

Программная инженерия

Пр 10
2015
ИН

Учредитель: Издательство "НОВЫЕ ТЕХНОЛОГИИ"

Издается с сентября 2010 г.

Редакционный совет

Садовничий В.А., акад. РАН
(председатель)
Бетелин В.Б., акад. РАН
Васильев В.Н., чл.-корр. РАН
Жижченко А.Б., акад. РАН
Макаров В.Л., акад. РАН
Панченко В.Я., акад. РАН
Стемпковский А.Л., акад. РАН
Ухлинов Л.М., д.т.н.
Федоров И.Б., акад. РАН
Четверушкин Б.Н., акад. РАН

Главный редактор

Васенин В.А., д.ф.-м.н., проф.

Редколлегия

Антонов Б.И.
Афонин С.А., к.ф.-м.н.
Бурдонов И.Б., д.ф.-м.н., проф.
Борзовс Ю., проф. (Латвия)
Гаврилов А.В., к.т.н.
Галатенко А.В., к.ф.-м.н.
Корнеев В.В., д.т.н., проф.
Костюхин К.А., к.ф.-м.н.
Липаев В.В., д.т.н., проф.
Махортов С.Д., д.ф.-м.н., доц.
Манцивода А.В., д.ф.-м.н., доц.
Назирова Р.Р., д.т.н., проф.
Нечаев В.В., д.т.н., проф.
Новиков Б.А., д.ф.-м.н., проф.
Павлов В.Л. (США)
Пальчунов Д.Е., д.ф.-м.н., доц.
Петренко А.К., д.ф.-м.н., проф.
Позднеев Б.М., д.т.н., проф.
Позин Б.А., д.т.н., проф.
Серебряков В.А., д.ф.-м.н., проф.
Сорокин А.В., к.т.н., доц.
Терехов А.Н., д.ф.-м.н., проф.
Филимонов Н.Б., д.т.н., проф.
Шапченко К.А., к.ф.-м.н.
Шундеев А.С., к.ф.-м.н.
Щур Л.Н., д.ф.-м.н., проф.
Язов Ю.К., д.т.н., проф.
Якобсон И., проф. (Швейцария)

Редакция

Лысенко А.В., Чугунова А.В.

Журнал издается при поддержке Отделения математических наук РАН, Отделения нанотехнологий и информационных технологий РАН, МГУ имени М.В. Ломоносова, МГТУ имени Н.Э. Баумана, ОАО "Концерн "Сириус"

СОДЕРЖАНИЕ

Кознов Д. В. Программная инженерия и визуальное моделирование: воспитание культуры работы с информацией	3
Мальцев А. В., Михайлюк М. В. Реализация эргономичного интерфейса управления виртуальной моделью антропоморфного робота с использованием устройства Kinect	12
Жаринов И. О., Жаринов О. О. Решение задачи оценки параметров математической модели цветовоспроизведения дисплея с максимальным цветовым охватом на основе интерполяции линии спектральных цветностей	19
Афонин С. А., Гаспарянц А. Э. Автоматическое построение функции оценки качества в задаче разрешения неоднозначности имен авторов научных публикаций	31
Киселёв Ю. А., Поршнева С. В., Мухин М. Ю. Метод извлечения родовидовых отношений между существительными из определений толковых словарей	38

Журнал зарегистрирован

в Федеральной службе

по надзору в сфере связи,

информационных технологий

и массовых коммуникаций.

Свидетельство о регистрации

ПИ № ФС77-38590 от 24 декабря 2009 г.

Журнал распространяется по подписке, которую можно оформить в любом почтовом отделении (индексы: по каталогу агентства "Роспечать" — 22765, по Объединенному каталогу "Пресса России" — 39795) или непосредственно в редакции.

Тел.: (499) 269-53-97. Факс: (499) 269-55-10.

Http://novtex.ru/pi.html E-mail: prin@novtex.ru

Журнал включен в систему Российского индекса научного цитирования.

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.

© Издательство "Новые технологии", "Программная инженерия", 2015

SOFTWARE ENGINEERING

PROGRAMMNAYA INGENERIA

№ 10

October

2015

Published since September 2010

Editorial Council:

SADOVNICHY V. A., Dr. Sci. (Phys.-Math.),
Acad. RAS (*Head*)
BETELIN V. B., Dr. Sci. (Phys.-Math.), Acad. RAS
VASIL'EV V. N., Dr. Sci. (Tech.), Cor.-Mem. RAS
ZHIZHCHEKNO A. B., Dr. Sci. (Phys.-Math.),
Acad. RAS
MAKAROV V. L., Dr. Sci. (Phys.-Math.), Acad.
RAS
PANCHENKO V. YA., Dr. Sci. (Phys.-Math.),
Acad. RAS
STEMPKOVSKY A. L., Dr. Sci. (Tech.), Acad. RAS
UKHLINOV L. M., Dr. Sci. (Tech.)
FEDOROV I. B., Dr. Sci. (Tech.), Acad. RAS
CHETVERTUSHKIN B. N., Dr. Sci. (Phys.-Math.),
Acad. RAS

Editor-in-Chief:

VASENIN V. A., Dr. Sci. (Phys.-Math.)

Editorial Board:

ANTONOV B.I.
AFONIN S.A., Cand. Sci. (Phys.-Math)
BURDONOV I.B., Dr. Sci. (Phys.-Math)
BORZOV JURIS, Dr. Sci. (Comp. Sci), Latvia
GALATENKO A.V., Cand. Sci. (Phys.-Math)
GAVRILOV A.V., Cand. Sci. (Tech)
JACOBSON IVAR, Dr. Sci. (Philos., Comp. Sci.),
Switzerland
KORNEEV V.V., Dr. Sci. (Tech)
KOSTYUKHIN K.A., Cand. Sci. (Phys.-Math)
LIPAEV V.V., Dr. Sci. (Tech)
MAKHORTOV S.D., Dr. Sci. (Phys.-Math)
MANCIVODA A.V., Dr. Sci. (Phys.-Math)
NAZIROV R.R., Dr. Sci. (Tech)
NECHAEV V.V., Cand. Sci. (Tech)
NOVIKOV B.A., Dr. Sci. (Phys.-Math)
PAVLOV V.L., USA
PAL'CHUNOV D.E., Dr. Sci. (Phys.-Math)
PETRENKO A.K., Dr. Sci. (Phys.-Math)
POZDNEEV B.M., Dr. Sci. (Tech)
POZIN B.A., Dr. Sci. (Tech)
SEREBRJKOV V.A., Dr. Sci. (Phys.-Math)
SOROKIN A.V., Cand. Sci. (Tech)
TEREKHOV A.N., Dr. Sci. (Phys.-Math)
FILIMONOV N.B., Dr. Sci. (Tech)
SHAPCHENKO K.A., Cand. Sci. (Phys.-Math)
SHUNDEEV A.S., Cand. Sci. (Phys.-Math)
SHCHUR L.N., Dr. Sci. (Phys.-Math)
YAZOV Yu. K., Dr. Sci. (Tech)

Editors: LYSENKO A.V., CHUGUNOVA A.V.

CONTENTS

Koznov D. V. Model-Based Development in Education: Teaching Way of Thinking	3
Maltsev A. V., Mikhayluk M. V. Implementation of Ergonomic Contactless Management Interface for Anthropomorphic Robot's Virtual Model by Using Kinect	12
Zharinov I. O., Zharinov O. O. The Solution to the Problem of Evaluation of Parameters of the Mathematical Model of Color Reproduction for Display with Maximal Gamut Based of Spectrum Locus Interpolation	19
Afonin S. A., Gasparianc A. E. Construction of Quality Function for Scientific Papers Author Names Disambiguation Problem Using Supervised Learning Techniques	31
Kiselev Yu. A., Porshnev S. V., Mukhin M. Yu. Method of Extracting Hyponym-Hypernym Relationships for Nouns from Definitions of Explanatory Dictionaries	38

Программная инженерия и визуальное моделирование: воспитание культуры работы с информацией

Обучение модельно-ориентированной инженерии является темой, которая активно обсуждается в международном сообществе. Однако до сих пор нет единого взгляда на этот вопрос. Различные курсы, посвященные преподаванию этого предмета, сильно отличаются друг от друга. В данной статье выдвинута концепция культуры визуального моделирования как умение видеть эффективные возможности применения модельно-ориентированной инженерии в программных проектах и создавать методики и решения, соответствующие потребностям конкретной ситуации. Представлены положения, детализирующие понятие культуры визуального моделирования, а также способы их изложения в курсе по модельно-ориентированной инженерии для бакалавров.

Ключевые слова: модельно-ориентированная инженерия, UML, программная инженерия

Введение

Модельно-ориентированная инженерия (Model Driven Engineering, MDE) — это подход к разработке программного обеспечения (ПО), использующий модели (другими словами — абстракции), которые более близки к предметной области, чем конструкции языков программирования и абстракции, используемые при разработке алгоритмов. Модели также помогают повысить совместимости различных программных систем, упростить процесс проектирования, повысить степень автоматизации процесса разработки, улучшить коммуникации между участниками и командами, работающими над созданием системы [1]. Под моделированием понимаем, главным образом, визуальное моделирование с помощью таких языков, как UML, SysML, BPMN и др., а также предметно-ориентированное моделирование¹ [2]. В настоящее время модельно-ориентированная инженерия из научных лабораторий и экспериментальных проектов активно проникает в индустрию, уже пройдя достаточно большой исторический путь. В 1950-е гг. появились блок-схемы, которые, однако, не нашли широкого распространения, несмотря на стандартизацию ANSI. В 1970—1980-е гг. возник структурный анализ, породивший активно используемые на практике методы. В 1990-е гг., в связи с развитием объектно-ориентированного подхода произошел методологический взрыв — было создано более 50 методологий анализа и проектирования ПО. В результате в конце 1990-х гг. появился универсальный

язык моделирования UML [3], а в 2000 г. вышел в свет стандарт OMG по модельно-ориентированной архитектуре — универсальному подходу к разработке ПО на основе моделей [4]. До настоящего времени UML выдержал более 10 версий, а в 2005 г. комитетом OMG был принят стандарт моделирования бизнес-процессов BPMN [5]. С начала 2000-х гг. и до настоящего времени продолжается активное исследование вопросов практического использования MDE в индустрии. Одно из последних и, по мнению автора, самых объективных исследований данного вопроса, представленное в работе [6], дает понять, что до полного успеха в этом вопросе еще далеко (70 % опрошенных ИТ-специалистов заявили, что не используют UML). В работе [7] указано, что большинство современных программистов получали образование в прошлые десятилетия, и их не учили MDE. Между тем, Тони Хоар еще в 1996 г. писал, что образование должно стать мостом для проникновения новаторских, исследовательских подходов в индустрию [8]. Таким образом, в настоящее время образование во многом ответственно за распространение MDE в индустрии.

Модельно-ориентированная инженерия и, в частности, UML очень популярны в университетских программах по информатике и программной инженерии, как, например, отмечено в работе [6]. Однако полного консенсуса о том, как именно следует преподавать MDE, еще нет. Так, в работе [9] приведены результаты исследования вопроса о том, какое место в различных образовательных стандартах и учебных программах занимает обучение моделированию. Представленные в работе [9] результаты противоречивы. С 2005 г. в рамках ведущей конференции по моделированию MODELS проводится Educational Symposium, на котором представляется и обсуждается опыт различ-

¹ Хотя в общем случае это и моделирование производительности систем, и формальные модели, используемые при верификации и модельно-ориентированном тестировании, и т. д.

ных педагогов по преподаванию MDE в университетах. Много работ по данной тематике публикуется в трудах различных ИТ-конференций, а также конференций, посвященных образованию. К настоящему времени издано несколько сотен научных работ, описывающих курсы, методы и подходы к обучению MDE в университетах². Поиск общих точек отсчета, единых концепций в этой области активно продолжается, а в качестве промежуточного итога такого поиска можно указать работы [7, 9]. Многие авторы указывают на то, что моделирование нужно преподавать как отдельную дисциплину, вне контекста тех или иных областей информатики и программной инженерии, а также курсов типа "Введение в программную инженерию", "Основы разработки архитектуры" и т. д. В качестве аргумента указывают, что варианты использования моделей при разработке ПО очень многообразны. Более того, согласно данным работы [14], преподавать моделирование нужно не "в глубину", а "в ширину", демонстрируя универсальность и широкую применимость модельно-ориентированной инженерии на практике.

В связи с отмеченным выше обстоятельством представляется важным *воспитание* у студентов *культуры моделирования* как способа работы с информацией. Здесь неслучайны и важны термины "*воспитание*" и "*культура*". Воспитание отличается от освоения и тренировки определенных навыков тем, что оно посвящено выработке более абстрактных профессиональных основ. Многие авторы (см., например, работу [15]) указывают на то, что при обучении программной инженерии нужно больше тратить времени и сил на воспитание принципов инженерии, чем на изучение конкретных прикладных технологий. Наравне с обучением принципам в процессе обучения можно также воспитывать определенную культуру, которая также (а может быть, даже более) абстрактна и включает в себя совокупность этих принципов. Такой подход позволяет не просто следовать при разработке ПО текущей прагматичной целесообразности, но создавать зрелые решения и продукты. *Культура визуального моделирования* основывается на умении видеть эффективные возможности применения модельно-ориентированной инженерии в программных проектах, а также на владении определенными навыками обобщения и систематизации информации (в виде визуальных моделей) и умении создавать "работающие", реально полезные диаграммы, облегчающие работу с информацией и повышающие ее целостность и непротиворечивость.

В данной работе описан курс по модельно-ориентированной инженерии для бакалавров, ориентированный на воспитание у студентов культуры визуального моделирования. Этот курс автор в течение 15 лет читает на математико-механическом факультете Санкт-Петербургского государственного университета.

² К сожалению, в России данный вопрос — т. е. проблемы и находки в области преподавания UML и MDE — освещен в научной литературе слабо: автору удалось найти лишь работы [10, 11]. С отечественными учебными пособиями в этой области дела обстоят лучше [12, 13, 32].

1. Культура визуального моделирования

Обучение модельно-ориентированной инженерии часто связывают с воспитанием у студентов абстрактного мышления³. При этом, согласно работам [15, 16], навыки абстрактного мышления относятся к ключевым компетенциями ИТ-специалистов⁴. Определение абстрактного мышления восходит еще к Аристотелю, который считал его важнейшей частью процесса научного познания. Однако при использовании абстрактного мышления в практической деятельности оказывается важным не только оперировать абстрактными понятиями и категориями, но также уметь поддерживать их связь с реальностью [16]: уметь выделять абстрактные концепции в рассматриваемой предметной области, совершать над ними, в их терминах, необходимые построения и снова "возвращать" их в предметную область, создавая окончательные продукты, хорошо служащие людям и бизнесу. Причем, по опыту автора, здесь важны все три фазы, и трудности в реализации какой-нибудь одной из них приводят к неукладке практического использования абстрактного мышления. По мнению автора, базовое развитие абстрактного мышления происходит путем изучения математики, философии и некоторых других наук. Во многом именно поэтому математика признана базовой наукой, изучение которой необходимо при подготовке ИТ-специалистов [15]. При этом, как указывалось в работе [18], речь идет не об использовании математических знаний *as is*, а именно о воспитании навыка оперирования абстракциями. Представляется, что для тех учебных ИТ-программ, которые включают в себя хорошую математическую подготовку (как, например, на математико-механическом факультете СПбГУ), преподавание моделирования не может ничего существенно добавить к воспитанию базового абстрактного мышления, и предлагаемые модельные формализмы типа UML и BPMN выглядят для студентов достаточно простыми и поэтому неинтересными⁵. Но моделирование может добавить кое-что для связей абстрактного мышления и реального мира, как указано в работе [16].

³ Интересная двойственность по отношению к моделированию в Software Engineering Curriculum [15] подмечена в работе [9]: с одной стороны, моделирование признается базовой концепцией в ИТ-образовании, с другой стороны, его изучение откладывается на завершающие стадии образования как требующее многих дополнительных знаний и навыков. Кажется, что в первом случае под моделированием в большей степени имелось в виду абстрактное мышление, в то время как во втором случае под моделированием понимают практики и технологии MDE, многие из которых действительно требуют обширной предваряющей образованности (например, метамоделирование и средства EMF/GMF).

⁴ Но этого недостаточно для успешного ИТ-специалиста. Важны также коммуникативность, широта профессионального кругозора и многое другое. Вообще, по мнению автора, сведение сложной проблемы к какому-то одному аспекту является упрощением.

⁵ Однако в иных случаях, например в бизнес-образовании, роль моделирования в воспитании абстрактного мышления может быть существенно иной.

Хотелось бы также заметить, что моделирование, в частности, визуальное, не является обязательным атрибутом инженерного творческого мышления и, соответственно, способом обучения навыкам такого мышления. Еще в 1970-х гг. эту точку зрения очень ясно аргументировал Д. Джонс, говоря о чертежном проектировании и инженерных решениях в широком спектре областей. Он писал, что рождение талантливых решений — это таинство, неформализуемый процесс, что такие решения появляются "на обратной стороне конверта" и лишь оформляются с помощью визуальных моделей, что позволяет снять "вторичные противоречия" и сделать найденные решения доступными для восприятия других⁶ [17].

Итак, обучение визуальному моделированию может помочь *применить* навыки абстрактного мышления к разработке ПО, а также развить и повысить коммуникативные способности студентов, приучая их лаконично и наглядно формулировать свои идеи, а также формализовывать информацию, необходимую для передачи коллегам и различным вовлеченным в проект сторонам.

Далее описано, как культура визуального моделирования конкретизируется в рамках представленного в работе учебного курса.

• **Уверенное практическое владение стандартными языками моделирования** — прежде всего UML и BPMN. Предполагается знание основных (но далеко не всех) конструкций этих языков и умение ими свободно пользоваться, "в идеале" — умение моделировать с их помощью любую предметную область. Такая свобода владения языками моделирования близка к педагогическим идеям В. Павлова [19] — в своих UML-тренингах он предлагал студентам моделировать не только ПО, но и произвольные предметные области, настаивая именно на свободе владения UML. Подчеркнем, что здесь речь идет именно о практическом овладении языками моделирования, а не об их формальном изучении. При этом важно обучать студентов использовать визуальные языки максимально строго, несмотря на неоднозначность и многовариантность UML и BPMN. В противном случае начинает происходить очень вольное использование языка, нарушающее все правила (автор неоднократно сталкивался с этим в индустрии). Только после овладения стандартными языками можно говорить об их расширении/изменении, а также о создании новых предметно-ориентированных языков. Сначала нужно уверенное владение базовыми средствами.

• **Концепция множественности точек зрения** на ПО в процессе его разработки. Многие студенты, привыкая использовать языки программирования, также привыкают к тому, что какая-либо информация должна быть определена точно, непротиворечиво и однозначно: вот код, который описывает данную функцию, он максимально собран в одном

⁶ При этом следует отличать действительно новые, творческие решения от решений, составленных из частей других, известных решений, — деятельность по разработке последних может быть высокопрофессиональной, но настоящим творчеством не является.

месте, и это приветствуется с точки зрения упрощения сопровождения. Вместе с тем, например, тот факт, что об одном и том же предмете можно писать в документе несколько раз, с разным уровнем детализации, воспринимается с трудом. Это очень заметно в студенческих письменных работах (курсовых и дипломных). Например, у студентов плохо получается коснуться разных вопросов во введении и потом развернуть их детально в основном тексте. Часто они начинают их подробно описывать сразу во введении и потом уже не упоминают в основном тексте, аргументируя это тем, что один раз они об этом уже писали. Однако повышение управляемости и коммуникативности информационного пространства проекта — это всегда избыточность, и она опирается на множественность точек зрения, которые необходимо учесть и соответствующим образом представить⁷. Нежелание программистов порождать что-либо, кроме кода, комментариев к нему и очень лаконичных документов (лучше всего — автоматически сгенерированных по коду и его комментариям), — большой недостаток нашей индустрии. Обучение моделированию может помочь в его устранении.

• **Умение видеть потенциальные модели в потоке информации о системе**, которые при этом максимально полно характеризуют систему или какой-то ее фрагмент. То есть важно воспитывать у студентов навыки извлечения из информации ее лаконичной сути и визуального представления этой сути. При этом, однако, визуальные модели не всегда могут подойти как средство описания получившейся абстракции, и это обстоятельство тоже нужно принимать во внимание. Если речь идет о структурных диаграммах UML, то в моделируемой предметной области должны быть сущности и связи (например, без связей модель выглядит как простое перечисление, т. е. оказывается вырожденной). Если речь идет, например, о диаграммах состояний и переходов UML, то они применимы в том случае, когда моделируемая система активно взаимодействует с внешним окружением (а не просто получает данные, преобразует их и передает дальше). При этом в разных ситуациях система по-разному обрабатывает одни и те же внешние воздействия (это упрощенное и неформальное определение реактивной системы [20]). Видение потенциальных моделей в потоке информации важно при неформальном моделировании, которое, согласно исследованиям [6], в настоящее время составляет основной способ применения UML. Это также важно при принятии решения использовать в проекте те или иные визуальные модели формальным способом, например, при моделировании телекоммуникационной системы, генерируя целевой код по моделям.

• **Искусство создания базовых моделей** системы, т. е. диаграмм, которые вешают на стену в главном кабинете команды проекта и показывают всем задействованным в проекте лицам (начальству, продавцам и пр.), с которых начинают знакомить с системой но-

⁷ Здесь автор объединяет декомпозицию и множественность точек зрения, поскольку это очень близкие подходы к работе с информацией.

вичков, указывая на которые можно много и долго рассказывать про разрабатываемую систему. Это могут быть как формальные, так и неформальные модели.

- **Умение выдерживать уровень абстракции и терминологию на диаграмме** — например, сущности, обозначаемые с помощью конструкции "Класс" на одной диаграмме, должны быть одного уровня абстракции. То же касается и связей. То есть модели, как правило, отражают некоторую классификацию предметной области, и она должна быть максимально однородной. При этом важно использовать согласованную терминологию, не называя, например, сотрудника кафедры то преподавателем, то сотрудником.

- **Навыки чтения диаграмм** необходимо развивать, следуя работам [19, 21], поскольку студенты должны получить собственный опыт того, сколь понятны/непонятны те или иные модели. При этом студенты смотрят на моделирование как бы с другой стороны и лучше видят свои ошибки. Важно обращать внимание студентов на то, чтобы они, создавая свои диаграммы, адресовали их конкретным читателям, а не создавали их вообще. Как правило, эти читатели или известны поименно, или можно их охарактеризовать как определенную целевую группу, имеющую определенный уровень подготовки, цели относительно данной информации и пр.

- **Умение создавать красивые диаграммы** — здесь имеется в виду движение в сторону обретения профессиональных художественно-дизайнерских навыков. Автор не склонен считать, что использование программных средств поддержки UML и других специализированных программных инструментов моделирования способно полностью решить эту задачу, хотя и не отрицает их пользы. На практике часто появляются небольшие, но важные диаграммы (см. выше про базовые модели), которые целесообразно нарисовать красиво, так как они предназначаются для многочисленных читателей. При этом если используются алгоритмы автоматических раскладок, то результаты их работы часто бывает полезно подправить. Красивые диаграммы повышают качество презентаций и документов, способствуют формированию позитивного эмоционального фона, облегчают общение и взаимодействие, а также восприятие сложной информации различными вовлеченными в проект сторонами.

- **Управляемость привнесения в модели собственных идей.** Здесь речь идет о том, что при моделировании часто оказывается важным формализовать уже имеющуюся информацию, например первичные понятия и представления заказчика при разработке требований, несмотря на их несовершенство. Чтобы, например, убедиться, что исполнители его (заказчика) правильно понимают, а уже потом предлагать собственные идеи и усовершенствования. Студенты очень часто игнорируют имеющуюся информацию, сразу же заменяя ее своими представлениями о том, как должно быть, и это происходит не только в рамках данного курса. Если эти тенденции заходят у работника на производстве очень далеко, то с ним оказывается непросто найти общий язык — ему трудно передать формулировку задачи, особенно если она сложная, он постоянно делает что-то свое. Отрабатывать навык восприимчивости (особенно

у талантливых студентов) можно при обучении моделированию, предлагая студентам точно выполнять небольшие задания. Еще раз подчеркнем, что речь не идет о том, что собственные идеи и соображения не нужны, но лишь о том, что нужно уметь управлять процессом их привнесения в работу.

- **О нецелесообразности создания очень больших диаграмм** — "если ты создал большую диаграмму классов, то это не значит, что ты много работал, а значит, что ты не владеешь культурой моделирования" [14]. Информацию, представленную на таких диаграммах, часто трудно воспринимать. Использование программных пакетов для работы с диаграммами отчасти устраняет эту трудность. Однако следует приучать студентов стремиться создавать диаграммы, уместающиеся на листе формата А4, и не перегружать их деталями. Здесь важным моментом оказывается также декомпозиция сложных моделей.

- **Введение в генерацию кода по моделям.** Многие исследователи видят основную задачу модельно-ориентированной инженерии в том, чтобы заменить языки программирования диаграммными языками. Студентам важно объяснить выгоды от решения этой задачи (прежде всего — это повышение уровня абстрактности средств разработки), а также показать ограничения, которые не дают ее решить (обсуждение этого вопроса см., например, в работе [32]). Целесообразно показать текущее состояние реализации данной идеи на следующих примерах: генерация описания классов Java/C# по диаграммам классов, генерация схемы базы данных по диаграммам классов, генерация поведенческих спецификаций по диаграммам конечных автоматов UML для телекоммуникационных систем и аппаратных систем. Причем это именно введение, демонстрация с примерами из реальной индустрии и минимальная практика. Серьезное практическое освоение этой темы, как указано в работе [23], требует значительной предварительной подготовки студентов.

- **Области и способы практического использования моделирования** — как непосредственно в разработке ПО, так и в смежных областях: при анализе предметной области; при разработке требований и архитектуры (включая образцы проектирования); в процессе подготовки документации и презентационных материалов; при создании корпоративных ИТ-архитектур для больших компаний; в бизнес-моделировании (бизнес-процессы и другие бизнес-модели). Тут необходимо показать важность организационного и бизнес-контекста разработки ПО и обозначить ту помощь, которую может здесь оказать визуальное моделирование [24].

Автору представляется, что обучение этим положениям способно сформировать у студентов основу для использования MDE на практике. Кроме того, такое обучение способствует формированию у студентов инженерной культуры, что может принести пользу даже в том случае, если они не будут использовать модельно-ориентированную инженерию. Заметим, что о возможности такого эффекта от изучения модельно-ориентированной инженерии было сказано в работе [6].

2. Эволюция курса

Курсы по программной инженерии, в отличие от курсов начертательной геометрии, математического анализа и т. д., часто меняются, что соответствует быстрому обновлению тенденций и технологий в ИТ-индустрии⁸. При этом кажется целесообразным выполнять ретроспективу эволюции учебных курсов в целях более качественного управления изменениями курсов впоследствии [25]. Эта информация может также помочь другим преподавателям улучшать свои курсы.

У представленного в данной работе курса было три описанных ниже этапа эволюции.

♦ 2000—2003 гг. Курс был посвящен объектно-ориентированному анализу и проектированию и следовал книге Г. Буча [27]. Курс посещало много студентов — объектно-ориентированные методы разработки ПО и недавно появившийся UML были модной темой, которая интересовала многих.

♦ 2003—2008 гг. Курс включил в себя различные аспекты визуального моделирования: историю предмета; языки и программные средства; описание типовой архитектуры CASE-средств; концепцию множественности точек зрения; обзор UML; тему генерации кода по моделям (моделирование с помощью UML баз данных и телекоммуникационных систем); визуальное моделирование бизнес-процессов; визуальное моделирование при разработке семейств программных продуктов; введение в семиотику и строение визуальных языков; метамоделирование; основы предметно-ориентированного визуального моделирования. Данный материал основывался на исследовательских проектах в области визуального моделирования, в которых автор в той или иной степени принимал участие — RTST [28], Real и Real-IT [29], QReal [30, 31]. Курс был оформлен как учебное пособие [32]. Однако автор получил негативную обратную связь от студентов — популярность курса падала год от года. Коллеги высказались в том смысле, что учебное пособие в большей степени является научной монографией. Вообще говоря, университетские курсы нередко посвящают обучению "экзотическим" методам и средствам, соответствующим научным интересам преподавателей, но имеющим незначительную практическую ценность.

♦ 2008—2014 гг. В эти годы автор существенно переработал курс, убрав из него исследовательские темы и ориентируя его на обучение практическим навыкам и культуре моделирования. В результате популярность курса стала расти — все больше студентов стали записываться на курс, регулярно его посещали и более качественно работали.

⁸ Нужно также упомянуть в целом о молодости программной инженерии и информатики по сравнению с такими классическими научными дисциплинами, как математика, биология, физика и т. д. При этом велика инерция разработки зрелых учебных программ, например, программа обучения математике на математико-механическом факультете СПбГУ складывалась и формировалась в течение последних 100 лет.

3. Общее описание курса

Курс проводится для бакалавров третьего года, обучающихся по специальности "Программная инженерия". Объем курса составляет 16 пар (32 академических часа), курс проводится по одной паре в неделю в течение одного семестра, число обучаемых — 10...16 человек. Предварительно студенты уже прошли курсы по программированию (Java/C#), математике, информатике (теория алгоритмов, формальные грамматики, математическая логика), а также — по базам данных. Особо отметим, что данный курс посвящен именно модельно-ориентированной инженерии — не проектированию ПО, но общим вопросам программной инженерии и пр. хотя и затрагивает многочисленные вопросы разработки ПО, в том числе и вопросы проектирования.

Подобно тому, как это принято при обучении программной инженерии в западных университетах (см., например, работу [22]), данный курс не является классическим лекционным, а также не подразумевает строго разделения на лекционную и семинарскую части⁹. Курс включает ряд лекций, но они непосредственно связаны с практическими заданиями и нужны для того, чтобы обеспечить студентов необходимой для выполнения заданий информацией. Исключение составляют вводные лекции, а также демонстрационные части курса, дающие представление студентам об областях, куда оказывается трудно "дотянуться" практическими заданиями (например, кодгенерация по моделям).

В рамках курса преподаются следующие языки моделирования: UML; BPMN; IDEF0. Однако эти языки преподаются не целиком в силу следующих причин. Во-первых, все они вместе оказываются весьма значительным объемом информации, который существенно превышает возможности одного семестрового курса (одна только спецификация UML занимает около 1000 страниц). Во-вторых, относительно UML и BPMN можно заметить, что на практике активно используется лишь подмножество этих языков — для UML, например, согласно работе [33], это диаграммы вариантов использования, классов и сценариев. В-третьих, согласно работе [14], преподавание визуальных языков в максимальном объеме является плохой практикой. В рамках курса были выбраны следующие подмножества этих языков: UML — диаграммы классов, объектов, компонентов, вариантов использования, конечных автоматов (*statecharts*); BPMN — участники (*participants*), процессы, декомпозиция процессов; IDEF0 — контекстные диаграммы.

В качестве программного средства в курсе используется пакет Microsoft Visio. Данный инструментарий широко распространен, легок в освоении, поддерживает большое количество визуальных нотаций, в частности все, используемые в рамках данного курса. Кроме того, распространяемая компанией Microsoft в последние годы программа DreamSpark позволяет вузам России получать этот (и другие про-

⁹ Как это принято в российских университетах, когда профессор читает лекции, а ассистент или лаборант ведет соответствующие семинарские занятия.

