

Программная инженерия



Пр **3**
ИН **2020**
Том 11



Рисунки к статье **В. А. Васенина, Е. Д. Явтушенко**
«СРЕДСТВА СОПРОВОЖДЕНИЯ ПРОЦЕССОВ РАЗГРАНИЧЕНИЯ ДОСТУПА
К БОЛЬШИМ НАУКОМЕТРИЧЕСКИМ ДАННЫМ С ИСПОЛЬЗОВАНИЕМ
МЕХАНИЗМОВ ВИЗУАЛЬНОГО ПРЕДСТАВЛЕНИЯ»

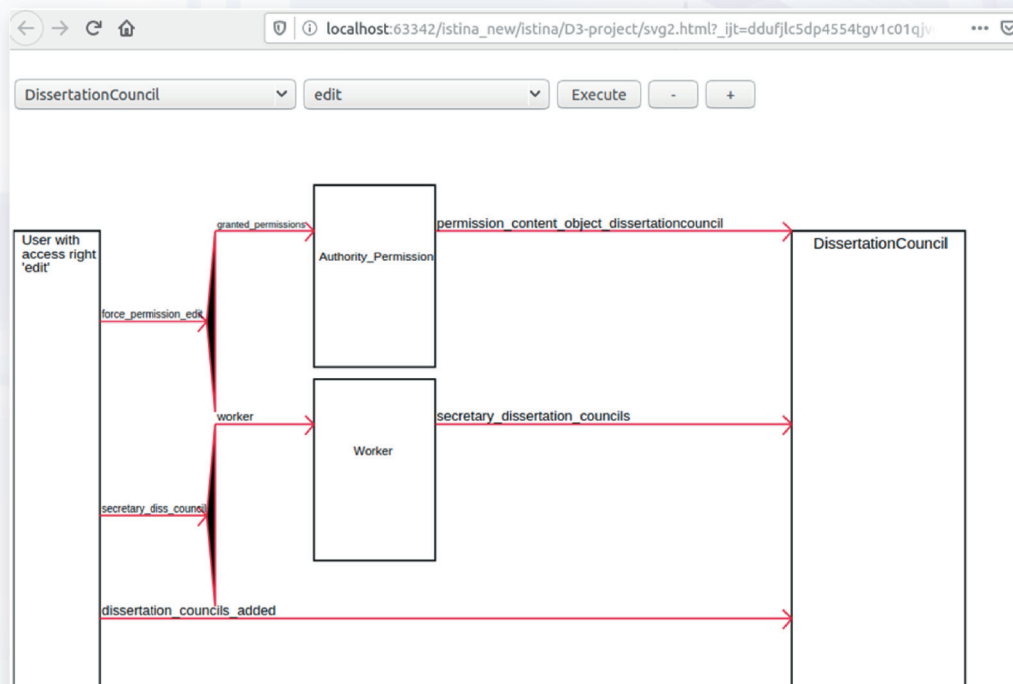


Рис. 2. Визуализация правил разграничения доступа к объекту «Диссертационный совет»

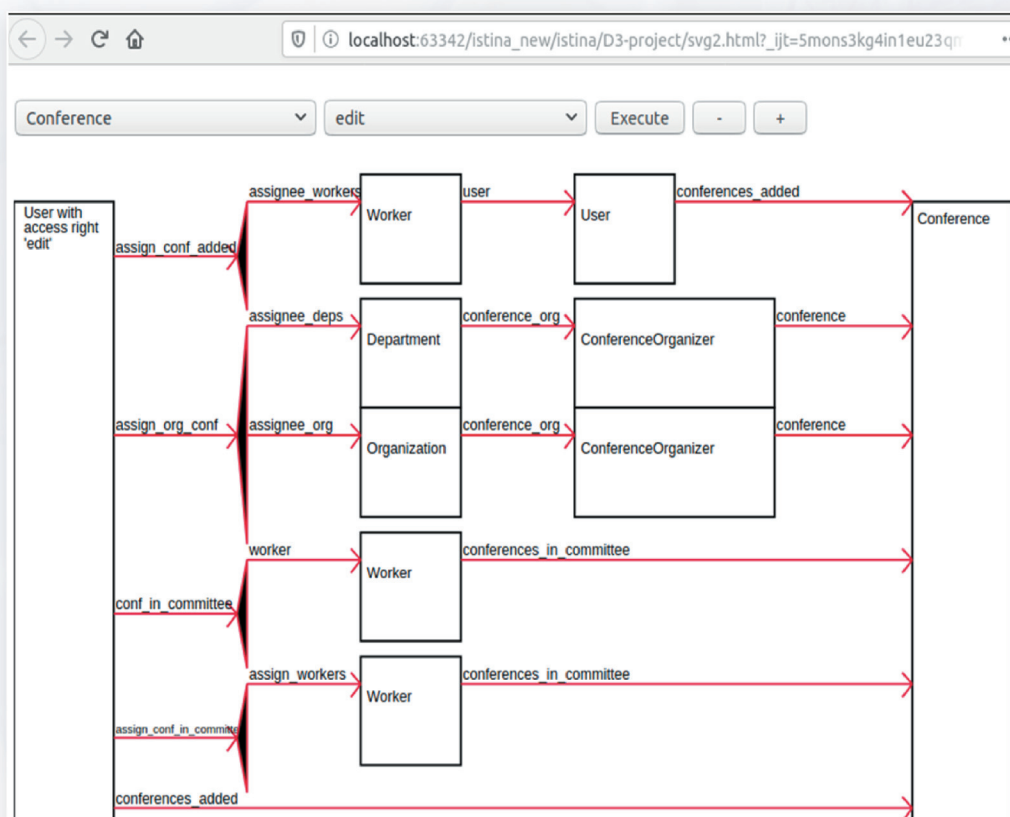


Рис. 3. Визуализация правил разграничения доступа к объекту «Конференция»

Программная инженерия

Том 11
№ 3
2020
Пр
ИН

Учредитель: Издательство "НОВЫЕ ТЕХНОЛОГИИ"

Издается с сентября 2010 г.

DOI 10.17587/issn.2220-3397

ISSN 2220-3397

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.
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

СОДЕРЖАНИЕ

- Васенин В. А., Явтушенко Е. Д.** Средства сопровождения процессов разграничения доступа к большим наукометрическим данным с использованием механизмов визуального представления 131
- Васильев В. И., Вульфин А. М., Черняховская Л. Р.** Анализ рисков инновационных проектов с использованием технологии многослойных нечетких когнитивных карт 142
- Бибило П. Н.** Минимизация *BDDI*-представлений систем неполностью определенных булевых функций 152
- Kalmychkov V. A., Matveeva I. V.** Automated Generation of Quantum Circuit Specifications Based on Reed—Muller Expressions 169
- Крюкова Е. С., Паращук И. Б.** Программное средство для моделирования смены состояний показателя качества электронной библиотеки: метод определения ключевых элементов модели 177
- Компаниец В. С., Казанская А. Ю., Лызь А. Е., Григорьев А. О.** Систематизация процедур проектирования пользовательского опыта в подготовке магистров прикладной информатики 183

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 (индекс по Объединенному каталогу "Пресса России" — 22765) или непосредственно в редакции.

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

Http://novtex.ru/prin/rus E-mail: prin@novtex.ru

Журнал включен в систему Российского индекса научного цитирования и базу данных RSCI на платформе Web of Science.

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

© Издательство "Новые технологии", "Программная инженерия", 2020

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)
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)
SEREBR'YAKOV 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

Vasenin V. A., Yavtushenko E. D. Tools for Supporting the Processes of Restricting Access to Large Scientometric Data Using Visual Presentation Mechanisms 131

Vasilyev V. I., Vulfin A. M., Chernyakhovskaya L. R. Risk Analysis of Innovative Projects with Use of Multilayer Fuzzy Cognitive Maps . . . 142

Bibilo P. N. Minimization of Binary Decision Diagrams for Systems of Incompletely Defined Boolean Functions Using Inverse Cofactors . . 152

Kalmychkov V. A., Matveeva I. V. Automated Generation of Quantum Circuit Specifications Based on Reed—Muller Expressions . . . 169

Kryukova E. S., Parashchuk I. B. Software Tool for Modeling the State Transition of a Quality Indicator of an Electronic Library: a Method for Determining the Key Elements of a Model 177

Kompaniets V. S., Kazanskaya A. Yu., Lyz A. E., Grigoriev A. O. Systematization of User Experience Design Procedures in the Applied Informatics Master's Training 183

В. А. Васенин, д-р физ.-мат. наук, проф., vasenin@msu.ru, **Е. Д. Явтушенко**, аспирант, evgeniya_40@mail.ru, Московский государственный университет им. М. В. Ломоносова

Средства сопровождения процессов разграничения доступа к большим наукометрическим данным с использованием механизмов визуального представления

Рассмотрены модели разграничения доступа субъектов ко всем объектам большой наукометрической системы и программные механизмы их визуализации, которые автоматизируют процессы сопровождения этой системы. Используются реляционная модель логического разграничения доступа и графовое представление правил разграничения. Сформированы требования как к самой системе, так и к модели разграничения. Модель визуализации, как составляющая более общей модели разграничения доступа к большим данным системы, позволяет упростить процесс управления таким доступом, устраняя узкие места, обусловленные языковыми средствами программной реализации модели разграничения доступа.

Ключевые слова: разграничение доступа, реляционная модель, визуализация, информационная безопасность, графовая модель

Введение

В современных условиях любая сложноорганизованная деятельность человека в различных сферах экономики связана с обработкой больших объемов данных и с интерактивным доступом к ним широкого круга лиц. Защита данных от несанкционированного доступа является одной из главных задач при проектировании, создании и сопровождении подобных систем. При этом к системам, осуществляющим доступ различных категорий пользователей к данным разной степени конфиденциальности, предъявляются высокие требования. В таких многопользовательских информационных системах ограничение доступа к тем или иным действиям с данными определяется на основе задаваемых правил. В условиях частого изменения связей между компонентами системы и атрибутами таких компонентов процессы адекватного и удобно воспринимаемого представления, контроля реализации правил доступа становятся самостоятельной, важной и сложной задачей. С учетом изложенного, разграничение доступа является одним из основных сервисов безопасности больших многопользовательских систем.

Программные механизмы логического разграничения доступа (далее используется термин "логическое разграничение доступа", ЛРД) к информационным активам, коммуникационным и вычислительным ресурсам являются важными элементами современных компьютерных систем, к защищенности которых предъявляют повышенные требования. С использованием механизмов ЛРД по заданному набору правил принимается решение о том, разрешено ли в системе некоторое действие, например, получение доступа к ее информационному ресурсу.

Алгоритм принятия решения о доступе функционирует согласно некоторой, как правило, математической модели, именуемой моделью ЛРД. Более точное определение такой модели будет дано в разд. 1.

Как правило, компоненты доступа (субъекты и объекты, определения которых будут также даны ниже) имеют большое число атрибутов, в зависимости от которых определяется, разрешен ли запрашиваемый доступ. Функционируют механизмы ЛРД на основе заданной для них конфигурации — набора *правил разграничения доступа*, представленных в форме входных данных из таких механизмов.

В статье в кратком изложении представлена формальная модель разграничения доступа, а также алгоритмы и программные механизмы, которые реализуют предъявляемые к ним требования в большой информационно-аналитической системе (ИАС) наукометрических данных. Такая система создается на основе принципов и методологии, уже апробированных на ИАС "ИСТИНА" — "Интеллектуальной Системе Тематического Исследования Наукометрических данных" [1]. Такая формальная модель именуется далее реляционной моделью разграничения доступа. Наукометрическая система позволяет обрабатывать данные о результатах научной деятельности. Исходная информация о результатах такой деятельности вносится в систему и корректируется преимущественно ее пользователями. Система "ИСТИНА" и аналогичные ей по назначению позволяют анализировать данные о результатах научной деятельности субъектов разных уровней по запросам многочисленных пользователей. Подобные системы имеют свойства, аналогичные свойствам социальных сетей: множество пользователей меняется без участия администратора системы; права доступа пользователей к объектам

определяются различными видами взаимосвязей между объектами, в том числе и опосредованными связями.

Особенностями наукометрических систем больших данных являются такие факторы, как большое число пользователей, зарегистрированных в системе, большое количество записей содержательной части системы, большое число часто изменяющихся связей между объектами системы.

В силу специфики данной системы существует ряд требований, отражающих функции и задачи, которые она и другие подобные ей наукометрические системы должны выполнять:

- автоматизированный ввод данных из файлов любой структуры, содержащих результаты научной и некоторых аспектов преподавательской деятельности сотрудников;
- быстрое вычисление типовых запросов пользователей на поиск определенных данных;
- корректный вывод данных;
- автоматизированная настройка и обновление данных для их поддержания в актуальном состоянии;
- возможность разграничивать доступ к данным и изменять настройки по мере изменения требования;
- возможность интеграции информации системы с другими хранилищами данных [1].

В контексте моделей и программных механизмов ЛРД в данной работе рассматривается важный компонент обеспечения информационной безопасности — визуализация процессов разграничения доступа. Визуализация информации — это процесс преобразования больших и сложных видов абстрактной информации в интуитивно понятную визуальную форму [2]. В общем случае этот компонент позволяет пользователю оперативно и в понятном для него виде получать необходимую информацию о возможных действиях нарушителя (под пользователем в этом случае подразумевается администратор безопасности). Подсистема визуализации особенно важна в рамках систем управления доступом, поскольку на основе отображаемых данных пользователь оценивает текущую политику безопасности, правила разграничения доступа и при необходимости выбирает и предпринимает соответствующие меры. Некорректная визуализация данных может привести к снижению качества выбираемых мер безопасности и, как следствие, к снижению уровня защищенности системы. В настоящее время в контексте визуализации данных большое внимание уделяется технологиям виртуальной и дополненной реальности [3—7]. Такие технологии широко применяются в различных областях человеческой деятельности — в проектировании и дизайне, добыче полезных ископаемых, военных технологиях и во многих других сферах. Виртуальная реальность — цифровой мир, полностью созданный с помощью современных компьютерных технологий, дополненная реальность — реальный мир, который "дополняется" виртуальными элементами и сенсорными данными. В системах виртуальной реальности основное внимание уделяется обеспечению зрительного канала, так как через него к человеку поступает основной объем информации.

Виртуальная среда создается посредством комплексного воздействия на восприятие ее человеком, использующим соответствующие шлемы виртуальной реальности или иные технические средства, которые динамически обновляют видимое пользователем пространство. Впервые подобного рода системы начали применять в различных военных тренажерах и для тренинга отдельных функций пилотов. В настоящее время такие технологии стали активно использовать и в других областях прикладной деятельности — от сферы развлечений, науки и медицины до оборонно-промышленного комплекса. К реализующим подобные технологии системам можно отнести шлемы и комнаты виртуальной реальности. Из первых реализованных на практике проектов, посвященных научной визуализации в виртуальном окружении, значимым является создание виртуальной аэродинамической трубы в исследовательском центре НАСА Эймс [7]. Целью этого проекта была визуализация трехмерных нестационарных потоков обтекания тел, моделирующих эксперименты в аэродинамической трубе. Дополненная реальность предлагает уникальные способы отображения визуальной информации, в частности, визуальное представление трехмерных объектов. В работе [8] в виде графов и плоских карт деревьев описаны модели визуализации маршрутов атак. Широкое распространение получают и технические системы дополненной реальности для экипажей боевых машин, танков и солдат, действующих в пешем порядке. Примером такого рода является американская нашлемная система ARC4.

При всех преимуществах перечисленных выше технологий для поддержки виртуальной и дополненной реальности необходимы значительные вычислительные ресурсы, которые может предоставить не любая система. В первую очередь, такие технологии используются в сложных киберфизических системах. Они обеспечивают ускорение и удешевление процессов обучения, тренировок и инструктажа для администраторов систем, увеличивают их эффективность; предотвращают угрозы здоровью и жизни сотрудников; сокращают число ошибок в процессе сборки, ремонта и эксплуатации специального оборудования; значительно снижают аварийность, а также стоимость эксплуатации единиц техники за счет своевременного выявления неисправностей. Для визуализации правил разграничения доступа в больших наукометрических системах использование технологий виртуальной и дополненной реальности в настоящее время не представляется целесообразным ввиду трудоемкости процессов внедрения и сопровождения этих технологий.

В общем случае процессу отображения информации должен предшествовать процесс ее обработки и приведения к структурированному виду, удобному для такого отображения. Универсальным средством представления структурированных данных являются графы. Графы применяют для представления любой информации, которую можно промоделировать в виде объектов и связей между ними. По этой причине визуализация графовых моделей является ключевым компонентом во многих приложениях в на-

уке и технике [9]. Разработка прикладных средств и методов визуализации, исследования в области визуализации графов и графовых моделей ведутся со второй половины 1980-х годов и имеют обширные и все более разнообразные сферы применения. Это связано с возникшей потребностью оперировать большими объемами информации и сложными структурами данных, которые естественным образом представляются графами. В настоящее время методы и средства визуализации графов и графовых моделей широко используются в таких областях, как информационно-коммуникационные системы и программное обеспечение, биологические науки, искусственный интеллект, анализ финансовой информации, компьютерное обучение [9–12].

Элементы (вершины и ребра) графа могут изображаться различными способами. Граф можно изобразить на плоскости или в трехмерном пространстве. Он может отображаться частично, несколько подграфов может быть стянуто в вершины, которые могут раскрываться по требованию. К основным графическим моделям, используемым при визуализации информации, можно отнести графы, различные графики и их вариации, матрицы, а также карты. При использовании классических моделей для наглядного представления данных применяют некоторые визуальные приемы, такие как расположение объекта в пространстве, кодирование информации с помощью его формы, размера и цвета. Формируемое при этом изображение должно быть понятным и информативным для человека.

Изображения графа могут быть не только статическими, но и интерактивными, поддерживающими различные способы навигации, адекватные потребностям пользователя. Если число элементов графа велико, его обработка может занимать неприемлемо большие ресурсы. Даже если возможно разместить и показать все элементы большого графа, зачастую возникают вопросы наглядности и удобства такой демонстрации. При графовом изображении не представляется возможным различать все элементы графа и их взаимосвязи. Интерактивная визуализация превращает статическую демонстрацию визуального представления информации в непрерывный процесс взаимодействия пользователя с информацией через ее визуальное отображение и доступные ему механизмы навигации. Пользователь может исследовать, рассматривать, открывать, узнавать и манипулировать данными через визуальные объекты. Есть ряд применений, требующих построения интерактивных изображений графов. Среди них инструментальные средства отладки, сохранение документов, модули отношений сущностей, веб-графы.

В работе, результаты которой представлены в статье, была поставлена задача разработать модель визуализации правил разграничения доступа субъектов ко всем объектам большой многопользовательской системы, в которой для хранения данных используется реляционная модель ЛРД на основе цепочек отношений. В качестве экспериментального полигона, на котором проводилась апробация программной реализации такой модели, рассматривалась ИАС

"ИСТИНА". Как упоминалось выше, эта система имеет характеристики социальных сетей. Социальные сети чаще всего представляются графовыми моделями, поэтому реляционная модель ЛРД имеет графовую структуру. Кроме того, принятый подход обосновывается тем, что связи между данными наукOMETрической системы имеют не менее большое значение, чем сами данные. В целом реляционная модель ЛРД представляется в виде совокупности правил, которые, в свою очередь, определяются цепочками отношений между объектами системы. Более подробное описание модели представлено в разд. 1.

При разработке модели визуализации процессов разграничения доступа необходимо выделить такие основные ее понятия, как субъект, объект, ресурс и отношение.

Субъект — это активный агент рассматриваемой системы, т. е. та сущность, которая выполняет действия, предполагающие осуществление доступа к некоторым ресурсам. В терминах визуализации процессов разграничения доступа к ресурсам системы субъектом всегда является пользователь, который хочет получить доступ к объекту. Средство визуализации показывает, с помощью каких цепочек отношений пользователь, обладающий заданным правом (имеющий возможность выполнять заданное действие), может получить доступ к определенному объекту.

Объекты — это информационные сущности, связанные между собой некоторыми отношениями. Объекты всегда принадлежат определенному классу.

Класс — это абстрактная информационная сущность, описывающая объект в общем виде, т. е. характеристики и методы, которыми может обладать объект.

Ресурсы системы — все объекты, не принадлежащие классу "пользователи".

Отношение — это определенная связь между двумя объектами (субъектами и ресурсами).

В случае информационно-аналитических наукOMETрических систем множество объектов может включать в себя учетные записи научных сотрудников, статьи, журналы, научные организации и подразделения. В множестве объектов выделено множество пользователей — активных сущностей-субъектов, совершающих операции над другими объектами. С каждым объектом ассоциирована метка, причисляющая его к определенному классу и определяющая, является ли объект пользователем или каким-либо ресурсом системы. Например, существуют классы "пользователь", "статья", "журнал", "подразделение" и др. Множество отношений — это множество разных видов взаимосвязей между объектами. Примерами отношений являются следующие взаимосвязи: авторство, связывающее пользователя с публикацией; место работы, связывающее пользователя с организацией или подразделением; связь пользователя с подразделением, для которого этот пользователь является ответственным за сопровождение информации в системе; связь публикации с подразделением, в котором работает один из ее авторов. Последнее отношение представляет собой пример порожденного отношения, которое автома-

тически возникает между двумя объектами, если существует третий объект, определенным образом связанный с ними обоими. Такими являются описанные выше отношения авторства и места работы. В больших данных используется множество цепочек отношений, каждая из которых представляет собой правило, по которому два объекта считают связанными определенным порожденным отношением, если существуют объекты, связанные с этими двумя объектами определенным образом. Более подробно виды отношений описаны разд. 1.

1. Реляционная модель ЛРД на основе цепочек отношений

С использованием механизмов ЛРД можно задать определенный набор правил, согласно которому пользователю системы разрешается или запрещается некоторое действие, например, получение доступа к информационному ресурсу системы. Для дальнейшего изложения необходимо определить несколько понятий, которые более подробно описаны в работе [13].

Определение 1. Функции логического разграничения доступа являются одними из функций, направленных на защиту информации. Функции ЛРД действуют над заданными в исследуемой автоматизированной системе множествами объектов, субъектов и типов доступа. Назначение функций ЛРД состоит в разрешении или запрете выполнения каждой операции (доступа или действия определенного типа, которое проводится субъектом доступа над объектом доступа) согласно заданным правилам разграничения доступа.

Определение 2. Программную реализацию функций ЛРД будем называть механизмом логического разграничения доступа, который реализует вычислительный алгоритм принятия решения о доступе. Механизм ЛРД встраивается в систему с целью реализовать назначение функций ЛРД, которые предполагают разрешение или запрет выполнения некоторого множества операций в системе. Входными данными для реализуемого алгоритма принятия решения о доступе является конфигурация механизма ЛРД и тройка \langle субъект доступа, объект доступа, тип доступа \rangle . В конфигурации механизма ЛРД некоторым зависящим от реализации способом задаются множества объектов и субъектов доступа, а также правила ЛРД.

Как правило, алгоритм принятия решения о доступе функционирует согласно некоторой математической модели, которая называется моделью ЛРД. Представим определение модели ЛРД.

Определение 3. Моделью логического разграничения доступа называется четверка $M = (S, O, A, P)$, где

- S — конечное и непустое множество субъектов доступа;
- O — конечное и непустое множество объектов доступа;
- A — конечное и непустое множество типов доступа;
- P — предикат над множеством $S \times O \times A$, определяющий для тройки $(s, o, a) \in S \times O \times A$, разрешен ли доступ типа a субъекта s к объекту o .

В случае рассмотрения доступа в качестве отдельной операции, принимаются следующие предположения:

- объекты доступа независимы, т. е. отдельный доступ к одному объекту не влияет на состояние других объектов;
- каждый отдельный доступ проводится только одним субъектом только к одному объекту.

Проблематике проектирования и разработки механизмов ЛРД, их интеграции в автоматизированные системы, информационные ресурсы которых подлежат защите, а также построению математических моделей, описывающих процессы функционирования таких механизмов, уделяется повышенное внимание с конца 1960-х — начала 1970-х годов. К настоящему времени созданы многочисленные механизмы ЛРД, традиционным примером которых являются механизмы ЛРД в операционных системах. Разработаны подходы к описанию математических моделей, составляющих основу функционирования подобных программных механизмов. К классическим формальным моделям ЛРД относятся модели дискреционного разграничения доступа, в том числе модель take-grant, модели мандатного многоуровневого разграничения доступа, включая фундаментальные модели Белла — ЛаПадула и Биба. Получили широкое распространение модели ЛРД на основе ролей, ключевые положения которых начали формироваться в 1990-е годы. Исследования моделей ЛРД и связанных с ними графов доступа проводятся во многих странах. Краткий анализ моделей ЛРД в социальных сетях описан в работе [13]. Среди отечественных ученых, работающих на этом направлении, можно выделить коллективы, возглавляемые П. Н. Девяниным [14–16], А. А. Грушо [19–21], П. Д. Зегждой [22, 23], а также коллектив, который представляют авторы [1, 13, 17, 18].

Зачастую для хранения свойств и связей между объектами в многопользовательской информационной системе используются реляционные системы управления базами данных (СУБД). В таких условиях модели ЛРД, использующие явным образом понятие отношения, которое легко транслируется в термины реляционных баз данных, имеют ряд преимуществ. Во-первых, языковые средства модели ЛРД могут быть расширены для использования других данных в реляционной СУБД в рамках единого языка с широкими выразительными возможностями. При этом можно ожидать, что такой язык будет оперировать схожим набором понятий вместе с другими программными компонентами целевой системы, которые связаны с той же базой данных. Во-вторых, механизм ЛРД может позволить получить доступ к дополнительным оптимизациям выполнения запросов к СУБД. Использование информации в реляционных СУБД в качестве основы для правил разграничения доступа поддерживается, например, многими типовыми наборами библиотек для веб-приложений, в которых такая информация может использоваться явным образом либо в рамках объектно-реляционного отображения.

Разграничение доступа к ресурсам системы, в представленной в настоящей статье реляционной

модели ЛРД, которая достаточно подробно описана в работе [13], реализуется на основе анализа цепочек отношений, определяющих такой доступ. Эта модель активно использует набор правил доступа для предотвращения случайных ошибок при редактировании ранее введенной информации, а также для ограничения доступа к определенным данным для их анализа на основе атрибутов пользователя и ресурса, к которому осуществляется доступ. В рамках настоящей статьи отметим следующие термины. Усеченным социальным графом будем называть нагруженный граф, вершинами которого являются пользователи системы, а ребрами — пары пользователей, связанных каким-либо отношением. Каждому ребру при этом соответствует набор отношений, связывающих пару пользователей. Полным социальным графом будем называть нагруженный граф, вершинами которого являются объекты (пользователи и ресурсы) системы, а ребрами — пары объектов, связанных каким-либо отношением. При этом каждому ребру соответствует набор отношений, связывающих пару объектов. В силу того, что в наукометрических системах важен учет отношений между ресурсами, а не только между пользователями, в дальнейшем под социальным графом будем подразумевать полный социальный граф, если явно не указано обратное.

Можно отметить две особенности спецификации правил контроля доступа, которые особенно важны для наукометрической системы, но не присутствуют в традиционных моделях ЛРД одновременно:

- использование полного социального графа с дополнительными отношениями "пользователь — объект" и "объект — объект" в совокупности с отношениями "пользователь — пользователь";
- использование атрибутов для сущностей и отношений в целях введения их параметров в правила доступа.

Отмеченные особенности означают, что объекты и их отношения могут иметь дополнительные атрибуты, влияющие на права доступа пользователей к объектам.

Таким образом, требования к модели управления доступом для большой наукометрической системы данных в кратком изложении включают в себя:

- определение разрешающих и запрещающих правил с использованием полного социального графа системы, атрибутов объектов и отношений между ними;
- ограничения на пользовательском уровне, соответствующие перечисленным выше правилам;
- учет отношений не только между пользователями, но и между ресурсами системы;
- возможность учитывать параметры внешней среды при задании правил разграничения доступа (например, текущее время, местоположение, IP-адрес и т. д.);
- широкие возможности настройки и изменения правил доступа;
- "легкость" восприятия основных постулатов и правил модели разграничения доступа конечным пользователем.

Модель должна допускать возможность разграничения доступа пользователей к объектам, учитывая

не только топологические свойства социального графа, но и дополнительные свойства объектов и отношений между ними. Рассмотрим пример. Пусть пользователь $u1$, назначенный ответственным за сопровождение данных в системе по некоторому подразделению $s1$, имеет право изменять атрибуты публикаций, выполненных в этом подразделении. Пусть сотрудник a , являющийся автором публикации p , работает в подразделении $s1$, однако на момент подготовки публикации p работал в подразделении $s2$, ответственным за сопровождение наукометрической информации по которому является пользователь $u2$. В этом случае, поскольку публикация подготовлена в подразделении $s2$, пользователь $u2$ должен иметь право изменять ее атрибуты, а пользователь $u1$ такого права иметь не должен. Вместе с тем пользователи $u1$ и $u2$ оба соединены с публикацией p цепочкой отношений "ответственный по подразделению" — "в подразделении работает сотрудник" — "сотрудник является автором статьи". Таким образом, модель, предполагающая разграничение доступа к объектам системы на основании только топологических свойств социального графа, не может дать пользователям $u1$ и $u2$ разные права доступа к публикации p . В связи с этим необходимо использовать модель, предполагающую учет дополнительных свойств вершин и ребер социального графа. В представленном примере ребрам, связывающим сотрудника с подразделением отношением "работает в", можно поставить в соответствие дату поступления на работу и увольнения, а ребрам, связывающим сотрудника с публикацией отношением авторства, — дату сдачи публикации в печать. При этом публикация должна считаться выполненной в определенном подразделении в том и только том случае, если один из ее авторов работал в этом подразделении на момент сдачи публикации в печать.

Реляционная модель ЛРД учитывает топологические свойства полного социального графа, включающего как пользователей системы, так и ее ресурсы, а также все отношения между ними. При этом в модели жестко фиксируется множество цепочек отношений, которые могут связывать субъекты и объекты доступа. Благодаря этому свойству реляционная модель ЛРД не требует от пользователей системы написания сложных правил для проверки существования пути в социальном графе. Кроме того, такая модель позволяет ограничивать использование вычислительных ресурсов для получения решения о запрашиваемом доступе, а также избегать отказов в обслуживании системы через исчерпание таких ресурсов.

Некоторые из отношений в системе можно отнести к *транзитивным* отношениям. Транзитивность означает, что если объект a связан с объектом b отношением r , а объект b связан отношением r с объектом c , то объект a считается связанным с объектом c отношением r , даже если это не указано в явном виде. Примером является отношение вложенности подразделений друг в друга. В частности, транзитивность отношений позволяет упростить запись правил предоставления системы по определенному подразделению, возможность изменять атрибуты всех сотрудников

этого подразделения и его дочерних подразделений всех уровней вложенности. Данное правило может быть сформулировано при условии, что отношение "одно подразделение является дочерним для другого" отмечено как транзитивное.

Другим видом отношений являются *порожденные* отношения. Данный вид взаимосвязи возникает между двумя объектами, если существует третий объект, определенным образом связанный с ними обоими. Это отношение определяет также тот факт, что два объекта системы s и f связаны порожденным отношением r , если существует такая цепочка отношений, порождающая отношение r , а также такая последовательность объектов, начинающаяся с s и заканчивающаяся f так, что каждый последующий объект в ней связан с предыдущим соответствующим отношением из цепочки, порождающей r [13]. Например, если цепочки отношений "в подразделении D работает сотрудник W ", "сотрудник W является автором публикации P " порождают отношение "публикация выполнена в подразделении D ", то публикация P связывается отношением "публикация выполнена в подразделении D ", если существует сотрудник W , работающий в подразделении D и являющийся автором публикации P .

Цепочка отношений реляционной модели ЛРД представляет собой правило, по которому два объекта считаются связанными определенным порожденным отношением, если существуют объекты, связанные с этими двумя объектами определенным образом. Цепочка, имеющая вид $(r12, r23) = > r13$ означает, что объекты считаются связанными отношением $r13$, если существует такой объект $o2$, что объекты $o1$ и $o2$ связаны отношением $r12$, а объекты $o2$ и $o3$ связаны отношением $r23$. Последовательность $(o1, o2, o3)$ при этом называется цепочкой объектов, соответствующей цепочке отношений $(r12, r23) = > r13$.

Другие отношения, для которых информация о связанных этими отношениями объектах хранится в базе данных системы и не зависит от других отношений, называются *примитивными*.

С каждым из отношений в системе при использовании реляционной модели ассоциированы множества разрешенных и запрещенных действий. Если действие a является разрешенным для отношения R , то пользователь, связанный с целевым объектом отношением R , имеет право осуществлять доступ a к целевому объекту. Если действие a является запрещенным для отношения R , то пользователь, связанный с целевым объектом отношением R , не имеет право осуществлять доступ a к целевому объекту даже в том случае, если существует другое отношение, связывающее данного пользователя с данным объектом и разрешающее действие a .

Пример использования реляционной модели ЛРД из работы [13] представлен на рис. 1. Пусть пользователь a является ответственным за сопровождение информации в системе в рамках подразделения b , включающего в себя дочернее подразделение c , в котором работает сотрудник d , являющийся автором статьи e . Тогда пользователь a имеет право изменять атрибуты статьи e за счет существования последовательности объектов (a, b, c, d, e) , связывающей пользователя с целевой статьей следующей после-

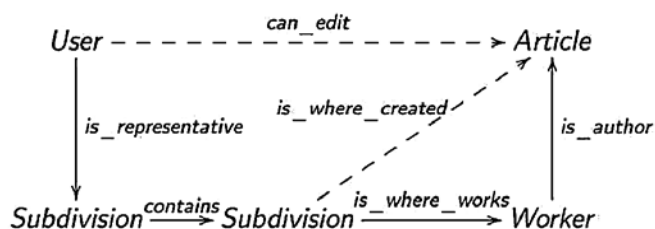


Рис. 1. Пример цепочки отношений

довательностью отношений: "пользователь является ответственным за сопровождение информации в рамках подразделения" — "одно подразделение содержит другое" — "в подразделении работает сотрудник" — "сотрудник является автором статьи". Таким образом, модель, предполагающая разграничение доступа к объектам системы на основании только топологических свойств социального графа, не может дать пользователям $u1$ и $u2$ разные права доступа к публикации p . В связи с этим обстоятельством необходимо использовать модель, предполагающую учет дополнительных свойств вершин и ребер социального графа. В представленном примере ребрам, связывающим сотрудника с подразделением отношением "работает в", можно поставить в соответствие дату поступления на работу и увольнения, а ребрам, связывающим сотрудника с публикацией отношением авторства, — дату сдачи публикации в печать. При этом публикация должна считаться выполненной в определенном подразделении в том и только том случае, если один из ее авторов работал в этом подразделении на момент сдачи публикации в печать.

В настоящее время реляционная модель содержит более 200 нетривиальных правил и представляется в виде xml-файла. Проблемным вопросом ее сопровождения является недостаточная степень адаптации модели ЛРД к своевременным механизмам обработки больших объемов информации и событий безопасности, которая необходима для оценки текущего состояния системы, управления инцидентами и выбора контрмер. Представление модели в виде xml-файла, с одной стороны, хорошо коррелирует с графовой основой xml-языка. С другой стороны, такое представление является сложным для восприятия администратором, которому необходимо получить информацию о разграничении доступа. Для представления модели ЛРД в удобном для пользователя виде необходимо конвертировать ее в более наглядный формат. Отметим, что под пользователем в этом случае подразумевается администратор безопасности. В настоящей работе предложено средство, использующее графовое представление данных для визуализации правил реляционной модели разграничения доступа.

2. Модель визуализации правил разграничения доступа

Главная задача, решаемая с помощью модели визуализации в рамках исследования, результаты которого представлены в настоящей работе, заключается в том, чтобы наглядно для каждого объекта

представить описанные выше цепочки отношений, определяющие доступ к нему. Посредством данных цепочек отношений пользователь (субъект), обладающий заданным правом, может получить доступ на выполнение определенного действия с запрашиваемым объектом. Как отмечалось ранее, реляционная модель ЛРД не удовлетворяет требованию удобства восприятия администратором определяемых ею правил разграничения доступа. Как следствие, ее программная реализация не предоставляет средств по адаптивной настройке и своевременному обнаружению ошибок доступа. Эти факторы существенно ограничивают возможности использования реляционной модели ЛРД в ходе сопровождения ее программной реализации:

- с точки зрения отсутствия механизмов оперативного контроля существующих отношений доступа;
- в плане реализации гибкого подхода к изменению правил доступа;
- с точки зрения отсутствия доступа к управлению системой внешних пользователей (включая администраторов безопасности), которые имеют для этого необходимые права.

Представленная в настоящей работе модель визуализации правил доступа к ресурсам большой наукометрической системы обеспечивает понятное и доступное даже для неподготовленных специальным образом пользователей графическое представление реляционной модели ЛРД. Данные об объектах предоставляются по запросу пользователя, как правило, администратора безопасности. Пользователь выбирает объект доступа, информацию о котором необходимо получить, и извлекает сведения о возможных действиях над этим объектом. Средство визуализации обеспечивает вывод изображения, которое наглядно демонстрирует цепочки отношений, связывающие пользователя, обладающего заданным правом доступа, с заданным объектом. Внесение изменений в политику безопасности не повлечет за собой значительных вычислительных или временных затрат. Разработанное средство визуализации позволяет создать дружественный интерфейс. Модель представляется в виде графической схемы или графа с указанием всех отношений между компонентами (субъектами и объектами доступа) в русскоязычном описании.

Описываемая модель визуализации имеет следующие особенности:

- модель представляет собой ориентированный граф, направление которого задается от субъекта к объекту;
- вершинами графа являются объекты и субъекты системы;
- ребрам графа соответствуют отношения, которыми связаны два объекта системы, обозначенные вершинами данного ребра;
- в графе могут присутствовать рекурсивные цепочки отношений, т. е. цепочки, в которых повторяются объекты системы.

Программное средство визуализации реляционной модели ЛРД является удобным и простым для применения. Интерфейс средства визуализации реализован в форме веб-приложения и предоставляет

возможности масштабирования данных из браузера, ввода необходимых данных и загрузки изображения.

При использовании интерфейса средства визуализации в панелях с раскрывающимся списком меню выбирается объект доступа и метка доступа. Метка доступа — фиксированное полномочие субъекта по отношению к объекту. Метка — объект, содержащий информацию о типе разрешения, а именно — имя, которое в дальнейшем используется для определения наличия разрешения на выполнение операций (редактирование, удаление, создание и др). В качестве примера рассмотрим рис. 2 (см. вторую сторону обложки), иллюстрирующий визуализацию доступа к объекту "Диссертационный совет" (прикладная подсистема в составе ИАС "ИСТИНА") для пользователя с правом на редактирование информации в этой подсистеме. В меню пользователем выбирается объект "Диссертационный совет" (DissertationCouncil), метка доступа "edit", затем выполняется нажатие на кнопку "Execute". После этого формируется графическая схема с указанием субъекта и объекта доступа и отношений между ними. Субъект доступа (пользователь) и целевой объект доступа изображены в виде именованных прямоугольников по краям окна браузера, между которыми определенным образом строятся цепочки отношений. В данном примере крайним слева будет построен прямоугольник, обозначающий пользователя с заданной меткой, содержащий текст "User with access right "edit"", крайним справа — прямоугольник, обозначающий объект доступа, содержащий имя "DissertationCouncil".

Цепочки отношений, которые связывают пользователя и объект доступа, визуализируются на основе данных, сформированных в JSON-файле, в связи с положениями политики безопасности системы. Основной файл, отображающий такую политику, содержит данные о перечне запрещенных и разрешенных действий пользователя с объектами системы, информацию о псевдонимах цепочек и входящих в эти цепочки отношениях. Таким образом, если пользователь имеет доступ к объекту, связанному с этим пользователем порожденным отношением, то указывается цепочка, порождающая это отношение, а также виды доступа, разрешенные пользователю по отношению к целевому объекту. Кроме того, одновременно с отмеченными выше данными о содержании рассматриваемого файла может быть указано имя или псевдоним порожденного отношения. Это имя или псевдоним позволяет в дальнейшем идентифицировать вновь созданное отношение, включая его в качестве элемента в другие цепочки.

Средство визуализации генерирует изображения правил разграничения доступа с помощью трех основных видов фигур — прямоугольников, стрелок, треугольников. Субъект и объект доступа изображаются в виде двух больших прямоугольников, стоящих по краям экрана. Маленькие прямоугольники определяют промежуточные объекты системы, через которые осуществляется доступ к целевому объекту. Отношения отображаются стрелками, направление которых задается от субъекта к объекту доступа. Треугольник определяет факт того, что опи-

сываемое отношение не является примитивным и не идентифицируется. Имена списков цепочек отношений прописываются над первой (в случае примитивных отношений — единственной) стрелкой, исходящей от прямоугольника, имитирующего пользователя-субъекта. Если отношение примитивное, то стрелка с именем отношения идет напрямую от крайнего прямоугольника-субъекта, соответствующего пользователю с меткой доступа, к прямоугольнику — объекту доступа. Иначе, именованная первая стрелка соединяет крайний прямоугольник-субъект с треугольником. Далее, если список цепочек отношений состоит из одной цепочки, то из треугольника исходит стрелка, направленная к маленькому прямоугольнику, определяющему промежуточный объект, на который указывает первое отношение. Таким образом, визуализируется первое отношение из цепочки. Затем из маленького промежуточного прямоугольника-объекта строится стрелка к следующему объекту, который изображается именованным прямоугольником. Эта стрелка определяет второе отношение цепочки и т. д. Последнее отношение цепочки связывает промежуточный прямоугольник-объект и целевой объект доступа. Таким образом визуализируется так называемый "путь доступа" от пользователя к заданному объекту системы. Список отношений может состоять из нескольких цепочек. В этом случае из треугольника (к которому проходит стрелка от прямоугольника-субъекта) исходит несколько стрелок, каждая из которых аналогичным образом определяет соответствующую цепочку. Важно отметить, что наличие нескольких цепочек отношений определяет разные способы получения доступа субъекта к объекту, и все цепочки одного списка несут одинаковый смысл, так как в конечном итоге приводят к одному и тому же результату. Поэтому псевдоним является общим для списка цепочек отношений.

Самый простой вид имеют примитивные отношения. Информация о связанных этими отношениями объектах хранится в базе данных системы и не зависит от других отношений. В программном коде политики безопасности системы такие отношения не инициализируются, они являются непосредственно полями используемого средства разработки и сопровождения системы, которым является Django. Однако при генерации JSON-файла им присваивается псевдоним, идентичный имени поля (листинг 1).

```
"DissertationCouncil": [
{
  "alias": "dissertation_councils_added",
  "nodes": [ [ "dissertation_councils_added" ] ],
  "actions": [ "edit" ]
}
]
```

Листинг 1. Пример примитивного отношения в файле JSON

На листинге 1 представлен фрагмент кода, сгенерированного на основе данных политики разграничения доступа реляционной модели, который демонстрирует хранение информации о примитив-

ном отношении с псевдонимом *"dissertation_councils_added"* объекта доступа "Диссертационный совет" (*DissertationCouncil*). Набор цепочек отношений состоит из одной цепочки, которая содержит только одно отношение, являющееся полем модели Django. Это означает, что в Django-модели *DissertationCouncil*, описывающей процессы, сопровождающие прохождение диссертаций на соискание ученой степени, есть поле *"creator"*, связанное с моделью *User*, и также определен связанный атрибут *"dissertation_councils_added"* для указания имени обратного отношения от модели *User* к модели *DissertationCouncil* (листинг 2).

```
class DissertationCouncil(LinkedToWorkersModel):
id=models.AutoField(primary_key=True, db_column="F_BOARD_ID")
members=models.ManyToManyField("workers.Worker",
related_name="dissertation_councils", through="DissertationCouncilMembership", verbose_name="Члены совета")
creator=models.ForeignKey(to=User, related_name="dissertation_councils_added", db_column="F_BOARD_USER",
null=True, blank=True, verbose_name="Создатель")
```

Листинг 2. Пример реализации Django-модели для объекта "Диссертационный совет"

Таким образом, пользователь имеет право редактировать информацию в подсистеме "Диссертационный совет" при наличии отношения *"dissertation_councils_added"*, которое имеется у создателя совета.

Средство визуализации представляет данное правило разграничения доступа с помощью единичной стрелки с именем *"dissertation_councils_added"*, ведущей от прямоугольника-пользователя к прямоугольнику, обозначающему подсистему "Диссертационный совет" (рис. 2, см. вторую сторону обложки).

