

Министерство образования Республики Беларусь  
Учреждение образования  
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ  
(БГУИР)

УДК681.513.6;681.513.7;007.52  
№ госрегистрации 20102753  
Инв. №

УТВЕРЖДАЮ  
Проректор по научной работе  
д-р техн. наук, проф.  
\_\_\_\_\_ А. П. Кузнецов  
«\_\_» \_\_\_\_\_ г.

ОТЧЕТ  
О НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ

МЕТОДЫ И ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА МОДЕЛИРОВАНИЯ И РАЗРАБОТКИ  
ИНФОРМАЦИОННЫХ, СОЦИАЛЬНО-ЭКОНОМИЧЕСКИХ И ТЕХНИЧЕСКИХ  
СИСТЕМ

(заключительный)

ГБ № 10-2030

Руководитель НИР  
зав. кафедрой ЭИ  
канд. техн. наук, доц.

\_\_\_\_\_ В.Н. Комличенко

Отв. исполнитель  
канд. экон. наук, доц.

\_\_\_\_\_ А.Э. Алехина

Минск 2015

## СПИСОК ИСПОЛНИТЕЛЕЙ

**Руководитель НИР**  
зав. кафедрой ЭИ  
канд. техн. наук, доц.

\_\_\_\_\_ В.Н. Комличенко  
(реферат, введение,  
раздел 1, 6, заключение)

**Отв. исполнитель**  
канд. экон. наук, доц.

\_\_\_\_\_ А.Э. Алехина  
(раздел 2)  
(

### Исполнители:

Ст. преподаватель

\_\_\_\_\_ Е.Н. Унучек  
(раздел 6)

Канд. техн. наук, доц.

\_\_\_\_\_ А.А. Бутов  
(раздел 4)

Канд. физ-мат. наук, доц.

\_\_\_\_\_ С.А. Потгосина  
(раздел 3)

Ст. преподаватель

\_\_\_\_\_ Д.А. Сторожев  
(раздел 5)

Ст. преподаватель

\_\_\_\_\_ Т.М. Унучек  
(раздел 1)

Канд. техн. наук, доц.

\_\_\_\_\_ Е.Н. Живицкая  
(раздел 6)

Ассистент

\_\_\_\_\_ В.В. Ярмольчик  
(раздел 6)

Ассистент

\_\_\_\_\_ Т.С. Космыкова  
(раздел 2)

Ассистент

\_\_\_\_\_ Т.Г. Пинчук  
(подраздел 2.3)

Ассистент

\_\_\_\_\_ П.В. Хомяков  
(подраздел 6.1)

Ст. преподаватель

\_\_\_\_\_ М.С. Рысевец  
(подраздел 3.3)

Магистрант

\_\_\_\_\_ И.В. Коваленко  
(раздел 3)

Ст. преподаватель

\_\_\_\_\_ Т.В. Русак  
(подраздел 1.4.2)

Ассистент

\_\_\_\_\_ И.Б. Валевская  
(подраздел 1.4.2)

Ассистент

\_\_\_\_\_ Е.А. Богданова  
(подраздел 5.2)

Ассистент

\_\_\_\_\_ Н.П. Мытник  
(подраздел 6.9)

**Нормоконтролёр**

\_\_\_\_\_ Л.А. Шичко

## РЕФЕРАТ

Отчет 155с, 107 рис, 17Таблиц, 51 источников.

ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ, ИНСТРУМЕНТАЛЬНЫЕ МЕТОДЫ МОДЕЛИРОВАНИЯ, УПРАВЛЕНИЯ ОСНОВНЫМИ ПРОЦЕССАМИ КАФЕДРЫ, ПРОГРАММНЫЙ КОМПЛЕКС, БИЗНЕС-ПРОЦЕССЫ, СЕТИ ПЕТРИ, ПРОЕКТИРОВАНИЕ ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ, ОЦЕНКИ РИСКА, ЭКОНОМЕТРИКА, МНОГОЗНАЧНАЯ ЛОГИКА, СБИС, ДНФ, ЭЛЕМЕНТЫ ПОКРЫТИЯ МНОГОУГОЛЬНИКА, МИНИМИЗАЦИИ ДНФ, КАЧЕСТВО ПРОГРАММНЫХ СРЕДСТВ, АВТОМАТИЗАЦИЯ ПРОЦЕССОВ УПРАВЛЕНИЯ

Объектом исследования являются инструментальные средства моделирования и разработки информационных, социально-экономических и технических систем.

Исследование проводилось с целью разработки и освоения методов и средств моделирования и программной реализации информационной поддержки социально-экономических и технических систем, а также внедрения передового опыта разработки информационных инструментальных средств в деятельность организаций, а также учебный процесс подготовки специалистов.

Методология проведения работы: анализ предметной области, формулировка гипотез, поиск и сравнительный анализ существующих аналогов в данной и смежных областях, выбор метода и проведение исследований, практический эксперимент, оценка результативности и перспективности полученных результатов, разработка выводов и предложений.

В процессе исследования использовались методы научного познания: диалектики, взаимосвязи исследовательской работы и практики, интеграции и обобщения опыта исследования и развития инструментальных средства моделирования и разработки информационных, социально-экономических и технических систем для расширения сфер практического применения и внедрения полученных знаний умения и опыта в учебный процесс.

Результатом работы явились 2 монографии, 21 учебных и учебно-методических пособий, 36 статей, 69 материалов конференций, 10 тезисов докладов на научных конференциях, около 30 переработанных и внедренных в учебный процесс электронных учебно-методических комплексов дисциплин.

Полученные результаты полностью соответствуют техническому (научному) заданию, а фактические сроки выполнения – установленным договором срокам.

## СОДЕРЖАНИЕ

Введение .....	6
1 Инструментальные методы моделирования и оценки образовательных бизнес-процессов кафедры .....	8
1.1 Описание системы управления основными процессами учебной деятельности кафедры .....	8
1.1.1 Описание системы управления основными процессами кафедры .....	8
1.1.2 Анализ и оценка ключевых бизнес-процессов кафедры .....	13
1.2 Методики проведения лабораторных работ, курсового и дипломного проектирования .....	19
1.3 Инструментальные средства программного комплекса поддержки виртуальной обучающей среды .....	32
1.3.1 Анализ подходов к моделированию бизнес-процессов виртуальной обучающей среды .....	32
1.3.2 Разработка подхода к описанию бизнес-процессов сетями Петри .....	36
1.3.3 Применение разработанного подхода для моделирования основных бизнес-процессов виртуальной обучающей среды .....	38
1.4 Руководство пользователя по эксплуатации инструментальных средств .....	41
1.4.1 Проектирование инструментального средства для моделирования бизнес-процессов деятельности кафедры .....	41
1.4.2 Руководство пользователя .....	43
2 Моделирование оценки риска банкротства предприятий .....	48
2.1 Диагностики банкротства на основе моделей бинарного выбора .....	48
2.2 Моделирование оценки риска банкротства предприятий с учетом личностно-профессиональных характеристик руководителя организации .....	50
2.3 Методика проведения лабораторных работ по дисциплине «Эконометрика» для студентов первой и второй ступени высшего образования .....	53
3 Исследование работоспособности организационных процессов в социально-экономических системах .....	75
3.1 Применение многозначной логики в исследовании работоспособности социально-экономических систем .....	75
3.2 Методика построения структурной функции и исследование ее чувствительности .....	78
3.3 Исследование работоспособности процесса организации олимпиады «Созвездие талантов» .....	86
4 Представление геометрических объектов с помощью булевых формул в задачах проектирования топологии СБИС .....	92
4.1 Введение и формулировка решаемых задач .....	92
4.2 Основные понятия и определения .....	93
4.2.1 Граница многоугольника, ориентация сторон, предикатные переменные .....	93
4.2.2 Выпуклые и вогнутые углы многоугольника .....	94
4.3 Способ поиска множества крайних вершин многоугольника .....	95
4.3.1 Каноническая булева формула многоугольника .....	95
4.3.2 Поиск множества крайних вершин многоугольника .....	97
4.4 Простой метод нахождения булевой формулы многоугольника в ДНФ .....	98
4.5 Метод нахождения булевой формулы многоугольника в ДНФ без использования дополнительных предикатных переменных .....	100
4.5.1 Элементы покрытия, порождаемые сторонами многоугольника .....	101
4.5.2 Базовый метод нахождения элементов покрытия многоугольника .....	103
4.5.3 Усовершенствованный метод нахождения булевой формулы многоугольника .....	104

4.6	Комплекс программ минимизации систем частичных булевых функций .....	106
4.6.1	Булевы функции .....	106
4.6.2	Частичные булевы функции .....	108
4.6.3	Системы булевых функций .....	108
4.6.4	Поиск кратчайшей ДНФ системы функций .....	111
4.6.5	Приближенные методы решения задачи минимизации ДНФ .....	119
4.6.6	Программная реализация методов минимизации ДНФ системы функций .....	122
5	Моделирование процессов усвоения знаний и методов извлечения знаний для управления качеством программных средств .....	126
5.1	Обобщенная характеристика проводимых исследований .....	126
5.2	Моделирование процессов усвоения знаний и методов извлечения знаний для управления качеством программных средств .....	127
6	Программная поддержка управления исполнением процессов на основе сценариев .....	130
6.1	Особенности управленческой деятельности в процессе обучения .....	130
6.2	Автоматизация процессов управления в образовательной сфере .....	131
6.3	Место сценарного подхода в управлении .....	132
6.4	Методы и способы повышения эффективности управления образовательным процессом компании .....	135
6.5	Постановка задачи и обзор методов её решения .....	138
6.6	Спецификация системы .....	139
6.7	Модели представления системы и их описание .....	141
6.8	Информационная модель системы и описание .....	143
6.9	Руководство пользователя .....	146
	Заключение .....	151
	Список использованных источников .....	153

## ВВЕДЕНИЕ

В условиях экономического обновления общества особую актуальность приобретает проблема повышения эффективности и качества деятельности человека в различных областях. Для решения задач повышения эффективности нужны подготовленные кадры. Главным источником подготовки квалифицированных кадров является высшая школа.

Профессиональное качество подготовки специалиста в образовательных учреждениях определяется не только научной базой и совокупностью получаемых в процессе обучения знаний, но и формированием заданного множества компетенций, понимаемого как подготовленность выпускника вуза к реальному владению методами, средствами деятельности, обладанию такой формы сочетания знаний, умений и навыков, которая позволяет достигать поставленной цели, способностью и готовностью выпускника эффективно применять имеющиеся у него знания и умения в конкретной ситуации для решения поставленной задачи. Такая особенность подготовки специалистов в современной высшей школе диктуется как требованием времени, так и современными стандартами - компетентностными обучающими стандартами третьего поколения.

Четкая направленность стандартов ориентирует центральные звенья обучения на развитие способностей обучаемых к практическому применению знаний и умений, т.е. компетентностный подход в обучении, и предполагает значительное усиление практической ориентации образования, что требует не только изменения структуры учебного процесса, но и специальной подготовки и переподготовки преподавателей.

Одним из реальных механизмов, способствующих реализации меняющихся требований в обучении и задачам переподготовки и «самообучения» преподавателей может служить ориентированная, разноплановая научно-исследовательская деятельность сотрудников кафедры, практической направленности, выполняемая в рамках исследований по второй половине дня.

В рамках рассматриваемой темы сотрудники кафедры ЭИ выполняли исследования в различных прикладных областях, что позволило расширить объем компетенций коллектива в целом, область возможного внедрения полученных результатов (знаний и навыков), а также количество предметных областей, рассматриваемых в учебном процессе, совершенствовать имеющуюся специальность ИСИТ (в экономике). Результатом проведенных исследований можно считать и открытие новой специальности ИСИТ (в логистике), что, как мы надеемся, внесет определенный вклад в инновационное развитие Республики Беларусь, активно осуществляющей поиск новых путей развития экономики.

Современный учебный процесс должен базироваться на деятельностном, системном подходе к усвоению знаний, использовать перспективные методы и техники передачи и усвоения знаний. В первую очередь речь идет о дисциплинах информатики и программирования, логистики, экономико-математических и инженерных дисциплин, составляющих основу формируемых специальностей. Очевидно, что в данных условиях процесс обучения должен иметь обучающе-исследовательский характер, использовать принципы проектного обучения, развивать системное, ситуационное и стратегическое мышление, навыки в области системного анализа, проектирования и программирования. Изучение теории и приобретение заданных компетенций в процессе подготовки специалистов должно осуществляться на основе современных интеграционных платформ информатики, логистических, экономических методов и технических решений, опираться на наиболее эффективные технологии, лучшие типовые и унифицированные решения, в целом, иметь практическую направленность и выходы на внедрение результатов в реальные производственные и социально-экономические процессы. Существенную роль должно играть и обратное, - полученные в проводимых НИР результаты должны внедряться в учебный процесс и повышать эффективность процессов подготовки специалистов.

В заключительном отчете представлены результаты исследований, выполненных сотрудниками кафедры экономической информатики по ГБ НИР, выполняемой по второй половине дня.

В первом разделе представлен материал, относящийся к разработке инструментальных методов моделирования и оценки образовательных бизнес-процессов кафедры. Описаны системы управления основными процессами учебной деятельности кафедры. Проведен анализ и оценка ключевых бизнес-процессов кафедры, а также анализ подходов к моделированию бизнес-процессов виртуальной обучающей среды

Во втором разделе рассматриваются перспективные методики преподавания дисциплины «Эконометрика», разработанные преподавателями кафедры экономической информатики, для студентов первой и второй ступени высшего образования. Целью данных технологий организации учебного процесса является создание условий для включения каждого обучаемого в деятельность, соответствующую зоне его ближайшего развития, а также условий для самостоятельного (и/или под контролем преподавателя) усвоения программного материала в том размере и с той глубиной, которые позволяют индивидуальные особенности обучаемого.

В третьем разделе представлен материал, иллюстрирующий целесообразность его внедрения в учебный процесс магистрантов по специальности «Математические и инструментальные методы в экономике» в рамках дисциплины «Моделирование и программная реализация сложных информационных систем» с модулем «Математическое моделирование в операционном менеджменте», где речь идет о системах с несколькими уровнями работоспособности, изменениях состояний работоспособности системы в зависимости от изменения состояний ее элементов и т.п.

В четвертом разделе приводятся результаты решения некоторых задач, связанных с подготовкой топологической информации для микрофотонаборных генераторов изображений, а именно – описаны методы представления топологических объектов (многоугольников) в виде булевых формул. Актуальность данных задач связана – ускоренным ростом развитием электроники и микроэлектроники, производством сверхбольших интегральных схем (СБИС), совмещающих тысячи и сотни тысяч электронных и оптических компонентов, формирующих суперскоростные устройства цифровой обработки информации. Все это требует ультратонких технологий создания в полупроводниковых материалах компонентов субмикронных размеров, отвечающих строгим расчетам физиков, химиков, схемотехников, образующих в итоге электронную схему. В современных условиях сложнейшие наукоемкие технологии производства микроэлектронных схем и устройств способны обеспечивать эффективность бизнеса, его конкурентоспособность и привлекательность, если эти технологии будут интегрированы в рамках так называемых CALS-технологий. Применение CALS-технологий позволяет эффективно, в едином ключе решать проблемы обеспечения качества выпускаемой продукции и ее конкурентоспособности, поскольку используемое электронное описание процессов разработки, производства, монтажа и т.д. полностью соответствует требованиям международных стандартов серии ИСО 9000, реализация которых гарантирует выпуск высококачественной продукции.

Пятый раздел посвящен интеллектуальным методам оценки качества программных средств, а также, методам моделирования процессов усвоения знаний и извлечения знаний для управления качеством программных систем.

В шестом разделе исследуются вопросы обеспечения управленческой деятельности учебного процесса на основе сценарного подхода и систем автоматизированной поддержки основных аспектов деятельности.

# 1 Инструментальные методы моделирования и оценки образовательных бизнес-процессов кафедры

## 1.1 Описание системы управления основными процессами учебной деятельности кафедры

### 1.1.1 Описание системы управления основными процессами кафедры

Кафедра Экономической информатики проводит подготовку специалистов в области информационных систем, технологий и экономики по специальности «Информационные системы и технологии (в экономике)» и «Информационные системы и технологии (в логистике)». Процесс подготовки охватывает четыре главные направления: инженерное, математическое, информационное и экономическое, – которые образуют сложную взаимосвязанную многоуровневую модель, формирующую целевую триаду знаний, умений и навыков выпускаемых специалистов. Одной из главных задач кафедры является подготовка высококвалифицированных специалистов, отвечающих требованиям современного рынка труда, в области экономики, программирования и информационных технологий. Специфика указанных областей требует от кафедры повышения эффективности ее деятельности посредством внедрения передовых методов и технологий обучения, постоянного мониторинга ее бизнес-процессов по основным направлениям деятельности [1, 2].

Можно выделить основные направления деятельности кафедры:

- обучение студентов;
- методическое обеспечение учебных процессов;
- НИРС, НИР;
- повышение квалификации;
- ведение вспомогательных процессов кафедры.

По каждому из указанных направлений можно выделить бизнес-процессы, тем или иным образом взаимодействующие между собой. Выделенные процессы представлены в таблице 1.1.