дукты компании) в целях обучения практически бесплатно, и функциональные возможности этой версии Visio вполне достаточны для целей данного курса. Наличие инструмента дисциплинирует студентов и облегчает их взаимодействие с преподавателем, в том числе — через сеть Интернет. Студенты не высылают сфотографированные листки с нарисованными от руки диаграммами, не пользуются произвольными программными средствами и, соответственно, произвольными форматами файлов. Кроме того, Microsoft Visio позволяет легко интегрировать разные нотации в один Visio-файл — имеется возможность открывать в одном файле несколько палитр, а также разбивать диаграммы на страницы, при необходимости связывая с элементами диаграмм другие страницы через гиперссылки и, таким образом, выполнять декомпозицию модели "в глубину". Наконец, с помощью Microsoft Visio можно обрабатывать тему с дизайном диаграмм, меняя визуальные атрибуты фигур, линий и надписей. Использование более профессиональных и, соответственно, сложных средств поддержки визуального моделирования усложнило бы курс (с дискуссией об использовании профессиональных средств моделирования в рамках учебных курсов по MDE можно ознакомиться в работе [21]).

Все выполненные в течение курса домашние задания студенты собирают в специальные "альбомы", которые представляют на экзамене. Альбом является Visio-файлом, в котором каждое задание выполнено на отдельной странице (*page*). Альбомы являются важным "собирающим" механизмом при освоении практических навыков. Сами студенты при этом имеют возможность вернуться к выполненным заданиям, что-то вспомнить, что-то уточнить, и преподаватель видит контекст очередных выполненных заданий студента.

Следуя статье [22], в предлагаемом курсе в качестве предметной области для упражнений используется университет — данная предметная область хорошо знакома студентам, и вместе с тем она является довольно содержательной.

Курс состоит из установочных лекций, основной части и групповых проектов.

- В рамках установочной части вначале предлагается введение в визуальное моделирование, а также настраивается порядок работы в классе/дома.

- Основная часть посвящена практическому освоению моделирования и устроена в виде перечисленных далее циклов. На занятии преподаватель дает лекцию по определенной теме и приводит примеры. После этого студентам выдают соответствующие упражнения для выполнения в аудитории, которые преподаватель сразу же проверяет и комментирует. Затем студентам выдают домашние задания. Результаты выполнения заданий студенты высылают преподавателю по электронной почте накануне следующего занятия (дата и контрольное время строго оговорены), преподаватель их распечатывает, проверяет и выдает с комментариями студентам на занятии. Особо интересные или трудные места разбирают в классе. Наличие проработанной системы заданий по принципу от простого к сложному повышает степень включения студентов в учебный процесс. У них хорошо получает-

ся выполнять простые задания, и это мотивирует их к выполнению более сложных. Кроме того, студенты постоянно работают сами, а не только слушают и кивают, что также благотворно сказывается как на их успеваемости, так и на атмосфере курса.

- В конце курса проводится небольшая серия групповых проектов, в рамках которых студенты закрепляют полученные навыки, выполняя более сложные практические задания.

5. Структура курса

Установочная часть курса включает основные определения, дается объяснение целей и задач визуального моделирования при разработке ПО, кратко описывается история развития данной области. Далее обсуждается основной вопрос визуального моделирования — невидимость ПО. Рассказывается о концепции множественности точек зрения при разработке ПО. Рассматривается роль визуального моделирования в установке и реализации внутренних и внешних коммуникаций проекта. Рассказывается о принципах объектно-ориентированного подхода к разработке ПО (по Г. Бучу).

Основная часть включает в себя следующие темы: анализ; проектирование; моделирование архитектуры предприятия. Рассмотрим эти темы более детально.

Анализ

- Анализ предметной области: выделение сущностей (объектов) и связей (диаграммы классов) с помощью диаграмм классов и объектов UML. В рамках этой темы студенты: осваивают принципы построения иерархий а part of (агрегирование) и as is (наследование); привыкают рассматривать визуальное моделирование как использование формального языка (в диаграммах классов содержится много полезных особенностей, например, множественность у ассоциаций, имена ролей: квалификаторы наследования); учатся точно излагать с помощью визуального языка существующую информацию без собственных интерпретаций (выполняя хорошо спроектированные небольшие упражнения). Также в рамках этой темы уделяется внимание дизайну диаграмм.

- Анализ предметной области: моделирование бизнес-процессов (BPMN). Фокус бизнес-процесса, выбор необходимых активностей для описания бизнес-процесса (предметная область "Университет").

- Контекст ПО: контекстные диаграммы IDEF0 для кристаллизации контекста программной системы или ее отдельной подсистемы в терминах вход/выход/управление/ресурсы, диаграммы вариантов использования для связи программных систем с потребностями будущих пользователей и выделения именно той верхнеуровневой функциональности, которая добавляет пользователям эффективности [26]. Кроме того, контекстные диаграммы IDEF0 оказываются удобными для первого знакомства с концепцией множественности точек зрения (например, можно рассмотреть университет как научно-образовательную систему, а можно — как финансовую). Кроме того, здесь студенты учатся выбирать подходящие названия для элементов модели. Для составления вари-

антов использования активно применяют различные сайты. Среди них как хорошо структурированные (например, сайты банков и различных интернет-магазинов), так и плохо структурированные (например, сайты некоторых университетов и компаний).

Проектирование

- Верхнеуровневое проектирование ПО с помощью диаграмм компонентов UML. Данная тема становится очевидна студентам только в том случае, если продемонстрировать ее на реальных проектах, а не с помощью искусственных упражнений. В качестве упражнений предлагается перерисовывать с помощью этих диаграмм верхнеуровневые схемы архитектуры различных существующих систем (у автора есть большая библиотека соответствующих документов). Предметная область каждого упражнения оказывается несложной, и само задание тоже не является сложным — всего только перерисовать. Однако потом рассматривают все диаграммы вместе — и исходные, и получившиеся, и студенты видят явную выгоду, полученную от унифицированного представления данной информации.

- Проектирование баз данных и объектно-ориентированных приложений. В рамках этой темы в курсе на примере диаграмм классов рассказывают о кодогенерации по моделям. Студентам показывают, как с помощью средств Visual Studio можно автоматически получать программный код на основе диаграмм (однако в качестве средства для домашних заданий Visual Studio не используется ввиду недостатка времени). При моделировании схем баз данных на примере концептуальных, логических и физических моделей данных еще раз рассматривают концепцию множественности точек зрения на ПО. На эту тему студенты выполняют ряд упражнений, пользуясь Microsoft Visio.

- Образцы (*patterns*): фабрика, фасад, прокси, адаптер (диаграммы классов). Поскольку данный курс не фокусируется только на проектировании ПО (курсы по проектированию студентам преподают в дальнейшем), то был выбран небольшой набор самых распространенных, по мнению автора, шаблонов проектирования с тем, чтобы продемонстрировать выразительную силу UML.

- Реактивные системы и телекоммуникационные алгоритмы (диаграммы состояний и переходов, диаграммы сценариев). Разработка диаграмм состояний и переходов вырабатывает отношение к визуальному моделированию как процедуре создания формальных спецификаций — последние имеют хорошо определенный синтаксис и ясную исполняемую семантику. Кроме того, наравне со средствами спецификации бизнес-процессов они являются самой распространенной нотацией для спецификации поведения систем.

- Метамоделирование (диаграммы классов UML), т. е. спецификация абстрактного синтаксиса небольших предметно-ориентированных языков моделирования, а также задание с помощью диаграмм классов небольших фрагментов языков программирования. Данная тема важна для развития у студентов навыков разработки метаструктур — схем баз данных, схем XML-документов, создания грамматик

для формальных языков и т. д. Практические задания для этой части курса отталкиваются от теории грамматик: студентам предлагается создавать грамматики в форме Бэкуса-Науэра для простых языков, а потом создавать для них метамодели.

Моделирование архитектуры предприятия (Enterprise Architecture Management, EAM)

Данная область бурно развивается в настоящее время [34], и основной ее инструментарий — это именно визуальное моделирование. При этом разработка корпоративных ИТ-архитектур для больших компаний очень близко подходит к разработке ПО как такового. Во многих компаниях в настоящее время требуются аналитики, знакомые с основами EAM. В рамках этой темы дают введение в EAM, рассказывают о бизнес-архитектуре и ИТ-архитектуре, приводят примеры. Данная тема не подкрепляется упражнениями и является иллюстративной.

Для выполнения групповых проектов студенты объединяются в группы по 3...5 человек. Для каждой группы выдают одно задание из предметной области "Университет", содержащее краткое описание предметной области и верхнеуровневые требования к программной системе. Каждая группа должна выполнить анализ и верхнеуровневое проектирование своей системы. Только одна группа знает свое задание, остальные — нет. Получившиеся результаты в виде диаграмм отдают для изучения соседним группам с тем, чтобы они попытались восстановить задание другой группы по диаграммам (это так называемый метод обратной трассировки В. Павлова [19]). Таким образом, студенты, в числе прочего, учатся читать визуальные спецификации.

Заключение

Предложены и подробно описаны концептуальные положения, определяющие культуру визуального моделирования как объекта преподавания в рамках обучения модельно-ориентированной инженерии. В работе представлен курс, реализующий эту концепцию для бакалавров по программной инженерии. Важно, что данный курс посвящен именно MDE, а не конкретной области разработки ПО — проектированию, анализу и т. д. Курс также не фокусируется на "экзотических" приемах и концепциях, например на модельно-ориентированном подходе с применением генерации кода по диаграммам. Его задача — воспитание культуры визуального моделирования и демонстрация применимости этой культуры в широком спектре прикладных задач. Следует также отметить, что данная статья ставит своей целью закрыть брешь и подтолкнуть преподавателей ИТ в России активнее делиться своими наработками — идеями, опытом преподавания и пр. — в области MDE. В качестве дальнейшего направления исследований можно обозначить интеграцию опыта преподавания модельно-ориентированного подхода программистам и преподавание модельных и визуальных техник при работе со знаниями [35, 36], а также распространение опыта преподавания визуального моделирования из области обучения программной инженерии в область обучения бизнес-информатики.

Список литературы

1. **Broy M.** Seamless Method- and Model-based Software and Systems Engineering / ed. S. Nanz // The Future of Software Engineering. Berlin Heidelberg: Springer-Verlag, 2011. P. 33–49.
2. **Kelly S., Tolvanen J.-P.** Domain-Specific Modeling: Enabling Full Code Generation. John Wiley&Sons, 2008. 340 p.
3. **OMG** Unified Modeling Language (OMG UML), Version 2.5, 2013. 786 p.
4. **MDA** Guide Version 1.0.1. omg/2003-06-01. June 2003. 62 p.
5. **Business Process Model and Notation (BPMN)**. Version 2.0. January 2011. OMG. 538 p.
6. **Petre M.** UML in Practice // Proc. of 35th International Conference on Software Engineering (ISCE). May 18–26, 2013, San Francisco, CA. IEEE Press, 2013. P. 722–731.
7. **Bezivin J., France R., Gogolla M.** et al. Teaching Modeling: Why, When, What? // MODELS'09: Proceedings of the 2009 International Conference on Models in Software Engineering. Berlin Heidelberg: Springer-Verlag, 2010. P. 55–62.
8. **Hoare C. A. R.** The Role of Formal Techniques: Past, Current and Future or How Did Software Get so Reliable without Proof? // Proceedings 18th International Conference on Software Engineering, Berlin, Germany, March 25–29, 1996. IEEE Computer Society, 1996. P. 233–234.
9. **Börstler J., Kuzniarz L., Alphonse C.** et al. Teaching software modeling in computing curricula // Proceedings of the final reports on Innovation and technology in computer science education 2012 (ITiCSE-WGR'12). NY, USA: ACM New York, 2012. P. 39–50.
10. **Иванов Д. Ю., Новиков Ф. А.** Моделирование на UML: опыт преподавания в Интернет // Компьютерные инструменты в образовании. 2009. № 4. С. 53–61.
11. **Тихомиров В. Н.** Обучение студентов методам графического программирования на примере ДРАКОН и UML // Сборник науч. трудов Московского машиностроительного университета. М.: МГОУ, 2014. С. 95–102.
12. **Иванов Д. Ю., Новиков Ф. А.** Унифицированный язык моделирования: учеб. пособие. СПб.: Изд-во гос. политехнического ун-та, 2010. Сер. Приоритетные национальные проекты "Образование".
13. **Леоненков А. В.** Самоучитель UML 2. СПб.: БХВ-Петербург, 2007. 576 с.
14. **Paige R. F., Polack F. A. C., Kolovos D. S.** et al. Bad Modelling Teaching Practices // Proceedings of the MODELS Educators Symposium co-located with the ACM/IEEE 17th International Conference on Model Driven Engineering Languages and Systems (MODELS 2014), Valencia, Spain, September 29, 2014. CEUR Workshop Proceedings, 2014. Vol. 1346. P. 1–12. URL: <http://ceur-ws.org/Vol-1346/>
15. **Joint ACM/IEEE Task Force on Computing Curricula.** Software engineering 2004: Curriculum guidelines for undergraduate degree programs in software engineering. Technical report, IEEE CS and ACM, 2004.
16. **Kramer J.** Is abstraction the key to computing? // Communications of the ACM. 2007. Vol. 50, N 4. P. 36–42.
17. **Джонс Дж. К.** Инженерное и художественное конструирование. Пер. с англ. М.: Мир, 1976. 374 с.
18. **Devlin K.** Why universities require computer science students to take math // Communication of ACM. 2003. Vol. 46, N 9. P. 37–39.
19. **Pavlov V. L., Yatsenko A.** "The Babel Experiment": An Advanced Pantomime-based Training in OOA&OOD with UML // Proceedings of the 36th SIGCSE technical symposium on Computer science education. New York: USA: ACM, 2005. P. 231–235.
20. **Harel D., Politi M.** Modeling Reactive Systems with Statecharts: state machine approach. McGraw-Hill, 1998, 258 p.
21. **Moisan S., Rigault J.-P.** Teaching object-oriented modeling and UML to various audiences // MODELS'09: Proceedings of the 2009 international conference on Models in Software Engineering. LNCS. 2010. Vol. 6002. P. 40–54.
22. **Burton P. J., Bruhn R. E.** Using UML to facilitate the teaching of object-oriented systems analysis and design // Journal of Computing Sciences in Colleges 2004. Vol. 19, N 3. P. 278–290.
23. **Flint S., Gardner H., Boughton C.** Executable/Translatable UML in computing education // Proceedings of the Sixth Australasian Conference on Computing Education (ACE'04). Darlinghurst, Australia: Australian Computer Society, Inc., 2004. P. 69–75.
24. **Bolognesi A., Ciancarini P., Moretti R.** On the Education of Future Software Engineering // In Software Engineering Education in the Modern Age. LNCS. 2006. Vol. 4309. P. 186–205.
25. **Lengyel L., Mezei G.** Model-driven paradigms: the evolution of a university course // Proc. of the 8th edition of the Educators' Symposium (EduSymp '12), 2012. P. 13–20.
26. **Booch G., Jacobson I., Rumbaugh J.** The Unified Software Development Process. Prentice Hall, 1999. 512 p.
27. **Booch G.** Object-Oriented Analysis And Design With Application, second edition. The Benjamin/Cummings Publishing Company, Inc., 1994. 589 p.
28. **Терехов А. Н.** RTST-технология программирования встроенных систем реального времени // Записки семинара Кафедры системного программирования "Case-средства RTST++". СПб.: Изд-во СПбГУ, 1998. С. 3–17.
29. **Терехов А., Долгов П., Иванов А.** и др. Объектно-ориентированное расширение технологии RTST // Записки семинара Кафедры системного программирования "Case-средства RTST++". СПб.: Изд-во СПбГУ, 1998. С. 17–37.
30. **Литвинов Ю. В.** Применение DSM-платформы QREAL при разработке среды программирования роботов QREAL:ROBOTS // Системное программирование. 2012. Т. 7, № 1. С. 161–186.
31. **Терехов А. Н., Брыксин Т. А., Литвинов Ю. В.** QREAL: платформа визуального предметно-ориентированного моделирования // Программная инженерия. 2013. № 6. С. 11–19.
32. **Кознов Д. В.** Основы визуального моделирования: учеб. пособие. М.: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2008. 246 с.
33. **Grossman M., Aronson J. E., McCarthy R. V.** Does UML make the grade? Insights from the software development community // Information and Software Technology. 2005. Vol. 47, N 6. P. 383–397.
34. **Кудрявцев Д. В., Арзумян М. Ю., Григорьев Л. Ю.** Технологии бизнес-инжиниринга. СПб.: Изд-во СПбГПУ, 2014. 427 с.
35. **Гаврилова Т. А., Лещева И. А., Страхович Э. В.** Об использовании визуальных концептуальных моделей в преподавании // Вестник Санкт-Петербургского университета. Серия 8: Менеджмент. 2011. № 4. С. 124–150.
36. **Гаврилова Т. А., Лещева И. А., Лещев Д. В.** Использование онтологий в качестве дидактического средства // Искусственный интеллект. 2000. № 3. С. 34–39.

D. V. Koznov, Associate Professor, e-mail: d.koznov@spbu.ru, Saint Petersburg State University

Model-Based Development in Education: Teaching Way of Thinking

Model-based software engineering teaching is actively discussed in the international education community at the moment. Nonetheless, there is no common viewpoint to the topic. Courses at different universities are radically different from each other. The variation points are focus and area of the teaching topic, student background required, connections with other software engineering courses, etc. In this paper an education concept for model based software engineering teaching is introduced. The concept is more a special way of thinking rather than a collection of techniques/tools. It consists of the following issues: background in standard modeling languages (UML, BPMN);

view point concept; skills for detection of information pieces, which suit/not suit for modeling; creating nice diagrams; balancing information on a diagram; diagram reading skills diagram decomposition/size management understanding of software modeling use cases; understanding advantages and limitations of codegeneration. Actually, the concept is a set of special mental skills and focuses on successful adoption of model based abstractions into a various software development activities meeting real industrial challenges. We present also a course based on the concept suggested. The course addresses to 3-4 year software engineering bachelor students.

Keywords: model based software development, UML, software development, domain-specific modeling, teaching software engineering

References

1. **Broy M.** Seamless Method- and Model-based Software and Systems Engineering / ed. S. Nanz. *The Future of Software Engineering*. Berlin Heidelberg, Springer-Verlag, 2011, pp. 33–49.
2. **Kelly S., Tolvanen J.-P.** *Domain-Specific Modeling: Enabling Full Code Generation*. John Wiley&Sons, 2008, 340 p.
3. **OMG** Unified Modeling Language (OMG UML), Version 2.5, 2013, 786 p.
4. **MDA** Guide Version 1.0.1. omg/2003-06-01. June 2003. 62 p.
5. **Business** Process Model and Notation (BPMN). Version 2.0. January 2011, OMG. 538 p.
6. **Petre M.** UML in Practice. *Proc. of 35th International Conference on Software Engineering (ISCE)*. May 18–26, 2013, San Francisco, CA. IEEE Press, 2013, pp. 722–731.
7. **Bezivin J., France R., Gogolla M., Haugen O., Taentzer G., Varro D.** Teaching Modeling: Why, When, What? *MODELS'09: Proceedings of the 2009 international conference on Models in Software Engineering*. Berlin Heidelberg, Springer-Verlag, 2010, pp. 55–62.
8. **Hoare C. A. R.** The Role of Formal Techniques: Past, Current and Future or How Did Software Get so Reliable without Proof? *Proceedings 18th International Conference on Software Engineering*. Berlin, Germany, March 25–29, 1996. IEEE Computer Society, 1996, pp. 233–234.
9. **Borstler J., Kuzniarz L., Alphonse C. Sanders W., Smialek M.** Teaching software modeling in computing curricula. *Proceedings of the final reports on Innovation and technology in computer science education 2012 (ITiCSE-WGR '12)*. NY, USA: ACM New York, 2012, pp. 39–50.
10. **Ivanov D. Ju., Novikov F. A.** Modelirovanie na UML: oypit prepodavaniya v Internet. *Komp'yuternye instrumenty v obrazovanii*, 2009, no. 4. pp. 53–61 (in Russian).
11. **Tihomirov V. N.** Obuchenie studentov metodam graficheskogo programirovaniya na primere DRAGON i UML. *Sbornik nauch. trudov Moskovskogo Mashinostroitel'nogo Universiteta*. Moscow, MGOU, 2014, pp. 95–102 (in Russian).
12. **Ivanov D. Ju., Novikov F. A.** *Unificirovannyj jazyk modelirovaniya: uch. posobie*. Saint Petersburg, Izd-vo gos. politehnicheskogo un-ta, 2010. Ser. Prioritetnye nacional'nye proekty "Obrazovanie" (in Russian).
13. **Leonenkov A. V.** *Samouchitel' UML 2*. Saint Petersburg, BHV-Peterburg, 2007, 576 p. (in Russian).
14. **Paige R. F., Polack F. A. C., Kolovos D. S., Rose L. M., Matragkas N., Williams J. R.** Bad Modelling Teaching Practices. *Proceedings of the MODELS Educators Symposium co-located with the ACM/IEEE 17th International Conference on Model Driven Engineering Languages and Systems (MODELS 2014)*, Valencia, Spain, September 29, 2014. CEUR Workshop Proceedings, 2014, vol. 1346, pp. 1–12. URL: <http://ceur-ws.org/Vol-1346/>
15. **Joint ACM/IEEE Task Force on Computing Curricula.** Software engineering 2004: Curriculum guidelines for undergraduate degree programs in software engineering. Technical report, IEEE CS and ACM, 2004.
16. **Kramer J.** Is abstraction the key to computing? *Communications of the ACM*, 2007, vol. 50, no. 4, pp. 36–42.
17. **Jones J. K.** *Inzhenernoe i khudozhestvennoe konstruirovaniye*. Per. s angl. Moscow, Mir, 1976, 374 p. (in Russian).
18. **Devlin K.** Why universities require computer science students to take math. *Communication of ACM*, 2003, vol. 46, no. 9, pp. 37–39.
19. **Pavlov V. L., Yatsenko A.** "The Babel Experiment": An Advanced Pantomime-based Training in OOA&OOD with UML. *Proceedings of the 36th SIGCSE technical symposium on Computer science education*. New York, USA, ACM, 2005, pp. 231–235.
20. **Harel D., Politi M.** *Modeling Reactive Systems with Statecharts: state machine approach*. McGraw-Hill, 1998, 258 p.
21. **Moisan S., Rigault J.-P.** Teaching object-oriented modeling and UML to various audiences. *MODELS'09: Proceedings of the 2009 international conference on Models in Software Engineering*. LNCS, 2010, vol. 6002, pp. 40–54.
22. **Burton P. J., Bruhn R. E.** Using UML to facilitate the teaching of object-oriented systems analysis and design. *Journal of Computing Sciences in Colleges*, 2004, vol. 19, no. 3, pp. 278–290.
23. **Flint S., Gardner H., Boughton C.** Executable/Translatable UML in computing education. *Proceedings of the Sixth Australasian Conference on Computing Education (ACE'04)*. Darlinghurst, Australia: Australian Computer Society, Inc., 2004, pp. 69–75.
24. **Bolognesi A., Ciancarini P., Moretti R.** On the Education of Future Software Engineering. In *Software Engineering Education in the Modern Age*. LNCS, 2006, vol. 4309, pp. 186–205.
25. **Lengyel L., Mezei G.** Model-driven paradigms: the evolution of a university course. *Proc. of the 8th edition of the Educators' Symposium (EduSymp '12)*, 2012, pp. 13–20.
26. **Booch G., Jacobson I., Rumbaugh J.** *The Unified Software Development Process*, Prentice Hall, 1999, 512 p.
27. **Booch G.** *Object-Oriented Analysis And Design With Application, second edition*. The Benjamin/Cummings Publishing Company, Inc., 1994, 589 p.
28. **Terekhov A. N.** RTST-tehnologiya programirovaniya vstroennykh sistem real'nogo vremeni. *Zapiski seminarov kafedry sistemnogo programirovaniya "Case-sredstva RTST++"*, Saint Petersburg, Izd-vo SPbGU, 1998, pp. 3–17 (in Russian).
29. **Terekhov A., Dolgov P., Ivanov A., Koznov Dm., Lebedev A., Murashova T., Parfenov V.** Ob'ektno-orientirovannoe rasshirenie tekhnologii RTST. *Zapiski Seminarov Kafedry Sistemnogo Programirovaniya "Case-sredstva RTST++"*, Saint Petersburg, Izd-vo SPbGU, 1998, pp. 17–37 (in Russian).
30. **Litvinov Yu.V.** Primenenie DSM-platformy QREAL, pri razrabotke sredy programirovaniya robotov QREAL:ROBOTS. *Sistemnoe Programirovanie*, 2012, vol. 7, no. 1, pp. 161–186 (in Russian).
31. **Terekhov A. N., Bryksin T. A., Litvinov Yu. V.** QREAL: platforma vizual'nogo predmetno-orientirovannogo modelirovaniya. *Programmnaya Ingeneria*, 2013, no. 6, pp. 11–19 (in Russian).
32. **Koznov D. V.** *Osnovy vizual'nogo modelirovaniya: uch. posobie*. Moscow, Internet-Universitet Informatsionnykh Tekhnologii; BINOM. Laboratoriya znaniy, 2008, 246 p. (in Russian).
33. **Grossman M., Aronson J. E., McCarthy R. V.** Does UML make the grade? Insights from the software development community. *Information and Software Technology*, 2005, vol. 47, no. 6, pp. 383–397.
34. **Kudryavtsev D. V., Arzumanyan M. Yu., Grigor'ev L. Yu.** *Tekhnologii biznes-inzhiniringa*. Saint Petersburg, Izd-vo SPbGPU, 2014, 427 p. (in Russian).
35. **Gavrilova T. A., Leshcheva I. A., Strakhovich E. V.** Ob ispol'zovanii vizual'nykh kontseptual'nykh modelei v prepodavanii. *Vestnik Sankt-Peterburgskogo universiteta. Seriya 8: Menedzhment*, 2011, no. 4, pp. 124–150 (in Russian).
36. **Gavrilova T. A., Leshcheva I. A., Leshchev D. V.** Ispol'zovanie ontologii v kachestve didakticheskogo sredstva. *Iskusstvennyi Intellect*, 2000, no. 3, pp. 34-39 (in Russian).

А. В. Мальцев, канд. физ.-мат. наук, ст. науч. сотр., e-mail: avmaltcev@mail.ru,
М. В. Михайлюк, д-р физ.-мат. наук, зав. отделом, e-mail: mix@niisi.ras.ru,
Научно-исследовательский институт системных исследований РАН, Москва

Реализация эргономичного интерфейса управления виртуальной моделью антропоморфного робота с использованием устройства Kinect

Рассмотрено создание эргономичного бесконтактного интерфейса управления виртуальной моделью человекоподобного робота в "копирующем режиме", обеспечивающем повторение этой моделью движений оператора. Предложены методы для расчета текущих углов в сочленениях модели робота по опорным точкам скелета оператора, найденным с помощью устройства Kinect, и передачи значений этих углов в систему управления, изменяющую в соответствии с ними позу робота.

Ключевые слова: антропоморфный робот, управление, виртуальная модель, опорные точки, бесконтактный интерфейс, устройство Kinect

Введение

В настоящее время для решения широкого круга задач в различных областях науки и техники проектируют и используют антропоморфных роботов [1] (т. е. роботов, по своему строению сходных с человеком) и их виртуальные модели [2]. Последние особенно распространены в имитационно-тренажерных комплексах [3] подготовки операторов для работы с механическими роботами — прототипами этих моделей. Применение роботов именно такого класса, в первую очередь, связано с тем, что объекты, созданные человеком в процессе его развития, приспособлены под самого человека. Ему удобно брать эти вещи руками, манипулировать ими, использовать для работы и т. д. Поэтому для замены человека роботом при выполнении многих работ наиболее привычным и подходящим является именно антропоморфный робот.

Одной из важнейших задач при использовании человекоподобного робота (или его виртуальной модели) является создание системы управления им. Использование для этих целей стандартных интерфейсных устройств компьютера (клавиатуры, мыши, джойстика) не отвечает уровню задач, решаемых с помощью такого робота. Применение пультов управления [4] затруднительно, поскольку каждый их элемент обычно управляет движением робота в одном сочленении (например, сгиб руки в локте или подъем руки в плече). Поэтому, используя пульт, сложные движения руки можно выполнить только с помощью последовательных движений в каждом сочленении, что является неэргономичным и неэф-

фективным процессом, занимающим длительное время. Одним из решений в данном случае является использование экзоскелета, который надевают на оператора. Он представляет собой металлическую конструкцию, состоящую из жестких звеньев и имеющую те же сочленения, что у человека. Повороты всех звеньев экзоскелета передаются в систему управления реальным роботом или его виртуальной моделью, обеспечивая повторение роботом (моделью робота) движений оператора. Такой тип управления принято называть копирующим режимом. Преимущество экзоскелета состоит в том, что он является как бы идеальным джойстиком, поскольку в любой момент времени определяет углы поворота звеньев одновременно во всех сочленениях. Однако существенным недостатком при этом является необходимость ношения оператором громоздкого и тяжелого оборудования. Зачастую ношение экзоскелета вообще не представляется возможным. Например, если оператор должен быть в специальном костюме или скафандре.

На настоящее время одним из перспективных направлений при разработке и реализации новых технологий управления как виртуальными, так и реальными объектами является так называемый бесконтактный интерфейс. В работе [5] приведен обзор методов отслеживания движения руки оператора на основе технического зрения. В ряде работ (например, [6]) для копирования движения предложено решать оптимизационную задачу по уменьшению различия между реальной позой оператора и позой 3D-модели. Работы в данной области ведутся и в России. Сравнительный анализ имеющихся способов управле-

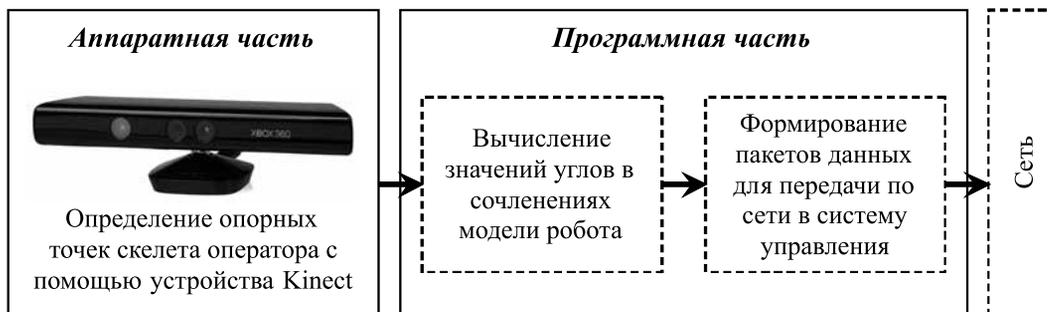


Рис. 1. Структурная схема предлагаемого интерфейса управления

ния андроидным роботом представлен в работе [7]. В этой работе отмечено, что управление андроидом с помощью устройства Kinect является перспективным, так как не требует затрат времени на создание управляющих команд. В работе [8] предложено использовать устройство Kinect для распознавания жестов оператора для управления программным обеспечением. В статье [9] продемонстрирован метод для управления виртуальной моделью кисти руки.

В настоящей работе предложен подход к использованию устройства Kinect для реализации эргономичного бесконтактного интерфейса управления в копирующем режиме трехмерной виртуальной моделью антропоморфного робота. Суть подхода подробно рассмотрена на примере управления рукой модели робота. Отработанные на виртуальных моделях решения затем можно без особых доработок перенести на реальные механические роботы. Далее приведено описание структуры предлагаемого интерфейса и реализации отдельных его частей.