Порожденные отношения зависят от других отношений и инициализируются в коде политики безопасности системы, т. е. им присваивается псевдоним (листинг 3).

```
"Conference": [
{
  "alias": "assign_org_conf",
  "nodes": [
[
"assignee_deps",
"conference_org",
"conference"
],
[
"assignee_org",
"conference_org",
"conference"
]
],
"actions": ["edit"]
}
```

Листинг 3. Пример порожденного отношения в файле JSON на примере объекта "Конференция"

Следует заметить, что отношения, входящие в цепочку, порождающую отношение, также могут

быть порожденными. Например, первое отношение *"assignee_deps"* само является порожденным. Это означает, что существует список цепочек отношений с псевдонимом *"assignee_deps"*. Он может относиться к данному объекту или какому-либо другому. В данном примере список цепочек отношений указывает на объект "Подразделение" (*Department*) (листинг 4).

```

    "Department": [
  {
    "alias": "assignee_deps",
    "nodes": [
      [
        "current_representative_departments"
      ],
      [
        "assignee_org",
        "departments"
      ],
      [
        "current_representative_departments",
        "actual_child_departments"
      ]
    ]
  },
  {
    "actions": [
      "human_resources"
    ]
  }
]
}

```

Листинг 4. Пример порожденного отношения в файле JSON на примере объекта "Подразделение"

Отношение *"assignee_deps"* порождается тремя способами с помощью трех различных цепочек отношений, элементы которых также являются порожденными. В настоящее время средство визуализации не предоставляет механизмов для полного вывода родительских отношений, порождающих подобного рода отношения, так как изображение может стать сложным для понимания в силу большого числа объектов и длины цепочки. При визуализации правил доступа к объекту (приложению) "Конференция" и изображении списка цепочек отношений с псевдонимом *"assign_org_conf"* средство визуализации проверяет, является ли первое отношение первой цепочки *"assignee_deps"* примитивным. Если оно является примитивным, то изображение строится по принципам, описанным выше. В противном случае средство осуществляет поиск по содержимому JSON-файла в целях определения объекта, на который указывает данное отношение (в данном случае это объект "Подразделение"). Найденное имя объекта используется для построения промежуточного прямоугольника (рис. 3, см. вторую сторону обложки).

На рис. 3 представлено пять списков цепочек отношений доступа пользователя к объектам "Конференция" (*Conference*), обозначенных псевдонимами. При этом список цепочек отношений с псевдонимом *"assign_org_conf"* состоит из двух цепочек отношений, равноправных между собой. Первая цепочка отношений указывает на промежуточные объекты "Подразделение" (*Department*) и "Организатор конференции" (*ConferenceOrganizer*), вторая — "Организация" (*Organization*) и "Организатор конференции" (*ConferenceOrganizer*).

Цепочки изображаются параллельно и находятся в пределах одного треугольника, определяющего псевдоним набора цепочек отношений. Таким образом, пользователь может получить доступ на редактирование объекта "Конференция" шестью различными способами; можно сделать заключение о том, что шесть различных категорий пользователей (с различными "путями доступа") могут редактировать объект "Конференция".

3. Реализация и тестирование механизмов визуализации правил разграничения доступа

При реализации программного средства визуализации правил доступа к ресурсам информационно-аналитической, наукометрической системы разработан генератор JSON-файла, в котором сосредоточена основная информация о разграничении доступа для всех объектов системы с указанием цепочек отношений и перечня запрещенных и разрешенных действий. Необходимыми данными для генератора являются объект "политика разграничения доступа", библиотеки реляционной модели разграничения доступа и метаданные модели данных системы. При внесении изменений в правила разграничения доступа файл необходимо генерировать заново. Реализована функция *toJson*, которая преобразует политику безопасности в объект словаря для записи в JSON-файл. На основе этого файла средствами формата векторных изображений SVG строится графовая модель визуализации правил доступа. Последовательно применяется функция, которая проверяет наличие порожденных отношений или примитивных связей, вызывает функцию построения соответствующей геометрической фигуры. Для построения цепочек осуществляется поиск соответствующих псевдонимов отношений по JSON-словарю. Таким образом, входными данными для средства визуализации являются описание политики разграничения доступа в формате JSON; тип объекта и метка доступа, относительно которых требуется получить изображение политики разграничения доступа. Для этих данных средство генерирует визуальное графическое представление цепочек отношений, которые используются для разграничения доступа с заданной меткой к данным заданного типа в рамках заданной политики. Средство визуализации предоставляет интерфейс с возможностью масштабирования данных из браузера, ввода необходимых данных и загрузки изображения.

Тестирование программной реализации механизмов визуализации правил, принятых реляционной моделью разграничения доступа, в условиях реального режима эксплуатации (сопровождения) ИАС "ИСТИНА" (коллекция данных более чем 95 тыс. пользователей) проводилось в целях подготовки оперативных ответов на обращения пользователей о наблюдаемых ими инцидентах.

В среднем в подсистему обращений поступало 5...10 таких запросов в течение суток. Некоторые из инцидентов были искусственно созданы для проверки сервисов всех подсистем без предварительного уведомления об этом пользователей (включая адми-

нистраторов безопасности). Результаты тестирования позволяют сделать следующие выводы.

1. Использование механизмов визуализации позволяет с достаточной степенью оперативности и аргументированности для пользователя ответить на поступающие от него обращения по поводу обнаруженных инцидентов нарушения принятой модели разграничения доступа к ресурсам системы.

2. Оперативный характер анализа и надежность представления результатов, полученных по обращениям, удовлетворяют и их авторов, и администраторов безопасности.

Заключение

Представленные механизмы визуального отображения процессов разграничения доступа к большим данным наукометрического содержания дополняют модель разграничения доступа к ресурсам сложно организованных интерактивных систем с большим объемом быстро меняющихся данных, которые хранятся в реляционных базах. Такое дополнение устраняет очень важный недостаток реляционной модели разграничения, обусловленный языком описания, который не позволяет должным образом использовать ее программную реализацию для эффективного сопровождения системы в процессе ее эксплуатации.

Результаты тестирования первой версии подсистемы визуализации правил разграничения доступа применительно к организации оперативной реакции на запросы пользователей большой наукометрической системы об инцидентах, связанных с нарушением регламентированного доступа к ее ресурсам, позволяют сделать следующие выводы.

1. Использование этих механизмов для оперативной реакции на сообщения об инциденте нерегламентированного доступа к ресурсам системы целесообразно. Они обеспечивают необходимые для решения задачи в такой постановке уровень оперативности анализа и степень наглядности для участвующих в таком анализе сторон.

2. Предлагаемые программные механизмы визуализации правил ЛРД имеют хорошие перспективы развития и решения других задач, востребованных на направлении сопровождения систем больших данных со сложной структурой, в том числе в иных, отличных от наукометрии областях.

В перспективе развития модели визуализации предполагается обеспечить возможность оперативной корректировки и модификации данных разграничения доступа по мере их изменения применительно к конкретным объектам доступа. Например, с помощью разработанной панели инструментальных средств администратору должна быть предоставлена возможность создать новые, уничтожить ранее созданные правила доступа, уточнить атрибуты существующих объектов.

Список литературы

1. **Интеллектуальная** система тематического исследования научно-технической информации (ИСТИНА) / Под ред. В. А. Садовниченко. — М.: Издательство Московского университета, 2014. — 262 с.

2. **Касьянов В. Н., Касьянова Е. В.** Визуализация графов и графовых моделей. — Новосибирск: Сибирское Научное Издательство, 2010. — 123 с.

3. **Иванова А. В.** Технологии виртуальной и дополненной реальности: возможности и препятствия применения. URL: <https://cyberleninka.ru/article/n/tehnologii-virtualnoy-i-dopolnennoy-realnosti-vozmozhnosti-i-prepyatstviya-primeneniya/viewer> (дата обращения 20.10.2019).

4. **Благовещенский И. А., Демьянков Н. А.** Технологии и алгоритмы для создания дополненной реальности // Моделирование и анализ информационных систем. — 2013. — Том 20, № 2. — С. 129—138.

5. **Дополненная** и виртуальная реальность. URL: <http://arnext.ru/news/2018-ar-vg-6503> (дата обращения 27.09.2019).

6. **Кравцов А. А.** Исследование и разработка информационной системы с технологией интерактивной визуализации средствами дополненной реальности: дис. ... канд. техн. наук. Кубанский государственный аграрный университет. Краснодар, 2016. 167 с. URL: <https://kubstu.ru/data/fdlist/FDD0463.pdf> (дата обращения 20.07.2019).

7. **Никитина Л. Д., Никитин И. Н., Клименко С. В.** Математическая визуализация в виртуальном окружении. [Электронный ресурс]. URL: <https://cyberleninka.ru/article/n/matematiceskaya-vizualizatsiya-v-virtualnom-okruzhenii/viewer> (дата обращения 10.07.2019).

8. **Котенко И. В., Новикова Е. С.** Визуальный анализ для оценки защищенности компьютерных сетей // Информационно-управляющие системы. — 2013. — № 3. — С. 55—61.

9. **Касьянов В. Н., Евстигнеев В. А.** Графы в программировании: обработка, визуализация и применение. — СПб.: БХВ-Петербург, 2003. — 1104 с.

10. **Касьянов В. Н.** Визуализация информации на основе графовых моделей. URL: <http://conf.ict.nsc.ru/files/conferences/MIT-2013/fulltext/143766/150440/Kasyanov.pdf> (дата обращения 10.07.2019).

11. **Herman I., Melancon G., Marshall M. S.** Graph visualization and navigation in information visualization: a survey // IEEE Trans. on Visualization and Computer Graphics. — 2000. — Vol. 6, No. 1. — P. 24—43.

12. **Di Battista G., Eades P., Tamassia R., Tollis I. G.** Graph Drawing: Algorithms for Visualization of Graphs. Prentice Hall, 1999. — 397 p.

13. **Васенин В. А., Иткес А. А., Шапченко К. А., Бухонов В. Ю.** Реляционная модель логического разграничения доступа на основе запечек отношений // Программная инженерия. — 2015. — № 9. — С. 11—19.

14. **Девянин П. Н.** Подходы к моделированию управления доступом в СУБД PostgreSQL в рамках МРОСЛ ДП-модели // Прикладная дискретная математика. — 2018. Приложение № 11. — С. 95—99.

15. **Девянин П. Н., Ефремов Д. В., Кулямин В. В.** и др. Моделирование и верификация политик безопасности управления доступом в операционных системах. — М.: Горячая линия — Телеком, 2019. — 214 с.

16. **Девянин П. Н.** Модели безопасности компьютерных систем. Управление доступом и информационными потоками. Уч. пособие для вузов. 2-е изд., испр. и доп. М.: Горячая линия — Телеком, 2013. — 338 с.

17. **Васенин В. А., Иткес А. А., Бухонов В. Ю., Галатенко А. В.** Модели логического разграничения доступа в многопользовательских системах управления наукометрическим контентом // Программная инженерия. — 2016. — Том 7, № 12. — С. 196—207.

18. **Васенин В. А., Иткес А. А., Шапченко К. А.** О применении моделей разграничения доступа в социальных сетях к одному классу много-пользовательских систем управления контентом // Программная инженерия. — 2015. — № 4. — С. 10—19.

19. **Грушо А. А., Тимонина Е. Е.** Теоретические основы защиты информации. Уч. пособие. — М.: Изд-во Агентства "Яхтсмен", 1996. — 176 с.

20. **Грушо А. А., Применко Э. А., Тимонина Е. Е.** Теоретические основы компьютерной безопасности. — М.: Academia, 2009. — 272 с.

21. **Грушо А. А., Тимонина Е. Е.** Модель случайных графов для описания взаимодействий в сети // Информатика и ее применение. — 2012. — Т. 6, № 4. — С. 61—64.

22. **Зегжда П. Д., Рудина Е. А.** Основы информационной безопасности. Уч. пособие. — СПб.: Изд-во Политехнического ун-та, 2008. — 224 с.

23. **Калинин М. О.** Адаптивное управление безопасностью информационных систем на основе логического моделирования: дис. ... д-ра техн. наук. Санкт-Петербург, 2011. 290 с.

Tools for Supporting the Processes of Restricting Access to Large Scientometric Data Using Visual Presentation Mechanisms

V. A. Vasenin, vasenin@msu.ru, E. D. Yavtushenko, evgeniya_40@mail.ru, Lomonosov Moscow State University, Moscow, 119991, Russian Federation

Corresponding author:

Vasenin Valery A., Professor, Lomonosov Moscow State University, Moscow, 119991, Russian Federation, E-mail: vasenin@msu.ru

Received on February 28, 2020

Accepted on March 31, 2020

In modern conditions, protection data from unauthorized access is one of the main tasks for the design, creation and support of the systems associated with big data processing. In such multi-user information systems, the defined access control rules determine the restriction of access to certain actions with system data. The process of adequately and conveniently presentation of these access rules is an important and difficult task. This article presents a mechanism for visualizing the processes of access control to multi-user system data using system "ISTINA" example.

The first section describes the relational model of logical access control based on relations chains, which is used in a big scientometric system. Requirements are formed, both to the system itself and to the access control model.

The second section of the article presents the visualization model, which is a component of the general access control model to big data of the system. The developed visualization subsystem provides data for administrator in an accessible and understandable form. On the basis of these data the administrator evaluates the current security policy, analyzes the access control rules. The third section briefly describes the software implementation and testing of the visualization subsystem.

The mechanisms for visualizing the processes of big data access control in the scientometric content system presented in this article are an important addition to the relational access control model. These mechanisms provide effective system support during its exploitation.

Keywords: access control, relational model, visualization, information security, graph model

For citation:

Vasenin V. A., Yavtushenko E. D. Tools for Supporting the Processes of Restricting Access to Large Scientometric Data Using Visual Presentation Mechanisms, *Programmnaya Ingeneria*, 2020, vol. 11, no. 3, pp. 131–141

DOI: 10.17587/prin.11.131-141

References

1. *Intellektual'naya sistema tematicheskogo issledovaniya nauchno-tekhnicheskoy informacii (ISTINA)*/ Eds. V. A. Sadovnichij, Moscow, Izdatel'stvo Moskovskogo universiteta, 2014, 262 p. (in Russian).
2. Kas'yanov V. N., Kas'yanova E. V. *Vizualizatsiya grafov i grafovyykh modelej*, Novosibirsk, Sibirskoe Nauchnoe Izdatel'stvo, 2010, 123 p. (in Russian).
3. Ivanova A. V. Tekhnologii virtual'noj i dopolnennoj real'nosti: vozmozhnosti i prepyatstviya primeneniya, available at: <https://cyberleninka.ru/article/n/tehnologii-virtual'noy-i-dopolnennoj-real'nosti-vozmozhnosti-i-prepyatstviya-primeneniya/viewer> (in Russian).
4. Blagoveshenskij I. A., Dem'yankov N. A. Tekhnologii i algoritmy' dlya sozdaniya dopolnennoj real'nosti, *Modelirovanie i analiz informacionnykh system*, 2013, vol. 20, no. 2, pp. 129–138 (in Russian).
5. *Dopolnennaya i virtual'naya real'nost'*, available at: <http://arnext.ru/news/2018-ar-vr-6503> (in Russian).
6. Kravcov A. A. Issledovanie i razrabotka informacionnoj sistemy' s tekhnologiej interaktivnoj vizualizacii sredstvami dopolnennoj real'nosti, available at: <https://kubstu.ru/data/fdlist/FDD0463.pdf> (in Russian).
7. Nikitina L. D., Nikitin I. N., Klimenko S. V. Matematicheskaya vizualizatsiya v virtual'nom okruzhenii, available at: <https://cyberleninka.ru/article/n/matematicheskaya-vizualizatsiya-v-virtual'nom-okruzhenii/viewer> (in Russian).
8. Kotenko I. V., Novikova E. C. Technologies and Algorithms for Building the Augmented Reality, *Informacionno-upravlyayushhie sistemy*, 2013, no. 3, pp. 55–61 (in Russian).
9. Kas'yanov V. N., Evstigneev V. A. *Grafy v programirovani: obrabotka, vizualizatsiya i primenenie*, Saint-Peterburg, BXV-Peterburg, 2003, 1104 p. (in Russian).
10. Kas'yanov V. N. Vizualizatsiya informacii na osnove grafovyykh modelej, available at: <http://conf.ict.nsc.ru/files/conferences/MIT-2013/fulltext/143766/150440/Kasyanov.pdf> (in Russian).
11. Herman I., Melancon G., Marshall M. S. Graph visualization and navigation in information visualization: a survey, *IEEE Trans. on Visualization and Computer Graphics*, 2000, vol. 6, no. 1, pp. 24–43.
12. Di Battista G., Eades P., Tamassia R., Tollis I. G. *Graph Drawing: Algorithms for Visualization of Graphs*, Prentice Hall, 1999, 397 p.
13. Vasenin V. A., Itkes A. A., Shapchenko K. A., Buxonov V. Yu. Relational Access Control Model Based on Chains of Relations, *Programmnaya ingeneria*, 2015, no. 9, pp. 11–19 (in Russian).
14. Devyanin P. N. Approaches to formal modelling access control in postgresql within framework of the mrosi DP-model, *Prikladnaya diskretnaya matematika*, 2018, Prilozhenie no. 11, pp. 95–99 (in Russian).
15. Devyanin P. N., Efremov D. V., Kulyamin V. V., Petrenko A. K., Horoshilov A. V., Shhepetkov I. V. *Modelirovanie i verifikatsiya politik bezopasnosti upravleniya dostupom v operacionnykh sistemah*, Goryachaya liniya — Telekom, 2019, 214 p. (in Russian).
16. Devyanin P. N. *Modeli bezopasnosti komp'yuternyx sistem. Upravlenie dostupom i informacionnyimi potokami*, Uchebnoe posobie dlya vuzov, 2 izd., Goryachaya liniya — Telekom, 2013, 338 p. (in Russian).
17. Vasenin V. A., Itkes A. A., Buhonov V. Yu., Galatenko A. V. Access Control Models in Multiuser Scientometric Content Management Systems, *Programmnaya ingeneria*, 2016? vol. 7, no. 12, pp. 196–207 (in Russian).
18. Vasenin V. A., Itkes A. A., Shapchenko K. A. On the Application of Social Networking Access Control Models to One Class of Multi-User Content Management Systems, *Programmnaya ingeneria*, 2015, no. 4, pp. 10–19. (in Russian).
19. Grusho A. A., Timonina E. E. *Teoreticheskie osnovy' zashchity' informacii*, Uchebnoe posobie, Moscow, Izdatelstvo Agentstva "Yaxtsmen", 1996, 176 p. (in Russian).
20. Grusho A. A., Primenko E. A., Timonina E. E. *Teoreticheskie osnovy' komp'yuternoy bezopasnosti*, Moscow, Academia, 2009, 272 p. (in Russian).
21. Grusho A. A., Timonina E. E. Random graphs model for description of interactions in the network, *Informatika i eyo primenenie*, 2012, vol. 6, no. 4, pp. 61–64 (in Russian).
22. Zegzhda P. D., Rudina E. A. *Osnovy' informacionnoj bezopasnosti*, Uchebnoe posobie. Saint-Petersburg, Izdatelstvo Politehnicheskogo universiteta, 2008, 224 p. (in Russian).
23. Kalinin M. O. *Adaptivnoe upravlenie bezopasnosti informacionnykh sistem na osnove logicheskogo modelirovaniya*. Dr. Tech. sci. diss. 05.13.19, Saint Petersburg, 2011, 290 p. (in Russian).

В. И. Васильев, д-р техн. наук, проф., vasilyev@ugatu.ac.ru,
А. М. Вульфин, канд. техн. наук, доц., vulfin.alexey@gmail.com,
Л. Р. Черняховская, д-р техн. наук, проф., lrchern@yandex.ru,
Уфимский государственный авиационный технический университет

Анализ рисков инновационных проектов с использованием технологии многослойных нечетких когнитивных карт*

Рассмотрена задача анализа рисков реализации инновационных проектов с использованием многослойных (вложенных) нечетких серых когнитивных карт. Предложена методика выбора структуры этих моделей на основе декомпозиции исходной укрупненной нечеткой когнитивной карты. Приведен пример использования предложенной методики для оценки рисков неблагоприятного развития событий при реализации инновационного проекта для предприятия в сфере инфотелекоммуникационных услуг.

Ключевые слова: инновационный проект, когнитивное моделирование, многослойная нечеткая серая когнитивная карта, декомпозиция, оценка рисков

Введение

Отличительной чертой цифровой экономики является переход к стратегии инновационного развития всех ее отраслей, направленной на стимулирование инновационной активности предприятий, на внедрение инновационных продуктов и технологий, на расширение тематики и на повышение результативности масштабных инновационных проектов. В современном понимании инновационный проект (ИП) — это проект, содержащий технико-экономическое, правовое и организационное обоснования конечной инновационной деятельности [1]. Итогом разработки ИП служит документ, включающий в себя: подробное описание инновационного продукта; обоснование его жизнеспособности; обоснование необходимости и возможности; форму привлечения инвестиций; сведения о сроках использования, исполнителях; учет организационно-правовых элементов его продвижения. Целями ИП являются создание новых или изменение существующих систем — технических, технологических, информационных, социально-экономических, организационных, а также — достижение в результате снижения затрат ресурсов (производительных, финансовых, человеческих) существенного повышения качества продукции, услуг и значимого коммерческого эффекта.

Согласно данным, приведенным в ежегодном "Национальном докладе об инновациях в России — 2017"**, Россия поднялась за последние 8 лет в рейтинге развития инноваций Global Innovation Index на 19 позиций и по итогам 2017 г. заняла 45-е место.

* Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 18-00-00238 КОМФИ.

** https://www.rvc.ru/upload/iblock/c64/RVK_innovation_2017.pdf

Учитывая, что опыт создания и внедрения рыночных инноваций в России составляет всего 20—25 лет, это можно считать обнадеживающим показателем. Однако на настоящее время число ежегодно реализуемых успешных ИП крайне невелико, что, в первую очередь, объясняется объективно существующими высокими рисками реализации этих проектов.

Природа указанных рисков ситуаций заключается прежде всего в невозможности точно предсказать заранее успешность осуществления того или иного ИП по причине значительного уровня неопределенности и сложности прогнозов реальных сроков и результатов реализации ИП [2, 3]. Известно большое число подходов к качественной и количественной оценке рисков ИП [4, 5]. Методы качественной оценки рисков ИП позволяют выявить и идентифицировать появление возможных видов рисков, свойственных данному ИП, а также определить и описать причины и факторы, влияющие на уровень каждого вида рисков. К этой группе методов относятся методы экспертной оценки, деревьев решений, ранжирование рисков, метод Дельфи и др. Методы количественной оценки рисков ИП предполагают численное определение величин отдельных рисков и проекта в целом. Они позволяют установить зависимости между ключевыми факторами риска, рассмотреть возможные альтернативы реализации ИП, выбрать оптимальный (подходящий) способ реализации ИП в условиях ограниченных ресурсов. К методам данной группы относятся метод Монте-Карло, метод анализа чувствительности, метод анализа сценариев и др. Общим недостатком существующих методов является отсутствие необходимого количества достоверной статистической информации о вероятности наступления рисков ситуаций, что, естественно, сказывается на результатах конечной оценки уровня рисков ИП.

Примеры НКК оценки рисков ИП

Авторы	Назначение НКК	Число концептов/число связей НКК
Al-Shebab A. J., Hughes R. T., Winstanley G. [6]	НКК анализа рисков ИП создания информационных систем	14/20
Rodriguez-Repiso L., Setchi R., Salmeron J. L. [7]	НКК оценки рисков ИП создания мобильной платежной системы	24/35
Миннибаев Д. М., Зайцева Ж. В. [8]	НКК для управления рисками предприятия строительного комплекса	14/25
Горелова Г. В. [9]	НКК анализа механизмов и стратегии развития социально-экономической системы	34/63
Градусов Д. А., Шутов А. В., Анцупова Д. П. [10]	НКК механизма внедрения интегрированной системы управления предприятием	28/42
Case D. M., Stylios Ch. D. [11]	НКК оценки сроков выполнения ИП	41/102
Проценко Е. В., Александрова А. В. [12]	НКК оценки рисков ИП модификации продукции при размещении заказа авиастроительной компанией	32/36
Ковалев С. М., Колоденкова А. Е. [13]	НКК оценки рисков ситуаций на этапе проектирования сложных технических систем	15/36

В последние годы большую популярность при решении задачи количественной оценки риска неблагоприятных событий при реализации проекта (инновационных рисков) получили методы когнитивного моделирования, основанные на построении и анализе нечетких когнитивных карт (НКК) [6–13]. Достоинствами этих методов являются их интерпретируемость и наглядность, выявление структуры причинно-следственных связей между элементами сложной системы, не поддающейся количественному анализу традиционными методами, использование знаний и опыта экспертов в рассматриваемой предметной области, адаптация к неопределенности исходных данных и условий решаемой задачи. В табл. 1 приведены известные в литературе примеры построения НКК оценки рисков ИП, где в последнем столбце даны значения общего числа концептов НКК и числа связей между ними. Вместе с тем на практике стремление получить достаточно подробное, развернутое описание исследуемой системы приводит к чрезмерному усложнению структуры НКК ("проклятие размерности").

Выходом из сложившейся ситуации является построение НКК в классе многослойных (вложенных) НКК, основанных на декомпозиции некоторой исходной (укрупненной) НКК. Такая карта представляется в виде совокупности нескольких вложенных друг в друга "частных" НКК, содержащих локальную информацию о концептах, составляющих исходную НКК [14–16]. Учитывая, что данный этап изложен в литературе достаточно поверхностно, далее рассмотрим подробнее методику построения многослойной НКК. Будем полагать для определенности, что как исходная, так и частные НКК представляют собой нечеткие серые когнитивные карты (НСКК) (соответствующие пояснения относительно преимуществ

и особенностей НСКК будут даны ниже). В завершении статьи рассмотрим конкретный пример построения многослойной НСКК для оценки рисков ИП (стартапа), связанного с созданием нового инновационного предприятия по оказанию услуг малому и среднему бизнесу в сфере инфотелекоммуникаций.

1. Методика построения многослойных нечетких когнитивных карт

Общую идею построения многослойной НКК может пояснить рис. 1.

Здесь НКК-0 — исходная (укрупненная) НКК, соответствующая нулевому (концептуальному) уровню представления исследуемой проблемы; 1, 2, 3, 4 — базовые концепты, образующие исходную НКК-0; НКК-1(1), НКК-3(1), НКК-4(1) — нечеткие когнитивные карты первого уровня декомпозиции исходной нечеткой когнитивной карты НКК-0, раскрывающие содержание (внутреннюю структуру) своих "родительских" концептов 1, 3, 4; НКК-113(2), НКК-114(2), НКК-411(2) — нечеткие когнитивные карты

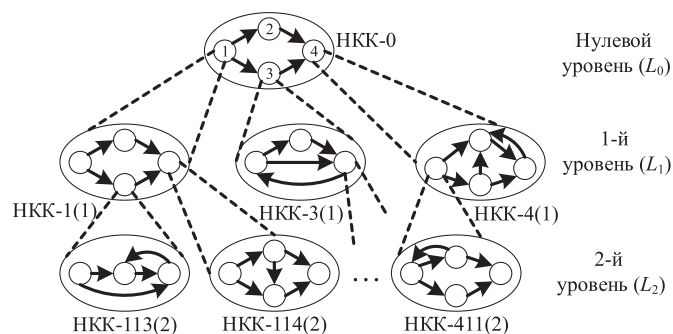


Рис. 1. Архитектура многослойной НКК

2-го уровня декомпозиции, раскрывающие (детализирующие) содержание концептов НКК-1(1), НКК-4(1) и т. д. Таким образом, верхний слой (нулевой уровень декомпозиции) НКК-0 отвечает за предоставление глобальной информации об исследуемой проблеме, тогда как последующие, ниже лежащие слои НКК (1-й, 2-й и т. д. уровни) обеспечивают дополнительную, локальную информацию о взаимодействиях и внутренних зависимостях, характеризующих поведение концептов вышележащего уровня (родительских концептов).

Согласно работе [16], разбиение НКК на слои (уровни) проводится экспертом или группой экспертов таким образом, что каждый слой L_i описывает определенный аспект понимания, глубины изучения проблемы и, следовательно, число слоев d многослойной НКК будет определяться числом принимаемых во внимание аспектов. В общем виде многослойная НКК представляет собой многослойный ориентированный граф G , определяемый кортежем множеств

$$G = \{V_M, E_M, V, L\}, \quad (1)$$

где V_M — множество вершин графа (концептов НКК), участвующих в формировании слоев в соответствии с принятым способом декомпозиции НКК; E_M — множество дуг, связывающих вершины графа (концепты НКК), входящие в V_M ; V — множество всех вершин (концептов НКК); L — множество слоев НКК.

В свою очередь, множества L , V_M , E_M определяются с помощью следующих отношений:

$$\begin{aligned} L &= \{L_a\}_{a=0}^{d-1} = L_0 \times L_1 \times \dots \times L_{d-1}; \\ V_M &\subseteq V \times L_0 L_1 \times \dots \times L_{d-1}; \\ E_M &\subseteq V_M \times V_M. \end{aligned} \quad (2)$$

Общее число концептов, входящих в многослойную НКК, равно $D = \sum_{a=0}^{d-1} |L_a|$, где $|L_a|$ — число концептов, принадлежащих слою L_a .

Ключевым вопросом построения многослойных НКК является разбиение исходной (укрупненной) НКК на слои, а также изучение взаимодействия между концептами как внутри слоя, так и между слоями. Рассмотрим подробнее методику построения многослойных НКК для анализа рисков ИП, представляющую в значительной степени ответы на указанный вопрос.

1. Нулевой уровень декомпозиции (слой L_0) НКК. Составляется исходная (укрупненная) НКК-0, которая включает в себя в качестве концептов наиболее значимые (существенные) факторы рисков ИП с указанием выявленных экспертами взаимосвязей между этими концептами (факторами).

В качестве рекомендаций общего характера при реализации данного этапа можно воспользоваться информацией, приведенной в работе [17], а также (с учетом специфики анализа рисков ИП) в работах [7, 10]. Оценка силы связей НКК-0 при этом является предварительной (приближенной), уточнение

силы связей между концептами откладывается до следующего этапа построения НКК.

2. Первый уровень декомпозиции (слой L_1) НКК.

На данном этапе проводится "раскрытие" всех или части концептов, входящих в состав укрупненной нечеткой когнитивной карты НКК-0. Рассмотрим, как это делается, на следующем примере. Допустим, что некоторый концепт C_1 , принадлежащий НКК-0, реализует причинно-следственную (каузальную) связь ЕСЛИ A_1 И A_2 , ТО F (рис. 2, а). Здесь A_1 , A_2 и F — события, которые могут произойти с определенной степенью вероятности (уверенности). Будем полагать, что вероятности наступления указанных событий определяются параметрами X_1 , X_2 , X_3 (переменными состояниями), принадлежащими интервалу $[0, 1]$. Предположим теперь, что рассматриваем отношение $(A_1 \text{ И } A_2)F$ как сложное событие, которое можно представить в виде цепочки последовательных элементарных событий: $A_1 \rightarrow B_1$; $A_2 \rightarrow B_2$; $(B_1 \text{ И } B_2)D_1$; $B_2 \rightarrow D_2$; $(D_1 \text{ И } D_2)F$. Тогда родительский концепт C_1 замещается частной НКК, состоящей из пяти концептов C_1^1 , C_1^2 , C_1^3 , C_1^4 , C_1^5 (см. рис. 2, б).

В основе данного преобразования лежит отношение эквивалентности [18], согласно которому родительский концепт и замещающая его частная НКК считаются эквивалентными, если они имеют одинаковые входы и выходы, реализуют эквивалентные (т. е. взаимно преобразуемые) схемы логического вывода и одинаково интерпретируемы (хотя и с разной глубиной понимания) в рамках общей (укрупненной) НКК исследуемой проблемы.

На этом же этапе уточняются значения весов связей между концептами (уже применительно к частным НКК) с учетом дополнительной информации, полученной от экспертов.

3. Следующие уровни декомпозиции (слои L_2 , L_3 , ...)

НКК. Аналогично, любой из концептов частных НКК, построенных на предыдущем уровне декомпозиции, может быть, в свою очередь, развернут (детализирован), что в итоге даст возможность еще более глубокого анализа всех аспектов изучаемой проблемной области за счет включения в рассмотрение дополнительных ниже лежащих слоев L_2 , L_3 и т. д. Соответственно, пересматриваются (уточняются) значения весов связей между концептами НКК.

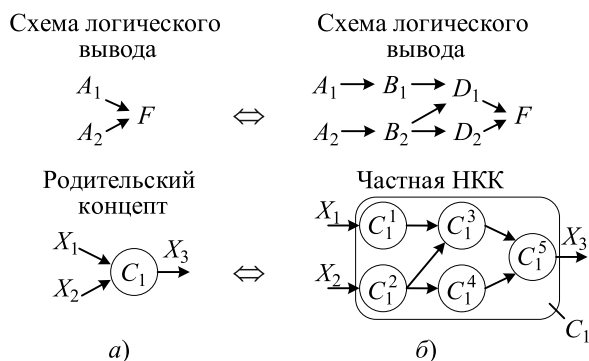


Рис. 2. Общая схема декомпозиции:

а — родительский концепт C_1 ; б — частная НКК

4. Моделирование поведения (динамики) полного набора нечетких когнитивных карт.

Для каждой НКК (начиная с частных НКК нижних уровней декомпозиции L_1, L_2, \dots и заканчивая укрупненной НКК самого верхнего уровня L_0) проводится расчет динамики изменения состояния концептов. Будем полагать, что каждая из указанных НКК представляет собой НСКК. Понятие НСКК впервые было введено в 2010 г. Хосе Салмероном [19]. Считается, что НСКК лучше учитывают объективно существующие факторы неопределенности, позволяют учесть разброс мнений экспертов и предоставляют больше степеней свободы лицу, принимающему решение (ЛПР) на основании результатов моделирования.

С формальной точки зрения НСКК — это когнитивная модель в виде ориентированного графа, заданного с помощью следующего набора множеств:

$$\text{НСКК} = \{C, E, W\}, \quad (3)$$

где $C = \{C_i\}$ — множество концептов (вершин графа), ($i = 1, 2, \dots, n$); $E = \{E_{ij}\}$ — множество связей между концептами (дуг графа); $W = \{W_{ij}\}$ — множество весов связей, $(i, j) \in \Omega$. Здесь $\Omega = \{(i_1, j_1), (i_2, j_2), \dots, (i_S, j_S)\}$ — множество пар индексов смежных (т. е. связанных между собой) вершин графа, $S \leq n(n-1)$.

В отличие от традиционного понимания НКК, веса связей НСКК задаются с помощью серых (интервальных) чисел $\otimes W_{ij}$, определяемых как

$$\otimes W_{ij} \in [\underline{W}_{ij}, \overline{W}_{ij}], \text{ где } \underline{W}_{ij} < \overline{W}_{ij}, \{\underline{W}_{ij}, \overline{W}_{ij}\} \in [-1, 1]. \quad (4)$$

Здесь \underline{W}_{ij} и \overline{W}_{ij} — нижняя и верхняя границы серого числа $\otimes W_{ij}$ соответственно. Таким образом, вес связи между i -м и j -м концептами ($C_i \rightarrow C_j$) может принимать любое значение в пределах заданного диапазона $[\underline{W}_{ij}, \overline{W}_{ij}] \in [-1, 1]$. В частном случае, когда $\underline{W}_{ij} = \overline{W}_{ij}$, получаем $\otimes W_{ij} \in [\underline{W}_{ij}, \underline{W}_{ij}]$ — "белое" (четкое, обычное) число.

Основные операции над серыми числами [20]:

$$1) \otimes W_1 + \otimes W_2 \in [\underline{W}_1 + \underline{W}_2, \overline{W}_1 + \overline{W}_2];$$

$$2) -\otimes W \in [-\overline{W}, -\underline{W}];$$

$$3) \otimes W_1 - \otimes W_2 \in [\underline{W}_1 - \overline{W}_2, \overline{W}_1 - \underline{W}_2];$$

$$4) \otimes W_1 \times \otimes W_2 \in [\min(S), \max(S)],$$

где $S = \{\underline{W}_1 \cdot \underline{W}_2, \underline{W}_1 \cdot \overline{W}_2, \overline{W}_1 \cdot \underline{W}_2, \overline{W}_1 \cdot \overline{W}_2\}$;

$$5) \text{ Если } \lambda > 0, \lambda \in \mathbb{R}, \text{ то } \lambda \cdot \otimes W \in [\lambda \underline{W}, \lambda \overline{W}].$$

Изменение состояния концептов во времени описывается уравнениями

$$\otimes X_i(k+1) = f\left(\otimes X_i(k) + \sum_{\substack{j=1 \\ (j \neq i)}}^n \otimes W_{ji} \otimes X_j(k)\right), \quad (5)$$

$$(i = 1, 2, \dots, n),$$

где $\otimes X_j(k)$ — "серая" (интервальная) переменная состояния i -го концепта C_j , которая в каждый мо-

мент времени принимает некоторое значение внутри интервала (диапазона изменения) $[\underline{X}_i(k), \overline{X}_i(k)]$; $f(\cdot)$ — нелинейная функция активации i -го концепта, принимающая значения из интервала $[-1, 1]$. Будем полагать далее, что функция активации — это двухполярная сигмоида (гиперболический тангенс):

$$f(x) = (1 - e^{-x}) / (1 + e^{-x}) = \text{th}(x/2). \quad (6)$$

При выборе серых значений весов $\otimes W_{ij}$ экспертам необходимо ориентироваться на некоторую нечеткую шкалу, наподобие той, которая представлена в табл. 2, начиная с выбора "центров" соответствующих интервалов W_{ij}^0 . В табл. 2 значения термов: Z — Zero; VS — Very Small; S — Small; M — Middle; L — Large; VL — Very Large. Следующим шагом, определяющим действия эксперта, будет выбор границ интервала $[\underline{W}_{ij}, \overline{W}_{ij}]$, определяющего серое значение силы связи $\otimes W_{ij}$. Это могут быть равноотстоящие от центрального значения W_{ij}^0 числа, например, $\otimes W_{ij} \in [W_{ij}^0 - \delta_{ij}, W_{ij}^0 + \delta_{ij}]$, где $\pm \delta_{ij}$ — разброс оценки относительно центра W_{ij}^0 , но возможны и другие варианты.

Для решения системы уравнений (5) требуется задать начальные условия для переменных состояния $\otimes X_i(k)$, которые также представляют собой серые числа $\otimes X_i(0) \in [\underline{X}_i(0), \overline{X}_i(0)]$. Наибольший интерес при расчетах представляет получение равновесного (установившегося) решения, которое представляет собой серый вектор $\lim_{k \rightarrow \infty} [\otimes X(k)] = X^* \in [\underline{X}^*, \overline{X}^*]$ или предельный цикл (странный аттрактор). Особенности обучения НСКК с применением нелинейного правила Хебба обсуждены в работе [21], где предлагается процедура обучения с учителем, основанная на коррекции весовых коэффициентов с помощью нелинейного правила обучения Хебба, для улучшения структуры нечеткой серой когнитивной карты. Введение процедуры обучения для связей концептов в НСКК повышает гибкость модели.

После выполнения всех расчетов для частных (локальных) НКК, расположенных на нижних слоях НКК, проводится агрегирование полученных результатов моделирования на верхнем уровне НКК (слой L_0) с вычислением интегральных показателей оценки рисков ИП.

Таблица 2

Оценка силы (весов) связей между концептами

Лингвистическое значение силы связи	Терм	Числовой диапазон
Не влияет	Z	0
Очень слабая	VS	(0; 0,15]
Слабая	S	(0,15; 0,35]
Средняя	M	(0,35; 0,06]
Сильная	L	(0,6; 0,85]
Очень сильная	VL	(0,85; 1]

5. Сценарное моделирование с использованием НКК. Завершающим этапом когнитивного моделирования является этап сценарного моделирования, на котором проводится анализ различных вариантов воздействия факторов риска с оценкой ожидаемых последствий, выбор возможных контрмер для снижения уровня рисков ИП, формирование рекомендаций ЛПР в отношении наиболее предпочтительного варианта реализации ИП.

2. Пример оценки и анализа рисков ИП с использованием технологии многослойных НКК

Рассмотрим в качестве примера ИП (стартап), направленный на создание нового инновационного предприятия по оказанию услуг малому и среднему бизнесу в сфере инфотелекоммуникаций. Предполагается, что инициатор ИП разработал бизнес-план, определил стратегию развития предприятия, провел маркетинговые исследования планируемой ниши рынка, договорился с предполагаемыми инвесторами ИП.

На этой стадии важно правильно провести оценку рисков реализации ИП, обусловленных наличием таких факторов риска, как:

- 1) несвоевременное финансирование ИП;
- 2) ошибки на этапе предпроектного исследования (в формулировке целей ИТ, оценке привлекательности предоставляемых услуг, сроков выхода на рынок, объемов ожидаемого рынка и т. п.);
- 3) задержки плановых сроков реализации отдельных этапов ИП (задержки с оформлением документации, сроков согласования ИП, поставок оборудования, ПО и т. п.);
- 4) недобросовестные действия со стороны конкурентов;
- 5) неэффективная рекламная (маркетинговая) компания.

Перечисленные факторы риска в свою очередь приводят к следующим последствиям:

- 6) срыв сроков начала реализации ИП;
- 7) прекращение финансирования ИП;
- 8) отсутствие привлекательности (востребованности) предлагаемого инновационного продукта;
- 9) просчеты в оценке рентабельности (окупаемости) вкладываемых средств на реализацию ИП.

Интегральные показатели риска ИП:

- 10) финансовый риск (потеря вложенных средств);
- 11) репутационный риск (прекращение деятельности предприятия).

Соответствующая укрупненная НКК (слой L_0), отражающая все перечисленные выше факторы риска, последствия от их реализации и интегральные показатели риска ИП, а также взаимодействия между ними, приведена на рис. 3.

Перейдем к 1-му уровню декомпозиции НКК (слой L_1). На рис. 4 приведена частная НКК, раскрывающая внутреннее содержание родительского концепта C_1 (несвоевременное финансирование ИП).

Приведем еще несколько вариантов декомпозиции концептов исходной (укрупненной) НКК.

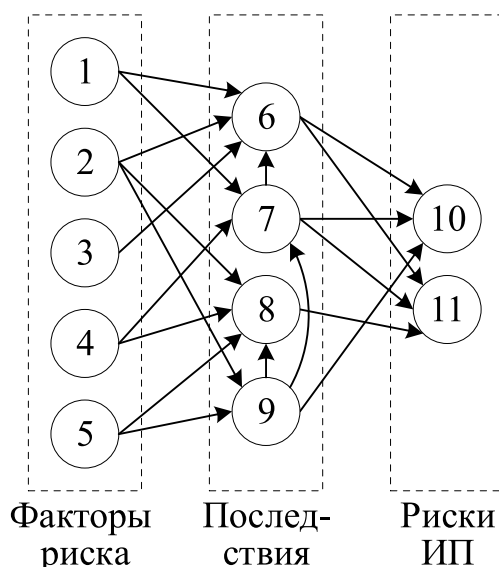


Рис. 3. Укрупненная НКК оценки рисков ИП

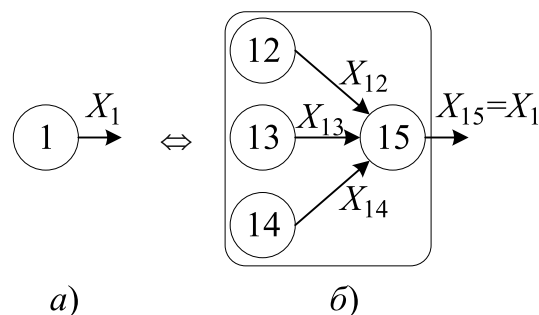


Рис. 4. Декомпозиция концепта C_1 :

a — концепт C_1 ; b — частная НКК; 12 — задержки с финансированием со стороны инвестора на предпроектной стадии ИП; 13 — перебой с банковским обслуживанием; 14 — пересмотр условий финансирования ИП (снижение суммы контракта); 15 — нарушения графика финансирования ИП; X_{12} , X_{13} , X_{14} , X_{15} — переменные состояния соответствующих концептов C_{12} , C_{13} , C_{14} , C_{15}

Концепт C_6 (срыв сроков начала реализации ИП) представлен на рис. 5.

Концепт C_7 (прекращение финансирования ИП) приведен на рис. 6.