Таблица 1.1 – Модель процессов кафедры по основным направлениям ее деятельности

Направления деятельности	Процессы
Обучение студентов	Обучение по лекционным курсам
	Обучение на практических занятиях
	Обучение на лабораторных занятиях
	Выполнение и защита курсовых проектов и курсовых работ
	Прием экзаменов и зачетов
	Переаттестация неуспевающих студентов
	Обучение на ДО
	Проведение консультаций
Методическое обеспечение учебных процессов	Разработка и издание методических пособий, лабораторных практикумов, методических рекомендаций, контрольных заданий и т.п.
	Подготовка пакетов обучающих программ для ДО
	Разработка и издание электронной литературы
	Организация и ведение электронной библиотеки

Продолжение таблицы 1.1

<b>Направления деятельности</b>	<b>Процессы</b>
НИРС, Н	Проведение семинаров
	Проведение производственной и преддипломной практики
	Подготовка и участие студентов в научных конференциях
	Проведение конференций
Повышение квалификации	Участие в конференциях и публикация результатов работ сотрудников кафедры
	Подбор высококвалифицированных кадров
	Обучение в докторантуре, аспирантуре, магистратуре, соискательство
	Самообучение
	Прохождение курсов повышения квалификации
	Участие в конференциях сотрудников
	Стажировки в других вузах, кафедрах
	Ведение вспомогательных процессов кафедры
Ведение вспомогательных процессов кафедры	Издание, хранение, анализ и обработка протоколов кафедры, регламентов, постановлений
	Хранение отчетов студентов, курсовых работ, проектов и т.п.
	Ведение электронного документооборота
	Разработка базовых, рабочих, типовых программ
	Разработка учебно-методической литературы
	Техническое и программное обеспечение
	Планирование деятельности кафедры: ресурсов, основных направлений работы, стратегическое и тактическое
	Анализ учебной информации и принятие управленческих решений

Также выделенные процессы кафедры были приведены в соответствие с моделью бизнес-процессов, описанной в стандарте ISO 9001: 2000. Согласно этому стандарту выделяют следующие четыре типа процессов: управления, менеджмента ресурсов, основные (или учебно-научной деятельности), мониторинга, анализа и улучшения. (таблице. 1.2).

Таблица 1.2 – Бизнес-процессы кафедры, сгруппированные по четырем типам процессомодели стандартов ISO серии 9000

<b>Тип процесса</b>	<b>Наименование процесса</b>
Процессы управления	Планирование деятельности кафедры: ресурсов, основных направлений работы, стратегическое и тактическое
	Разработка учебных планов и программ
	Информирование работников кафедры
	Координация деятельности
	Анализ информации и принятие управленческих решений

Продолжение таблицы 1.2

Тип процесса	Наименование процесса
Процессы менеджмента ресурсов	Разработка и издание методических пособий, лабораторных практикумов, методических рекомендаций, контрольных заданий и т.п.
	Подбор высококвалифицированных кадров
	Разработка и издание электронной литературы
	Организация и поддержка кафедрального сайта
	Подготовка пакетов обучающих программ для ДО
	Издание, хранение, анализ и обработка протоколов кафедры, регламентов, постановлений
	Хранение отчетов студентов, курсовых работ, проектов и т.п.
	Ведение электронного документооборота
	Разработка базовых, рабочих, типовых программ
	Разработка учебно-методической литературы
	Техническое и программное обеспечение
	Процессы учебно-научной деятельности
Обучение на практических занятиях	
Обучение на лабораторных занятиях	
Выполнение и защита курсовых проектов и курсовых работ	
Прием экзаменов и зачетов	
Переаттестация неуспевающих студентов	
Обучение на ДО	
Проведение консультаций	
Проведение семинаров	
Проведение производственной и преддипломной практики	
Подготовка и участие студентов в научных конференциях	
Проведение конференций	
Участие в конференциях и публикация результатов работ сотрудников кафедры	
Ведение совместных проектов с другими организациями	
Процессы мониторинга, анализа и улучшения	Сбор и анализ информации об успеваемости студентов
	Сбор и анализ информации о научно-исследовательской работе кафедры
	Сбор и анализ информации кадровом потенциале кафедры
	Сбор и анализ информации о учебно-методической деятельности кафедры
	Сбор и анализ прочей информации
	Выработка корректирующих и предупреждающих мероприятий

Содержание деятельности кафедры находит свое отражение в плане работы кафедры, который составляется на каждый учебный год, и включает проведение учебной, учебно-методической, организационно-методической, научно-исследовательской работы, а также воспитательной работы со студентами и повышение профессиональной квалификации преподавателей кафедры. План работы кафедры согласовывается с проректорами по учебно-методической и научной работе и утверждается ректором БГУИР. По результатам учебного года кафедра оформляет отчет о выполнении плана работ, который заслушивается на заседании кафедры, согласовывается и утверждается проректором по учебно-методической работе.

Кафедра осуществляет следующие виды деятельности:

а) В области организационно-методической работы:

- 1) определение перечня дисциплин кафедры;
- 2) закрепление дисциплин за преподавателями кафедры;
- 3) составление кафедрального плана работы и представление его в УМУ;
- 4) рассмотрение и утверждение индивидуальных планов учебной, методической, научной и другой работы преподавателей кафедры и отчетов об их выполнении;
- 5) составление и утверждение отчетов по учебной работе (в установленные сроки);
- 6) изучение, обобщение и распространение опыта работы преподавателей;
- 7) оказание методической помощи начинающим преподавателям в овладении педагогическим мастерством;
- 8) проведение и обсуждение результатов взаимопосещений учебных занятий;
- 9) направление сотрудников кафедры на обучение на факультетах повышения квалификации (ФПК), в институтах повышения квалификации (ИПК), организация других форм повышения их квалификации и стажировки на предприятиях и в организациях;
- 10) участие в разработке кафедрального уровня системы менеджмента качества университета (описание процессов организации и ведения учебной и методической работы на кафедре, полное документирование процессов и регламентов в соответствии со стандартами серии ИСО).

б) В области учебной работы:

- 1) обеспечение высокого научного и методического уровня преподавания дисциплин путем применения традиционных и внедрения новых технологий обучения;
- 2) регулярное проведение и обсуждение на кафедре открытых учебных занятий;
- 3) использование различных форм контроля учебной деятельности студентов и анализ результатов текущего контроля и промежуточной аттестации студентов;
- 4) руководство самостоятельной работой студентов;
- 5) назначение консультаций, утверждение заведующим кафедрой графика консультаций преподавателей и размещение его на доске объявлений кафедры;

в) В области учебно-методической работы:

- 1) подготовка необходимых для учебного процесса учебно-методических материалов: рабочих программ учебных курсов, методических пособий, указаний и рекомендаций, материалов для проведения текущего контроля и промежуточной аттестации студентов, самостоятельной и индивидуальной работы студентов, по выполнению курсовых и выпускных квалификационных работ;
- 2) подготовка учебников, учебных пособий, учебно-методических комплексов, практикумов, хрестоматий и других руководств и наглядных пособий для студентов всех форм обучения;
- 3) обеспечение подготовки учебно-методических материалов для проведения обучения с элементами дистанционных образовательных технологий;

4) осуществление связи кафедры с библиотекой по вопросам книгообеспеченности дисциплин, закрепленных за кафедрой, формирование заказа на приобретение необходимых учебников и учебных пособий;

5) разработка и осуществление мероприятий по использованию в учебном процессе современных технических средств;

6) внедрение инновационных методов в образовательный процесс, т.е. методов основанных на использовании современных достижений науки и информационных технологий и направленных на повышение качества подготовки путем развития у студентов творческих способностей и самостоятельности (методы проблемного и проективного обучения, исследовательские методы, тренинговые формы, предусматривающие актуализацию творческого потенциала и самостоятельности студентов и др.);

7) организация работы научно-практических, учебно-методических семинаров с обсуждением теоретических и методических вопросов по актуальной тематике;

8) проведение и/или участие в меж кафедральных семинарах, научно-методических конференциях.

г) В области научно-исследовательской работы:

1) проведение научных исследований по теоретическим и прикладным проблемам по профилю кафедры;

2) рецензирование научных работ членов кафедры, соискателей извне;

3) обсуждение законченных диссертационных работ и принятие решения о рекомендации их к защите; рецензирование представленных к защите диссертаций, выполненных вне кафедры и присланных для подготовки отзыва;

4) руководство научно-исследовательской работой студентов;

5) составление программы кандидатских экзаменов, участие в приеме кандидатских экзаменов по специальности, ведение научной работы с аспирантами по научным специальностям, реализуемым в университете;

6) организация работы научных семинаров, участие в научно-практических и научно-методических конференциях;

7) подготовка к изданию научной и учебной литературы самостоятельно или совместно с другими подразделениями; внесение предложений по включению в План изданий актуальной научной и учебной литературы;

8) подготовка рекомендаций к переводу лиц из числа профессорско-преподавательского состава в научные сотрудники для подготовки докторской диссертации, ходатайств о предоставлении научным сотрудникам и ППС творческих отпусков;

9) участие в реализации договоров с иностранными вузами и научными центрами по вопросам, связанным с учебной и научной работой по профилю кафедры.

д) В области воспитательной работы со студентами:

1) участие в организации творческой деятельности студентов (фестивали, конкурсы, викторины, брейн-ринги, выставки студенческих работ);

2) проведение бесед, диспутов на темы нравственного поведения студентов;

3) проведение работы по формированию у студентов традиций вуза (вовлечение в общественные мероприятия, привлечение к работе в газете, музее).

е) В области довузовской подготовки:

- 1) участие выпускающих кафедр в профориентационной работе по направлениям подготовки и специальностям кафедры;
- 2) участие в проведении олимпиад и конкурсов для школьников, научных конференций и семинаров для работников образования;
- 3) участие выпускающих кафедр в проведении Дня знаний и Дней открытых дверей. Дополнительные виды деятельности выпускающих кафедр
- 4) участие в разработке документов для лицензирования вновь открываемых образовательных программ высшего профессионального образования;
- 5) руководство всеми видами практик: учебной, производственной, преддипломной; написание программ практик, организация защиты отчетов по практике;
- 6) обеспечение подготовки студентов к итоговой государственной аттестации, организация предзащиты выпускных квалификационных работ;
- 7) осуществление связи с предприятиями, организациями и учреждениями, являющимися потенциальными работодателями выпускников университета; оказание содействия в трудоустройстве выпускникам.

Определив систему управления основными процессами кафедры, перейдем к анализу и оценке ключевых бизнес-процессов кафедры.

#### **1.1.2 Анализ и оценка ключевых бизнес-процессов кафедры**

Основными направлениями деятельности кафедры экономической информатики являются научная и учебная работа. Поэтому проведем анализ этих направлений деятельности.

Кафедра обеспечивает учебный процесс на инженерно-экономическом факультете, факультете заочного обучения, факультете непрерывного и дистанционного обучения, институте информационных технологий. Ведет следующие дисциплины: «Базы данных», «Визуальные средства разработки программных приложений», «Делопроизводство», «Информатика», «Информационные технологии реинжиниринга бизнес-процессов», «Исследование операций в экономике», «Компьютерные информационные технологии», «Компьютерные сети», «Математика рынка ценных бумаг», «Объектно-ориентированное программирование», «Основы информатики и программирование», «Объектно-ориентированное проектирование и программирование», «Основы алгоритмизации и программирования», «Основы дискретной математики и теории алгоритмов», «Основы и лингвистическое обеспечение баз данных», «Основы конструирования программ», «Программирование сетевых приложений», «Прикладные системы обработки данных», «Проектирование распределенных информационных систем», «Разработка информационных систем», «Сетевые информационные технологии», «Системный анализ и проектирование информационных систем», «Современные информационные технологии», «Современные технологии обработки экономической информации», «Технологии автоматизации делопроизводства», «Эконометрика», «Эконометрика и экономико-математические методы и модели», «Экономико-математические методы и модели», «Языки программирования для разработки сетевых» [1, 2].

От количества и состава кафедры зависит успешность всех направлений деятельности кафедры. На конец учебного 2009/2010 года профессорско-преподавательский состав включал 38 сотрудников, на конец учебного 2010/2011 года – 38 сотрудников, на конец учебного 2012/2013 года – 34 сотрудника. Наблюдается небольшое сокращение количества профессорско-преподавательского состава, но

количество сотрудников остается на достаточном для успешного выполнения функций кафедры уровне.

Структура кадрового состава кафедры по годам приведена на рисунках 1.1 – 1.3. Основу кафедры составляют кандидаты наук, доценты и ассистенты. Это говорит о том, что кафедра обеспечена опытными научными сотрудниками и одновременно проводит работу по привлечению молодых специалистов.

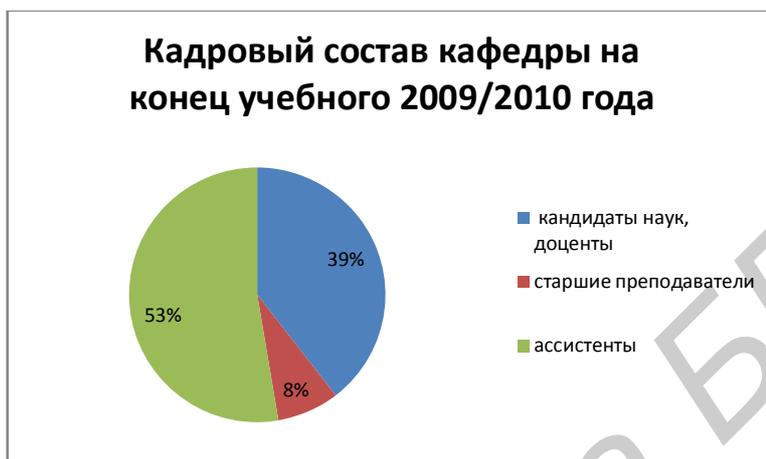


Рисунок 1.1 – Кадровый состав кафедры на конец учебного 2009/2010 года



Рисунок 1.2 – Кадровый состав кафедры на конец учебного 2010/2011 года

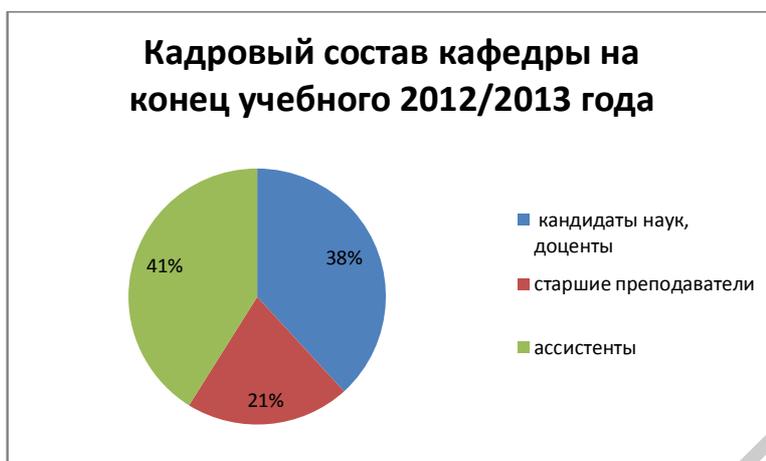


Рисунок 1.3 – Кадровый состав кафедры на конец учебного 2012/2013 года

Проводится активная работа по повышению квалификации профессорско-преподавательского состава. На рисунке 1.4 представлен график роста количества сотрудников, прошедших повышение квалификации.



Рисунок 1.4 – Количество сотрудников прошедших повышение квалификации

Проводится работа по совершенствованию материально-технической базы кафедры. Ежегодно издается по три методических пособия (диаграмма представлена на рисунке 1.5). На рисунке 1.6 отражена диаграмма, иллюстрирующая общий тираж всех выпущенных методических пособий для каждого учебного года.

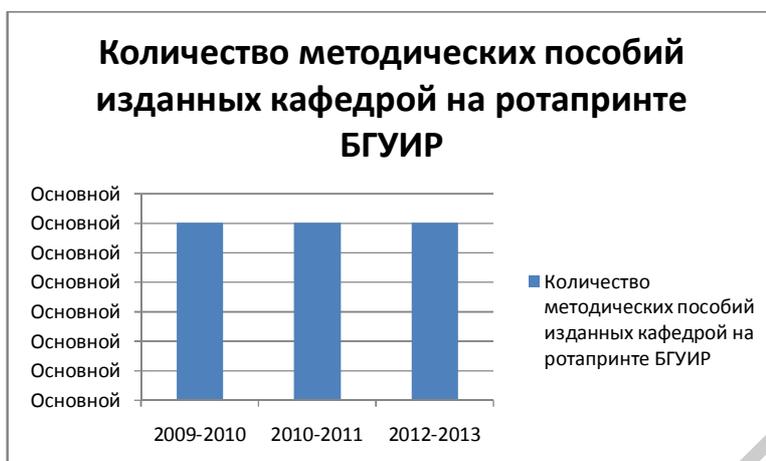


Рисунок 1.5 – Количество изданных методических пособий

Наибольший общий тираж был в 2009/2010 учебном году, наименьший – в 2010/2011. Сейчас наблюдается тенденция к увеличению общего тиража выпускаемой методической литературы.

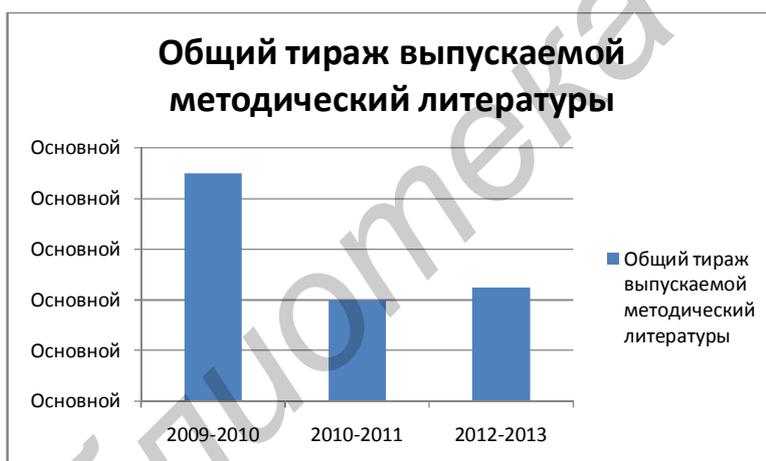


Рисунок 1.6 – Общий тираж выпускаемой методической литературы

Результат научной деятельности отражается в количестве напечатанных статей и в участии в международных, республиканских и др. конференциях, симпозиумах ППС кафедры экономической информатики. На рисунках 1.7 и 1.8 приведена динамика количества напечатанных статей и количества докладов на конференциях соответственно.