1. Структура интерфейса управления

Разработанный интерфейс управления виртуальной моделью человекоподобного робота включает аппаратную и программную части. Его структурная схема представлена на рис. 1.

Аппаратную часть предлагается реализовывать на базе Kinect — мультikomпонентного устройства, подключаемого к персональному компьютеру через USB-интерфейс, включающего:

- инфракрасный излучатель, испускающий ИК-лучи, которые далее отражаются от окружающих предметов;
- инфракрасный сенсор глубины, который принимает отраженные ИК-лучи и на их основе вычисляет расстояния от сенсора до точек объектов — строит карту глубины рабочей области;
- цветную (RGB) видеокамеру.

Используя полученную с помощью своих камер и сенсоров информацию, устройство Kinect вычисляет с частотой 30 раз в секунду трехмерные координаты (в своей системе координат) точек сочленения костей оператора, находящегося в рабочей области устройства. Далее будем называть эти точки опорными точками скелета оператора.

Программная часть предлагаемого интерфейса обеспечивает вычисление углов между звеньями конечностей модели управляемого робота (на основе полученных аппаратной частью координат опорных точек), а также передачу значений этих углов в систему управления виртуальной моделью по специальному информационному протоколу. Система управления, в свою очередь, будет выполнять изменение соответствующих углов в виртуальной модели робота.

2. Расчет углов в сочленениях руки модели робота

Рассмотрим реализацию программной части предлагаемого интерфейса на примере управления левой рукой виртуальной модели робототехнической системы SAR-401, имитирующей верхнюю половину человеческого тела. Исходное положение рук модели робота представлено на рис. 2. В начале процесса управления оператор должен принять аналогичную позу в целях проведения калибровки интерфейса,



Рис. 2. Модель робота SAR-401

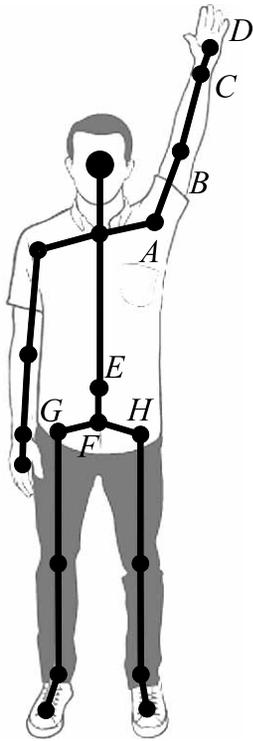


Рис. 3. Точки скелета оператора, определяемые устройством Kinect

которая необходима для обеспечения правильного повторения роботом дальнейших движений оператора.

Вычисление требуемых углов между звеньями руки робота будем проводить на основе координат опорных точек, получаемых с устройства Kinect (аппаратная часть интерфейса) и представленных в системе координат KCS, связанной с устройством. На рис. 3 кружками показаны все опорные точки скелета оператора, доступные после работы аппаратной части интерфейса [10]. Из них для использования в системе управления потребуются точки *A* (плечевой сустав), *B* (локтевой сустав), *C* (лучезапястный сустав) и *D* (центр кисти), относящиеся к левой руке оператора. Кроме того, в процессе расчетов будем также использовать точки *E*, *F*, *G* и *H*, которые необходимы для установления ориентации оператора относительно сенсора устройства Kinect. Эти четыре

точки выбраны, так как они образуют жестко связанную структуру в пространстве. В данной работе не затрагиваем движение пальцами руки робота, поскольку устройство Kinect не отслеживает для них опорные точки. Вычисление таких точек может быть в перспективе реализовано путем программного анализа изображений оператора, которые можно получить с устройства Kinect, а также путем добавления в аппаратную часть интерфейса устройства Leap Motion [11] для бесконтактного отслеживания скелета кистей рук.

На рис. 4 представлена схема механического устройства левой руки виртуальной модели робота SAR-401. Движения руки и повороты звеньев друг относительно друга обеспечиваются системой иерархически связанных шарниров, каждый из которых в данной модели робота обладает одной степенью свободы. Шарниры P_1 и P_2 расположены в точке крепления руки робота к его плечу (соответствует точке *A* оператора), P_3 и P_4 — между плечевым звеном и предплечьем (у оператора это точка *B*), P_5 и P_6 — в месте соединения предплечья и кисти (соответствует точке *C*).

Для определения углов поворота φ , ψ , θ , γ , ω и σ в шарнирах P_1 , P_2 , P_3 , P_4 , P_5 и P_6 соответственно введем в пространстве оператора правосторонние локальные системы координат (СК) CS_1 и CS_2 с центром в точке *A*, CS_3 и CS_4 с центром в точке *B*, CS_5 и CS_6 с центром в точке *C*, связанные с этими шарнирами. На рис. 5 данные системы координат показаны в тот момент, когда рука находится в исходном калибровочном положении, при котором все углы равны нулю. Направления координатных осей

Шарнир предплечья (крен) Шарнир плеча Шарнир кисти (курс) Шарнир руки Шарнир локтя Кисть Шарнир предплечья

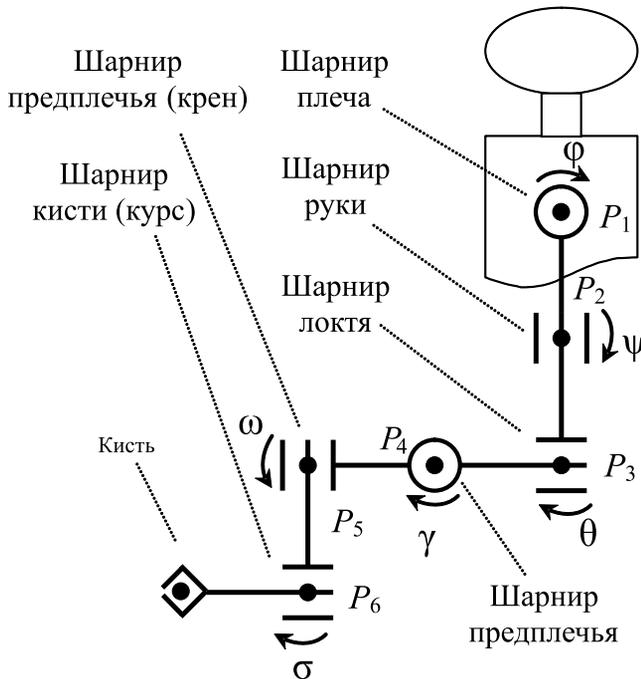


Рис. 4. Схема левой руки робота SAR-401

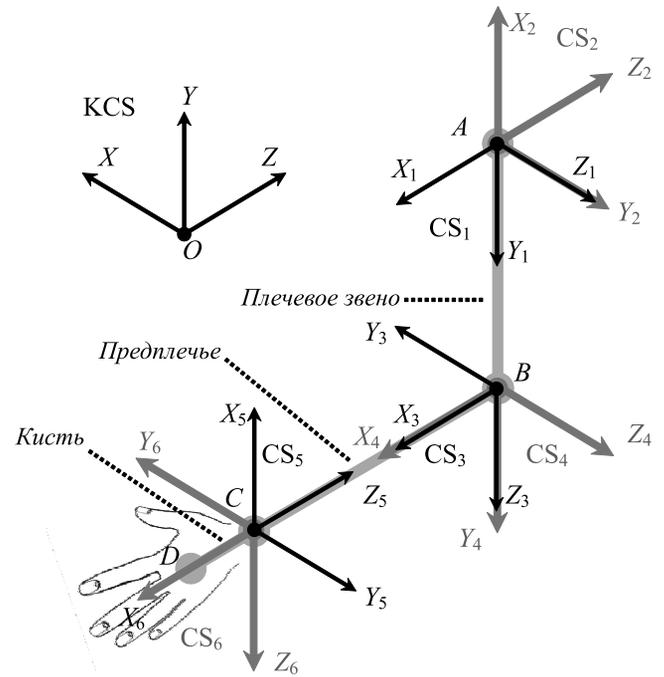


Рис. 5. Системы координат в точках руки

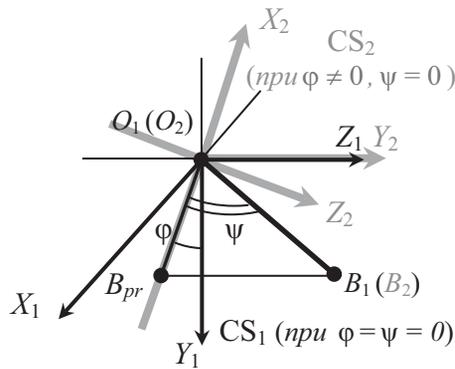


Рис. 6. Вычисление углов φ и ψ

систем выбраны так, что они совпадают с направлениями осей в соответствующих шарнирах виртуальной модели. Поскольку рассматриваемые шарниры обладают лишь одной степенью свободы, то повороты в каждой из систем возможны только вокруг одной оси, в рассматриваемом случае для системы с номером i такой осью является ось Z_i . Положительным поворотом (угол > 0) будем считать поворот против часовой стрелки, если смотреть с конца направляющего вектора оси Z_i . Заметим, что системы координат CS_i имеют такую же иерархическую связь, как и шарниры в модели робота, т. е. поворот СК с индексом i влечет изменение всех СК, чей индекс больше i .

Чтобы вычислить угол φ , найдем координаты B_1 (рис. 6) точки B в системе координат CS_1 левого плечевого звена, находящегося в исходном калибровочном положении (т. е. когда $\varphi = 0, \psi = 0$). Матрица перехода из системы координат KCS устройства Kinect в СК CS_1 в таком случае будет иметь вид

$$\mathbf{M}_{CS_1} = \mathbf{M}_R \mathbf{M}_{T,OA}, \quad (1)$$

$$\mathbf{M}_R = \begin{pmatrix} e_{1,x} & e_{2,x} & e_{3,x} & 0 \\ e_{1,y} & e_{2,y} & e_{3,y} & 0 \\ e_{1,z} & e_{2,z} & e_{3,z} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{M}_{T,OA} = \begin{pmatrix} 1 & 0 & 0 & -A_x \\ 0 & 1 & 0 & -A_y \\ 0 & 0 & 1 & -A_z \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

где \mathbf{M}_R — матрица поворота осей, $\mathbf{M}_{T,OA}$ — матрица переноса начала СК из точки O (начало системы KCS) в точку $A (A_x, A_y, A_z)$ левого плеча; $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ — базисные векторы СК CS_1 (соответствующие осям X_1, Y_1, Z_1) с координатами в СК KCS. Данные векторы можно вычислить с помощью определяемых устройством Kinect координат точек E, F, G и H скелета оператора, которые, как уже было упомянуто выше, образуют жестко связанную структуру:

$$\mathbf{e}_2 = \frac{F - E}{|F - E|}, \quad \mathbf{e}_3 = \frac{H - G}{|H - G|}, \quad \mathbf{e}_1 = \frac{[\mathbf{e}_2, \mathbf{e}_3]}{||[\mathbf{e}_2, \mathbf{e}_3]||}.$$

Тогда $B_1 = \mathbf{M}_{CS_1} B$. Угол φ определим как угол между вектором $\mathbf{O}_1 \mathbf{B}_{pr}$ (где $\mathbf{B}_{pr} = (B_{1,x}, B_{1,y}, 0)$ — проекция точки B_1 на плоскость $X_1 Y_1$ системы CS_1)

и направляющим вектором $\mathbf{Y}_1 = (0, 1, 0)$ координатной оси Y_1 :

$$\varphi = \begin{cases} -\arccos(c_\varphi), & B_{1,x} > 0 \\ \arccos(c_\varphi), & B_{1,x} \leq 0, \end{cases}$$

$$c_\varphi = \cos \varphi = \frac{(\mathbf{O}_1 \mathbf{B}_{pr}, \mathbf{Y}_1)}{|\mathbf{O}_1 \mathbf{B}_{pr}|} = \frac{B_{1,y}}{\sqrt{B_{1,x}^2 + B_{1,y}^2}}.$$

Для определения ψ (рис. 6) вычислим координаты B_2 точки B в положении локальной СК CS_2 плечевого звена, когда $\varphi \neq 0, \psi = 0$. Исходя из выбранного расположения осей систем CS_1 и CS_2 (см. рис. 5) и принимая во внимание тот факт, что CS_2 иерархически зависит от CS_1 (поворачивается вместе с ней на угол φ), имеем:

$$B_2 = \mathbf{M}_{CS_2} B = \mathbf{M}_{12} \mathbf{M}_Z^T(\varphi) \mathbf{M}_{CS_1} B, \quad (2)$$

где \mathbf{M}_{CS_2} — матрица перехода из СК KCS в СК CS_2 , \mathbf{M}_{CS_1} — матрица перехода из равенства (1); $\mathbf{M}_Z(\varphi)$ — матрица поворота на угол φ вокруг оси Z_1 системы CS_1 ; \mathbf{M}_{12} — матрица "переименования" осей координат;

$$\mathbf{M}_Z(\varphi) = \begin{pmatrix} \cos \varphi & -\sin \varphi & 0 & 0 \\ \sin \varphi & \cos \varphi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{M}_{12} = \begin{pmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (3)$$

Угол ψ найдем из скалярного произведения между вектором $\mathbf{O}_2 \mathbf{B}_2$ и вектором $-\mathbf{X}_2 = (-1, 0, 0)$, противоположным направляющему вектору координатной оси X_2 системы CS_2 (O_2 — начало CS_2):

$$\psi = \begin{cases} -\arccos(c_\psi), & B_{2,y} > 0 \\ \arccos(c_\psi), & B_{2,y} \leq 0, \end{cases}$$

$$c_\psi = \cos \psi = \frac{(\mathbf{O}_2 \mathbf{B}_2, -\mathbf{X}_2)}{|\mathbf{O}_2 \mathbf{B}_2|} = \frac{-B_{2,x}}{\sqrt{B_{2,x}^2 + B_{2,y}^2 + B_{2,z}^2}}.$$

Отметим, что при $|\mathbf{O}_1 \mathbf{B}_{pr}| = 0$, т. е. когда ψ достигает -90° (плечевое звено перпендикулярно боковой поверхности туловища), однозначное определение угла φ по точкам A и B невозможно. Выходом в этом случае является присвоение φ предыдущего вычисленного значения до тех пор, пока $|\mathbf{O}_1 \mathbf{B}_{pr}| = 0$.

Для расчета углов θ и γ (рис. 7) вычислим координаты C_3 точки C в системе координат CS_3 левого предплечья, когда оно находится в исходном калибровочном положении ($\theta = 0, \gamma = 0$), а плечевое звено при этом уже повернуто на углы φ и ψ . Учитывая иерархическую зависимость CS_3 от CS_2 и первоначальное расположение их осей (см. рис. 5), получим:

$$C_3 = \mathbf{M}_{CS_3} C = \mathbf{M}_{23} \mathbf{M}_Z^T(\psi) \mathbf{M}_{T,AB} \mathbf{M}_{CS_2} C, \quad (4)$$

где \mathbf{M}_{CS_3} — матрица перехода из СК KCS в СК CS_3 ; \mathbf{M}_{CS_2} — матрица перехода из равенства (2); $\mathbf{M}_{T,AB}$ —

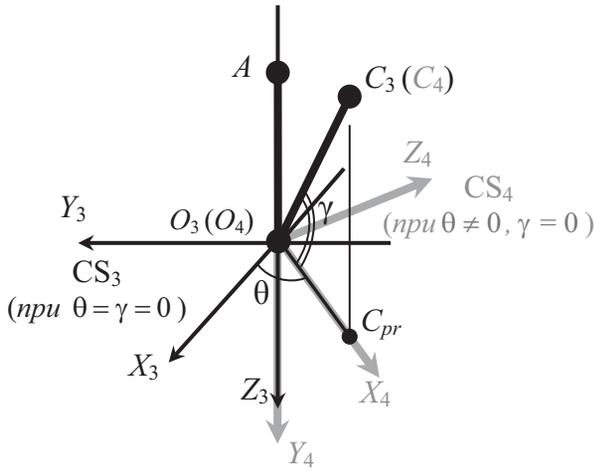


Рис. 7. Вычисление углов θ и γ

матрица переноса начала СК из точки A в точку B ; $M_Z(\psi)$ — матрица поворота на угол ψ вокруг оси Z_2 системы CS_2 (аналогична матрице, приведенной в формуле (3)); M_{23} — матрица "переименования" осей координат;

$$M_{T,AB} = \begin{pmatrix} 1 & 0 & 0 & -B_{2,x} \\ 0 & 1 & 0 & -B_{2,y} \\ 0 & 0 & 1 & -B_{2,z} \\ 0 & 0 & 0 & 1 \end{pmatrix}, M_{23} = \begin{pmatrix} 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Угол θ найдем из скалярного произведения между вектором O_3C_{pr} , где $C_{pr} = (C_{3,x}, C_{3,y}, 0)$ — проекция точки C_3 на плоскость X_3Y_3 системы CS_3 , и направляющим вектором $X_3 = (1, 0, 0)$ координатной оси X_3 системы CS_3 (см. рис. 7):

$$\theta = \begin{cases} \arccos(c_\theta), C_{3,y} \geq 0 \\ -\arccos(c_\theta), C_{3,y} < 0, \end{cases}$$

$$c_\theta = \cos \theta = \frac{(O_3C_{pr}, X_3)}{|O_3C_{pr}|} = \frac{C_{3,x}}{\sqrt{C_{3,x}^2 + C_{3,y}^2}}.$$

Чтобы найти угол γ (см. рис. 7), определим координаты C_4 точки C в положении локальной СК CS_4 предплечья, когда $\theta \neq 0$, $\gamma = 0$. Исходя из выбранного расположения осей систем CS_3 и CS_4 (рис. 5) и учитывая зависимость CS_4 от CS_3 (поворачивается вместе с ней на угол θ), имеем:

$$C_4 = M_{CS_4} C = M_{34} M_Z^T(\theta) M_{CS_3} C, \quad (5)$$

где M_{CS_4} — матрица перехода из СК KCS в СК CS_4 ; M_{CS_3} — матрица перехода из равенства (4); $M_Z(\theta)$ — матрица поворота на угол θ вокруг оси Z_3 системы CS_3 (см. (3)); M_{34} — матрица "переименования" осей координат;

$$M_{34} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Определим γ , используя скалярное произведение вектора O_4C_4 и вектора $-Y_4 = (0, -1, 0)$, противоположного направляющему вектору координатной оси Y_4 системы CS_4 :

$$\gamma = \arccos\left(\frac{(O_4C_4, -Y_4)}{|O_4C_4|}\right) - 90^\circ =$$

$$= \arccos\left(\frac{-C_{4,y}}{\sqrt{C_{4,x}^2 + C_{4,y}^2 + C_{4,z}^2}}\right) - 90^\circ.$$

Заметим, что при $|O_3C_{pr}| = 0$, т. е. если $\gamma = \pm 90^\circ$, однозначное определение угла θ по точкам B и C невозможно. Значение 90° соответствует случаю, когда рука оператора полностью разогнута в локте. Случай $\gamma = -90^\circ$, означающий полностью согнутую руку, не может быть получен в связи с особенностями строения человеческой руки. Для устранения неоднозначности зафиксируем предыдущее рассчитанное значение θ до тех пор, пока $|O_3C_{pr}| = 0$.

Чтобы определить углы ω и σ (рис. 8), вычислим координаты D_5 точки D в системе координат CS_5 левой кисти, когда она находится в исходном калибровочном положении ($\omega = 0$, $\sigma = 0$), а плечевое звено и предплечье при этом уже повернуты на углы φ , ψ и θ , γ соответственно. Аналогично формуле (4) получаем:

$$D_5 = M_{CS_5} D = M_{45} M_Z^T(\gamma) M_{T,BC} M_{CS_4} D, \quad (6)$$

M_{CS_5} — матрица перехода из СК KCS в СК CS_5 ; M_{CS_4} — матрица перехода из равенства (5); $M_{T,BC}$ — матрица переноса начала СК из точки B в точку C ; $M_Z(\gamma)$ — матрица поворота на угол γ вокруг оси Z_4 системы CS_4 (аналогична матрице, приведенной в формуле (3)); $M_{45} = M_{12}$ — матрица "переименования" осей координат (см. (3));

$$M_{T,BC} = \begin{pmatrix} 1 & 0 & 0 & -C_{4,x} \\ 0 & 1 & 0 & -C_{4,y} \\ 0 & 0 & 1 & -C_{4,z} \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Учитывая биомеханику человеческой руки, можно понять, что при $D_{5,y} \geq 0$ кисть разогнута наружу ($\sigma < 0$), а при $D_{5,y} < 0$ — согнута внутрь ($\sigma \geq 0$). Поэтому угол σ (рис. 8), который задает поворот вокруг оси Z_6 системы CS_6 , определим следующим образом:

$$\sigma = \begin{cases} -\arccos(c_\sigma), D_{5,y} \geq 0 \\ \arccos(c_\sigma), D_{5,y} < 0, \end{cases}$$

$$c_\sigma = \cos \sigma = \frac{(O_5D_5, -Z_5)}{|O_5D_5|} = \frac{-D_{5,z}}{\sqrt{D_{5,x}^2 + D_{5,y}^2 + D_{5,z}^2}},$$

где $-Z_5 = (0, 0, -1)$ — вектор, противоположный направляющему вектору координатной оси Z_5 системы CS_5 с началом в точке $O_5(0,0,0)$. Вычисление угла ω также зависит от того, согнута кисть внутрь

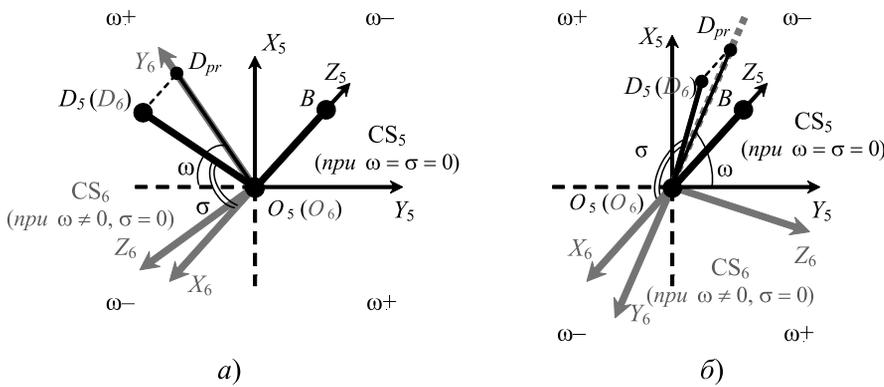


Рис. 8. Вычисление углов ω и σ при согнутой внутрь (а) и разогнутой наружу (б) кисти

(рис. 8, а) или разогнута наружу (рис. 8, б). Поскольку в рамках поставленной выше задачи $\omega > 0$ задает поворот против часовой стрелки вокруг оси Z_5 , если смотреть из конца направляющего этой оси, то

$$\omega = \begin{cases} 90^\circ - \arccos(c_\omega), D_{5,y} \leq 0 \\ \arccos(c_\omega) - 90^\circ, D_{5,y} > 0, \end{cases}$$

$$c_\omega = \cos \omega = \frac{(\mathbf{O}_5 \mathbf{D}_{pr}, \mathbf{X}_5)}{|\mathbf{O}_5 \mathbf{D}_{pr}|} = \frac{D_{5,x}}{\sqrt{D_{5,x}^2 + D_{5,y}^2}},$$

где $\mathbf{D}_{pr} = (D_{5,x}, D_{5,y}, 0)$ — проекция точки D_5 на плоскость $X_5 Y_5$ системы CS_5 ; $\mathbf{X} = (1, 0, 0)$ — направляющий вектор оси X_5 системы CS_5 .

Заметим, что при $|\mathbf{O}_5 \mathbf{D}_{pr}| = 0$, т. е. если $\sigma = 0$, однозначное определение угла ω по точкам C и D невозможно. В таком случае зафиксируем предыдущее рассчитанное значение ω до тех пор, пока $|\mathbf{O}_5 \mathbf{D}_{pr}| = 0$.

Таким образом находят требуемые углы в шарнирах между звеньями левой руки виртуальной модели робота SAR-401 на основе опорных точек скелета оператора, координаты которых определяют устройством Kinect. Углы в других шарнирах модели робота определяют аналогичным образом. Далее рассмотрим передачу вычисленных углов в систему управления моделью робота.

3. Передача данных в систему управления и реализация копирующего режима

При обмене информацией между разработанным беспроводным интерфейсом на основе устройства Kinect и системой управления виртуальной моделью робота передача данных реализуется по локальной вычислительной сети. Информационный обмен при этом осуществляется пакетами постоянной длины. Каждый пакет начинается с заголовка, за которым следуют информационные блоки, соответствующие подвижным объектам (шарнирам и связанным с ними звеньями робота). В заголовке пакета указывается номер пакета и его длина в байтах, а также число передаваемых параметров. Каждый информационный блок содержит идентификатор одного из шарниров модели робота и вычисленное для этого шарнира значение угла, которое требуется установить.

Обмен данными осуществляется по протоколу UDP с использованием несвязанных сокетов

(unconnected sockets). Порты для приема и передачи данных определяются при отладке. Период посылки пакетов в адрес системы управления не должен составлять более 40 мс. В этом случае движения визуализируемой виртуальной модели робота выглядят непрерывными и связными.

Предложенные методы и алгоритмы реализованы в виде программных модулей на языке Microsoft Visual Studio C++ для персональных компьютеров и встроены в разработанный в НИИСИ РАН имитационно-тренажерный комплекс [5]. Комплекс

состоит из подсистемы управления, подсистемы динамики и подсистемы визуализации. Для тестирования рассмотренного интерфейса была создана 3D-модель антропоморфного робота SAR-401, реальный прототип которого разработан НПО "Андроида техника" (г. Магнитогорск). В системе управления с помощью специального редактора создана функциональная схема, вычисляющая по поступающим на вход углам значения напряжений на электрические двигатели в шарнирах модели робота. Эти значения передаются в подсистему динамики, которая моделирует работу двигателей и осуществляет необходимый поворот всех звеньев модели. Подсистема визуализации синтезирует изображение виртуальной 3D-сцены с новым состоянием модели робота. Работа всех подсистем осуществляется в масштабе реального времени, т. е. весь цикл обработки одного шага интерфейса, моделирования и визуализации занимает не более 40 мс, что обеспечивает частоту шагов не менее 25 раз в секунду. Тестирование предложенного интерфейса проводилось по визуальному соответствию углов поворота звеньев робота углам поворота в сочленениях руки оператора. Испытания показали достаточно хорошую точность повторения моделью робота движений оператора, т. е. качество реализации копирующего режима.

Заключение

Предложен подход к реализации эргономичного бесконтактного интерфейса управления виртуальной моделью антропоморфного робота в копирующем режиме. Предложены и подробно описаны методы определения углов в шарнирах, расположенных между звеньями руки виртуальной модели антропоморфного робота SAR-401 на основе данных об операторе, получаемых с устройства Kinect. Рассмотрен способ передачи найденных значений углов в систему управления такой моделью. Используя описанный подход, можно осуществлять управление и другими моделями антропоморфных робототехнических систем, включающими не только верхние, но и нижние конечности. Предложенное решение применимо как для виртуальных моделей роботов, так и в случае реализации управления реальным механическим человекоподобным роботом.

Работа выполняется при поддержке РФФИ, проект 15-07-04544.

Список литературы

1. Михайлюк М. В., Сохин И. Г., Торгашев М. А., Бурдин Б. В. Исследование возможности использования андроидов в космосе с помощью технологий виртуальной реальности // Материалы 10-й Междунар. науч.-практ. конф. "Пилотируемые полеты в космос". ФГБУ НИИ ЦПК им. Ю. А. Гагарина, Звездный городок, 2013. С. 29—31.
2. Бурдин Б. В., Михайлюк М. В., Сохин И. Г., Торгашев М. А. Использование виртуальных 3D-моделей для экспериментальной отработки бортовых полетных операций, выполняемых с помощью антропоморфных роботов // Робототехника и техническая кибернетика. 2013. № 1. С. 42—46.
3. Михайлюк М. В. Видеотренажерный комплекс управления роботами и манипуляторами // Труды Междунар. симп. "Инновационные технологии в исследовании окружающей среды". МФТИ, Москва, 2013. С. 84—85.
4. Михайлюк М. В. Двумерные виртуальные пульта управления в тренажерных комплексах // Программная инженерия. 2014. № 5. С. 20—25.
5. Erol A., Bebis G., Nicolescu M. et al Vision-based Hand Pose Estimation: A review // Computer Vision and Image Understanding. 2007. Vol. 108 (1—2). P. 52—73.
6. Ramey A. A., González-Pacheco V., Salichs M. A. Integration of a low-cost rgb-d sensor in a social robot for gesture recognition // Proceedings of the 6th International Conference on Human Robot Interaction, HRI 2011, Lausanne, Switzerland, March 6—9, 2011. P. 229—230.
7. Шеломентев Е. Е., Дусеев В. Р. Сравнительный анализ различных способов управления роботом андроидного типа // Труды XVIII Междунар. науч.-практ. конф. "Современная техника и технологии". Томск: Изд-во ТПУ, 2012. С. 439—440. URL: http://www.lib.tpu.ru/fulltext/v/Conferences/2012/C2/V2/v2_216.pdf (дата обращения 26.05.2015).
8. Котюжанский Л. А. Интерфейс бесконтактного управления // Фундаментальные исследования. 2013. № 4. С. 44—48.
9. Oikonomidis I., Kyriazis N., Argyros A. Efficient model based 3D tracking of hand articulations using Kinect // In Proceedings of the 22nd British Machine Vision Conference. 2011. Dundee, Scotland: BMVA Press, 2011. P. 101.1—101.11.
10. Kinect for Windows SDK 1.8 URL: <https://msdn.microsoft.com/en-us/library/hh973074.aspx> (дата обращения 26.05.2015).
11. Weichert F., Bachmann D., Rudak B., Fisseler D. Analysis of the Accuracy and Robustness of the Leap Motion Controller // Sensors. 2013. Vol. 13. P. 63—80.

A. V. Maltsev, Senior Researcher, e-mail: avmaltcev@mail.ru, M. V. Mikhayluk, Head of Department, e-mail: mix@niisi.ras.ru, Scientific Research Institute of System Analysis (Russian Academy of Sciences), Moscow

Implementation of Ergonomic Contactless Management Interface for Anthropomorphic Robot's Virtual Model by Using Kinect

At this work the approach is considered to implementation ergonomic contactless management interface of anthropomorphic (humanoid) robot's virtual model in so-called copying mode which provides a repetition of operator's movements by this model. This interface includes hardware and software parts. The hardware part is based on using the Microsoft Kinect device that connects to a personal computer via USB port. It provides a coordinate determination of skeleton's reference points for operator which is in the device's working area. The software part of proposed interface calculates current values of angles in robot model's joints by using found reference points and send these values to management system. The management system in its turn changes a posture of robot model in accordance with received parameters. The essence of proposed methods for calculating angles and transfer of them by a network to management system is described in detail on the example of left hand control of the humanoid robot's virtual model (SAR-401). The methods for implementation of ergonomic contactless interface, considered in this article, allow managing not only this virtual model but also other models of anthropomorphic robotic systems. The described solutions are also applicable in the case of the control interface creation for real mechanical humanoid robot.