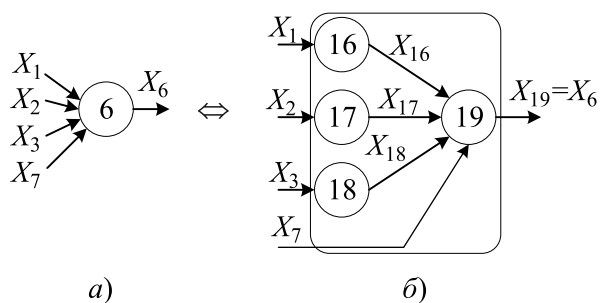


Рис. 5. Декомпозиция концепта C_6 :

a — концепт C_6 ; b — частная НКК; 16 — привлечение дополнительного инвестора; 17 — необходимость выполнения дополнительных работ; 18 — невыполнение запланированного графика работ над ИП; 19 — срыв сроков начала реализации ИП

Концепт C_{10} (финансовый риск) изображен на рис. 7.

Концепт C_{11} (репутационный риск) представлен на рис. 8.

Аналогичным образом можно (при необходимости) провести декомпозицию всех оставшихся

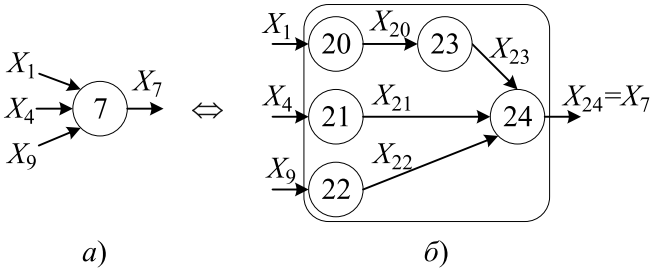


Рис. 6. Декомпозиция концепта C_7 :

a — концепт; b — частная НСКК; 20 — низкая исполнительская дисциплина участников ИП; 21 — декомпозиция инвестора и потребителей инновационного продукта; 22 — отсутствие преимуществ инновационного продукта по сравнению с аналогами; 23 — низкое качество ИП; 24 — прекращение финансирования ИП

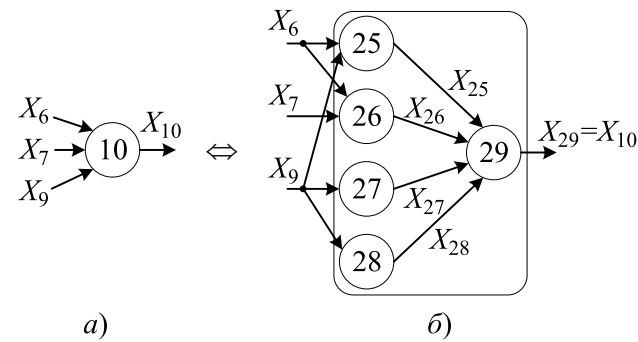


Рис. 7. Декомпозиция концепта C_{10} :

a — концепт C_{10} ; b — частная НСКК; 25 — невыход инновационного продукта на рынок; 26 — срыв обязательств перед заказчиком (инвестором); 27 — невыполнение ожиданий потребителя (уменьшение числа заказов); 28 — прекращение реализации ИП (с возвратом полностью или части вложенных средств инвестору); 29 — потеря финансовых средств (убыток предприятия)

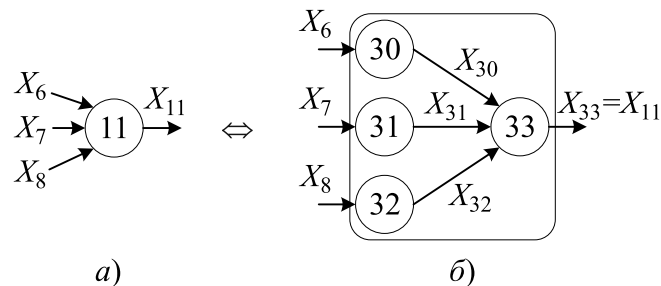


Рис. 8. Декомпозиция концепта C_{11} :

a — концепт C_{11} ; b — частная НСКК; 30 — потеря интереса к инновационному предприятию со стороны потребителей; 31 — прекращение деловых контактов с инвестором; 32 — потеря планируемой доли рынка; 33 — снижение уровня (имиджа) предприятия по отношению к партнерам (деловому сообществу)

концептов укрупненной НСКК (см. рис. 3), а также провести более детальную декомпозицию полученных частных НСКК (слой L_2, L_3, \dots) с учетом дополнительной информации о внутренней структуре и характеристиках соответствующих концептов.

3. Результаты когнитивного моделирования с использованием НСКК

На этапе численного моделирования НСКК использовалось инструментальное средство Cognitive Map Constructor, разработанное на кафедре вычислительной техники и защиты информации УГАТУ.

Будем полагать, что укрупненная НСКК принимает вид, изображенный на рис. 3. Рассмотрим в качестве примера один из вариантов сценарного моделирования, связанный с оценкой финансового и репутационного рисков (переменные X_{10} и X_{11}), которые вызваны задержками финансирования со стороны инвестора на предпроектной стадии ИП (входной концепт C_{12} в составе частной НСКК на рис. 4). Определим значения переменной состояния $X_{12}(k)$ указанного концепта как $\otimes X_{12}(k) \in [0, 9; 1]$ для всех моментов времени $k = 0, 1, 2, \dots$; начальные условия для всех остальных переменных состояния (как исходной НСКК, так и частных НСКК) принимаем нулевыми, т. е. равными $[0; 0]$.

Изменение во времени переменных состояния НСКК будет определяться уравнениями динамики (5). Однако заметим, что в рассматриваемом случае из общего числа (11 концептов) лишь пять концептов (C_1, C_6, C_7, C_{10} и C_{11}) являются активными, т. е. участвуют в процедуре оценки искомых рисков. Влияние остальных шести концептов ($C_2, C_3, C_4, C_5, C_8, C_9$) на показатели рисков X_{10} и X_{11} здесь не учитывается, так как переменные состояния этих концептов сохраняют нулевые значения для всех $k = 0, 1, 2, \dots$. Соответствующие частные (детализированные) НСКК для указанных активных концептов ($C_1, C_6, C_7, C_{10}, C_{11}$) приведены на рис. 4—8. Далее допустим, что эксперты вычислили значения весов связей внутри каждой из частных НСКК определенным образом (табл. 3—7).

Полагаем, что решение уравнений динамики (5) осуществляется последовательно для перечисленных выше частных НСКК. Нас при этом интересуют только установившиеся значения переменных состояния $\otimes X_1^*, \otimes X_6^*, \otimes X_7^*, \otimes X_{10}^*, \otimes X_{11}^*$. Общую схему вычислений указанных переменных можно представить в виде, приведенном на рис. 9. Как видно на рис. 9, за k_1 итераций (этап I) вначале вычисляется значение $\otimes X_{15}^* = \otimes X_1^*$, затем за $\Delta k_2 = k_2 - k_1$ итераций (этап II) вычисляются значения $\otimes X_6^*$ и $\otimes X_7^*$, и, наконец, за $\Delta k_3 = k_3 - k_2$ итераций (этап III) вычисляются

Таблица 3

Весы связей частной НСКК для концепта C_1

Вес связи	$W_{12, 15}$	$W_{13, 15}$	$W_{14, 15}$
Терм	VL	L	VL
Числовой диапазон	[0,9; 1]	[0,7; 0,8]	[0,9; 1]

Веса связей частной НСКК для концепта C_6

Вес связи	$W_{1,16}$	$W_{2,17}$	$W_{3,18}$	$W_{7,19}$	$W_{16,19}$	$W_{17,19}$	$W_{18,19}$
Терм	М	L	VL	VL	L	L	VL
Числовой диапазон	[0,5; 0,6]	[0,75; 0,85]	[0,9; 1]	[0,9; 1]	[0,7; 0,8]	[0,75; 0,85]	[0,9; 1]

Таблица 5

Веса связей частной НСКК для концепта C_7

Вес связи	$W_{1,20}$	$W_{4,21}$	$W_{9,24}$	$W_{20,23}$	$W_{21,24}$	$W_{22,24}$	$W_{23,24}$
Терм	L	М	L	VL	S	VL	L
Числовой диапазон	[0,6; 0,8]	[0,4; 0,5]	[0,7; 0,8]	[0,85; 0,9]	[0,25; 0,35]	[0,85; 1]	[0,75; 0,85]

Таблица 6

Веса связей частной НСКК для концепта C_{10}

Вес связи	$W_{6,25}$	$W_{7,26}$	$W_{9,25}$	$W_{9,27}$	$W_{9,28}$	$W_{25,29}$	$W_{26,29}$	$W_{27,29}$	$W_{28,29}$
Терм	М	VL	L	М	L	VL	L	L	VL
Числовой диапазон	[0,5; 0,6]	[0,85; 1]	[0,7; 0,8]	[0,4; 0,6]	[0,8; 0,85]	[0,9; 1]	[0,75; 0,85]	[0,7; 0,8]	[0,9; 1]

Таблица 7

Веса связей частной НСКК для концепта C_{11}

Вес связи	$W_{6,30}$	$W_{7,31}$	$W_{9,32}$	$W_{30,33}$	$W_{31,33}$	$W_{32,33}$
Терм	М	VL	L	L	L	L
Числовой диапазон	[0,5; 0,6]	[0,9; 1]	[0,8; 0,85]	[0,7; 0,8]	[0,6; 0,8]	[0,75; 0,85]

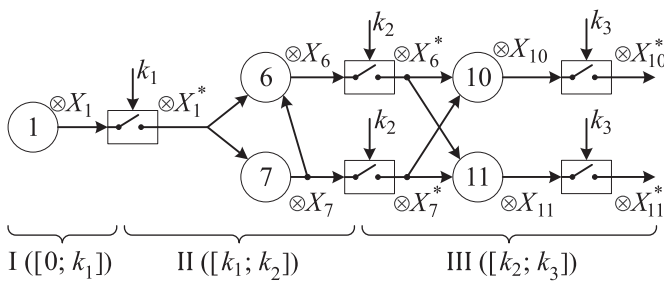


Рис. 9. Схема вычисления рисков (установившихся значений переменных состояния $\otimes X_{10}^*$ и $\otimes X_{11}^*$)

искомые значения рисков $\otimes X_{10}^*$ (финансовый риск) и $\otimes X_{11}^*$ (репутационный риск).

Фактически, с учетом принятых ранее нулевых начальных условий (за исключением концепта C_{12}), будет моделироваться следующая детализированная схема НСКК (рис. 10), где $\otimes X_{15}^* = \otimes X_1^*$, $\otimes X_{19}^* = \otimes X_6^*$, $\otimes X_{24}^* = \otimes X_7^*$, $\otimes X_{29}^* = \otimes X_{10}^*$, $\otimes X_{33}^* = \otimes X_{11}^*$.

Результаты моделирования: $\otimes X_{15}^* \in [0,61; 0,69]$ (достигается за $k_1 = 10$ итераций); $\otimes X_{19}^* \in [0,35; 0,50]$ и $\otimes X_{24}^* \in [0,20; 0,31]$ (потребовалось $\Delta k_2 = 20$ итераций); $\otimes X_{29}^* \in [0,36; 0,57]$ и $\otimes X_{33}^* \in [0,22; 0,41]$ (потребовалось $\Delta k_3 = 15$ итераций).

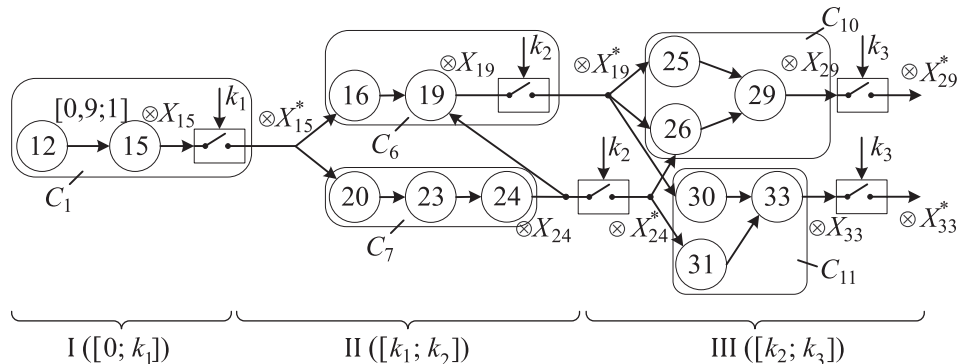


Рис. 10. Детализированная схема вычисления рисков ИП

Скорректированные значения весов связей

Вес связи	$\tilde{W}_{15,16}$	$\tilde{W}_{16,19}$	$\tilde{W}_{24,19}$	$\tilde{W}_{15,20}$	$\tilde{W}_{20,23}$	$\tilde{W}_{23,24}$
Терм	S	M	M	M	M	S
Числовой диапазон	[0,25; 0,3]	[0,45; 0,5]	[0,45; 0,5]	[0,45; 0,5]	[0,55; 0,6]	[0,25; 0,3]

Допустим (в качестве еще одного сценария), что за счет применения заранее продуманных организационных контрмер, по мнению экспертов, удастся снизить влияние отдельных факторов риска на результирующие значения прогнозируемых рисков, что сказывается на изменении весов частных НСКК:

$$W_{1,16} \rightarrow \tilde{W}_{15,16}; W_{16,19} \rightarrow \tilde{W}_{16,19}; W_{7,19} \rightarrow \tilde{W}_{24,19}; \\ W_{1,20} \rightarrow \tilde{W}_{15,20}; W_{20,23} \rightarrow \tilde{W}_{20,23}; W_{23,24} \rightarrow \tilde{W}_{23,24}.$$

Новые значения скорректированных весов приведены в табл. 8.

Результаты моделирования для данного сценария:

$$\otimes X_1^* \in [0,61; 0,69]; \otimes X_6^* \in [0,08; 0,12];$$

$$\otimes X_7^* \in [0,04; 0,05]; \otimes X_{10}^* \in [0,07; 0,11];$$

$$\otimes X_{11}^* \in [0,03; 0,06].$$

Таким образом, применение указанных контрмер позволило уменьшить финансовый риск, определяемый показателем $\otimes X_{10}^*$, в среднем в 5,1 раза, а репутационный риск, оцениваемый показателем $\otimes X_{11}^*$, в среднем в 7 раз.

Аналогичным образом можно рассмотреть другие варианты сценариев с использованием предложенной методики когнитивного моделирования, оценивая уровень рисков ИП от воздействия других дестабилизирующих факторов и эффективность принятия тех или иных мер по их снижению.

Заключение

Вопросы анализа и управления рисками на настоящее время являются одними из ключевых при разработке и реализации крупномасштабных инновационных проектов в силу наличия большого числа факторов неопределенности (факторов риска), связанных с возможными ошибками в ходе проектирования, недостаточно высокой компетентностью персонала, сбоями в финансировании проекта и т. д. Отмеченные обстоятельства зачастую приводят к увеличению стоимости проекта, затягиванию сроков его реализации, снижению качества ожидаемых результатов ИП. В качестве перспективного подхода к оценке рисков ИП сегодня все шире используются методы когнитивного моделирования, основанные на построении и анализе нечетких когнитивных карт. В настоящей работе для решения задачи оценки рисков ИП предложено использование многослойных нечетких серых когнитивных карт,

представляющих несомненный интерес для широкого круга специалистов. Преимуществом данного класса когнитивных моделей, кроме их наглядности, является возможность последовательного раскрытия неопределенности на каждом последующем (нижележащем) слое нечеткой когнитивной карты, с привлечением дополнительной информации от экспертов о содержании базовых концептов (факторов риска). Кроме того, использование данного класса нечетких когнитивных моделей обеспечивает возможность учета разброса мнений экспертов о взаимном влиянии концептов и, как следствие, позволяет обеспечить больше степеней свободы лицу, принимающему решение на основании результатов моделирования. Особенности предложенной методики анализа рисков иллюстрируются на примере построения нечеткой когнитивной карты оценки рисков ИП по созданию инновационного предприятия в сфере оказания инфотелекоммуникационных услуг малому и среднему бизнесу.

Список литературы

1. **Википедия.** Инновационный проект. URL: https://ru.wikipedia.org/wiki/Инновационный_проект.
2. **Калиновская С. Ю.** Анализ инновационных рисков с учетом современных тенденций развития инновационной деятельности // Вестник Волжского университета им. В. Н. Татищева. — 2013. — № 1 (27). — С. 65–72.
3. **Русинов В. М., Ситников В. В.** Управление рисками инновационных проектов: проблемы и решения // Международный журнал гуманитарных и естественных наук. 2018. Т. 6, Часть 1. С. 97–99.
4. **Прокопьева М. В.** Методы управления рисками инновационного проекта // Молодой ученый. — Февраль 2009. — № 2 (2). — С. 75–81.
5. **Deptula A. M., Knosala R.** Risk assessment of the innovative projects implementation // Management and Production Engineering Review. — 2015. — Vol. 6, No. 4. — P. 15–25.
6. **Al-Shebab A. J., Hughes R. T., Winstanley G.** Modelling Risks in IS/IT Projects through Causal and Cognitive Mapping // Electronic Journal of Information Systems Evaluation. — 2005. — Vol. 8, Issue 1. — P. 1–10.
7. **Rodriguez-Repiso L., Setchi R., Salmeron J. L.** Modelling IT Projects Success with Fuzzy Cognitive Maps // Expert Systems with Applications. — 2007. — Vol. 32, No. 2. — P. 543–559.
8. **Миннибаев Д. М., Зайцева Ж. В.** Когнитивные модели для управления инвестиционными рисками предприятий строительного комплекса // Вестник ОГУ. — Октябрь 2008. — № 10 (92). — С. 91–95.
9. **Горелова Г. В.** Когнитивный подход к имитационному моделированию сложных систем // Известия ЮФУ. Технические науки. — 2013. — № 3 (140). — С. 239–250.
10. **Градусов Д. А., Шутов А. В., Анцупова Д. П.** Моделирование взаимовлияний целей и рисков проектов внедрения интегрированной системы управления предприятием на основе нечетких когнитивных карт // Экономический анализ: Теория и практика. — 2013. № 37 (340). — С. 50–59.
11. **Case D. M., Stylios Ch. D.** Fuzzy Cognitive Map to model project management problems // 2016 Annual Conference of the

North American Fuzzy Information Processing Society (NAFIPS). IEEE, 2016. — P. 1–6.

12. **Проценко Е. В., Александрова А. В.** Когнитивный подход в оценке рисков инновационных проектов // Экономические аспекты развития промышленности в условиях глобализации. Матер. Междунар. науч.-практ. конф. — М.: Ун-т машиностроения, Изд-во "Научный консультант", 2015. — С. 213–216.

13. **Ковалев С. М., Колоденкова А. Е.** Построение базы знаний интеллектуальной системы контроля и предупреждения рисков ситуаций для этапа проектирования сложных технических систем // Онтология проектирования. — 2017. — Т. 7, № 4 (26). — С. 398–409.

14. **Боженюк А. В., Гинис Л. А.** Об использовании нечетких баз и антибаз при анализе нечетких когнитивных карт // Искусственный интеллект. — 2004. — № 4. — С. 276–285.

15. **Aguilar J.** Multilayer Cognitive Maps in the Resolution of Problems using the FCM Designer Tool // Applied Artificial Intelligence. — 2016. — Vol. 30, No. 7. — P. 720–743.

16. **Mohr S.** Modelling Approaches for Multilayer Fuzzy Cognitive Maps. URL: <https://www.researchgate.net/>

publication/332158518_Modelling_Approaches_for_Multilayer_Fuzzy_Cognitive_Maps.

17. **Кулинич А. А.** Компьютерные системы моделирования когнитивных карт: подходы и методы // Проблемы управления. — М.: Изд-во ИПУ РАН им. В. А. Трапезникова, 2010. — Вып. 3. — С. 2–16.

18. **Zhang J. Y., Liu Z. Q., Zhou S.** Quotient FCMs — A Decomposition Theory for Fuzzy Cognitive Maps // IEEE Transactions on Fuzzy Systems. Oct. 2003. Vol. 11, No. 5. P. 593–604.

19. **Salmeron J. L.** Modelling grey uncertainty with Fuzzy Grey Cognitive Maps // Expert Systems with Applications. — 2010. Vol. 37, No. 12. — P. 7581–7588.

20. **Harmati I. S., Koczy L. T.** On the Convergence of Fuzzy Grey Cognitive Maps // Information Technology, Systems Research, and Computational Physics. Springer Verlag. — 2018. — Vol. 945. — P. 74–84.

21. **Papageorgiou E. I., Salmeron J. L.** Learning Fuzzy Grey Cognitive Maps using Nonlinear Hebbian-based approach // International Journal of Approximate Reasoning. — 2012. — Vol. 53, No. 1. — P. 54–65.

Risk Analysis of Innovative Projects with Use of Multilayer Fuzzy Cognitive Maps

V. I. Vasilyev, vasilyev@ugatu.ac.ru, **A. M. Vulfin**, vulfin.alexey@gmail.com,
L. R. Chernyakhovskaya, lrchern@yandex.ru, Ufa State Aviation Technical University,
Ufa, 450000, Russian Federation

Corresponding author:

Vulfin Aleksey M., Associate Professor, Ufa State Aviation Technical University, Ufa, 450000,
Russian Federation
E-mail: vulfin.alexey@gmail.com

Received on November 25, 2019

Accepted on December 26, 2019

The problem of innovative projects risks assessment with use of fuzzy cognitive maps is considered. The special attention is paid to applying a new advanced class of cognitive models for these purposes — multilayer (nested) fuzzy grey cognitive maps (FGCM). The advantages of this class of models, allowing us to reveal the structure of causal relationships between elements of explored system with account of knowledge and experience of experts in considered subject domain, are described. The technique of choosing the rational structure of these models on the basis of formal decomposition rules for initial fuzzy cognitive map taking into account the specifics of the task being solved is offered. In accordance with this technique, at first the construction of initial (enlarged) FGCM is carried out by highlighting its basic concepts, including the risk factors, their consequences, and resulting risks of innovative project. The next stage consists in construction of the set of partial FGCMs revealing the content of each basic concept with use of additional local information about their inner structure. Further, the experts assign weights to the grey connections of the partial FGCMs. The final stage of the offered technique consists in modeling the obtained FGCM for different scenarios taking into account the influence of the different risk factors on potential risks of the innovative project implementation. The example of using this technique for risk assessment of the creation of a new innovative product in the sphere of infotelecommunication services is considered. The results of conducted computer experiments with use of the developed instrumental software tool "Cognitive Map Constructor" are considered.

Keywords: innovative project, cognitive modeling, multilayer fuzzy grey cognitive map, decomposition, risk assessment

For citation:

Vasilyev V. I., Vulfin A. M., Chernyakhovskaya L. R. Risk Analysis of Innovative Projects with Use of Multilayer Fuzzy Cognitive Maps, *Programmная Ingeneria*, 2020, vol. 11, no. 3, pp. 142–151

DOI: 10.17587/prin.11.142-151

References

1. **Wikipedia**. Innovative project, available at: https://www.ru.wikipedia.org/wiki/Инновационный_проект (in Russian).
2. **Kalinovskaya S. U.** Research of innovation risks in view of modern trends of innovative activity development, *Vestnik Volzhskogo universiteta im. V. N. Tatishcheva*, 2013, no. 1 (27), pp. 65–72 (in Russian).
3. **Rusinov V. M., Sitnikov V. V.** Risk management of innovative projects implementation, *Mezhdunarodnyj zhurnal gumanitarnykh i estestvennykh nauk*, 2018, vol. 6, part 1, pp. 97–99 (in Russian).
4. **Prokop'eva M. V.** Innovative project risks management methods, *Molodoy uchenyj*, February 2009, no. 2 (2), pp. 75–81 (in Russian).
5. **Deptula A. M., Knosala R.** Risk assessment of innovative projects implementation, *Management and Production Engineering Review*, 2015, vol. 6, no. 4, pp. 15–25.
6. **Al-Shebab A. J., Hughes R. T., Winstanley G.** Modelling Risks in IS/IT Projects through Causal and Cognitive Mapping, *Electronic Journal of Information Systems Evaluation*, 2005, vol. 8, issue 1, pp. 1–10.
7. **Rodriguez-Repiso L., Setchi R., Salmeron J. L.** Modelling IT Projects Success with Fuzzy Cognitive Maps, *Expert Systems with Applications*, 2007, vol. 32, no. 2, pp. 543–559.
8. **Minnibaev D. M., Zajceva Zh. V.** Cognitive models for investment risk management of construction enterprises complex, *Vestnik OGU*, October 2008, no. 10(92), pp. 91–95 (in Russian).
9. **Gorelova G. V.** Cognitive approach to simulation of large systems, *Izvestiya YuFU. Tekhnicheskie nauki*, 2013, no. 3 (140), pp. 239–250 (in Russian).
10. **Gradusov D. A., Shutov A. V., Ancupova D. P.** Simulation of the mutual influence of goals and risks of projects implementing an integrated enterprise management system based on fuzzy cognitive maps, *Ekonomicheskij analiz: Teoriya i praktika*, 2013, no. 37 (340), pp. 50–59 (in Russian).
11. **Case D. M., Stylios Ch. D.** Fuzzy Cognitive Map to model project management problems, *2016 Annual Conference of the North American Fuzzy Information Processing Society (NAFIPS)*, IEEE, 2016, pp. 1–6.
12. **Protsenko E. V., Aleksandrova A. V.** Cognitive approach to risk assessment of innovative projects, *Ekonomicheskie aspekty razvitiya promyshlennosti v usloviyah globalizacii, Materialy Mezhdunarodnoj nauchno-prakticheskoy konferencii*. Moscow, Un-t mashinostroeniya, Izd-vo "Nauchnyj konsul'tant", 2015, pp. 213–216 (in Russian).
13. **Kovalev S. M., Kolodenkova A. E.** Knowledge base design for intelligent system of control and prevention of risk situations for design stage of complex technical systems, *Ontologiya proektirovaniya*, 2017, vol. 7, no. 4 (26), pp. 398–409. (in Russian).
14. **Bozhenuk O. V., Ginis L. O.** On Use of Fuzzy Bases and Antibases at the Analysis of Fuzzy Cognitive Maps, *Iskusstvennyj intellect*, 2004, no. 4, pp. 276–285 (in Russian).
15. **Aguilar J.** Multilayer Cognitive Maps in the Resolution of Problems using the FCM Designer Tool, *Applied Artificial Intelligence*, 2016, vol. 30, no. 7, pp. 720–743.
16. **Mohr S.** Modelling Approaches for Multilayer Fuzzy Cognitive Maps, available at: https://www.researchgate.net/publication/332158518_Modelling_Approaches_for_Multilayer_Fuzzy_Cognitive_Maps
17. **Kulinich A. A.** Computer systems for cognitive maps simulation: Approaches and methods, *Problemy upravleniya*. Moscow, Izd-vo IPU RAN im. V. A. Trapeznikova, 2010, no. 3, pp. 2–16 (in Russian).
18. **Zhang J. Y., Liu Z. Q., Zhou S.** Quotient FCMs — A Decomposition Theory for Fuzzy Cognitive Maps, *IEEE Transactions on Fuzzy Systems*, Oct. 2003, vol. 11, no. 5, pp. 593–604.
19. **Salmeron J. L.** Modelling grey uncertainty with Fuzzy Grey Cognitive Maps, *Expert Systems with Applications*, 2010, vol. 37, no. 12, pp. 7581–7588.
20. **Harmati I. S., Koczy L. T.** On the Convergence of Fuzzy Grey Cognitive Maps, *Information Technology, Systems Research, and Computational Physics*, Springer Verlag, 2018, vol. 945, pp. 74–84.
21. **Papageorgiou E. I., Salmeron J. L.** Learning Fuzzy Grey Cognitive Maps using Nonlinear Hebbian-based approach, *International Journal of Approximate Reasoning*, 2012, vol. 53, no. 1, pp. 54–65.

ИНФОРМАЦИЯ

Продолжается подписка на журнал "Программная инженерия" на второе полугодие 2020 г.

Оформить подписку можно в любом отделении Почты России,
через подписные агентства или непосредственно в редакции журнала.

Подписной индекс по Объединенному каталогу

"Пресса России" — 22765

Сообщаем, что с 2020 г. возможна подписка
на электронную версию нашего журнала через:

ООО "ИВИС": тел. (495) 777-65-57, 777-65-58; e-mail: sales@ivis.ru,
ООО "УП Урал-Пресс". Для оформления подписки (индекс 013312)
следует обратиться в филиал по месту жительства — <http://ural-press.ru>

Адрес редакции: 107076, Москва, Стромынский пер., д. 4,
Издательство "Новые технологии",
редакция журнала "Программная инженерия"

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

П. Н. Бибило, д-р техн. наук, проф., зав. лабораторией, bibilo@newman.bas-net.by, Объединенный институт проблем информатики Национальной академии наук Беларуси, Минск

Минимизация *BDDI*-представлений систем неполностью определенных булевых функций

Предложена модификация *BDD*-представлений систем неполностью определенных (частичных) булевых функций, названная *BDDI* (*Binary Decision Diagram with Inverse cofactors*), и алгоритмы минимизации *BDDI*, результатом работы которых являются многоуровневые представления систем полностью определенных функций в виде формул разложений Шеннона с использованием как одинаковых, так и взаимно инверсных подфункций. Минимизация числа взаимно инверсных частичных подфункций на одном уровне *BDDI*-представления системы частичных функций сведена к решению комбинаторной задачи нахождения кратчайшей имплицитной формы троичной матрицы, задающей значения подфункций этого уровня *BDDI*-представления.

Ключевые слова: булева функция, дизъюнктивная нормальная форма (ДНФ), разложение Шеннона, *Binary Decision Diagram* (*BDD*), раскраска графа, синтез логической схемы

Введение

Полностью определенные булевы функции являются математическими моделями комбинационных логических схем. В практике проектирования применяют также модели неполностью определенных (частичных) булевых функций в тех случаях, когда проектировщики задают требуемые двоичные значения выходных сигналов для заданного множества наборов значений входных сигналов, причем на остальных наборах (комбинациях значений входных сигналов) значения выходных сигналов являются неопределенными [1, 2]. Неопределенные значения в процессе проектирования доопределяются до полностью определенных (0,1) в целях получения лучших характеристик логических схем. Основными характеристиками логических схем являются сложность (площадь кристалла), быстродействие, энергопотребление [1]. В настоящее время схемная реализация систем частичных функций осуществляется по двум направлениям. Первое направление связано с минимизацией исходного представления систем частичных функций в классе дизъюнктивных нормальных форм (ДНФ), получением минимизированных ДНФ функций системы, являющихся *двухуровневыми* (И-ИЛИ) формами представления полностью определенных функций [2, 3], после чего проводится схемная реализация минимизированных представлений.

Второе направление связано с получением минимизированных *многоуровневых* представлений системы функций. Нахождение многоуровневых представлений осуществляется на основе решения различных задач декомпозиции (разложения) систем частичных функций [4–8]. Также как и при минимизации, в процессе разложения исходные

частичные функции доопределяются до полностью определенных, которые на втором этапе — этапе технологического отображения [1, 2, 9–12] — реализуются логическими схемами в целевой библиотеке проектирования (технологическом базисе логических элементов). Среди многочисленных видов функциональных разложений и соответствующих им форм представлений систем функций, по мнению автора, наиболее эффективными оказались многоуровневые представления систем функций на основе разложения Шеннона, называемые в литературе *BDD*-представлениями [11, 13–20]. Широкое применение *BDD*, т. е. ориентированных графов, задающих формулы разложений Шеннона, основывается на компактности графов и эффективности вычислительных процедур. Оно обусловлено тем обстоятельством, что *BDD*-представления булевых функций соответствуют ортогональным формам задания булевых функций.

Обширные вычислительные эксперименты, приведенные в работе [21], позволили сделать вывод о том, что логическая оптимизация *BDD*-представлений систем полностью определенных функций, в процессе которой ищутся *равные и взаимно инверсные подфункции*, позволяет получать более компактные *BDD*-представления, по сравнению с результатами поиска только одинаковых (равных) подфункций. Уменьшение сложности *BDD*-представлений при последующем синтезе логических схем приводит к меньшим по площади кристаллам заказных цифровых сверхбольших интегральных схем (СБИС).

В данной работе предложены алгоритмы логической минимизации *BDDI*-представлений систем частичных булевых функций. Результатом работы алгоритмов являются многоуровневые представления систем полностью определенных функций, задаваемые в виде формул разложений Шеннона

с использованием как одинаковых, так и взаимно инверсных подфункций. Показывается, что минимизация числа взаимно инверсных частичных подфункций *BDDI*-представлений связана с решением комбинаторных задач нахождения кратчайших имплицитующих форм троичных матриц.

1. Основные определения

Булевыми называются двоичные (0, 1) функции $f(\mathbf{x}) = f(x_1, x_2, \dots, x_n)$ двоичных (булевых) переменных x_1, x_2, \dots, x_n . Пусть V^x — булево пространство, построенное над переменными булева вектора $\mathbf{x} = (x_1, \dots, x_n)$. Элементами этого пространства являются n -компонентные наборы (векторы) \mathbf{x}^* нулей и единиц. Булева функция, значения 0, 1 которой определены на всех элементах $\mathbf{x}^* \in V^x$, называется *полностью определенной*. Множество M_f^1 элементов \mathbf{x}^* булева пространства, на которых полностью определенная булева функция принимает значение 1, называется *характеристическим множеством функции*. Если же на некоторых элементах булева пространства V^x значения функции не определены, то такая функция называется *неполностью определенной*, или *частичной*. Частичная булева функция принимает единичное значение на элементах \mathbf{x}^* подмножества M_f^1 булева пространства V^x и нулевое значение на элементах подмножества M_f^0 . На всех остальных элементах пространства V^x , образующих подмножество M_f^- пространства V^x , значения частичной функции не определены. Неопределенное значение функции обозначается символом "–". Очевидно, что $M_f^1 \cap M_f^0 = \emptyset$, $M_f^1 \cap M_f^- = \emptyset$, $M_f^0 \cap M_f^- = \emptyset$, $V^x = M_f^0 \cup M_f^1 \cup M_f^-$.

Частичные функции $f^1(\mathbf{x})$ и $f^2(\mathbf{x})$ равны, если и только если

$$M_{f^1}^1 = M_{f^2}^1, M_{f^1}^0 = M_{f^2}^0, M_{f^1}^- = M_{f^2}^- \quad (1)$$

Будем называть частичные функции $f^1(\mathbf{x})$ и $f^2(\mathbf{x})$ *взаимно инверсными* ($f^2 = \overline{f^1}$, $f^1 = \overline{f^2}$), если и только если

$$M_{f^1}^1 = M_{f^2}^0, M_{f^1}^0 = M_{f^2}^1, M_{f^1}^- = M_{f^2}^- \quad (2)$$

Согласно выражению (2) инверсией частичной функции является частичная функция, при этом взаимно инверсные частичные функции имеют одну и ту же область неопределенных значений.

Частичная функция f^i реализуется частичной функцией f^j (обозначается $f^i \prec f^j$), если и только если

$$M_{f^i}^1 \subseteq M_{f^j}^1, M_{f^i}^0 \subseteq M_{f^j}^0 \quad (3)$$

Для полностью определенных булевых функций отношение реализации является отношением равенства. Под векторной булевой функцией $f(\mathbf{x})$ будем понимать упорядоченную систему частичных булевых функций $f(\mathbf{x}) = (f^1(\mathbf{x}), \dots, f^m(\mathbf{x}))$, значениями векторных функций на элементах \mathbf{x}^* булева пространства являются m -компонентные троичные векторы $f(\mathbf{x}^*)$. Отношение реализации векторных функций сводится к выполнению отношения реализации между ком-

понентными функциями: если каждая компонентная функция $f^{j^*}(\mathbf{x})$, $j = 1, \dots, m$, векторной функции $f^*(\mathbf{x}) = (f^{1^*}(\mathbf{x}), \dots, f^{m^*}(\mathbf{x}))$ реализует соответствующую компонентную функцию f^j ($f^j \prec f^{j^*}$) векторной функции $f(\mathbf{x}) = (f^1(\mathbf{x}), \dots, f^m(\mathbf{x}))$, то тогда выполняется отношение реализации между векторными функциями: векторная функция $f^*(\mathbf{x})$ реализует векторную функцию $f(\mathbf{x})$: $f(\mathbf{x}) \prec f^*(\mathbf{x})$. В этом случае векторная функция $f^*(\mathbf{x})$ называется *доопределением* векторной функции $f(\mathbf{x})$.

2. Формы задания частичных векторных булевых функций

Интервальная форма задания векторной функции $f(\mathbf{x}) = (f^1(\mathbf{x}), \dots, f^m(\mathbf{x}))$ состоит из троичной матрицы T^x задания элементарных конъюнкций и троичной матрицы T^g вхождений конъюнкций в ДНФ, представляющих области нулевых и единичных значений компонентных функций.

Пример 1. Частичная векторная функция $g(\mathbf{x}) = (g^1(\mathbf{x}), \dots, g^2(\mathbf{x}))$, $n = 4$, $m = 2$, заданная в интервальной форме (табл. 1).

Неопределенное значение "–" элемента t_{ij} ($i = 1, \dots, m$; $j = 1, \dots, k$) матрицы T^g не означает, что функция g^i принимает неопределенное значение на наборах, которые порождает троичный вектор строки i матрицы T^x . Например, на пересечении девятой строки и первого столбца матрицы T^g (табл. 1) находится "–", однако это не означает, что на четырех наборах, образующих интервал (троичный вектор) (0 0 – –) функция g^1 не определена. Рассмотрев пятую строку матриц T^x и T^g , можно убедиться, что на наборах (0 0 1 0), (0 0 1 1) функция g^1 принимает единичное значение. Если же все строки матрицы T^x являются попарно ортогональными, то в строках матрицы T^g задаются значения 0, 1, "–" компонентных частичных функций g^i . Троичные векторы, задающие элементарные конъюнкции из левой части табл. 1, называются также *интервалами* булева пространства V^x [2, с. 148]. Заметим, что часто в матрицу T^g не вносятся строки, на которых все компонентные функции имеют неопределенные значения (в табл. 1 это девятая и десятая строки). Это

Таблица 1

Интервальная форма частичной векторной функции

№ строки	T^x	T^g
	$x_1 x_2 x_3 x_4$	$g^1 g^2 g^3 g^4$
1	0 0 – –	0 0 – –
2	0 1 0 1	0 0 0 1
3	1 – 0 –	0 – 0 1
4	– 1 1 0	1 – – –
5	– 0 1 –	1 – – –
6	– 1 1 1	1 0 – –
7	0 0 1 –	– 1 1 –
8	1 0 1 –	– 1 – 0
9	0 0 – –	– – – –
10	0 1 0 0	– – – –

связано с тем, что в практических примерах большой размерности задаются только области единичных и нулевых значений функций. В данном примере все как раз наоборот — общую область неопределенных значений для всех компонентных функций составляет только один (десятый) набор (0100) из левой части табл. 1.

Частичная функция g^1 (см. табл. 1) может быть задана парой ДНФ $D_{g^1}^1, D_{g^1}^0$: множество $M_{g^1}^1$ задается в виде ДНФ $D_{g^1}^1 = x_2x_3\bar{x}_4 \vee \bar{x}_1x_2 \vee x_2x_3x_4$; множество $M_{g^1}^0$ — в виде ДНФ $D_{g^1}^0 = \bar{x}_1\bar{x}_2\bar{x}_3 \vee \bar{x}_1x_2\bar{x}_3x_4 \vee x_1\bar{x}_3$. Область неопределенных значений — множество M_{f^1} может быть задано в виде ДНФ $D_{g^1}^1 = D_{g^1}^1 \vee D_{g^1}^0 = \bar{x}_1x_2\bar{x}_3\bar{x}_4$. Аналогично может быть задана функция g^2 .

Ортогонализованная форма системы частичных булевых функций. Будем говорить, что троичные векторы $\mathbf{a} = (a_1, \dots, a_m), \mathbf{b} = (b_1, \dots, b_m)$ ортогональны, если найдется хотя бы одна пара одноименных компонент a_i, b_i такая, что значения их определены и не равны. Например, троичные векторы $\mathbf{a} = (0 - 10), \mathbf{b} = (-100)$ ортогональны, так как для $i = 3$ выполняется условие ортогональности: $a_3 = 1, b_3 = 0$. Если в матричном задании в троичной матрице T^x все строки (троичные векторы) ортогональны, то такая матричная форма представления векторной булевой функции называется *ортогонализованной*. Алгоритмы получения ортогонализированных форм систем функций по исходным интервальным формам известны в литературе [22].

3. Проверка отношения реализуемости булевых функций

При задании систем функций в интервальной форме следует уточнить проверку отношения реализации. Пусть заданы две частичные функции: функция f^i задана парой ортогональных ДНФ $D_{f^i}^0, D_{f^i}^1$, а функция f^j — парой ортогональных ДНФ $D_{f^j}^0, D_{f^j}^1$. ДНФ называются ортогональными, если их произведение (конъюнкция) равно нулю. Проверка отношения $f^i < f^j$, т.е. проверка $M_{f^i}^1 \subseteq M_{f^j}^1, M_{f^i}^0 \subseteq M_{f^j}^0$ сводится к проверке выполнения импликаций $D_{f^i}^0 \rightarrow D_{f^j}^0, D_{f^i}^1 \rightarrow D_{f^j}^1$, что эквивалентно одновременному выполнению двух условий $D_{f^i}^0 \bar{D}_{f^j}^0 = 0, D_{f^i}^1 \bar{D}_{f^j}^1 = 0$. Алгоритмы перемножения и инверсирования матричных форм ДНФ описаны в работе [23, с. 34], структуры данных — в работе [24]. В работе [21] приведены алгоритмы и примеры перехода от ДНФ к полиномам Жегалкина, для которых легко выполняются логические операции инверсирования и перемножения булевых функций. Проверка условий $D_{f^i}^0 \bar{D}_{f^j}^0 = 0, D_{f^i}^1 \bar{D}_{f^j}^1 = 0$ может быть также сведена к задаче "выполнимость" (SAT-problem [25, 26]), если перевести ДНФ в конъюнктивную нормальную форму (КНФ). Покажем это на примере. Пусть $D_{f^i}^0 = \bar{x}_1\bar{x}_2x_3 \vee x_1x_2, D_{f^i}^1 = x_1 \vee \bar{x}_2x_3\bar{x}_4$. Проверим равенство $D_{f^i}^0 \bar{D}_{f^j}^0 = 0$. Чтобы перевести ДНФ $D_{f^i}^0$ в КНФ, выполним операцию отрицания $D_{f^i}^0$, а затем к полученной ДНФ $\bar{D}_{f^i}^0$ применим принцип двойственности [2]:

$$\bar{D}_{f^i}^0 = (x_1 \vee x_2 \vee \bar{x}_3)(\bar{x}_1 \vee \bar{x}_2) = \bar{x}_1x_2 \vee \bar{x}_1\bar{x}_3 \vee \bar{x}_2\bar{x}_3 \vee x_1\bar{x}_2.$$

КНФ, полученная по $\bar{D}_{f^i}^0$, имеет вид $(x_1 \vee \bar{x}_2)(x_1 \vee x_3)(x_2 \vee x_3)(\bar{x}_1 \vee x_2)$. Отрицание $\bar{D}_{f^i}^0$ ДНФ $D_{f^i}^0$ имеет вид КНФ: $\bar{D}_{f^i}^0 = \bar{x}_1(x_2 \vee \bar{x}_3 \vee x_4)$. Таким образом, проверка $D_{f^i}^0 \bar{D}_{f^j}^0 = 0$ сводится к проверке выполнимости КНФ

$$D_{f^i}^0 \bar{D}_{f^j}^0 = (x_1 \vee \bar{x}_2)(x_1 \vee x_3)(x_2 \vee x_3) \times (\bar{x}_1 \vee x_2)(\bar{x}_1)(x_2 \vee \bar{x}_3 \vee x_4).$$

Выполняющим набором для КНФ, представленной в правой части последнего равенства, является набор 0011, что будет свидетельствовать о том, что $D_{f^i}^0 \bar{D}_{f^j}^0 \neq 0$, т.е. набор 0011 является элементом множества $M_{f^i}^0$ и не является элементом множества $M_{f^j}^0$, именно на наборе 0011 нарушается импликация $D_{f^i}^0 \rightarrow D_{f^j}^0$.

Проверка отношения реализации становится гораздо более простой, если области значений частичных функций представлены не произвольными ДНФ, а соответствующими ортогонализированными формами — таблицами истинности, либо ортогонализированными ДНФ.