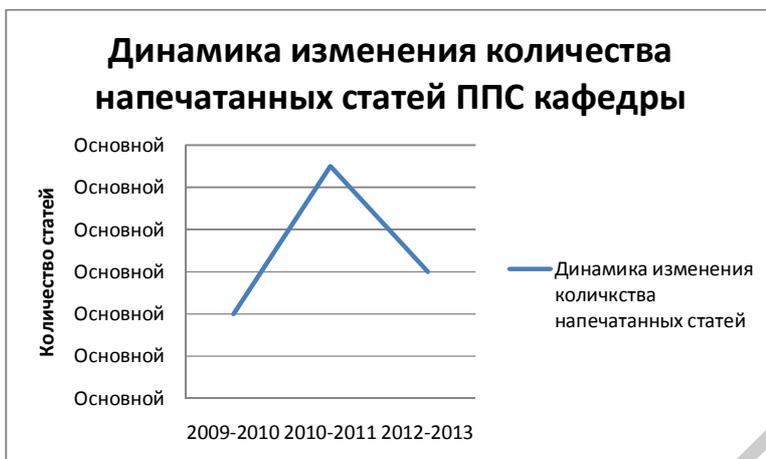


Рисунок 1.7 – Динамика изменения количества напечатанных статей

По сравнению с 2009/2010 учебным годом количество опубликованных статей увеличилось, пик пришелся на 2010/2011 учебный год, когда было напечатано 11 статей.

Также по сравнению с 2009/2010 учебным годом увеличилось количество докладов ППС на международных, республиканских и др. конференциях, симпозиумах. Пик пришелся на 2010/2011 учебный год, когда было напечатано 11 статей.

Показателем учебной работы кафедры также является количество выполняемых на кафедре дипломных проектов. Общее количество проектов уменьшилось по сравнению с 2009/2010 учебным годом. Изменилась и структура по формам обучения. Так, количество дипломных проектов заочников и дистанционщиков увеличилось, а студентов очной формы обучения – уменьшилось. Динамика изменения количества дипломных проектов по формам обучения представлена на рисунке 1.9.



Рисунок 1.8 – Динамика изменения количества докладов

На рисунке 1.10 представлена динамика изменения средней оценки по дипломным проектам с учетом формы обучения. Стабильно высокие результаты демонстрируют студенты очной формы обучения. Средний балл превышает отметку «восемь». Для заочной и дистанционной формы обучения отмечается резкий рост результата (в 2012/2013 учебном году средняя оценка превысила 8 баллов). Этот факт свидетельствует об улучшении методов обучения и контроля на заочной и дистанционной формах обучения.



Рисунок 1.9 – Динамика изменения количества дипломных проектов

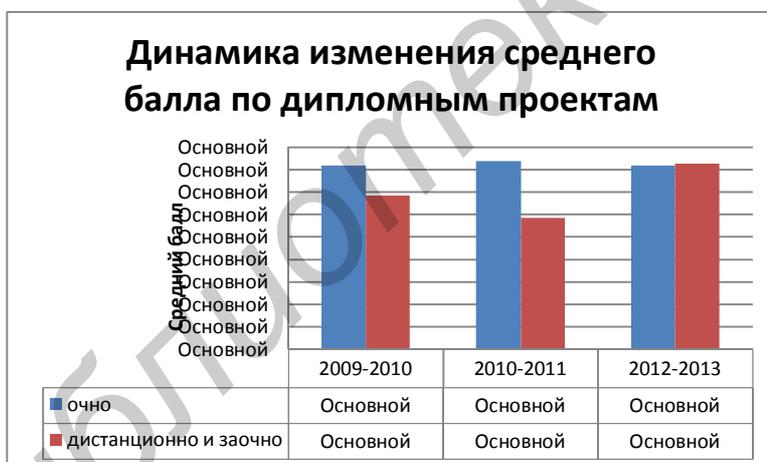


Рисунок 1.10 – Динамика изменения средней оценки

В итоге, на основании проведенного анализа можно утверждать, что кафедра экономической информатики осуществляет подготовку высококвалифицированных кадров для ИТ предприятий страны, а также ведет интенсивную научную и учебно-методическую деятельность.

## 1.2 Методики проведения лабораторных работ, курсового и дипломного проектирования

На кафедре Экономической информатики разработана методика выполнения и защиты курсовых проектов «Спиральная модель выполнения и защиты курсовых проектов студентами». Ниже перечислены основные этапы работы студента над проектом и критерии оценивания результатов его работы в соответствии с разработанной моделью:

–Студент выполняет работу над курсовым проектом в течение 12 недель со дня выдачи ему задания на курсовое проектирование.

–Задание на курсовое проектирование должно быть выдано студенту в течение 2 недель от начала семестра.

–Выполнение курсового проекта реализуется по спиральной модели проектирования: работа выполняется в несколько этапов, и каждый предыдущий этап является основой (базой, прототипом) для последующего. Спиральная модель процесса разработки курсового проекта представлена на рисунке 1.11.

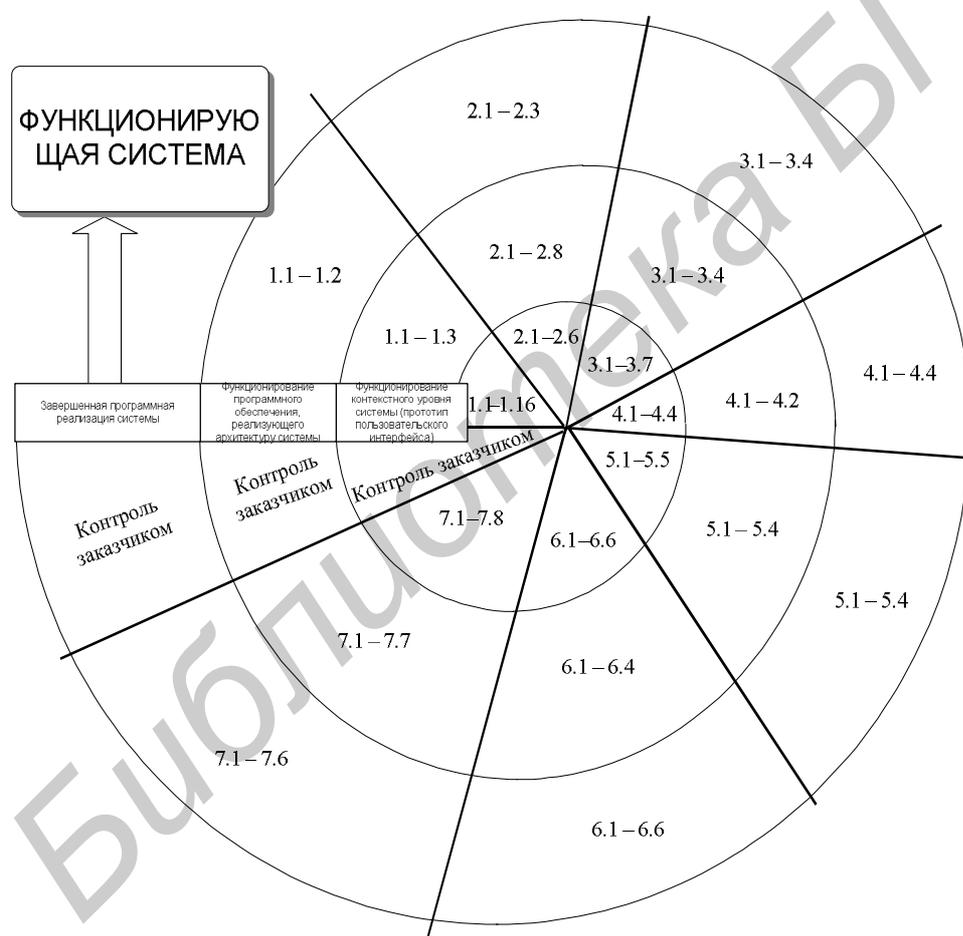


Рисунок 1.11 – Спиральная модель процесса «Выполнение и защита курсовых проектов и курсовых работ»

Выполнение курсового проекта представлено в виде семи основных процессов:

- Анализ предметной области
- Разработка методов и моделей представления системы
- Разработка информационного обеспечения системы
- Алгоритмизация системных решений
- Разработка пользовательского интерфейса программы
- Кодирование и тестирование программы
- Разработка пояснительной записки (проектной и технической документации)

Описание этих процессов представлено в разработанной на кафедре экономической информатики документации. Каждый процесс имеет свои цели выполнения, способствующие достижению общей цели выполнения курсового проекта (таблица 1.3):

Таблица 1.3 – Цели семи процессов выполнения курсового проекта

Процесс	Цель процесса
1. Анализ предметной области	Разработка концептуальной модели как абстракции предметной области.
2. Разработка методов и моделей представления системы	Формализованное описание модели предметной области с точки зрения решаемой задачи
3. Разработка информационного обеспечения системы	Информационная поддержка функционирования системы
4. Алгоритмизация системных решений	Алгоритмическое описание процессов функционирования разрабатываемой системы
5. Разработка пользовательского интерфейса программы	Обеспечение эффективного взаимодействия конечного пользователя и системы
6. Кодирование и тестирование программы	Получение надежной программной реализации системы
7. Разработка пояснительной записки (проектной и технической документации)	7.1 Эффективное использование системы 7.2 Освоение методики и стандартов разработки проектной и технической документации

Согласно методике оценки успеваемости, утвержденной ректором, установлены три этапа выполнения курсового проекта. В конце каждого этапа проводится оценка результатов его выполнения в соответствии с установленными контрольными точками. После выполнения всех этапов осуществляется защита курсового проекта (таблица 1.4).

Таблица 1.4 – Даты и тип контроля для каждого этапа

Наименование этапа	Дата	Тип контроля
этап 1 (зимний семестр)	не позднее 15 октября (для зимнего семестра)	оценка руководителем результатов выполнения этапа
этап 2 (зимний семестр)	не позднее 15 ноября (для зимнего семестра)	оценка руководителем результатов выполнения этапа
этап 3 (зимний семестр)	не позднее 15 декабря (для зимнего семестра) для IV и V курса не позднее ... декабря	оценка руководителем результатов выполнения этапа
этап 1 (весенний семестр)	не позднее 15 марта (для весеннего семестра)	оценка руководителем результатов выполнения этапа
этап 2 (весенний семестр)	не позднее 15 апреля (для весеннего семестра)	оценка руководителем результатов выполнения этапа
этап 3 (весенний семестр)	не позднее 15 мая (для весеннего семестра) для IV курса не позднее 30 апреля	оценка руководителем результатов выполнения этапа
этап 1, этап 2, этап 3	14-15 неделя от начала семестра для IV курса 14-я неделя от начала семестра	защита курсового проекта студентом

Цели, достигаемые на каждом из трех этапов, способствуют достижению целей последующих этапов и проекта в целом. Цели каждого этапа и выполнения всего курсового проекта представлены на рисунке 1.12.



Рисунок 1.12 – Цели этапов и всего курсового проекта

Выделенные этапы логически взаимосвязаны: каждый последующий этап является логическим продолжением выполнения предыдущего этапа (либо предыдущий этап предоставляет базу для выполнения последующего этапа). В своей совокупности они способствуют достижению целей выполнения всего курсового проекта.

Определенные выше семь процессов выполняются на каждом этапе. Цель (результаты) выполнения каждого из семи процессов для этапа 1 приведены на рисунке 1.13, для этапа 2 приведены на рисунке 1.14, для этапа 3 приведены на рисунке 1.15.

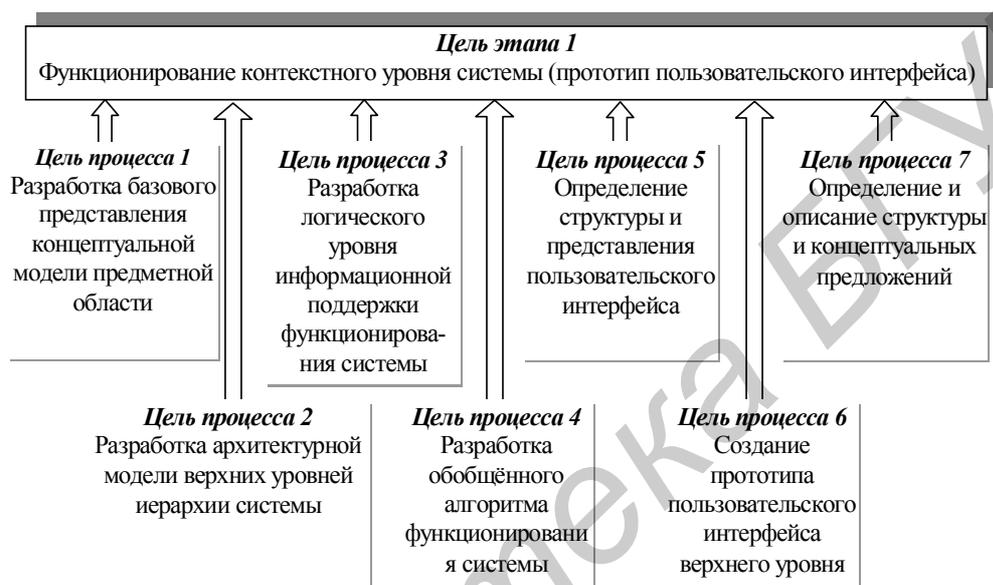


Рисунок 1.13 – Декомпозиция цели этапа 1 в виде целей каждого из семи процессов

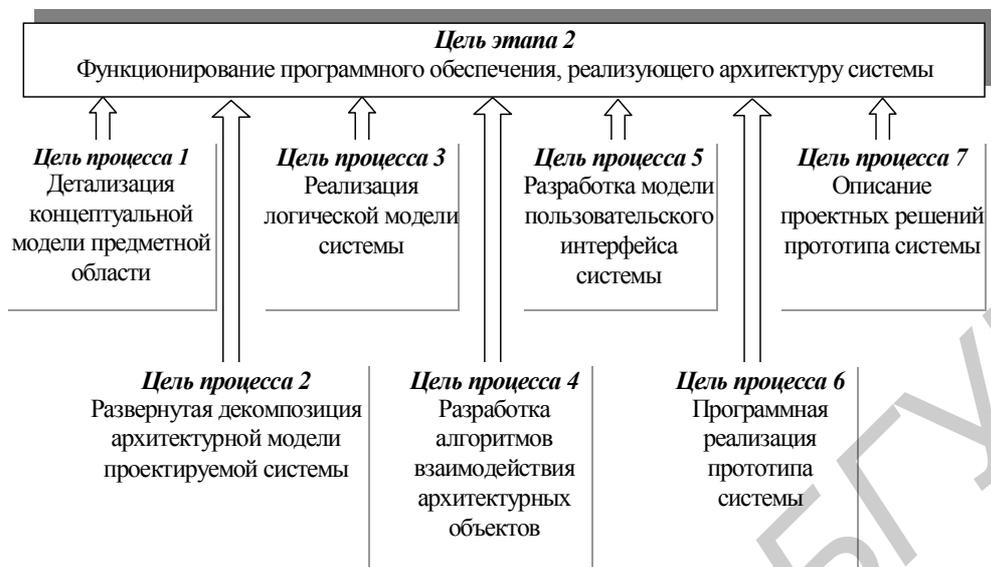


Рисунок 1.14 – Декомпозиция цели этапа 2 в виде целей каждого из семи процессов

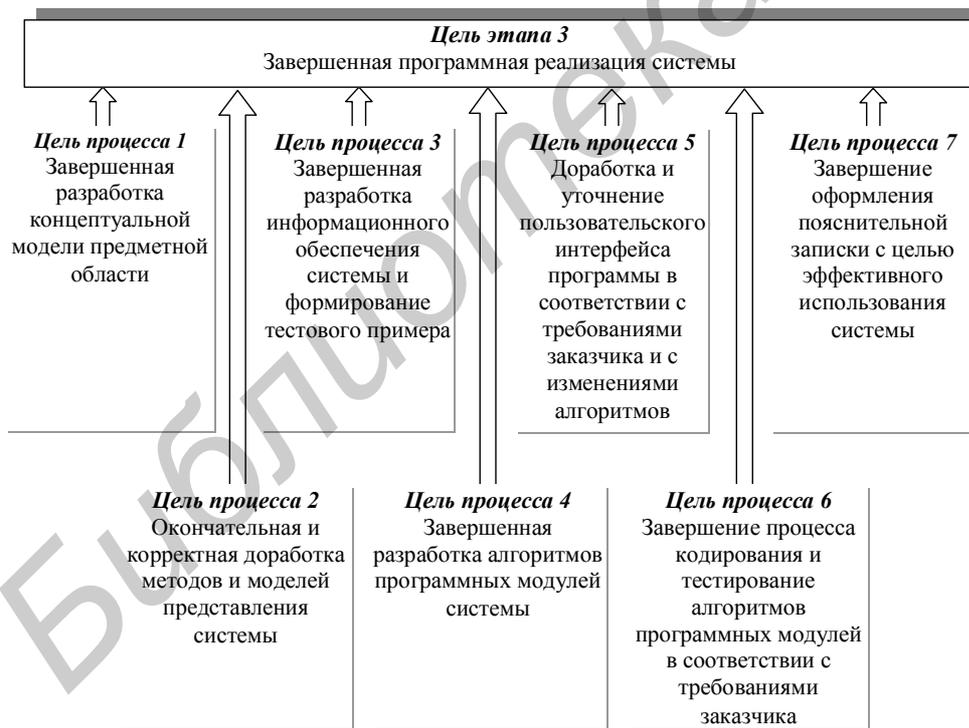


Рисунок 1.15 – Декомпозиция цели этапа 3 в виде целей каждого из семи процессов

К контролю должны быть предоставлены результаты, запланированные и утвержденные кафедрой, приведенные в таблицах 1.5, 1.6 и 1.7.