Keywords: anthropomorphic robot, management, virtual model, reference points, contactless interface, Kinect

References

1. Mikhayluk M. V., Sokhin I. G., Torgashev M. A., Burdin B. V. Issledovanie vozmozhnosti ispol'zovaniya androidov v kosmose s pomoshch'yu tekhnologiy virtual'noy real'nosti (Study of a possibility for using androids in space by dint of virtual reality technology). *Materialy 10-y Mezhdunar. nauch.-prakt. konf. "Pilotiруемые полеты в космос"*. FGBU NII CPK im. Yu. A. Gagarina, Zvezdnyy gorodok. 2013, pp. 29—31 (in Russian).
2. Burdin B. V., Mikhajljuk M. V., Sokhin I. G., Torgashev M. A. Ispol'zovanie virtual'nykh 3D-modely dlya eksperimental'noy otrabotki bortovykh poletnykh operatsiy, vypolnyaemykh s pomoshch'yu antropomorfnykh robotov (The usage of virtual 3D models for experimental exercising the flight operations performed with the help of anthropomorphic robots). *Robototekhnika i Tekhnicheskaya Kibernetika*, 2013, no. 1, pp. 42—46 (in Russian).
3. Mikhayluk M. V. Videotrenazhernyy kompleks upravleniya robotami i manipulyatorami (The video-training complex for managing robots and manipulators). *Trudy Mezhdunar. simp. "Innovatsionnye tekhnologii v issledovanii okruzhayushchey sredy"*. MFTI, Moscow, 2013, pp. 84—85 (in Russian).
4. Mikhayluk M. V. Dvumernye virtual'nye pul'ty upravleniya v trenazhernyykh kompleksakh (Two-dimensional virtual control panels in training complexes). *Programmnaya Inzheneriya*, 2014, no. 5, pp. 20—25 (in Russian).
5. Erol A., Bebis G., Nicolescu M., Boyle R. D., Twombly X. Vision-based Hand Pose Estimation: A review. *Computer Vision and Image Understanding*, 2007, vol. 108 (1—2), pp. 52—73.
6. Ramey A. A., Gonzalez-Pacheco V., Salichs M. A. Integration of a low-cost rgb-d sensor in a social robot for gesture recognition. *Proceedings of the 6th International Conference on Human Robot Interaction, HRI 2011*. Lausanne, Switzerland, March 6—9, 2011, New York, USA, ACM, 2011, pp. 229—23.
7. Shelomentsev E. E., Duseev V. R. Sravnitel'nyy analiz razlichnykh sposobov upravleniya robotom androidnogo tipa. (A comparative analysis of the various ways to control the robot android type). *Trudy XVIII Mezhdunar. nauch.-prakt. konf. "Sovremennaya tekhnika i tekhnologii"*. Tomsk, Ivd-vo TPU, 2012, pp. 439—440, available at: http://www.lib.tpu.ru/fulltext/v/Conferences/2012/C2/V2/v2_216.pdf (date of access 26.05.2015) (in Russian).
8. Kotyuzhanskiy L. A. Interfeys beskontaktnogo upravleniya. (Contactless Interface Control). *Fundamental'nye Issledovaniya*, 2013, no. 4, pp. 44—48 (in Russian).
9. Oikonomidis I., Kyriazis N., Argyros A. Efficient model based 3D tracking of hand articulations using Kinect. *In Proceedings of the 22nd British Machine Vision Conference*, 2011 Dundee, Scotland: BMVA Press, 2011, pp. 101.1—101.11.
10. Kinect for Windows SDK 1.8, available at: <https://msdn.microsoft.com/en-us/library/hh973074.aspx> (date of access 26.05.2015).
11. Weichert F., Bachmann D., Rudak B., Fisseler D. Analysis of the Accuracy and Robustness of the Leap Motion Controller. *Sensors*, 2013, vol. 13, pp. 63—80.

И. О. Жаринов, д-р техн. наук, доц., зав. каф., e-mail: igor_rabota@pisem.net, Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики (НИУ ИТМО), руководитель учебно-научного центра, Санкт-Петербургское ОКБ "Электроавтоматика" им. П. А. Ефимова,
О. О. Жаринов, канд. техн. наук, доц., Санкт-Петербургский государственный университет аэрокосмического приборостроения (ГУАП)

Решение задачи оценки параметров математической модели цветовоспроизведения дисплея с максимальным цветовым охватом на основе интерполяции линии спектральных цветностей

Рассмотрена задача и в рамках ее решения разработана методика оценки параметров математической модели цветовоспроизведения жидкокристаллических и светодиодных дисплеев, обладающих максимальным цветовым охватом. Методика основана на интерполяции таблично заданной линии спектральных цветностей цветового локуса на XU -плоскости и последующем вычислении координат цветности вершин треугольника цветового охвата, вписанного в цветовой локус и обладающего максимальной возможной площадью. Оценка параметров математической модели цветовоспроизведения дисплеев выполнена на основе решения системы уравнений, исходными данными для которой являются координаты цветности вершин треугольника цветового охвата с максимальной площадью и координаты цветности точки белого цвета, отображаемого на экране дисплея. Рассмотрены три способа интерполяции таблично заданной линии спектральных цветностей: на основе интерполяции в пространстве удельных координат цвета; на основе интерполяции в пространстве координат цветности; на основе интерполяции спектрального локуса как некоторой таблично заданной кривой, обладающей признаками формы. Получены числовые значения для профилей дисплеев, обладающих максимально возможной площадью треугольника цветового охвата и соответствующих колориметрическим стандартам D-75, D-65, D-55, D-50.

Ключевые слова: колориметрический стандарт, локус, координаты цветности, интерполяция, кривая Безье, треугольник цветового охвата, дисплей

Введение

Основным средством отображения информации в современной компьютерной технике является жидкокристаллический или светодиодный дисплей [1]. Цвет свечения каждого пикселя дисплея определяется значениями управляющих кодов трех основных цветов RGB (R — Red, G — Green, B — Blue).

Модель кодирования цвета, основанная на смешении кодов RGB , определяет [2, 3] на колориметрической XU -плоскости цветовой охват, границы которого аппроксимируются геометрической фигурой треугольной формы. Треугольник цветового охвата (ТЦО) находится внутри спектрального локуса, характеризующего область всех видимых цветов. Таким

образом, дисплей, схема управления пикселями которого основана на трехкомпонентной модели RGB , воспроизводит не все видимые человеческим глазом цвета и оттенки, а только часть из них, т. е. цвета и оттенки, (x, y) -координаты цветности которых находятся внутри ТЦО.

Координаты цветности вершин ТЦО (x_R, y_R) , (x_G, y_G) , (x_B, y_B) определяются [4] через параметры $X_r, X_g, X_b, Y_r, Y_g, Y_b, Z_r, Z_g, Z_b$ профиля дисплея — математической модели цветовоспроизведения дисплея, а площадь ТЦО и его местоположение на XU -плоскости характеризуют способность дисплея воспроизводить определенные цвета и оттенки цветов. Параметры профиля дисплея определяются технологией изготовления жидких кристаллов или светодиодов и мо-

гут быть рассчитаны [4] теоретически или оценены по результатам колориметрических измерений.

Дисплей, обладающий максимально возможной площадью S_{\max} ТЦО, вписанного в спектральный локус, будет иметь профиль с наилучшими свойствами по цветовоспроизведению.

Выпускаемые в настоящее время образцы дисплеев имеют профили, существенно отличающиеся от профиля с максимальной площадью ТЦО (площадь ТЦО обычного дисплея составляет примерно 30...40 % от площади ТЦО дисплея с профилем максимальной площади), в связи с чем актуальной является задача [5] поиска профиля дисплея, обладающего максимальной площадью S_{\max} треугольника цветового охвата, вписанного в спектральный локус.

1. Математическая и геометрическая модель цветовоспроизведения дисплея

Известно [6], что математическая модель цветовоспроизведения дисплея, основанная на смешении кодов *RGB* трех основных цветов, имеет следующий вид:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} X_r & X_g & X_b \\ Y_r & Y_g & Y_b \\ Z_r & Z_g & Z_b \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}, \quad \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} X_r & X_g & X_b \\ Y_r & Y_g & Y_b \\ Z_r & Z_g & Z_b \end{bmatrix}^{-1} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}, \quad (1)$$

$$x = \frac{X}{X+Y+Z}, \quad y = \frac{Y}{X+Y+Z}, \quad z = \frac{Z}{X+Y+Z}, \quad (2)$$

$$X = \frac{xY}{y}, \quad Z = \frac{(1-x-y)Y}{y}, \quad z = 1-x-y,$$

где x, y, z — координаты цветности; X, Y, Z — координаты цвета; $X_r, X_g, X_b, Y_r, Y_g, Y_b, Z_r, Z_g, Z_b$ — коэффициенты профиля дисплея; R, G, B — десятичные компоненты кода цвета. Прямое преобразование $RGB \rightarrow XYZ$ позволяет осуществлять переход от значений кодов *RGB* к значениям координат цвета, обратное преобразование $XYZ \rightarrow RGB$ — наоборот. Модель (1), (2) соответствует аддитивному закону смешения цветов Грассмана, используемому в колориметрических стандартах Международной комиссии по освещению (МКО).

Координаты цветности $(x_R, y_R), (x_G, y_G), (x_B, y_B)$ вершин ТЦО в соответствии с математической моделью (1), (2) определяются из следующих соотношений [7]:

$$x_R = X_r / (X_r + Y_r + Z_r), \quad y_R = Y_r / (X_r + Y_r + Z_r),$$

$$x_G = X_g / (X_g + Y_g + Z_g), \quad y_G = Y_g / (X_g + Y_g + Z_g),$$

$$x_B = X_b / (X_b + Y_b + Z_b), \quad y_B = Y_b / (X_b + Y_b + Z_b).$$

Площадь ТЦО дисплея как геометрической фигуры может быть вычислена через значение его полупериметра p и значения длин сторон треугольника l_{BG}, l_{BR}, l_{GR} по формуле Герона:

$$S_{\max} = \sqrt{p(p-l_{BG})(p-l_{BR})(p-l_{GR})}, \quad (3)$$

где

$$p = (l_{BG} + l_{BR} + l_{GR})/2, \quad l_{BG} = \sqrt{(x_B - x_G)^2 + (y_B - y_G)^2},$$

$$l_{BR} = \sqrt{(x_B - x_R)^2 + (y_B - y_R)^2},$$

$$l_{GR} = \sqrt{(x_G - x_R)^2 + (y_G - y_R)^2}.$$

2. Оценка параметров математической модели цветовоспроизведения дисплея

При известных значениях $(x_R, y_R), (x_G, y_G), (x_B, y_B)$ координат цветности вершин ТЦО оценки коэффициентов математической модели цветовоспроизведения дисплея могут быть получены из следующих выражений [4]:

$$\hat{X}_r = \frac{x_R}{y_R} \hat{Y}_r, \quad \hat{X}_g = \frac{x_G}{y_G} \hat{Y}_g, \quad \hat{X}_b = \frac{x_B}{y_B} \hat{Y}_b, \quad (4)$$

$$\hat{Y}_r = 1 - \hat{Y}_g - \hat{Y}_b, \quad \hat{Y}_g = \frac{\frac{x_W}{y_W} - \frac{x_R}{y_R} + \left(\frac{x_R}{y_R} - \frac{x_B}{y_B}\right) \hat{Y}_b}{\frac{x_G}{y_G} - \frac{x_R}{y_R}}, \quad (5)$$

$$\hat{Y}_b = \frac{\frac{1-x_W}{y_W} + \frac{x_R-1}{y_R} + \frac{x_W y_R - y_W x_R}{x_G y_R - y_G x_R} \frac{y_G}{y_W} \left(\frac{1-x_R}{y_R} + \frac{x_G-1}{y_G}\right)}{\frac{x_R y_B - y_R x_B}{x_G y_R - y_G x_R} \frac{y_G}{y_B} \left(\frac{1-x_G}{y_G} + \frac{x_R-1}{y_R}\right) + \frac{1-x_B}{y_B} + \frac{x_R-1}{y_R}}, \quad (6)$$

$$\hat{Z}_r = \hat{Y}_r \left(\frac{1-x_R-y_R}{y_R}\right), \quad \hat{Z}_g = \hat{Y}_g \left(\frac{1-x_G-y_G}{y_G}\right), \quad (7)$$

$$\hat{Z}_b = \hat{Y}_b \left(\frac{1-x_B-y_B}{y_B}\right),$$

где (x_W, y_W) — координаты цветности точки белого цвета, воспроизводимого на дисплее.

Таким образом, для вычисления коэффициентов (4)–(7) профиля дисплея с максимальной площадью ТЦО должны быть:

- определены координаты цветности $(x_R, y_R), (x_G, y_G), (x_B, y_B)$ вершин ТЦО, вписанного в спектральный локус и обладающего максимально возможной площадью S_{\max} ;

- заданы координаты цветности (x_W, y_W) точки белого цвета, воспроизводимого на дисплее, соответствующие одному из принятых в МКО колориметрических стандартов, например D-75, D-65, D-55, D-50 и др.

Определение координат цветности $(x_R, y_R), (x_G, y_G), (x_B, y_B)$ вершин ТЦО основано на решении задачи интерполяции спектрального локуса.

3. Задача интерполяции для линии спектральных цветностей

Международной комиссией по освещению в 1931 г. установлены значения удельных координат $\bar{x}(\lambda), \bar{y}(\lambda), \bar{z}(\lambda)$ спектральных излучений посто-

янной мощности для длин волн $\lambda[i]$ монохроматических цветов в видимом человеком спектральном диапазоне от 380 до 780 нм с шагом $\Delta\lambda = 5$ нм, $i = 0, 1, \dots, N_\lambda - 1$. Общее число N_λ таблично заданных отсчетов удельных координат цвета $\bar{x}(\lambda)$, $\bar{y}(\lambda)$, $\bar{z}(\lambda)$ составляет 81. Соответствующая таблица данных также утверждена рядом отечественных стандартов (в частности, в ГОСТ 23198—94 "Лампы электрические. Методы измерения спектральных и цветовых характеристик").

По этим табличным данным с использованием соотношений (2) для каждого значения длины волны $\lambda[i]$ могут быть получены соответствующие значения координат цветности $(x[i], y[i])$ монохроматических цветов. Оценки координат цветности $(x[i], y[i])$ монохроматических цветов для стандартов МКО 1931 г. и 1964 г. (индекс 10) приведены в табл. 1. Первое и последнее значения координат в табл. 1 совпадают, так как форма спектрального локуса описывается замкнутой кривой с соединенными прямой линией — линией пурпурных цветов (ЛПЦ) — координатами цветности, соответствующими длинам волн 380 и 780 нм.

Набор координат цветности $\{x[i], y[i]\}$ для $\lambda \in [380, 780]$ образует (рис. 1) на XY -плоскости точно заданный с шагом $\Delta\lambda$ контур, интерполяция значений отсчетов которого позволяет получить замкнутую кривую спектральных цветности для монохроматических цветов, ограничивающую область цветового локуса. На рис. 1 отмечены спектральные диапазоны следующих цветов: Ф — фиолетового, С — синего, Г — голубого, Ж-З — желто-зеленого, Ж — желтого, О — оранжевого, К — красного.

Для решения задачи нахождения (x, y) -координат цветности вершин ТЦО с максимальной площадью,

вписанного в область цветового локуса, необходимо предварительно определить методику интерполяции набора точек $\{x[i], y[i]\}$. Интерполяция набора точек $\{x[i], y[i]\}$ должна обеспечивать возможность получения промежуточных значений (x, y) -координат цветности монохроматических цветов с произвольно заданным шагом изменения длины волны, а не с фиксированным шагом в 5 нм, как в стандартах МКО, соединение которых позволит получить "гладкую" кривую спектрального локуса.

Задача интерполяции таблично заданных спектральных кривых имеет важное практическое значение [8] для повышения точности выполнения колориметрических расчетов световых величин при различном спектральном составе воспринимаемого излучения. Ряд результатов интерполяции спектральных кривых образует основу государственных стандартов, действующих в Российской Федерации.

В частности, в ГОСТ 8.332—78 "Световые измерения. Значения относительной спектральной световой эффективности монохроматического излучения для дневного зрения" приведены результаты интерполяции через 1 нм кривой относительной спектральной световой эффективности монохроматического излучения ("кривой видности" [8]) для длин волн в видимом диапазоне 380...780 нм.

Основным требованием для методики интерполяции набора точек $(x[i], y[i])$ является непрерывность первой производной получаемой интерполирующей функции в заданных узловых точках $(x[i], y[i])$, за исключением крайних точек $(x[0], y[0])$ и $(x[80], y[80])$.

Одним из эффективных подходов к решению задачи интерполяции является подход на основе построения сплайн-функций. В рамках предметной области

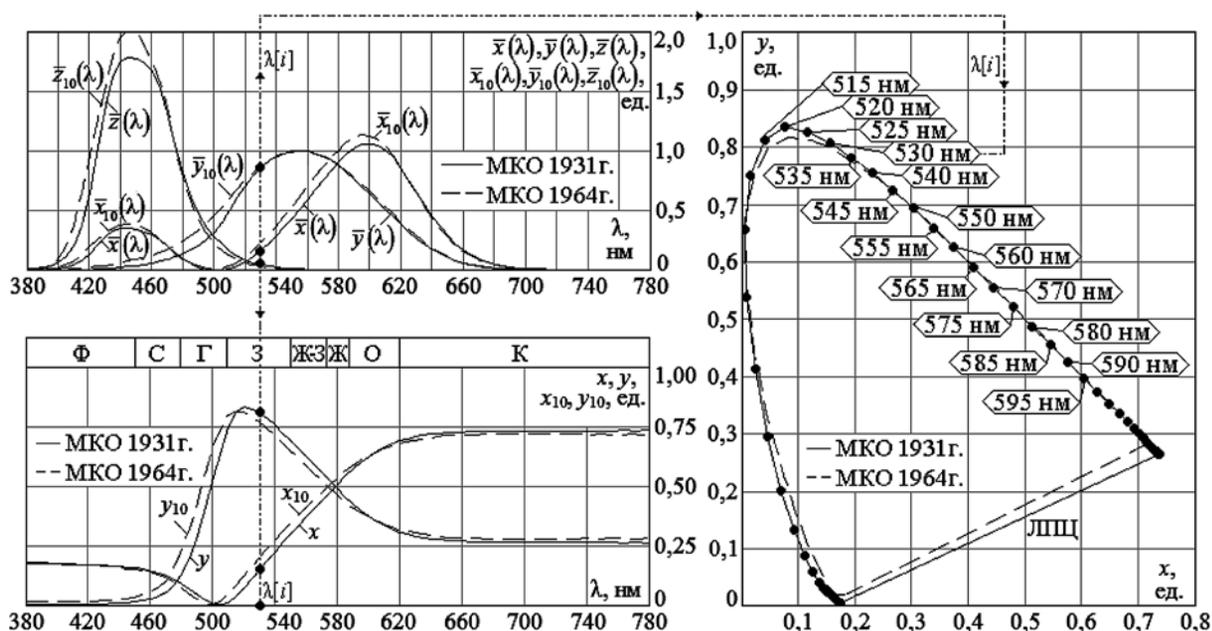


Рис. 1. Семейство колориметрических графиков, принятых МКО: сплошная линия — МКО 1931 г.; штриховая линия — МКО 1964 г.

Таблица оценок (x, y)-координат цветности монохроматических цветов для стандартов МКО 1931 г. и 1964 г.

λ, нм	МКО 1931 г.		МКО 1964 г.		λ, нм	МКО 1931 г.		МКО 1964 г.	
	x, ед.	y, ед.	x ₁₀ , ед.	y ₁₀ , ед.		x, ед.	y, ед.	x ₁₀ , ед.	y ₁₀ , ед.
380	0,174112	0,004964	0,181406	0,019274	585	0,544787	0,454434	0,565444	0,434556
385	0,174008	0,004981	0,180776	0,019661	590	0,575151	0,424232	0,58996	0,410040
390	0,173801	0,004915	0,180347	0,019317	595	0,602933	0,396497	0,611597	0,388403
395	0,173560	0,004923	0,179457	0,019056	600	0,627037	0,372491	0,630629	0,369371
400	0,173337	0,004797	0,178390	0,018707	605	0,648233	0,351395	0,647127	0,352873
405	0,173021	0,004775	0,177122	0,018402	610	0,665764	0,334011	0,661224	0,338776
410	0,172577	0,004799	0,175488	0,018134	615	0,680079	0,319747	0,673055	0,326945
415	0,172087	0,004833	0,173231	0,017806	620	0,691504	0,308342	0,68266	0,317340
420	0,171407	0,005102	0,170634	0,017849	625	0,700606	0,299301	0,689759	0,310241
425	0,170301	0,005789	0,167902	0,018708	630	0,707918	0,292027	0,695483	0,304517
430	0,168878	0,00690	0,165027	0,020283	635	0,714032	0,285929	0,700989	0,299011
435	0,166895	0,008556	0,162170	0,022487	640	0,719033	0,280935	0,705873	0,294127
440	0,164412	0,010858	0,159022	0,025725	645	0,723032	0,276948	0,710249	0,289751
445	0,161105	0,013793	0,155391	0,030017	650	0,725992	0,274008	0,713713	0,286287
450	0,156641	0,017705	0,151001	0,036439	655	0,728272	0,271728	0,715619	0,284381
455	0,150985	0,022740	0,145945	0,045217	660	0,729969	0,270031	0,716790	0,283210
460	0,143960	0,029703	0,138922	0,058920	665	0,731089	0,268911	0,717887	0,282113
465	0,135503	0,039879	0,129520	0,077870	670	0,731993	0,268007	0,718736	0,281264
470	0,124118	0,057803	0,115180	0,109040	675	0,732719	0,267281	0,719341	0,280659
475	0,109594	0,086843	0,095732	0,159090	680	0,733417	0,266583	0,719765	0,280235
480	0,091294	0,132702	0,072777	0,229239	685	0,734047	0,265953	0,720021	0,279979
485	0,068706	0,200723	0,045194	0,327537	690	0,734390	0,26561	0,720152	0,279848
490	0,045391	0,294976	0,020987	0,440113	695	0,734592	0,265408	0,720300	0,279700
495	0,023460	0,412703	0,007302	0,562523	700	0,734687	0,265313	0,720346	0,279654
500	0,008168	0,538423	0,005586	0,674543	705	0,734692	0,265308	0,720284	0,279716
505	0,003859	0,654823	0,021874	0,752578	710	0,734678	0,265322	0,720297	0,279703
510	0,013870	0,750186	0,049540	0,802302	715	0,734668	0,265332	0,720174	0,279826
515	0,038852	0,812016	0,085024	0,816976	720	0,734668	0,265332	0,719960	0,280040
520	0,074302	0,833803	0,125236	0,810194	725	0,734672	0,265328	0,719885	0,280115
525	0,114161	0,826207	0,166408	0,792172	730	0,734694	0,265306	0,719697	0,280303
530	0,154722	0,805864	0,207057	0,766282	735	0,734754	0,265246	0,719090	0,280910
535	0,192876	0,781629	0,243642	0,739873	740	0,734824	0,265176	0,718529	0,281471
540	0,229620	0,754329	0,278588	0,711300	745	0,734568	0,265432	0,717742	0,282258
545	0,265775	0,724324	0,313230	0,681278	750	0,734513	0,265487	0,719198	0,280802
550	0,301604	0,692308	0,347296	0,650090	755	0,734375	0,265625	0,717742	0,282258
555	0,337363	0,658848	0,381161	0,618164	760	0,734513	0,265487	0,715909	0,284091
560	0,373102	0,624451	0,414213	0,585787	765	0,735849	0,264151	0,714286	0,285714
565	0,408736	0,589607	0,446924	0,553076	770	0,734513	0,265487	0,722222	0,277778
570	0,444062	0,554714	0,479038	0,520962	775	0,737500	0,262500	0,718750	0,281250
575	0,478775	0,520202	0,509641	0,490359	780	0,736842	0,263158	0,717391	0,282609
580	0,512486	0,486591	0,538560	0,461440	380	0,174112	0,004964	0,181406	0,019274

колориметрии практический интерес представляют три различных способа конструирования кривой, интерполирующей линию спектральных цветностей:

1) интерполяция в пространстве удельных координат цвета — интерполяция осуществляется по трем таблично заданным последовательностям узловых точек $(\lambda[i], \bar{x}(\lambda[i]))$, $(\lambda[i], \bar{y}(\lambda[i]))$, $(\lambda[i], \bar{z}(\lambda[i]))$ независимо друг от друга, с последующим использованием выражений (2) для получения значений (x, y) -координат цветности монохроматических цветов;

2) интерполяция в пространстве (x, y) -координат цветности — независимо друг от друга осуществляется интерполяция по двум последовательностям узловых точек $(\lambda[i], x[i])$, $(\lambda[i], y[i])$ табл. 1;

3) интерполяция таблично заданной линии спектральных цветностей, заданной множеством узловых точек $(x[i], y[i])$ как некоторой кривой, обладающей характерными признаками формы, с определением соответствующего геометрического места точек на XY -плоскости.

4. Решение задачи интерполяции для линии спектральных цветностей

Для решения задачи интерполяции в рамках первого и второго способов могут быть использованы стандартные методы [9] построения интерполирующих сплайнов, в частности методы построения кубических сплайнов. Основным недостатком применения кубических сплайнов заключается в существенной "колебательности" формы кривой интерполирующей функции. Свойство нежелательной "колебательности" проявляется на участках, где исходные (таблично заданные) узловые точки координат цветности с очевидностью могли бы быть соединены практически прямой линией.

В качестве альтернативного подхода предлагается оригинальный метод интерполяции линии спектральных цветностей как некоторой траектории, которую описывает точка с координатами цветности на XY -плоскости при дискретном изменении длины волны $\lambda[i]$ от 380 до 780 нм. Для интерполяции предлагаются кривые Безье [9], совмещенные друг с другом в узлах интерполяции, образующие в совокупности сплайн Безье. Используются квадратичные кривые Безье и кубические кривые Безье.

Интерполирующая функция, используемая для представления формы спектрального локуса на участке, соответствующем длинам волн от $\lambda[i]$ до $\lambda[i+1]$, на базе квадратичной кривой Безье, имеет следующий вид:

$$\begin{cases} x(\lambda) = t_2^2(\lambda)x[i] + 2t_1(\lambda)t_2(\lambda)x_Q[i] + t_1^2(\lambda)x[i+1] \\ y(\lambda) = t_2^2(\lambda)y[i] + 2t_1(\lambda)t_2(\lambda)y_Q[i] + t_1^2(\lambda)y[i+1], \end{cases} \quad (8)$$

$$\text{где } t_1(\lambda) = \frac{\lambda - \lambda[i]}{\lambda[i+1] - \lambda[i]}; \quad t_2(\lambda) = 1 - t_1(\lambda) = \frac{\lambda[i+1] - \lambda}{\lambda[i+1] - \lambda[i]};$$

$x_Q[i], y_Q[i]$ — координаты управляющей точки $Q[i]$ на XY -плоскости, определяющей свойства интерполирующей функции на каждом интервале.

Интерполирующая функция, используемая для представления формы спектрального локуса на участке, соответствующем длинам волн от $\lambda[i]$ до $\lambda[i+1]$, на базе кубической кривой Безье, имеет следующий вид:

$$\begin{cases} x(\lambda) = t_2^3(\lambda)x[i] + 3t_1(\lambda)t_2^2(\lambda)x_{P_1}[i] + \\ + 3t_1^2(\lambda)t_2(\lambda)x_{P_2}[i] + t_1^3(\lambda)x[i+1] \\ y(\lambda) = t_2^3(\lambda)y[i] + 3t_1(\lambda)t_2^2(\lambda)y_{P_1}[i] + \\ + 3t_1^2(\lambda)t_2(\lambda)y_{P_2}[i] + t_1^3(\lambda)y[i+1], \end{cases} \quad (9)$$

где $x_{P_1}[i], y_{P_1}[i], x_{P_2}[i], y_{P_2}[i]$ — координаты управляющих точек $P_1[i]$ и $P_2[i]$ на XY -плоскости, определяющих свойства интерполирующей функции на каждом интервале.

Расчет координат управляющих точек $Q[i], P_1[i]$ и $P_2[i]$ предполагает предварительное вычисление в каждой узловой точке $(x[i], y[i])$ угла наклона линии, касательной к искомой интерполирующей кривой, по формуле

$$\alpha[i] = \text{angle}(x[i+1] - x[i-1], y[i+1] - y[i-1]), \quad (10)$$

$$i = 1, 2, \dots, N_\lambda - 2,$$

где функция $\text{angle}(x, y)$ вычисляет значение угла (в радианах) направляющего вектора, начинающегося в начале системы координат XY -плоскости, и имеющего конец в точке с координатами (x, y) , $\alpha[i] \in [0, 2\pi]$. На i -м интервале интерполяции касательная линия КН (касательная начальная) с углом наклона $\alpha[i]$ к оси абсцисс проводится через точку $(x[i], y[i])$ начала интервала интерполяции, касательная линия КК (касательная конечная) с углом наклона $\alpha[i+1]$ к оси абсцисс проводится через точку $(x[i+1], y[i+1])$ конца интервала интерполяции.

Принцип выбора типа кривой Безье и методики ее построения проиллюстрирован на рис. 2 на примере двух участков графика спектрального локуса.

Узловой точке $(x[i], y[i])$ на рис.2, а, через которую проведена касательная линия КН, соответствует длина волны $\lambda[i] = 515$ нм. Линия КН представляет собой линию, параллельную линии (показана на рис. 2, а, штриховой линией), соединяющей точки $(x[i-1], y[i-1])$ для $\lambda[i-1] = 510$ нм и $(x[i+1], y[i+1])$ для $\lambda[i+1] = 520$ нм. Линия КК, касающаяся цветового локуса в точке $(x[i+1], y[i+1])$ для $\lambda[i+1] = 520$ нм, представляет собой линию, параллельную линии (показана на рис. 2, а, штриховой линией), соединяющей точки $(x[i], y[i])$ для $\lambda[i] = 515$ нм и $(x[i+2], y[i+2])$ для $\lambda[i+2] = 525$ нм. На рис. 2 касательные линии КН и КК проведены штрих-пунктирными линиями.