4. Проверка равенства двух частичных функций

Пусть заданы две частичные функции: функция f^i задана тройкой попарно ортогональных ДНФ $D_{f^i}^0, D_{f^i}^1, D_{f^i}^{-1}$, а функция f^j — тройкой попарно ортогональных ДНФ $D_{f^j}^0, D_{f^j}^1, D_{f^j}^{-1}$. Частичные функции f^i, f^j равны, если равны две (из трех) соответствующих ДНФ, например, если $D_{f^i}^0 = D_{f^j}^0$ и $D_{f^i}^1 = D_{f^j}^1$. Равенство $D_{f^i}^0 = D_{f^j}^0$ требует выполнения условий

$$D_{f^i}^0 (\bar{D}_{f^j}^0 \vee D_{f^j}^{-1}) = 0, D_{f^i}^0 (\bar{D}_{f^j}^0 \vee D_{f^j}^{-1}) = 0. \quad (4)$$

Равенство $D_{f^i}^1 = D_{f^j}^1$ требует выполнения условий

$$D_{f^i}^1 (\bar{D}_{f^j}^1 \vee D_{f^j}^{-1}) = 0, D_{f^i}^1 (\bar{D}_{f^j}^1 \vee D_{f^j}^{-1}) = 0. \quad (5)$$

Легко видеть, что проверки условий (4), (5) также могут быть сведены к задаче "выполнимость".

5. Обобщенно-ортогонализованная форма частичной векторной булевой функции

Более компактной по сравнению с ортогонализованной является *обобщенно-ортогонализованная* форма, когда векторная функция задается на попарно ортогональных ДНФ, а не на попарно ортогональных интервалах (конъюнкциях), как это осуществляется в ортогонализированных формах. Данная форма представления более компактная и удобная для решения задач, возникающих при доопределении частичных функций с целью сокращения сложности BDD.

Нахождение обобщенно-ортогонализованной формы сводится к задаче нахождения минимального дизъюнктивного базиса для частичной векторной

Задание областей значений компонентных функций из примера 1

Частичная функция	$x_1 x_2 x_3 x_4$	Значение	Область значений	Функция λ
g^1	0 0 0 – 0 1 0 1 1 – 0 –	0 0 0	$M_{g^1}^0$	λ_1^0
	– 1 1 0 – 0 1 – – 1 1 1	1 1 1	$M_{g^1}^1$	λ_1^1
	0 1 0 0	–	$M_{g^1}^-$	λ_1^-
g^2	0 0 0 – 0 1 0 1 – 1 1 1	0 0 0	$M_{g^2}^0$	λ_2^0
	0 0 1 – 1 0 1 –	1 1	$M_{g^2}^1$	λ_2^1
	1 – 0 – – 1 1 0 0 1 0 0	– – –	$M_{g^2}^-$	λ_2^-
g^3	0 1 0 1 1 – 0 –	0 0	$M_{g^3}^0$	λ_3^0
	0 0 1 –	1	$M_{g^3}^1$	λ_3^1
	0 0 0 – 1 0 1 – – 1 1 1 0 1 0 0 1 1 1 0	– – – – –	$M_{g^3}^-$	λ_3^-
	1 0 1 – 0 1 0 1 1 – 0 –	0 1 1	$M_{g^4}^0$ $M_{g^4}^1$	λ_4^0 λ_4^1
g^4	0 0 – – – 1 1 – 0 1 0 0	– – –	$M_{g^4}^-$	λ_4^-

булевой функции $f(\mathbf{x}) = (f^1(\mathbf{x}), \dots, f^m(\mathbf{x}))$. Зададим каждую компонентную частичную булеву функцию $f^i(\mathbf{x})$ тремя полностью определенными булевыми функциями $\lambda_i^0(\mathbf{x})$, $\lambda_i^1(\mathbf{x})$, $\lambda_i^-(\mathbf{x})$ такими, что функция $\lambda_i^0(\mathbf{x})$ имеет характеристическое множество $M_{f^i}^0$, функция $\lambda_i^1(\mathbf{x})$ — характеристическое множество $M_{f^i}^1$, функция $\lambda_i^-(\mathbf{x})$ — характеристическое множество $M_{f^i}^-$.

Минимальным дизъюнктивным базисом для $f(\mathbf{x}) = (f^1(\mathbf{x}), \dots, f^m(\mathbf{x}))$ называется минимальная по мощности система попарно ортогональных полностью определенных булевых функций $\Pi = \{\Pi^1(\mathbf{x}), \dots, \Pi^k(\mathbf{x})\}$ такая, что каждая полностью определенная функция $\lambda_i^j(\mathbf{x})$, $\lambda_i^0(\mathbf{x})$, $\lambda_i^-(\mathbf{x})$ равна дизъюнкции некоторого подмножества функций системы Π .

Частичная булева функция $f^i(\mathbf{x})$ задает трехблочное разбиение $R^i = \{M_{f^i}^0, M_{f^i}^1, M_{f^i}^-\}$ булева пространства V^x на попарно непересекающиеся подмножества. Нахождение для $f(\mathbf{x}) = (f^1(\mathbf{x}), \dots, f^m(\mathbf{x}))$ минимального дизъюнктивного базиса (в теоретико-множественной интерпретации) может быть сведено к нахождению произведения разбиений R^i , $i = 1, \dots, m$, и составлению из непустых блоков полученного произведения разбиений каждого из исходных блоков $M_{f^i}^0$, $M_{f^i}^1$, $M_{f^i}^-$ [27, с. 12].

Пусть каждая компонентная функция f^i задана тройкой попарно ортогональных ДНФ $D_{f^i}^0$, $D_{f^i}^1$, $D_{f^i}^-$, данные ДНФ представляют функции $\lambda_i^0(\mathbf{x})$, $\lambda_i^1(\mathbf{x})$, $\lambda_i^-(\mathbf{x})$, соответственно. В этом случае нахождение минимального дизъюнктивного базиса Π может быть сведено к выполнению операций перемножения, инверсирования и сравнения на равенство ДНФ, алгоритм описан в работе [28, с. 124]. Он оперирует исходными данными для случая, когда каждая из функций задана элементарной конъюнкцией. Алгоритм пригоден и для случая задания функций $\lambda_i^j(\mathbf{x})$, $\lambda_i^0(\mathbf{x})$, $\lambda_i^-(\mathbf{x})$ в виде ДНФ, он программно реализован и использован при решении задачи декомпозиции системы булевых функций [29]. Примеры разбиения булева пространства на непересекающиеся подмножества (характеристические множества попарно ортогональных базисных функций $\Pi^1(\mathbf{x}), \dots, \Pi^k(\mathbf{x})$) показаны в работах [28, с. 120] и [20, с. 135].

Пусть векторная функция $\mathbf{g}(\mathbf{x}) = (g^1(\mathbf{x}), \dots, g^4(\mathbf{x}))$, $\mathbf{x} = (x_1, x_2, x_3, x_4)$ задана в табл. 2 ДНФ, представляющими области значений компонентных функций.

В табл. 3 представлены попарно ортогональные функции Π^i минимального (по мощности) дизъюнктивного базиса для $g^1(\mathbf{x}), \dots, g^4(\mathbf{x})$ (см. табл. 1). Дизъюнкция некоторых функций Π^i , представленных в табл. 4 матричными формами ДНФ, позволяет получить ДНФ $D_{g^i}^0$, $D_{g^i}^1$, $D_{g^i}^-$, задающие области значений каждой из компонентных функций g^i частичной векторной функции $\mathbf{g}(\mathbf{x}) = (g^1(\mathbf{x}), \dots, g^4(\mathbf{x}))$:

$$D_{g^1}^0 = \Pi^1 \vee \Pi^2 \vee \Pi^3;$$

$$D_{g^1}^1 = \Pi^4 \vee \Pi^5 \vee \Pi^6 \vee \Pi^7; D_{g^1}^- = \Pi^8;$$

$$D_{g^2}^0 = \Pi^1 \vee \Pi^2 \vee \Pi^7; D_{g^2}^1 = \Pi^5 \vee \Pi^6;$$

$$D_{g^2}^- = \Pi^3 \vee \Pi^4 \vee \Pi^8;$$

$$D_{g^3}^0 = \Pi^2 \vee \Pi^3 \vee \Pi^7; D_{g^3}^1 = \Pi^5;$$

$$D_{g^3}^- = \Pi^1 \vee \Pi^4 \vee \Pi^6 \vee \Pi^7 \vee \Pi^8;$$

$$D_{g^4}^0 = \Pi^6; D_{g^4}^1 = \Pi^2 \vee \Pi^3;$$

$$D_{g^4}^- = \Pi^1 \vee \Pi^4 \vee \Pi^5 \vee \Pi^7 \vee \Pi^8.$$

Каждая из попарно ортогональных базисных ДНФ представлена одной элементарной конъюнкцией, поэтому обобщенно-ортогонализованная форма может вырождаться (редуцироваться) до ортогонализованной формы. В табл. 3 троичные векторы в правой колонке задают одинаковые значения компонентных функций g^1 , g^2 , g^3 , g^4 на булевых векторах, входя-

Таблица 3

Задание функций на ДНФ минимального дизъюнктивного базиса

Функция Π^i	$x_1 x_2 x_3 x_4$	$g^1 g^2 g^3 g^4$
Π^1	0 0 0 –	0 0 – –
Π^2	0 1 0 1	0 0 0 1
Π^3	1 – 0 –	0 – 0 1
Π^4	– 1 1 0	1 – – –
Π^5	0 0 1 –	1 1 1 –
Π^6	1 0 1 –	1 1 – 0
Π^7	– 1 1 1	1 0 – –
Π^8	0 1 0 0	– – – –

щих в характеристические множества функций Π^i , $i = 1, \dots, 8$. Например, характеристическое множество $M_{\Pi^6}^1 = \{1010, 1011\}$, вектор значений компонентных функций g^1, g^2, g^3, g^4 на каждом из наборов 1010, 1011 равен (11-0). Легко видеть, что наборы 1010, 1011 входят в характеристические множества функций $\lambda_1^1, \lambda_2^1, \lambda_3^1, \lambda_4^1$.

6. Нахождение минимального числа реализующих функций для системы частичных функций

Задача 1. Для заданной неупорядоченной системы F частичных булевых функций требуется найти минимальную по мощности подсистему $H \subseteq F$ функций таких, что каждая из функций исходной системы F реализуется хотя бы одной из функций подсистемы H .

Функции f^i, f^j являются *несовместимыми* (не доопределяемыми до одной функции) тогда и только тогда, когда выполняется хотя бы одно из условий

$$M_{f^i}^0 \cap M_{f^j}^1 \neq \emptyset, \quad (6)$$

$$M_{f^j}^0 \cap M_{f^i}^1 \neq \emptyset. \quad (7)$$

Если функции f^i, f^j заданы в ДНФ: f^i задана парой ортогональных ДНФ $D_{f^i}^0, D_{f^i}^1$, а функция f^j – парой попарно ортогональных ДНФ $D_{f^j}^0, D_{f^j}^1$, то условия (6), (7) эквивалентны следующим:

$$D_{f^i}^0 D_{f^j}^1 \neq 0, \quad (8)$$

$$D_{f^j}^0 D_{f^i}^1 \neq 0. \quad (9)$$

Зададим значения функций примера 1 троичной матрицей T (табл. 4). Решение задачи 1 сводится к построению неориентированного графа G отношения несовместимости (ортогональности) строк матрицы T и раскраске графа в минимальное число цветов. Каждой вершине графа G соответствует частичная функция системы F . Две вершины графа G смежны тогда и только тогда, когда соответствующие данным

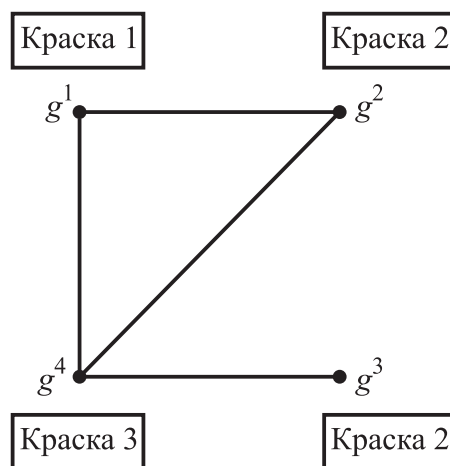


Рис. 1. Граф отношения несовместимости функций

Таблица 4

Задание значений функций троичной матрицей

	Π^1	Π^2	Π^3	Π^4	Π^5	Π^6	Π^7	Π^8	
$T =$	0	0	0	1	1	1	1	–	g^1
	0	0	–	–	1	1	0	–	g^2
	–	0	0	–	1	–	–	–	g^3
	–	1	1	–	–	0	–	–	g^4

Таблица 5

Реализующие функции, соответствующие раскраске графа (рис. 1)

Π^1	Π^2	Π^3	Π^4	Π^5	Π^6	Π^7	Π^8	Функция
0	0	0	1	1	1	1	–	δ^1 (краска 1)
0	0	0	–	1	1	0	–	δ^2 (краска 2)
–	1	1	–	–	0	–	–	δ^3 (краска 3)

вершинам функции системы F являются несовместимыми. Постановка задачи раскраски неориентированного графа и эффективный эвристический алгоритм ее решения даны в работе [2, с. 47].

Граф отношения несовместимости и его раскраска в минимальное число цветов (красок) изображен на рис. 1. Например, вершины g^1 и g^2 графа G смежны (соединены ребром), так как $D_{g^1}^1 D_{g^2}^0 = \Pi^7 \neq 0$.

Каждой краске соответствует своя частичная функция подсистемы H (табл. 5). Подсистема H включает три функции $\delta^1, \delta^2, \delta^3$. Данные функции реализуют исходные: $g^1 \prec \delta^1, g^2 \prec \delta^2, g^3 \prec \delta^3, g^4 \prec \delta^4$.

7. Нахождение минимального числа взаимно инверсных функций

Задача 2. Для заданной системы F частичных булевых функций требуется найти минимальную по мощности подсистему $H \subseteq F$ функций таких, что

каждая из функций исходной системы F реализуется хотя бы одной из функций (либо инверсией хотя бы одной из функций) подсистемы H .

Решение задачи 2 сводится к решению задачи нахождения кратчайшей имплицитной формы троичной матрицы T , задающей значения системы F :

- на наборах (правая часть таблицы истинности);
- на ортогональных интервалах (ортогонализованная форма);
- на ДНФ минимального дизъюнктивного базиса (при представлении системы функций обобщенно-ортогонализованной формой).

Введем требуемые определения. Троичный вектор \mathbf{a} имплицитует троичный вектор \mathbf{b} , если \mathbf{b} получается из \mathbf{a} заменой некоторых нулей или единиц значением "–" и, возможно, инвертированием полученного результата [2, с. 499]. Например, вектор (1 0 – – 1 0 1) имплицитует каждый из векторов (1 0 – – – 0 1) и (0 1 – – – 1 –). Будем говорить, что троичная матрица R имплицитует троичную матрицу T , если для каждой строки матрицы T в матрице R найдется имплицитующая ее строка. В работе [2, с. 499] описан точный метод нахождения кратчайшей (содержащей минимальное число строк) имплицитующей формы троичной матрицы. Метод основывается на нахождении всех максимальных i -совместимых подмножеств строк матрицы T , после чего требуется решить задачу кратчайшего покрытия — получить кратчайшее покрытие строк матрицы T этими множествами. Множество строк матрицы T называется i -совместимым, если существует вектор, имплицитующий каждую строку этого множества. *Максимальным* называется i -совместимое множество, если оно не является собственным подмножеством другого i -совместимого множества.

При решении практических задач, когда число аргументов и функций достигает многих десятков и сотен, для решения задачи нахождения кратчайшей имплицитующей формы используются эвристические алгоритмы, позволяющие находить приближенное решение. Эвристический алгоритм решения задачи 2 основан на итеративном выполнении шагов 1–3 алгоритма, которые позволяют сформировать очередную строку r^i матрицы R , по которой затем формируется очередная функция системы H .

Шаг 1. Выбрать строку q матрицы T , которая имеет максимальное число определенных (0,1) элементов, занести ее в подмножество R^i . Исключить строку q из матрицы T .

Шаг 2. Среди оставшихся строк матрицы T найти строку p , имеющую максимальное число определенных элементов, которая будет образовывать i -совместимое множество $R^i = \{q, p\}$. Исключить строку p из матрицы T .

Шаг 3. Добавлять (аналогично шагу 2) в множество R^i строки (и исключать их из матрицы T) до тех пор, пока множество R^i является i -совместимым. Если среди оставшихся строк матрицы T нельзя выбрать ни одной строки для R^i , то сформировать строку r^i матрицы R .

Шаг 4. Шаги 1–3 выполнять до тех пор, пока из матрицы T не будут исключены все строки.

Выбор формы представления системы F обусловлен числом аргументов функций системы F : если

Реализующие функции, построенные по кратчайшей имплицитующей форме для троичной матрицы (табл. 4)

Π^1	Π^2	Π^3	Π^4	Π^5	Π^6	Π^7	Π^8	
0	0	0	1	1	1	1	–	φ^1
0	0	0	–	1	1	0	–	φ^2

число аргументов не превышает десятка, то можно использовать таблицы истинности, в случае большего числа аргументов требуются более компактные (ортогонализованные) формы.

Рассмотрим решение задачи 2 для примера 1. Троичная матрица T , задающая значения функций на ДНФ дизъюнктивного базиса, дана в табл. 4. Решение — кратчайшая имплицитующая форма — дана в табл. 6, каждой строке кратчайшей имплицитующей формы соответствует частичная функция, заданная на ДНФ дизъюнктивного базиса: первой строке соответствует частичная функция φ^1 , второй — функция φ^2 . Данные функции реализуют исходные: $g^1 < \varphi^1$, $g^2 < \varphi^2$, $g^2 < \varphi^2$, $g^4 < \bar{\varphi}^2$.

8. Представление частичной векторной функции в виде BDD

Разложением Шеннона полностью определенной либо частичной булевой функции $f = f(\mathbf{x})$, $\mathbf{x} = (x_1, x_2, \dots, x_n)$, по переменной x_i называется представление

$$f = f(\mathbf{x}) = \bar{x}_i f_0(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n) \vee x_i f_1(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n). \quad (10)$$

Функции $f = f_0(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n)$, $f_1 = f_1(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n)$ в правой части (10) называются коэффициентами разложения по переменной x_i , кофакторами (cofactors), остаточными подфункциями, либо просто *подфункциями*. Они получаются из функции $f = f(x_1, \dots, x_n)$ подстановкой вместо переменной x_i константы 0 и 1 соответственно. Каждая из подфункций f_0 и f_1 может быть разложена по одной из переменных из множества $\{x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n\}$. Процесс разложения подфункций заканчивается, когда все n переменных будут использованы для разложения, либо когда все подфункции вырождаются до констант 0, 1, "–". На каждом шаге разложения выполняется сравнение на равенство полученных подфункций и оставляется одна из нескольких попарно равных подфункций.

Под BDD частичной векторной булевой функции $f(\mathbf{x}) = (f^1(\mathbf{x}), \dots, f^m(\mathbf{x}))$ понимается ориентированный ациклический граф, задающий последовательные разложения Шеннона всех компонентных функций $f^i(\mathbf{x})$, $i = 1, \dots, m$, по всем ее переменным x_1, \dots, x_n при одном и том же заданном порядке (перестановке) переменных, по которому проводятся разложения.

Наиболее близкими к графам BDD для частичных векторных функций являются широко извест-

ные в литературе сокращенные упорядоченные *BDD* (*Reduced Ordered BDD, ROBDD*) для одной полностью определенной булевой функции, в которых каждой функциональной вершине соответствует одна функция (подфункция разложения Шеннона), при этом функциональные вершины лишь подразумеваются (отождествляются с вершинами-переменными). Подробное описание *OBDD* (упорядоченных *BDD*) дано в работах [16, 17], *ROBDD* — в работе [18]. Далее под *BDD* будем понимать *ROBDD* для систем функций (векторных функций).

Будем изображать на рисунках графы *BDD*, содержащие три вида вершин: функциональные вершины, соответствующие разлагаемым функциям (и их инверсиям); вершины-переменные; листовые вершины, соответствующие константам 0, 1 и — [20, с. 16].

Замечание. Рисунки и структуры данных классических *BDD* не содержат функциональных вершин. Функциональные вершины будем показывать на рисунках для удобства изложения методов минимизации *BDD* и иллюстрации зависимостей переменных в формулах разложений Шеннона, в компьютерных структурах данных функциональные вершины не указывают [20, с. 177].

Пример 2. Построим граф *BDD* для векторной частичной функции, заданной таблицей истинности (табл. 7), по перестановке $\langle x_1, x_2, x_3, x_4 \rangle$. Алгоритм построения *BDD* описан в работе [20, с. 56] и сводится к нахождению одинаковых подфункций $s^i, i = 1, \dots, 16$, полученных в результате проведения разложения Шеннона исходных компонентных функций, либо подфункций, получен-

ных на очередном шаге разложения. Полученный граф *BDD* для векторной функции из табл. 7 показан на рис. 2, а. На рис. 2, б показан граф *BDD*, по которому строятся структуры данных для компьютерных программ.

Таблица 7

Таблица истинности

$x_1 x_2 x_3 x_4$	$f^1 f^2$
0 0 0 0	1 —
0 0 0 1	0 1
0 0 1 0	— 0
0 0 1 1	1 1
0 1 0 0	— 1
0 1 0 1	1 —
0 1 1 0	1 0
0 1 1 1	0 0
1 0 0 0	0 —
1 0 0 1	— —
1 0 1 0	1 —
1 0 1 1	0 1
1 1 0 0	— 0
1 1 0 1	— 1
1 1 1 0	— 1
1 1 1 1	1 —

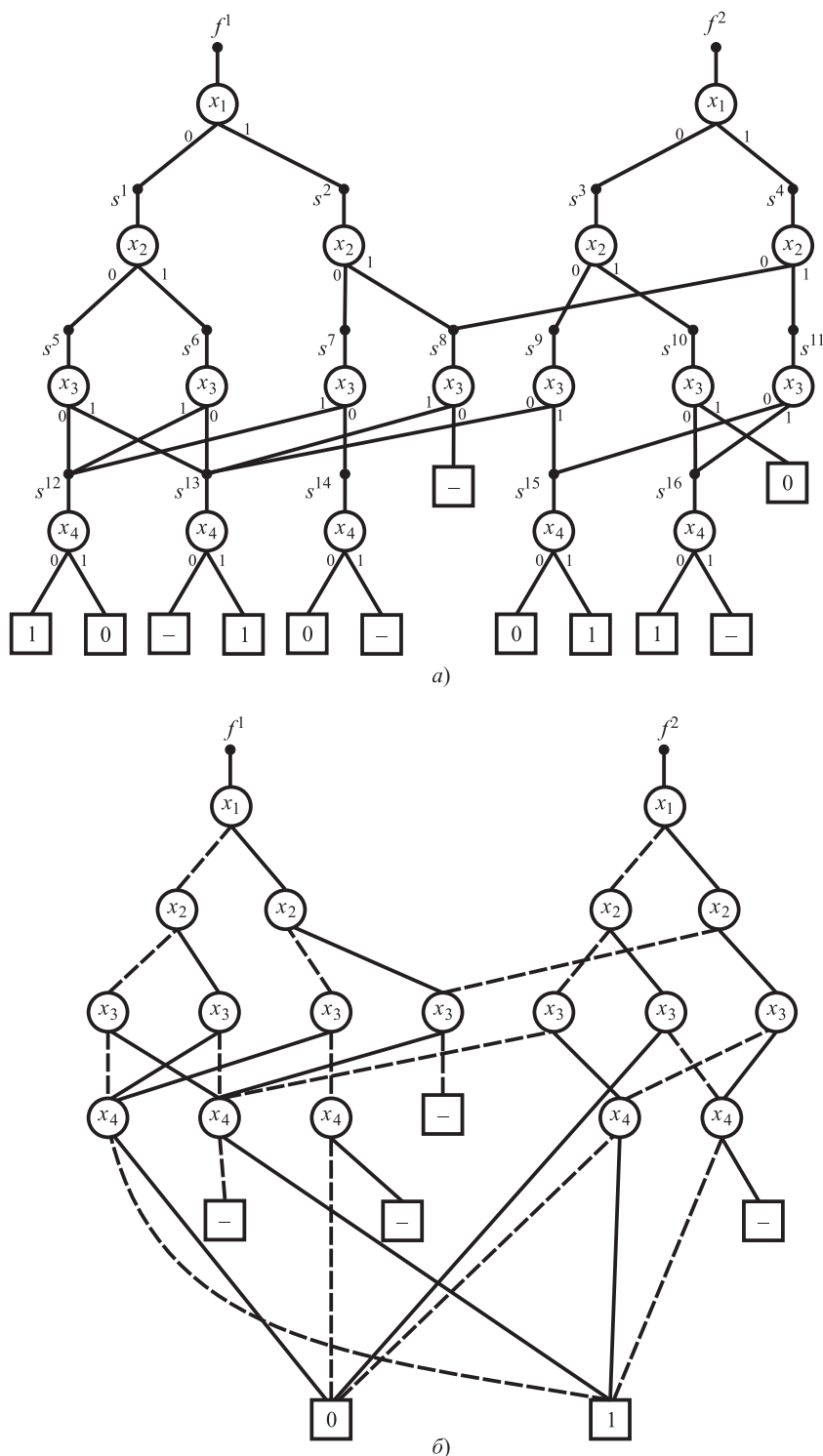


Рис. 2. Граф *BDD* частичной векторной функции из примера 2:

а — граф с функциональными вершинами; б — граф без функциональных вершин

BDD задают в компактном виде ортогонализированные формы функций: для этого достаточно рассмотреть пути из корневых вершин, ведущие в листовые вершины 0, 1, —. Пути, ведущие из корневых вершин в листовые вершины "—", задают интервалы булева пространства, на которых соответствующая функция принимает неопределенное значение. Например, на наборах (0001), (0111), (1000), (1011) функция f^1 принимает нулевое значение, а на интервале (110—) эта же функция не определена (принимает значение "—"). В работе [20, с. 94] показано, что логические операции над *BDD* полностью определенных компонентных функций векторной функции могут быть обобщены на случай *BDD*, представляющих частичные компонентные булевы функции.

BDD является более компактной формой по сравнению с ортогонализированными формами, так задается граф вместо перечисления всех путей (из корневых вершин в листовые) в этом графе. Задание графа более компактно, чем перечисление всех путей в нем. Поэтому *BDD* и нашли очень широкое применение не только при проектировании цифровых (логических) схем, но и в информатике в целом [18], [19, с. 295].

Существенным отличием *BDD*, представляющих частичные функции, от полностью определенных *BDD* является то, что все листовые вершины "—" рассматриваются как различные, в отличие от вершин-констант 0, 1. Каждая вершина "—" помещается на свой уровень *BDD* и отдельно доопределяется впоследствии до 0 либо 1 (рис. 2, б). Если же отождествить все различные листовые вершины "—" одной вершине "—" (как это сделано для констант 0, 1), то задача выбора доопределения до нуля либо единицы различных листовых вершин "—", по сути, не возникает. Доопределив, например, до 0, такую "глобальную" листовую вершину "—", требуется выполнить процедуру редукции (сокращения) *BDD*, что не является трудоемкой задачей.

Рассмотренные в работе [20, с. 63] эвристические алгоритмы минимизации *BDD*-представлений частичных векторных функций основывались на решении задачи 1 раскраски графа несовместимости для подфункций, получаемых в процессе разложения Шеннона по общей перестановке переменных.

9. Представление векторной функции в виде *BDDI*

Если при разложении Шеннона по какой-либо из переменных находить взаимно инверсные подфункции и оставлять одну из них, то число различных подфункций можно сократить — вместо двух взаимно инверсных подфункций останется одна из них. На таком виде оптимизации и основывается введенное в работе [21] понятие *BDDI* (*Binary Decision Diagram with Inverse cofactors*) для полностью определенной векторной булевой функции. Под *BDDI* частичной векторной булевой функции $f(x) = (f^1(x), \dots, f^m(x))$ понимается ориентированный ациклический граф, задающий последовательные разложения Шеннона всех компонентных функций $f^i(x)$, $i = 1, \dots, m$, по всем ее переменным x_1, \dots, x_n при одном и том же задан-



Рис. 3. Функциональная вершина *BDDI*

ном порядке (перестановке) переменных, по которым проводятся разложения, при этом функциональная вершина *BDDI* (рис. 3) реализует одну подфункцию h , либо две подфункции — h и ее инверсию \bar{h} .

BDD либо *BDDI* назовем полностью определенной, если она представляет систему полностью определенных булевых функций, такие представления не содержат листовых вершин "—". Именно по таким *BDD* либо *BDDI*, вернее говоря, по соответствующим взаимосвязанным логическим уравнениям вида (10), осуществляется последующий синтез логической схемы. Представление многоуровневого разложения системы булевых функций в виде графа полностью определенной *BDD* либо *BDDI* в данной статье используется для иллюстрации зависимостей булевых переменных в логических выражениях. Целью является получение возможно меньшего (минимального) числа логических выражений вида (10), по которым впоследствии осуществляется синтез логической схемы в заданном технологическом базисе (библиотеке) логических элементов.

Функциональные вершины нижнего (неконстантного) уровня *BDD* (*BDDI*) соответствуют функции от одной переменной. В уравнениях вида (10) такие функции не содержат логических операторов дизъюнкции и конъюнкции, поэтому такие уравнения (вершины) не будут учитываться и при подсчете сложности многоуровневого представления, соответствующего *BDDI*. Сложность S_{BDDI} полностью определенной *BDDI* — число ее функциональных вершин без учета функциональных вершин-литералов. Итак, сложность *BDDI* (либо *BDD*) определяется по числу логических уравнений, которые содержат хотя бы один оператор дизъюнкции либо конъюнкции.

Переход от полностью определенных *BDDI* к *BDD* рассмотрен в работе [21] и заключается в "расщеплении" каждой функциональной вершины, содержащей инверсный (светлый) кружок на две функциональные вершины, и замене на противоположные значения всех листовых вершин, на которые опирается данная функциональная вершина. При этом может понадобиться выполнение процедуры сокращения *BDD*, так как могут появиться одинаковые функциональные вершины.

Задача 3. Задана матричная форма частичной векторной булевой функции $f(x)$ и перестановка $\langle x_{j_1}, x_{j_2}, \dots, x_{j_n} \rangle$ переменных для построения многоуровневого представления. Требуется для заданной перестановки переменных получить полностью определенную *BDDI*, реализующую исходную векторную

булеву функцию, при этом *BDDI* должна быть возможно меньшей сложности.

Предлагаемый эвристический алгоритм 1 решения задачи 3 является итерационным — число итераций не превышает число n переменных x_i векторной функции. Алгоритм выполняется до тех пор, пока текущая система функций (подфункций — коэффициентов разложений) не выродится до констант 0, 1, "—". На каждой i -й итерации ($i = 1, 2, \dots, n$) для очередной переменной x_j , $i = 1, \dots, n$, перестановки $\langle x_{j_1}, x_{j_2}, \dots, x_{j_n} \rangle$ выполняются шаги 1–3. После выполнения всех итераций выполняется переход на шаг 4 и проводится дополнительное сокращение *BDDI* (выполняется шаг 5) и запись уравнений и их возможное упрощение (шаги 6 и 7).

Шаг 1. Построение коэффициентов разложения (частичных функций) по переменной x_j каждой из функций системы, построенных на шаге $i - 1$. На первой итерации на первом шаге ($i = 1$) в качестве такой системы выступает система компонентных функций, составляющих исходную векторную функцию.

Шаг 2. Нахождение для построенных коэффициентов минимального по мощности множества H частичных функций, к которым доопределяются (возможно, после инверсирования) все коэффициенты (решение задачи 2).

Замечание. Задача нахождения минимального числа реализующих функций сводится к построению графа G отношения несовместимости частичных функций и раскраске его в минимальное число цветов — это задача 1.

Шаг 3. Построение i -го яруса графа *BDD*, число функциональных вершин *BDDI* в i -м ярусе соответствует мощности множества H . Множество H составляет исходную систему функций для следующей итерации $i + 1$. Переход на шаг 1.

Шаг 4. Конец итераций проведения разложений Шеннона по одной переменной. Переход на шаг 5.

Шаг 5. Дополнительное сокращение *BDDI* выполняется в том случае, если имеются листовые вершины, помеченные символом "—" неопределенного значения. На данном шаге рассматривается функциональный уровень *BDDI*, на котором располагаются частичные функции, зависящие от одной переменной. Данные функции доопределяются до констант 0, 1 (рис. 4).

Осуществляется редукция (сокращение) нижнего яруса *BDD* согласно проведенному доопределению функций. После этого выполняется редукция остальных уровней *BDDI*, правила редукции показаны на рис. 4, 5. Поясним последнее правило редук-

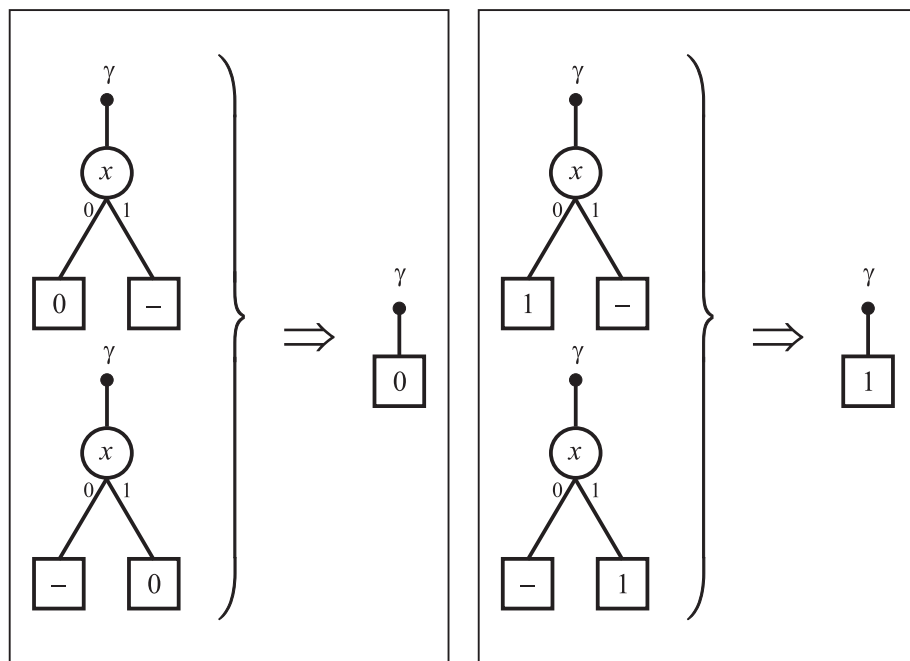


Рис. 4. Доопределение до констант частичных функций от одной переменной

ции на рис. 5. На этом рисунке функции s^i , s^j задают формулы разложения Шеннона по переменной x : $s^i = \bar{x}\alpha \vee x\bar{\alpha}$, $s^j = \bar{x}\bar{\alpha} \vee x\alpha$. Выполним инверсию функции s^i :

$$\begin{aligned} \bar{s}^i &= \overline{\bar{x}\alpha \vee x\bar{\alpha}} = (\overline{\bar{x}\alpha})(\overline{x\bar{\alpha}}) = (x \vee \bar{\alpha})(\bar{x} \vee \alpha) = \\ &= x\bar{x} \vee x\bar{\alpha} \vee x\alpha \vee \bar{\alpha}\alpha = \bar{x}\bar{\alpha} \vee x\alpha = s^j. \end{aligned}$$

После выполнения шага 5 выполняется переход на шаг 6.

Шаг 6. По полученной *BDDI* записывают формулы многоуровневого разложения Шеннона: каждой функциональной вершине соответствует функция, если же функциональная вершина *BDDI* содержит инверсный кружок, то соответствующая булева переменная (промежуточная функция) берется с инверсией. Выполняется переход на шаг 7.

Шаг 7. Для полученных формул многоуровневого разложения проводится дополнительное сокращение путем нахождения вырожденных формул разложения Шеннона: формула разложения Шеннона $f = \bar{x}_i s^1 \vee x_i s^2$, которая имеет вид $f = \bar{x}_i s^1 \vee x_i$ (в которой $s^2 = 1$, заменяется более простой формулой: $f = s^1 \vee x_i$, так как $\bar{x}_i s^1 \vee x_i = s^1 \vee x_i$. Аналогично, если формула разложения Шеннона для какой-то из подфункций имеет вид $f = \bar{x}_i \vee x_i s^2$ (в этой формуле $s^1 = 1$), то данная формула заменяется формулой $f = \bar{x}_i \vee s^2$.

После выполнения шага 7 выполняется переход на шаг 8.

Шаг 8. Конец алгоритма.

Рассмотрим решение задачи 3 для примера 2 с помощью алгоритма 1. Пусть задана частичная векторная функция (см. табл. 7) примера 2, требуется построить по перестановке $\langle x_1, x_2, x_3, x_4 \rangle$ полностью определен-

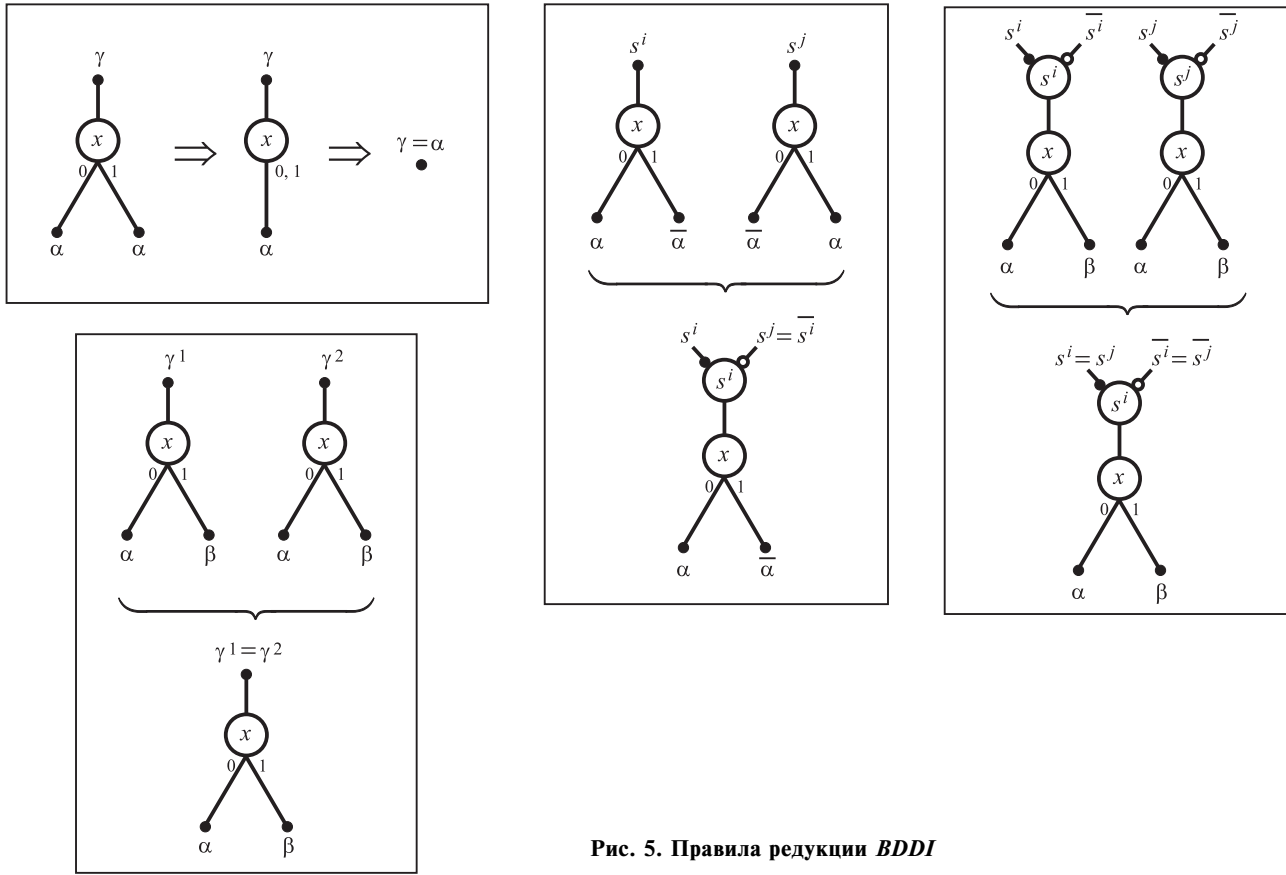


Рис. 5. Правила редукции BDDI

ную BDDI возможно меньшей сложности, которая будет реализовать исходную частичную векторную функцию.

Итерация 1. Разложение по переменной x_1 . Запишем формулы разложения Шеннона $f^1 = \bar{x}_1 a^1 \vee x_1 a^2$, $f^2 = \bar{x}_1 a^3 \vee x_1 a^4$ и для подфункций a^1, a^2, a^3, a^4 решим задачу 2. Подфункции, полученные разложением Шеннона по переменной x_1 , и результат решения задачи 2 для них даны в табл. 8. Три полученные функции реализуют исходные: $a^1 < b^1$, $a^2 < \bar{b}^1$, тогда разложения Шеннона принимают вид $f^1 = \bar{x}_1 b^1 \vee x_1 \bar{b}^1$, $f^2 = \bar{x}_1 a^3 \vee x_1 b^1$.

Таблица 8

Реализующие функции, построенные на первой итерации алгоритма 1

$x_2 x_3 x_4$	a^1	a^2	a^3	a^4	Реализующие функции		
					b^1	a^3	a^4
0 0 0	1	0	—	—	1	—	—
0 0 1	0	—	1	—	0	1	—
0 1 0	—	1	0	—	0	0	—
0 1 1	1	0	1	1	1	1	1
1 0 0	—	—	1	0	—	1	0
1 0 1	1	—	—	1	1	—	1
1 1 0	1	—	0	1	1	0	1
1 1 1	0	1	0	—	0	0	—

Итерация 2. Разложение по переменной x_2 . Запишем формулы разложения Шеннона $b^1 = \bar{x}_2 c^1 \vee x_2 c^2$, $a^3 = \bar{x}_2 c^3 \vee x_2 c^4$, $a^4 = \bar{x}_2 c^5 \vee x_2 c^6$ и для подфункций c^1, c^2, c^3, c^4 решим задачу 2. Подфункции c^1, \dots, c^6 , полученные разложением Шеннона, и результат решения задачи 2 для них даны в табл. 9. Среди шести функций находятся три реализующие: $c^2 < c^1$, $c^5 < c^1$, $c^6 < \bar{c}^1$. Разложения Шеннона принимают вид $b^1 = \bar{x}_2 c^1 \vee x_2 \bar{c}^1$, $a^3 = \bar{x}_2 c^3 \vee x_2 c^4$, $a^4 = \bar{x}_2 c^1 \vee x_2 \bar{c}^1$.

Итерация 3. Разложение по переменной x_3 . Запишем формулы разложения Шеннона $c^1 = \bar{x}_3 d^1 \vee x_3 d^2$, $c^3 = \bar{x}_3 d^3 \vee x_3 d^4$, $c^4 = \bar{x}_3 d^5 \vee x_3 d^6$ и для подфункций d^1, \dots, d^6 решим задачу 2. Подфункции d^1, \dots, d^6 , полученные разложениями Шеннона, и результат

Таблица 9

Реализующие функции, построенные на второй итерации алгоритма 1

$x_3 x_4$	c^1	c^2	c^3	c^4	c^5	c^6	Реализующие функции		
							c^1	c^3	c^4
0 0	1	—	—	1	—	0	1	—	1
0 1	0	1	1	—	—	1	0	1	—
1 0	0	1	0	0	—	1	0	0	0
1 1	1	0	1	0	1	—	1	1	0

Таблица 10

Реализующие функции, построенные на третьей итерации алгоритма 1

x_4	d^1	d^2	d^3	d^4	d^5	d^6	Реализующие функции		
							d^1	$e^2 = 0$	$e^3 = 1$
0	1	0	—	0	1	0	1	0	1
1	0	1	1	1	—	0	0	0	1

решения задачи 2 для них даны в табл. 10. Среди шести функций находятся три реализующие: $d^2 < \bar{d}^1$, $d^3 < 1$, $d^4 < \bar{d}^1$, $d^5 < 1$, $d^6 < 0$. Разложения Шеннона принимают вид $c^1 = \bar{x}_3 d^1 \vee x_3 \bar{d}^1$, $c^3 = \bar{x}_3 \vee x_3 \bar{d}^1$, $c^4 = \bar{x}_3 d^5 = \bar{x}_3$.

Итерация 4. На данной итерации требуется провести разложение Шеннона одной неконстантной функции $d^1 = \bar{x}_4$.