Таблица 1.5 – Задачи, реализуемые на этапе 1 «Функционирование контекстного уровня системы (прототип пользовательского интерфейса)», для достижения поставленных целей (каждого из процессов и всего этапа 1)

Процессы	Цель этапа 1	Задачи, решаемые на этапе 1
1. Анализ предметной области	Разработка базового представления концептуальной модели предметной области.	<p>1.1. Ознакомление с предметной областью, явлениями, закономерностями ее функционирования используемой терминологией.</p> <p>1.2. Осмысление цели и основных задач.</p> <p>1.3. Поиск аналогов решений и прототипов, близких к решаемым задачам.</p> <p>1.4. Определение критериев анализа.</p> <p>1.5. Оценка и выбор методов и средств анализа.</p> <p>1.6. Техническое предложение и обоснование выбора</p> <p>1.7. Определение реальных и виртуальных сущностей, их классификация и абстрагирование с позиций решаемой задачи.</p> <p>1.8. Определение жизненных циклов объектов и критические ситуации.</p> <p>1.9. Определение критических точек</p> <p>1.10. Разработка объектных абстракций</p> <p>1.11. Определение основных процессов.</p> <p>1.12. Определение основных принципов реализации.</p> <p>1.13. Выбор способа представления концептуальной модели.</p> <p>1.14. Разработка локальных представлений как элементов концептуальной модели</p> <p>1.15. Интеграция локальных представлений и построение концептуальной модели.</p> <p>1.16. Согласование с заказчиком</p>
2. Разработка методов и моделей представления системы	Разработка архитектурной модели верхних уровней иерархии системы.	<p>2.1. Анализ и систематизация требований к моделям представления системы.</p> <p>2.2. Изучение стандартов моделирования</p> <p>2.3. Выбор и обоснование методов и средств представления системы</p> <p>2.4. Моделирование контекстных диаграмм системы (0-уровней)</p> <p>2.5. Декомпозиция контекстных диаграмм</p> <p>2.6. Согласование с заказчиком</p>
3. Разработка информационного обеспечения системы	Разработка логического уровня информационной поддержки функционирования системы	<p>3.1. Анализ возможных архитектурных решений информационного обеспечения</p> <p>3.2. Анализ и классификация сущностей решаемой задачи</p> <p>3.3. Абстрагирование сущностей архитектурной модели</p> <p>3.4. Определение возможных взаимосвязей абстрагированных сущностей</p> <p>3.5. Выделение и обоснование атрибутов сущностей</p> <p>3.6. Разработка логической модели информационной поддержки функционирования системы</p>

		<b>3.7. Согласование с заказчиком</b>
--	--	---------------------------------------

Продолжение таблицы 1.5

<b>Процессы</b>	<b>Цель этапа 1</b>	<b>Задачи, решаемые на этапе 1</b>
4. Алгоритмизация системных решений	Разработка обобщённого алгоритма функционирования системы	<p>4.1. Выбор и обоснование средств и стандарта представления алгоритма функционирования системы</p> <p>4.2. Разработка обобщённого алгоритма функционирования системы</p> <p>4.3. Формализованное описание обобщённого алгоритма функционирования системы</p> <p>4.4. Согласование с заказчиком</p>
5. Разработка пользовательского интерфейса программы	Определение структуры и представления пользовательского интерфейса	<p>5.1. Анализ требований к пользовательскому интерфейсу</p> <p>5.2. Определение структуры пользовательского интерфейса</p> <p>5.3. Согласование возможностей реализации пользовательского интерфейса и функциональных требований к разрабатываемой системе</p> <p>5.4. Разработка представления структуры пользовательского интерфейса</p> <p>5.5. Согласование с заказчиком</p>
6. Кодирование и тестирование программы	Создание прототипа пользовательского интерфейса верхнего уровня	<p>6.1. Анализ требований к программной реализации</p> <p>6.2. Выбор и обоснование технологии и языка реализации алгоритмов пользовательского интерфейса</p> <p>6.3. Освоение технологии и языка реализации</p> <p>6.4. Разработка прототипа пользовательского интерфейса верхнего уровня</p> <p>6.5. Кодирование и тестирование прототипа пользовательского интерфейса верхнего уровня</p> <p>6.6. Согласование с заказчиком</p>
7. Оформление технической и проектной документации	Определение и описание структуры и концептуальных предложений.	<p>7.1. Формирование титульного листа и содержания</p> <p>7.2. Описание предметной области</p> <p>7.3. Описание требований к системе</p> <p>7.4. Описание контекстных диаграмм системы и их декомпозиций</p> <p>7.5. Описание логической модели информационной поддержки функционирования системы</p> <p>7.6. Описание обобщённого алгоритма функционирования системы</p> <p>7.7. Описание прототипа пользовательского интерфейса верхнего уровня.</p> <p>7.8. Согласование с заказчиком</p>

Таблица 1.6 – Задачи, реализуемые на этапе 2 «Функционирование программного обеспечения, реализующего архитектуру системы», для достижения поставленных целей (каждого из процессов и всего этапа 2)

Процессы	Цель этапа 2	Задачи, решаемые на этапе 2
1. Анализ предметной области	Детализация концептуальной модели предметной области.	<p><b>1.1.</b> Дополнительное исследование предметной области, явлений и закономерностей ее функционирования.</p> <p><b>1.2.</b> Детализация предметной области с целью выявления особенностей решаемой задачи.</p> <p><b>1.3.</b> Согласование с заказчиком</p>
2. Разработка методов и моделей представления системы	Развернутая декомпозиция архитектурной модели проектируемой системы.	<p><b>2.1.</b> Определение уровней архитектуры проектируемой системы</p> <p><b>2.2.</b> Декомпозиция верхнего уровня архитектуры проектируемой системы. Выделение архитектурных объектов системы.</p> <p><b>2.3.</b> Определение правил взаимодействия выделенных объектов</p> <p><b>2.4.</b> Определение жизненных циклов архитектурных объектов</p> <p><b>2.5.</b> Разработка модели взаимодействия объектов проектируемой системы</p> <p><b>2.6.</b> Разработка диаграммы классов</p> <p><b>2.7.</b> Разработка структурной схемы</p> <p><b>2.8.</b> Согласование с заказчиком</p>
3. Разработка информационного обеспечения системы	Реализация логической модели системы.	<p><b>3.1.</b> Выбор и обоснование используемой модели и средств представления данных.</p> <p><b>3.2.</b> Разработка модели физического представления информационной поддержки системы</p> <p><b>3.3.</b> Реализация разработанной модели физического представления информационной поддержки системы.</p> <p><b>3.4.</b> Согласование с заказчиком</p>
4. Алгоритмизация системных решений	Разработка алгоритмов взаимодействия архитектурных объектов.	<p><b>4.1.</b> Разработка алгоритмов взаимодействия выделенных архитектурных объектов.</p> <p><b>4.2.</b> Согласование с заказчиком</p>
5. Разработка пользовательского интерфейса программы	Разработка модели пользовательского интерфейса системы.	<p><b>5.1.</b> Разработка и согласование пользовательского интерфейса в соответствии с требованиями заказчика</p> <p><b>5.2.</b> Определение требований и решений пользовательского интерфейса с позиции реализации архитектуры системы</p> <p><b>5.3.</b> Разработка согласованной модели пользовательского интерфейса</p> <p><b>5.4.</b> Согласование с заказчиком</p>
6. Кодирование и тестирование программы	Программная реализация прототипа системы.	<p><b>6.1.</b> Кодирование и тестирование подсистемы взаимодействия архитектурных объектов.</p> <p><b>6.2.</b> Кодирование и тестирование модели пользовательского интерфейса системы.</p> <p><b>6.3.</b> Интеграция и программная реализация разработанных решений в прототип системы.</p> <p><b>6.4.</b> Согласование с заказчиком</p>

Продолжение таблицы 1.6

Процессы	Цель этапа 2	Задачи, решаемые на этапе 2
7. Оформление технической и проектной документации и	Описание проектных решений прототипа системы.	<p>7.1. Внесение дополнений и изменений в описание предметной области.</p> <p>7.2. Описание архитектурных решений проектируемой системы</p> <p>7.2.1 Описание классификации архитектурных объектов</p> <p>7.2.2 Описание модели взаимодействия выделенных архитектурных объектов</p> <p>7.2.3. Описание диаграммы классов</p> <p>7.2.4. Описание схемы взаимодействия программных модулей.</p> <p>7.3. Описание и обоснование выбора средств и методов физического представления информационной поддержки системы.</p> <p>7.4. Описание реализации модели физического представления информационной поддержки системы.</p> <p>7.5. Описание алгоритма взаимодействия выделенных архитектурных объектов</p> <p>7.6. Описание пользовательского интерфейса системы.</p> <p>7.7. Согласование с заказчиком</p>

Таблица 1.7 – Задачи, реализуемые на этапе 3 «Завершенная программная реализация системы», для достижения поставленных целей (каждого из процессов и всего этапа 3)

Процессы	Цель этапа 3	Задачи, решаемые на этапе 3
1. Анализ предметной области	Завершенная разработка концептуальной модели предметной области	<p>1.1. Законченное и документированное исследование предметной области с учетом тематики разрабатываемой системы</p> <p>1.2. Согласование с заказчиком</p>
2. Разработка методов и моделей представления системы	Окончательная и корректная доработка методов и моделей представления системы	<p>2.1. Завершенная разработка требуемых моделей</p> <p>2.2. Обоснование корректности их представления</p> <p>2.3. Согласование с заказчиком</p>
3. Разработка информационного обеспечения системы	Завершенная разработка информационного обеспечения системы и формирование тестового примера	<p>3.1. Завершенная разработка комплекса моделей информационного обеспечения разрабатываемой системы</p> <p>3.2. Обоснование корректности их представления</p> <p>3.3. Внесение тестовых данных в модель физического представления информационной поддержки системы</p> <p>3.4. Согласование с заказчиком</p>

Продолжение таблицы 1.7

Процессы	Цель этапа 3	Задачи, решаемые на этапе 3
4. Алгоритмизация системных решений	Завершенная разработка алгоритмов программных модулей системы	4.1. Разработка алгоритмов программных модулей 4.2. Описание алгоритмов программных модулей 4.3. Завершенная разработка алгоритмов с учетом замечаний и предложений 4.4. Согласование с заказчиком
5. Разработка пользовательского интерфейса программы	Доработка и уточнение пользовательского интерфейса программы в соответствии с требованиями заказчика и с изменениями алгоритмов	5.1. Внесение изменений (при необходимости) в связи с изменениями в алгоритмах программных модулей 5.2. Предоставление окончательного пользовательского интерфейса заказчику 5.3. Внесение изменений в пользовательский интерфейс с учетом всех изменений и требований заказчика 5.4. Согласование с заказчиком
6. Кодирование и тестирование программы	Завершение процесса кодирования и тестирования алгоритмов программных модулей в соответствии с требованиями заказчика	6.1. Кодирование и тестирование алгоритмов программных модулей 6.2. Тестирование программных модулей 6.3. Тестирование всей системы 6.4. Разработка тестового примера 6.5. Завершенная программная реализация и представление программного кода с тестовыми примерами 6.6. Согласование с заказчиком
7. Оформление технической и проектной документации	Завершение оформления пояснительной записки с целью эффективного использования системы	7.1. Внесение изменений в техническую и проектную документацию в связи с изменениями в процессах 1, 2, 3, 4, 5, 6 7.2. Описание наиболее важных алгоритмов программных модулей 7.3. Описание тестового примера и руководства пользователя 7.4. Оформление прочих элементов пояснительной записки по согласованию с заказчиком 7.5. Окончательное оформление пояснительной записки, подготовленной к защите 7.6. Согласование с заказчиком

В конце каждого этапа руководитель курсового проекта должен оценить предоставленные студентом результаты выполнения каждого из семи процессов. Контролю должны быть предоставлены результаты выполнения процессов, предварительно согласованные с руководителем в течение работы на каждом этапе. Для каждого из семи процессов на каждом этапе выполнения курсового проекта должен быть выполнен запланированный объем работы, установленный в процентном отношении от общего объема выполняемых работ для каждого процесса (таблица 1.8).

Таблица 1.8 – Обобщенная карта выполнения процессов

Процессы	Процентное выполнение процесса, %			
	Этап 1	Этап 2	Этап 3	Всего
1. Анализ предметной области	80	15	5	100
2. Разработка моделей представления системы	50	40	10	100
3. Разработка информационной модели системы	35	50	15	100
4. Разработка алгоритмов программных модулей	10	45	45	100
5. Разработка пользовательского интерфейса программы	50	45	5	100
6. Кодирование и тестирование программы	5	30	65	100
7. Оформление технической и проектной документации	20	40	40	100

Работа, выполненная на каждом этапе, оценивается в максимальное количество баллов по схеме, приведенной в таблице 1.9.

Таблица 1.9 – Максимальное количество баллов за % выполнение объема работ

Процессы	Максимальное количество баллов за % выполнение объема работ						Всего, баллов
	Этап 1		Этап 2		Этап 3		
	% выполнения от общего объема	баллы	% выполнения от общего объема	баллы	% выполнения от общего объема	баллы	
1	2	3	4	5	6	7	8
1. Анализ предметной области	80	<b>80</b>	15	<b>15</b>	5	<b>5</b>	<b>100</b>
2. Разработка моделей представления системы	50	<b>50</b>	40	<b>40</b>	10	<b>10</b>	<b>100</b>
3. Разработка информационной модели системы	35	<b>35</b>	50	<b>50</b>	15	<b>15</b>	<b>100</b>
4. Разработка алгоритмов программных модулей	10	<b>10</b>	45	<b>45</b>	45	<b>45</b>	<b>100</b>
5. Разработка пользовательского интерфейса программы	50	<b>50</b>	45	<b>45</b>	5	<b>5</b>	<b>100</b>
6. Кодирование и тестирование программы	5	<b>5</b>	30	<b>30</b>	65	<b>65</b>	<b>100</b>

Продолжение таблицы 1.9

1	2	3	4	5	6	7	8
7. Оформление технической и проектной документации	20	20	40	40	40	40	100
<b>Всего, баллов</b>		<b>250</b>		<b>265</b>		<b>185</b>	

За выполнение запланированного объема работ (таблица 1.9 графа «% выполнения») за каждый этап начисляется указанное количество баллов (таблица 1.9 графа «баллы»). Руководитель должен адекватно оценить работу студента, исходя из запланированного объема и фактически предоставленных ему результатов. Если студент в предыдущем этапе не выполнил установленный объем работы (не набрал допустимое количество баллов), то он обязан его выполнить на следующем этапе. Если студент предоставляет результаты, которые есть выполнение упущенного объема предыдущего этапа и выполнение объема текущего этапа, то начисляемое количество баллов состоит только из баллов, начисленных за выполненную работу текущего этапа (выставляется в соответствии с таблицей 1.9).

Общее набранное количество баллов за выполнение каждого этапа соответствует оценкам, приведенным в таблице 1.10:

Таблица 1.10 – Соответствие оценки набранному количеству баллов

Оценка	Баллы, начисленные за выполнение этапов		
	Этап 1	Этап 2	Этап 3
«отлично»	от 231 до 250	от 246 до 265	от 166 до 185
«хорошо»	от 211 до 230	от 226 до 245	от 146 до 165
«удовлетворительно»	от 191 до 210	от 206 до 225	от 126 до 145
«неудовлетворительно»	меньше 190	меньше 205	меньше 125

В случае нестандартного творческого подхода студента к решению поставленной задачи в конце каждого этапа он может получить большее количество баллов, чем то, что определено в таблице 9. Если студент проявил творческий подход к выполнению какого-то этапа проекта, то ему могут быть начислены дополнительные 10 баллов. Во время защиты курсового проекта студент может набрать максимальное количество баллов равное 60.

Если студент не может предоставить запланированный объем работ к концу этапа (контрольной точке) и причиной этого является продолжительная болезнь, занявшая более 50 % времени, отведенного на выполнение этапа, и имеется соответствующий подтверждающий документ, то может быть рассмотрен вопрос о переносе контроля результатов выполнения данного этапа на следующий этап.

Итоговая оценка выставляется студенту, исходя из общего набранного им количества баллов за выполнение трех этапов и баллов, полученных на защите, по следующей схеме (таблица 1.11):

Таблица 1.11 – Соответствие итоговой оценки студента баллам, полученным на трех этапах и на защите

Оценка	Общее количество баллов	Примечания
«отлично»	от 643 до 700	если ни на одном из этапов не получена оценка «неудовлетворительно»
«хорошо»	от 583 до 640	если ни на одном из этапов не получена оценка «неудовлетворительно»
«удовлетворительно»	от 523 до 580	допускаются оценки «неудовлетворительно», но не для всех этапов вместе
«неудовлетворительно»	меньше 522	

Если студент проявил творческий подход к выполнению курсового проекта, то, несмотря на набранное общее им количество баллов, итоговая оценка может быть скорректирована в пользу студента.

В стандарте IDEF3 разработана модель процесса «Выполнение и защита курсовых проектов и курсовых работ» рисунок 1.16.