Узловой точке $(x[i], y[i])$ на рис.2, б, через которую проведена касательная линия КН, соответствует длина волны $\lambda[i] = 390$ нм. Линия КН представляет собой линию, параллельную линии (показана на рис. 2, б, штриховой линией), соединяющей точки $(x[i-1], y[i-1])$ для $\lambda[i-1] = 385$ нм и $(x[i+1], y[i+1])$ для $\lambda[i+1] = 395$ нм. Линия КК, касающаяся

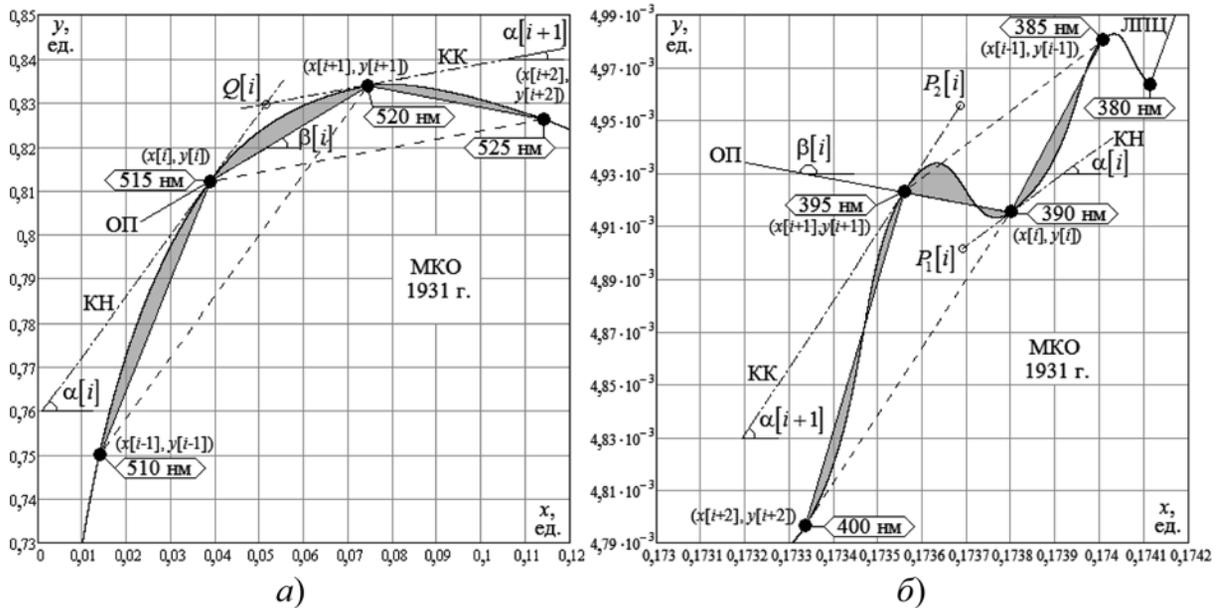


Рис. 2. Принцип выбора типа кривой Безье:

а — квадратичная кривая; б — кубическая кривая для интерполяции спектрального локуса на i -м участке

цветового локуса в точке $(x[i+1], y[i+1])$ для $\lambda[i+1] = 395$ нм, представляет собой линию, параллельную линии (показана на рис. 2, б, штриховой линией), соединяющей точки $(x[i], y[i])$ для $\lambda[i] = 390$ нм и $(x[i+2], y[i+2])$ для $\lambda[i] = 400$ нм.

Для выбора типа кривой Безье также вычисляется угол наклона опорной прямой (ОП) между узловыми точками $(x[i], y[i])$ и $(x[i+1], y[i+1])$:

$$\beta[i] = \text{angle}(x[i+1] - x[i], y[i+1] - y[i]), \quad (11)$$

$$i = 0, 1, \dots, N_\lambda - 2.$$

Для крайних узловых точек спектрального локуса в выражениях (10), (11) фиксируются граничные условия:

$$\alpha[0] = \beta[0], \quad \alpha[N_\lambda - 1] = \beta[N_\lambda - 2].$$

Таким образом, на каждом i -м интервале интерполяции, соответствующем длине волны $\lambda[i]$, оказываются заданными три параметра, каждый из которых может изменяться в пределах от 0 до 2π :

- угол $\alpha[i]$ наклона касательной линии КН к искомой кривой в начале интервала;
- угол $\beta[i]$ наклона ОП между узловыми точками $(x[i], y[i])$ и $(x[i+1], y[i+1])$;
- угол $\alpha[i+1]$ наклона касательной линии КК к искомой кривой в конце интервала.

Взаимные соотношения между параметрами $\alpha[i]$, $\beta[i]$, $\alpha[i+1]$ определяют выбор вида кривой Безье для интерполяции участка кривой на данном интервале:

- квадратичная кривая Безье с одной управляющей точкой $Q[i]$;
- кубическая кривая Безье с двумя управляющими точками $P_1[i]$ и $P_2[i]$.

Управляющая точка $P_1[i]$ лежит на касательной линии КН на расстоянии $L_1[i]$ от узловой точки с координатами $(x[i], y[i])$ при движении вдоль линии КН в направлении соответствующего ей направляющего вектора.

Управляющая точка $P_2[i]$ лежит на касательной линии КК на расстоянии $L_2[i]$ от точки $(x[i+1], y[i+1])$ при движении вдоль линии КК против направления соответствующего ей направляющего вектора. Управляющие точки на рис. 2 отмечены мнемознаком "о", узловые точки — мнемознаком "•".

Координаты управляющих точек $P_1[i]$ и $P_2[i]$, используемых для интерполяции формы спектрального локуса кубической кривой (9), вычисляются по следующим формулам:

$$x_{P_1}[i] = x[i] + L_1[i] \cos(\alpha[i]), \quad (12)$$

$$y_{P_1}[i] = y[i] + L_1[i] \sin(\alpha[i]),$$

$$x_{P_2}[i] = x[i+1] - L_2[i] \cos(\alpha[i+1]), \quad (13)$$

$$y_{P_2}[i] = y[i+1] - L_2[i] \sin(\alpha[i+1]).$$

В случае, когда управляющие точки $P_1[i]$ и $P_2[i]$ оказываются по разные стороны от ОП, проведенной через узловые точки $(x[i], y[i])$ и $(x[i+1], y[i+1])$, управляющие точки фиксируются в указанных положениях (12), (13) и с выбранным шагом интерполяции $\Delta\lambda$ на данном интервале строится кубическая кривая Безье.

Если управляющие точки $P_1[i]$ и $P_2[i]$ окажутся по одну сторону (обе сверху, обе снизу, обе слева или обе справа) от ОП, то строить кубическую кривую Безье на данном интервале интерполяции нецелесообразно. В этом случае задается одна управляющая

точка $Q[i]$, соответствующая точке пересечения касательных линий КН и КК, и строится квадратичная кривая Безье.

Координаты управляющей точки $Q[i]$, используемой для интерполяции формы спектрального локуса квадратичной кривой (8), вычисляются по следующим формулам:

$$x_Q[i] = \begin{cases} \frac{b[i+1] - b[i]}{\operatorname{tg}(\alpha[i]) - \operatorname{tg}(\alpha[i+1])}, \operatorname{tg}(\alpha[i]) \neq \operatorname{tg}(\alpha[i+1]) \\ (x[i] + x[i+1])/2, \operatorname{tg}(\alpha[i]) = \operatorname{tg}(\alpha[i+1]), \end{cases} \quad (14)$$

$$\begin{aligned} y_Q[i] &= \operatorname{tg}(\alpha[i])x_Q[i] + b[i], \\ b[i] &= y[i] - \operatorname{tg}(\alpha[i])x[i]. \end{aligned} \quad (15)$$

Расположение управляющих точек (12), (13) и (14), (15) на касательных линиях КН и КК обеспечивает совмещение кривых Безье в узловых точках так, чтобы обеспечивалась непрерывность первой производной интерполирующей функции.

Решающая функция, обеспечивающая выбор в пользу квадратичной (8) или кубической (9) интерполированной кривой на i -м интервале интерполяции, имеет вид

$$\begin{aligned} n[i] &= (\operatorname{tg}(\beta[i])(x_{P_1}[i] - x[i]) + y[i] - y_{P_1}[i]) \times \\ &\times (\operatorname{tg}(\beta[i])(x_{P_2}[i] - x[i]) + y[i] - y_{P_2}[i]). \end{aligned} \quad (16)$$

При $n[i] > 0$ выбор происходит в пользу квадратичной кривой Безье, при $n[i] \leq 0$ — в пользу кубической кривой Безье.

В итоге система интерполирующих функций, определяющая форму спектрального локуса на XY -плоскости, имеет вид

$$\begin{cases} x(\lambda) = \begin{cases} t_2^2(\lambda)x[i] + 2t_1(\lambda)t_2(\lambda)x_Q[i] + \\ + t_1^2(\lambda)x[i+1], n[i] > 0 \\ t_2^3(\lambda)x[i] + 3t_1(\lambda)t_2^2(\lambda)x_{P_1}[i] + \\ + 3t_1^2(\lambda)t_2(\lambda)x_{P_2}[i] + t_1^3(\lambda)x[i+1], n[i] \leq 0 \end{cases} \\ y(\lambda) = \begin{cases} t_2^2(\lambda)y[i] + 2t_1(\lambda)t_2(\lambda)y_Q[i] + \\ + t_1^2(\lambda)y[i+1], n[i] > 0 \\ t_2^3(\lambda)y[i] + 3t_1(\lambda)t_2^2(\lambda)y_{P_1}[i] + \\ + 3t_1^2(\lambda)t_2(\lambda)y_{P_2}[i] + t_1^3(\lambda)y[i+1], n[i] \leq 0, \end{cases} \end{cases} \quad (17)$$

$$i = 0, 1, \dots, N_\lambda - 2.$$

Задание расстояний $L_1[i]$ и $L_2[i]$ оказывает существенное влияние на изгибы формы кривой интерполирующей функции на интервалах, где формируется кубическая кривая Безье. Для решения задачи интерполяции линии спектральных цветностей в предлагаемом методе используются следующие выражения:

$$L_1[i] = \frac{L[i-1]L[i]}{L[i-1] + L[i+1]}, \quad L_2[i] = \frac{L[i+1]L[i]}{L[i-1] + L[i+1]}.$$

Величина $L[i]$ представляет собой евклидово расстояние между точками $(x[i], y[i])$ и $(x[i+1], y[i+1])$ на XY -плоскости. Для начального и конечного интервалов интерполяции спектрального локуса используют следующие расстояния:

$$\begin{aligned} L_1[0] &= L_2[0] = L[0]/2, \\ L_1[N_\lambda - 2] &= L_2[N_\lambda - 2] = L[N_\lambda - 2]/2. \end{aligned}$$

На рис. 2 соединение узловых точек выполнено прямыми сплошными линиями, позволяющими наглядно оценить эффективность предложенного метода интерполяции по сравнению с методом линейной интерполяции.

Качественную оценку результатов интерполяции для различных способов конструирования кривой, интерполирующей форму спектрального локуса, можно получить путем анализа графиков функций в увеличенном масштабе отображения. Интерполирующие кривые спектрального локуса, полученные тремя различными способами, приведены на рис. 3.

Наибольший интерес представляет анализ спектральных интервалов, где расположение таблично заданных в нормативной документации узловых точек спектрального локуса соответствует участкам с максимальной кривизной функции:

- на начальном участке в спектральном диапазоне от 380 до 420 нм;
- на участке, соответствующем длинам волн от 515 до 525 нм.

Анализ рис. 3 показывает, что применение методов сплайн-интерполяции в пространстве координат цвета (способ 1) и координат цветности (способ 2) приводит к выраженной "колебательности" формы интерполирующей функции на начальном участке 380...415 нм спектрального диапазона видимых длин волн. Такое свойство интерполирующей функции представляется неоправданным при интерполяции формы спектрального локуса. Для спектрального интервала 515...525 нм, где изменения кривизны формы спектрального локуса существенно меньше, способ 1 и способ 2 обеспечивают приемлемое для практики качество интерполяции.

Количественную оценку точности интерполяции можно получить путем анализа отклонений Δx , Δy полученных кривых от таблично установленных значений (x, y) спектрального локуса в точках, заданных вне шага $\Delta\lambda = 5$ нм и априори не использованных в способах интерполяции. В частности, в ГОСТ 23198—94 приведено четыре таких "контрольных" точки: на длинах волн 436, 546, 578 и 589 нм.

Результаты интерполяции, полученные численными методами по способам 1—3 при выборе шага интерполяции $\Delta\lambda = 0,0002$ нм, т. е. когда между каждой парой узловых точек рассчитывалось по 25 000 точек координат цветности интерполирующей функции, приведены в табл. 2. Величина d рассчитана и приведена в табл. 2 как евклидово расстояние $d = \sqrt{\Delta x^2 + \Delta y^2}$.

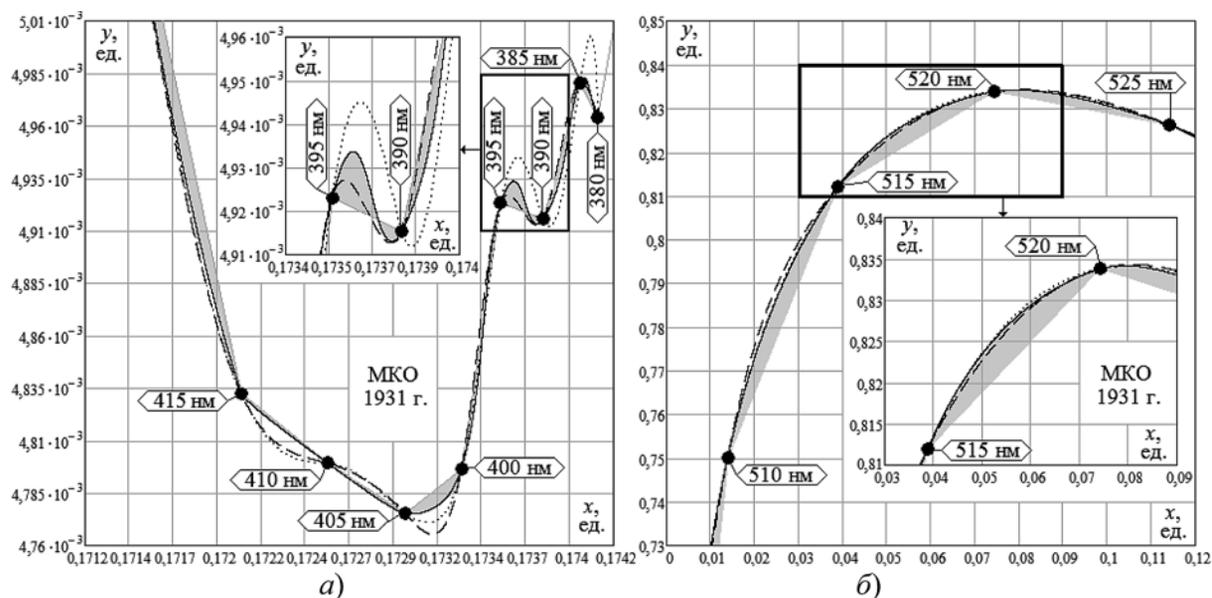


Рис. 3. Результаты интерполяции спектрального локуса:

a — в диапазоне 380...420 нм; *б* — в диапазоне 510...525 нм (штриховая кривая получена на основе методики интерполяции в пространстве координат цвета, пунктирная кривая — интерполяция в пространстве координат цветности, сплошная кривая — интерполяция сплайнами Безье)

Нетрудно видеть, что все рассмотренные способы интерполяции обеспечивают соизмеримые характеристики точности представления спектрального локуса и в этом смысле одинаково пригодны для решения задачи интерполяции линии спектральных цветностей. Тем не менее следует отметить лучшую точность интерполяции

для способа 3, основанного на сплайнах Безье, в точке, соответствующей длине волны 436 нм, где проявляется существующая кривизна исходной формы спектрального локуса, что особенно важно, поскольку одна из вершин ТЦО максимальной площади оказывается в окрестности от этого значения.

Таблица 2

Количественные оценки отклонений значений интерполирующей функции от таблично заданных значений координат цветности спектрального локуса в контрольных точках, выборочно заданных вне стандартного шага 5 нм

Способ интерполяции	Параметр	Длина волны λ в контрольной точке, нм			
		436	546	578	589
Табличные данные	x	0,1664468788	0,2729572515	0,4991508363	0,5692571629
	y	0,0089607742	0,7180626962	0,4998872841	0,4301020072
1 (по координатам цвета)	Δx	$3,163 \cdot 10^{-7}$	$3,427 \cdot 10^{-6}$	$8,306 \cdot 10^{-7}$	$8,591 \cdot 10^{-7}$
	Δy	$3,911 \cdot 10^{-7}$	$5,168 \cdot 10^{-6}$	$1,852 \cdot 10^{-7}$	$7,466 \cdot 10^{-8}$
	d	$5,030 \cdot 10^{-7}$	$6,201 \cdot 10^{-6}$	$8,510 \cdot 10^{-7}$	$8,654 \cdot 10^{-7}$
2 (по координатам цветности)	Δx	$2,125 \cdot 10^{-6}$	$2,307 \cdot 10^{-7}$	$2,937 \cdot 10^{-7}$	$6,875 \cdot 10^{-7}$
	Δy	$2,287 \cdot 10^{-6}$	$2,254 \cdot 10^{-8}$	$6,960 \cdot 10^{-7}$	$1,255 \cdot 10^{-6}$
	d	$3,122 \cdot 10^{-6}$	$2,316 \cdot 10^{-7}$	$7,554 \cdot 10^{-7}$	$1,431 \cdot 10^{-6}$
3 (по кривой локуса сплайном Безье)	Δx	$1,538 \cdot 10^{-6}$	$9,057 \cdot 10^{-8}$	$5,575 \cdot 10^{-7}$	$1,292 \cdot 10^{-7}$
	Δy	$1,639 \cdot 10^{-6}$	$5,311 \cdot 10^{-7}$	$2,076 \cdot 10^{-7}$	$7,331 \cdot 10^{-7}$
	d	$2,248 \cdot 10^{-6}$	$5,388 \cdot 10^{-7}$	$5,949 \cdot 10^{-7}$	$7,444 \cdot 10^{-7}$

5. Оценка значений координат цветности вершин треугольника с максимальным цветовым охватом

В результате решения задачи интерполяции образуется множество точек координат цветности, соединение которых позволяет получить замкнутую линию спектральных цветностей (16), (17). Крайние точки линии спектральных цветностей, соответствующие координатам цветности для длин волн $\lambda[0] = 380$ и $\lambda[N-1] = 780$ нм, соединяются прямой линией — линией пурпурных цветов.

Для определения координат цветности вершин ТЦО с максимальной площадью, вписанного в цветовой locus, предложен алгоритм, заключающийся в пошаговом улучшении точности интерполяции формы локуса и уточнении получаемых оценок координат вершин ТЦО. Помимо экономии машинного времени вычислений алгоритм позволяет оценить влияние точности используемого метода интерполяции на основе кривых Безье на получаемые оценки значений координат цветности вершин ТЦО.

Алгоритм поиска значений координат цветности вершин ТЦО предполагает, что все точки, находящиеся внутри ТЦО и вдоль его границ, принадлежат видимой области цветов спектрального локуса. Сначала осуществляется определение треугольника с максимальной площадью цветового охвата по исходным данным, таблично заданным с шагом длины волны в $\Delta\lambda = 5$ нм. Таким образом формируется начальное приближение. Найденным вершинам треугольника на линии спектральных цветностей соответствуют точки с координатами (x_R, y_R) , (x_G, y_G) , (x_B, y_B) и длины волн $\lambda_R, \lambda_G, \lambda_B$. Далее в алгоритме проводится поиск уточненных положений вершин треугольника по данным, полученным в результате интерполяции формы спектрального локуса с меньшим шагом, $\Delta\lambda = 1$ нм, уже в окрестности точек, определенных на предыдущем шаге, т. е.: $\lambda_R \pm 10\Delta\lambda$, $\lambda_G \pm 10\Delta\lambda$, $\lambda_B \pm 10\Delta\lambda$.

Уточненный набор параметров $\lambda_R, \lambda_G, \lambda_B$ используется в дальнейшем в алгоритме в качестве нового приближения, и итерация повторяется до значения шага $\Delta\lambda = 0,0002$ нм. Значение шага $\Delta\lambda = 0,0002$ нм принято в алгоритме в качестве минимально допустимого, после которого приращение вычисляемой площади ТЦО по формуле (3) становится соизмеримым с точностью представления результатов вычислений в примененной среде моделирования Mathcad 15.0, а именно 10^{-15} .

Треугольники цветового охвата, получаемые в процессе моделирования и содержащие точки, координаты которых выходят за пределы формы спектрального локуса, в алгоритме отбраковывались. Появление таких ТЦО обусловлено существенной нелинейностью (см. рис. 3, а) линии спектральных цветностей в диапазоне длин волн 380...410 нм, при которой сторона ТЦО, соединяющая вершины (x_B, y_B) и (x_R, y_R) , частично окажется вне области видимых цветов.

В результате моделирования с применением интерполяции таблично заданной линии спектральных цветностей сплайнами Безье получены следующие оценки координат цветности вершин треугольника с максимальным цветовым охватом:

- $x_R = 0,7368421053$, $y_R = 0,2631578947$, $x_G = 0,060196208$, $y_G = 0,8294588866$, $x_B = 0,1676393663$, $y_B = 0,0079187467$ (МКО 1931 г.);

- $x_R = 0,7173913043$, $y_R = 0,2826086957$, $x_G = 0,0620826166$, $y_G = 0,8110570357$, $x_B = 0,1631430859$, $y_B = 0,0216556487$ (МКО 1964 г.).

Оценки (4)—(7) коэффициентов профилей дисплеев, соответствующих принятым в МКО колориметрическим стандартам D-75, D-65, D-55, D-50 задания точки белого цвета для МКО 1931 г. и для МКО 1964 г., приведены в табл. 3 и табл. 4 соответственно.

Любой ТЦО, образованный смешением кодов RGB, для дисплеев с 8-битным двоичным представлением каждого кода основного цвета содержит в себе $2^8 \cdot 2^8 \cdot 2^8 = 2^{24}$ координат цветности цветов и оттенков, воспроизводимых на экране. Плотность заполнения площади ТЦО координатами цветности тем выше, чем меньше площадь это треугольника, т. е. с уменьшением площади ТЦО уменьшается расстояние [6, 10] между соседними координатами цветности.

Практические расчеты площадей ТЦО для некоторых промышленно изготавливаемых дисплеев и для дисплея с максимальным цветовым охватом следующие:

- площадь ТЦО для жидкокристаллического дисплея производства AU Optronics Corporation, модель G104SN03 V.0 с газоразрядным подсветом, равна $0,07665$ ед.²;

- площадь ТЦО для жидкокристаллического дисплея производства Chi Mei Optoelectronics Corporation, модель R190E1-L01 с газоразрядным подсветом равна $0,116427$ ед.²;

- площадь ТЦО с максимальным цветовым охватом для стандарта МКО 1931 г. равна $S_{\max} = 0,247523298979138$ ед.²;

- площадь ТЦО с максимальным цветовым охватом для стандарта МКО 1964 г. равна $S_{\max} = 0,231948174865851$ ед.².

Значения координат цветности (x_R, y_R) , (x_G, y_G) , (x_B, y_B) вершин ТЦО промышленных дисплеев, необходимые для расчетов (3), взяты из технической документации:

- модель G104SN03 V.0: $x_R = 0,57$, $y_R = 0,32$, $x_G = 0,3$, $y_G = 0,56$, $x_B = 0,145$, $y_B = 0,13$;

- модель R190E1-L01: $x_R = 0,648$, $y_R = 0,333$, $x_G = 0,284$, $y_G = 0,612$, $x_B = 0,15$, $y_B = 0,075$.

Таким образом, площадь ТЦО с максимальным цветовым охватом для стандарта МКО 1931 г. превосходит в 3,23 раза площадь ТЦО модели G104SN03 V.0 и в 2,13 раза площадь ТЦО модели R190E1-L01. Положение ТЦО модели G104SN03 V.0 и модели R190E1-L01 на XY-плоскости, а также положение ТЦО с максимальным цветовым охватом приведены на рис. 4.

Коэффициенты профиля дисплеев с максимальным цветовым охватом для стандарта МКО 1931 г.

Стандарт для белого цвета	Коэффициенты профиля дисплеев	
	Прямое преобразование: $RGB \rightarrow XYZ$	Обратное преобразование: $XYZ \rightarrow RGB$
D-50 $x_W = 0,34567, y_W = 0,35850$	$\begin{bmatrix} 0,7636324442 & 0,0522719525 & 0,1483075977 \\ 0,2727258729 & 0,7202685512 & 0,0070055759 \\ 0 & 0,0958190532 & 0,7293692314 \end{bmatrix}$	$\begin{bmatrix} 1,331 & -0,061 & -0,27 \\ -0,505 & 1,413 & 0,089 \\ 0,066 & -0,186 & 1,359 \end{bmatrix}$
D-55 $x_W = 0,33242, y_W = 0,34743$	$\begin{bmatrix} 0,7362338752 & 0,0529157938 & 0,1676473836 \\ 0,2629406697 & 0,7291402049 & 0,0079191254 \\ 0 & 0,0969992705 & 0,8244813155 \end{bmatrix}$	$\begin{bmatrix} 1,381 & -0,063 & -0,28 \\ -0,499 & 1,396 & 0,088 \\ 0,059 & -0,164 & 1,203 \end{bmatrix}$
D-65 $x_W = 0,31271, y_W = 0,32902$	$\begin{bmatrix} 0,6952285443 & 0,0538631148 & 0,2013368863 \\ 0,2482959086 & 0,7421935818 & 0,0095105096 \\ 0 & 0,0987357926 & 0,9901645782 \end{bmatrix}$	$\begin{bmatrix} 1,462 & -0,067 & -0,297 \\ -0,49 & 1,371 & 0,086 \\ 0,049 & -0,137 & 1,001 \end{bmatrix}$
D-75 $x_W = 0,29902, y_W = 0,31485$	$\begin{bmatrix} 0,6661515281 & 0,054521759 & 0,2290488028 \\ 0,23791126 & 0,7512692082 & 0,0108195318 \\ 0 & 0,0999431449 & 1,1264503758 \end{bmatrix}$	$\begin{bmatrix} 1,526 & -0,07 & -0,31 \\ -0,484 & 1,355 & 0,085 \\ 0,043 & -0,12 & 0,88 \end{bmatrix}$

Таблица 4

Коэффициенты профиля дисплеев с максимальным цветовым охватом для стандарта МКО 1964 г.

Стандарт для белого цвета	Коэффициенты профиля дисплеев	
	Прямое преобразование: $RGB \rightarrow XYZ$	Обратное преобразование: $XYZ \rightarrow RGB$
D-50 $x_W = 0,34773, y_W = 0,35952$	$\begin{bmatrix} 0,7736447921 & 0,0517760327 & 0,1417854502 \\ 0,3047691606 & 0,6764102079 & 0,0188206315 \\ 0 & 0,1057997531 & 0,7084803982 \end{bmatrix}$	$\begin{bmatrix} 1,316 & -0,06 & -0,262 \\ -0,595 & 1,512 & 0,079 \\ 0,089 & -0,226 & 1,4 \end{bmatrix}$
D-55 $x_W = 0,33411, y_W = 0,34877$	$\begin{bmatrix} 0,7450005985 & 0,0524494563 & 0,1605165135 \\ 0,2934850843 & 0,6852079188 & 0,0213069969 \\ 0 & 0,1071758347 & 0,8020766812 \end{bmatrix}$	$\begin{bmatrix} 1,367 & -0,062 & -0,272 \\ -0,588 & 1,492 & 0,078 \\ 0,079 & -0,199 & 1,236 \end{bmatrix}$
D-65 $x_W = 0,31382, y_W = 0,33100$	$\begin{bmatrix} 0,7017743278 & 0,0534238892 & 0,1928984597 \\ 0,2764565535 & 0,6979380631 & 0,0256053835 \\ 0 & 0,1091670024 & 0,9638843571 \end{bmatrix}$	$\begin{bmatrix} 1,451 & -0,066 & -0,289 \\ -0,577 & 1,465 & 0,077 \\ 0,065 & -0,166 & 1,029 \end{bmatrix}$
D-75 $x_W = 0,29968, y_W = 0,31740$	$\begin{bmatrix} 0,6707629962 & 0,0540905738 & 0,2193178225 \\ 0,2642399683 & 0,7066477363 & 0,0291122955 \\ 0 & 0,1105293137 & 1,0958979075 \end{bmatrix}$	$\begin{bmatrix} 1,518 & -0,069 & -0,302 \\ -0,57 & 1,447 & 0,076 \\ 0,057 & -0,146 & 0,905 \end{bmatrix}$

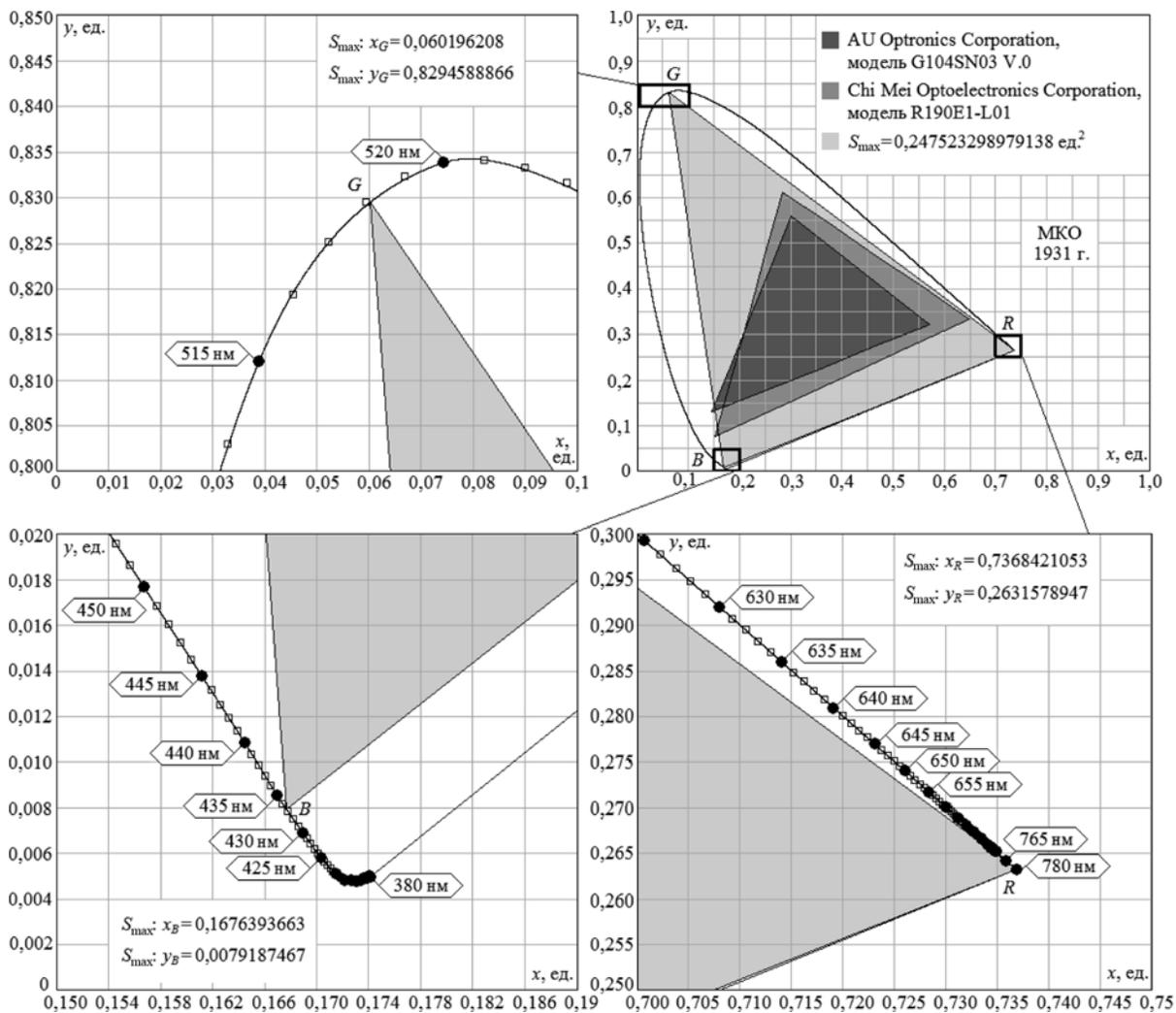


Рис. 4. Треугольник с максимальным цветовым охватом для стандарта МКО 1931 г.

Заключение

В результате интерполяции спектрального локуса получена система (17) интерполирующих функций, определяющих в аналитической форме зависимость (x, y) -координат цветности монохроматических цветов от длины волны λ в видимом диапазоне 380...780 нм. Решение задачи интерполяции спектрального локуса имеет важное практическое значение для повышения точности выполнения колориметрических расчетов световых величин при различном спектральном составе воспринимаемого излучения. В частности, аналитическая форма представления линии спектральных цветностей позволяет повысить точность оценки значения доминирующей длины волны светового излучения, определяющей цветовой тон, повысить точность оценки значения насыщенности излучаемого цвета и пр.