Шаги 1—3. Не выполняются, так как текущая система функций включает только одну функцию, зависящую от одной переменной x_4 .

Шаг 4. Полученная полностью определенная BDDI для функций из примера 2 дана на рис. 6, а. Листовых неопределенных вершин нет.

Шаг 5. Не выполняется, так как листовых неопределенных вершин нет.

Шаг 6. Многоуровневое представление в виде формул разложения Шеннона с использованием взаимно инверсных подфункций имеет сложность $S_{BDDI} = 6$ и записывается восемью уравнениями

$$f^1 = \bar{x}_1 b^1 \vee x_1 \bar{b}^1; \quad f^2 = \bar{x}_1 a^3 \vee x_1 b^1;$$

$$b^1 = \bar{x}_2 c^1 \vee x_2 \bar{c}^1; \quad a^3 = \bar{x}_2 c^3 \vee x_2 c^4;$$

$$c^1 = \bar{x}_3 d^1 \vee x_3 \bar{d}^1; \quad c^3 = \bar{x}_3 \vee x_3 \bar{d}^1;$$

$$c^4 = \bar{x}_3; \quad d^1 = \bar{x}_4.$$

Уравнения $c^4 = \bar{x}_3$; $d^1 = \bar{x}_4$ не учитываются при подсчете сложности BDDI.

Шаг 7. Уравнение $c^3 = \bar{x}_3 \vee x_3 \bar{d}^1$ упрощается: $c^3 = \bar{x}_3 \vee \bar{d}^1$.

На рис. 6, б дана BDD, полученная заменой взаимно инверсных функциональных вершин BDD на функциональные вершины BDD с последующей редукцией BDD. Данной BDD соответствует следующее многоуровневое представление

$$f^1 = \bar{x}_1 b^1 \vee x_1 b^2; \quad f^2 = \bar{x}_1 a^3 \vee x_1 b^1;$$

$$b^1 = \bar{x}_1 c^1 \vee x_1 c^2; \quad b^2 = \bar{x}_2 c^2 \vee x_2 c^1;$$

$$a^3 = \bar{x}_2 c^3 \vee x_2 c^4; \quad c^1 = \bar{x}_3 d^1 \vee x_3 d^2;$$

$$c^2 = \bar{x}_3 d^2 \vee x_3 d^1; \quad c^3 = \bar{x}_3 \vee x_3 d^2;$$

$$c^4 = \bar{x}_3; \quad d^1 = \bar{x}_4; \quad d^2 = x_4$$

со сложностью $S_{BDD} = 8$.

Задача 4. Задана BDD частичной векторной булевой функции $f(x)$ по заданной перестановке $\langle x_{j_1}, x_{j_2}, \dots, x_{j_n} \rangle$ переменных. Требуется для этой перестановки переменных получить полностью определенную BDDI, реализующую исходную векторную булеву функцию, при этом BDDI должна быть возможно меньшей сложности.

Эвристический алгоритм 2 (решения задачи 4) основан на постепенном доопределении BDD "снизу", т. е. от листовых вершин. На каждом шаге рассматриваются подфункции, располагаемые на одном (следующем по порядку) уровне BDD, и для этих подфункций решается задача 2.

На первом шаге троичная матрица T для нахождения кратчайшей имплицитной формы задана в табл. 11 (элементы троичной матрицы T выделены полужирным шрифтом), в последнем столбце данной таблицы указана реализующая функция для каждой из подфункций s^{12}, \dots, s^{16} , те же реализующие функции для BDD показаны на рис. 7, зависимость подфункций следующего верхнего уровня показана на рис. 8.

На втором шаге рассматриваются семь подфункций s^5, \dots, s^{11} (табл. 12), которые заменяются четырьмя реализующими подфункциями, указанными в последнем столбце табл. 12. Редуцированная BDDI показана на рис. 9, она является результирующей, так как все подфункции следующего верхнего уровня являются различными и нет взаимно инверсных

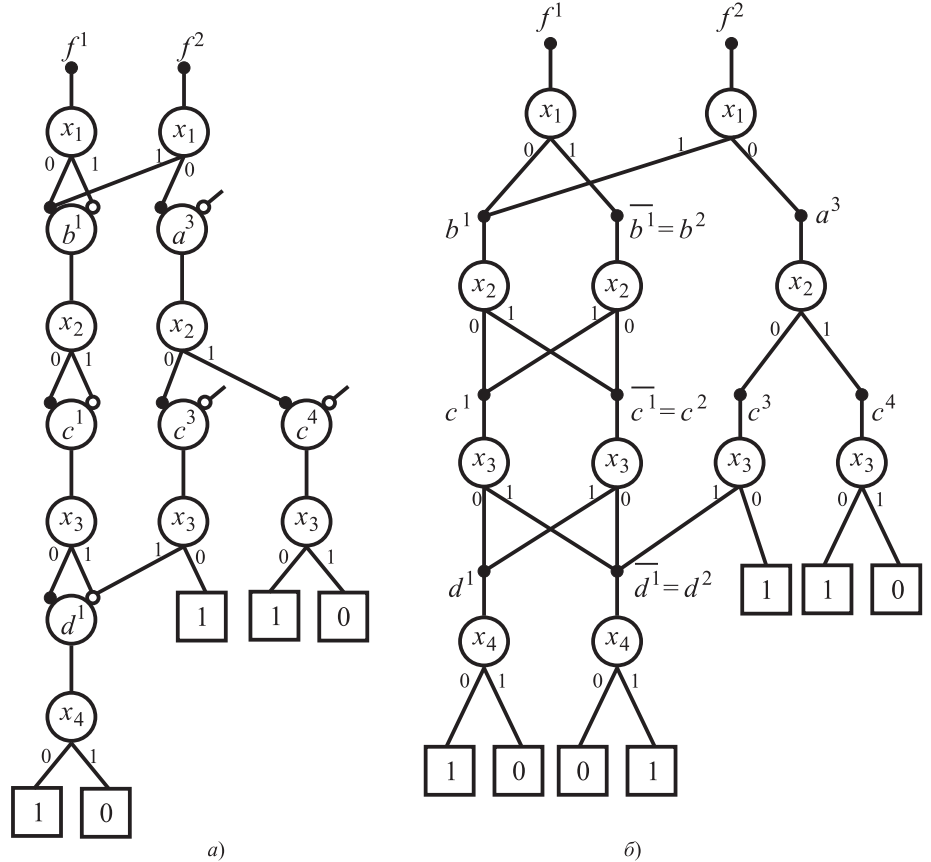


Рис. 6. Сложность $S_{BDDI} = 6$ (а) меньше сложности $S_{BDD} = 8$ (б)

Таблица 11

Реализующие функции, зависящие от переменной x_4

	$x_4 = 0$	$x_4 = 1$	Частичные подфункции	Реализующая функция
$T =$	1	0	s^{12}	φ^1
	–	1	s^{13}	$\bar{\varphi}^1$
	0	–	s^{14}	$\bar{\varphi}^1$
	0	1	s^{15}	$\bar{\varphi}^1$
	1	–	s^{16}	φ^1
	0	0	0	0
	–	–	–	0

подфункций. При этом единственное неопределенное значение, являющееся листовым при разложении подфункции s^8 по переменной x_3 (см. табл. 12), доопределено до нуля.

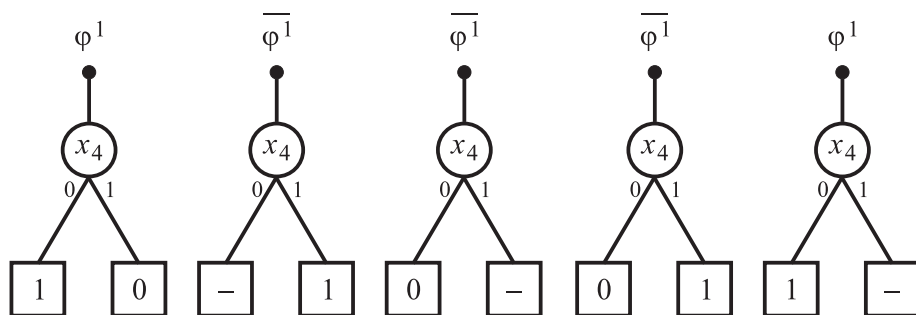


Рис. 7. Доопределение подфункций, зависящих от переменной x_4

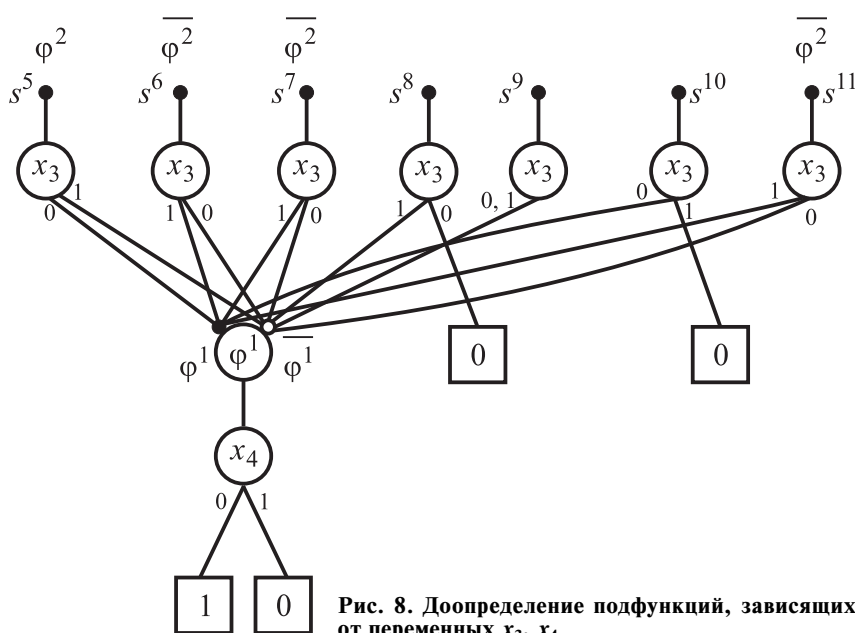


Рис. 8. Доопределение подфункций, зависящих от переменных x_3, x_4

Многоуровневое представление, соответствующее *BDDI* (рис. 9) имеет сложность $S_{BDDI} = 9$ и записывается десятью уравнениями

$$\begin{aligned} f^1 &= \bar{x}_1 s^1 \vee x_1 s^2; & f^2 &= \bar{x}_1 s^3 \vee x_1 s^4; \\ s^1 &= \bar{x}_1 \varphi^2 \vee x_2 \bar{\varphi}^2; & s^2 &= \bar{x}_2 \bar{\varphi}^2 \vee x_2 s^8; \\ s^3 &= \bar{x}_2 \bar{\varphi}^1 \vee x_2 s^{10}; & s^4 &= \bar{x}_2 s^8 \vee x_2 \bar{\varphi}^2; \\ \varphi^2 &= \bar{x}_3 \varphi^1 \vee x_3 \bar{\varphi}^1; & s^8 &= x_3 \bar{\varphi}^1; & s^{10} &= \bar{x}_3 \varphi^1; & \varphi^1 &= \bar{x}_4. \end{aligned}$$

Недостаток алгоритма 2 — минимизация числа функциональных вершин только на отдельном (очередном) уровне *BDD*.

Эвристический алгоритм 3 (решения задачи 4) использует доопределение *BDD* снизу, рассматривая на каждом шаге не одну переменную, а подмножество из нескольких переменных, которое назовем "блоком".

Проиллюстрируем алгоритм 3 на примере *BDD* (см. рис. 2). В качестве первого блока будут переменные подмножества $\{x_3, x_4\}$.

Шаг 1. По исходной *BDD* (см. рис. 2) запишем подфункции s^5, \dots, s^{11} , зависящие от переменных x_3, x_4 , в виде конъюнкций, соответствующих путям на графе *BDD* (см. рис.2), получим табл. 13.

Для удобства зададим подфункции s^5, \dots, s^{11} в виде таблицы истинности (табл. 14). Для блока переменных, содержащего достаточно большое число переменных (десятки и более), вместо таблицы истинности данные функции можно задать ортогонализованной либо обобщенно-ортогонализованной формой.

Однако для *BDDI* нахождение минимального дизъюнктивного базиса для подфункций выбранного уровня *BDDI* значительно упрощается за счет того, что ДНФ для задания областей значений каждой из подфункций являются попарно ортогональными и ортогонализированными. Задача нахождения минимального дизъюнктивного базиса для ДНФ, полученных из *BDD*, гораздо легче, чем задача нахождения минимального дизъюнктивного базиса для системы произвольных (неортогонализированных) ДНФ, т. е. ДНФ, конъюнкции которых находятся в различных отношениях друг с другом: ортогональности, неортогональности, равенства, неравенства, импликации. Кроме того, сами произвольные ДНФ (т. е. не попарно ортогональные ДНФ) также могут находиться в отношениях, аналогичных перечисленным отношениям конъюн-

Таблица 12

Таблица 13

Реализующие функции, зависящие от переменных x_3, x_4

Подфункции *BDD* (рис. 2), зависящие от переменных x_3, x_4

Частичные подфункции	$x_3 = 0$	$x_3 = 1$	Реализующая функция
	Подфункции разложения Шеннона		
s^5	φ^1	$\bar{\varphi}^1$	φ^2
s^6	$\bar{\varphi}^1$	φ^1	$\bar{\varphi}^2$
s^7	$\bar{\varphi}^1$	φ^1	$\bar{\varphi}^2$
s^8	—	$\bar{\varphi}^1$	s^8
s^9	$\bar{\varphi}^1$	$\bar{\varphi}^1$	s^9
s^{10}	φ^1	0	s^{10}
s^{11}	$\bar{\varphi}^1$	φ^1	$\bar{\varphi}^2$

Частичная подфункция	$x_3 x_4$	Значение
s^5	0 1	0
	0 0	1
	1 1	1
s^6	1 0	—
	1 1	0
	0 1	1
s^7	1 0	1
	0 0	—
	0 1	—
s^8	0 0	0
	1 1	0
	1 0	1
s^9	0 1	—
	1 1	1
	0 0	—
s^{10}	1 —	0
	0 0	1
	0 1	—
s^{11}	0 0	0
	0 1	1
	1 0	1
	1 1	—

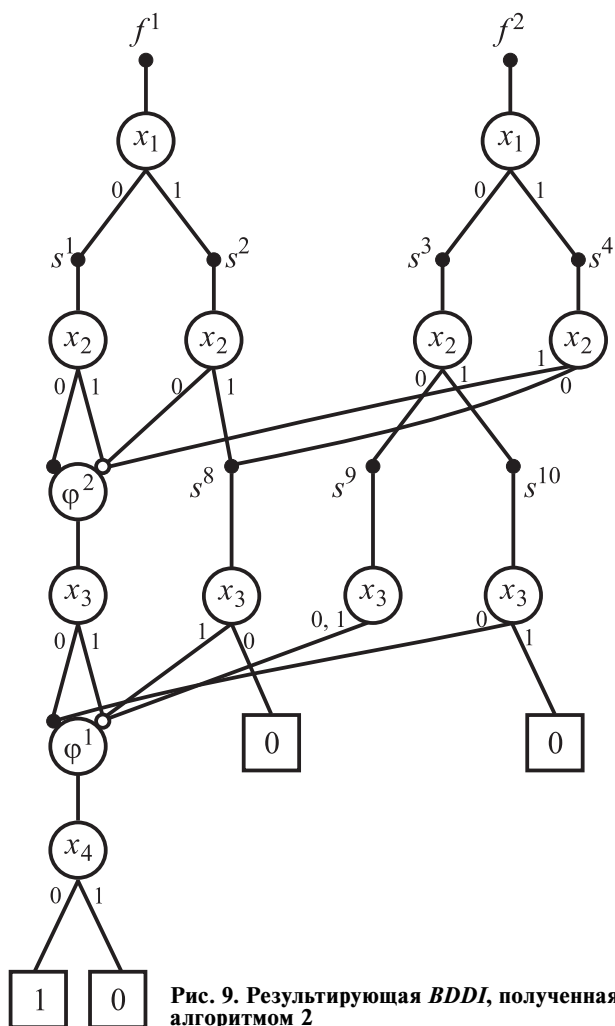


Рис. 9. Результирующая *BDDI*, полученная алгоритмом 2

юнкций. Именно эти ортогонализированные формы задания подфункций, являющихся функциональными вершинами *BDD*, обеспечивают эффективные алгоритмы доопределения *BDDI*, так как обобщенно-ортогонализированная форма системы частных функций, извлеченных из *BDD*, сводится к операциям перемножения ДНФ, представляющих области значений неконстантных подфункций, находящихся на одном уровне *BDD*, и сравнению на равенство ортогонализированных ДНФ, что гораздо более просто, чем сравнение произвольных ДНФ. По сути, в табл. 14 задана (столбцами) троичная матрица *T* значений подфункций s^5, \dots, s^{11} . Для троичной матри-

Таблица 14

Частичные булевы подфункции и их реализующие

$x_3 x_4$	$s^5 s^6 s^7 s^8 s^9 s^{10} s^{11}$	$\psi^1 \bar{\psi}^1 \psi^2 \psi^3$
0 0	1 — 0 — — 1 0	1 0 — 1
0 1	0 1 — — 1 — 1	0 1 1 —
1 0	— 1 1 — 0 0 1	0 1 0 0
1 1	1 0 0 1 1 0 —	1 0 1 0

Таблица 15

Задание значений частичных подфункций строками троичной матрицы

	$x_3 \ x_4$				Частичные подфункции	Реализующая подфункция
	0 0	0 1	1 0	1 1		
$T =$	1	0	–	1	s^5	ψ^1
	–	1	1	0	s^6	$\bar{\psi}^1$
	0	–	1	0	s^7	$\bar{\psi}^1$
	–	–	–	1	s^8	ψ^1
	–	1	0	1	s^9	ψ^2
	1	–	0	0	s^{10}	ψ^3
	0	1	1	–	s^{11}	$\bar{\psi}^1$

цы T (табл. 15) решим задачу нахождения кратчайшей имплицитной формы. Получим, что подфункции s^5, \dots, s^{11} можно заменить тремя новыми подфункциями ψ^1, ψ^2, ψ^3 (табл. 15), эти подфункции требуется реализовать в виде подграфов *BDDI*.

На рис. 10 видно, что данные подфункции записываются в виде разложений Шеннона: $\psi^1 = \bar{x}_3 p \vee x_3 \bar{p}$, $\psi^2 = \bar{p}$, $\psi^3 = \bar{x}_3 p$.

Шаг 2. Рассмотрим подфункции s^1, s^2, s^3, s^4 на следующем верхнем уровне и проведем редукцию, выполнив поиск одинаковых и взаимно инверсных подфункций (см. табл. 14, 15). Легко видеть, что $s^1 = s^4$, тогда алгоритм заканчивает свою работу, так как функции f^1, f^2 нельзя доопределить до одинаковых либо взаимно инверсных.

Результирующая *BDDI* имеет сложность $S_{BDDI} = 6$ и показана на рис. 11, данной *BDDI* соответствует следующее многоуровневое представление:

$$f^1 = \bar{x}_1 s^1 \vee x_1 \bar{s}^1; \quad f^2 = \bar{x}_1 s^3 \vee x_1 \bar{s}^3;$$

$$s^1 = \bar{x}_1 \psi^1 \vee x_1 \bar{\psi}^1; \quad s^3 = \bar{x}_2 \bar{p} \vee x_2 \psi^3;$$

$$\psi^1 = \bar{x}_3 p \vee x_3 \bar{p}; \quad \psi^3 = \bar{x}_3 p; \quad p = \bar{x}_4.$$

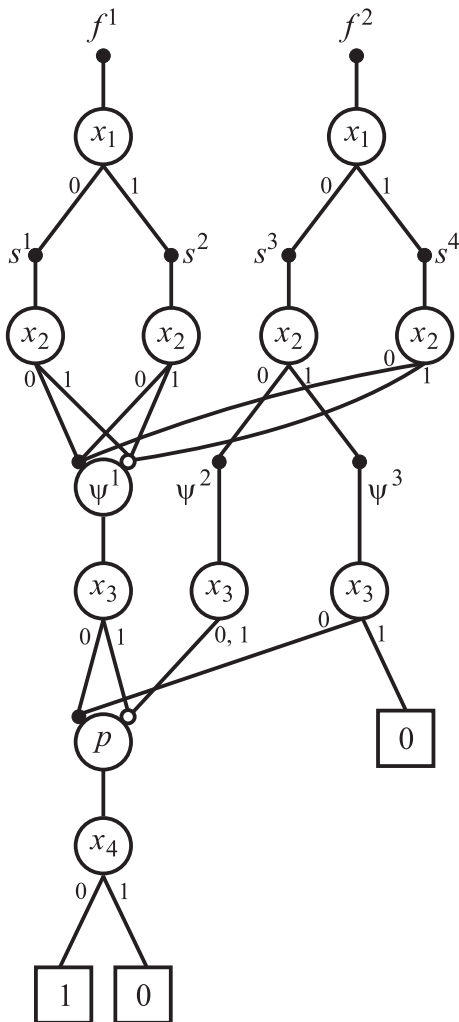


Рис. 10. Доопределенная *BDDI*, полученная после рассмотрения блока переменных $\{x_3, x_4\}$

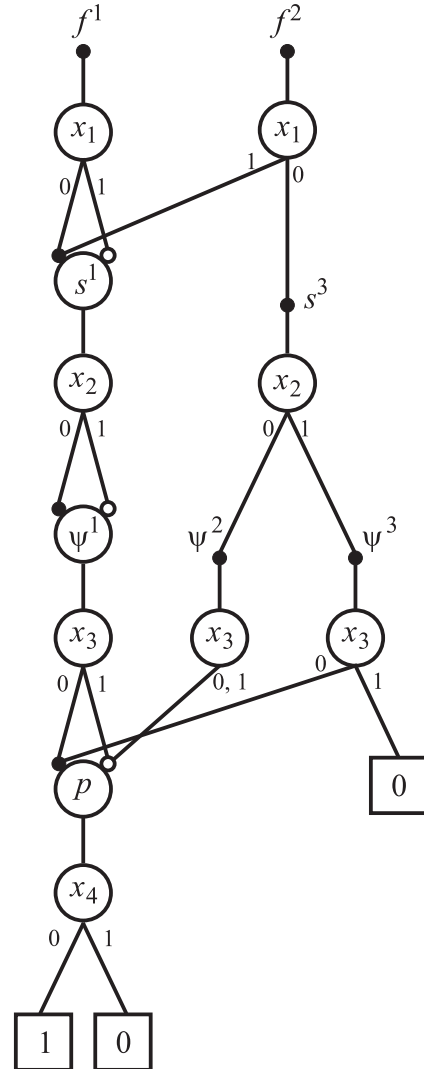


Рис. 11. Результирующая *BDDI* для алгоритма 3

Таблица 16

Доопределение частичных функций из примера 2 различными алгоритмами оптимизации BDDI

Исходные частичные функции		Полностью определенные функции — результат доопределения частичных функций			
$x_1 x_2 x_3 x_4$	$f^1 f^2$	Алгоритм 1	Алгоритм 2	Алгоритм 3	Доопределение, полученное программой [30]
		$f^1 f^2$	$f^1 f^2$	$f^1 f^2$	
0 0 0 0	1 –	1 1	1 0	1 0	1 1
0 0 0 1	0 1	0 1	0 1	0 1	0 1
0 0 1 0	– 0	0 0	0 0	0 0	1 0
0 0 1 1	1 1	1 1	1 1	1 1	1 1
0 1 0 0	– 1	0 1	0 1	0 1	1 1
0 1 0 1	1 –	1 1	1 0	1 0	1 1
0 1 1 0	1 0	1 0	1 0	1 0	1 0
0 1 1 1	0 0	0 0	0 0	0 0	0 0
1 0 0 0	0 –	0 1	0 0	0 0	0 0
1 0 0 1	– –	1 0	1 0	1 0	0 1
1 0 1 0	1 –	1 0	1 0	1 0	1 1
1 0 1 1	0 1	0 1	0 1	0 1	0 1
1 1 0 0	– 0	1 0	0 0	0 0	1 0
1 1 0 1	– 1	0 1	0 1	0 1	1 1
1 1 1 0	– 1	0 1	0 1	0 1	1 1
1 1 1 1	1 –	1 0	1 0	1 0	1 1

Заметим, что в данных формулах подфункция ψ^2 заменена на \bar{p} .

В табл. 16 даны полностью определенные функции, полученные различными алгоритмами. Полужирным шрифтом выделены полностью определенные значения (0, 1) функций, которые заменили неопределенные значения исходных частичных функций. В последнем столбце приводится для сравнения доопределение, полученное в результате совместной минимизации системы частичных функций с помощью программы [30], включенной в систему логической оптимизации [31]. Заметим, что BDDI-представление

$$f^1 = \bar{x}_4 v^2 \vee x_4 v^3; \quad f^2 = \bar{x}_4 v^0 \vee x_4 v^1;$$

$$v^0 = \bar{x}_1 \bar{x}_3 \vee x_1 x_3; \quad v^1 = \bar{x}_2 \vee x_2 v^4;$$

$$v^2 = \bar{x}_2 v_7 \vee x_2; \quad v^3 = \bar{x}_2 \bar{v}_4 \vee x_2 v^4;$$

$$v^4 = \bar{x}_1 \bar{x}_3 \vee x_1; \quad v^7 = \bar{x}_1 \vee x_1 x_3$$

для минимизированной в классе ДНФ системы полностью определенных функций построено с помощью программы [21] по перестановке $\langle x_4, x_2, x_1, x_3 \rangle$ и имеет сложность $S_{BDDI} = 8$ (функциональных вершин).

Задача 5. Задана матричная форма частичной векторной булевой функции $f(x)$. Требуется найти перестановку $\langle x_{j_1}, x_{j_2}, \dots, x_{j_n} \rangle$ переменных, по которой BDDI будет иметь, возможно, меньшую сложность.

Эвристический алгоритм последовательного нахождения перестановки (эвристический алгоритм решения задачи 5).

Шаг 1. Рассматриваются все упорядоченные пары входных переменных. Для каждой пары $\langle x_{i_1}, x_{i_2} \rangle$ строятся два начальных уровня BDDI и оценивается сложность (без учета функциональных вершин-переменных либо их инверсий). Выбирается пара $\langle x_{i_1}, x_{i_2} \rangle$, для которой начальный фрагмент BDDI имеет наименьшую сложность, данная пара определяет две переменные искомой перестановки: $x_{j_1} = x_{i_1}$, $x_{j_2} = x_{i_2}$.

Шаг 2 (итеративный). К полученной перестановке поочередно добавляется одна из оставшихся переменных и оценивается сложность BDDI. Выбирается та из переменных x_{j_s} , которая обеспечивает наименьшую сложность BDDI. Такой выбор переменных продолжается до тех пор, пока не будут определены все переменные в искомой перестановке $\langle x_{j_1}, x_{j_2}, \dots, x_{j_n} \rangle$.

Заметим, что данный алгоритм оказался эффективным для минимизации сложности BDDI, реализующей систему полностью определенных функций, полученные результаты его экспериментального исследования даны в работе [21].

Заключение

Предложены методы рационального доопределения систем частичных булевых функций, позволяющие получать минимизированные многоуровневые BDDI-представления с использованием одинаковых и взаимно инверсных подфункций. Такие представления являются более простыми по сравнению с BDD-представлениями по критерию числа литералов в функциональных описаниях, по которым осуществляется синтез логических схем в различных технологических базисах.

Список литературы

1. Брейтон Р. К., Хэчтел Г. Д., Санджованни-Винченцелли А. Л. Синтез многоуровневых комбинационных логических схем // ТИИЭР. — 1990. — Т. 78, № 2. — С. 38–83.
2. Закревский А. Д., Поттосин Ю. В., Черемисинова Л. Д. Логические основы проектирования дискретных устройств. — М.: Физматлит, 2007. — 592 с.
3. Brayton K. R., Hachtel G. D., McMullen C. T., Sangiovanni-Vincentelli A. L. Logic minimization algorithm for VLSI synthesis. — Boston, e.a.: Kluwer Academic Publishers, 1984. — 193 p.
4. Ashenurst R. L. The Decomposition of Switching Functions // Annals of Computation Laboratory of Harvard University. — 1959. — Vol. 29. — P. 74 — 116.
5. Curtis H. A. A new approach to the design of switching circuit. — Princeton, Van Nostrand, 1962. — 635 p.
6. Roth J. P., Karp R. M. Minimization over Boolean Graphs // IBM J. Res. and Dev. — 1962. — Vol. 6, No. 2. — P. 227–238.
7. Поттосин Ю. В., Шестаков Е. А. Табличные методы декомпозиции систем полностью определенных булевых функций. — Минск: Беларус. наука, 2006. — 327 с.
8. Бибило П. Н. Декомпозиция булевых функций на основе решения логических уравнений. — Минск: Беларус. наука, 2009. — 211 с.
9. Mailhot F., Micheli G. Algorithms for technology mapping based on binary decision diagrams and on Boolean operations // IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems. — 1993. — Vol. 12, No. 5. — P. 599–620.
10. Advanced Techniques in Logic Synthesis, Optimizations and Applications /Eds: S. P. Khatiri, K. Gulati. Springer, 2010. 423 p.
11. Amaru L. G. New Data Structures and Algorithms for Logic Synthesis and Verication. — Springer, 2017. — 156 p.

12. **Advanced** Logic Synthesis / Eds: A. I. Reis, R. Drechsle. Springer, 2017. — 232 p.
13. **Кузнецов О. П.** О программной реализации логических функций и автоматов // Автоматика и телемеханика. — 1977. № 7. — С. 63—74; № 9. — С. 138—149.
14. **Akers S. B.** Binary Decision Diagrams // IEEE Trans. on Computers. — 1978. — Vol. C-27, No. 6. — P. 509—516.
15. **Bryant R. E.** Graph-based algorithms for boolean functions manipulation // IEEE Trans. on Computers. — 1986. — Vol. C-35, No. 8. — P. 677—691.
16. **Bryant R. E., Meinel C.** Ordered Binary Decision Diagrams // Logic synthesis and verification / Eds. S. Hassoun, T. Sasao, R. K. Brayton. Kluwer Academic Publishers, 2002. P. 285—307.
17. **Meinel C., Theobald T.** Algorithms and Data Structures in VLSI Design: OBDD Foundations and Applications. — Berlin Heidelberg. Springer-Verlag, 1998. — 267 p.
18. **Кнут Д. Э.** Искусство программирования. — М.: Вильямс, 2013. Т. 4, А: Комбинаторные алгоритмы, ч. 1. — 960 с.
19. **Карпов Ю. Г.** MODEL CHECKING. Верификация параллельных и распределенных программных систем. — СПб.: БХВ-Петербург, 2010. — 560 с.
20. **Бибилло П. Н.** Применение диаграмм двоичного выбора при синтезе логических схем. — Минск: Беларус. навука, 2014. — 231 с.
21. **Бибилло П. Н., Ланкевич Ю. Ю.** Использование полиномов Жегалкина при минимизации многоуровневых представлений систем булевых функций на основе разложения Шеннона // Программная инженерия. — 2017. Том 8, № 8. — С. 369—384.
22. **Поттосин Ю. В., Шестаков Е. А.** Ортогонализация системы полностью определенных булевых функций // Логическое проектирование: Сб. науч. тр. Минск: Ин-т техн. кибернетики НАН Беларуси. — 2000. — Вып. 5. — С. 107—115.
23. **Уткин А. А.** Анализ логических сетей и техника булевых вычислений. — Минск: Наука и техника, 1979. — 152 с.
24. **Романов В. И.** Программные средства для решения логико-комбинаторных задач // Информатика. — 2005. — № 4. — С. 114—123.
25. **Goldberg E., Novikov Y.** BerkMin: a Fast and Robust SAT-Solver // Discrete Applied Mathematics. — 2007. — Vol. 155, No. 12. — P. 1549 — 1561.
26. **Handbook of Satisfiability** / Eds. A. Biere, M. Heule, H. Van Maaren, T. Walsh. IOS Press, 2009. 980 p.
27. **Романовский И. В.** Дискретный анализ: Учебное пособие для студентов, специализирующихся по прикладной математике и информатике. 4-е изд., испр. и доп. — СПб.: Невский Диалект; БХВ-Петербург, 2008. — 336 с.
28. **Бибилло П. Н., Енин С. В.** Синтез комбинационных схем методами функциональной декомпозиции. — Минск: Наука и техника, 1987. — 189 с.
29. **Бибилло П. Н., Леончик П. В.** Декомпозиция систем булевых функций, заданных диаграммами двоичного выбора // Известия РАН. Теория и системы управления. — 2011. — № 4. — С. 86—101.
30. **Торопов Н. Р.** Минимизация систем булевых функций в классе ДНФ // Логическое проектирование. — 1999. Вып. 4. — С. 4—19.
31. **Бибилло П. Н., Романов В. И.** Логическое проектирование дискретных устройств с использованием продукционно-фреймовой модели представления знаний. — Минск: Беларус. навука, 2011. — 279 с.

Minimization of Binary Decision Diagrams for Systems of Incompletely Defined Boolean Functions Using Inverse Cofactors

P. N. Bibilo, bibilo@newman.bas-net.by, The United Institute of Informatics Problems of the National Academy of Sciences of Belarus, Minsk, 220012, Belarus

Corresponding author:

Bibilo Petr N., Head of Laboratory, The United Institute of Informatics Problems of the National Academy of Sciences of Belarus, 220012, Minsk, Belarus
E-mail: bibilo@newman.bas-net.by

Received on January 09, 2020

Accepted on March 03, 2020

Completely defined Boolean functions are mathematical models of combinational logic circuits. In design practice, models of incompletely defined (partial) Boolean functions are also used. Uncertain values of such functions in the design process are determined in order to obtain the best characteristics of logic circuits. The main characteristics of logic circuits are chip size, delays and power consumption. Currently, the circuit implementation of systems of partial functions uses two approaches. The first approach assumes minimization of the initial representation of systems of partial functions in the class of disjunctive normal forms (DNF). The second approach is related to obtaining minimized multilevel representations of a system of functions. Finding multilevel representations corresponds to solving various problems of decomposition of systems of partial functions. Among the many types of functional decomposition and the corresponding forms of representations of function systems, the multilevel representations of function systems based on the Shannon expansion were the most effective, which are also known as BDD representations in the literature.

In this article, algorithms for the logical minimization of BDD representations (BDD — Binary Decision Diagram) of systems of partial Boolean functions are proposed. The result of the minimization is the multilevel BDD representations of systems of completely defined functions in the form of Shannon expansion formulas using both identical and mutually inverse cofactors. The minimization of the number of mutually inverse partial cofactors at the same level of the BDD representation of the system of partial functions is reduced to solving the combinatorial problem of

finding the shortest implicating form of the ternary matrix that defines the values of the cofactors of this level of the BDD representation.

Keywords: Boolean function, disjunctive normal form, Shannon expansion, binary decision diagram (BDD), graph coloring, synthesis of logical circuit

For citation:

Bibilo P. N. Minimization of Binary Decision Diagrams for Systems of Incompletely Defined Boolean Functions Using Inverse Cofactors, *Programmnaya Ingeneria*, 2020, vol. 11, no. 3, pp. 152–168

DOI: 10.17587/prin.11.152-168

References

1. **Brayton R. K., Hachtel G. D., Sangiovanni-Vincentelli A. L.** Multilevel Logic Synthesis, *Trudy institute inzhenerov po jelektronike i radiotekhnike*, 1990, vol. 78, no. 2, pp. 38–83 (in Russian).
2. **Zakrevskij A. D., Pottosin Ju.V., Cheremisinova L. D.** Logical bases of design of discrete devices, Moscow, Fizmatlit, 2007, 589 p. (in Russian).
3. **Brayton K. R., Hachtel G. D., McMullen C. T., Sangiovanni-Vincentelli A. L.** *Logic minimization algorithm for VLSI synthesis*, Boston, e.a.: Kluwer Academic Publishers, 1984, 193 p.
4. **Ashenurst R. L.** The Decomposition of Switching, *Annals of Computation Laboratory of Harvard University*, 1959, vol. 29, pp. 74–116.
5. **Curtis H. A.** *A new approach to the design of switching circuit*, Princeton, Van Nostrand, 1962, 635 p.
6. **Roth J. P., Karp R. M.** Minimization over Boolean Graphs, *IBM J. Res. and Dev.*, 1962, vol. 6, no. 2, pp. 227–238.
7. **Pottosin Yu. V., Shestakov E. A.** Tabular methods for decomposition of systems of completely defined Boolean functions, Minsk, Belarus. navuka, 2006, 327 p. (in Russian).
8. **Bibilo P. N.** Decomposition of Boolean functions based on solving logical equations, Minsk, Belarus. navuka, 2009. 211 p. (in Russian).
9. **Mailhot F., Micheli G.** Algorithms for technology mapping based on binary decision diagrams and on Boolean operations, *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 1993, vol. 12, no. 5, pp. 599–620.
10. **Advanced Techniques in Logic Synthesis, Optimizations and Applications.** / Eds. S. P. Khatri, K. Gulati, Springer, 2010, 423 p.
11. **Amaru L. G.** *New Data Structures and Algorithms for Logic Synthesis and Verification*, Springer, 2017, 156 p.
12. **Advanced Logic Synthesis** / Eds. A. I. Reis, R. Drechsler, Springer, 2017, 232 p.
13. **Kuznecov O. P.** On the software implementation of logical functions and automata, *Avtomatika i telemekhanika*, 1977, no. 7, pp. 63–74 (in Russian).
14. **Akers S. B.** *Binary Decision Diagrams*, *IEEE Trans. on Computers*, 1978, vol. C-27, no. 6, pp. 509–516.
15. **Bryant R. E.** Graph-based algorithms for Boolean function manipulation, *IEEE Transactions on Computers*, 1986, vol. 35, no. 8, pp. 677–691.
16. **Bryant R. E.** *Ordered Binary Decision Diagrams, Logic synthesis and verification* / Eds. S. Hassoun, T. Sasao, R. K. Brayton, Kluwer Academic Publishers, 2002, pp. 285–307.
17. **Meinel C.** *Algorithms and Data Structures in VLSI Design: OBDD — Foundations and Applications*, Berlin, Heidelberg, Springer-Verlag, 1998, 267 p.
18. **Knut D. Je.** *The Art of Computer Programming, Combinatorial Algorithms, Part 1*, Moscow, Vil'jams, 2013, vol. 4, A: Kombinatornye algoritmy, 960 p. (in Russian).
19. **Karpov Yu. G.** *MODEL CHECKING*. Verification of parallel and distributed software systems, Saint Petersburg, BHV-Peterburg, 2010, 560 p. (in Russian).
20. **Bibilo P. N.** Application of Binary Decision Diagrams at synthesis of logical circuits, Minsk, Belarus. navuka, 2014, 231 p. (in Russian).
21. **Bibilo P. N., Lankevich Yu. Yu.** Using Zhegalkin polynomials to minimize multilevel representations of Boolean function systems based on the Shannon expansion, *Programmnaya ingeneria*, 2017, no. 3, pp. 369–384 (in Russian).
22. **Pottosin Yu. V., Shestakov E. A.** Orthogonalization of a system of completely defined Boolean functions, *Logicheskoe proektirovanie: Sb. nauch. tr.*, Minsk, In-t tekhn. kibernetiki NAN Belarusi, 2000, vol. 5, pp. 107–115 (in Russian).
23. **Utkin A. A.** The analysis of logical networks and the technique of Boolean calculations, Minsk, Nauka i tekhnika, 1979, 152 p. (in Russian).
24. **Romanov V. I.** Software tools for the solution of the logical-combinatorial task, *Informatika*, 2005, no. 4, pp. 114–123 (in Russian).
25. **Goldberg E., Novikov Y.** BerkMin: a Fast and Robust Sat-Solver, *Discrete Applied Mathematics*, 2007, vol. 155, no. 12, pp. 1549–1561.
26. **Handbook of Satisfiability** / Eds. A. Biere, M. Heule, H. Van Maaren, T. Walsh, IOS Press, 2009, 980 p.
27. **Romanovskij I. V.** *Discrete analysis* Uchebnoe posobie dlya studentov, specializiruyushchihsvya po prikladnoj matematike i informatike, 4-e izd., ispr. i dop., Saint-Petersburg, Nevskij Dialekt, BHV-Peterburg, 2008, 336 p.
28. **Bibilo P. N., Enin S. V.** *Synthesis of combination circuits by methods of functional decomposition*, Minsk, Nauka i tekhnika, 1987, 189 p. (in Russian).
29. **Bibilo P. N., Leonchik P. V.** Decomposition of systems of Boolean functions defined by binary decision diagrams, *Izvestiya RAN. Teoriya i sistemy upravleniya*, 2011, no. 4, pp. 86 — 101 (in Russian).
30. **Toropov N. R.** Minimization of systems of Boolean functions in the class DNF, *Logicheskoe proektirovanie*, Minsk, In-t tekhn. kibernetiki NAN Belarusi, 1999, iss. 4, pp. 4–19 (in Russian).
31. **Bibilo P. N., Romanov V. I.** *Logical design of discrete devices with use of productional and frame model of representation of knowledge*, Minsk, Belarus. navuka, 2011, 279 p. (in Russian).

Automated Generation of Quantum Circuit Specifications Based on Reed — Muller Expressions

V. A. Kalmychkov, vakalmychkov@etu.ru, I. V. Matveeva, ir_mat@mail.ru, Faculty of Computer Science and Technology, Saint Petersburg Electrotechnical University "LETI", Saint Petersburg, 197376, Russian Federation

Corresponding author:

Kalmychkov Vitaly A., Associate Professor, Saint Petersburg Electrotechnical University "LETI", Saint Petersburg, 197376, Russian Federation
E-mail: vakalmychkov@etu.ru

Received on February 20, 2020

Accepted on March 27, 2020

Quantum computing is gradually reaching the applied level from the theoretical field due to the rapid progress of the technological base and the emergence of real quantum computer prototypes. Logical operations are the basis of computing devices, therefore quantum logical operations must be present in the quantum circuit, which is a model of quantum computing at the logical level.

This paper presents our experience in automated synthesis and analysis of quantum circuit specifications and the results of using a set of developed applications to support these processes. The design, generation of possible variants and selection of a suitable quantum circuit are the iterative problem of multi-criteria search. The article considers the stages of a set of polarized Reed—Muller expressions of fixed polarity (FPRM) synthesis at the logical level by the truth table and automatic generation of sets of quantum Boolean schemes in notation of quantum gates. Results of using auto-minimize templates for SWAP gates composition in the transition to the architecture of linear nearest neighbor (LNN) are demonstrated.

For qualitative and quantitative evaluation of automatically obtained specifications, the developed applications provide lexical verification of the generated quantum circuits based on symbolic notation for the coefficients of the involved qubits. The results of the automatic formation of the statistical analysis table for the choice of the quantum circuit according to the criteria set by the assumed technology of physical realization are presented.

Keywords: qubit, quantum gate, quantum circuits, linear nearest neighbor architectures, quantum computer, quantum cost, fixed-polarity Reed—Muller expressions (FPRM)

For citation:

Kalmychkov V. A., Matveeva I. V. Automated Generation of Quantum Circuit Specifications Based on Reed—Muller Expressions, *Programmnaya Ingeneria*, 2020, vol. 11, no. 3, pp. 169—176

DOI: 10.17587/prin.11.169-176

УДК: 004.421, 004.04, 519.688

В. А. Калмычков, канд. техн. наук, доц., vakalmychkov@etu.ru,

И. В. Матвеева, канд. техн. наук, доц., ir_mat@mail.ru, Санкт-Петербургский государственный электротехнический университет "ЛЭТИ"

Автоматизированная генерация спецификаций квантовых цепей на основе полиномов Рида — Маллера

Представлены современный уровень квантовых вычислений и авторский опыт разработки приложений для синтеза и анализа спецификаций квантовых цепей для генерируемых поляризованных полиномов Рида—Маллера. Описана лексическая верификация квантовых цепей на основе символической нотации коэффициентов кубитов. Приведены результаты применения шаблонов автоматической минимизации преобразователей SWAP в квантовых цепях в архитектуре линейно ближайшего соседства и автоматически формируемые таблицы статистического анализа для многокритериального выбора квантовой цепи.

Ключевые слова: кубит, квантовая цепь, квантовая стоимость, полиномы Рида—Маллера фиксированной поляриности

Introduction

More than 30 years ago, the proposed approach [1] for a quantum computer could seem like a distant fantastic event. The general principles of operation and its description required significant knowledge in atomic physics and were radically different from the traditional computer architecture, well-known algorithms and programming languages.