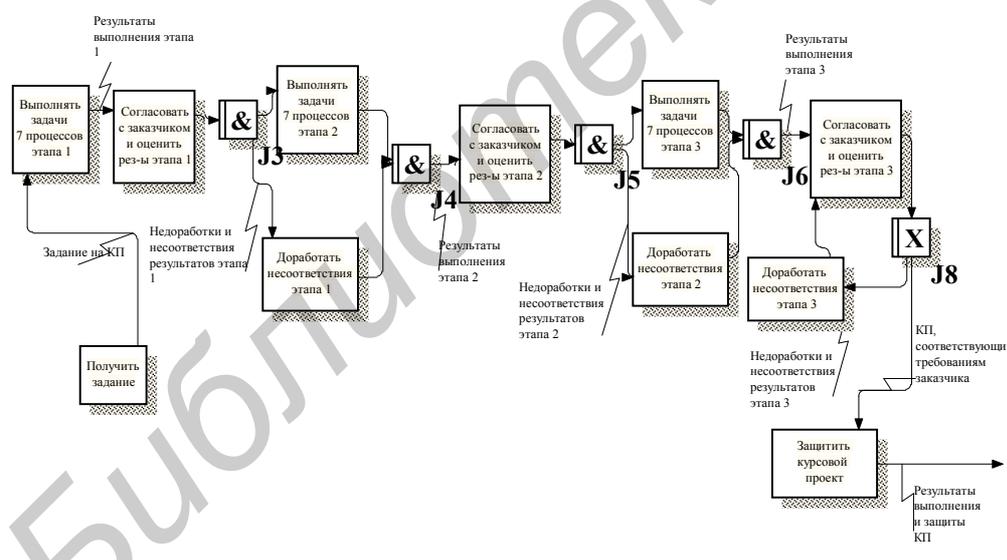


Рисунок 1.16 – Модель процесса «Выполнение и защита курсовых проектов и курсовых работ» (стандарт IDEF3)

### 1.3 Инструментальные средства программного комплекса поддержки виртуальной обучающей среды

#### 1.3.1 Анализ подходов к моделированию бизнес-процессов виртуальной обучающей среды

Проведен анализ подходов, применяемых для моделирования бизнес-процессов. Такой анализ показал, что сегодня разработан достаточно широкий спектр методологий, позволяющих проектировать бизнес-процессы. Наиболее часто используемыми среди них являются следующие (рисунок 1.17):

–BPEL (Business Process Execution Language) – язык на основе XML для формального описания бизнес-процессов и протоколов их взаимодействия между собой.

–Диаграммы BPD (Business Process Diagram), определённые спецификацией BPMN, разработанной организацией Business Process Management Initiative (BPMI) в 2001–2004 гг. с учётом ряда ранее существовавших диаграмм.

–IDEF (ICAM Definition Methods)

–ARIS – методология и одноименный программный продукт компании IDS Sheer

–Диаграммы деятельности (activity diagrams) языка UML (Unified Modeling Language – унифицированного языка моделирования).

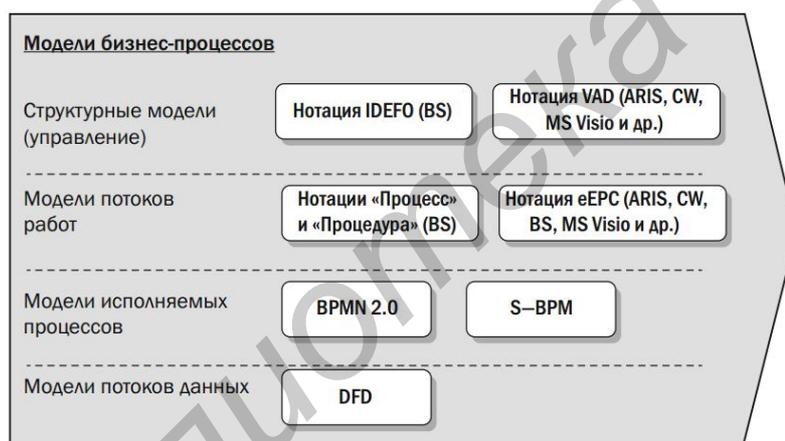


Рисунок 1.17 – Методологии описания бизнес-процессов

Наиболее распространенными среди методологий моделирования являются стандарты семейства IDEF:

–IDEF0 – методология функционального моделирования, позволяющая отобразить систему в виде набора взаимосвязанных функций (функциональных блоков). Основана на ранее разработанной Дугласом Т. Росс методике моделирования SADT.

–IDEF1 – методология моделирования информационных потоков внутри системы, позволяющая отображать и анализировать их структуру и взаимосвязи.

–IDEF1X (IDEF1 Extended) – методология построения реляционных структур по принципу “Сущность–взаимосвязь” (ER – Entity–Relationship), используется для моделирования реляционных баз данных.

–IDEF2 – методология динамического моделирования развития систем. Используется как расширение IDEF0 для описания динамических систем.

–IDEF3 – методология документирования процессов, происходящих в системе. Позволяет описывать сценарий и последовательность операций для каждого процесса. Средствами IDEF3 каждый функциональный блок IDEF0 может быть представлен в виде отдельного процесса.

–IDEF4 – методология построения объектно–ориентированных систем. Позволяет наглядно отображать структуру объектов и принципы их взаимодействия, анализировать и оптимизировать сложные объектно–ориентированные системы.

–IDEF5 – методология онтологического исследования сложных систем. Позволяет описать онтологию системы при помощи специального словаря терминов и правил, на основании которых могут быть сформированы достоверные утверждения о состоянии рассматриваемой системы в некоторый момент времени и сделаны выводы о дальнейшем развитии системы и возможности её оптимизации.

В настоящее время наиболее часто используемыми являются методологии IDEF0 и IDEF1.

IDEF0 основана на подходе SADT (Structured Analysis & Design Technique /Development Technology) и используется для создания функциональной модели системы или процесса, отображающей его структуру, функции, а также информационные и материальные потоки, преобразуемые данными функциями.

Сравнительный анализ нотаций ARIS и IDEF приводится в таблице 12.

В качестве программного обеспечения, поддерживающего стандарт IDEF0, можно использовать Design IDEF (Meta Software) или BPWin (CA).

Ценность полученного в результате моделирования описания процесса оценивают эксперты. Причем, толкование модели должно быть однозначным без личного присутствия авторов модели и без дополнительных пояснений. Окончательная модель соответствует представлениям автора и экспертов о системе, смоделированной с данной точки зрения и для данной цели [3, 4, 5, 6].

Таблица 1.12 – Сравнительный анализ нотаций ARIS и IDEF (на основе [7])

Критерии сравнения	ARIS	IDEF0	IDEF3
1	2	3	4
Принцип построения диаграммы / логика процесса	Временная последовательность выполнения процедур	Принцип доминирования	Временная последовательность выполнения процедур
Описание процедуры процесса	Объект на диаграмме	Объект на диаграмме	Объект на диаграмме
Входящий документ	Используется отдельный объект для описания («документ»)	Стрелка слева, стрелка сверху	Нет (может быть отражен в модели только привязкой объекта–комментария)

Продолжение таблицы 1.12

1	2	3	4
Входящая информация	Используется отдельный объект для описания («кластер», «технический термин»)	Стрелка слева, стрелка сверху	Нет (может быть отражен в модели только привязкой объекта–комментария)
Исходящий документ	Используется отдельный объект для описания («документ»)	Стрелка справа	Нет (может быть отражен в модели только привязкой объекта–комментария)
Исходящая информация	Используется отдельный объект для описания («кластер», «технический термин»)	Стрелка справа	Нет (может быть отражен в модели только привязкой объекта–комментария)
Исполнитель процедуры	Используется отдельный объект для описания («позиция», «организационная единица»)	Стрелка снизу	Нет (может быть отражен в модели только привязкой объекта–комментария)
Используемое оборудование	Используется отдельный объект для описания	Стрелка снизу	Нет (может быть отражен в модели только привязкой объекта–комментария)
Управление процедурой	Нет. Может быть отражено только символами логики и событий (последовательность выполнения процедур) и/или указанием входящих документов	Стрелка сверху	Только временная последовательность выполнения процедур и логика процесса
Контроль выполнения процедуры	Нет. Может быть отражен указанием входящих документов	Стрелка сверху	Нет
Обратная связь по управлению /контролю	Нет. Может быть отражена только символами логики (последовательность выполнения процедур)	Стрелка сверху	Нет

Следует отметить, что приведенные подходы к моделированию не учитывают тот факт, что бизнес–процессы по своей природе являются динамичными (со временем меняются их свойства). Поэтому для моделирования и анализа бизнес–процессов целесообразно использовать динамические модели, позволяющие описывать особенности бизнес–процессов в различные временные моменты.

Анализ показал, что для представления динамической модели бизнес–процессов могут использоваться сети Петри [8]. С их помощью можно осуществлять проверку работоспособности алгоритмов и программ. Сеть Петри отражает логическую последовательность событий, позволяет проследить потоки информации, отрабатывает взаимодействие процессов. В связи с этим сеть Петри представляется достаточно удобным аппаратом для моделирования множества взаимосвязанных, параллельных и асинхронных процессов бизнес–системы. Использование сетей Петри также привлекательно наглядностью, адекватностью и технологичностью при реализации моделей на ПК.

Выделяют следующие типы сетей Петри: автоматные сети Петри (state machine), маркированные сети (MG–сети или market graph), сети свободного выбора (FC–сети или free choice), простые сети (SN–сети или Simple nets), ординарные сети (ON–сети или Ordinary nets), раскрашенные сети (CPN или Colored Petri nets), временные сети Петри (Time Petri nets), потоковые сети (FN–сети или Flow nets), в последнее время получили развитие вложенные сети Петри (NPN–сети или Nested Petri nets) [8].

Сеть Петри представляется графом, узлами которого служат позиции (обозначаемые кружками) и переходы (обозначаемые черточками), связанные направленными дугами [8]. Сеть Петри формально задается пятеркой [8]:

$$N = (P, T, F, H, M_0), \quad (1.1)$$

где  $P$  – непустое конечное множество позиций (мест),

$T$  – непустое конечное множество переходов,

$F$  и  $H$  – матрицы инцидентности, причем.

$F : P \times T$  – отображение множества  $P$  на  $T$ ,

$H : T \times P$  – отображение множества  $T$  на  $P$ ,

$M_0$  – начальная разметка (маркировка) – вектор, который содержит информацию о количестве меток (фишек, маркеров) в соответствующих позициях (местах) сети.

Существуют три формы математического представления сетей Петри: графовая, матричная и подстановочная. Для удобства отображения в дальнейшем будем использовать матричную форму представления сетей Петри.

Согласно матричной форме функция  $F$  задается матрицей  $F_{m \times n}$  размерностью  $m \times n$ , где  $m$  – количество позиций в сети,  $n$  – количество переходов. Элемент матрицы  $F_{m \times n}$  равен “1”, если есть дуга, направленная из позиции  $P_i$  в переход  $t_j$ , и “0” – в противном случае ( $i=1 \dots m$  и  $j=1 \dots n$ ). Функция  $H$  задается матрицей  $H_{n \times m}$  размерностью  $n \times m$ . Элемент матрицы  $H_{n \times m}$  равен “1”, если есть дуга, направленная из перехода  $t_i$  в позицию  $P_j$ , и “0” – в противном случае ( $i=1 \dots n$  и  $j=1 \dots m$ ).

Смена маркировок сети или функционирование сети Петри, может быть описано с помощью алгебраического подхода к описанию работы сетей Петри, то есть следующим уравнением:

$$M_k = M_{k-1} + A^T \cdot U_{k-1}, \quad A = H - F^T, \quad (1.2)$$

где  $k=1,2,\dots$

$U_k$  – управляющий вектор в  $k$ -ый момент функционирования сети, размерность которого совпадает с количеством переходов в сети.

$$U_0 = \begin{pmatrix} \tau_1 \\ \vdots \\ \tau_n \end{pmatrix}, \quad \text{причем } \tau_i=1$$

Управляющий вектор в начальном состоянии имеет вид  $\tau_i$  означает готовность перехода  $t_i$  к срабатыванию,  $\tau_i=0$  – отсутствие готовности. Наличие нескольких единиц в векторе соответствует одновременному возбуждению соответствующих переходов и наличию разветвления процесса.

Однако, следует отметить, что описание бизнес-процессов сетями Петри имеет ряд недостатков. В частности, не понятно, каким образом можно сети Петри применить к процессам, представленным в перечисленных ранее нотациях (ARIS, BPEL, BPD, диаграммы деятельности, семейство IDEF). В связи с чем, разработан подход к описанию бизнес-процессов, представленных в нотации IDEF0, сетями Петри (основные результаты показаны в [9], [10], [11], [12]).

### 1.3.2 Разработка подхода к описанию бизнес-процессов сетями Петри

Опишем особенности представления бизнес-процессов с помощью сетей Петри. В дальнейшем будем использовать неординарные сети Петри. Ординарная сеть – это сеть, которая не имеет ограничений, кроме одного – кратность дуг должна быть не более единицы. Неординарная сеть – это сеть, у которой сняты все ограничения, в том числе и кратности единице дуг.

Если система бизнес-процессов представляет собой совокупность бизнес-процессов, в свою очередь, бизнес-процесс – совокупность функций, преобразующих его входные ресурсы в выходные, то целесообразно ввести определения с позиции сетей Петри функций, входов, выходов и того, кто выполняет процесс или функцию и что этим всем управляет.

С позиции IDEF0 функция (function, F) – совокупность действий (операций), направленных на преобразование входных ресурсов в выходные для получения определенного результата. Функции реализуются в соответствии с заданными целями и направлены на удовлетворение требований потребителя: смежных функций, процессов или потребителей вне системы бизнес-процессов. В дальнейшем будем предполагать, что функции являются логически неделимой единицей процесса.

В функционировании бизнес-процессов и их функций участвуют следующие ресурсы (рис. 1а): управление (Control resources, C)– ресурсы, регламентирующие порядок выполнения процесса; входы (Input resources, I) – ресурсы, подаваемые на вход

процесса или функции, подвергаемые преобразованию в ходе выполнения процесса или функции; выходы (Output resources,  $O$ ) – ресурсы, получаемые в результате выполнения процесса или функции; механизмы (Machine resources,  $M$ ) – ресурсы, непосредственно выполняющие процесс или функцию, сами при этом не подвергается видоизменению или преобразованию (по сравнению с входными ресурсами).

На рисунке 1.18 показан пример отображения функции процесса, описанной в стандарте IDEF0 (а) и с помощью сети Петри (б), а также сеть Петри после срабатывания перехода  $F_s$  (в).

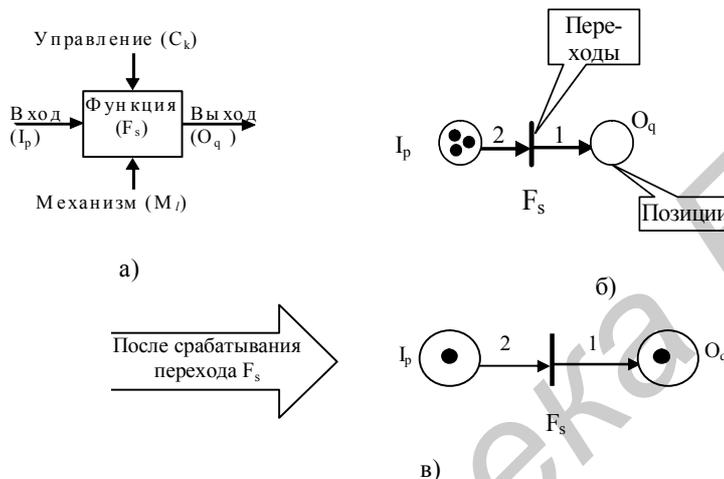


Рисунок 1.18 – Представление функции процесса в стандарте IDEF0 (а) и с помощью сети Петри (б) (механизмы и управления не показаны); сеть Петри после срабатывания перехода  $F_s$  (в)

Функционирование и поведение бизнес-процессов представим с помощью алгебраического подхода к описанию работы сетей Петри:

$$M_k = M_{k-1} + A^T * U_{k-1}, \quad (1.3)$$

$$A = T - R^T, \quad (1.4)$$

где  $U_k$  – управляющий вектор на  $k$ -ом шаге функционирования сети Петри (или бизнес-процесса), размерность которого совпадает с количеством переходов (функций) в сети Петри –  $M$ ;

$R_{N \times M} = \{r_{ij}\}$  – матрица, описывающая связи вершин  $\{I_p\}$  и  $\{F_s\}$ , где значение каждого элемента матрицы соответствует весу дуги, исходящей от вершины (позиции), обозначающей входы процессов, к вершинам-функциям;

$T_{M \times N} = \{t_{ji}\}$  – матрица, описывающая связи вершин  $\{F_s\}$  и  $\{O_q\}$ , где значение каждого элемента матрицы соответствует весу дуги, исходящей из вершины-функции к вершине, обозначающей выходы процесса;

$M_k$  – вектор текущего состояния бизнес–процесса, размерности  $N = P + Q$ , каждый элемент которого  $(m_{\xi})_k$  (будем обозначать  $m_{\xi \rightarrow k}$ ), описывает количество маркеров помещенных в позиции множеств  $\{I_p\}$  и  $\{O_q\}$  на шаге  $k$ ,  $k$  – шаг функционирования бизнес–процесса.

Управляющий вектор для сети Петри в начальный момент времени имеет вид  $U_0 = (u_1, u_2, \dots, u_M)^T$ , причем  $u_s = 1$  означает готовность функции  $F_s$  к срабатыванию,  $F_s = 0$  – отсутствие готовности. Наличие нескольких единиц в векторе  $U_k$  соответствует одновременному срабатыванию соответствующих переходов на шаге  $k$  и наличию разветвления работы сети Петри.