В результате решения задачи оценки значений координат цветности вершин треугольника цветового охвата с максимальной площадью, вписанного в спектральный локус, получены числовые значения профилей дисплеев, соответствующих колориметрическим стандартам МКО 1931 г. и МКО 1964 г. для задания точки белого цвета D-75, D-65, D-55, D-50. Анализ технической документации на дисплеи различных производителей показал, что все они обладают сравнимыми между собой площадями ТЦО, варьируемыми в пределах 0,06...0,12 ед.², что не превышает 50 % площади ТЦО с максимальным цветовым охватом. В связи с этим основной задачей разработчиков дисплеев с трехкомпонентной схемой управления пикселями является создание такой технологии изготовления дисплеев, которая обеспечивала бы цветовоспроизведение с максимально возможным цветовым охватом.

Список литературы

1. **Жаринов И. О., Жаринов О. О.** Бортовые средства отображения информации на плоских жидкокристаллических панелях: учеб. пособие. Информационно-управляющие системы. СПб: ГУАП, 2005, 144 с.
2. **Luo Zh., Chen Y., Wu Sh.-T.** Wide color gamut LCD with a quantum dot backlight // *Optics Express*. 2013. Vol. 21, N 22. P. 1–16.
3. **Gatchin Y. A., Zharinov I. O., Korobeynikov A. G., Zharinov O. O.** Theoretical estimation of Grassmann's transformation resolution in avionics color coding systems // *Modern Applied Science*. 2015. Vol. 9, N 5. P. 197–210.
4. **Жаринов И. О., Жаринов О. О.** Оценка параметров математической модели цветопередачи жидкокристаллической панели // Информационно-управляющие системы. 2015. № 2. С. 49–56.
5. **Rodriguez-Pardo C. E., Gaurav Sh., Feng X.-F., Speigle J., Sezan I.** Optimal gamut volume design for three primary and multiprimary display systems // *Proc. of SPIE-IS&T Electronic Imaging*. 2012. Vol. 8292, 82920C. P. 1–7.
6. **Жаринов И. О., Жаринов О. О.** Исследование распределения оценки разрешающей способности преобразования Грассмана в системах кодирования цвета, применяемых в авионике // Программная инженерия. 2014. № 8. С. 40–47.
7. **Жаринов И. О., Жаринов О. О.** Исследование свойства равноконтрастности цветовых пространств, применяемых в авионике // Программная инженерия. 2014. № 11. С. 35–43.
8. **Зайцева Е. В.** Погрешности аппроксимации усредненных кривых видности при определении освещенности оптического изображения // Доклады ТУСУРа. 2010. № 1, Ч. 2. С. 69–73.
9. **Роджерс Д., Адамс Дж.** Математические основы машинной графики. М.: Мир, 2001.
10. **Kwak Y., Lee S., Choe W., Kim Ch.-Y.** Optimal chromaticities of the primaries for gamut 3-channel display // *Proc. of SPIE-IS&T Electronic Imaging*. 2005. Vol. 5667. P. 319–327.

I. O. Zharinov, Associate Professor, Chief of Department, e-mail: igor_rabota@pisem.net, Saint Petersburg National Research University of Information Technologies, Mechanics and Optics (University ITMO), Chief of Learning-Scientists Center, SPb Scientific Design Bureau "Electroavtomatika" n. a. P. A. Efimov, **O. O. Zharinov**, Associate Professor, Saint Petersburg State University of Aerospace Equipment

The Solution to the Problem of Evaluation of Parameters of the Mathematical Model of Color Reproduction for Display with Maximal Gamut Based of Spectrum Locus Interpolation

The problem and methods for its solution for estimation of parameters of the mathematical model of color reproduction, which are suitable for LCD and LED displays with maximum gamut, were considered. The technique is based on interpolation of the given spectrum locus on the XY-plane and the subsequent calculation of the chromaticity coordinates of the vertices of the color gamut triangle, inscribed in the color locus and having the maximum possible area. Estimation of parameters of the mathematical model of color reproduction of the display was performed on the basis of the solution of the system of equations, the initial data for which are the chromaticity coordinates of the vertices of the color gamut triangle with maximum area and the chromaticity coordinates of the white point of the given display unit. There three ways for interpolation of the spectrum gamut were developed and tested: first technique, based on interpolation in the space of color coordinates; second one, based on the interpolation in the space of chromaticity coordinates; and third technique, based on the interpolation of the spectral locus as some tabular curve having its specific shape. There numerical values were obtained for profiles of displays, with a maximum possible area of the color gamut triangle, corresponding to the colorimetric standards D-75, D-65, D-55, D-50.

Keywords: colorimetric standard, spectrum locus, chromaticity coordinates, interpolation, Bezier curve, gamut triangle, display

References

1. **Zharinov I. O., Zharinov O. O.** *Bortovye sredstva otobrazheniya informacii na ploskih zhidkokristallicheskih paneljah* (On-board Display on Flat Liquid Crystal Panels). Saint-Petersburg, Informacionno-upravljajushhie sistemy, 2005, 144 p. (in Russian).
2. **Luo Zh., Chen Y., Wu Sh.-T.** Wide color gamut LCD with a quantum dot backlight. *Optics Express*, 2013, vol. 21, no. 22, pp. 1–16.
3. **Gatchin Y. A., Zharinov I. O., Korobeynikov A. G., Zharinov O. O.** Theoretical estimation of Grassmann's transformation resolution in avionics color coding systems. *Modern Applied Science*, 2015, vol. 9, no. 5, pp. 197–210.
4. **Zharinov I. O., Zharinov O. O.** Ocenka parametrov matematicheskoj modeli cvetopredachi zhidkokristallicheskoj paneli (Evaluation of Parameters of the Mathematical Model of Color Response Performances of LCD-panel). *Informacionno-Upravljajushhie Sistemy*, 2015, no. 2, pp. 49–56 (in Russian).
5. **Rodriguez-Pardo C. E., Gaurav Sh., Feng X.-F., Speigle J., Sezan I.** Optimal gamut volume design for three primary and multiprimary display systems. *Proc. of SPIE-IS&T Electronic Imaging*, 2012, vol. 8292, 82920C, pp. 1–7.
6. **Zharinov I. O., Zharinov O. O.** Issledovanie raspredelenija ocenki razreshajushhej sposobnosti preobrazovanija Grassmana v sistemah kodirovanija cveta, primenjaemyh v avionike (The Evaluation of Hardware-Caused Inaccuracy of Indirect Measurements of Chromaticity Coordinates in Color Model Data Used in Avionics). *Programmnaya Ingeneria*, 2014, no. 12, pp. 39–46 (in Russian).
7. **Zharinov I. O., Zharinov O. O.** Issledovanie svojstva ravnokonttrastnosti cvetovyh prostranstv, primenjaemyh v avionike (The Research of Uniform Chromaticity Property of Color Spaces, Used in Avionics Equipment). *Programmnaya Ingeneria*, 2014, no. 11, pp. 35–43 (in Russian).
8. **Zajceva E. V.** Pogreshnosti approksimacii usrednennyh krivyh vidnosti pri opredelenii osveshennosti opticheskogo izobrazhenija (Errors of approximation of averaged curves of visibility at determination of luminosity of optical image). *Lectures of Tomsk State University of Systems Managements and Radio Electronics*, 2010, no. 1, pp. 69–73 (in Russian).
9. **Rogers D., Adams G.** *Matematicheskie osnovy mashinnoj grafiki* (Mathematical bases of machine graphic arts). M.: Mir, 2001 (in Russian).
10. **Kwak Y., Lee S., Choe W., Kim Ch.-Y.** Optimal chromaticities of the primaries for gamut 3-channel display. *Proc. of SPIE-IS&T Electronic Imaging*, 2005, vol. 5667, pp. 319–327.

С. А. Афонин, канд. физ.-мат. наук, вед. науч. сотр., e-mail: serg@msu.ru, НИИ механики МГУ имени М. В. Ломоносова,

А. Э. Гаспарянц, студент, e-mail: gaspariants@mail.ru, МГУ имени М. В. Ломоносова

Автоматическое построение функции оценки качества в задаче разрешения неоднозначности имен авторов научных публикаций

Описан метод построения решающего правила для задачи автоматического разрешения неоднозначности имен авторов научных публикаций, основанный на методах машинного обучения. Предложен ряд эвристик, которые позволяют сократить время поиска решения. Результаты тестирования программного комплекса, проведенного на коллекции реальных данных, показали высокую эффективность и точность предлагаемого метода.

Ключевые слова: автоматическое обучение, классификация, электронная библиотека, библиографическая запись, имя автора, разрешение неоднозначности, поиск дубликатов

Введение

Настоящая работа является продолжением работы [1], в которой был предложен алгоритм для автоматического распознавания авторов при добавлении новой публикации в библиографическую базу данных. Задача распознавания авторов состоит в следующем. При добавлении новой библиографической записи требуется сопоставить авторов этой публикации с имеющимися в базе данных записями (пользователями системы). Сложность задачи состоит в наличии большого числа однофамильцев и неочевидных критериях оценки качества ответа. Если, например, для каждого из трех соавторов статьи в базе найдется по 10 однофамильцев, то существует 1000 различных вариантов выбора соответствующих пользователей.

Предложенное в работе [1] решение учитывает историю совместных публикаций и основано на переборе в пространстве возможных ответов. Для оценки каждого потенциального ответа используется ряд количественных показателей и эвристическая функция интегральной оценки качества. В настоящей работе предлагается использовать методы автоматического обучения для построения функции оценки качества набора пользователей. Поскольку проблемные вопросы перебора возможных ответов сохраняются, предлагается ряд эвристик в целях уменьшения времени поиска ответа. Рассматривается также близкая задача поиска дублирующих записей об авторах. Данная задача может иметь самостоятельный интерес, так как вычисление наукометрических показателей осложняется наличием дубликатов. С точки зрения библиографической базы данных каждая дублирующая учетная запись

будет обрабатываться как новый автор, поэтому показатели цитирования будут подсчитаны неверно. В связи с этим обстоятельством возникает необходимость в нахождении всех дубликатов в целях их дальнейшего слияния.

1. Задача распознавания авторов новой публикации

Автором будем называть тройку слов $\langle f, i, o \rangle$, которые соответствуют фамилии, имени и отчеству (или инициалам). Имя и отчество могут быть пустыми словами. *Пользователем* называется запись в базе данных, соответствующая некоторому ученому. С каждым пользователем связаны следующие *атрибуты*: имя, список соавторов, список публикаций и др. Множество всех авторов (троек слов в соответствующем алфавите) обозначим через N , а множество всех пользователей — через R . Заметим, что R — конечное множество в силу конечности базы данных. Пусть NS обозначает множество всех конечных наборов N , а RS — множество всех наборов R .

Пусть авторами публикации является набор $\langle N_1, \dots, N_k \rangle \in NS$. Предположим, что задана некоторая функция $F: RS \times NS \rightarrow [0, 1]$ оценки качества выбора пользователей для заданных авторов. Тогда задача разрешения имен авторов сводится к определению такого набора $\langle R_1, \dots, R_k \rangle \in RS$, который имеет максимальное значение функции F . Поиск теоретически может быть осуществлен полным перебором в силу конечности множества R и естественного ограничения, что число пользователей не превосходит число авторов работы. Наибольшую трудность составляет построение функции F . Эта функция должна

обеспечивать возможность получения "адекватных" результатов. Критерии построения такой функции не формализуемы, поскольку в данном случае нельзя определить генеральную совокупность.

Общий метод решения задачи совпадает с тем, который был предложен в работе [1]. Каждому автору новой библиографической записи можно сопоставить некоторый набор пользователей из базы данных на основе текстовой близости между именами пользователей и авторов. Функция текстовой близости может учитывать варианты транслитерации, опечатки, сокращенные или опущенные части имени и другие модификации. Декартово произведение полученных множеств образует пространство решений. Оптимальным решением является такой набор авторов, на котором достигается максимальное значение функции качества. Поиск решения проводится перебором с применением алгоритмов, основанных на эвристиках.

Для построения функции F в настоящей работе предлагается использовать метод классификации данных. Классификатор h для каждого набора пользователей I определяет, правильный он или нет. Классификатор $h(I)$ принимает значения $\{-1, +1\}$, причем значение 1 соответствует правильному набору.

1.1. Задача бинарной классификации

Пусть X — пространство признаков, Y — пространство классов. Задача классификации состоит в определении того, к какому из классов принадлежит новая наблюдаемая величина. В случае, если $Y = \{-1, +1\}$, задача называется бинарной классификацией. Функция $h: X \rightarrow Y$ называется гипотезой и $\forall x \in X, y = h(x)$ она определяет класс, которому принадлежит x . Эта функция строится на основании обучающей выборки $\{(x^{(i)}, y^{(i)})\}$, $x^{(i)} \in X, y^{(i)} \in Y$. Для данной задачи предлагается использовать метод опорных векторов. Поскольку для каждой статьи существует только один правильный ответ, выборка может оказаться сильно несбалансированной. Классификаторы, как правило, работают плохо на несбалансированных наборах данных [2], поскольку они предназначены обобщать данные и выводить в качестве результата наиболее простую гипотезу, которая лучше всего соответствует данным. Тем не менее, показано, что метод опорных векторов эффективен при работе с несбалансированной выборкой [2].

Объектом классификации является набор пользователей. Классифицирование объекта происходит следующим образом: вычисляются признаки, описывающие объект, значения признаков образуют вектор, который классифицируется в один из классов функцией h . Для того чтобы оценить правильность набора, нужно ответить на следующий вопрос: "Насколько вероятно, что данная комбинация пользователей написала работу в соавторстве?" Для этого определим признаки, оценивающие связанность пользователей историей публикаций. Описывать связанность удобно в терминах гиперграфа, в котором вершины — это пользователи, а ребра отражают

совместные публикации пользователей. В качестве дополнительного признака, позволяющего разделить ученых из различных предметных областей, предлагается использовать информацию о журнале, при условии, что обрабатываемая запись описывает публикацию в научном журнале. По журналу можно определить рубрики, которые сравним с рубриками, соответствующими выбранным пользователям.

Каждому набору пользователей $R = \{R_1, \dots, R_n\}$ можно поставить в соответствие взвешенный гиперграф истории совместных публикаций $H = (V, E)$. Множество вершин H совпадает с R , а набор $u = \langle v_1, \dots, v_n \rangle$ образует ребро, если пользователи u имели совместные публикации. Весом ребра является число совместных публикаций. Обозначим через $A(u)$ все публикации, написанные в соавторстве вершинами из u . Будем считать, что все ребра максимальны в том смысле, что добавление в ребро новой вершины уменьшает множество совместных публикаций: $\forall e \in E, v \in V \setminus e, A(e \cup \{v\}) \subset A(e)$.

Ранг $rank(H)$ гиперграфа H — это максимальная степень его ребра. Трансверсаль гиперграфа $H = (V, E)$ — это множество $T \subseteq V$, которое имеет непустое пересечение с каждым ребром. Трансверсаль минимальна, если она состоит из наименьшего числа вершин. Показатель $\tau(H)$ — мощность минимальной трансверсали. Понятия пути и связности определяются аналогично тому, как это делается применительно к графам.

В качестве числовых характеристик, описывающих заданный набор R , предлагается использовать следующие шесть значений.

1. Значение $feature_1 = rank(H)$ — ранг гиперграфа. Этот признак определяет, какое число пользователей из набора ранее публиковались вместе. Если много пользователей ранее публиковались в соавторстве, то такой набор можно считать "хорошим".

2. Значение $feature_2 = |\bigcap_{e \in E} e|$. Если это значение отлично от нуля, то существует один или несколько пользователей, которые раньше публиковались в соавторстве со всеми остальными в этом наборе. Можно условно считать таких пользователей "руководителями". На основе этого можно предположить, что такая комбинация пользователей могла публиковаться в соавторстве.

3. Значение $feature_3 = cc_h$ — число компонент связности. Нахождение двух вершин v и v' в одной компоненте связности означает наличие хотя бы одного пути, соединяющего v с v' . Это может означать, что эти два пользователя работают в одной области или в одной организации. Очевидно, чем меньше число компонент связности, тем больше вершин, связанных путями между собой.

4. Значение $feature_4 = \tau(H)$ — число вершин минимальной трансверсали. Этот признак является обобщением признака $feature_2$ на тот случай, когда в наборе не оказалось "руководителя". В этом случае вершины гиперграфа можно разбить на подмножества, в каждом из которых будет "руководитель". Можно считать набор тем лучше, чем меньше значение $\tau(H)$.

5. $feature_5 = 1/m \sum_{i=1}^m w_i$ — средний вес ребер. Данный

признак является противопоставлением признаку $feature_1$ с целью определить, насколько "хорошим" является гиперграф с ребрами небольшой мощности и большого веса.

6. Следуя импликации статья → журнал → рубрики, можно определить список рубрик, соответствующих данной статье. Предположим, что имеется функция — "черный ящик" $br(x, R)$ — степень принадлежности пользователя R к рубрике x . Пусть $rubrics$ — список рубрик, относящихся к новой статье. Выражение

$feature_6 = 1/n \sum_{i=1}^n \max_{x \in rubrics} br(x, R_i)$ — функция схо-

жести автора с рубриками журнала. Этот признак определяет степень схожести научных интересов пользователей в наборе с темой публикации.

1.2. Обучение классификатора

На основе 15 000 статей системы ИСТИНА [3] была сгенерирована обучающая выборка. Из каждой статьи выбирались имена авторов и название журнала и запоминался правильный ответ. Далее для каждой комбинации похожих пользователей вычислялись признаки и вектор

$$features = (feature_1, feature_2, feature_3, feature_4, feature_5, feature_6)$$

добавлялся к одному из классов. Среди всех статей 12 000 представляли собой обучающую выборку, остальные — тестовую выборку. Поскольку заранее неизвестно, являются ли данные линейно разделимыми, применим к выборке классификаторы с различными ядерными функциями. Общий вид классификатора

$$h(\mathbf{x}) = \text{sign} \left(\sum_{i=1}^L \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) - b \right), \quad (1)$$

• $k: X \times X \rightarrow \mathbb{R}$ — ядерная функция, которая измеряет схожесть между парой признаков $\mathbf{x}, \mathbf{x}' \in X$;

• суммирование проводится по L примерам $\{(\mathbf{x}_i, y_i)\}_{i=1}^L$ обучающей выборки, $y_i \in \{-1, +1\}$;

• $\alpha_i \in \mathbb{R}, b \in \mathbb{R}$ — веса и значение смещения, которые определяют с помощью алгоритма обучения: \mathbb{R} — множество действительных чисел.

Для обучения классификатора использовалась библиотека shogun [4]. Сравним различные ядерные функции, предоставляемые этой библиотекой на их точность. Пусть Pos — число положительных экземпляров в выборке, Neg — число отрицательных экземпляров. Обозначим через TP и TN число корректно определенных объектов положительного и отрицательного классов, а через FP и FN — число объектов, ошибочно отнесенных к этим классам, соответственно. Тогда:

$$\text{Точность} = \frac{(TP + TN)}{(Pos + Neg)};$$

$$F_1 = \frac{2TP}{(2TP + FP + FN)}.$$

В табл. 1 представлены ядра, показавшие наибольшие значения точности и F_1 -меры. Некоторые из них содержат дополнительные параметры, такие как τ, γ и c , для оптимизации которых применялся метод сеточного поиска, предоставляемый средствами библиотеки shogun.

Таблица 1

Сравнение классификаторов

Ядерная функция $k(\mathbf{x}, \mathbf{x}')$	F_1	Точность
$\exp(-\ \mathbf{x} - \mathbf{x}'\ ^2/\tau)$	0,787	0,933
$\tanh(\gamma \mathbf{x} \mathbf{x}' + c)$	0,772	0,832
$\exp(-\ \mathbf{x} - \mathbf{x}'\ /\tau)$	0,789	0,931
$\mathbf{x} \mathbf{x}'$	0,801	0,936

Из данных табл. 1 видно, что на данной обучающей выборке наилучшие результаты показывает линейный метод опорных векторов (функция $\mathbf{x} \mathbf{x}'$). Учитывая объем выборки, можно предположить, что линейная функция лучше всего соответствует рассматриваемой задаче в целом. Функцию-гипотезу (1) можно переписать в следующем виде:

$$h(\mathbf{x}) = \text{sign}(\mathbf{w} \mathbf{x} - b), \quad \mathbf{w} = \sum_{i=1}^L \alpha_i y_i \mathbf{x}_i. \quad (2)$$

Получены следующие значения вектора \mathbf{w} и смещения b :

$$\mathbf{w} = (-0,852613, 0,277309, 1,32469, -0,114332, 0,184657, 1,23419);$$

$$b = -1,69425.$$

Значение функции $F(I, S)$ на наборе пользователей I при заданном множестве авторов S вычисляется следующим образом:

1) вычисление вектора $features$ по набору I и множеству S ;

2) вычисление значения $h(features)$;

3) $F(I, S) = 0$, если $h(features) = -1$, иначе 1.

1.3. Эвристики поиска оптимального решения

Функция F позволяет оценить степень соответствия выбранных пользователей заданному набору авторов. Поиск решения может быть осуществлен полным перебором. Однако, учитывая специфику задачи, можно предложить следующие эвристики, которые могут в некоторых случаях уменьшить перебор комбинаций пользователей.

• При составлении множества пользователей, "похожих" на заданного автора (фамилии и, воз-

можно, имени и отчества), используется высокое пороговое значение.

- Будем предполагать, что классификатор верно определяет правильный набор и принимает значение +1 на единственном правильном наборе. Соответственно, поиск проводится до нахождения первого набора, на котором $F = 1$.

- Считаем, что наиболее вероятные "руководители" авторского коллектива — это пользователи, у которых большее число публикаций. Следовательно, первыми в наборы добавляем пользователей с максимальным числом публикаций.

1.4. Численные эксперименты

Эксперименты проводили с помощью системы ИСТИНА [3] на выборке статей, не участвующих в обучении классификатора. Проводили ряд тестов в целях оценивания качества распознавания авторов. Эксперименты представляли собой моделирование добавления новой публикации в базу данных. Из каждой публикации выбирали всю необходимую информацию, а также запоминали правильный набор пользователей. В обоих экспериментах статьи были ограничены числом авторов, равным 14. Это обосновывается тем обстоятельством, что число статей с большим количеством авторов слишком мало, чтобы по нему можно было судить о правильности работы алгоритма. Также, поскольку предполагается использование предложенного алгоритма в реальном времени, в процессе тестирования измеряли среднее время обработки наборов пользователей. Результат работы алгоритма сравнивали с правильным ответом.

Первый эксперимент состоял в проверке правильности распознавания авторов на случайной выборке статей. Вся выборка была разбита по числу авторов

Таблица 2

Результаты работы алгоритма на случайном наборе статей

Число авторов в статье	Всего статей	Среднее число комбинаций пользователей	Среднее время обработки статьи, с	Точность, %
2	592	6	0,1254	97,12
3	754	16	0,1860	97,08
4	797	21	0,2384	95,10
5	897	53	0,2997	94,09
6	899	142	0,3438	92,32
7	904	277	0,3807	91,92
8	912	716	0,4340	92,10
9	911	884	0,4938	91,00
10	919	1263	0,5538	92,05
11	804	3474	0,6128	91,04
12	568	42632	0,6527	89,96
13	420	22654	0,77	91,90
14	341	7544	0,7681	90,61

Таблица 3

Точность распознавания авторов, для которых выбор пользователей неоднозначен

Число неоднозначностей авторов в статье	Всего статей	Среднее число комбинаций пользователей	Среднее время обработки статьи, с	Точность, %
1	1497	5	0,11	96,99
2	830	15	0,1265	96,74
3	687	66	0,1920	96,21
4	614	287	0,2784	92,18
5	540	1238	0,3497	89,44
6	533	3020	0,3832	87,05
7	524	12060	0,4223	87,59
8	349	133788	0,4690	90,54
9	340	811432	0,5427	86,17
10	225	2544441	0,5949	88,00
11	210	686361	0,6688	88,57
12	216	2439936	0,7071	93,51
13	115	11441664	0,7912	85,22
14	108	51321600	0,8632	86,11

в статьях. Результат отдельного теста представляет собой процентное соотношение правильно распознанных авторов ко всем авторам. Результаты эксперимента приведены в табл. 2.

Второй эксперимент отличался от первого тем, что результирующий процент правильно распознанных авторов считали только по отношению к тем авторам, для которых выбор пользователя был неоднозначен. Результаты эксперимента приведены в табл. 3.

На основе результатов проведенных экспериментов можно утверждать, что предложенный метод правильно определяет авторов публикаций в более чем 86 % случаев. Анализ тех статей, на которых алгоритм выдал результат меньше 100 %, показал, что причинами ошибок являются перечисленные далее.

- Наличие у пользователей соавторов, являющихся однофамильцами.

Например, соавторами пользователя R с фамилией Иванов могут оказаться два различных пользователя R' , R'' с фамилией Петров. Поэтому, если в новой статье среди авторов указано Иванов, Петров без инициалов, то набор с пользователем R' может быть неотличим от набора с пользователем R'' .

- Погрешность классификатора, связанная с недостаточным числом признаков. Поскольку не существует методов, помогающих угадать наилучший набор признаков для классификацию, классификатор в некоторых случаях может принимать неправильный ответ за правильный.

- Наличие в базе данных записей-дубликатов. Это учетные записи, заведенные для одного и того же ученого. При этом в них может содержаться одинако-

вая информация. Как следствие, два различных набора с такими пользователями могут быть неотличимы.

Отметим, что при увеличении множества потенциальных ответов время работы программы растет незначительно, что позволяет говорить о практической применимости данного метода.

2. Задача определения дубликатов

Данная задача состоит в определении записей, которые ссылаются на одну сущность. Пусть $R = \{R_1, \dots, R_n\}$ — все записи в базе данных. Необходимо построить разбиение множества R на m множеств $\{A_1, \dots, A_m\}$, где $A_i = \{R_{k1}, \dots, R_{ki}\}$ содержит записи, представляющие одну сущность. Традиционный метод решения такой задачи состоит из двух частей: попарно сравнивающая функция и алгоритм кластеризации. Сравнивающая функция определяет, ссылаются ли две записи на одну сущность, основываясь на некоторых атрибутах. Результат такой функции может быть как бинарный (да/нет), так и коэффициент схожести между 0 и 1. Алгоритм кластеризации далее кластеризует записи, основываясь на значениях этой функции.

2.1. Связанные работы

Существует большое число работ, относящихся к задаче определения дубликатов в некоторой базе данных или социальной сети. В этих работах исследования варьируются от использования простых строковых метрик до использования техник машинного обучения [5–7]. В работе [7] использованы два классификатора: гибридный наивный байесовский классификатор и метод опорных векторов, в работе [8] использован алгоритм DBSCAN для кластеризации авторов на основе таких метаданных, как имена, электронная почта, адреса. Алгоритм DBSCAN использует метрику, построенную с помощью метода опорных векторов. В работе [9] описан алгоритм, основанный на алгоритмах PLSA и LDA для устранения неоднозначности имен авторов с учетом содержания их статей.

Эти работы можно разделить по предложенным методам решения задачи. Некоторые авторы используют алгоритмы обучения с учителем, что неприменимо в данном случае, поскольку заранее нет информации о наличии дубликатов в системе. В других работах предложено использовать совместно классификацию данных (для построения метрики) и кластеризацию с помощью построенной метрики. Данный метод также неприменим, поскольку классификация реализуется по некоторой обучающей выборке. К третьей группе можно отнести работы, в которых предложено использовать только алгоритмы кластеризации. Оценку точности кластеризации в этом случае проводят с помощью вручную предоставленных отметок для записей. Результаты работы алгоритма при этом оценивают на заранее известном множестве авторов. Данный способ оценивания работы алгоритма неприменим,

поскольку нельзя достоверно утверждать о наличии дубликатов в системе для какого-нибудь автора. Поэтому предлагаемый метод решения следует использовать в полуавтоматическом режиме. Его результатом является некоторая подсказка для пользователей, которые принимают окончательное решение об объединении дубликатов.

Определение дубликатов пользователей. Пусть $G = (V, E)$ неориентированный взвешенный граф с множеством вершин $V = R$. Пусть вес между вершинами R_i и R_j равен $w_{ij} \geq 0$. Взвешенная матрица смежности графа $\mathbf{W} = (w_{ij})$. Коэффициенты w_{ij} являются значениями симметричной сравнивающей функции на парах вершин:

$$fsim: R \times R \rightarrow [0, 1], \\ w_{ij} = fsim(R_i, R_j).$$

Определим матрицу расстояний $\mathbf{D} = (d_{ij})$: $d_{ij} = \sqrt{w_{ii} + w_{jj} - 2w_{ij}}$.

Поскольку мощность множества R может быть достаточно велика, необходимо (в целях улучшения производительности), чтобы матрица \mathbf{D} была разреженной. Для этого отбросим все ребра с весом меньше ϵ для некоторого $\epsilon > 0$. Задача разбиения множества R сводится к задаче разбиения графа G .

Отметим перечисленные далее особенности разбиения графа.

- Неизвестно m — число элементов разбиения.
- $\forall a, b \in A_i: fsim(a, b) \approx 1$.
- Большинство кластеров — одноэлементные.

Предполагается, что в базе данных число дубликатов невелико. Это достигается за счет проверки данных самими пользователями.

Разбиение графа строится с учетом связности графа. Каждое множество-разбиение представляется в виде клики графа с ребрами весом выше некоторого порогового значения τ . Для нахождения всех клик используется измененный алгоритм Брона—Кербоша, приведенный в листинге. Отличие от стандартного алгоритма заключается в том, что на каждом шаге к окрестности $N(v)$ вершины v добавляются только те вершины, которые соединены с v ребрами весом больше τ .

```

1  R=0, P=V, X=0.
2  BronKerbosh(R, P, X)
3      Если P=0 и X=0
4          R — максимальная клика
5      Для всех v из P:
6          Добавляем в N(v) все вершины, соединенные
7          с v ребром весом больше  $\tau$ 
8          BronKerbosh(R+{v}, P ∩ N(v), X ∩ N(v))
9          P=P-{v}
10         X=X+{v}

```

Алгоритм поиска клики с ограничением на минимальные веса ребер

2.2. Построение функции схожести

Функцию схожести двух записей определим на основе признаков, по которым будем сравнивать эти записи. Общий вид функции:

$$dsim_i(R_k, R_l) = \sum_{i=1}^m \omega_i fsim_i(R_k, R_l),$$

где $fsim_i(R_k, R_l)$ — результат сравнения по i -му признаку; m — число признаков; $\sum_{i=1}^m \omega_i = 1$.

Веса ω_i определяют по формулам, которые использовались в работе [10] при решении аналогичной задачи:

$$\omega_i^d = p_i / \left(\sum_{i=1, \dots, m} p_i \right), p_i = \sum_{k, l=1, \dots, m; k > l} fsim(R_k, R_l);$$

$$\omega_i^n = q_i / \left(\sum_{i=1, \dots, m} q_i \right), q_i = \sum_{k, l=1, \dots, m; k > l} (1 - fsim(R_k, R_l)); \quad (3)$$

$$\omega_i = \alpha_i \omega_i^d + (1 - \alpha_i) \omega_i^n, \alpha_i \in [0, 1], i = 1, \dots, m.$$

Признаки. Рассмотрим следующие данные, которые можно использовать при сравнении пользователей на идентичность:

- фамилия;
- имя или первый инициал;
- отчество или второй инициал;
- список соавторов;
- список рубрик (области науки, к которым относятся работы автора).