The next 15 years of research were conducted in the field of theoretical understanding of general approaches to the organization of quantum computing. The specification of quantum circuit (quantum algorithm) [2, 3] started using for the model description of quantum information processes. It is a fixation of the method of transition from input to output states of the qubits set (quantum register). The change of states is carried out on the basis of the arrangement of quantum gates, each of which can be implemented as a sequence of elementary gates for the unitary matrix — a universal set of single-qubit operations [2] which consist of rotation gates for one qubit and the controlled not (CNOT) gate was determined in 1995.

About 20 years ago the first physical prototypes of a quantum computing device [4] providing elementary operations for a small number (3-7) of qubits and available only to their creators appeared. This was followed by a series of experimental devices with significant costs and different levels of success. There are many problems on this difficult path: research and search for adequate technological solutions at the nano-level, the need to maintain the physical environment of a quantum device in a stable state (the lifetime of the elementary base is measured in nanoseconds), obtaining entangled states of qubits in the quantum register, detection and elimination of errors during quantum computing in the physical environment, and many other significant issues [5]. At the same time various research groups developed specifications of quantum algorithms at the logical level to solve tasks of different complexity.

In the last 3 years there has been a significant breakthrough in the technological support of quantum computing — prototypes of quantum processors and programmable simulators are available via the Internet and cloud technologies for testing research solutions appeared. Largest companies (IBM [6, 7], D-Wave [8, 9], Google [10], Intel, Microsoft) have offered limited access to the actual quantum devices at physical level that operate on the basis of different physical principles: superconducting circuits, nuclear magnetic resonance (NMR) [11], ion traps [12], quantum dots systems, photons [13]. The adaptation of standard methods and approaches to the peculiarities of generation and implementation of quantum circuits with a basic set of quantum gates is carried out [14]. An important task is to ensure (if possible) simultaneous (parallel) operation of the gates in order to reduce the size (depth) of the quantum circuit and accordingly time costs. During the pilot operation the researchers note the difficulties in matching the theoretical developments and their direct implementation at the level of the physical device: not being able to provide quantum entanglement for a significant number of qubits [15] at present, practical results on the actual mapping of the CNOT and exchange SWAP gates

in prototypes IBM-Q5 and IBM-Q20 machine [16] specified the need to identify and eliminate purported computing errors in the implementation of these elementary two-qubits operations, the peculiar properties of transfer of the quantum assembler language from one- and two-qubits gates to the quantum computer environment appear when they are translated into control signals on physical devices [17]. One of the most common for the physical implementation of the quantum circuit is to limit the distance of interaction between the qubits of gates (location of interacting qubits in the linear nearest neighbor).

1. Research background: mathematical foundations

1.1 Quantum circuit model

The basis of the model representation of a quantum (vector) computing circuit is a system of components that can be in two states and on information content are related to a two-dimensional Hilbert space with a computational basis:

$$\{|0\rangle, |1\rangle\} : |0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \quad (1)$$

A qubit (a unit vector in a two-dimensional complex vector space) can be in a superposition of base states $|0\rangle$ and $|1\rangle$: $a|0\rangle + b|1\rangle$, where the probability amplitudes a and b are complex numbers such that $|a|^2 + |b|^2 = 1$. $|a|^2$ and $|b|^2$ estimate the probability to obtain one of the states $|0\rangle$ or $|1\rangle$. Description of the system is based on the tensor product of a set of components. For example, a two-qubits system based on the four-component basis is shown in fig. 1.

The gate $C^k\text{NOT}$ implements the function $(x_1, \dots, x_k, y) \rightarrow (x_1, \dots, x_k, y \oplus x_1 \cdot x_2 \cdot \dots \cdot x_k)$ and inverts the target qubit when all control qubits have a state of $|0\rangle$ or $|1\rangle$ (vectors $\mathbf{0}$ or $\mathbf{1}$) depending on the control type. This is implemented by AND operation for control qubits and EXOR operation for the target qubit. Some variants of the $C^k\text{NOT}$ (class k -qubits gates) are shown in table 1. The gate $C^0\text{NOT}$ inverts the target qubit, that is implemented by the EXOR operation with $\mathbf{1}$. The gate $C^1\text{NOT}$ inverts the target qubit if the control qubit is $\mathbf{1}$ and the EXOR operation is performed for both qubits. The gate $C^2\text{NOT}$ (Toffoli) inverts the target qubit when both control qubits have state $\mathbf{1}$, that is implemented by AND operation for both control qubits and EXOR operation for the target qubit. In addition, the SWAP gate (exchange of states of two qubits) which is necessary to bring the quantum circuit to the LNN-notation is also shown in table 1.

1.2. Graphical notation for a quantum circuit

Deutsch graphical notation [2] is used to visualize basic quantum gates in the works of many authors and in our applications. The qubits in this notation are represented by the horizontal lines on which the gates are placed. Different gates are depicted as circles (sometimes squares) with corresponding symbols. The quantum computing process evolves from left to right. Control gates are placed on the target qubit, from which a vertical control line is drawn to the corresponding control qubits which is marked on it by symbols "•" ("control by 1"). The EXOR operation in this notation denotes a symbol "⊕".

$$\begin{aligned}
 |00\rangle &= |0\rangle|0\rangle = |0\rangle \otimes |0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \\
 |01\rangle &= |0\rangle|1\rangle = |0\rangle \otimes |1\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \\
 |10\rangle &= |1\rangle|0\rangle = |1\rangle \otimes |0\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \\
 |11\rangle &= |1\rangle|1\rangle = |1\rangle \otimes |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}
 \end{aligned}$$

Fig. 1. The basis for a quantum register from two qubit

Table 1

The use of lexemes for basic gates

Gate	Number of qubits and graphical notation	Lexemes for the coefficients of a qubit
NOT (C ⁰ NOT)	1 	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \otimes \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} b \\ a \end{bmatrix}$
C ¹ NOT	2 	$\begin{bmatrix} a \\ c \end{bmatrix} \otimes \begin{bmatrix} b \\ d \end{bmatrix} \xrightarrow{\text{CNOT}} \begin{bmatrix} ab \\ cd \\ cb \\ ad \end{bmatrix} = \begin{bmatrix} ab \\ cd \end{bmatrix} \otimes \begin{bmatrix} 0 \\ d \end{bmatrix} + \begin{bmatrix} a \\ c \end{bmatrix} \otimes \begin{bmatrix} b \\ 0 \end{bmatrix}$
C ² NOT (Toffoli)	3 	$\begin{bmatrix} a \\ c \end{bmatrix} \otimes \begin{bmatrix} b \\ d \end{bmatrix} \otimes \begin{bmatrix} e \\ f \end{bmatrix} \xrightarrow{\text{Tof}} \begin{bmatrix} abe \\ abf \\ ade \\ adf \\ cbe \\ cbf \\ cde \\ cdf \end{bmatrix} = \begin{bmatrix} abe \\ abf \\ ade \\ adf \\ cbe \\ cbf \\ cde \\ cdf \end{bmatrix} = \begin{bmatrix} a \\ c \end{bmatrix} \otimes \begin{bmatrix} 0 \\ d \end{bmatrix} \otimes \begin{bmatrix} 0 \\ f \end{bmatrix} + \begin{bmatrix} a \\ c \end{bmatrix} \otimes \begin{bmatrix} b \\ 0 \end{bmatrix} \otimes \begin{bmatrix} e \\ 0 \end{bmatrix} + \begin{bmatrix} a \\ c \end{bmatrix} \otimes \begin{bmatrix} 0 \\ d \end{bmatrix} \otimes \begin{bmatrix} e \\ 0 \end{bmatrix} + \begin{bmatrix} a \\ c \end{bmatrix} \otimes \begin{bmatrix} b \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ f \end{bmatrix}$
SWAP (through 3 C ¹ NOT)	2 	$\begin{bmatrix} a \\ c \end{bmatrix} \otimes \begin{bmatrix} b \\ d \end{bmatrix} \xrightarrow{\text{SWAP}} \begin{bmatrix} ba \\ da \\ dc \end{bmatrix} = \begin{bmatrix} b \\ d \end{bmatrix} \otimes \begin{bmatrix} a \\ c \end{bmatrix}$

1.3. Lexical verification used by us

Review and analysis of transformations on the qubits is carried out on the basis of the space of lexemes [18] for the coefficients of a qubit in the computational basis of the formula (1). The results of the basic gates [2] for the qubit and the quantum register of two and three qubits are presented in table 1.

1.4. Reed—Muller expressions

The article [19] considers the possibility of comparison between quantum logical operations and certain forms of classical Boolean logic on the basis of Reed—Muller decomposition (RM). This idea allows the transformation of logical circuits into their quantum analogs. RM logic is based on algebra modulo 2.

A Boolean function f of n variables $f: \{0, 1\}^n \rightarrow \{0, 1\}$ can be represented in RM logic where in formula n variables x are in direct or additional forms:

$$f(x_0, \dots, x_{n-1}) = \bigoplus_{i=0}^{2^n-1} b_i z_i, \quad z_i = \prod_{m=0}^{n-1} x_m^{i_m} \quad (2)$$

The symbols $b_i \in \{0, 1\}$, x_m and i_m are binary numbers; z_i — terms multiplication; b_i determine whether the product of terms is represented or not. In formula (2) the symbol \oplus denotes the EXOR operation, the multiplication symbol Π denotes the AND operation. If n variables are not written simultaneously in direct or in additional forms then it is a fixed polarity of the generalized Reed—Muller decomposition (FPRM, fixed polarity Reed—Muller expressions). As a result, the task of synthesis and subsequent minimization of the quantum logic circuit can be solved on the basis of the RM logic.

2. A common description of our approach

The need for research on the automated generation of quantum circuits for different types of quantum computing has recently become apparent [20]. In this context, the quantum compilation procedure provides obtaining and arrangement of a sequence of elementary gates for solving a specific computational task. Quantum compilation is the beginning of the procedure of quantum computing decomposition into an effective quantum circuit consisting of elementary unitary operations.

2.1. General information on quantum compilation

Different approaches are proposed for quantum circuit generation. The most used method is based on the decomposition of the matrix of the unitary operator [2]. The main ideas and different variants of decomposition are well described [21, 22]. Therefore, equivalent quantum circuits can be proposed for the implementation of quantum computing. A significant problem is the exponential increase in the size of the state space of the quantum register with the appearance of additional qubits in it, which complicates the implementation of the quantum circuit at the logical and even more physical levels. A brief overview of the articles on the discussion of this problem for the set of quantum logic gates with multiple control C^k NOT is presented in [23–25].

An important component in the synthesis of quantum circuits is their implementation in the architecture of the linear nearest neighbor (LNN) [26–32]. This attention is explained by the fact that the physical implementation of the quantum computer is more simple to organize the interaction between adjacent (nearest) qubits with the aim of obtaining lower level errors than when placing qubits at a significant distance from each other. Currently used prototypes with ion traps and NMR operate on the basis of qubits placement in the LNN. The representation of the quantum circuit specification in the LNN-notation can help in solving the problem of providing a scalable quantum architecture.

2.2. Our version of the common architecture for the quantum compilation system

In our research, we propose the automation of the design process of quantum circuit specifications based on the connection of classical mathematical foundations with a common basis for quantum algorithms based on multiple control C^k NOT set taking into account LNN-architecture. Our approach assumes a comprehensive view of the procedure for generating the quantum circuit specifications in automatic mode.

Our toolkit provides automatic compilation of all possible variants of quantum circuits according to Reed—Muller expressions including automatic modes for: transition to linearly nearest neighbor while minimizing the number of

SWAP gates based on our scalable templates, automatic statistics collection, visualization, lexical verification of the equivalence of the results for quantum circuits compilation. This allows you to select the appropriate options from the all automatically generated specifications according to different criteria.

For the versatile analysis, automatic generation and use of the quantum circuit model the authors propose a multi-level scheme of synthesis flows, taking into account various aspects of the circuit implementation including in the LNN-notation (fig. 2): the transition (compilation) from the mathematical description to the logical level of quantum circuit specifications is carried out, after which it becomes possible to adapt to the conditions of physical realization. The authors develop a set of applications for the logical level.

If some gate interacts in a quantum circuit at a logical level with multiple non-adjacent qubits then for the transition to LNN-notation a simple rule applies for changing the order of qubits lines by primitive arrangement of the quantum gates SWAP. One SWAP is placed on the corresponding lines before the gate and the second is placed symmetrically after the gate to restore the original order of the qubit lines. The number of arranged pairs of SWAP gates is determined by the number of shifted lines of qubits in the process of bringing this gate to the LNN-notation and is large enough (there are often mutually absorbing among them). After that it begs the transition to the minimum LNN-specification (MLNN-specification) [33], in which the minimization of the number of SWAP gates is carried out on the basis of taking into account their mutual action when working together or in parallel with the help of specially selected templates [34].

According to the results of practical research [14], it is noted that an essential factor in the physical implementation of quantum computing is the search for effective (at quantum cost) strategies for placing gates in LNN-notation with an optional desire to minimize the gates of qubits permutations SWAP. Therefore, we propose to consider all potential variants for quantum circuits with the collection of statistical indicators, leaving the choice of a suitable specification from proposed to the stage of physical implementation in a particular environment.

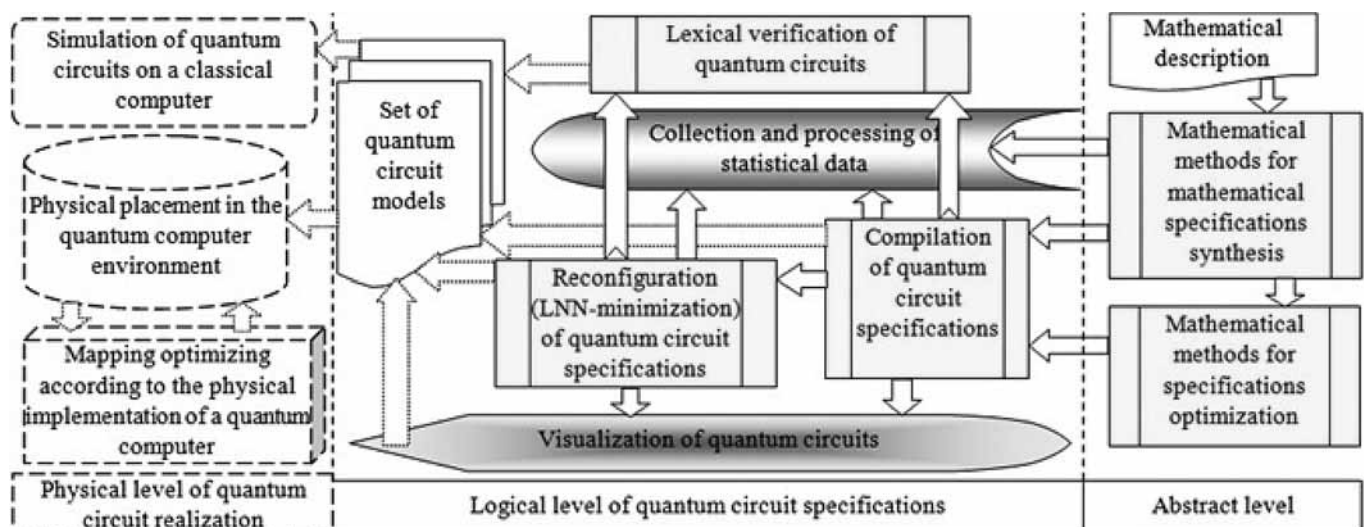


Fig. 2. The main stages in the compilation of equivalent quantum circuits specifications

Each of the generated quantum circuit specifications can be represented graphically using a separate visualization application. To check the equivalence of specifications obtained automatically in the process of transformations, we have implemented a special application that carries out the procedure of lexical analysis of the states of the qubits at the output of each quantum circuit.

Multistage automatic quantum compilation (fig. 2) provides preparation of quantum circuit specifications for different levels of implementation and perception. A step-by-step translation of the specification from an abstract description (strict mathematical semantics) into model representations at different levels (specifications of the quantum circuit in different notations) is carried out, which allows researchers to proceed further to the physical implementation (simulation) of the quantum circuit in accordance with the principles of functioning of the real computing device (analog quantum computer). The process of quantum compilation is accompanied by the automatic collection of statistical data in Excel format (fig. 3) and the formation of protocols of lexical analysis for the verification of all received specifications in different notations.

2.3. Metrics for quantum circuit

Various metrics [25, 35] have been proposed for a comparative evaluation of the efficiency of the generated quantum circuits for the implementation of the quantum computing process. In our research the following evaluation characteristics are used: quantum gates number (QGN), quantum circuit depth (QCD) when parallel arrangement of simultaneously running gates to account for the reduction of time costs, quantum cost (QCost) for assessing the complexity of the physical implementation of the gate, the nearest neighborhood complexity (NNC) by distance of interacting qubits in the gate (LNN assumes NNC = 0).

QCost for basic gates in statistical analysis can be specified in the Excel table parametrically because it depends on the complexity of the implementation of each type of gates in a particular nanotechnology. Usually single-qubit gate NOT is selected as the unit of quantum cost. In this

case the QCost of two-qubit gate SWAP is determined on the basis of a standard decomposition into single-qubit gates with a similar approach and for the class C^k NOT.

In our case the calculation of the QCost is as follows: $KC_{NOT} = KC_{CNOT} = 1$, $KC_{SWAP} = 3$ ($KC_{SWAP} = 3 * KC_{CNOT}$ according to the representation of the SWAP gate by means of CNOT gates in table 1), $KC_{CNOT}^k = 5^{k-1}$.

3. Results

To test and clarify the features of the proposed architecture of the quantum circuit generation system, we have implemented an open set of applications that allows us to synthesize and analyze various combinations of quantum circuit specifications.

Let us consider the first example of generation the set of quantum circuits for a Boolean function for a given truth table (left in table 2 and in fig. 3) of 3 logical variables.

Obtaining an extensive experimental set of specifications is carried out on the example of the considered Reed—Muller expressions [34], which are used in other researches [36—38]. In this case, the synthesis of the corresponding quantum circuit is possible on the basis of automatic mapping of the FPRM-representation fragments in notation of quantum gates.

According to formula (2) the set of all Boolean functions of 3 variables can be represented as $(x_k$ and $b_i \in \{0, 1\}$, $0 \leq k < n$, $0 \leq i \leq 2^n - 1$, $n = 3$, the coefficients b_i determine whether the minterm is represented in the expression or not):

$$f(x_0, x_1, x_2) = b_0 \oplus b_1 x_0 \oplus b_2 x_1 \oplus b_3 x_1 x_0 \oplus b_4 x_2 \oplus b_5 x_2 x_0 \oplus b_6 x_2 x_1 \oplus b_7 x_2 x_1 x_0. \quad (3)$$

Columns are formed in the table on fig. 3: "Polarity" — polarity number, "NOT-x" — number of NOT for changing qubits $|x_0\rangle \dots |x_{n-1}\rangle$, where n is the control qubits number, columns "k-control_qubit" ($k \in \{1, n\}$) are the numbers of C^k NOT, "NOT-f" is the number of NOT which acting on the functional qubit $|f\rangle$, "C^gates" is the total number of C^k NOT and NOT, "SWAPs" is the number of SWAPs to

M10																					
fx =F10*\$F\$12^2+G10*\$F\$12^1+H10*\$F\$12^0+(E10+I10)*\$G\$12+K10*\$H\$12																					
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1						Control qubits number									Minimization result				Reduction result		
2	x	f	Polarity	NOT-x	3	2	1	NOT-f	C^gates	SWAPs	C^gates+SWAPs	QCost	Depth	SWAPs	C^gates+SWAPs	Qcost	SWAPs	C^gates+SWAPs	Qcost		
3	000	1	0	0	1	3	2	1	7	10	17	73	14	8	15	67	20,00%	11,76%	8,22%		
4	001	0	1	2	1	2	2	0	7	4	11	51	10	4	11	51	0,00%	0,00%	0,00%		
5	010	1	2	2	1	2	0	1	6	4	10	50	9	4	10	50	0,00%	0,00%	0,00%		
6	011	1	3	4	1	1	1	1	8	2	10	42	7	2	10	42	0,00%	0,00%	0,00%		
7	100	0	4	2	1	2	2	0	7	12	19	75	15	8	15	63	33,33%	21,05%	16,00%		
8	101	0	5	4	1	1	1	0	7	4	11	47	9	4	11	47	0,00%	0,00%	0,00%		
9	110	1	6	4	1	1	1	1	8	6	14	54	9	4	12	48	33,33%	14,29%	11,11%		
10	111	1	7	6	1	0	1	1	9	2	11	39	6	2	11	39	0,00%	0,00%	0,00%		
11	Gate:	C^kNOT	NOT	SWAP					Minimum:				Maximum:								
12	Qcost:		5	1	3					6	2	10	39	6	2	10	39	33,33%	21,05%	16,00%	

Fig. 3. Statistics analysis for equivalent quantum circuits compiled for the logic function of 3 variables

A comparison of the FPRM set and compiled quantum circuits

No.	x_0	x_1	x_2	$f(x_0, x_1, x_2)$	FPRM expressions of all polarities	Quantum circuit for FPRM and its LNN-notation
0	0	0	0	1	$f(x_0, x_1, x_2) = x_0x_1x_2 \oplus x_0x_1 \oplus x_1x_2 \oplus x_0x_2 \oplus x_2 \oplus x_0 \oplus 1$	Fig. 4, a
1	0	0	1	0	$f(x_0, x_1, x_2) = x_0x_1\bar{x}_2 \oplus x_1\bar{x}_2 \oplus x_1 \oplus x_0\bar{x}_2 \oplus \bar{x}_2$	Fig. 4, b
2	0	1	0	1	$f(x_0, x_1, x_2) = x_0\bar{x}_1x_2 \oplus x_0\bar{x}_1 \oplus \bar{x}_1x_2 \oplus 1$	Fig. 4, c
3	0	1	1	1	$f(x_0, x_1, x_2) = x_0\bar{x}_1\bar{x}_2 \oplus \bar{x}_1\bar{x}_2 \oplus \bar{x}_1 \oplus 1$	Fig. 4, d
4	1	0	0	0	$f(x_0, x_1, x_2) = \bar{x}_0x_1x_2 \oplus \bar{x}_0x_1 \oplus x_1 \oplus \bar{x}_0x_2 \oplus \bar{x}_0$	Fig. 4, e
5	1	0	1	0	$f(x_0, x_1, x_2) = \bar{x}_0x_1\bar{x}_2 \oplus x_1 \oplus \bar{x}_0\bar{x}_2$	Fig. 4, f
6	1	1	0	1	$f(x_0, x_1, x_2) = \bar{x}_0\bar{x}_1x_2 \oplus \bar{x}_0\bar{x}_1 \oplus \bar{x}_1 \oplus 1$	Fig. 4, g
7	1	1	1	1	$f(x_0, x_1, x_2) = \bar{x}_0\bar{x}_1\bar{x}_2 \oplus \bar{x}_1 \oplus 1$	Fig. 4, h

convert the circuit in terms of nearest neighbor without regard to possible minimization, "C[^]gates + SWAPs" is the total number of gates, "QCost" is the quantum cost of circuit (the calculation formula is given for cell M10 and can be changed depending on the technological implementation and physical processes).

According to the general formula (3) FPRM of all polarities are automatically generated in the mathematical form of the record (table 2) and the corresponding specifications of quantum circuits are compiled. The basic quantum circuits according to 8 polarities of FPRM are shown in fig. 4 together with their respective LNN-notations.

Reciprocal pairs (highlighted in fig. 4 by rectangles) appear in LNN-notation in the SWAPs arrangement. Their removal allows to reduce the QCD which is shown in fig. 3 under 'Minimization result'. From the collected statistics it is seen that according to different characteristics the minimum values are for different quantum circuits. This allows you to consider different options further at the level of physical implementation taking into account the feature of the analog quantum computer environment.

The following example represents the result of quantum compilation by the truth table for 8 variables. Mathematical forms of record are generated by the formula (2). Quantum compilation provides for them an automatic receive of the quantum circuit specifications in LNN- and MLNN-notations.

The software modules implemented by us allow for automatic multi-stage minimization of the quantum circuit with repeatedly (if necessary) practical applications of individual stages again. At the first stage of minimization, the initial removal of paired SWAP gates occurs during the conversion of the quantum circuit to LNN-notation. At the second stage, multi-pass minimization is carried out using scalable templates [34], which leads to a significant reduction in SWAP gates with a decrease in QGN and QCost. In the third stage, parallel arrangement of the gates with the minimization of the quantum circuit for QCD is executed to reduce the time of operation of the circuit at the expense of parallelism.

An example of the quantum circuit with a direct SWAP gates arrangement for automatic transition to LNN-notation is shown in fig. 5 (see the third side of cover). The

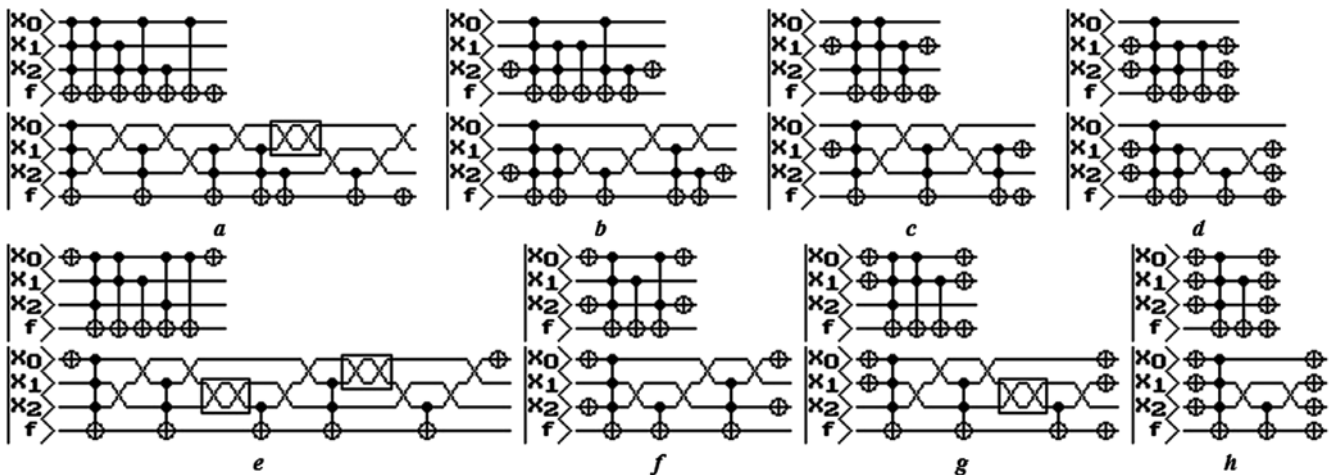


Fig. 4. Specifications of equivalent quantum circuits (initial and reduced to LNN-notation) compiled in accordance with FPRM

=C15*\$P\$1^7+D15*\$P\$1^6+E15*\$P\$1^5+F15*\$P\$1^4+G15*\$P\$1^3+H15*\$P\$1^2+I15*\$P\$1^1+J15*\$P\$1^0+(B15+K15)*\$N\$1+M15*\$L\$1																							
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	
1	Control qubits number										SWAP: 3		NOT: 1		C ⁷ NOT: 5		Minimization result			Reduction result			
2	Polarity	NOT-x	8	7	6	5	4	3	2	1	NOT-f	C ⁷ gater	SWAPs	C ⁷ gater+SWAPs	QCost	SWAPs	C ⁷ gater+SWAPs	Qcost	SWAPs	C ⁷ gater+SWAPs	Qcost		
3	0	0	1	4	14	37	41	32	14	3	1	147	2076	2223	219727	1026	1173	216577	50,58%	47,23%	1,43%		
4	1	2	1	3	13	29	36	28	18	4	1	135	1892	2027	194723	870	1005	191657	54,02%	50,42%	1,57%		
5	40	4	1	2	15	32	39	31	13	3	1	141	1860	2001	187553	892	1033	184649	52,04%	48,38%	1,55%		
6	41	6	1	1	14	27	37	30	17	5	1	139	1848	1987	165391	798	937	162241	56,82%	52,84%	1,90%		
7	42	6	1	3	13	34	42	38	16	6	0	159	2134	2293	199569	976	1135	196095	54,26%	50,50%	1,74%		
8	131	6	1	3	15	23	23	16	9	2	1	99	1322	1421	193545	632	731	191475	52,19%	48,56%	1,07%		
9	132	4	1	4	14	31	34	20	7	1	1	117	1656	1773	213509	646	763	210479	60,99%	56,97%	1,42%		
10	133	6	1	3	15	23	24	16	6	2	0	96	1268	1364	193492	574	670	191410	54,73%	50,88%	1,08%		
11	140	6	1	3	13	24	22	21	4	2	1	97	1424	1521	188201	574	671	185651	59,69%	55,88%	1,35%		
12	141	8	1	2	13	18	17	15	7	2	0	83	1090	1173	167065	520	603	165355	52,29%	48,59%	1,02%		
13	142	8	1	4	12	24	20	17	7	2	1	96	1284	1380	199948	640	736	198016	50,16%	46,67%	0,97%		
14	168	6	1	1	14	31	30	22	9	2	1	117	1464	1581	165621	678	795	163263	53,69%	49,72%	1,42%		
15	169	8	1	0	14	28	28	22	12	4	0	117	1496	1613	147985	650	767	145447	56,55%	52,45%	1,72%		
16	170	8	1	2	13	30	34	19	8	1	1	117	1562	1679	178211	702	819	175631	55,06%	51,22%	1,45%		
17	194	6	1	5	14	24	26	17	5	0	1	99	1296	1395	222595	580	679	220447	55,25%	51,33%	0,96%		
18	195	8	1	4	14	17	23	12	4	2	1	86	1036	1122	201314	548	634	199850	47,10%	43,49%	0,73%		
19	196	6	1	5	14	30	33	16	4	1	1	111	1452	1563	227659	598	709	225097	58,82%	54,64%	1,13%		
20	254	14	1	5	20	28	29	19	8	0	1	125	1524	1649	244977	764	889	242697	49,87%	46,09%	0,93%		
21	255	16	1	4	20	29	36	28	12	4	1	151	1900	2051	232231	922	1073	229297	51,47%	47,68%	1,26%		
23	Minimum:		1	0	10	17	17	10	2	0		083	1036	1122	147985	520	603	145447	42,40%	45,86%	0,73%		
24	Maximum:		1	8	20	39	42	38	23	8		171	2284	2455	274406	1034	1203	271598	60,99%	56,97%	1,90%		

Fig. 7. Analysis of statistics fragment for equivalent quantum circuits compiled for a logical function of 8 variables (minimum and maximum values are given for all 256 polarities)

resulting quantum circuit in MLNN-notation is shown in fig. 6 (see the third side of cover). In both figures, the quantum circuits are represented in several bars with a continuation marked with 4 lines to compare the fragments with a significant reduction in the number of SWAP gates. Note that the penultimate part of the second fragment after automatic minimization led to an increase in the number of SWAP gates but the depth of the fragment remained unchanged (9). Common QCD in MLNN-notation for FPRM 141 is equal 214.

The results of the collection of statistical data on the parameters of the circuits in the process of minimization are automatically transferred to Excel (the right part in fig. 7 represents the result of the reduction according to the criteria).

In the second example quantum circuits with different polarities turned out to be minimal according to different criteria. By QGN (without SWAP gates): circuit for polarity 141 is minimum. For initial and final QCost: circuit for polarity 169 (note that it is the only one where C⁷NOT gate is missing).

Maximum ratio when removing SWAPs: circuit for polarity 132. By QGN in MLNN-notation, SWAPs number and QCD: circuit for polarity 141.

The result of the procedure of automatically obtaining a set of equivalent quantum circuit specifications and a set of LNN-specifications at the logical level is demonstrated by examples. After the minimization of quantum circuits their MLNN-specifications and the final statistical description are formed. The minimization by QGN, QCD and QCost is significantly different for different variants of the quantum circuit depending on its structural composition. The choice of a specific quantum circuit depends on the nanotechnology

in which it is supposed to be implemented, as well as on the type of minimization used.

Conclusion

Quantum computing devices as physical equipment have received their real embodiment — we are on the threshold of their availability in the next decade, albeit for a limited number of qubits measured today by dozens. IBM has launched the IBM Q Experience quantum cloud platform into open access, including a 20-qubit quantum device [39], according to preliminary information, not all of qubits can be in an entangled state, and the retention time of the quantum state of the qubit is estimated to be about 75 μs. The researchers noted that the improvement of quantum technologies will lead to a significant increase in the capacity of the quantum register [40], consequently, there is an urgent need for computer specialists [41], prepared for practical activities on the use of theoretical developments on real quantum computers. Any theoretical and practical experience in the field of quantum computing will be in demand.

In this article we presented our approach and practical experience in automatic compilation of quantum circuits specifications for a relatively small task of preparation at the logical level of the scheme of quantum gates placement in accordance with the presented logical function by means of FPRM. The results of adaptation in accordance with the requirements of the linear nearest neighbor as well as automatic minimization of the number of SWAP gates on the basis of a set of templates and the formation of appropriate statistics are demonstrated.

References

1. Feynman R. P. Quantum Mechanical Computers, *Found. Phys.*, 1986, vol. 16, no. 6, pp. 507–531.
2. Barenco A., Bennett C. H., Cleve R., DiVincenzo D. P., Margolus N., Shor P., Sleator T., Smolin J., Weinfurter H. Elementary gates for quantum computation, *Phys. Rev. A*, 1995, vol. 52, no. 5, pp. 3457–3467.
3. Nielsen M. A., Chuang I. L. *Quantum Computation and Quantum Information*, Cambridge, Cambridge Univ. Press, 2001, 674 p.
4. Vandersypen L. M. K., Steffen M., Breyta G., Yannoni C. S., Sherwood M. H., Chuang I. L. Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance, *Nature*, 2001, vol. 414, pp. 883–887.
5. Li G., Ding Y., Xie Y. Tackling the Qubit Mapping Problem for NISQ-Era Quantum Devices, *Proceedings of the Twenty-Fourth ASM Int. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS'19)*, Providence, RI, USA, Apr. 13–17, 2019, 14 p.
6. Conover E. Quantum computers are about to get real: As the first qubit-based machines come online, scientists are just beginning to imagine the possibilities, *Science News*, 2017, vol. 191, no. 13, available at: <https://www.sciencenews.org/article/quantum-computers-are-about-get-real>
7. Conover E. Quantum computers take a step forward with a 50-qubit prototype, *Science News*, 2017, available at: <https://www.sciencenews.org/blog/science-ticker/quantum-computers-take-step-forward-50-qubit-prototype>
8. Vert D., Sirdrey R., Louise S. On the limitations of the Chimera graph topology in using analog quantum computers, *Proceedings of the 16th Conf. on Computing Frontiers (CF'19)*, Alghero, Italy, Apr. 30 – May 2, 2019, 4 p., DOI: 10.1145/3310273.3322830.
9. Mueller F., Byrd G., Dreher P. Programming quantum computers: a primer with IBM Q and D-Wave exercises, *Proceedings of the 24th Symposium on Principles and Practice of Parallel Programming (PPoPP'19)*, Washington, District of Columbia, USA, Feb. 16–20, 2019, 1 p., DOI: 10.1145/3293883.3302578.
10. Arute F., Arya K., Babbush R. et al. Quantum supremacy using a programmable superconducting processor, *Nature*, 2019, vol. 574, pp. 505–510, available at: <https://www.nature.com/articles/s41586-019-1666-5>
11. Kane B. A silicon-based nuclear spin quantum computer, *Nature*, 1998, vol. 393, pp. 133–137.
12. Friis N., Marty O., Maier C., Hempel C., Holzäpfel M., Jurcevic P., Plenio M. B., Huber M., Roos C., Blatt R., Lanyon B. Observation of Entangled States of a Fully Controlled 20-Qubit System, *Phys. Rev. X*, 2018, vol. 8, article 021012, 20 p., DOI: 10.1103/PhysRevX.8.021012.
13. Alonso Calafell I., Cox J. D., Radonjic M., Saavedra J. R. M., Garcia de Abajo F. J., Rozema L. A., Walther P. Quantum computing with graphene plasmons, *Nature: npj Quantum Information*, 2019, vol. 5, no. 1, DOI: 10.1038/s41534019-0150-2.
14. Krylov G., Lukac M. Quantum Encoded Quantum Evolutionary Algorithm for the Design of Quantum Circuits. *Proceedings of the 16th ASM Conf. on Computing Frontiers (CF'19)*, Alghero, Italy, Apr. 30 – May 2, 2019, 6 p., DOI: 10.1145/3310273.3322826.
15. Saggio V., Dimic A., Greganti C., Rozema L. A., Walther P., Dakic B. Experimental few-copy multipartite entanglement detection, *Nature Phys.*, 2019, vol. 15, no. 9, pp. 935–940, DOI: 10.1038/S41567-019-0550-4.
16. Tannu S. S., Qureshi M. K. Not All Qubits Are Created Equal: A Case for Variability-Aware Policies for NISQ Era Quantum Computers, *Proceedings of the Twenty-Fourth ASM Int. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS'19)*, Providence, RI, USA, Apr. 13–17, 2019, 13 p., DOI: 10.1145/3297858.3304007.
17. Shi Y., Leung N., Gokhale P., Rossi Z., Schuster D. I., Hoffmann H., Chong F. T. Optimized Compilation of Aggregated Instructions for Realistic Quantum Computers. *Proceedings of the Twenty-Fourth ASM Int. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS'19)*, Providence, RI, USA, Apr. 13–17, 2019, 14 p., DOI: 10.1145/3297858.3304018.
18. Kalmaychikov V. A., Krasilnikov A. V., Matveeva I. V. Quantum circuits specifications design with lexical verification. *Proceedings of the XVIII Int. Conf. on Soft Computing and Measurements (SCM'2015)*, Saint-Petersburg, Russia, May 19–21, 2015, pp. 87–89.
19. Younes A., Miller J. Representation of Boolean Quantum Circuits as Reed-Muller Expansions. *Int. J. Electron.*, 2004, vol. 91, no. 7, pp. 431–444.
20. Soeken M., Haener T., Roetteler M. Programming Quantum Computers Using Design Automation. *Proceedings of the Design, Automation and Test in Europe (DATE'2018)*, Dresden, Germany, March 19–23, 2018, 10 p., DOI: 10.23919/DATE.2018.8341993.
21. Paige C. C., Wie M. History and generality of the CS decomposition, *Linear Algebra and its Applications*, 1994, vol. 208–209, pp. 303–326.
22. Stewart G. W. Computing the CS Decomposition of a Partitioned orthogonal Matrix, *Numerische Mathematik*, 1982, vol. 40, pp. 297–306.
23. Arpita P. M., Datta K., Vemula R., Sengupta I. Optimization of reversible circuits using triple-gate templates at quantum gate level, *Proceedings of the Int. Conf. on Electronic Design, Computer Networks & Automated Verification (EDCAV'2015)*, Shillong, India, Jan. 29–30, 2015, 5 p.
24. Wille R., Soeken M., Otterstedt C., Drechsler R. Improving the mapping of reversible circuits to quantum circuits using multiple target lines, *Proceedings of the 18th Asia and South Pacific Design Automation Conf. (ASP-DAC)*, Yokohama, Japan, Jan. 22–25, 2013, pp. 145–150.
25. Szyptowski M., Kerntopf P. Reducing Quantum Cost in Reversible Toffoli Circuits, *Proceedings of Reed-Muller 10th Workshop*, Tuusula, Finland, May 25–26, 2011, pp. 127–136.
26. Alfailakawi M. G., Alterkawi L., Ahmad I., Hamdan S. Line Ordering of Reversible Circuits for Linear Nearest Neighbor Realization, *Quantum Information Processing*, 2013, vol. 12, no. 10, pp. 3319–3339.
27. Alfailakawi M. G., Alterkawi L., Ahmad I. Harmony Search Algorithm for 2D Neighbor Quantum Circuits Realization, *Int. J. Expert Systems with Applications*, 2016, vol. 61, pp. 16–27.
28. Liu Y., Guan Z.-J., Cheng X.-Y., Tan Y.-Y. Linear Nearest Neighbor Quantum Circuit Synthesis and Optimization Based on the Matrix, *Tien Tzu Hsueh Pao/Acta Electronica Sinica*, 2018, vol. 46, no. 3, pp. 688–694.
29. Tan Y.-Y., Cheng X.-Y., Guan Z.-J., Liu Y. Multi-strategy based quantum cost reduction of linear nearest-neighbor quantum circuit, *Quantum Information Processing*, 2018, vol. 17, no. 3, pp. 14.
30. Cheng X.-Y., Guan Z.-J., Xu H., Tan Y.-Y. The nearest neighbor arrangement of quantum circuits based on MCT reversible circuits, *Tien Tzu Hsueh Pao/Acta Electronica Sinica*, 2018, vol. 46, no. 8, pp. 1891–1897.
31. Kole A., Datta K., Sengupta I. A New Heuristic for N-Dimensional Nearest Neighbor Realization of a Quantum Circuit, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2017, vol. 37, no. 1, pp. 182–192.
32. Marbaniang L., Kole A., Datta K., Sengupta I. Design of Efficient Quantum Circuits Using Nearest Neighbor Constraint in 2D Architecture, *Proceedings of 9th Int. Conf. of Reversible Computation (RC'2017)*, Kolkata, India, Jul. 6–7, 2017, 6 p.
33. Shafaei A., Saeedi M., Pedram M. Optimization of quantum circuits for interaction distance in linear nearest neighbor architectures, *Proceedings of 50th Design Automation Conf. (DAC'2013)*, Austin, Texas, USA, Jun. 2–6, 2013, pp. 41–46.
34. Matveeva I., Kalmaychikov V., Dorokhov A. Model design of quantum logic circuits based on Reed–Muller expressions, *Proceedings of IEEE Int. Russian Automation Conf. (RusAutoCon'2018)*, Sochi, Russia, Sept. 9–16, 2018, 6 p.
35. Ali M. B., Hirayama T., Yamanaka K., Nishitani Y. Quantum Cost Reduction of Reversible Circuits Using New Toffoli Decomposition Techniques, *Proceedings of Int. Conf. on Computational Science and Computational Intelligence (CSCI'2015)*, Jakarta, Indonesia, Aug. 24–26, 2015, pp. 59–64, DOI: 10.1109/CSCI.2015.41.
36. Wang L., Xia Y., Chen X., Almaini A. E. A. Reed-Muller function optimization techniques with onset table, *J. of Zhejiang University-SCIENCE C (Computers & Electronics)*, 2011, vol. 12, no. 4, pp. 288–296.
37. He Z., Xiao L., Gu F., Wang X. An efficient and fast polarity optimization approach for mixed polarity Reed–Muller logic circuits, *Frontiers of Computer Science*, 2017, vol. 11, no. 4, pp. 728–742, DOI: 10.1007/s11704-016-5259-2.
38. Suprun V. P. The Complexity of Boolean Functions in the Reed-Muller Polynomials Class, *Automatic Control and Computer Sciences*, 2017, vol. 51, no. 5, pp. 285–293.
39. International Business Machines Corporation. IBM Q System One, 2019, available at: <https://www.research.ibm.com/ibm-q/system-one/>
40. Butko A., Michelogiannakis G., Donofrio D., Shalf J. Extending Classical Processors to Support Future Large Scale Quantum Accelerators: Invited Talk Abstract., *Proceedings of the 16th Conf. on Computing Frontiers (CF'19)*, Alghero, Italy, Apr. 30 – May 2, 2019, 1 p., available at: <https://doi.org/10.1145/3310273.3324898>.
41. Chong F. T. Quantum Computing is Getting Real: Architecture, PL, and OS Roles in Closing the Gap between Quantum Algorithms and Machines, *Proceedings of the Twenty-Third Int. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS'18)*, Williamsburg, VA, USA, March 24–28, 2018, 1 p., DOI: 10.1145/3296957.3177152.

Е. С. Крюкова, адъюнкт (аспирант), e.krukovaa69@yandex.ru,
И. Б. Паращук, д-р техн. наук, проф., shchuk@rambler.ru, Военная академия связи
имени Маршала Советского Союза С. М. Буденного, г. Санкт-Петербург

Программное средство для моделирования смены состояний показателя качества электронной библиотеки: метод определения ключевых элементов модели

Предложен подход к математическому описанию и программной реализации алгоритма определения элементов матрицы переходных вероятностей как ключевого объекта марковской вероятностно-временной модели процесса смены состояний среднего времени ожидания доступа пользователей к электронному информационному ресурсу. С точки зрения математической реализации подход основан на положениях теории массового обслуживания, опирается на исходные данные об интенсивности заявок на информационный ресурс электронной библиотеки и параметры ее очереди на обслуживание. Программная реализация учитывает особенности и достоинства современных языков программирования.