Управляющий вектор  $U_k$  формируется на основании следующего правила: определяется, обеспечена ли ресурсами функция  $F_s$  (переход). Для определения обеспеченности необходимо элементы текущего вектора  $M_k$  на шаге  $k$  сравнить с соответствующими элементами столбца матрицы  $R$ , номер которого совпадает с номером рассматриваемого перехода  $s$ . Если все элементы вектора текущего состояния не меньше соответствующих элементов  $s$ -го столбца матрицы  $R$ , то это означает возможность срабатывания  $s$ -ой функции, и, следовательно,  $u_s = 1$ . В противном случае  $u_s = 0$ . Это правило можно представить следующим выражением:

$$u_{s \rightarrow k} = \begin{cases} 1, & \text{если } r_{is} \geq m_{i \rightarrow k} \\ 0, & \text{если } r_{is} < m_{i \rightarrow k} \end{cases} \text{ для перехода } s, \text{ на шаге } k, \quad (1.5)$$

Таким способом с помощью сетей Петри можно моделировать процессы, в том числе параллельные, синхронные и асинхронные, как это показано в [11].

### 1.3.3 Применение разработанного подхода для моделирования основных бизнес-процессов виртуальной обучающей среды

Смоделируем процесс «Провести лабораторную работу» в нотациях IDEF0 и с помощью сетей Петри.

Упрощенно бизнес–процесс «Провести лабораторную работу у студентов» описывается в IDEF0 так, как показано на рисунке 1.19. Сеть Петри, описывающая данный процесс, представлена на рисунке 20.

Позиции: р1 – преподаватель, р2 – офисная мебель, р3 – компьютер, р4 – электроэнергия, р5 – результаты проведения лабораторных работ.

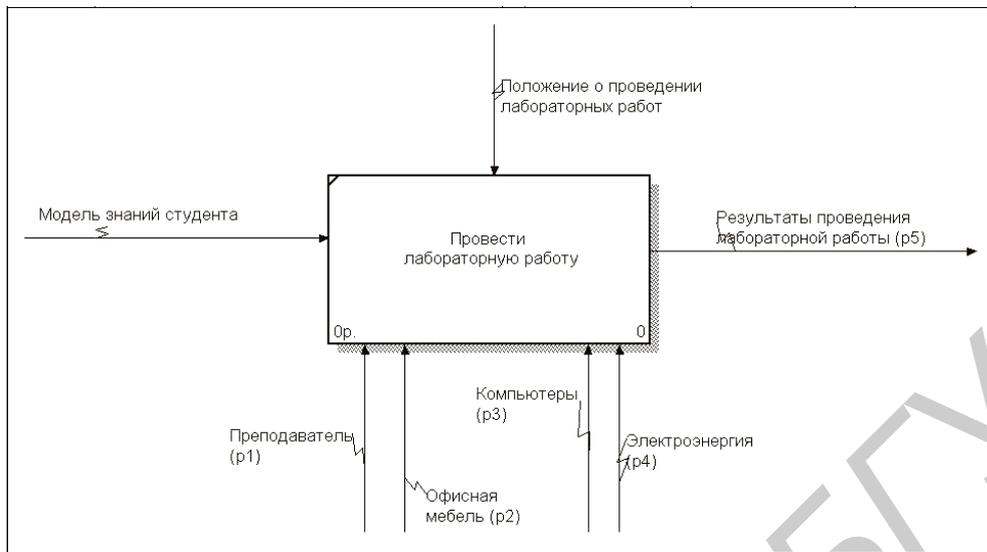


Рисунок 1.19 – Модель бизнес-процесса «Провести лабораторную работу у студентов» в нотации IDEF0

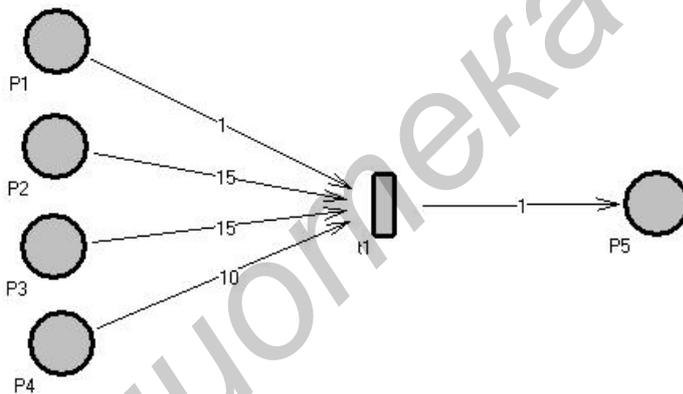


Рисунок 1.20 – Сеть Петри для бизнес-процесса «Провести лабораторную работу у студентов»

Смоделируем несколько процессов. Пусть итоговый процесс отображает проведение двух лабораторных работ и двух лекций. Описание процесса в нотации IDEF0 представлено на рисунке 1.21, с помощью сети Петри – на рисунке 1.22.

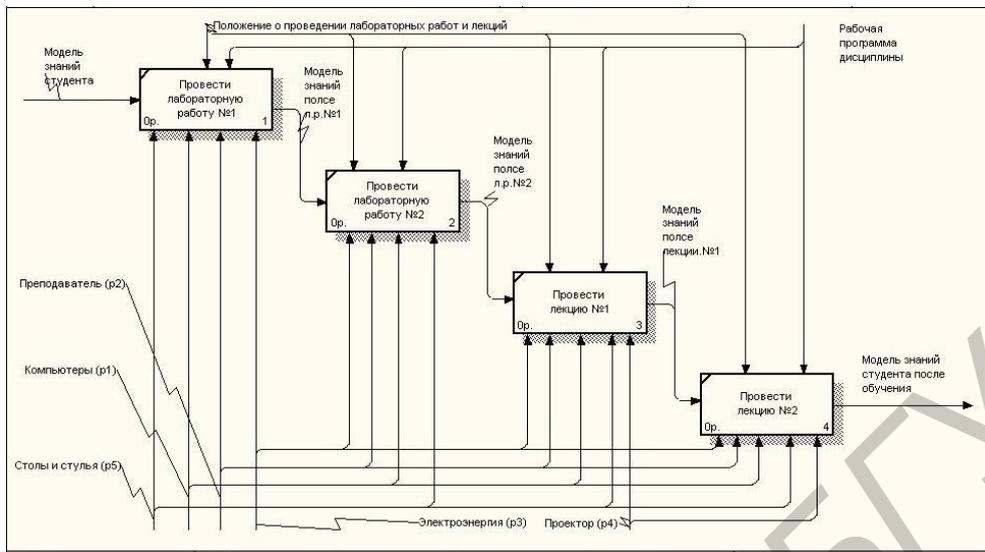


Рисунок 1.21 – Описание бизнес-процесса в нотации IDEF0

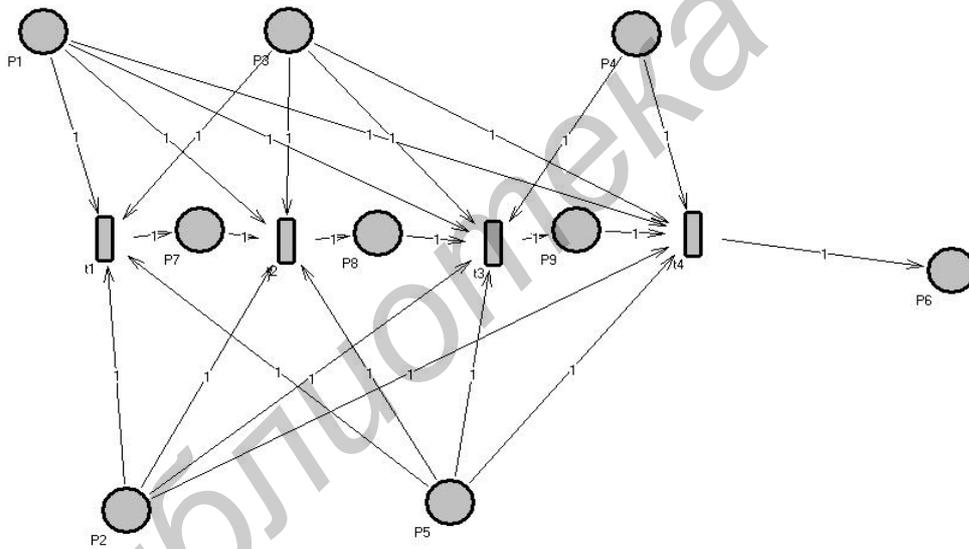


Рисунок 1.22 – Сеть Петри для процесса проведения двух лабораторных работ и двух лекций

## 1.4 Руководство пользователя по эксплуатации инструментальных средств

### 1.4.1 Проектирование инструментального средства для моделирования бизнес-процессов деятельности кафедры

Разработано инструментальное средство для моделирования бизнес-процессов деятельности кафедры, смоделированных с учетом разработанного подхода (IDEF0 и теории сетей Петри). При проектировании инструментального средства были составлены следующие диаграммы UML: диаграмма вариантов использования (рисунок 1.23), диаграмма состояний (рисунок 1.24), диаграмма последовательностей (рисунок 1.25), диаграмма компонентов (рисунок 1.26), диаграмма развертывания (рисунок 1.27), диаграмма классов (рисунок 1.28).

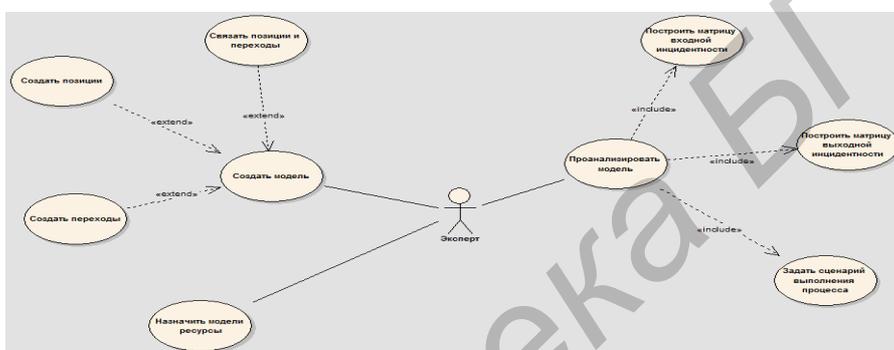


Рисунок 1.23 – Диаграмма вариантов использования

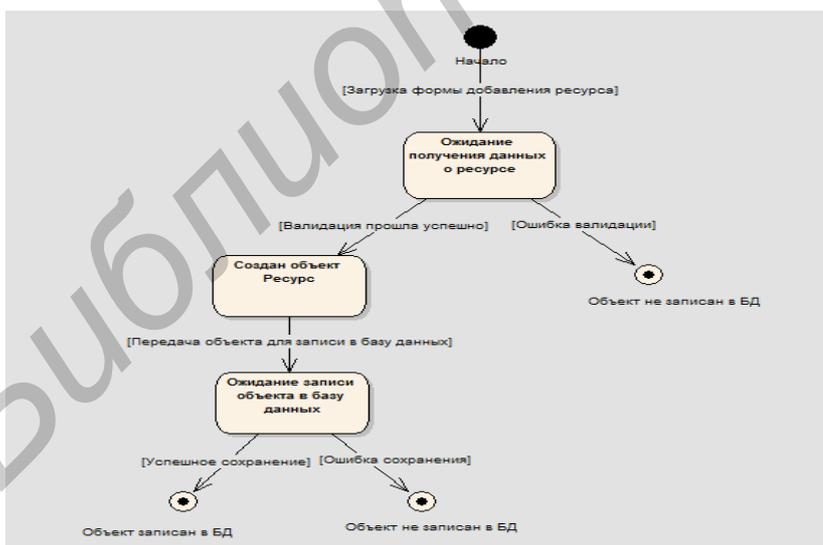


Рисунок 1.24 – Диаграмма состояний

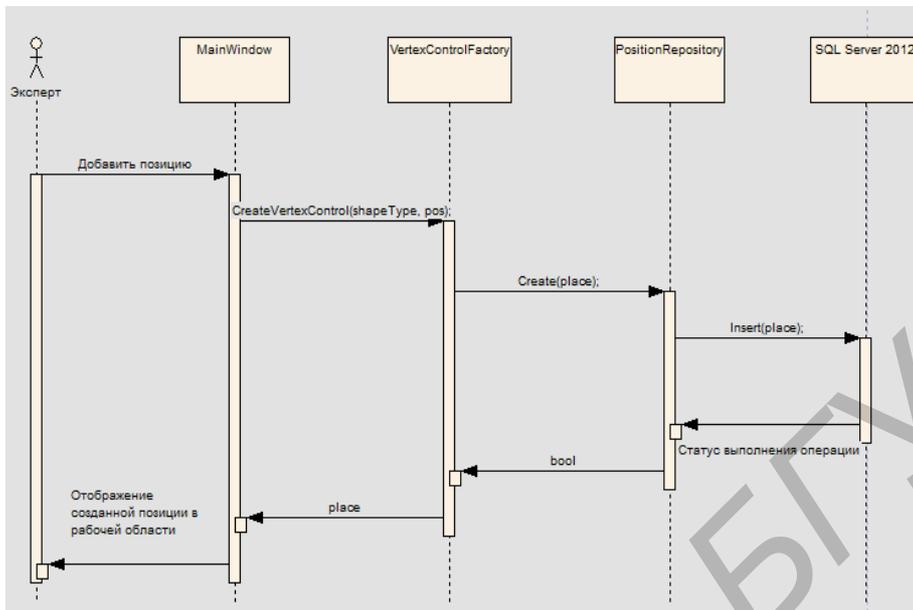


Рисунок 1.25 – Диаграмма последовательностей

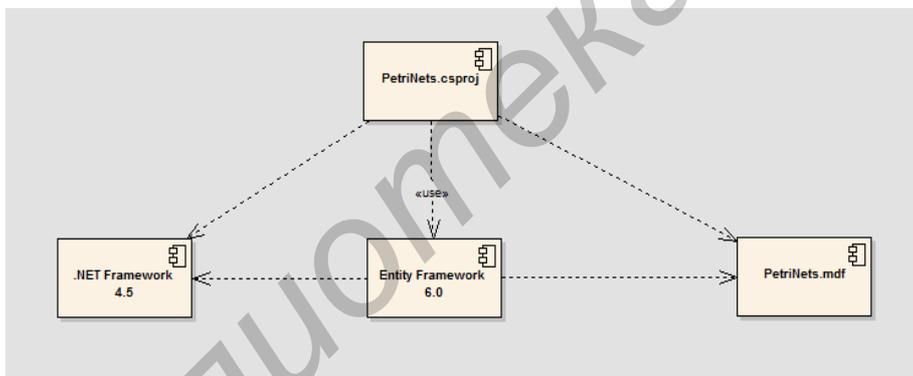


Рисунок 1.26 – Диаграмма компонентов

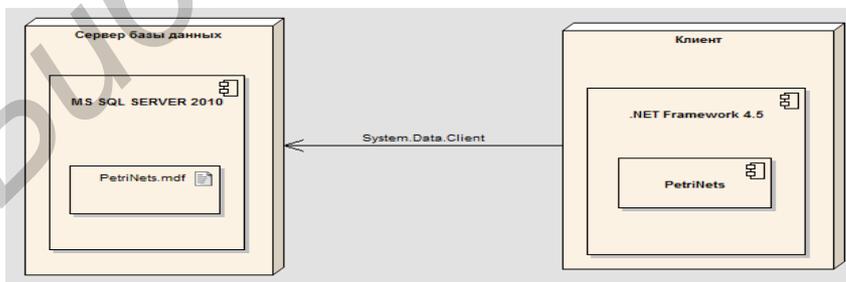


Рисунок 1.27 – Диаграмма развертывания

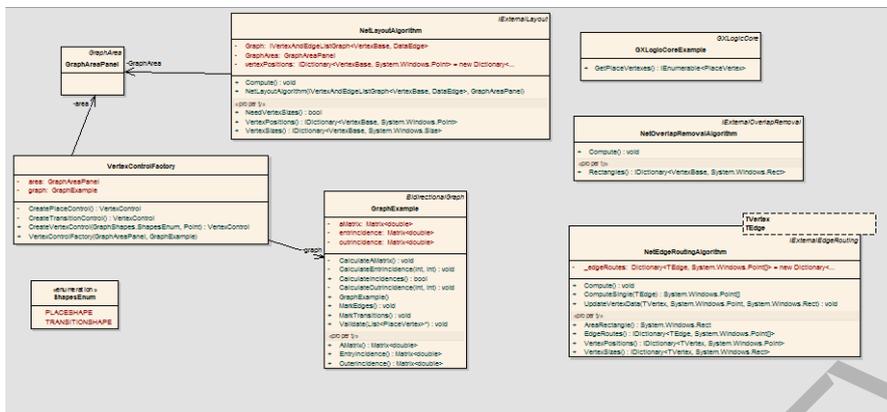


Рисунок 1.28 – Диаграмма классов

Разработана информационная модель системы на логическом и физическом уровне (рисунок 1.29).

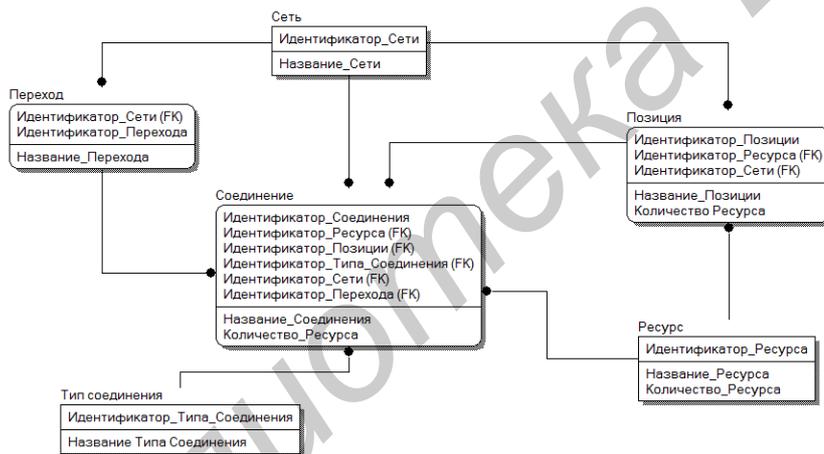


Рисунок 1.29 – Логическая модель базы данных

В качестве сервера баз данных выбран **SQLServer 2012**.