Пусть $lname_i, fname_i, mname_i$ — соответственно фамилия, имя и отчество пользователя R_i . Для оценки их схожести используют расстояние Jaro-Winkler [11]:

- $fsim_1(R_i, R_j) = d_w(lname_i, lname_j)$;
- $fsim_2(R_i, R_j) = d_w(mname_i, mname_j)$;
- $fsim_3(R_i, R_j) = d_w(fname_i, fname_j)$;
- пусть $coauthors_i \subset R$ — список соавторов пользователя R_i , тогда

$$fsim_4(R_i, R_j) = \frac{|coauthors_i \cap coauthors_j|}{|coauthors_i + coauthors_j|};$$

- пусть $rubrics_i$ — список рубрик, соответствующих записи R_i , тогда

$$fsim_5(R_i, R_j) = \frac{|rubrics_i \cap rubrics_j|}{|rubrics_i + rubrics_j|}.$$

2.3. Результаты

Предложенный метод был реализован программно и использован для определения дубликатов в базе данных системы ИСТИНА [3]. Из базы данных были извлечены данные по более чем 130 000 авторов. В результате работы алгоритма было выявлено 2363 пары дубликатов. Поскольку изначально неизвестно о на-

Оценка работы алгоритма

Известных дубликатов	Возможных дубликатов	Распознанных достоверных	Точность, %
237	2367	206	87

личии каких-либо дубликатов в системе, автоматическими средствами проверить работу алгоритма не представлялось возможным. Однако при некоторых допущениях можно оценить точность предлагаемого метода. В базе данных системы ИСТИНА существует информация о пользователях, которые регистрировались в системе. Эта информация содержит имя пользователя и адрес электронной почты. Предположим, что две различные записи являются дубликатами, если у них совпадают фамилии и адреса электронной почты до символа "@". Таким образом, получим множество пар пользователей, которые являются дубликатами. Проверим, сколько таких пар из этого множества распознает предложенный алгоритм. Результат представлен в табл. 4. Столбец "Возможных дубликатов" представляет число кластеров в результате работы алгоритма. Каждый кластер разбиваем на множество пар. Далее каждую такую пару сравниваем с парой известных дубликатов. Столбец "Распознанных достоверных" определяет, сколько среди пар известных дубликатов распознано алгоритмом. Заметим, что предложенный алгоритм просматривает данные не только пользователей, а вообще всех авторов системы. При этом нельзя определить, насколько правильно были распознаны дубликаты среди остальных пользователей.

Заключение

Рассмотрены две задачи, необходимость решения которых возникает в библиографических базах данных. Первая задача состоит в распознавании авторов новой статьи, когда необходимо определить, кто из пользователей системы является авторами статьи. Предложен и реализован программно метод решения задачи с использованием классификации данных. Проведено тестирование на реальных данных, которое показало, что предлагаемый алгоритм в зависимости от числа авторов и количества вариантов корректно распознает 86...97 % авторов, что превосходит значения, полученные в работе [1] с использованием эвристической функции оценки.

Вторая задача состоит в определении дубликатов среди пользователей системы. Для ее решения предложена метрика, оценивающая близость двух пользователей, на основе которой проведена кластеризация пользователей. Результат работы программной реализации метода проверен на множестве пользователей, для которого известны все дубликаты. Предложенный алгоритм распознал 87 % известных дубликатов. Возможные направления развития рассматриваемого проекта связаны с автоматическим исправлением данных в библиографической базе данных. Примером такой задачи является задача определения неправильно привязанных к авторам публикаций.

Список литературы

1. **Афонин С. А., Гаспарянц А. Э.** Разрешение неоднозначности авторства публикаций при автоматической обработке библиографических данных // Программная инженерия. 2014. № 1. С. 25—29.
2. **Akbani R., Kwek S., Japkowicz N.** Applying support vector machines to imbalanced datasets // In Proceedings of the 15th European Conference on Machine Learning (ECML). 2004. P. 39—50.
3. **Васенин В. А., Афонин С. А., Голомазов Д. Д., Козыцын А. С.** Интеллектуальная Система Тематического Исследования Научно-технической информации (ИСТИНА) // Информационное общество. 2013. № 1—2. С. 21—36.
4. **Sonnenburg S., Rätsch G., Henschel S.** et al. The SHOGUN Machine Learning Toolbox // Journal of Machine Learning Research. 2010. June. Vol. 11. P. 1799—1802.
5. **Bekkerman R., McCallum A.** Disambiguating Web Appearances of People in a Social Network // Proceedings of the 14th International Conference on World Wide Web. WWW'05. New York, NY, USA: ACM, 2005. P. 463—470.
6. **Christen P.** A Comparison of Personal Name Matching: Techniques and Practical Issues // Data Mining Workshops, 2006.

ICDM Workshops 2006. Sixth IEEE International Conference on. 2006. P. 290—294.

7. **Han H., Giles L., Zha H.** et al. Two Supervised Learning Approaches for Name Disambiguation in Author Citations // Proceedings of the 4th ACM/IEEE-CS Joint Conference on Digital Libraries. JCDL'04. — New York, NY, USA: ACM, 2004. P. 296—305.
8. **Huang H., Ertekin S., Giles C. L.** Efficient Name Disambiguation for Large-scale Databases // Proceedings of the 10th European Conference on Principle and Practice of Knowledge Discovery in Databases.— PKDD'06. Berlin, Heidelberg: Springer-Verlag, 2006. P. 536—544.
9. **Song Y., Huang J., Council I. G.** et al. Efficient Topic-based Unsupervised Name Disambiguation // Proceedings of the 7th ACM/IEEE-CS Joint Conference on Digital Libraries. JCDL '07. New York, NY, USA: ACM, 2007. P. 342—351.
10. **Fouzia S. Manjusha K.** Record Matching in Web Databases Using Unsupervised Approach // International Journal of Scientific Engineering and Technology. 2013. Vol. 9, N 2. P. 895—899.
11. **Winkler W. E.** String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage // Proceedings of the Survey Research Methods Section, American Statistical Association. ERIC, 1990. P. 354—359. URL: http://www.amstat.org/sections/srms/Proceedings/papers/1990_056.pdf

S. A. Afonin, Leading Researcher, Institute of Mechanics, Lomonosov Moscow State University,
A. E. Gasparianc, Student, e-mail: gaspariants@mail.ru, Lomonosov Moscow State University

Construction of Quality Function for Scientific Papers Author Names Disambiguation Problem Using Supervised Learning Techniques

In this paper the authors' names disambiguation problem is considered. This problem consists of finding a mapping between given set of scientific paper authors, i.e. tuple of strings, and the set of all authors' records of a bibliographic database. More than one database record may be matched by author's name. In this case disambiguation is conducted using information about previous joint works recorded in the database. A number of numerical features reflecting quality of the mapping proposed, and supervised learning techniques were applied in order to obtain final decision rule. Simulation on real a data set shows 86...97 % accuracy depending on number of coauthors and number of matching records.

Keywords: automatic learning, classification, e-library, bibliographic record, author name, disambiguation, duplicate search

References

1. **Afonin S. A., Gasparianc A. E.** Razreshenie neodnoznachnosti avtorstva publikacii pri avtomaticheskoi obrabotke bibliograficheskikh dannyh. *Programmnaya Ingeneria*, 2014, no. 1, pp. 25—29 (in Russian).
2. **Akbani R., Kwek S., Japkowicz N.** Applying support vector machines to imbalanced datasets. *In Proceedings of the 15th European Conference on Machine Learning (ECML)*, 2004, pp. 39—50.
3. **Vasenin V. A., Afonin S. A., Golomazov D. D., Kozitsyn A. S.** Intellectualnaia Sistema Tematicheskogo Issledovaniia Nauchnoi Informacii (ISTINA). *Informacionnoe Obschestvo*, 2013, no. 1—2, pp. 21—36 (in Russian).
4. **Sonnenburg S., Rätsch G., Henschel S., Widmer C., Behr J., Zien A., de Bona F., Binder A., Gehl C., Franc V.** The SHOGUN machine learning toolbox. *Journal of Machine Learning Research*, 2010, vol. 11, pp. 1799—1802.
5. **Bekkerman R., McCallum A.** Disambiguating Web Appearances of People in a Social Network. *Proceedings of the 14th International Conference on World Wide Web. WWW'05*. New York, NY, USA, ACM, 2005, pp. 463—470.
6. **Christen P.** A Comparison of Personal Name Matching: Techniques and Practical Issues. *Data Mining Workshops*, 2006.

ICDM Workshops 2006. Sixth IEEE International Conference on, 2006, pp. 290—294.

7. **Han H., Giles L., Zha H., Li C., Tsioutsoulklis K.** Two Supervised Learning Approaches for Name Disambiguation in Author Citations. *Proceedings of the 4th ACM/IEEE-CS Joint Conference on Digital Libraries. JCDL'04*. New York, NY, USA, ACM, 2004, pp. 296—305.
8. **Huang J., Ertekin S., Giles C. L.** Efficient Name Disambiguation for Large-scale Databases. *Proceedings of the 10th European Conference on Principle and Practice of Knowledge Discovery in Databases. PKDD'06*. Berlin, Heidelberg, Springer-Verlag, 2006, pp. 536—544.
9. **Song Y., Huang J., Council I. G., Li J., Giles C. L.** Efficient Topic-based Unsupervised Name Disambiguation. *Proceedings of the 7th ACM/IEEE-CS Joint Conference on Digital Libraries. JCDL '07*. New York, NY, USA, ACM, 2007, pp. 342—351.
10. **Fouzia S. Manjusha K.** Record Matching in Web Databases Using Unsupervised Approach. *International Journal of Scientific Engineering and Technology*, 2013, vol. 9, no. 2, pp. 895—899.
11. **Winkler W. E.** String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. *In Proceedings of the Section on Survey Research Methods*, 1990, pp. 354—359, available at: http://www.amstat.org/sections/srms/Proceedings/papers/1990_056.pdf

Ю. А. Киселёв, мл. науч. сотр., e-mail: ykiselev.loky@gmail.com,
 С. В. Поршнев, д-р техн. наук, проф., зав. каф., e-mail: sergey_porshnev@mail.ru,
 М. Ю. Мухин, д-р филол. наук, доц., проф., e-mail: mfly@sky.ru,
 Уральский федеральный университет имени первого Президента России Б. Н. Ельцина,
 г. Екатеринбург

Метод извлечения родовидовых отношений между существительными из определений толковых словарей

Предложен метод, позволяющий автоматизировать нахождение пар существительных, связанных родовидовыми отношениями, на основе обработки определений из толковых словарей. Получены количественные оценки точности и полноты предлагаемого метода, свидетельствующие о его эффективности, достаточной для использования этого метода при наполнении базы данных тезауруса родовидовыми отношениями. Проведенный среди носителей русского языка анализ согласованности при определении корректности выявленных родовидовых отношений позволил сделать вывод о возможности применения краудсорсинга для наполнения тезауруса родовидовыми парами.

Ключевые слова: тезаурус, словарь, семантические отношения, гипо-гиперонимические отношения, краудсорсинг, русский язык

Введение

Родовидовые (гипо-гиперонимические) отношения представляют собой разновидность семантических отношений между словами. Такая связь имеет иерархический характер и объединяет два слова: *гипоним* — слово с более частным (конкретным) значением по отношению к более общему понятию, и *гипероним* — слово с более общим значением, составляющим категорию, в которую попадают слова с более частным значением [1]. Например, для пары *клен—дерево* слово *клен* является гипонимом, а слово *дерево* — гиперонимом. Гиперонимы часто становятся основными операторами толкований гипонимов в словарях (*Клен — дерево с широкими резными листьями*). В реальной речи такая пара слов может проявлять функциональную эквивалентность, т. е. в широком смысле синонимию (*У дома рос клен. Дерево было очень высоким*). Родовидовые отношения наряду с меронимическими (отношениями части и целого: *страница—книга*) составляют основу "вертикальной" классификации семантических словарей, словарей-тезаурусов.

Под *тезаурусом* будем понимать особый вид словаря, отражающий семантические отношения между словами. Отношения и их типы — важные элементы структуры тезаурусов. Одним из ключевых отношений является синонимия, так как именно набор синонимов (или *синсет*, от англ. *synset — set of synonyms*) является основной структурной единицей, между которыми устанавливаются все прочие отношения,

такие как антонимия (в основном для прилагательных и наречий), тропонимия¹ (для глаголов), отношение ассоциативности и др.

Необходимо отметить, что семантическая организация лексики, в том числе формирование родовидовых связей между понятиями, — это один из наиболее важных видов структуризации лексики на основе смысловых отношений. Она является одним из основных направлений в комплексе исследований, посвященных системности словарного состава языка [3–5]. Описание семантической организации лексики имеет важное практическое значение. Знание родовидовых отношений между словами необходимо для изучения языка, а представление гипо-гиперонимических связей в машиночитаемом виде используется в системах информационного поиска, машинного перевода — т. е. в направлениях ИТ (информационных технологий), связанных с развитием семантического анализа и синтеза.

Учитывая, что отношение гипонимии является транзитивным и несимметричным [6] и что обычно у некоторого значения существует только один гипероним, данное отношение естественно представлять в виде некоторой иерархической структуры. Именно такое представление и используется в тезаурусах. Данный тип отношений вполне естественен для носителя языка, поэтому для его выявления

¹ Тип отношений, называемый тропонимией, можно описать следующим образом: делать *a* — это делать *b* особым способом [2], например *хромать—идти*.

человеку не требуется дополнительная подготовка. Это делает возможным ручное добавление гипонимических связей в базу данных тезауруса, однако этот процесс связан с некоторыми трудностями. Во-первых, установить наличие или отсутствие родовидовой связи между словами обычно быстрее и проще, чем определить род для имеющегося вида (сравните установление верности утверждения "*пот* — это жидкость" и определение рода для слова *пот*). Во-вторых, нужно собрать отдельные пары слов в иерархию, причем, с одной стороны, иерархия должна быть достаточно полной, т. е. многоуровневой, с другой стороны, число этих уровней не должно быть слишком большим, для того чтобы иерархию было удобно использовать. Как следствие, возникают вопросы при определении рода для того или иного понятия. Например, для слова *каменщик* родом может быть как *строитель*, так и *человек* или *живое существо*. Следовательно, "правильный" род будет зависеть от степени детализации конкретной иерархии. Поэтому определение требуемой granularity создаваемой иерархии на начальном этапе представляет собой существенную сложность.

В такой ситуации понятно, что полностью ручное установление родовидовых связей в тезаурусе — это процесс трудозатратный и достаточно медленный. Поэтому необходимо разрабатывать методы, позволяющие автоматизировать выделение родовидовых отношений. В то же время ясно, что задача выявления таких отношений является очень сложной, поэтому при ее решении исключить участие человека при настоящем уровне развития программных систем оказывается невозможным [7].

В настоящей статье описаны подход, позволяющий автоматизировать нахождение пар существительных, связанных родовидовыми отношениями (*родовидовые пары*), и количественные оценки его эффективности. Этот подход основан на анализе дефиниций толковых словарей, которые обычно конструируют с помощью некоторых шаблонов, из которых возможно извлечь информацию о роде описываемого слова. В то же время развитие задачи "связывание выявленных пар в иерархию" не являлось предметом данного этапа исследований.

1. Краудсорсинг как метод создания электронного тезауруса

В настоящее время в различных задачах автоматической обработки текстов активно используют *электронные тезаурусы* (ЭТ), например, бесплатный англоязычный ЭТ WordNet уже существует и поддерживается на протяжении нескольких десятилетий [2]. Такие тезаурусы используют для снятия лексической многозначности, расширения перечня слов при формулировке поисковых запросов, автоматической классификации и рубрикации информации и др. [1].

Один из возможных и наиболее очевидных подходов к созданию тезаурусов заключается в переводе уже существующего ресурса на другой язык [8].

Его преимущество основано на априорном предположении о том, что при переводе синсетов с языка оригинала тезауруса на другой язык сохраняются все отношения между ними, например родовидовая иерархия. По этой причине рассматриваемая задача может быть сведена к задаче перевода синсетов ЭТ на другой язык.

Однако на практике оказывается, что установление взаимно однозначного соответствия между синсетами на разных языках является непростой задачей, требующей для ее решения привлечения профессиональных лингвистов. При решении данной задачи, возникает в том числе сложность, связанная с наличием понятий, существующих в одном языке и нелексикализованных в другом. В датском языке, например, есть слово *hoofd*, означающее *голова человека*, однако аналогичное понятие в русском языке отсутствует. В связи с этим в переводной тезаурус неизбежно приходится вводить дополнительные отношения между словами и понятиями, отсутствующие в языке оригинала, и удалять имеющиеся. Отметим, что общепринятой концепции, следуя которой можно автоматизировать процесс перевода синсетов (в нашем случае с английского языка на русский) и обеспечить его приемлемое качество, на настоящее время не существует. Подробный анализ качества переводных ЭТ русского языка, в том числе открытого ЭТ, размещенного по адресу <http://wordnet.ru>, представлен в работе [9].

Отсутствие "стандартного" русскоязычного ресурса, удовлетворяющего всем требованиям пользователей ЭТ, привело к тому, что на настоящее время существует достаточно большое число проектов по созданию русскоязычных ЭТ, структурно независимых от уже имеющихся ЭТ для других языков. Одним из них является проект по созданию большого открытого русскоязычного ЭТ — YARN (Yet Another RussNet) [10], реализуемый с 2013 г. группой ученых из Уральского федерального университета (УрФУ) (г. Екатеринбург) и Высшей школы экономики (г. Москва). Главной целью проекта YARN является обеспечение пользователей сети Интернет свободно распространяемым (а потому — общедоступным) и дополняемым пользователями большим ЭТ русского языка. Важное преимущество YARN в сравнении с другими ЭТ (и не только русского языка) состоит в том, что работу над ним могут осуществлять не только непосредственные участники проекта, но и другие заинтересованные лица с совершенно различными навыками, интересами и уровнями лингвистических знаний. По сути, данный проект является экспериментом, цель которого состоит в сочетании традиционных принципов создания ЭТ и вики-подхода к наполнению и редактированию лингвистических ресурсов [11].

Подобный метод создания ресурса (краудсорсинг) позволяет применять смесь автоматического и ручного подходов, что, как ожидают разработчики YARN, позволит получить достаточно качественные ЭТ. В частности, его можно также применять к за-

даче определения родовидовых отношений между словами. Для этого на первом этапе необходимо разработать метод, позволяющий с достаточным качеством генерировать кандидаты в родовидовые пары. В связи с этим в следующем разделе рассмотрены существующие методы автоматического определения родовидовых отношений.

2. Обзор подходов

В одной из первых работ [12], которая описывает автоматическое извлечение слов, связанных родовидовыми отношениями, из текстов, написанных на естественном языке, использован метод, основанный на применении лексико-синтаксических шаблонов (ЛСШ). В качестве такого шаблона можно рассмотреть, например, следующий:

such NP as {NP,} {(or | and)} NP,*

перевод²:— *такая ИГ³ как {ИГ,*} {(и|или)} ИГ,*

где *{*}* означает, что заключенная в скобки часть может повторяться произвольное число раз, *{}* обозначает необязательную часть, *(a|b)* означает, что в конкретном месте может встретиться *a* или *b*. Примером, соответствующим данному шаблону, является выражение, "*такая ягода как, вишня или смородина*". Здесь слово *ягода* — это гипероним, а слова *вишня* и *смородина* — гипонимы. Разработанные шаблоны затем применялись к английским текстам, написанным на естественном языке. Каждый шаблон применяется независимо и детектирует различные языковые конструкции.

В силу того что в работе [12] были получены такие шаблоны, которые позволяли детектировать нужные отношения с хорошей точностью, оказалось, что число извлеченных пар в сравнении с общим объемом текста, на котором тестировались эти шаблоны, невелико. Точность данного метода по разным оценкам составляет 40...52 % [13]. Для выявления существенного числа отношений шаблоны, во-первых, должны часто встречаться в текстах, а во-вторых, они должны быть универсальными и присутствовать в текстах разных жанров. Универсальность шаблонов, разработанных М. Hearst, была продемонстрирована в работе [14] на примере извлечения родовидовых отношений из набора фармацевтических текстов. То есть даже на коллекции текстов определенного жанра с помощью ЛСШ были получены вполне удовлетворительные результаты. В этой же работе был изложен метод автоматического расширения списка ЛСШ, предложенных в работе [12]. Кроме того, известны и другие попытки автоматического выделения ЛСШ [15].

² Перевод примерный — он не учитывает грамматические особенности русского языка.

³ ИГ (именная группа, от англ. noun phrase — NP) — компонент предложения, который обладает синтаксическими свойствами существительного (может быть просто одним словом), например: *рукав моей куртки*.

Несмотря на относительно невысокую точность извлечения пар слов, связанных родовидовыми соотношениями, подход М. Hearst использовался и совершенствовался многими другими исследователями, а также применялся для других языков. Например, в работе [16] был осуществлен перевод этих шаблонов на русский язык, при этом авторы предложили несколько новых шаблонов, разработав их также вручную. По их оценкам, точность извлечения родовидовых отношений составила 79,5 %.

В работе [17] отмечено, что в некоторых подходах выделения родовидовых пар даже ошибочные пары могут быть использованы в дальнейшем. Так, например, неверно сгенерированные пары могут содержать меронимы (например, *перо* для категории *животные*) и просто слова, которые ассоциируются с заданной категорией (например, *зоопарк* — *животные*). Эта информация действительно может быть полезна для разных приложений и использоваться, например, для формирования прочих отношений тезауруса.

Отметим, что в описанных методах присутствует тот недостаток, что из отобранных отношений достаточно сложно сформировать полноценную иерархию связей. Однако известны работы, в которых данный подход использован для построения иерархии, например работа [18]. При этом было обнаружено, что отношениями связываются не сами слова, а их смыслы (в терминах тезаурусов — синсеты). Однако в иерархии, полученной в работе [18], каждое слово встречается ровно один раз и связано только с одним гиперонимом. На практике слово может встречаться в количестве выделяемых у него значений, причем использоваться оно должно не само по себе, а именно в составе синсетов.

Наиболее близким к подходу, предлагаемому в настоящей работе, оказывается подход, изложенный в работе [19]. Он основан на том, что в словарях толкования терминов обычно имеют определенные структуры, разбор которых позволяет выявлять семантические отношения между словами. Например, из определения вида "*Стебель — надземная часть растения ...*" можно выделить отношение вида часть — целое между словами *стебель* и *растение*. Данный подход позволяет детектировать не только отношения, но и некоторые свойства у понятий, такие как место, время, объект и пр. По оценкам авторов, средняя точность извлечения различных типов получаемых отношений оказалась равной 78 %, а родовидовых отношений — 87 %.

Рассмотренные выше подходы позволяют сделать вывод о том, что задача выделения родовидовых отношений может решаться по-разному. Отметим, что применительно к русскому языку удалось обнаружить единственную работу [17], посвященную автоматизации поиска родовидовых отношений. Однако полученные там результаты так и не были использованы разработчиками тезаурусов русского языка. При этом оказывается, что выделять родовидовые отношения с приемлемой точностью удается даже при использовании достаточно простых методов. Этот факт особенно важен при построении ЭТ методом

краудсорсинга, так как позволяет в кратчайшие сроки реализовать программное приложение для извлечения пар-кандидатов с точностью, достаточной для того, чтобы далее их было целесообразно размечать человеку.

Отметим, что краудсорсинговый подход, используемый для наполнения ЭТ YARN, описанный в разд. 1, можно применять при выделении, например, родовидовых отношений. Можно ожидать, что он позволит снизить требования к разрабатываемому методу автоматической генерации родовидовых пар, так как необходимо получить не гарантированные пары, а лишь вероятные. При этом понятно, что основным критерием качества метода извлечения синонимических отношений является его точность (доля верных пар от общего числа сгенерированных), так как именно она оказывает основное влияние на скорость человеческой разметки. Однако получить формальную оценку минимально приемлемой точности весьма непросто, в то время как сформулировать разумное требование для качества генерируемых кандидатов несложно. Оно должно существенно уменьшать время, необходимое для фильтрации пар слов, действительно связанных родовидовыми отношениями, по сравнению с полностью ручным формированием таких пар.

Предлагаемый метод получения вероятных родовидовых пар описан в следующем разделе.

3. Метод автоматического извлечения родовидовых отношений из словарных определений

Результаты анализа существующих методов извлечения родовидовых отношений позволили сделать вывод о том, что наиболее популярными на настоящее время являются методы, основанные на ЛСП. Успешный опыт их применения для выделения родовидовых отношений из англоязычных толковых словарей описан в работе [19]. Принимая во внимание наличие толковых словарей русского языка, переведенных в электронный вид, представляется целесообразным провести анализ возможности использования данного подхода для установления родовидовых отношений между словами русского языка.

3.1. Описание метода

Лексическое значение, отражающее связь между словами и явлениями реальной действительности, фиксируется в словаре в виде определения или толкования. Существуют различные подходы к построению определений лексикализованных концепций [20]. Первый подход заключается в том, что определение должно содержать информацию, достаточную для точного воспроизведения соответствующей концепции. Однако определения, соответствующие данному подходу, являются чрезвычайно громоздкими и на практике встречаются достаточно редко. По этой причине есть все основания полагать, что определение большинства словарей не удовлетво-

ряют этому требованию [21]. Согласно второму подходу определение должно быть таким, чтобы при его использовании описываемую сущность можно было отличить от любой другой [20]. Другими словами, дефиниция должна содержать информацию, достаточную для человека, знакомого с данной предметной областью, с тем, чтобы он мог ее идентифицировать. В работе [20] также отмечено, что в большинстве случаев для этого достаточно перечисления нескольких синонимов, возможно, даже одного. Этот способ широко используется в толковых словарях, например (здесь и далее примеры определений даны из Малого академического словаря, МАС [22]):

- *булат* — *стальной клинок, меч;*
- *ищейка* — *сыщик, шпион.*

В случае имен существительных другим универсальным способом их определения является указание рода для описываемого понятия с уточнением отличительных свойств, например:

- *звукоспись* — *один из видов инструментовки стиха, заключающийся в...;*
- *лавсан* — *вид синтетического волокна, получаемого из...*

В большинстве случаев часть *вид, один из видов* и т. п. опускается, неявно при этом подразумеваясь, например:

- *ненависть* — *чувство сильнейшей вражды, неприязни;*
- *грызун* — *травоядное млекопитающее (белка, суслик...) с длинными, сильно развитыми передними зубами.*

Рассмотрение определений существительных показывает, что род описываемого понятия не просто встречается, но часто оказывается первым существительным в дефиниции. Из приведенных выше определений: *грызун* (вид) — *млекопитающее* (род), *ненависть* — *чувство*. В большинстве случаев первое существительное стоит в именительном падеже, хотя возможно, и во множественном числе, например:

- *концентры* — *круги различной величины, имеющие общий центр.*

Дальнейший анализ показывает, что в определениях, являющихся распространенными предложениями, род понятия обозначает первое существительное, стоящее именно в именительном падеже, например:

- *перина* — *набитый пухом или пером продолговатый мешок, который...*
- *клирошанин* — *поющий на клиросе причетник, дьячок.*

При этом в ряде случаев родовое понятие оказывается первым существительным, хотя и не в именительном падеже, например:

- *молокане* — *одна из сект, вышедших из православия...*
- *пигмей* — *о человеке очень низкого роста.*

Таким образом, для определения вероятного рода существительного *w* при наличии его определения *def* можно применить следующий алгоритм:

- 1) выбрать первое существительное *s1* из определения *def*;

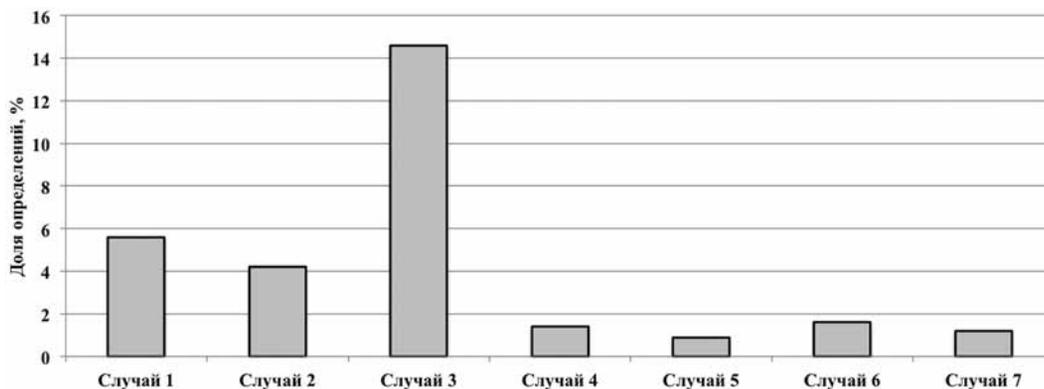


Рис. 1. Гистограмма распределения проблемных случаев, связанных с определениями, по долям

2) в случае если падеж *c1* не именительный, выбрать из определения *def* первое существительное *c2*, стоящее в именительном падеже.

Проведенный выше анализ показывает, что одно из слов *c1* или *c2* является вероятным родом для существительного *w*. Причем ожидается, что для *w* будет только один кандидат для проверки, так как все-таки обычно первое существительное стоит в именительном падеже.

Для практического использования описанного метода извлечения родовидовых пар была написана программа на языке программирования Python⁴ с использованием только его стандартной библиотеки. Выбор данного языка был обусловлен возможностью создания на нем кросс-платформенных приложений, а также наличием средств работы с текстовой информацией. Данные словаря МАС, из которого извлекались родовидовые пары, были представлены в кодировке UTF-8, поддержка которой есть в Python.

3.2. Уточнение метода

Описанный выше алгоритм был опробован⁵ на определениях существительных из словаря МАС. Для этого из 1000 случайно выбранных определений были автоматически сгенерированы пары *w — c1* и *w — c2*. Далее был проведен ручной анализ найденных пар, позволивший выделить типичные проблемные случаи, которые при этом появляются. К их числу относятся перечисленные далее.

1. Определения, содержащие только один синоним (типичные конструкции "*см. слово*", "*то же, что слово*" и др.). Для предлагаемого подхода к решению задачи определения родовидовых отношений такие "синонимические" определения бесполезны, поэтому из них кандидаты не выбирались. При этом авторы не делали попытку детектировать определения, в которых дается целый синонимический ряд, так как

⁴ Официальный сайт <https://www.python.org/>

⁵ Вся необходимая грамматическая информация о словах (часть речи, число и падеж) была получена с помощью программного средства *mystem* (<http://tech.yandex.ru/mystem>).

он может содержать и словосочетания, что бывает достаточно сложно определить.

2. Определения, описывающие производные от существительных, часто — с использованием стилистических помет, из которых невозможно извлечь род (например, "*молочко — ласк. к молоко*"). Был составлен список таких помет; определения, их содержащие, не рассматривались.

3. Определения, являющиеся отсылочными (вида "*абстрагирование — действие по знач. глаг. абстрагироваться*"). Здесь формально из них можно выделить род, однако он получается очень общим. По этой причине генерируемая пара слов является плохим кандидатом для включения в тезаурус. Подобные отсылочные определения не рассматривались.