Ключевые слова: программное средство, модель, электронная библиотека, показатель качества, интенсивность, матрица вероятности перехода, состояние

Введение

Электронная библиотека является одним из решений проблемы долговременного хранения информационных ресурсов, их организации, реализации способов и средств доступа к ним пользователей в условиях быстрых эволюционных изменений в области современных технологий передачи данных [1–3]. Электронная библиотека — сложная информационно-техническая система, представляющая собой упорядоченную коллекцию разнородных электронных документов и снабженная средствами навигации и поиска [1, 2].

Электронные библиотеки приобретают всё большую популярность благодаря тому, что их применяют в широком спектре информационно-справочных услуг, они предоставляют доступ к большому объёму информационных ресурсов и их развертывание не требует значительных затрат времени и средств. Это емкие по памяти и высокопроизводительные информационные системы для обеспечения поиска и управляемого доступа по информационно-телекоммуникационным сетям к электронным документам без индивидуального носителя, профессиональным базам данных, информационным справочным и поисковым системам, а также иным информационным ресурсам [4–6].

Одним из возможных и широко распространенных математических инструментальных средств для вероятностно-временного моделирования процесса смены состояний показателей качества сложных информационно-технических систем, например, таких как электронные библиотеки, являются управляемые цепи Маркова (УЦМ). При этом необходимыми

исходными данными для адекватной работы модели в данной постановке являются: система показателей качества (ПК) электронной библиотеки (ЭБ); требования к этим ПК; возможные значения, принимаемые данными параметрами (ПК) в процессе функционирования ЭБ; данные об управляющих и внешних воздействиях на ЭБ, характеризующие для каждого ПК (для модели в виде УЦМ) элементами матрицы одношаговых переходных вероятностей (ОПВ), переводящими этот ПК из одного состояния в другое в процессе функционирования ЭБ.

Вариант определения элементов матрицы ОПВ для ПК $\bar{t}_{ок}(k)$ — среднего времени ожидания доступа легальных (авторизированных) пользователей к информационному ресурсу ЭБ на k -м шаге ее функционирования (времени ожидания предоставления ЭБ конкретной услуги), а также возможный подход к его программной реализации предложены в данной статье.

Постановка задачи

Для определения элементов матрицы ОПВ марковской модели смены состояний ПК $\bar{t}_{ок}(k)$ воспользуемся статистическими данными относительно потоков требований на предоставление информационного ресурса в системах поиска и хранения данных со свободным доступом.

В работе [7] получен ряд аналитических выражений, определяющих взаимосвязь между временем ожидания $\bar{t}_{ок}(k)$ и числом запросов (требований) на информационный ресурс, находящихся на обслуживании (в очереди) $\eta_{ю}(k)$. С использованием этих

данных изменения состояния процесса ожидания в ЭБ достаточно полно можно проследить, наблюдая изменения числа запросов (требований), находящихся на обслуживании $\eta_{\text{но}}(k)$ в системе распределения ресурса электронной библиотеки. Необходимым условием построения математической модели данного процесса в рамках УЦМ является наличие матрицы ОПВ, характеризующей вероятность перехода процесса изменения числа запросов на ресурс ЭБ, находящихся на обслуживании (в очереди) ($\eta_{\text{но}}(k)$), из одного состояния в другое. При этом с точки зрения системы массового обслуживания (СМО) относительно процесса функционирования ЭБ необходимо сделать следующие допущения [8].

1. Входящие потоки заявок (требований) на информационный ресурс ЭБ ($\eta_{\text{т}}(k)$) являются простейшими с интенсивностями λ_{ij} , причем $\Lambda = \sum_{i=1}^I \sum_{j=1}^J \lambda_{ij}$ —

суммарная интенсивность трафика запросов (требований) на информационный ресурс ЭБ ($i = 1, \dots, I$ — источники заявок на ресурс ЭБ (пользователи); $j = 1, \dots, J$ — виды запрашиваемого ресурса ЭБ).

2. Распределение длительности обслуживания запросов (требований) на информационный ресурс ЭБ является экспоненциальным с параметром μ , где μ — интенсивность обслуживания, измеряемая в запросах/ч.

3. Число n единиц ресурса ЭБ (порты) в режиме приема запросов (требований) на информационный ресурс и число мест в очереди N ограничено.

4. Время ожидания в очереди $\xi(v)$ является ограниченным и распределенным по экспоненциальному закону с параметром v .

5. Дисциплина обслуживания трафика запросов (требований) на информационный ресурс ЭБ — в порядке поступления.

В соответствии с принятой классификацией, данную модель СМО для ЭБ можно отнести к типу $M/M/n/N$. Модель СМО такого типа подразумевает, что тип распределения входящего потока требований — марковский поток (распределение Пуассона), тип распределения времени обслуживания — марковский поток (показательное экспоненциальное распределение), а также имеется n обслуживающих приборов и N мест в очереди на обслуживание.

Описание математической реализации метода решения задачи

Для марковских моделей СМО такого класса воспользуемся алгоритмом определения значений элементов матрицы интенсивностей перехода (интенсивность перехода обозначается q) процесса $\eta_{\text{но}}(k)$ из одного состояния в другое, например, из m -го в l -е состояние:

$$|q_{ml}| = \begin{vmatrix} q_{11}(k+1, k) & q_{12}(k+1, k) & q_{13}(k+1, k) \\ q_{21}(k+1, k) & q_{22}(k+1, k) & q_{23}(k+1, k) \\ q_{31}(k+1, k) & q_{32}(k+1, k) & q_{33}(k+1, k) \end{vmatrix}.$$

Учитывая взаимосвязь значений элементов матрицы ОПВ и интенсивностей изменения состояния трафика запросов (требований) на информационный ресурс ЭБ, находящегося на обслуживании, получим:

$$\varphi(k+1, k): \begin{cases} p_{ml}(k+1, k) = Tq_{ml}(k+1, k) + OT \\ p_{mm}(k+1, k) = 1 - Tq_{mm}(k+1, k) + OT \end{cases} \quad (1)$$

где $\varphi(k+1, k)$ — матрица ОПВ, переводящая ПК из одного состояния в другое в процессе функционирования ЭБ; $T = (t_k - t_{k-1})$ — интервал времени между сменой состояний, который должен удовлетворять условию $0 \leq T \leq \tau_{\text{кор}}$ — временной интервал, за который проявляется корреляционная зависимость — статистическая взаимосвязь двух случайных величин (в нашем случае — двух состояний), а O — символ операции обслуживания заявок (требований) на информационный ресурс ЭБ. Символ O является безразмерным, условным обозначением факта отдельной операции (ожидания, обслуживания) и применяется для удобства описания процесса удовлетворения заявок пользователей.

Известно [8–10], что понимая под состоянием ЭБ в k -й момент времени число заявок (требований) на ее информационный ресурс (заявок, находящихся на обслуживании в системе распределения ресурса и очереди), можно представить множество возможных состояний процесса $\eta_{\text{но}}(k)$ в следующем виде:

$$M = \{m\}, m = \overline{0, n+N}.$$

Тогда интенсивности перехода процесса $\eta_{\text{но}}(k)$ из одного состояния в другое могут быть определены следующим образом [8].

1. При $0 \leq m \leq n$ могут выполняться одна фиктивная операция ожидания заявок (требований) на информационный ресурс ЭБ O_0 и m операций обслуживания заявок (требований) на информационный ресурс ЭБ O_1, \dots, O_m , а соответствующая им интенсивность перехода равна

$$q_{ml}(k+1, k) = \begin{cases} \lambda & \text{при } l = m+1 \\ m\mu & \text{при } l = m-1 \\ 0 & \text{при } |l-m| \geq 2 \end{cases} \quad (2)$$

2. При $m = 0$ имеем

$$q_{0l}(k+1, k) = \begin{cases} \lambda & \text{при } l = 1 \\ 0 & \text{при } l \geq 2 \end{cases} \quad (3)$$

3. При $n < m < n+N$ выполняется одна фиктивная операция ожидания заявок (требований) на информационный ресурс ЭБ O_0 , n операций обслуживания заявок (требований) на информационный ресурс ЭБ O_1, \dots, O_n , и $(m-n)$ фиктивных операций "ожидания потери заявок (требований) на информационный ресурс ЭБ". Тогда

$$q_{ml}(k+1, k) = \begin{cases} \lambda & \text{при } l = k+1 \\ n\mu + (m-n)v & \text{при } l = k-1 \\ 0 & \text{при } |m-l| \geq 2 \end{cases} \quad (4)$$

4. Для случая $m = n+N$ интенсивность перехода определяется следующим образом:

$$q_{(n+N)l}(k+1, k) = \begin{cases} n\mu + Nv & \text{при } l = (n+N) \\ 0 & \text{при } l < (n+N) - 1 \end{cases} \quad (5)$$

Начальное распределение вероятностей состояний показателей качества электронной библиотеки, при условии, что в момент времени $k = 0$ очередь запросов в ЭБ свободна от заявок (требований) на информационный ресурс библиотеки, определяют как

$$P(0; \eta_{\text{но}}(0)) = \begin{cases} 1 & \text{при } m = 0 \\ 0 & \text{при } m > 0 \end{cases} \quad (6)$$

Нетрудно показать [8, 9], что механизм вероятностно-временного перехода полностью задает модель процесса функционирования ЭБ — модель процесса обслуживания заявок (требований) на информационный ресурс ЭБ — и может быть представлен в терминах цепей Маркова в форме разностных стохастических уравнений. Уравнения состояния в этом случае будут иметь следующий вид:

$$\begin{aligned} \eta_{\text{но}}(k) &= C^T(k; \eta_{\text{но}} = m) \bar{\Theta}(k); \quad \eta_{\text{но}}(0) = 0 \\ \bar{\Theta}(k) &= [A + Q^T(k, k-1; u)] \bar{\Theta}(k-1) + \\ &+ \Gamma(k-1) \bar{V}(k-1), \end{aligned} \quad (7)$$

где $\eta_{\text{но}}(k)$ — число заявок на обслуживание в очереди; C^T — диагональная матрица возможных значений показателей качества электронной библиотеки (т. е. числа заявок на обслуживание); $\bar{\Theta}(k)$ — вектор значений индикаторов состояния процесса на k -м шаге (элементы вектора принимают значения 0 или 1); A — диагональная единичная матрица; $Q(k, k-1; u) = q_{ml}(k, k-1; u)$ — матрица интенсивностей перехода; $\bar{\Theta}(k-1)$ — вектор значений индикаторов состояния процесса на предыдущем, $(k-1)$ -м шаге; $\Gamma(k-1)$ — матрица возбуждения процесса смены состояний $\bar{\Theta}(k-1)$, которая во взаимодействии с вектором дискретных по состоянию последовательностей возбуждения $\bar{V}(k-1)$, позволяют компенсировать дробные части в уравнении (7) и получить целочисленные значения $\bar{\Theta}(k-1)$.

Тогда, по аналогии с результатами, представленными в работах [9, 10], финальные значения вероятностей значений числа заявок (требований) на информационный ресурс ЭБ, находящихся на обслуживании для установившегося режима работы ЭБ, могут быть определены из следующего выражения:

$$\pi_m = \begin{cases} \frac{\alpha^m / m!}{\sum_{s=1}^n \frac{\alpha^s}{s!} + \frac{\alpha^n}{n!} \sum_{s=1}^{N-1} \frac{\alpha^s}{\prod_{r=1}^s (n+r\beta)}} & \text{при } m = \overline{0, n} \\ \frac{\alpha^m / (n! \prod_{r=1}^s (n+r\beta))}{\sum_{s=1}^n \frac{\alpha^s}{s!} + \frac{\alpha^n}{n!} \sum_{s=1}^N \frac{\alpha^s}{\prod_{r=1}^s (n+r\beta)}} & \text{при } m = \overline{n+1, n+N} \end{cases}, \quad (8)$$

где $\alpha = \lambda/\mu$, $\beta = \nu/\mu$.

Пользуясь этим выражением, можно определить среднее число заявок (требований) на информационный ресурс ЭБ в очереди на обслуживание:

$$M[\eta_o(\infty)] = N_Q = \sum_{r=1}^N r \pi_{n+r}, \quad (9)$$

а также дисперсию

$$D[\eta_o(\infty)] = \sum_{r=1}^N (r - m_{\eta_o})^2 \pi_{n+r}. \quad (10)$$

Выражение (10) характеризует дисперсию числа требований на ресурс ЭБ в очереди (дисперсию числа заявок (требований) на информационный ресурс ЭБ, находящихся в очереди на обслуживание).

Интерпретация и иллюстрация полученных результатов

Рассмотрена марковская модель смены состояний ПК $\bar{t}_{\text{ож}}(k)$ — среднего времени ожидания доступа легальных (авторизованных) пользователей к информационному ресурсу ЭБ на k -м шаге ее функционирования (времени ожидания предоставления ЭБ конкретной услуги). Вместе с тем существует ряд особенностей и иных ПК, характеризующих существенные (наиболее важные с точки зрения качества обслуживания пользователей) свойства ЭБ, например, безопасность, устойчивость обслуживания.

К особенностям ЭБ, характеризующим их сложность и отличающим их от традиционных систем массового обслуживания, можно отнести:

- глобальный доступ к ним в среде web;
 - поддержку метаданных для системы и пользователей;
 - включение неоднородных информационных ресурсов, каталогов, электронных книг, справочников, аудиовизуальных и электронных документов;
 - распределение информационных ресурсов по назначению — государственные (национальные), корпоративные (предприятия, организации, ведомства), индивидуальные;
 - наличие материализованных (в виде систем хранения данных, баз данных, дата-центров и дисков) и виртуальных коллекций информационных ресурсов (электронных версий печатных изданий, аудио и видеоданных и т. д.);
 - интеграцию информационных ресурсов на различных уровнях;
 - использование пользовательских интерфейсов с возможностью их адаптации к контексту (семантике, смысловой сущности) поисковых запросов пользователей на получение конкретного информационного ресурса [1, 2].
- Ключевые показатели качества функционирования современных ЭБ должны, по мнению авторов, характеризовать ряд этих существенных свойств таких систем и процессов, ими реализуемых. Эти ПК могут численно характеризовать:
- своевременность, как в нашем случае — ПК: среднее время ожидания доступа легальных (авторизованных) пользователей к информационному ресурсу ЭБ, $\bar{t}_{\text{ож}}(k)$;
 - достоверность — ПК: коэффициент правильного (безошибочного, без потерь) хранения,

обработки и выдачи информации, предоставляемой пользователю;

- безопасность — ПК: время вскрытия процессов, протекающих в контуре управления ЭБ, и коэффициент конфиденциальности информации, циркулирующей по каналам управления и сигнализации ЭБ, а также показатели надежности, отказоустойчивости, доступности, объема хранения данных и масштабируемости.

В рамках данной статьи рассматриваем модель смены состояния одного из этих ПК — показателя, характеризующего своевременность (оперативность) ЭБ: среднего времени ожидания доступа легальных (авторизированных) пользователей к информационному ресурсу ЭБ — $\bar{t}_{ож}(k)$.

Опираясь на выражения (1)–(10), рассмотрим, как интерпретируются элементы матрицы ОПВ состояний процесса $\eta_{но}(k)$ в элементы матрицы ОПВ процесса смены состояний ПК $\bar{t}_{ож}(k)$ — среднего времени ожидания доступа легальных (авторизированных) пользователей к информационному ресурсу ЭБ на k -м шаге ее функционирования (времени ожидания предоставления ЭБ конкретной услуги). Очевидно, что:

$$\begin{aligned} \bar{t}_{ож}(k) &= 0 && \text{при } m \leq n; \\ 0 < \bar{t}_{ож}(k) &\leq \bar{t}_{ож}^{пор} && \text{при } m \leq n + N; \\ \bar{t}_{ож}(k) &> \bar{t}_{ож}^{пор} && \text{при } m > n + N, \end{aligned} \quad (11)$$

где $\bar{t}_{ож}^{пор}(k)$ — пороговые (граничные, требуемые) значения среднего времени ожидания доступа легальных (авторизированных) пользователей к информационному ресурсу ЭБ, а m , n и N — обозначения, используемые для процесса $\eta_{но}(k)$. Если ограничить процесс смены состояний $\bar{t}_{ож}(k)$ тремя состояниями (например, "исправная работа", "предупреждение" и "авария"), т. е. условно считать, что элементы вспомогательного вектора индикаторов в уравнении состояния

$$\Theta_1(\bar{t}_{ож}) = \begin{cases} 1 & \text{— если } 0 \leq \bar{t}_{ож} \leq \bar{t}_{ож}^{пор_1}, \\ 0 & \text{— в противном случае;} \end{cases} \quad (12)$$

$$\Theta_2(\bar{t}_{ож}) = \begin{cases} 1 & \text{— если } \bar{t}_{ож}^{пор_1} \leq \bar{t}_{ож} \leq \bar{t}_{ож}^{пор_2}, \\ 0 & \text{— в противном случае;} \end{cases} \quad (13)$$

$$\Theta_3(\bar{t}_{ож}) = \begin{cases} 1 & \text{— если } \bar{t}_{ож} > \bar{t}_{ож}^{пор_2}, \\ 0 & \text{— в противном случае;} \end{cases} \quad (14)$$

где $\bar{t}_{ож}^{пор_1}$ и $\bar{t}_{ож}^{пор_2}$ — граничные значения между тремя состояниями, а число уже задействованных мест в очереди обозначим d , то можно записать

$$\bar{\Theta}(\bar{t}_{ож}) = I$$

$$= \begin{cases} \Theta_1 = 1; \Theta_2 = \Theta_3 = 0 & \text{при } 0 < m \leq n + N - d, \\ \Theta_2 = 1; \Theta_1 = \Theta_3 = 0 & \text{при } n + N - d < m \leq n + N, \\ \Theta_3 = 1; \Theta_1 = \Theta_2 = 0 & \text{при } m > n + N, m = \overline{0, M}. \end{cases} \quad (15)$$

Уравнение состояния для среднего времени ожидания доступа легальных (авторизированных)

пользователей к информационному ресурсу ЭБ на k -м шаге ее функционирования (времени ожидания предоставления ЭБ конкретной услуги) определяется выражением:

$$\bar{t}_{ож}(k+1) = C^T \bar{\Theta}(k+1),$$

где C^T — диагональная матрица возможных значений $\bar{t}_{ож}$, элементы которой для трех состояний $\bar{t}_{ож}$ принимают следующие значения:

$$|C^T| = \begin{vmatrix} 0 \leq \bar{t}_{ож} \leq \bar{t}_{ож}^{пор_1} & 0 & 0 \\ 0 & \bar{t}_{ож}^{пор_1} \leq \bar{t}_{ож} \leq \bar{t}_{ож}^{пор_2} & 0 \\ 0 & 0 & \bar{t}_{ож} > \bar{t}_{ож}^{пор_2} \end{vmatrix}. \quad (16)$$

С учетом того, что на основе теоремы Литтла [11]

$$\bar{t}_{ож} = \frac{N_Q}{\lambda},$$

где N_Q — общее число требований, находящихся в системе, для $\eta_{но}(k)$ получим выражение для среднего значения времени ожидания

$$\bar{t}_{ож} = \frac{\sum_{r=1}^N r \pi_{n+r}}{\Lambda},$$

где $\Lambda = \sum_{i=1}^I \sum_{j=1}^J \lambda_{ij}$ — суммарная интенсивность трафика,

интенсивность потока заявок (требований) на информационный ресурс ЭБ.

Таким образом, имея матрицу вероятностей перехода процесса $\eta_{но}(k)$ для m множества состояний при различных n и N , учитывая зависимость пороговых значений состояний процессов $\eta_{но}(k)$ и $\bar{t}_{ож}(k)$ и вводя пороговые значения для состояний $\bar{t}_{ож}(k)$, можем получить элементы матрицы ОПВ для ПК ЭБ $\bar{t}_{ож}(k)$ для любого k -го момента времени ($0 \leq k < \infty$).

Обозначим состояния ПК $\bar{t}_{ож}(k)$ — среднего времени ожидания доступа легальных (авторизированных) пользователей к информационному ресурсу ЭБ на k -м шаге ее функционирования (времени ожидания предоставления ЭБ конкретной услуги) и соответствующие им вероятности.

1. Состояние "исправная работа", вероятность $P_1(0 \leq \bar{t}_{ож} \leq \bar{t}_{ож}^{пор_1})$, что соответствует наличию в очереди на ресурс ЭБ от 0 до $n + N - d$ требований (наличию от 0 до $n + N - d$ заявок (требований) на информационный ресурс ЭБ, находящихся в очереди на обслуживание).

2. Состояние "предупреждение", вероятность $P_2(\bar{t}_{ож}^{пор_1} \leq \bar{t}_{ож} \leq \bar{t}_{ож}^{пор_2})$, что соответствует наличию в очереди на использование ресурса ЭБ от $n + N - d$ до $n + N$ требований (наличию от $n + N - d$ до $n + N$ заявок (требований) на информационный ресурс ЭБ, находящихся в очереди на обслуживание).

3. Состояние "авария", вероятность $P_3(\bar{t}_{ож} > \bar{t}_{ож}^{пор_2})$, что соответствует наличию в очереди на использование ресурса ЭБ больше $n + N$ требований (наличию более

чем $n + N$ заявок (требований) на информационный ресурс ЭБ, находящихся в очереди на обслуживание).

При этом элементы матрицы ОПВ принимают вид

$$\begin{aligned}
 p_{11} &= \sum_{i=1}^{n+N-d} \sum_{j=1}^{n+N-d} p_{ij}; & p_{21} &= \sum_{i=n+N-d}^{n+N} \sum_{j=1}^{n+N-d} p_{ij}; \\
 p_{12} &= \sum_{i=1}^{n+N-d} \sum_{j=n+N-d}^{n+N} p_{ij}; & p_{32} &= \sum_{i=n+N}^{n+N+m} \sum_{j=n+N-d}^{n+N} p_{ij}; \\
 p_{22} &= \sum_{i=n+N-d}^{n+N} \sum_{j=n+N-d}^{n+N} p_{ij}; & p_{33} &= \sum_{i=n+N}^{n+N+m} \sum_{j=n+N}^{n+N+m} p_{ij}; \\
 p_{23} &= 1 - p_{21} - p_{22}; & p_{13} &= p_{31} = 0.
 \end{aligned} \quad (17)$$

Здесь n — число единиц ресурса ЭБ (число портов) в режиме приема запросов (требований) на информационный ресурс, N — число мест в очереди, d — число уже задействованных мест в очереди.

Таким образом, предложены аналитические выражения для нахождения элементов матрицы переходных вероятностей для моделирования трех состояний рассматриваемого показателя качества.

Использованные в статье известные и описанные в работах [7, 8, 11] данные и выражения из теории массового обслуживания (и не только) применены как исходные данные (в качестве напоминания и попытки установить логический "мостик" между традиционными постулатами систем массового обслуживания и современных ЭБ). Научная новизна, по мнению авторов, заключается в попытке применить эти традиционные механизмы к анализу и управлению современными сложными информационно-поисковыми системами, такими как ЭБ. Главным образом она состоит в разработке подхода (показан на примере ПК ЭБ $\bar{t}_{ок}(k)$) к определению элементов матрицы вероятностей перехода процесса смены состояний ПК электронной библиотеки с учетом пороговых значений этих состояний.

Программное средство для реализации алгоритмов определения элементов матрицы переходных вероятностей (в интересах моделирования смены состояний трех показателей качества электронной библиотеки) исполняется на языке C++ в среде разработки C++ Builder и Visual Studio 8, при этом программа будет работать под MS Windows. При разработке приложения для определения элементов матрицы переходных вероятностей использованы стандартные модули и процедуры языка программирования. Они позволяют построить логически правильный алгоритм для решения поставленной задачи. Для реализации данного программного продукта использование дополнительных нестандартных модулей и процедур не требуется.

Интегрированным средам разработки C++ Builder и Visual Studio 8 отдано предпочтение, потому что они обладают большим числом функций, включая собственный компилятор, поддержку баз данных, поддержку технологий Windows Forms, объекты модулей данных, открытые инструментальные средства интерфейса программирования приложений и т. д. Кроме того, такой выбор обусловлен возможностью более легкого создания в этих средах приложения с графическим интерфейсом.

Заключение

С учетом взаимосвязи количества требований на ресурс ЭБ, находящихся на обслуживании, и времени ожидания доступа легальных (авторизированных) пользователей к информационному ресурсу ЭБ на k -м шаге ее функционирования (времени ожидания предоставления ЭБ конкретной услуги), предложен метод расчета ключевых элементов модели. Сформулированы понятия значений состояний $\eta_{но}(k)$, соответствующие пороговым значениям состояний ПК ЭБ $\bar{t}_{ок}(k)$, предложены аналитические выражения для нахождения элементов матрицы ОПВ, которые в нашем случае характеризуют вероятности перехода из состояния в состояние такого ПК ЭБ, как среднее время ожидания доступа легальных (авторизированных) пользователей к информационному ресурсу ЭБ на k -м шаге ее функционирования (времени ожидания предоставления ЭБ конкретной услуги).

Учитывая линейную взаимосвязь ПК $\bar{t}_{ок}(k)$ и его отклонений $\Delta \bar{t}_{ок}(k)$ от требуемых значений, можно предположить дальнейшее развитие и применение предложенного подхода. Необходимо, например, доказать, что полученные элементы матрицы ОПВ справедливы и для такого ПК ЭБ $\Delta \bar{t}_{ок}(k)$, как отклонение от требуемых значений среднего времени ожидания доступа авторизированных пользователей к информационному ресурсу электронной библиотеки (времени ожидания предоставления ЭБ конкретной услуги). В статье описаны подходы к созданию программных продуктов в среде разработки C++ Builder и Visual Studio, которые могут быть исследованы пользователями электронных библиотек.

Список литературы

1. Зуйкина К. Л., Соколова Д. В., Скалабан А. В. Электронные библиотеки в России. Текущий статус и перспективы развития. — М.: Ваш формат, 2017. — 120 с.
2. Национальный стандарт Российской Федерации ГОСТ Р 7.0.96-2016. Электронные библиотеки. Основные виды. Структура. Технология формирования. — М.: Стандартинформ, 2016. — 13 с.
3. Ramaiah S. K. Electronic Books: A Bibliography // DESIDOC Journal of Library & Information Technology. March. — 2012. — Vol. 32, No. 2. — P. 116—166.
4. Aaltonen M., Mannonen P., Nieminen S., Nieminen M. Usability and compatibility of e-book readers in an academic environment: A collaborative study // IFLA Journal. — 2011. — Vol. 37, Iss. 1. — P. 16—27.
5. Abdulmumin I., Batlang Comma S., Athulang M., Lekoko K. Digital Libraries: Analysis of Delos Reference Model and 5S Theory // Journal of Information Science Theory and Practice. — 2013, 1 (4). — P. 38—47.
6. Замятин А. М. Модель электронной библиотечной системы вуза // Молодой ученый. — 2011. Т. 1, № 4. — С. 89—91.
7. Лагутин В. С., Степанов С. Н. Телетрафик мультисервисных сетей связи. — М.: Радио и связь, 2000. — 320 с.
8. Ивченко Г. И., Каштанов В. А., Коваленко И. Н. Теория массового обслуживания. — М.: Высшая школа, 1982. — 256 с.
9. Парашук И. Б., Деньжонков К. А., Зубакин В. В. и др. Основы передачи данных: Учебник / Под ред. проф. И. Б. Парашука. — СПб.: ВАС, 2015. — 216 с.
10. Терентьев В. М., Парашук И. Б. Теоретические основы управления сетями многоканальной радиосвязи. — СПб.: ВАС, 1995. — 195 с.
11. Bertsekas D. P., Gallager R. G. Data Network. — USA: Prentice Hall, 1992. — 556 p.

Software Tool for Modeling the State Transition of a Quality Indicator of an Electronic library: a Method for Determining the Key Elements of a Model

E. S. Kryukova, e.kkrukovaa69@yandex.ru, **I. B. Parashchuk**, e-mail: shchuk@rambler.ru,
The Military Academy of Telecommunications named after Marshal of the Soviet Union S. M. Budenny,
St. Petersburg, 194064, Russian Federation

Corresponding author:

Parashchuk Igor B., Professor, The Military Academy of Telecommunications named after Marshal of the Soviet Union S. M. Budenny, St. Petersburg, 194064, Russian Federation
E-mail: shchuk@rambler.ru

Received on March 06, 2020

Accepted on March 25, 2020

An approach to the mathematical description and software implementation of the algorithm for determining the elements of the transition probability matrix as a key object of the Markov probability-time model of the process of changing the states of the average waiting time for users to access an electronic information resource is proposed. From the point of view of mathematical implementation, the approach is based on the provisions of the theory of queuing, relies on the initial data on the intensity of applications for the information resource of the electronic library and the parameters of its queue for service. Software implementation takes into account the features and advantages of modern programming languages.

Keywords: software tool, model, electronic library, quality indicator, intensity, matrix, transition probabilities, state, information resource, request, queue, timeout, access

For citation:

Kryukova E. S., Parashchuk I. B. Software Tool for Modeling the State Transition of a Quality Indicator of an Electronic library: a Method for Determining the Key Elements of a Model, *Programmnyaya Ingeneria*, 2020, vol. 11, no. 3, pp. 177–182

DOI: 10.17587/prin.11.177-182

References

1. **Zuykina K. L., Sokolova D. V., Skalaban A. V.** *Digital Libraries in Russia. Current status and development prospects*, Moscow, Vash format, 2017, 120 p. (in Russian).
2. **National** standard of the Russian Federation GOST R 7.0.96–2016. *Electronic libraries. The main types. Structure. Formation technology*, Moscow, Standardartinform, 2016, 13 p. (in Russian).
3. **Ramaiah C. K.** Electronic Books: A Bibliography. *DESIDOC Journal of Library & Information Technology*, March 2012, vol. 32, no. 2, pp. 116–166.
4. **Aaltonen M., Mannonen P., Nieminen S., Nieminen M.** Usability and compatibility of e-book readers in an academic environment: A collaborative study, *IFLA Journal*, 2011, vol. 37, iss. 1, pp. 16–27.
5. **Abdulmumin I., Batlang Comma S., Athulang M., Lekoko K.** Digital Libraries: Analysis of Delos Reference Model and 5S Theory, *Journal of Information Science Theory and Practice*, 2013, vol. 4, no. 1, pp. 38–47.
6. **Zamyatin A. M.** Model of the electronic library system of the university, *Molodoj uchenij*, 2011, vol. 1, no. 4, pp. 89–91 (in Russian).
7. **Lagutin V. S., Stepanov S. N.** *Teletraffic of multiservice communication networks*, Moscow, Radio and Communications, 2000, 320 p. (in Russian).
8. **Ivchenko G. I., Kashtanov V. A., Kovalenko I. N.** *Queuing theory*, Moscow, Higher School, 1982, 256 p. (in Russian).
9. **Parashchuk I. B., Denzhonkov K. A., Zubakin V. V.** et al. *Fundamentals of data transfer* / Ed. prof. I. B. Parashchuk. Saint-Petersburg: Military Academy of Telecommunications, 2015, 216 p. (in Russian).
10. **Terentyev V. M., Parashchuk I. B.** *Theoretical foundations of multichannel radio network management*, St. Petersburg, Military Academy of Telecommunications, 1995, 195 p. (in Russian).
11. **Bertsekas D. P., Gallager R. G.** *Data Network*, USA, Prentice Hall, 1992, 556 p.

В. С. Компаниец, канд. техн. наук, доц., kompaniets@sfnedu.ru,
А. Ю. Казанская, канд. экон. наук, доц., akazanskaya@sfnedu.ru,
А. Е. Лызь, канд. техн. наук, доц., aelyz@sfnedu.ru,
А. О. Григорьев, зав. лабораторией, agrigorev@sfnedu.ru, Инженерно-технологическая академия ФГАОУ ВО Южный федеральный университет, г. Таганрог

Систематизация процедур проектирования пользовательского опыта в подготовке магистров прикладной информатики

Описан опыт проектирования и реализации в вузе магистерской программы "Эргодизайн пользовательского интерфейса", в которой выполнено формирование компетентностной модели выпускника по актуальным требованиям нового поколения федерального стандарта (ФГОС ВО 3++) и функциональной карте профильного профстандарта "Специалист по дизайну графических и пользовательских интерфейсов". Кратко представлено содержательное наполнение учебного плана, в котором важная роль отведена профессиональному модулю "UX/UI-дизайн". Подробно описан и проиллюстрирован пошаговым примером пятилетний опыт обучения студентов по классической уровневой системе элементов Дж. Гарретта. Показано, как каркас уровней системы может быть наполнен набором конкретных методов и инструментальных средств проектирования пользовательского опыта, доказавшим на практике свою эффективность и дидактическую ценность.

Ключевые слова: UX/UI-дизайн, классический процесс UX, элементы Дж. Гарретта, эргодизайн пользовательского интерфейса, айтрекинг, Eye Tribe, нейрогарнитура, NeuroSky, электроэнцефалография

Введение

За последние три десятилетия технологические прорывы в области ИТ сформировали экосистемы тотального пользования компьютерами. Расширение использования информационных технологий не только для решения профессиональных задач, но и для удовлетворения человеком разнообразных информационных, досуговых и коммуникативных потребностей предъявляет к пользовательскому интерфейсу ряд новых требований. Интерфейс должен не только быть ориентирован на эффективность, продуктивность и удовлетворенность пользователей решением задачи, но и иметь эстетическую привлекательность, мотивационную притягательность для пользователя, а значит опираться на его целостный пользовательский опыт. В общем случае пользовательский опыт (опыт взаимодействия, *User experience*, UX) — это восприятие и ответные действия пользователя, возникающие в результате использования и/или предстоящего использования продукции, системы или услуги (ISO 9241—210¹). В сфере ИТ в процессе использования как информационной системы в целом, так и отдельной ее части (веб-

сайта, приложения и пр.) термин UX применяется для описания субъективного отношения, возникающего у пользователя. Однако при внешней простоте термина анализ и оценка пользовательского опыта — сложная и неоднозначная задача, включающая слабоформализуемые методы (опросы, наблюдение за работой пользователей, тестирование производительности и анализ журналов их проведению и наполнению конкретным содержанием). Так возникла задача систематизации процедур проектирования пользовательского опыта для включения в образовательный процесс будущих UX/UI-дизайнеров (UI, *User interface* — пользовательский интерфейс).

Цель настоящей статьи — представить опыт решения данной задачи при реализации в вузе авторской магистерской программы "Эргодизайн пользовательского интерфейса" по направлению подготовки "Прикладная информатика".

1. Магистерская программа "Эргодизайн пользовательского интерфейса"

Эргодизайн пользовательского интерфейса является наименованием и центральным понятием реализуемой в Южном федеральном университете с 2015 г. магистерской программы по прикладной информатике. Программа — победитель грантового конкурса Стипендиальной программы Фонда В. Потанина (2017 г.).

¹ ГОСТ Р ИСО 9241-210—2012 "Эргономика взаимодействия человек—система. Часть 210. Человеко-ориентированное проектирование интерактивных систем" (ISO 9241-210:2010 "Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems").

Цель программы — научить магистрантов проектировать целеориентированный, эргономичный дизайн интерфейса и проводить его экспертизу [1–3].

При разработке образовательной программы были рассмотрены следующие близкие по концепции магистерские программы российских вузов: "Человеческое взаимодействие в информационных системах" (УрФУ) и "Эргономика" (СПбГЭТУ "ЛЭТИ"), которые характеризуются доминирующим уклоном либо только в технологические дисциплины инженерии ПО (УрФУ), либо в спецдисциплины по эргономике (ЛЭТИ). В большей степени концепции эргоди-зайна UI соответствует новая российская программа "Мультимедиа-технологии, дизайн и юзабилити" (специализация "Дизайн человеко-компьютерных систем", Университет ИТМО) и международная программа *Human Computer Interaction and Design* (HCID), реализуемая семью европейскими техническими вузами в рамках проекта EIT Digital Master School. Но на рынке образовательных услуг южного региона наша программа уникальна и востребована как абитуриентами (в 2019 г. успешно состоялся третий набор на программу), так и работодателями (все студенты трудоустраиваются еще до выпуска, и растет число профильных предприятий-партнеров). Внешние партнеры, предоставляя места практики и трудоустройства, играют также важную экспертную роль в образовательном процессе, выборе актуальных тем, рецензировании и защите выпускных квалификационных работ.

Для проектируемых результатов обучения в магистратуре — профессиональных компетенций выпускника — необходимо было разработать перечень индикаторов формирования компетенций и систему требований в традиционном для системы образования формате: знания, умения, навыки. Дополнительная сложность состояла в том, что профессиональные компетенции федеральными государственными стандартами (ФГОС ВО поколения 3++) явно не определены. Перечень этих компетенций разработчики образовательной программы должны формировать самостоятельно на основе так называемых "обобщенных трудовых функций" из действующих в конкретной предметной области профессиональных стандартов. В области дизайна UI существует лишь один профессиональный стандарт 06.025 "Специалист по дизайну графических и пользовательских интерфейсов", содержащий для образовательных программ магистратуры (седьмой квалификационный уровень) две обобщенные трудовые функции: проектирование сложных пользовательских интерфейсов и экспертный анализ эргономических характеристик программных продуктов и/или аппаратных средств. Дальнейшая декомпозиция обобщенных функций до уровня решаемых выпускником профессиональных задач позволила сформировать компетентностную модель выпускника магистратуры, а также представление об основных блоках, модулях и дисциплинах учебного плана [3].

Дальнейшим предметом методической разработки магистерской программы стал учебный план, в профессиональную часть которого включены обзорные курсы по основам эргономики и инженерной психологии, методам инженерно-психологической экспертизы программных продуктов, когнитивной психологии и нейронауке; а также большие курсы по программированию (6 зачетных единиц трудоемко-

сти, ЗЕТ), основам дизайна и технической эстетики (8 ЗЕТ). Важная роль в учебном плане отведена модулю "UX/UI-дизайн", содержащему два последовательных курса: "UX-практикум" и "Проектирование сложных пользовательских интерфейсов".

2. Содержание модуля "UX/UI-дизайн"

Более подробное описание учебного модуля будет описано далее.

2.1. Формирование общих требований к учебному модулю

Согласно компетентностной модели программы в результате освоения модуля студенты должны знать:

- тенденции в проектировании интерфейсов;
- паттерны поведения людей при использовании программных продуктов и аппаратных средств;
- техники сбора этнографической и социологической информации;
- нотации записи структурных схем, описания логики работы приложения;
- общие практики и методы проектирования интерфейсов.

Они должны уметь:

- проводить фокусированные и этнографические интервью с пользователями;
- эскизировать интерфейсы;
- читать, создавать, модифицировать и оформлять структурные схемы интерфейса;
- выявлять особенности интерфейса, которые влияют на выполнение задач пользователем;
- разрабатывать интерфейсные решения и прототипировать интерфейсы.

Поэтому, работая над содержанием модуля, потребовалось изучить, систематизировать и методически выстроить известные подходы, методы и инструменты методологии UX/UI-дизайна, а также проанализировать собственный опыт инженерно-психологических исследований и разработки прикладных информационных систем.

Были изучены публикации об актуальном Dual Task-дизайне, соединившем дизайн-мышление и модель дизайна спринта от Google Ventures с методологией Lean UX Джеффа Готельфа [4–9]. Так же как и ряд авторитетных авторов, по результатам такого изучения авторами был сделан вывод, что современная методология UX содержит в своей основе классический каскадный UX-процесс, максимально ускоренный (цель "наскоро и грубо" получить MVP¹) вслед за гибкой разработкой [10–12]. Поэтому в учебные курсы решено было включить именно классический процесс UX, постепенно овладевая которым студенты могли бы в дальнейшем развивать практические навыки, совершенствоваться и, повышая темп работы, более успешно включаться в современные процессы дизайна и разработки. В качестве опорной модели была выбрана классическая система элементов пользовательского опыта, разработанная Джессом Гарреттом [13]: от абстрактного

¹ MVP (от англ. *Minimum Viable Product* — минимально жизнеспособный продукт) — простейший работающий прототип продукта, которым тестируют спрос до полномасштабной разработки.

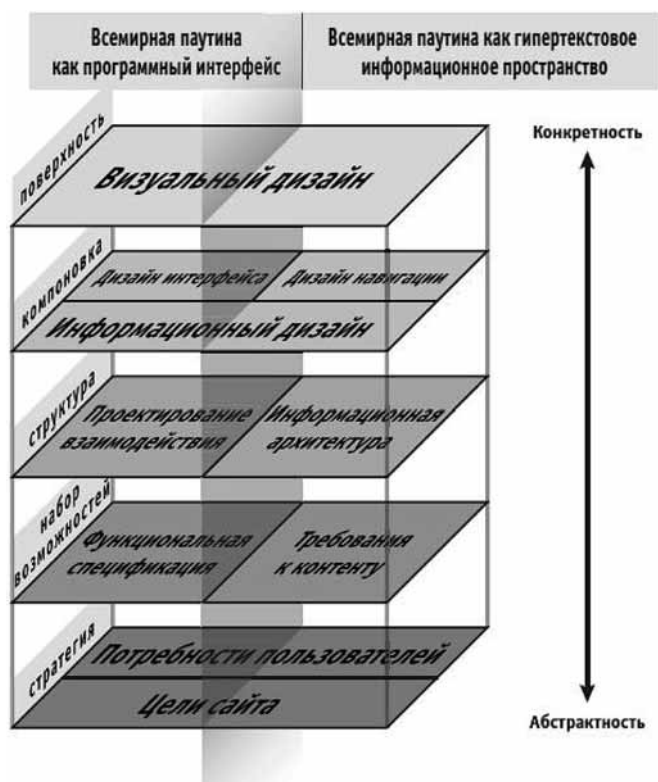


Рис. 1. Элементы Дж. Гарретта (<http://jig.net/elements/>)

уровня "стратегии" до детализированного уровня "поверхности" программного продукта (рис. 1).

По сути, система элементов Гарретта дает хорошо применимую в учебном процессе наглядную структуру классического исследования пользовательского опыта (от общего к частному). Систему необходимо было встроить в учебный модуль и наполнить актуальным набором методов и инструментальных средств. Было выполнено следующее разделение: уровни стратегии и набора возможностей системы Гарретта вынесены в курс "UX-практикум", а последующие уровни — в курс "Проектирование сложных пользовательских интерфейсов". Благодаря указанному разделению удалось добиться, чтобы освоению детальных уровней системы Гарретта предшествовал технологический курс по программированию. На лекционных занятиях курсов приводится требуемая теоретическая информация по каждому уровню системы Гарретта, также дается краткий обзор применяемых методов и инструментов. На практических занятиях методы и средства рассматривают более подробно, разбирают конкретные примеры, решают задачи и упражнения.

Далее представлено описание работы со студентами по каждому из пяти уровней системы Гарретта.

2.2. Уровень стратегии

Согласно системе Гарретта на уровне стратегии должно быть получено абстрактное описание создаваемого или исследуемого программного продукта с учетом ожиданий заинтересованных лиц (потенциальных пользователей, заказчиков, экспертов предметной области и т. п.). Первоначально планировалось, что выданное всей группе студентов учебное задание (кейс) будет прорабатываться в командах.

Каждая из команд будет самостоятельно разрабатывать гипотезы о персонажах целевой аудитории, планировать опросы и т. п. [14]. Однако оказалось, что команды допускают многочисленные ошибки моделирования (стереотипные решения, проектирование "под себя", "пластилиновые"¹ персонажи, более четырех второстепенных персонажей и др.) по причине размытого видения будущего продукта. Тогда был изменен формат занятий — с командного до общегруппового — и применен проблемно-центрированный подход [15]. После выдачи учебного задания и недельного периода на подготовку проходят дискуссии о предметной области с всеми студентами группы, где обсуждаются факты несоответствия между желаемым и реальным ее состоянием. Выявленные противоречия записываем как "проблемы" на стикерах и формируем флипчарт "Проблемное поле предметной области". Далее выполняем структурирование и ранжирование выявленных проблем. При структурировании решаются задачи поиска и анализа причинно-следственных связей проблемного поля, декомпозиции сложных проблем до более простых и/или их перегруппировки и ранжирования по степени важности. В итоге в центре проблемного поля оставляем главную проблему, решение которой может быть выполнено в терминах и средствами информатики: автоматизацией бизнес-процессов, разработкой информационных систем, приложений и т. п. Для наглядности результаты анализа представляем в виде дерева проблемы [15]. Иллюстрация дерева хорошо отражает причинно-следственные связи главной проблемы: корни обозначают иерархию ее причин, ветви являются следствиями, а листья — симптомами, явными признаками, по которым мы определяли наличие проблемы в начале анализа.