### 1.4.2 Руководство пользователя

Для корректной работы программы необходимо запустить **SQLServer 2012** с развернутой в нем базой данных **PetriNets**. При запуске программы перед пользователем появляется окно, изображенное на рисунке 1.30:

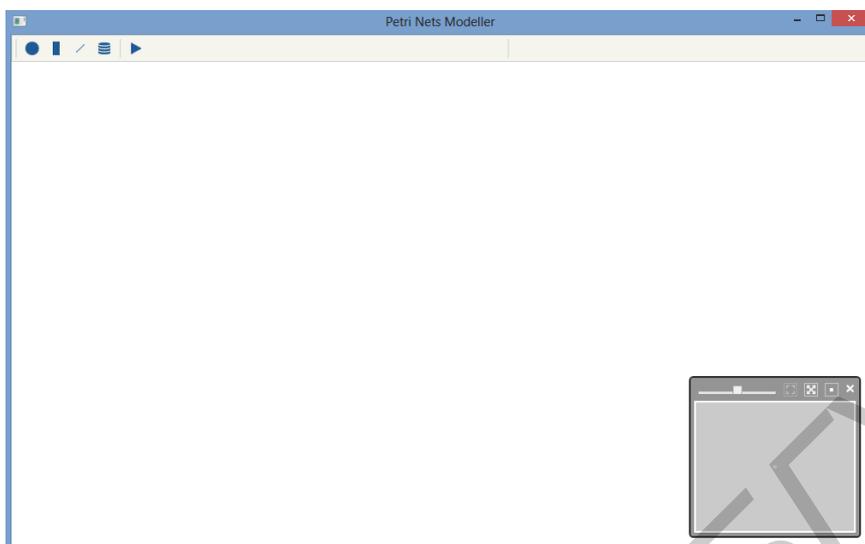


Рисунок 1.30 – Главное окно программы

В верхнем левом углу находится панель инструментов, позволяющая пользователю создавать вершины графа, управлять ресурсами и задавать сценарии выполнения процесса. В правом нижнем углу находится элемент, позволяющий изменять размеры и масштаб рабочей области.

Создание визуальных представлений позиций и переходов реализовано по принципу drag & drop. Для этого необходимо левой кнопкой мыши нажать на интересующий элемент и, не отпуская левую кнопку мыши, перетащить его на рабочую область. В результате в рабочей области появится фигура, изображающая позицию (круг) либо переход (прямоугольник).

Рабочая область с созданными вершинами представлена на рисунке 1.31.

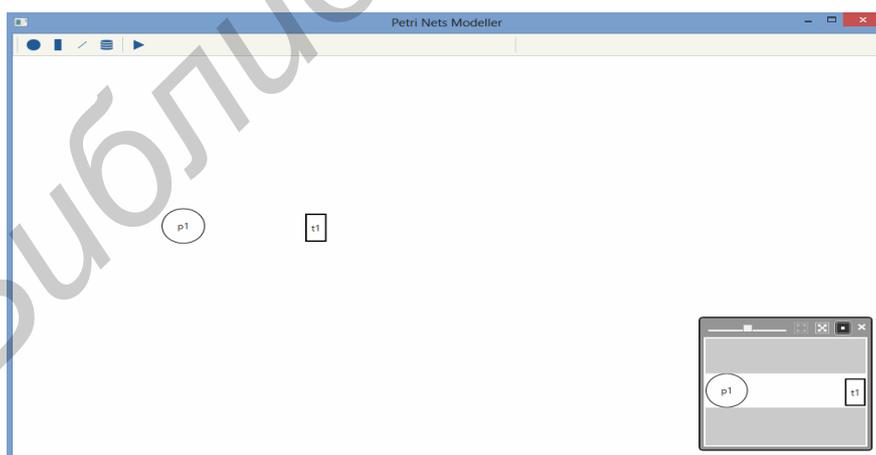


Рисунок 1.31 – Рабочая область с созданными вершинами

Для соединения вершин между собой необходимо нажать на кнопку, на которой изображена линия, затем выбрать начальную и конечную вершины. Необходимо заметить, что аппарат сетей Петри разрешает соединение вершин разных типов, т.е. нельзя соединить позицию с позицией или переход с переходом, поэтому в программе допускаются соединения вершин разных типов. Результат соединения вершин показан на рисунке 1.32.

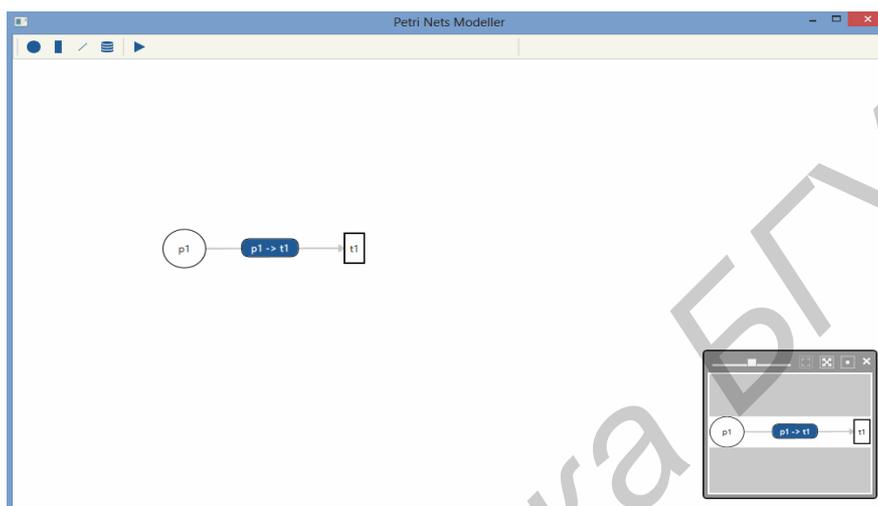


Рисунок 1.32 – Соединение вершин разных типов

Для добавления ресурсов к модели необходимо нажать на соответствующую кнопку в панели инструментов. После клика появится форма, позволяющая добавлять ресурсы к модели. Для добавления ресурса достаточно внести необходимые данные в форму и закрыть её, все данные сохранятся автоматически. Окно с формой изображено на рисунке 1.33.

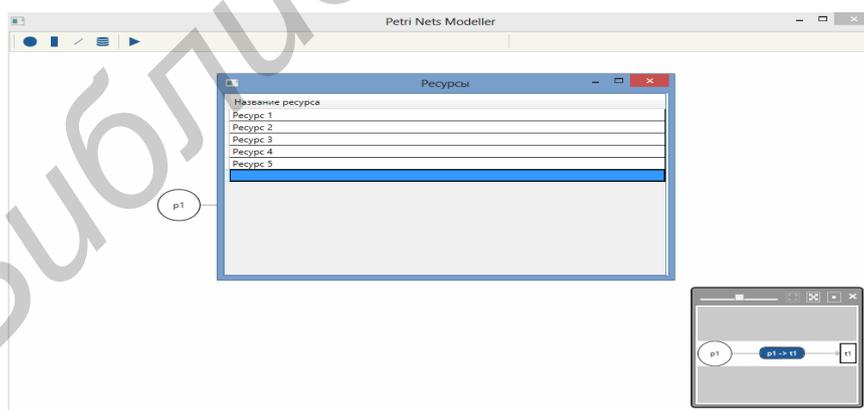


Рисунок 1.33 – Окно добавления ресурсов к модели

Для назначения ресурсов позиции необходимо два раза кликнуть на её соответствующее визуальное представление. После двойного нажатия появится форма, позволяющая изменять название позиции, а также добавлять ресурс в позицию. Добавление ресурса для позиции реализовано по принципу drag&drop. Для этого необходимо выбрать нужную строку в верхней таблице и перетянуть в нижнюю, затем задать необходимое количество ресурса. Также в форме можно изменить название позиции.

Форма редактирования параметров позиции изображена на рисунке 1.34.

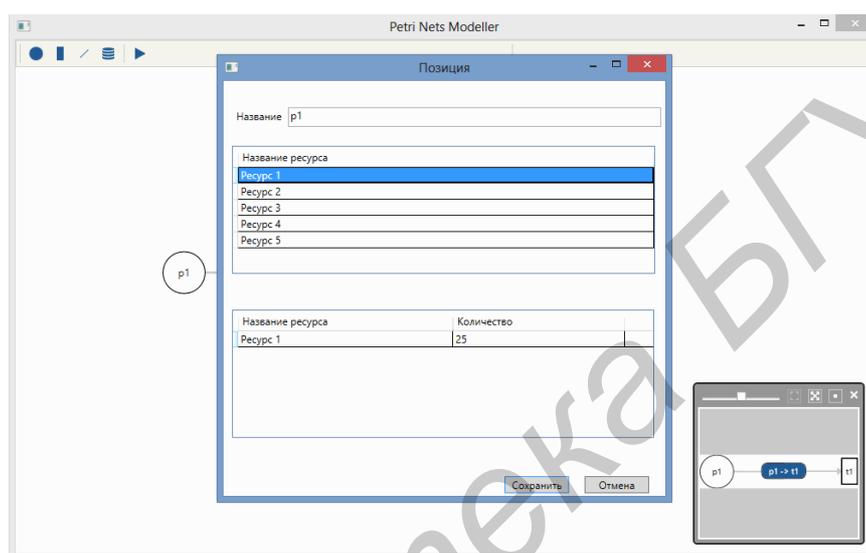


Рисунок 1.34 – Форма изменения параметров позиции

Для задания сценариев выполнения бизнес-процесса необходимо нажать на соответствующую кнопку в панели управления. После клика перед пользователем появится окно, предоставляющее возможность задать сценарии выполнения процесса. Если в позициях недостаточно ресурсов для срабатывания связанного с ними перехода, то чекбокс будет неактивен. При каждом последующем задании сценария выполнения процесса активность чекбоксов будет изменяться в зависимости от доступности ресурсов. Данный механизм позволяет избежать ошибок при имитации срабатывания переходов.

При нажатии на кнопку сохранения результатов в окне симуляции все рассчитанные данные в матричном виде сохраняются в файл.

После окончания симуляции данные о количестве ресурсов в позициях, а также о количестве потребляемых ресурсов принимают начальные значения, заданные пользователем.

Окно для задания сценариев изображено на рисунке 1.35.

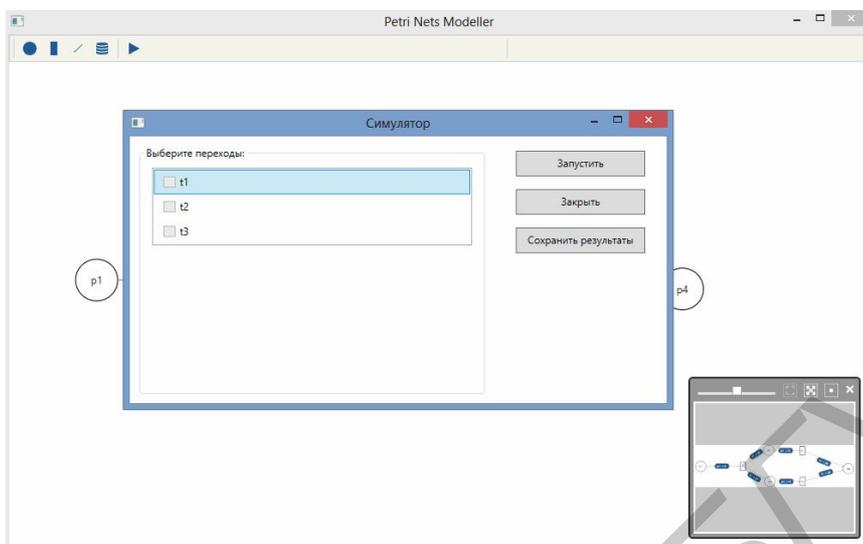


Рисунок 1.35 – Окно для задания сценариев срабатывания переходов

Ведутся работы по адаптации и внедрению разработанного приложения в деятельность кафедры экономической информатики БГУИР.

## 2 Моделирование оценки риска банкротства предприятий

На современном этапе развития образовательных технологий одним из важнейших факторов повышения качества подготовки специалистов в высших учебных заведениях является рационализация учебного процесса посредством оптимальных учебных планов, программ нового поколения, новых форм и методов преподавания.

В данном разделе рассматриваются перспективные методики преподавания дисциплины «Эконометрика», разработанные преподавателями кафедры экономической информатики, для студентов первой и второй ступени высшего образования.

Целью данных технологий организации учебного процесса является *создание условий для включения каждого обучаемого в деятельность, соответствующую зоне его ближайшего развития*, а также условий для самостоятельного (и/или под контролем преподавателя) усвоения программного материала в том размере и с той глубиной, которые позволяют индивидуальные особенности обучаемого.

### 2.1 Диагностики банкротства на основе моделей бинарного выбора

Модели, в которых переменная принимает только два различных значения, используются при исследовании влияния тех или иных субъективных и объективных факторов на наличие или отсутствие некоторого признака у отдельных домашних хозяйств (наличие или отсутствие в семье автомобиля), у отдельных индивидуумов (занятый – безработный), у отдельных фирм (обанкротилась или нет в течение определенного периода) и т.п. Если исследование затрагивает  $n$  субъектов, т.е. если имеется  $n$  наблюдений, то факт наличия или отсутствия такого признака в  $i$ -м наблюдении удобно индексировать числами 1 (наличие признака) и 0 (отсутствие признака). Тем самым можно определить индикаторную (дихотомическую, бинарную) переменную, которая принимает в  $i$ -м наблюдении значение  $y_i$ . При этом  $y_i=1$  при наличии рассматриваемого признака у  $i$ -го субъекта и  $y_i=0$  – при отсутствии рассматриваемого признака у  $i$ -го субъекта.

В качестве объясняющих факторов в моделях бинарного выбора могут выступать несколько переменных, и тогда имеем модель вида

$$y_i = G(\theta_1 x_{i1} + \dots + \theta_p x_{ip}) + \varepsilon_i = G(z) + \varepsilon_i, i = 1, \dots, n, \quad (2.1)$$

При этом в качестве  $G(z)$  функций может быть использована одна из следующих:

– функция стандартного нормального распределения  $N(0,1)$  (пробит-модель):

$$G(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^z e^{-t^2/2} dt; \quad (2.2)$$

– функция стандартного логистического распределения (логит-модель):

$$G(z) = \frac{e^z}{1 + e^z}; \quad (2.3)$$

Предсказанные значения ( $y$ ) в этой модели будут лежать в диапазоне от 0 до 1 и интерпретироваться как вероятность наличия качественного признака.

Для построения модели бинарного выбора, прогнозирующей вероятность банкротства субъекта хозяйствования, из 38 финансовых показателей 665 отечественных предприятий на основе корреляционного анализа отобраны 6 независимых количественных переменных, среди которых:

– Коэффициент текущей ликвидности (K1) (характеризует общую обеспеченность организации собственными оборотными средствами для ведения хозяйственной деятельности и своевременного погашения срочных обязательств организации).

– Коэффициент обеспеченности собственными оборотными средствами (K2) (характеризует наличие у организации собственных оборотных средств, необходимых для ее финансовой устойчивости).

– Коэффициент обеспеченности финансовых обязательств активами (K3) (характеризует способность организации рассчитаться по своим финансовым обязательствам после реализации активов).

– Коэффициент обеспеченности просроченных финансовых обязательств активами (K4) (характеризует способность организации рассчитаться по своим просроченным финансовым обязательствам после реализации активов).

– Отношение просроченных финансовых обязательств к общей сумме финансовых обязательств (K5) (показывает долю просроченных финансовых обязательств в общей сумме кредиторской задолженности).

– Отношение кредиторской задолженности по отчислениям в бюджет и ФСЗН к общей сумме финансовых обязательств (K6) (показывает долю задолженности перед бюджетом по обязательным платежам в общей сумме кредиторской задолженности).

Бинарная регрессионная пробит-модель в данном случае имеет вид:

$$G(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^z e^{-t^2/2} dt, \quad (2.4)$$

где

$$z = -8,23 + 6,91K1 + 6,31K2 + 1,56K3 - 3,30K4 - 4,48K5 + 0,81K6$$

Для анализа статистического качества модели используются:  $\chi^2 = 407,924$ ,  $df=6$ ,  $p$ -уровень  $< 0,05$  коэффициент детерминации  $R^2_{\text{predict}} = 0,96137$ .

Прогностические качества модели представлены в таблице 2.1

Таблица 2.1 – Процент верных предсказаний и отношение несогласия

Отношение несогласия:	19933		Процент верных предсказаний (общий):	99,55
	Предсказания			
Наблюдения	Значение 1	Значение 0	Процент верных предсказаний	
	Значение 1	67	1	98,529
Значение 0	2	595	99,665	

По представленным характеристикам модель может быть признана удовлетворительной.

## **2.2 Моделирование оценки риска банкротства предприятий с учетом лично-профессиональных характеристик руководителя организации.**

Любая организация в своей деятельности постоянно сталкивается с рисками различного рода, которые лежат в основе принятия всех управленческих решений. Одними из наиболее опасных видов рисков для предприятий являются финансовые риски, крайним проявлением которых является риск банкротства организации, ведущий ко временному или полному прекращению ее функционирования как субъекта хозяйствования, а при сохранении организации (ее неликвидации) – существенным изменениям в бизнес–процессах или профиле ее деятельности.