4. Существительные, определяющиеся как часть чего-то (например, "*запястье — часть кисти руки ...*"). Такое определение свидетельствует о наличии отношения часть — целое между определяемым словом и существительным следующим за словом *часть*, а не о родовидовом. Такие определения не рассматривались.

5. Определения, содержащие вводные части (отделяемые обычно двоеточием или тире, например: "*бурлак — в старину: рабочий на реке...*"), которые, ожидается, не содержат родовое понятие для описываемого слова. Такие вводные части из определений удалялись.

6. Определения, которые первым существительным в явном виде содержат леммы *род*, *вид*, *сорт* и т. п. Был составлен список таких слов. Формы этих слов игнорировались, т. е. в кандидаты выбирались следующие за ними существительные. Отметим, что подобные определения содержат само родовое понятие не в именительном падеже (например, "*мото-роллер — разновидность мотоцикла...*").

7. Определения, составленные по схеме "*слово — название [рода/вида/сорта]*" (например, "*Абиссинцы — устарелое название населения Эфиопии*"). В таких конструкциях слово *название* и слова из списка из п. 6 игнорировались.

Распределение выявленных проблемных случаев по долям от общего числа определений приведено на рис. 1.

На рис. 1 видно, что проблемный случай 1 возник в 5,6 % случаев от общего числа определений; случай 2 — в 4,2 %; случай 3 — в 14,6 %; случай 4 — в 1,4 %; случай 5 — в 0,9 %; случай 6 — в 1,6 %; случай 7 — в 1,2 %. Отмеченные проблемные случаи, встречающиеся почти в 30 % всех определений, были автоматически разрешены.

Количественные оценки эффективности предлагаемого метода рассмотрены в следующем разделе.

4. Эксперименты и результаты

Описанный метод извлечения родовидовых отношений между словами русского языка является полностью автоматическим, однако оценить его точность без участия человека (носителя русского языка) оказывается затруднительным. В связи с этим для оценки точности извлечения родовидовых отношений был проведен опрос среди носителей русского языка.

4.1. Методика проведения эксперимента

Для оценки точности извлечения родовидовых пар и обоснования возможности использования предложенного метода для формирования родовидовых связей тезауруса с последующим формированием родовидовой иерархии было использовано анкетирование. Анкета, пример которой приведен в Приложении к статье, состояла из двух частей: инструкция для оценки, включающая в себя определение родовидовых отношений и их примеры, и 50 пар слов, которые необходимо было оценить. Для каждой пары предлагалось ответить на следующий вопрос: "Является ли второе слово общим понятием по отношению к первому в каком-либо его значении?" Участникам следовало дать один из ответов:

- 1) да;
- 2) да, но между этими понятиями пропущено смысловое звено;
- 3) нет, слова являются синонимами;
- 4) нет;
- 5) не знаю.

Ответ 2 был предусмотрен, чтобы оценить близость правильно извлеченных пар в родовидовой иерархии. Например, пара *хирург — человек* является родовидовой, но здесь естественно выделить еще одну промежуточную категорию — *врач* или *доктор*.

Всего в опросе приняло участие 18 человек. Все они являлись носителями русского языка; других требований к ним не предъявлялось, хотя более половины участников — ученики филологического факультета. Всего были получены оценки по 300 различным парам. Чтобы учесть возможные ошибки при разметке и случаи, которые даже носителям языка сложно трактовать однозначно, каждая пара оценивалась тремя различными экспертами. Таким образом, всего было получено 900 оценок.

Пары слов, которые оценивались информантами, были сформированы следующим образом.

1. 100 пар было получено путем случайного выбора из всего набора пар, выявленных из словаря.

2. 100 пар было получено путем случайного выбора из тех, в которых видовое понятие входит в 2000 самых часто употребительных⁶ (частотных) слов русского языка. При этом в качестве определений, которые обрабатывались в целях выделения из них родового понятия, выбиралось первое (возможно, единственное), что обычно соответствует основному значению трактуемого слова.

3. 100 пар было получено из определений слов группы "Одежда". Во-первых, авторы хотели поработать с целым семантическим классом. Во-вторых, эта группа состоит из конкретных по значению слов, и в ней достаточно прозрачно можно выделить родовидовые отношения.

Оценки по категории 1 (далее называемая категория "Случайные") позволяют делать выводы о точности предлагаемого метода в общем; оценки по категории 2 (категория "Частотные") — о точности метода относительно тех слов и значений, которые, очевидно, должны быть в тезаурусе общего назначения; оценки по категории 3 (категория "Одежда") — о точности метода по некоторой предметной области. Всего было сформировано шесть анкет, каждая из которых содержала пары из всех трех категорий в равных пропорциях.

4.2. Оценка точности метода

Для оценки точности предлагаемого метода пять типов ответов приводились к трем: верная пара, ошибочная пара, неизвестно. Приведение оценок выполнялось тремя способами, указанными в таблице.

Из таблицы видно, что в случае сильного приведения верной парой считаются только "близкие" родовидовые пары, в случае среднего — любые, в случае слабого — все родовидовые пары и синонимы. С точки зрения решаемой задачи выявление синонимичных пар вместо родовидовых является ошибкой, однако их получение само по себе является важным и полезным результатом. Действительно, синонимы — это базовые структурные элементы тезауруса, которые необходимы для его создания.

В ходе анкетирования для каждой родовидовой пары были получены три оценки, которые далее приводились к одной общей с помощью следующей процедуры: для каждой пары брали оценку, которая была поставлена хотя бы два раза (по приведенной тернарной шкале). Эта оценка считалась единственно верной для пары. В связи с этим точность нахождения родовидовых пар далее оценивали как отношение верных пар к их общему количеству.

При этом пары, у которых все три оценки оказывались различными или преобладала оценка "неизвестно", исключались, потому что для них нельзя сделать однозначного вывода о правильности или ошибочности.

⁶ Данная информация была получена на основе списка частот словоформ, взятых с сайта проекта Национального корпуса русского языка (<http://www.ruscorpora.ru>).

Виды приведений пяти типов ответов к тернарной шкале

Типы ответов	Сильное приведение	Среднее приведение	Слабое приведение
Да ("близкая" родовидовая пара)	Верная пара	Верная пара	Верная пара
Да, но между этими понятиями пропущено смысловое звено	Ошибочная пара	Верная пара	Верная пара
Нет, слова являются синонимами	Ошибочная пара	Ошибочная пара	Верная пара
Нет	Ошибочная пара	Ошибочная пара	Ошибочная пара
Не знаю	Неизвестно	Неизвестно	Неизвестно

сти определения родовидовой пары. Всего из 300 были исключены 15, 28 и 37 пар соответственно для сильного, среднего и слабого приведений оценок (5, 9 и 12 %). Очевидно, что все пары, являющиеся верными для сильного приведения, являются также верными для среднего и слабого приведения, а верные пары для среднего приведения — верными для слабого, поэтому вклад каждого из этих типов естественно представить в виде накопительной диаграммы (рис. 2).

Из данных, приведенных на рис. 2, можно сделать следующие выводы.

1. Самая низкая точность извлечения "близких" родовидовых пар у категории "Случайные" (46 %), а с учетом всех родовидовых пар — у категории "Одежда" (60 %). Вероятно, последнее объясняется спецификой выборки, а значит, можно считать, что точность выявления родовидовых пар для конкретной предметной области не отличается от случайных.

2. Самая высокая доля правильно извлеченных родовидовых пар, в том числе и с учетом синонимов, у категории "Частотные". Этот факт означает, что при выявлении родовидовых отношений между словами, которые должны быть в универсальном тезаурусе, точность предлагаемого метода составляет 68 %.

3. Среди всех извлеченных пар семантической близостью обладают 76 %, а среди частотных слов — 84 %. Значит, разметка таких пар является "полезной", так как они могут быть включены в тезаурус.

4. 12...16 % слов определяются через синонимы и при этом не являются отсылочными. Отметим, что из подобных определений извлечь родовые понятия нельзя, так как они там отсутствуют.

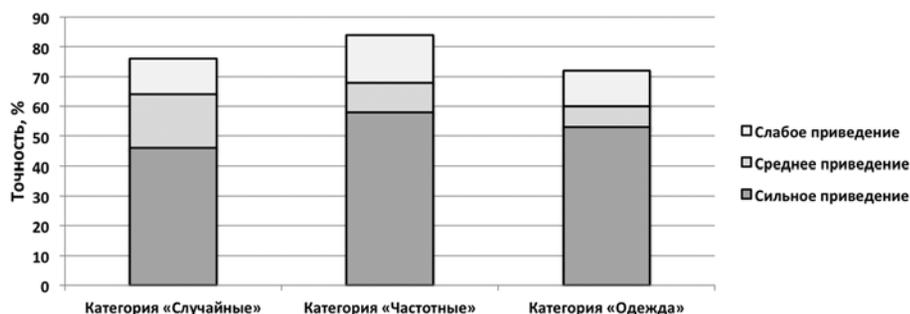


Рис. 2. Точность метода извлечения родовидовых пар для разных видов приведений оценок

4.3. Оценка согласованности оценок

Как было отмечено выше, при среднем приведении, когда верными считаются любые родовидовые пары и только они, было исключено 28 пар, т. е. в 9 % случаев носителям языка не удастся достигнуть согласия по поводу их верности или ошибочности. В связи с этим возникает объективный вопрос о возможности использования краудсорсинга с получением нескольких оценок на каждую пару с последующим мажорированием и выявлением единой оценки для определения верных родовидовых пар и формирования иерархии.

Для ответа на данный вопрос была получена количественная оценка согласованности оценок между разными информантами. В этих целях была вычислена мера согласованности между всеми парами людей, оценивавшими одинаковые анкеты. В качестве попарной меры согласованности была выбрана каппа-статистика k [23], которая вычисляется следующим образом. Пусть есть n оцениваемых пар и множество возможных оценок мощности m . Обозначим через f_{ij} число ситуаций, когда один человек выбрал i -ую категорию ответа, а второй — j -ую. Тогда можно рассчитать наблюдаемую долю согласия в оценках p_0 и ожидаемую долю случайного согласия p_e :

$$p_0 = \frac{1}{n} \sum_{i=1}^m f_{ii},$$

$$p_e = \frac{1}{n^2} \sum_{i=1}^m f_{i+} f_{+i},$$

$$\text{где } f_{i+} = \sum_{j=1}^m f_{ij}; \quad f_{+i} = \sum_{j=1}^m f_{ji}.$$

Тогда каппа-статистика k может быть вычислена следующим образом:

$$k = \frac{p_0 - p_e}{1 - p_e}.$$

Значения каппа-статистики всегда не больше 1 (1 соответствует

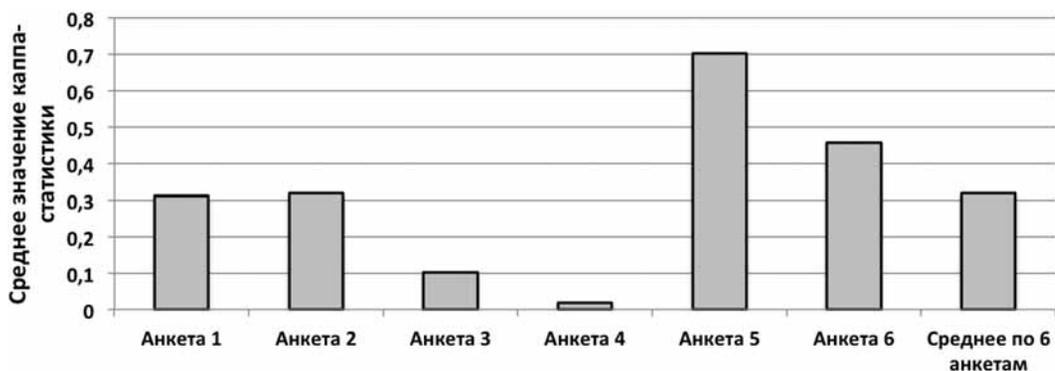


Рис. 3. Средние значения каппа-статистики для всех анкет

идеальной согласованности), 0 соответствует случайному согласию, значения, меньшие 0, — "расхождения больше, чем может объяснить случайность" [23]. Каппа-статистика учитывает согласие мнений двух экспертов. В случае, когда экспертов больше двух, вычисляется среднее попарных значений каппа-статистики [23].

Средние попарные значения каппа-статистики, вычисленные по анкетам, представлены на рис. 3. Для получения соответствующих значений использовалось среднее приведение с последующим удалением пар, в которых хотя бы от одного человека была получена оценка "неизвестно". Таким образом, множество оценок состояло из двух: верная пара и ошибочная.

На рис. 3 видно, что разброс значений достаточно большой: от 0,01 до 0,70, при этом среднее значение равно 0,32. Согласно различным исследованиям поведения каппа-статистики (см., например, работы [24, 25]), это значение можно интерпретировать по-разному. Однако, по мнению многих исследователей, значения k , большие 0,2, свидетельствуют о наличии определенного согласия между оценками, хотя и небольшого.

В связи с этим можно сделать вывод, что полученное среднее значение каппа-статистики свидетельствует, что носители языка могут договориться между собой, какие понятия можно считать родовидовыми, а какие нет. Следовательно, задачу определения рода для понятия действительно можно решить с помощью краудсорсинга, обязательно собирая несколько оценок на каждую пару. Участие специалиста необходимо лишь в 9 % случаев, когда носители языка расходятся во мнении по поводу правильности или ошибочности предлагаемых родовидовых пар.

4.4. Итоговые результаты

Предложенный метод был применен к словарю МАС. Всего этот словарь содержит 33,5 тыс. существительных, для которых дана 51 тыс. определений. Хотя бы одну пару удалось извлечь из 37,2 тыс. определений, всего было получено 38,7 тыс. пар, в том числе 36,8 тыс. разных⁷. Из этих данных видно, что

⁷ Отметим, что даже из разных значений слов могут извлекаться одинаковые пары, например: "граната — разрывной артиллерийский снаряд", "граната — в легкой атлетике: снаряд для метания".

из 1,5 тыс. определений (что составляет 4 % от их общего числа) было извлечено две пары.

Учитывая полученные в разделе 4.2. значения точности метода, можно сделать вывод, что из МАС можно верно извлечь 23,5 тыс. родовидовых пар и 28,0 тыс. семантически близких слов (т. е. в том числе и синонимов). Эти значения свидетельствуют о существенной полноте предлагаемого метода.

Отметим, что по понятным причинам ошибки метода носят случайный характер, т. е. в качестве ошибочного родового понятия выбираются не одинаковые слова, а разные. Поэтому, если некоторое понятие выбирается как родовое во многих парах, то это с большой вероятностью означает, что оно должно быть представлено в родовидовой иерархии, так как случайные ошибки не должны приводить к появлению "крупных" родов. Тем не менее, как уже отмечалось во введении, связывание выявленных пар в иерархию не являлось предметом данного этапа проводимых исследований.

Заключение

Результаты анализа предлагаемого метода извлечения родовидовых отношений между существительными из определений толковых словарей позволяют сделать следующие выводы.

- Средняя точность предлагаемого метода составляет 46 % для "близких" родовидовых пар и 64 % — для пар, в которых, возможно, пропущено смысловое звено (58 и 68 % соответственно для пар, полученных из слов, где гипонимом является часто используемое слово). Эта точность сравнима с методами, предлагаемыми для английского языкового материала.

- Точность извлечения родовидовых пар для некоторой тематической категории несущественно отличается от случайно выбранных пар.

- Проведенный анализ результатов извлечения родовидовых пар показал, что 23,5 тыс. пар было извлечено верно. Этого количества достаточно для формирования полноценной иерархии. Так, например, в работе [16], в которой также использовался русский языковой материал, сообщают о 3,9 тыс. верных парах.

• Задачу определения рода для понятия можно решить с помощью краудсорсинга. Для этого необходимо получение нескольких оценок для каждой извлеченной родовидовой пары. Неоднозначные пары должны передаваться для разметки специалистам.

• Предложенный метод позволяет выделить слова, которые часто являются родом, т. е. понятия-кандидаты, в узлы, которые формируют общую родовидовую иерархию. Разработка автоматического метода визуализации родовидовой иерархии является предметом дальнейших исследований.

Все кандидаты в родовидовые пары, полученные из словаря МАС в ходе проведенного исследования, будут использованы в проекте создания большого открытого ЭТ YARN в целях верификации их правильности и связывания концепций в единую родовидовую иерархию.

Исследование выполняется при финансовой поддержке РГНФ (проект № 13-04-12020 "Новый открытый электронный тезаурус русского языка") и научной группы "Разработка методов анализа, обработки, визуализации и прогнозирования многомерных данных для современных информационных систем" Уральского федерального университета имени первого Президента России Б. Н. Ельцина.

Авторы выражают глубокую признательность всем, кто принимал участие в проведенном в ходе исследования анкетировании.

Список литературы

1. Лукашевич Н. В. Тезаурусы в задачах информационного поиска. М.: Изд-во МГУ, 2011. 512 с.
2. Fellbaum C. WordNet: An Electronic Lexical Database. Cambridge, 1988. 447 p.
3. Апресян Ю. Д. Лексическая семантика (синонимические средства языка). М.: Наука, 1974. 367 с.
4. Караулов Ю. Н. Лингвистическое конструирование и тезаурус литературного языка. М.: Наука, 1981. 367 с.
5. Кронгауз М. А. Семантика: учеб. для вузов. М.: Рос. гос. гуманит. ун-т, 2001. 399 с.
6. Lyons J. Semantics. Vol 2. New York: Cambridge University Press, 1977. 540 p.
7. Лукашевич Н. В. Проблемы установления родовидовых отношений в лингвистических онтологиях // Тр. Всерос. конф. "Знания-Онтологии-решения" (ЗОНТ-07). Новосибирск, 2007. С. 211—220.
8. Vossen P. WordNet, EuroWordNet and Global WordNet // J. Revue française de linguistique appliquée. 2002. Vol. 7, N 1. P. 27—38.

ИНСТРУКЦИЯ

для оценки родовидовых отношений между словами

1. Родовидовые отношения — это отношения между частным и общим понятиями: *роза — цветок, любовь — чувство, бег — движение.*

2. Слова, имеющие несколько значений, могут образовывать различные пары: *окно — отверстие* или *проем, окно — просвет* (например, между тучами), *окно — промежуток* или *интервал* (между занятиями, поездами), *лиса — животное, лиса — человек.*

9. Киселёв Ю. А., Поршнева С. В., Мухин М. Ю. Современное состояние электронных тезаурусов русского языка: качество, полнота и доступность // Программная инженерия. 2015. № 6. С. 34—40.

10. Yet Another RussNet. URL: <http://russianword.net> (дата обращения: 20.06.2015).

11. Браславский П. И., Мухин М. Ю., Ляшевская О. Н. и др. YARN: начало // Компьютерная лингвистика и интеллектуальные технологии: конференция "Диалог-2013": Доклады, принятые к публикации на сайте. Москва, 2013. URL: http://www.dialog-21.ru/digests/dialog2013/materials/pdf/Braslavskiy_P_YARN.pdf (дата обращения: 20.06.2015).

12. Hearst M. Automatic acquisition of hyponyms from large text corpora // Proc. of the 14th conf. on Computational linguistics. Vol. 2. Stroudsburg, PA, USA: ACL, 1992. P. 539—545.

13. Cederberg S., Widdows D. Using LSA and noun coordination information to improve the precision and recall of automatic hyponymy extraction // Proc. of the 7th conf. on Natural language learning at HLT-NAACL 2003, Vol. 4. Stroudsburg, PA, USA, ACL, 2003. P. 111—118.

14. Oakes M. Using Hearst's Rules for the Automatic Acquisition of Hyponyms for Mining a Pharmaceutical Corpus // Proc. of the Workshop Text Mining Research, 24 Sept. 2005. Borovets, Bulgaria, 2005. P. 63—67.

15. Snow R., Jurafsky D., Ng V. A. Learning Syntactic Patterns for Automatic Hypernym Discovery // Proc. of NIPS 2004/ ed. L. K. Saul, Y. Weiss, L. Bottou. Cambridge, Mass: MIT Press, 2004. P. 1297—1304.

16. Sabirova K., Lukanin A. Automatic Extraction of Hypernyms and Hyponyms from Russian Texts // Supplementary Proc. of the 3rd Int. Conf. AIST-2014. Vol. 1197. Yekaterinburg, Russia: CEUR-WS.org, 2014. P. 35—40.

17. Roark B., Charniak E. Noun-phrase co-occurrence statistics for semiautomatic semantic lexicon construction // Proc. of the 36th Annual Meeting of the ACL and 17th Int. Conf. on Computational Linguistics, Vol. 2. Stroudsburg, PA, USA: ACL, 1998. P. 1110—1116.

18. Caraballo S. Automatic Acquisition of a Hypernym-labeled Noun Hierarchy from Text // Proc. of the 37th Annual Meeting of the ACL. Baltimore, MD, 1999. P. 120—126.

19. Dolan W., Vanderwende L., Richardson S. Automatically Deriving Structured Knowledge Bases From on-Line Dictionaries // Proc. of the Pacific Association for Computational Linguistics. PACL Press, 1993. P. 5—14.

20. Miller G. A., Beckwith R., Fellbaum C. et al. Introduction to Wordnet: An Online Lexical Database // Int. J. of Lexicography. 1990. Vol. 3, N 4. P. 235—244.

21. Miller G. A., Gildea P. M. How Children Learn Words // J. Scientific American. 1987. Vol. 257, N 3. P. 94—99.

22. Малый академический словарь: в 4-х т. / Под ред. А. П. Евгеньевой. 4-е изд. М.: Рус. яз.; Полиграфресурсы, 1999.

23. Маннинг К., Рагхаван П., Шютце Х. Введение в информационный поиск: Пер. с англ. М.: Вильямс, 2011. 528 с.

24. Landis J. R., Koch G. G. Measurement of Observer Agreement for Categorical Data // J. Biometrics. 1977. Vol. 33, N 1. P. 159—174.

25. Altman D. G. Practical Statistics for Medical Research. 1st ed. Chapman and Hall/CRC, 1990. 624 pp.

Приложение

3. Возможна ситуация, когда между общим и частным понятиями существуют другие смысловые звенья:

хирург — человек → хирург — врач
(или *доктор*) — человек

изба — строение → изба — дом — строение

4. Слова-синонимы примером отношений общего и частного не являются: *стюардесса — бортпроводница, лиса — хитрец.*

Вопрос для оценки: является ли второе слово общим понятием по отношению к первому в каком-либо его значениях?

- Варианты ответа
1. **Да** (поставить знак "+"). См. примеры 1, 2 (в таблице).
 2. **Нет** (поставить знак "-"). См. пример 4.

3. **Нет, слова являются синонимами** (вписать слово "Синоним"). См. пример 3.
4. **Не знаю** (поставить знак "?"). См. пример 5.
5. **Да, но между этими понятиями пропущено смысловое звено** (вписать это слово). См. пример 6.

ЗАДАНИЕ

Пожалуйста, оцените следующие пары слов

№	Слово 1 — частное	Слово 2 — общее	Ваша оценка
1	слово	слово	
2	слово	слово	
...			
50	слово	слово	

Примеры ответов

№	Слово 1 — частное	Слово 2 — общее	Оценка
1	Дуб	Дерево	+
2	Дуб	Человек	+
3	Дуб	Дурак	Синоним
4	Дуб	Инструмент	-
5	Дуб	Древесина	?
6	Дуб	Растение	Дерево

Yu. A. Kiselev, Junior Research Fellow, e-mail: yuri.kiselev@urfu.ru, **S. V. Porshnev**, Head of Department, e-mail: sergey_porshnev@mail.ru, **M. Yu. Mukhin**, Professor, e-mail: mfly@sky.ru, Ural Federal University named after the first President of Russia B. N. Yeltsin, Ekaterinburg

Method of Extracting Hyponym-Hypernym Relationships for Nouns from Definitions of Explanatory Dictionaries

Method allowing automating retrieving the pairs of nouns linked with hyponym-hypernym relations based on the processing of definitions from explanatory dictionaries is proposed. Quantitative assessments of the precision and the recall of the proposed method, testifying that the effectiveness of the method is sufficient for filling the database thesaurus with hyponym-hypernym relationships, are obtained. The average precision of the method is 0.58. The precision equals to 0.68 for the most frequent words of the Russian language. It's ascertained that the precision for the words belonging to a certain semantic class does not significantly vary from random words. 23,500 hyponym-hypernym pairs were correctly retrieved from the Dictionary of the Russian Language. Such recall is enough for the further constructing of hyponym-hypernym hierarchy between concepts, based on these pairs. All the retrieved data will be used in YARN project — a large open WordNet-like machine-readable thesaurus for the Russian language through crowdsourcing. The analysis of agreement between Russian native speakers in determining the correctness of the retrieved hyponym-hypernym pairs let us to make a conclusion about the possibility of the use of crowdsourcing for filling the thesaurus.

Keywords: thesaurus, dictionary, semantic relationships, hyponym-hypernym relationships, crowdsourcing, Russian language

References

1. **Lukashevich N. V.** *Tezaurusy v zadachah informacionnogo poiska* (Thesauri in application to problems of information retrieval). Moscow: Izd-vo MSU, 2011, 512 p. (in Russian).
2. **Fellbaum C.** *WordNet: An Electronic Lexical Database*. Cambridge, 1988. 447 p.
3. **Apresjan Ju. D.** *Leksicheskaja semantika (sinonimicheskie sredstva jazyka)* (Lexical semantics (synonymy means of language)), Moscow, Nauka, 1974, 367 p. (in Russian).
4. **Karaulov Ju. N.** *Lingvisticheskoe konstruirovanie i tezaurus literaturnogo jazyka* (Linguistic engineering and thesaurus of literary language). Moscow, Nauka, 1981, 367 p. (in Russian).

5. **Krongauz M. A.** *Semantika: Uchebnik dlja vuzov* (Semantics: Textbook for universities.). Moscow: Russian State University for the Humanities, 2001, 399 p. (in Russian).
6. **Lyons J.** *Semantics*. Vol 2. New York: Cambridge University Press, 1977. 540 p.
7. **Lukashevich N. V.** Problemy ustanovlenija rodovidovyh ot-noshenij v lingvisticheskix ontologijah. *Trudy vsrossijskoj konferencii "Znanija-Ontologii-Teorii" (ZONT-07)*. Novosibirsk, 2007, pp. 211–220 (in Russian).
8. **Vossen P.** WordNet, EuroWordNet and Global WordNet. *J. Revue francaise de linguistique appliquee*, 2002, vol. 7, no. 1, pp. 27–38.
9. **Kiselev Yu. A., Porshnev S. V., Mukhin M. Yu.** Sovremennoe sostojanie jelektronnyh tezaurusov russkogo jazyka: kachest-

vo, polnota i dostupnost'. *Programmnaya Ingeneria*, 2015, no. 6, pp. 34–40 (in Russian).

10. **Yet Another RussNet**, available at: <http://russianword.net> (accessed 20.06.2015).

11. **Braslavskij P. I., Muhin M. Ju., Ljashevskaja O. N., Bonch-Osmolovskaja A. A., Krizhanovskij A. A., Egorov P. E.** YARN: nachalo. *Konferencija "Dialog-2013": Doklady, prinjatye k publikacii na sajte*, 2013, available at: http://www.dialog-21.ru/digests/dialog2013/materials/pdf/BraslavskiyP_YARN.pdf (accessed 20.06.2015) (in Russian).

12. **Hearst M.** Automatic acquisition of hyponyms from large text corpora. *Proc. of the 14th conf. on Computational linguistics*, Vol. 2. Stroudsburg, PA, USA, ACL, 1992, pp. 539–545.

13. **Cederberg S., Widdows D.** Using LSA and noun coordination information to improve the precision and recall of automatic hyponymy extraction. *Proc. of the 7th conf. on Natural language learning at HLT-NAACL 2003*, Vol. 4. Stroudsburg, PA, USA, ACL, 2003, pp. 111–118.

14. **Oakes M.** Using Hearst's Rules for the Automatic Acquisition of Hyponyms for Mining a Pharmaceutical Corpus. *Proc. of the Workshop Text Mining Research*, 24 September. 2005. Borovets, Bulgaria, 2005, pp. 63–67.

15. **Snow R., Jurafsky D., Ng V. A.** Learning Syntactic Patterns for Automatic Hyponym Discovery. *Proc. of NIPS 2004* / ed. L. K. Saul, Y. Weiss, L. Bottou. Cambridge, Mass, MIT Press, 2004, pp. 1297–1304.

16. **Sabirova K., Lukanin A.** Automatic Extraction of Hyponyms and Hyponyms from Russian Texts. *Supplementary Proc. of the 3rd*

Int. Conf. AIST-2013, vol. 1197. Yekaterinburg, Russia, CEUR-WS.org, 2014, pp. 35–40.

17. **Roark B., Charniak E.** Noun-phrase co-occurrence statistics for semiautomatic semantic lexicon construction. *Proc. of the 36th Annual Meeting of the ACL and 17th Int. Conf. on Computational Linguistics*, Vol. 2. Stroudsburg, PA, USA: ACL, 1998, pp. 1110–1116.

18. **Caraballo S.** Automatic Acquisition of a Hyponym-labeled Noun Hierarchy from Text. *Proc. of the 37th Annual Meeting of the ACL*. Baltimore, MD, 1999, pp. 120–126.

19. **Dolan W., Vanderwende L., Richardson S.** Automatically Deriving Structured Knowledge Bases From on-Line Dictionaries. *Proc. of the Pacific Association for Computational Linguistics*. PACL Press, 1993, pp. 5–14.

20. **Miller G. A., Beckwith R., Fellbaum C., Gross D., Miller K. J.** Introduction to Wordnet: An Online Lexical Database. *Int. J. of Lexicography*, 1990, vol. 3, no. 4, pp. 235–244.

21. **Miller G. A., Gildea P. M.** How Children Learn Words. *J. Scientific American*, 1987, vol. 257, no. 3, pp. 94–99.

22. **Malyy akademicheskij slovar'**. (Dictionary of the Russian Language) Four volumes/ed. A. P. Evgenyeva, Moscow, Russkij yazyk; Poligrafresursy, 1999 (in Russian).

23. **Manning C., Raghavan P., Schutze H.** *Vvedenie v informacionnyj poisk* (Introduction to Information Retrieval), Moscow, Williams, 2001, 528 p. (in Russian).

24. **Landis J. R., Koch G. G.** Measurement of Observer Agreement for Categorical Data. *J. Biometrics*, 1977, vol. 33, no. 1, pp. 159–174.

25. **Altman D. G.** *Practical Statistics for Medical Research*. 1st ed. Chapman and Hall/CRC, 1990, 624 p.

ИНФОРМАЦИЯ

**Начинается подписка на журнал
"Программная инженерия" на первое полугодие 2016 г.**

Оформить подписку можно через подписные агентства
или непосредственно в редакции журнала.

Подписные индексы по каталогам:

Роспечать — 22765; Пресса России — 39795

Адрес редакции: 107076, Москва, Стромьинский пер., д. 4,
редакция журнала "Программная инженерия"

Тел.: (499) 269-53-97. Факс: (499) 269-55-10. E-mail: prin@novtex.ru

ООО "Издательство "Новые технологии". 107076, Москва, Стромьинский пер., 4
Технический редактор *Е. М. Патрушева*. Корректор *З. В. Наумова*

Сдано в набор 06.08.2015 г. Подписано в печать 15.09.2015 г. Формат 60×88 1/8. Заказ П11015
Цена свободная.

Оригинал-макет ООО "Авансед солюшнз". Отпечатано в ООО "Авансед солюшнз".
119071, г. Москва, Ленинский пр-т, д. 19, стр. 1. Сайт: www.aov.ru