Полученная иллюстрация дерева облегчает формирование согласованного видения цели продукта как средства разрешения главной проблемы. Цели по устранению причин проблемы становятся подцелями и задачами продукта. А листья кроны исходного дерева проблем, отражавшие симптомы и признаки проблемы, в дереве целей становятся индикаторами успешности решения задач (рис. 2).

Полученные материалы (флипчарт, стикеры, иллюстрации деревьев) в совокупности успешно формируют видение продукта и дают непротиворечивую, более конкретную информацию для создания описательной модели пользователя — персонажа: его цели и мотивация для использования продукта, а также контекст использования продукта (когда, почему и как).

Приведем пример постановки и решения задачи формирования уровня стратегии. В рамках одного из семинаров со студентами был поднят вопрос "Почему молодежь слабо знает и мало интересуется историей родного города?" В ходе дискуссии были выявлены несколько существенных причин, связанных с доступностью и привлекательностью исторической информации. В частности, существующие информационные ресурсы (библиотека, музейные экспозиции, традиционные экскурсии) проигры-

¹ Проблема «пластилинового» пользователя возникает при ошибках проектирования, например, когда в некоторых ситуациях пользователь представляется проектировщиком как новичок, а в некоторых — как опытный пользователь.

ДЕРЕВО ПРОБЛЕМЫ

Следствия проблемы

Прямые эффекты

Центральная
проблема

Прямые причины

Косвенные причины



ДЕРЕВО ЦЕЛЕЙ

Индикаторы
достижения цели

Основная цель

Подцели

Задачи, необходимые
условия

Рис. 2. Иерархическая структура дерева проблемы и дерева целей

вают современным средствам развлечения и досуга (соцсети, компьютерные игры, кино и т. д.) и мало востребованы молодежью.

Проблема несовременного состояния информационных ресурсов об истории города стала главной в учебном проекте. Было предложено и проработано оригинальное решение — создание мобильного приложения "Квест-экскурсия". Основная цель проекта — повышение у молодежи (14–35 лет) интереса к изучению истории родного города. Учитывая возраст и интересы целевой аудитории, выбрана именно форма квест-экскурсий, поскольку она в большей мере способна привлечь внимание молодого поколения, нежели классический вариант экскурсии, в котором часто практически нет обратной связи и активности со стороны слушателя. Формат квеста примечателен тем, что помимо классических функций экскурсии (познавательная, образовательная, культурно-досуговая), он несет ряд социально значимых функций. Участие в групповой игре и выполнение различных заданий на пути позволяет настроить коллектив на командную работу, выявить скрытые таланты у участников, поднять их самооценку, пробудить интерес к изучению истории родного города [16].

Далее было разработано моделируемое представление о пользователе приложения — ключевом персонаже. Это Яна Гаврилова, молодая спортивная девушка, студентка первого курса, открытая веселому времяпровождению с друзьями на открытом воздухе. С началом учебного года в институте Яна обзавелась большим числом новых знакомств. Широкий круг знакомств и интересов располагает к поиску альтернативных вариантов досуга, где можно не только общаться и обсуждать вместе различные темы, но и узнавать что-то

новое, командно решать задачи. В хорошую погоду Яна предпочитает объединиться с друзьями-единомышленниками и пройти увлекательный квест по городу с помощью мобильного приложения.

Претензии и болевые точки персонажа:

- скучно просто гулять по городу;
- неудобно и некомфортно ходить по улицам за экскурсоводом с толпой других людей;
- отсутствие доступного культурного досуга.

На рис. 3 представлена наглядная визуализация — фотоколлаж Яны Гавриловой [17].

Достоинство фотоколлажа в описании модели пользователя — он передает больше информации, которую трудно выразить словами.



Рис. 3. Фотоколлаж ключевого персонажа Яны Гавриловой

Приведенный пример иллюстрирует практические аспекты использования результатов внедренного в учебный процесс проблемно-центрированного подхода. Также хорошо прослеживается логика дальнейшей работы над учебным проектом — поиск ответов на вопросы о функциональных возможностях и требованиях к контенту продукта.

2.3. Уровень возможностей

Согласно системе Гарретта на данном уровне должны быть получены представления о функциональных возможностях и требованиях к контенту продукта. Однако учитывая результаты предыдущего этапа по моделированию персонажей, задачи уровня были дополнены анализом существующих продуктов-аналогов. Помимо списка функциональных возможностей и особенностей контента необходимо получить представление о возможных проблемах и причинах недовольства пользователей существующими продуктами.

Для этого были использованы инструментальные механизмы (анкеты, опросные листы, методы обработки) разработанной на нашей кафедре методологии экспертизы программных продуктов [18, 19]. Методология учитывает ряд показателей, разработанных Шнейдерманом для оценки удобства использования интерфейса, опирается на комплекс стандартов и требований в области оценки эргономичности программных продуктов, на исследования в области инженерной психологии и эргономики [20—22]. Оцениваемые показатели методологии:

- скорость работы программного продукта;
- число ошибок пользователей;
- скорость обучения;
- субъективная удовлетворенность;
- техническая эстетика.

Балльные оценки формировались по каждому показателю и по системе весовых коэффициентов пересчитывались в итоговую оценку эргономичности программного продукта. Данный подход помимо сравнимых количественных показателей позволяет выявлять конкретные ошибки и недостатки интерфейсных решений, прогнозировать трудности взаимодействия с продуктом и делать выводы о возможных причинах недовольства пользователей.

В учебном проекте по разработке мобильного приложения "Квест-экскурсия" в исследование эргономичности были включены следующие приложения-аналоги: City Stories, Ubego, Surprise Me (представленные на момент публикации в Google Play в разделе "Экскурсии и аудиогиды"). Полученные результаты анкет показывают лучшую, чем у аналогов реализацию пользовательского интерфейса в приложении City Stories. Приложение не требует дополнительного обучения, интерфейс хорошо структурирован и понятен, что обеспечивает уверенное эффективное взаимодействие. В то же время пользовательский опыт приложений Ubego и Surprise Me спроектирован неоднородно: некоторые функции лежат на поверхности, а поиск или неожиданное местоположение других заставляет совершать ошибки при первом использовании. Существенными минусами при работе с данными приложениями стала плохо структурированная информационная архитектура приложений, что и привело к низкой оценке удовлетворенности.

Стоит отметить, что общий характер приложений такого типа развлекательный, что означает в первую очередь то, что ожидание пользователей — это получение удовольствия от взаимодействия с продуктом и процесса прохождения заданий. Всеми экспертами, так или иначе, были отмечены проблемные вопросы с регистрацией во всех трех рассматриваемых продуктах. Настроенный в позитивном ключе пользователь, будучи разочарованным в долгой и утомительной регистрации, может потерять интерес к продукту уже на старте и предпочесть другой вид досуга разбирательствам, как работает приложение или попыткам понять, как пройти процесс регистрации.

Таким образом, важно ускорить и облегчить процесс регистрации пользователя, или даже дать ему возможность проходить задания без регистрации в тестовом режиме.

На основе выводов были даны следующие рекомендации для интерфейса разрабатываемого приложения:

- упростить процесс регистрации, либо сделать возможным пользование приложением без привязки к аккаунту;
- проработать функцию сортировки квестовых заданий, добавить фильтр и возможность поиска для лучшей ориентации пользователя в контенте;
- тщательно продумать цветовую палитру приложения;
- дополнительно выверить логичность структуры приложения;
- использовать пиктограммы одного стиля, подбирать их по смыслу и следить за выравниванием (аккуратность в работе с графическими элементами приложения в целом влияет на общую удовлетворенность пользователя продуктом).

Итогом работы данного этапа стал список с иерархической нумерацией строк, в котором вся функциональность приложения формализована: <кто> <что делает> <с чем>. В результате функциональные требования идентифицируются и прослеживаются (по уникальным номерам) на последующих этапах разработки.

2.4. Уровень структуры

На этом уровне описанные ранее возможности продукта приобретают структуру и информационное наполнение. Должен быть документирован процесс взаимодействия пользователя с продуктом и информационная архитектура интерфейса. По Гарретту главным инструментом документирования является схема, в которой важно отразить концептуальные связи: какие категории сгруппированы, а какие остаются сами по себе, как согласуются шаги в данной процедуре взаимодействия и т. п.

Однако вместо техники Гарретта Visual Vocabulary сейчас используется User Flow — это гибрид классической блок-схемы и элементов визуального интерфейса. В основе User Flow лежат задачи, которые выполняет пользователь, а также возможные альтернативные исходы [23]. По сравнению со схемами Гарретта User Flow более функциональный инструмент — в нем есть точки принятия решений, пути, режимы и петли, с помощью которых можно иллюстрировать все взаимодействие с продуктом и, что важно, отслеживать, все ли процессы в продукте имеют логическое завершение.

В нашем учебном проекте на уровне структуры сначала была разработана достаточно простая блок-схема информационной архитектуры, которую затем дополнили эскизами визуального интерфейса (вайрфреймами) для получения User Flow.

User Flow наглядно демонстрирует распределение сервисов создаваемого продукта между окнами и позволяет создать цельное представление концепции конечного дизайна, проверить ошибки в логике и деталях взаимодействия.

2.5. Уровень компоновки

Задача уровня по Гарретту — на основе архитектурной схемы предыдущего уровня, предлагающей общий взгляд на проект, необходимо документировать детальные прототипы всех экранных форм продукта. Должны быть решены вопросы наиболее эффективного расположения различных элементов интерфейса — кнопок, полей ввода/вывода, текстовых блоков и иллюстраций на каждой экранной форме.

Значимость прототипов страниц определяется тем, как они объединяют все три элемента уровня структуры: дизайн интерфейса — через расположение и выбор элементов интерфейса; дизайн навигации — через идентификацию и задание главных навигационных систем; информационный дизайн — через размещение и расстановку по приоритету информационных компонентов. Собрав эти три составляющие в одном документе, прототип способен задать компоновку, в полной мере опирающуюся на концептуальную структуру сайта и указывающую путь к визуальному дизайну.

Однако находясь на стыке информационной архитектуры и визуального дизайна, прототипы страниц становятся предметом множества дискуссий и споров. При поиске компромисса требуется рассмотреть альтернативные варианты решений и при этом снизить затраты ресурсов на их разработку.

Поэтому прототипирование интерфейса в нашем практикуме начинается с вайрфреймов (*wireframe*) — низко детализированного, схематичного представления компоновки элементов интерфейса каждого экрана продукта, выполненное, как правило, в оттенках серого [24].

Поскольку вайрфреймы создаются быстро, то значительную часть времени можно провести за обсуждениями и размышлениями. Следует отметить, что обсуждается в том числе и соответствие альтернатив критериям эргономичности [18, 19]. Так, например, если было разработано несколько логически верных вариантов компоновки элементов, то выбрать предпочтительнее решение, требующее минимальных затрат времени пользователя. При этом оценка временных затрат для каждого варианта компоновки может быть рассчитана априорно по методу GOMS [25] или определена экспериментально тестированием производимости прототипа. Согласованные в итоге решения используются далее как документация по проекту, сопровождаемая текстовыми комментариями.

Далее на основе вайрфреймов создается прототип. Мы перешли на использование инструмента Adobe XD, позволяющего создавать средне или высоко детализированное представление конечного продукта в виде кликабельного интерактивного прототипа. Пользователь может оценить интерфейс, протести-

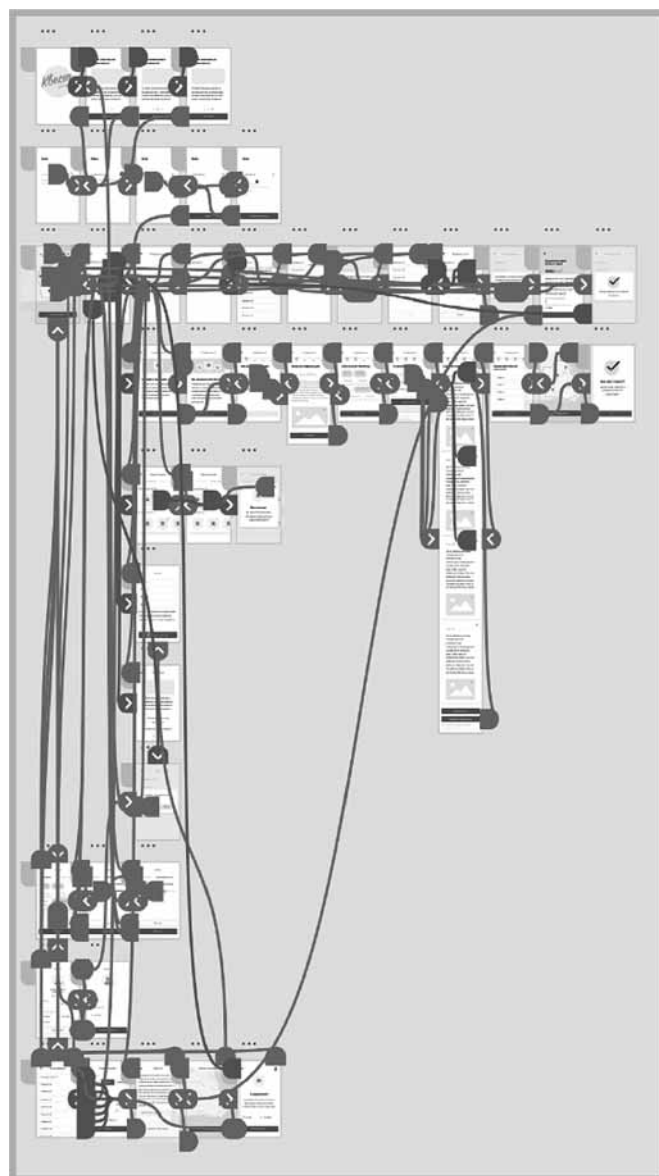


Рис. 4. Пример работы с прототипом учебного проекта

ровав основные способы взаимодействия так, как если бы это был готовый продукт.

На рис. 4 в качестве примера приведена иллюстрация прототипа нашего учебного проекта: вайрфреймы связаны друг с другом переходами.

Потенциал прототипов в Adobe XD полностью раскрывается при пользовательском тестировании. С помощью данной имитации можно получить много материала для улучшения юзабилити еще до фактического начала разработки.

2.6. Уровень поверхности

Этот уровень по Гарретту определяет визуальное представление логического порядка элементов, образующих компоновку продукта. Созданный ранее прототип получает цветовое оформление, добавляются декоративные элементы, навигационные и информационные пиктограммы.

Задача уровня — не заикливаясь на оценках визуального дизайна, исключительно в терминах

эстетической привлекательности, следует сосредоточить внимание на результативности интерфейса: насколько эффективно дизайн поддерживает цели, определенные на каждом из нижележащих уровней. Частые проблемы, затрудняющие взаимодействие и достижение пользовательских целей: перегруженность элементами декора, несогласованные или неудачные цветовые комбинации с фоном, непродуманные скевоморфизм¹ или параллакс-эффект², низкий контраст элементов и дизайнерские шрифты в плоском дизайне. В результате проверенные ранее варианты компоновки на уровне поверхности могут перестать эффективно работать: пользователи могут кликать некликабельные элементы декора или наоборот, пропускать элементы управления (например, кнопки), считая их статичными элементами. Требуется тщательное пользовательское тестирование макета.

В связи с этим весьма перспективными в задачах исследования пользовательского опыта, в том числе его бессознательных компонентов, представляются инструментальные методы окулографии и электроэнцефалографии, реализуемые сейчас относительно недорогими и компактными устройствами, в частности, айтрекером Eye Tribe и нейрогарнитурой NeuroSky MindWave.

Айтрекинг — это метод окулографии, позволяющий отслеживать взгляд человека и определять его координаты. Несмотря на то, что отечественные исследователи отмечают ограниченность использования айтрекеров при юзабилити-тестировании [26] и необходимость стандартизации метрологических параметров айтрекинга [27], за рубежом с начала 2000-х гг. он активно используется для анализа пользовательских интерфейсов и веб-дизайна [28, 29]. Айтрекинг позволяет дополнить традиционные юзабилити-исследования за счет анализа связи между кликами мышкой и визуальным поведением пользователя, позволяет выявить компоненты веб-сайта, привлекательные для пользователя, вызывающие трудности и не замечаемые им, позволяет оценить эффективность поиска, правильность концепции бренда и других аспектов веб-дизайна [28].

В учебном процессе задействованы два прибора Eye Tracker модели Eye Tribe. В качестве пользователей-экспертов участвуют сами магистранты (10—12 человек), которым ставится задача по тестированию ключевой функциональности программного продукта. Интерактивность макета пользовательского интерфейса продукта реализуется средствами программного обеспечения OGAMA [30], позволяющего из статичных изображений страниц получить визуально скролируемую и кликабельную демонстрацию. Порядок подготовки и проведения процедуры исследования: запуск эксперимента в приложении OGAMA; ввод идентификационных данных о пользователе; индивидуальная калибровка прибора (каж-

дый пользователь мог выбрать комфортную позу, положение монитора); запуск записи отслеживания движения глаз; решение пользователем тестовой задачи; завершение записи [31].

По данным, полученным от приборов, составляется тепловая карта — пространственная характеристика движения глаз, отражающая плотность точек фиксации взгляда. Красный и желтый цвета отражают зоны, на которых внимание было сфокусировано дольше остальных. Анализируются тепловые карты, полученные от всех испытуемых, и проводится автоматическая статистическая обработка результатов. В качестве дополнительного этапа иногда проводится сравнительный анализ результатов работы с другими аналогичными программными продуктами. Результаты исследования позволяют в целом оценить эффективность решения тестовой задачи, наглядно обозначить визуальные фрагменты веб-сайта, которым пользователь уделяет наибольшее/наименьшее внимание, и проанализировать их функциональность (например, часты ситуации, когда декоративные элементы интерфейса излишне долго удерживают внимание пользователя или наоборот, кликабельные, но слабо различимые функциональные элементы пропускаются).

Беспроводная нейрогарнитура NeuroSky MindWaveMobile относится к компактным недорогим устройствам одноканальной электроэнцефалографии (ЭЭГ). Bluetooth-модуль позволяет сопрягать гарнитуру с планшетом, смартфоном или ноутбуком. Гарнитура позволяет изучать процессы ментальной активности пользователя, тренировать его способности самоконтроля эмоционального состояния, улучшать работоспособность и устойчивость к стрессам [32—39].

В устройстве реализована аппаратная технология съема и аналого-цифрового преобразования сигнала ThinkGear. Частотный и амплитудный анализ регистрируемых сигналов позволяет дифференцировать пять традиционных ритмов головного мозга: дельта, тета, альфа, бета и гамма [40, 41]. Наблюдаемое у пользователя соотношение ритмов может использоваться для диагностики его функционального состояния и учитываться при интерпретации результатов деятельности. Например, фиксируемые в окулографическом исследовании ошибочные действия пользователя оказывались обусловлены не проблемами в проектировании или реализации пользовательского интерфейса, а повышенными уровнями тревожности, психоэмоционального напряжения или депрессии данного испытуемого [32].

Однако профессиональная расшифровка, анализ и полная интерпретация ЭЭГ представляют собой очень сложные задачи, требующие от исследователя специальной квалификации. Поэтому в устройстве NeuroSky MindWave реализован алгоритм eSense [42], позволяющий упростить интерпретацию сигналов за счет выделения всего двух обобщенных показателей — уровня концентрации внимания (*Attention*) и уровня релаксации (*Meditation*), каждый из которых измеряется по 100-балльной шкале. С помощью визуализации биологической обратной связи в приложении BrainWave Visualizer испытуемые добивались сочетания максимального уровня концентрации внимания (85...100) со спокойным "медитативным"

¹ Скевоморфизм в дизайне — это использование формы, дизайна и стиля реальных объектов при создании виртуальных. Например, изображение кнопки в приложении похоже на фотографию реального объекта.

² Параллакс-эффект — это техника в веб-разработке, предполагающая разбиение изображений на слои. При прокрутке окна браузера слои движутся с разной скоростью, создавая у пользователя иллюзию объема.

состоянием "беспристрастного наблюдателя" (уровень релаксации 65...80). И только потом приступали к окулографическому исследованию.

Заключение

Обобщая представленные выше результаты, можно сделать вывод о по-прежнему высокой практической значимости системы элементов Дж. Гарретта как наглядного инструментария для обучения классическому процессу UX. Благодаря апробированному на практике набору эффективных методов и средств, дополняющих систему элементов, создаются условия для подготовки в вузе востребованных на рынке труда специалистов по UX/UI-дизайну.

Список литературы

1. **Компаниец В. С.** Эргодизайн пользовательского интерфейса: опыт проектирования междисциплинарной магистерской программы // Безопасность и развитие личности в образовании. Материалы III Всероссийской научно-практической конференции. Таганрог, 19—21 мая 2016 г. — Таганрог: Изд-во ЮФУ, 2016. — С. 199—207.
2. **Компаниец В. С., Лызь А. Е.** Эргодизайн пользовательского интерфейса: методы юзабилити-исследований // Инженерный вестник Дона. — 2017. — Т. 46. № 3 (46). — С. 56—64.
3. **Компаниец В. С., Казанская А. Ю.** Опыт реализации магистерской образовательной программы "Эргодизайн пользовательского интерфейса" // III Международная конференция "Человеческий фактор в сложных технических системах и средах: ЭРГО 2018". — СПб.: СПбГЭТУ "ЛЭТИ", 2018. — С. 60—68.
4. **Designing for Product Strategy**, O'Reilly Media, 2016. — 95 p.
5. **Gothelf J.** Lean UX: Applying Lean Principles to Improve User Experience, O'Reilly Media 2013. — 152 p.
6. **Vigil D.** Dual Track Agile: Level Up Enterprise Product Design with UX Part 3. 2019. URL: <https://medium.com/@dannylvigil/dual-track-agile-level-up-enterprise-product-design-with-ux-part-3-25cafc1d6aa5>
7. **Дирекова Н.** Официальное руководство по Google Design Sprint (методология дизайн-спринтов). URL: <https://ux.pub/officialnoe-rukovodstvo-po-google-design-sprint-metodologiya-dizajn-sprintov/>
8. **Kleynhans H.** 9 Principles for Lean UX. URL: <https://medium.com/@geekrebel/9-principles-for-lean-ux-64d9fc4da41b#vqxh3319m>
9. **The Essential Elements of Successful UX Design: Best Practices for Meaningful Product Design.** 2015. URL: <https://www.uxpin.com/studio/ebooks/ux-design-successful-elements-for-products/>
10. **Бэнк К.** Минимально жизнеспособный продукт: инструкция. URL: <https://cmsmagazine.ru/journal/items-the-minimum-viable-product-user/>
11. **Armstrong I.** The Evolution of UX Process Methodology: The UX Process is Confusing, Even to Most Designers. URL: <https://uxplanet.org/the-evolution-of-ux-process-methodology-47f52557178b>
12. **Сао J.** The Skeptical Designer's UX Process. 2016. URL: <https://www.uxpin.com/studio/blog/the-skeptic-designer-ux-process/>
13. **Гарретт Д.** Веб-дизайн: книга Джесса Гарретта. Элементы опыта взаимодействия. — СПб.: Символ-Плюс, 2008. — 186 с.
14. **Купер А.** Интерфейс: Основы проектирования взаимодействия. — СПб.: Питер, 2018. — 720 с.
15. **Трофимова Л. А., Трофимов В. В.** Управленческие решения. — СПб.: Изд-во СПбГУ ИТМО, 2011. — 190 с.
16. **Алексеева Н. Д., Рябова Е. В.** Квест-экскурсия как инновационная форма экскурсионной деятельности // Вектор науки Тольяттинского государственного университета. Серия: Педагогика, психология. — 2015. — № 1. — С. 14—17.
17. **Задерака А. Ю.** Пользовательский интерфейс мобильного приложения "Квест-экскурсия" Выпускная квалификационная работа. — Таганрог: ИКТИБ ЮФУ, 2019. — 119 с. URL: <https://hub.lib.sfedu.ru/repository/material/801264211/>
18. **Картавенко М. В.** Методология эргономической оценки персональных антивирусов. URL: <https://www.anti-malware.ru/node/9570>
19. **Картавенко М. В.** Методология эргономической оценки программного обеспечения в области информационной безопасности // Известия Южного федерального университета. Технические науки. — 2012. — Т. 135, № 10. — С. 199—204.
20. **Баканов А. С., Обознов А. А.** Эргономика пользовательского интерфейса: от проектирования к моделированию человеко-компьютерного взаимодействия. — М.: Институт психологии РАН, 2011. — 176 с.
21. **Речинский А. В., Сергеев С. Ф.** Разработка пользовательских интерфейсов. Юзабилити-тестирование интерфейсов информационных систем: уч. пособие. — СПб.: Изд-во Политехнич. ун-та, 2012. — 145 с.
22. **Сергеев С. Ф.** Метод тестирования и оптимизации интерфейсов информационных систем — СПб.: Изд-во университета ИТМО, 2013. — 117 с.
23. **Baraniak P. (Thalion)** User Journey Maps or User Flows, what to do first? 2018. URL: <https://medium.com/sketch-app-sources/user-journey-maps-or-user-flows-what-to-do-first-48e825e73aa8>
24. **Григорьев А.** Вайрфреймы, прототипы и мокапы. 2013. URL: <http://projectorator.ru/wireframes-prototypes-mockups/>
25. **Раскин Д.** Интерфейс: новые направления в проектировании компьютерных систем. Пер. с англ. — СПб.: Символ-Плюс, 2004. — 272 с.
26. **Костин А. Н.** Круглый стол "Юзабилити как новое направление исследований в инженерной психологии" // Психологический журнал. — 2011. — Т. 32, № 4. — С. 113—124.
27. **Фазылыянова Г. И., Балалов В. В.** Применение метода айтрекинга для оценки качества графической и мультимедийной продукции // Наука и Мир. — 2014. — Т. 3, № 3 (7). — С. 172—179.
28. **Bojko A., Stephenson A.** It's All in the Eye of the User: How eye tracking can help answer usability questions// User experience. — 2005. — Vol. 4, No. 1. — P. 4—6.
29. **Goldberg J. H., Stimson M. J., Lewenstein M.** et al. Eye Tracking in Web Search Tasks: Design Implications// In Eye Tracking Research & Applications (ETRA) Symposium. New Orleans, LA USA. — 2002. — P. 51—58.
30. **OGAMA** — Open Gaze And Mouse Analyzer. URL: <http://www.ogama.net/sites/default/files/pdf/OGAMA-DescriptionV25.pdf>
31. **Лызь А. Е., Компаниец В. С.** Айтрекинг как метод оценки пользовательских интерфейсов // Новые задачи технических наук и пути их решения: Сборник статей по итогам международной научно-практической конференции. — Оренбург, 2017. — С. 31—33.
32. **Рувинская А. О.** Исследование мозговой активности с помощью нейрорегистратуры NeuroSky MindWave //Сборник статей II Всероссийской научно-практической конференции молодых ученых, аспирантов, магистрантов и студентов "Информационные системы и технологии: фундаментальные и прикладные исследования". — Таганрог, 2017. — С. 101—104.
33. **MindWave** Mobile User Guides. RL: <https://ru.scribd.com/doc/191259445/Mindwave-Mobile-User-Guide>
34. **Hu J., Nakanishi M., Matsumoto K.** et al. A Method of Usability Testing by Measuring Brain Waves //Proceedings of the International Symposium on Future Software Technology (ISFST-2000). Guiyang, China, 2000. — P. 159—164.
35. **Lee H., Seo S.** A Comparison and Analysis of Usability Methods for Web Evaluation: The Relationship Between Typical Usability Test and Bio-Signals Characteristics (EEG, ECG) //

Proceedings of the 2010 conference of the Design Research Society. Montréal, Canada, 2010. — P. 893–904.

36. **Chuang J., Nguyen H., Wang C., Johnson B. I.** Think, Therefore I Am: Usability and Security of Authentication Using Brainwaves // Lecture Notes in Computer Science, 2013. — Vol. 786. P. 1–16.

37. **Bos D. P.-O., Reuderink B., van de Laar B.** et al. Human-Computer Interaction for BCI Games: Usability and User Experience // Proceedings of International Conference on Cyberworlds. Singapore, 2010. — P. 277–281.

38. **Kimura M., Uwano H., Ohira M., Matsumoto K.** Toward Constructing an Electroencephalogram Measurement Method for Usability Evaluation // Proceedings of the 13th International Conference on Human-Computer Interaction. — San Diego, USA. Part I: New Trends. 2009. — P. 95–104.

39. **Alfimtsev A. N., Basarab M. A., Devyatkov V. V., Levanov A. A.** A New Methodology of Usability Testing on the Base of the Analysis of User's Electroencephalogram // Journal of Computer Sciences and Applications. — 2015. — Vol. 3, No. 5. — P. 105–111. URL: <http://pubs.sciepub.com/jcsa/3/5/1>

40. **Кирой В. Н., Ермаков П. Н.** Общая характеристика ритмов ЭЭГ человека // Электроэнцефалограмма и функциональные состояния человека. — Ростов-на-Дону: Изд-во Рост. ун-та, 1998. — С. 48–76.

41. **Larsen E. A.** Classification of EEG Signals in a Brain-Computer Interface System. Norwegian University of Science and Technology. Department of Computer and Information Science. NTNU. 2011. — 58 p.

42. **NeuroSky** Developers Program. eSense (tm) Meters. URL: <http://developer.neurosky.com/>

Systematization of User Experience Design Procedures in the Applied Informatics Master's Training

V. S. Kompaniets, kompaniets@sfedu.ru, **A. Yu. Kazanskaya**, akazanskaya@sfedu.ru, **A. E. Lyz**, aelyz@sfedu.ru, **A. O. Grigoriev**, agrigorev@sfedu.ru, Engineering and Technology Academy, Federal State Autonomous Educational Institution of Higher Education Southern Federal University, Taganrog, 347928, Russian Federation

Corresponding author:

Kompaniets Vitaly S., Associate Professor, Engineering and Technology Academy of Southern Federal University, Taganrog, 347928, Russian Federation
E-mail: kompaniets@sfedu.ru

Received on February 18, 2020

Accepted on March 11, 2020

The design and implementation experience of the university master program "User Interface Ergodesign" is described. The competence model of the graduate is formed in this program according to the current requirements of the new generation of the federal standard (federal state educational standard of higher education hereafter — FSES HE 3++) and in accordance with the functional map of the profile professional standard "Specialist in the design of graphic and user interfaces". The content of the curriculum, in which an important role is assigned to the professional module "UX/UI-design", is briefly presented. The five-year experience of teaching students in the classical level system of elements of J. Garrett is described and illustrated in detail with a step-by-step example. It is shown how the framework of system levels can be filled with a minimum set of specific methods and tools that have proven their effectiveness and didactic value.

It is concluded that J. Garrett's system of elements still has high practical importance as a visual tool for teaching students the classical UX process. Thanks to the approach, that was proven in practice, the conditions for university training of UX/UI design specialists, demanded on the labor market, are created.

Keywords: classical process UX, the UI, the elements of J. Garrett, user interface Ergodesign, eye tracking, Eye Tribe, neuro headset, NeuroSky, electroencephalography

For citation:

Kompaniets V. S., Kazanskaya A. Yu., Lyz A. E., Grigoriev A. O. Systematization of User Experience Design Procedures in the Applied Informatics Master's Training, *Programmная Ingeneria*, 2020, vol. 11, no. 3, pp. 183–192

DOI: 10.17587/prin.11.183-192

References

1. **Kompaniets V. S.** Ergodesign of the user interface: experience of designing an interdisciplinary master's program. *Proceedings of the III All-Russian Scientific and Practical Conference "Safety and Personal Development in Education"*, Taganrog, Southern Federal University, Russia, May 19–21, 2016, pp. 199–207 (in Russian).

2. **Kompaniets V. S., Lyz A. E.** User interface ergodesign: methods of usability research. *Inzhenernyy Vestnik Dona*, 2017, vol. 46. no. 3, pp. 56–64 (in Russian).

3. **Kompaniets V. S., Kazanskaya A.Yu.** The experience of the master's educational program implementation "User Interface Ergodesign". *Proceedings of III International Conference "The Human Factor in Complex Technical Systems and Environments (ERGO'18)*. St. Petersburg, Russia, Jul. 4–7, 2018. pp. 60–68 (in Russian).

4. **Designing** for Product Strategy, O'Reilly Media, 2016, 95 p.

5. **Gothelf J.** *Lean UX: Applying Lean Principles to Improve User Experience*. O'Reilly Media, 2013, 152 p.

6. **Vigil D.** Dual Track Agile: Level Up Enterprise Product Design with UX, available at: <https://medium.com/@dannylvigil/>

dual-track-agile-level-up-enterprise-product-design-with-ux-part-3-25cafd6aa5

7. **Direkova N.** Official Guide to Google Design Sprint (design sprint methodology), available at: <https://ux.pub/oficialnoe-rukovodstvo-po-google-design-sprint-metodologiya-dizajn-sprintov/> (in Russian).
8. **Kleynhans H.** 9 Principles for Lean UX available at: <https://medium.com/@geekrebel/9-principles-for-lean-ux-64d9fc4da41b#.vqxh3319m>
9. **The Essential Elements of Successful UX Design: Best Practices for Meaningful Product Design**, available at: <https://www.uxpin.com/studio/ebooks/ux-design-successful-elements-for-products/>
10. **Bank C.** Minimum Viable Product: instruction, available at: <https://cmsmagazine.ru/journal/items-the-minimum-viable-product-user/>
11. **Armstrong I.** The Evolution of UX Process Methodology: The UX Process is Confusing, Even to Most Designers, available at: <https://uxplanet.org/the-evolution-of-ux-process-methodology-47f52557178b>
12. **Cao J.** The Skeptical Designer's UX Process, available at: <https://www.uxpin.com/studio/blog/the-skeptic-designer-ux-process/>
13. **Garrett J. J.** *Web Design: A Book by Jesse Garrett. Elements of interaction experience*, St. Petersburg, Simvol Plyus, 2008, 186 p. (in Russian).
14. **Cooper A.** *Interface: Basics of interaction design*, St. Petersburg, Piter, 2018, 720 p. (in Russian).
15. **Trofimova L. A., Trofimov V. V.** *Management decisions (methods of adoption and implementation): textbook*, St. Petersburg, Izd. Sankt-Peterburgskogo Gosudarstvennogo Universiteta Ekonomiki I Finansov, 2011, 190 p. (in Russian).
16. **Alekseeva N. D., Ryabova E. V.** Quest excursion as an innovative form of excursion activity, *Vektor Nauki Tol'yatinskogo Gosudarstvennogo Universiteta. Seriya: Pedagogika, Psikhologiya*, 2015, no. 1, pp. 14–17 (in Russian).
17. **Zaderaka A. Yu.** The user interface of the Quest-excursion mobile application: Final qualification work, Taganrog, Southern Federal University, 2019, 119 p., available at: <https://hub.lib.sfedu.ru/repository/material/801264211/> (in Russian).
18. **Kartavenko M. V.** Methodology of the ergonomics test of personal antiviruses, available at: <https://www.anti-malware.ru/node/9570> (in Russian)
19. **Kartavenko M. V.** Methodology of ergonomic software evaluation in the field of information security, *Izv. Yuzhnogo Federal'nogo Universiteta. Tekhnicheskiye Nauki*, 2012, vol. 135, no. 10, pp. 199–204 (in Russian).
20. **Bakanov A. S., Oboznov A. A.** *Ergonomics of the user interface*, Moscow, Institut Psikhologii RAN, 2011, 176 p. (in Russian).
21. **Rechinsky A. V., Sergeev S. F.** *Development of user interfaces. Usability testing of information system interfaces* St. Petersburg, Izd. Politehnicheskogo Universiteta, 2012, 145 p. (in Russian).
22. **Sergeev S. F.** *Methods of testing and optimizing the interfaces of information systems: a training manual*, St. Petersburg, Izd. Universiteta ITMO, 2013, 117 p. (in Russian).
23. **Baraniak P. (Thalion)** User Journey Maps or User Flows, what to do first? available at: <https://medium.com/sketch-app-sources/user-journey-maps-or-user-flows-what-to-do-first-48e825e73aa8>
24. **Grigoriev A.** Wireframes, prototypes and mockups, available at: <http://projectorat.ru/wireframes-prototypes-mockups/>
25. **Raskin J.** *Interface: new directions in the design of computer systems*, St. Petersburg, Simvol Plyus, 2004, 272 p. (in Russian).
26. **Kostin A. N.** Round table "Usability as a new direction of research in engineering psychology", *Psikhologicheskij Zhurnal*, 2011, vol. 32, no. 4, pp. 113–124 (in Russian).
27. **Fazylyzyanova G. I., Balalov V. V.** Application of Eyetracking method for assessing the quality of graphic and multimedia products, *Science and World*, 2014, vol. 3, no. 7, pp. 172–179 (in Russian).
28. **Bojko A., Stephenson A.** It's All in the Eye of the User: How eye tracking can help answer usability questions, *User experience*, 2005, vol. 4, no. 1, pp. 4–6.
29. **Goldberg J. H., Stimson M. J., Lewenstein M., Scott N., Wichansky A. M.** Eye Tracking in Web Search Tasks: Design Implications, *Proceedings of the 2002 Symposium Eye Tracking Research & Applications (ETRA)*, New Orleans, LA USA, 2002, pp. 51–58.
30. **GAMA** — Open Gaze And Mouse Analyzer, available at: <http://www.ogama.net/sites/default/files/pdf/OGAMA-DescriptionV25.pdf>
31. **Lyz A. E., Kompaniets V. S.** Eyetracking as a method for evaluating user interfaces. *Proceedings of the International Scientific and Practical Conference "New problems of technical sciences and ways to solve them"*, Orenburg, Aug. 28, 2017, pp. 31–33 (in Russian).
32. **Ruvinskaya A. O.** The study of brain activity using the NeuroSky MindWave neuro headset, *Proceedings of The II All-Russian scientific and practical conference of young scientists, graduate students, undergraduates and students "Information systems and technologies: fundamental and applied research"*, Taganrog, Southern Federal University, Oct. 23–27, 2017, pp. 101–104 (in Russian).
33. **MindWave** Mobile User Guides, available at: <https://ru.scribd.com/doc/191259445/Mindwave-Mobile-User-Guide>
34. **Hu J., Nakanishi M., Matsumoto K., Tagaito H., Inoue K., Shima K., Torii K.** A Method of Usability Testing by Measuring Brain Waves, *Proceedings of the International Symposium on Future Software Technology (ISFST-2000)*, Guiyang, China, 2000, pp. 159–164.
35. **Lee H., Seo S.** A Comparison and Analysis of Usability Methods for Web Evaluation: The Relationship Between Typical Usability Test and Bio-Signals Characteristics (EEG, ECG), *Proceedings of the conference of the Design Research Society*, Montréal, Canada, 2010, pp. 893–904.
36. **Chuang J., Nguyen H., Wang C., Johnson B.** I Think, Therefore I Am: Usability and Security of Authentication Using Brainwaves, *Lecture Notes in Computer Science*, 2013, vol. 7862, pp. 1–16.
37. **Bos D. P.-O., Reuderink B., van de Laar B., Gürkök H., Mühl C., Poel M., Heylen D., Nijholt A.** Human-Computer Interaction for BCI Games: Usability and User Experience, *Proceedings of International Conference on Cyberworlds*, Singapore, 2010, pp. 277–281.
38. **Kimura M., Uwano H., Ohira M., Matsumoto K.** Toward Constructing an Electroencephalogram Measurement Method for Usability Evaluation, *Proceedings of the 13th International Conference on Human-Computer Interaction*, San Diego, USA, 2009, pp. 95–104.
39. **Alfimtsev A. N., Basarab M. A., Devyatkov V. V., Levakov A. A.** A New Methodology of Usability Testing on the Base of the Analysis of User's Electroencephalogram, *Journal of Computer Sciences and Applications*, 2015, vol. 3, no. 5, pp 105–111, available at: <http://pubs.sciepub.com/jcsa/3/5/1>
40. **Kirov V. N., Ermakov P. N.** General characteristics of human EEG rhythms, *Electroencephalogram and functional state of a person*, Rostov-on-Don, Izdatel'stvo Rostovskogo Gos. Universiteta, 1998, pp. 48–76 (in Russian).
41. **Larsen E. A.** *Classification of EEG Signals in a Brain-Computer Interface System*, Norwegian University of Science and Technology, 2011, 58 p.
42. **NeuroSky** Developers Program. eSense (tm) Meters, available at: <http://developer.neurosky.com/>

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

Сдано в набор 15.04.2020 г. Подписано в печать 25.05.2020 г. Формат 60×88 1/8. Заказ П1320
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

V. A. Kalmychkov, I. V. Matveeva

«AUTOMATED GENERATION OF QUANTUM CIRCUIT SPECIFICATIONS
BASED ON REED – MULLER EXPRESSIONS»



Fig. 5. Direct arrangement of SWAP gates according to FPRM for 141 polarity

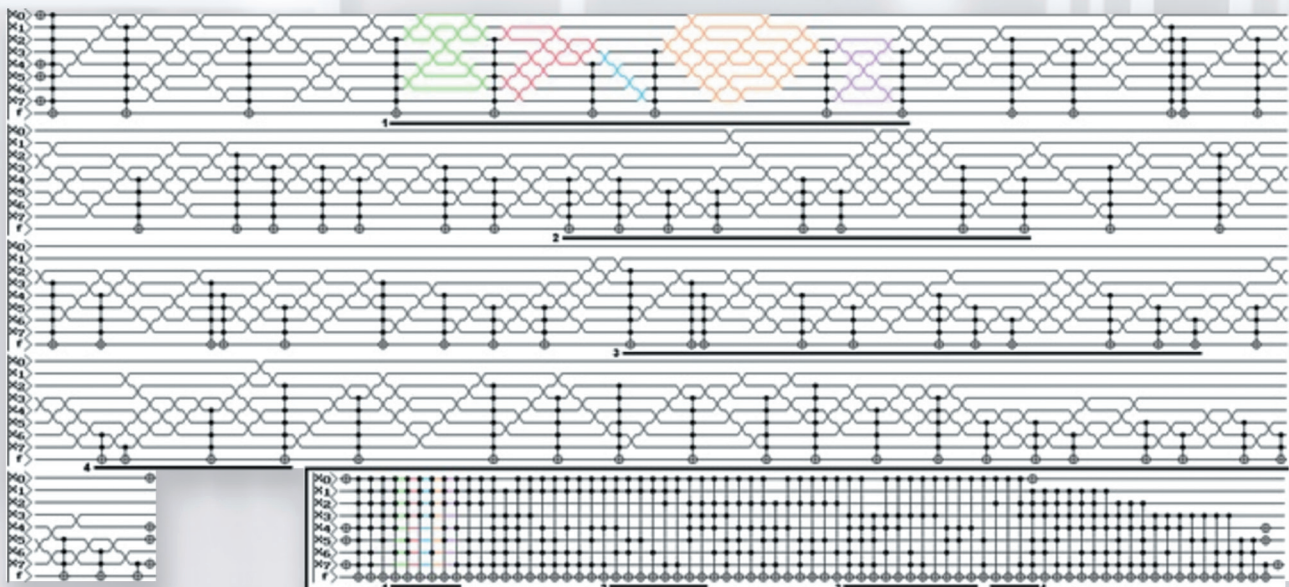


Fig. 6. Minimum arrangement of SWAP gates after applying templates according to the original FPRM for 141 polarity (its original quantum circuit is visualized at the bottom right)

Издательство «НОВЫЕ ТЕХНОЛОГИИ» выпускает научно-технические журналы



Теоретический и прикладной научно-технический журнал **ПРОГРАММНАЯ ИНЖЕНЕРИЯ**

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

Подписной индекс по Объединенному каталогу
«Пресса России» – 22765



Ежемесячный теоретический
и прикладной научно-
технический журнал

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

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

Подписной индекс по
Объединенному каталогу
«Пресса России» – 72656

Междисциплинарный
теоретический и прикладной
научно-технический журнал

НАНО- и МИКРОСИСТЕМНАЯ ТЕХНИКА

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



Подписной индекс по
Объединенному каталогу
«Пресса России» – 79493



Ежемесячный теоретический
и прикладной
научно-технический журнал

МЕХАТРОНИКА, АВТОМАТИЗАЦИЯ, УПРАВЛЕНИЕ

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

Подписной индекс по
Объединенному каталогу
«Пресса России» – 79492

Научно-практический
и учебно-методический журнал

БЕЗОПАСНОСТЬ ЖИЗНЕДЕЯТЕЛЬНОСТИ

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



Подписной индекс по
Объединенному каталогу
«Пресса России» – 79963

Адрес редакции журналов для авторов и подписчиков:

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