Таким образом, прогнозирование риска банкротства приобретает особую роль, а изучению методов, с помощью которых оно производится, уделяется серьезное внимание.

Актуальной тема исследования является еще и потому, что, по данным Национального статистического комитета Республики Беларусь, в 2013 году убыточными были признаны 752 организации, или 9,1% от общего количества организаций. В 2012 году количество убыточных организаций было меньшим и составляло 426 организаций, или 5 %. Сумма чистого убытка убыточных организаций за 2013 год составила 9 трлн. рублей, что в 2,5 раза больше, чем за 2012 год. Общее количество дел об экономической несостоятельности (банкротстве), находящихся в производстве хозяйственных судов Беларуси по состоянию на 01.01.2014 года, составило 1648, что на 5,8% больше по сравнению с аналогичным периодом 2013 года (1557 дел). Кроме того, количество дел, по которым принято решение о банкротстве с ликвидацией, преобладает над количеством дел, по которым вынесено решение об экономической несостоятельности с санацией, в 1,8 раза.

Поэтому важным представляется своевременно обнаружить и предвидеть вероятность наступления неблагоприятных ситуаций в деятельности предприятия. Для прогнозирования риска банкротства предприятия разработано большое количество подходов, основное число которых опирается на анализ финансовых показателей, таких как показатели рентабельности, ликвидности, стоимости капитала и прочих. Наибольшее распространение получили модели, основанные на методах дискриминантного анализа. Однако их использование в современных отечественных условиях затруднено особенностями национальной экономики, временными рамками, особенностями предприятий (их отраслевой принадлежностью, спецификой, условиями осуществления финансово–хозяйственной деятельности, местом расположения).

Также следует отметить, что в условиях финансово–экономического кризиса вопрос прогнозирования риска банкротства становится особенно острым, так как критерии оценки финансовых рисков часто являются главной основой для принятия правильных управленческих решений. В этой связи риск банкротства предприятия зависит не только от финансовых показателей его деятельности, но и от лица принимающего решение (директора либо другого должностного лица компании, ответственного за принятие решений в области финансов), его личностных качеств и поведенческих особенностей. Однако методов и моделей, учитывающих не только финансово–экономические факторы деятельности организации, но и поведенческие особенности в настоящее время не разработано.

Недостаточная научная и методическая разработанность проблемы анализа риска банкротства, и, как следствие, необходимость дальнейшей разработки и

совершенствования методов его оценки в условиях экономической нестабильности, определяют новизну и значимость настоящего исследования, а также обусловили выбор темы научного исследования и позволили сформулировать ее цель и научные задачи.

Таким образом, целью научного исследования (широко) является разработка эконометрической модели оценки риска банкротства как с учетом только финансовых показателей, так и с включением в модель факторов, характеризующих личные характеристики руководителя организации.

Реализация данной цели предусматривает решение взаимосвязанных научных задач, которые заключаются в следующем:

- изучить теоретические основы рисков и банкротства;
- провести сравнительный анализ существующих зарубежных и отечественных методов и моделей прогнозирования риска банкротства;
- изучить психологию поведения руководителей в ситуациях риска, выявить личные характеристик менеджеров, которые могут оказывать влияние на возникновение рисков ситуации.
- построить эконометрическую модель оценки риска банкротства в форме модели бинарного выбора как с учетом финансовых показателей, так и с включением в модель факторов, характеризующих личные качества руководителя организации.

В ходе проведенного исследования были получены следующие результаты:

а) Изучены теоретические основы рисков и банкротства. Отмечено, что риск – это возможность возникновения неблагоприятной ситуации или неудачного исхода производственно–хозяйственной или какой–либо другой деятельности организации. Выявлено, что риск банкротства является одними из наиболее опасных видов рисков и ведет к временному или полному прекращению функционирования предприятия как субъекта хозяйствования, а при сохранении организации (ее неликвидации) – существенным изменениям в бизнес–процессах или профиле ее деятельности. Рассмотрена такая категория как банкротство, под которой согласно Закону о банкротстве Республики Беларусь понимается неплатежеспособность, имеющая или приобретающая устойчивый характер, признанная решением хозяйственного суда о банкротстве с ликвидацией должника – юридического лица, прекращением деятельности должника – индивидуального предпринимателя (далее – решение об открытии ликвидационного производства). В свою очередь, под неплатежеспособностью понимают неспособность в полном объеме удовлетворить требования кредитора (кредиторов) по платежным обязательствам, а также по обязательствам, вытекающим из трудовых и связанных с ними отношений. Различают следующие виды неплатежеспособности организаций:

- 1) временная неплатежеспособность,
- 2) неплатежеспособность, приобретающая устойчивый характер,
- 3) неплатежеспособность, имеющая устойчивый характер. Причем

неплатежеспособность, приобретающая устойчивый характер, может считаться критическим состоянием организации, так как на данном уровне возможно как улучшение финансового состояния (восстановление платежеспособности) так и его ухудшение (потенциальное банкротства).

б) Проведен сравнительный анализ существующих зарубежных и отечественных методов и моделей оценки риска банкротства, который показал, что среди современных методов и методик оценки риска банкротства широкое применение получили дискриминантные модели (или Z–модели), основу которых составляет расчет интегрального показателя, производимый с помощью ряда коэффициентов финансовой деятельности предприятий. Применение данных моделей оценки риска банкротства в отечественных условиях сопровождается определенными трудностями: особенностями национальной экономики, временными рамками, особенностями предприятий (их

отраслевой принадлежностью, спецификой, условиями осуществления финансово-хозяйственной деятельности, местом расположения).

в) Изучена психология поведения руководителей в ситуациях риска, выявлены личные характеристики менеджеров, которые могут оказывать влияние на возникновение рискованной ситуации и в конечном итоге привести организацию к банкротству. Среди них особо отмечены пол, возраст, уровень образования и опыт руководящей работы.

г) Разработана эконометрическая модель прогнозирования риска банкротства в форме нелинейной модели бинарного выбора.

Исходными данными для построения данной модели риска банкротства являлась выборка, состоящая из показателей, характеризующих финансовое состояние организаций Республики Беларусь, а также факторов, влияющих на личностные качества ЛПР, таких как: пол, возраст, образование и опыт работы на руководящей должности. Признак состояния платежеспособности определяется экспертом на основании отечественной методики оценки платежеспособности и в соответствии с нормативами для коэффициентов, характеризующих состояние платежеспособности.

Принимая во внимание то, что заключение о банкротстве организации можно сделать на основании критериев платежеспособности, в качестве зависимой переменной представляется был выбран признак состояния платежеспособности организации ( $y$ ), принимающий значения 0 (предприятие платежеспособно) или 1 (предприятие неплатежеспособно).

Для выбора объясняющих переменных проведен анализ 43 фактора (38 количественных и 5 качественных), всесторонне характеризующих финансово-экономическое состояние предприятий, а также самого ЛПР. На основе предварительного анализа и анализа корреляционной матрицы группа факторов была сужена до 6 независимых количественных переменных, среди которых: коэффициент текущей ликвидности ( $K1$ ), коэффициент обеспеченности собственными оборотными средствами ( $K2$ ), коэффициент обеспеченности финансовых обязательств активами ( $K3$ ), коэффициент обеспеченности просроченных финансовых обязательств активами ( $K4$ ), отношение просроченных финансовых обязательств к общей сумме финансовых обязательств ( $K5$ ), отношение кредиторской задолженности по отчислениям в бюджет и ФСЗН к общей сумме финансовых обязательств ( $K6$ ), а также 4 качественных переменных, которыми являются следующие: пол ( $K7$ ), возраст ( $K8$ ), образование ( $K9$ ) и опыт работы на руководящей должности ( $K10$ ).

Построенная бинарная пробит-модель приняла следующий вид:

$$G(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^z e^{-t^2/2} dt \quad (2.5)$$

где

$$z = 2,474 + 3,70K1 + 4,59K2 - 4,02K3 + 10,55K4 - 10,20K5 + 1,25K6 - 1,41K7 - 9,34K8 + 2,71K9 - 8,66K10.$$

Все коэффициенты модели являются статистически значимыми на 5%-ом уровне. Гипотеза о статистической значимости модели не отклоняется:  $p$ -значение статистики  $\chi^2$  меньше 0,05.

Для построенной модели отношение несогласия составляет 19966, указывая, что построенная классификация также во много раз лучше, чем проведенная классификация наугад. Кроме того, процент правильно предсказанных моделью организаций с неплатежеспособностью, приобретающей устойчивый характер, составляет 98,529, а процент правильно предсказанных моделью платежеспособных предприятий равен 99,832. Эти значения показывают, что построенная модель дает хороший результат на

рассматриваемых данных. Если получить остатки для модели и возвести их в квадрат, то сумма квадратов остатков составит 1,193.

Исходя из полученных данных рассчитанный коэффициент детерминации  $R^2$ , доля неверных предсказаний по выбранной модели и доля неверных предсказаний по тривиальной модели (модели, в которой в качестве единственной объясняющей переменной выступает константа), составят 0,98246; 0,00179 и 0,10226 соответственно.

Это свидетельствует о том, что существует сильная связь между рассматриваемыми параметрами и вероятностью возникновения неплатежеспособности. Также рассчитанный показатель показывает, что построенная модель прогнозирования риска банкротства дает довольно точный результат (на 98,24%).

Таким образом, задачи, поставленные для научного исследования, выполнены полностью, цель достигнута. Модель прогнозирования риска банкротства построены по актуальным данным для отечественных предприятий, оценивает вероятность наступления риска банкротства, следовательно, она может быть использована для прогнозирования банкротства как на уровне предприятий Республики Беларусь, так и на уровне органов отраслевой принадлежности, отвечающих за назначение руководителя организации.

### **2.3 Методика проведения лабораторных работ по дисциплине «Эконометрика» для студентов первой и второй ступени высшего образования**

Эконометрические модели, а также методы статистического и математического анализа и прогнозирования в последнее время приобретают особую роль и активно применяются для исследования различных явлений и процессов в национальной экономике. В силу того, что дисциплина «Эконометрика» является одной из фундаментальных для современного образования, ее изучение студентами экономических и технических ВУЗов Республики Беларусь является необходимым, а ее преподаванию отводится особая роль.

Основные задачи преподавания эконометрики состоят в том, чтобы показать особенность эконометрики как науки, расположенной между экономикой, статистикой и математикой; научить студентов использовать данные или наблюдения для построения количественных зависимостей для экономических соотношений, для выявления связей, закономерностей и тенденций развития экономических явлений, а также выработать у студентов умение формировать экономические модели, основываясь на экономической теории или на эмпирических данных, оценивать неизвестные параметры в этих моделях, делать прогнозы и оценивать их точность.

Экономические процессы развиваются во времени, поэтому большое место в эконометрике занимают вопросы анализа и прогнозирования временных рядов, в том числе многомерных. При этом в одних задачах больше внимания уделяют изучению трендов (средних значений, математических ожиданий), например, при анализе динамики цен. В других же — важны отклонения от средней тенденции, например, при применении контрольных карт (кумулятивных сумм и др.).

Выполняя все эти задачи, будущий специалист сталкивается с необходимостью сбора и переработки информации, передаваемой по каналам средств массовой информации, в том числе и с использованием информационных технологий.

К сожалению, сложившаяся система образования часто не справляется с задачами, поставленными перед ней обществом, одна из которых — воспитание специалистов с высоким уровнем информационно-технологической подготовки, способных применять в своей работе инновационные методы, готовых быстро адаптироваться к новым требованиям производства. Возникает необходимость совершенствования подготовки студентов, осваивающих экономические и технические специальности, к использованию компьютерных технологий в будущей профессиональной деятельности.

С целью повышения качества преподавания дисциплины «Эконометрика» преподавателями кафедры экономической информатики учреждения образования «Белорусский государственный университет информатики и радиоэлектроники» разработана эффективная методика проведения лабораторных работ по данной дисциплине, позволяющая улучшить качество подготовки специалистов специальности ИСИТ в экономике, ИСИТ в логистике, электронный маркетинг и других специальностей экономической и технической направленности.

Разработанная методика позволяет закрепить основные понятия эконометрики, применяемых в корреляционно-регрессионном анализе, а также выработать практические рекомендации, призванные оказать помощь при выполнении индивидуальных заданий.

В процессе участия в такой лабораторной работе студенты имеют возможность использовать личный опыт, материалы из книг, журналов, газетных и интернет-изданий.

При этом они должны уметь обосновать выбор того или иного фактора для модели, пояснить, какое влияние (прямое или обратное, сильное или слабое и т.д.) он может оказывать на результивный признак, пояснить спецификацию выбранной модели, проверить ее качество.

В проведении такого занятия важен принцип «мозгового штурма», когда участники поощряются на высказывание любых мнений, творческое и конструктивное обсуждение вопросов. Вопросы могут задавать как преподаватель, так и другие студенты. Итогом становится качественно построенная регрессионная модель.

Демонстрацию разработанной методики проведем на основании темы: «Моделирование временных рядов с помощью программного пакета EViews (построение регрессионной модели временного ряда и проверка ее качества)».

Задание для данной лабораторной работы является следующим:

По предложенным данным построить качественную модель временного ряда с использованием пакета EViews.

Предлагаемые эндогенные переменные: прибыль или рентабельность, или себестоимость (выбираются студентами самостоятельно).

Выполнение задания предполагает следующее:

– проверка временных рядов на стационарность (для качественного подбора факторов модели);

– построение регрессионной модели временного ряда;

– проверка качества построенной модели (предполагает проверку остатков на нормальное распределение, на отсутствие автокорреляции, на наличие гомоскедастичности с помощью соответствующих тестов).

Исходные данные приведены на рисунке 2.1.

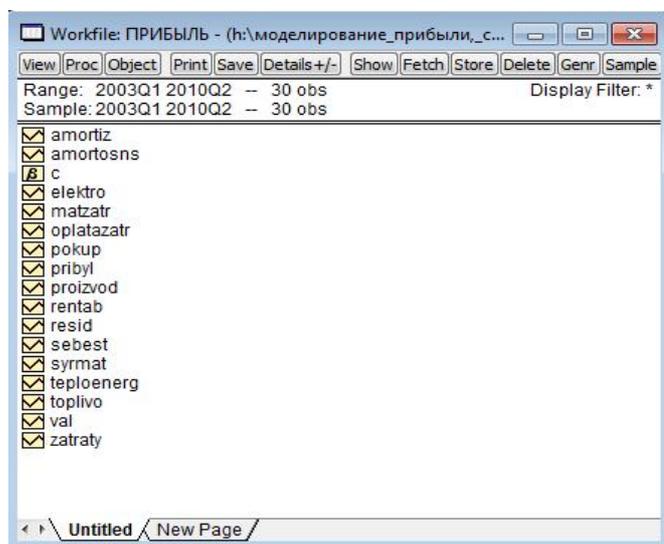


Рисунок 2.1 – Исходные данные для моделирования временных рядов в пакете EViews.

Представляется целесообразным расшифровать приведенные на рисунке 2.1 переменные:

- amortize (общие амортизационные отчисления),
- amortosns (амортизация основных средств),
- elektro (затраты на электроэнергию),
- matzatr (материальные затраты),
- oplatazatr (оплата затрат),
- pokup (оплата материалов),
- pribyl (прибыль),
- proizvod (производственные затраты),
- rentab (рентабельность),
- sebest (себестоимость),
- syrmat (затраты на сырьё и материалы),
- teploenerg (затраты на тепло и энергию),
- toplivo (затраты на топливо),
- val (выпуск продукции),
- zraty (прочие виды затрат).

Выполнение лабораторной работы предполагает последовательное выполнение следующих этапов:

**Этап предположения о характере регрессионной функции.** Методы подбора вида функции: *графический, аналитический и экспериментальный*. Существуют следующие виды функции:

- линейная:  $y = b_0 + bx$ ;
- полиномиальная:  $y = a_0 + a_1x + a_2x^2 \dots a_nx^n$ ;

- степенная:  $y=ax^b$ ;
- экспоненциальная:  $y=ae^x$ ;
- логистическая:  $y=k_1+ae^b x$ .

Для поставленной задачи предполагается линейная множественная регрессионная функция.

**Определение зависимой (эндогенной) и независимых (экзогенных) переменных.**

**Анализ исходных данных.** На этом этапе выполняется отбор параметров для множественной регрессии с учетом следующего:

Эндогенные и экзогенные переменные должны быть стационарны в одних и тех же типах (уровнях, первых разностях или вторых разностях), так как построение эконометрической модели по временным рядам, относящимся к разным типам стационарности, может привести к получению неадекватной модели, для которой не будут выполняться предпосылки метода наименьших квадратов, что приведет к невыполнению условий несмещенности, состоятельности и эффективности полученных оценок.

Факторы не должны быть взаимно коррелированы и, тем более, находиться в точной функциональной связи. Наличие высокой степени коррелированности между факторами может привести к неустойчивости и ненадежности оценок коэффициентов регрессии, а также к невозможности выделить изолированное влияние факторов на результативный показатель.

Реализация данного этапа предполагает выполнение следующих последовательных шагов:

- из предложенных переменных (прибыль, себестоимость и рентабельность) осуществить выбор эндогенной переменной. Пусть этой переменной будет прибыль (pribyl);
- проверить эндогенную переменную на стационарность (выполнить тест Дикки-Фуллера (тест единичного корня)). Для этого в пакете EViews выполняется следующая последовательность действий: в открытом окне временного ряда прибыль (pribyl) выбирается View→Unit Root Test. На рисунке 2.1.2 представлено окно проверки временного ряда на стационарность.

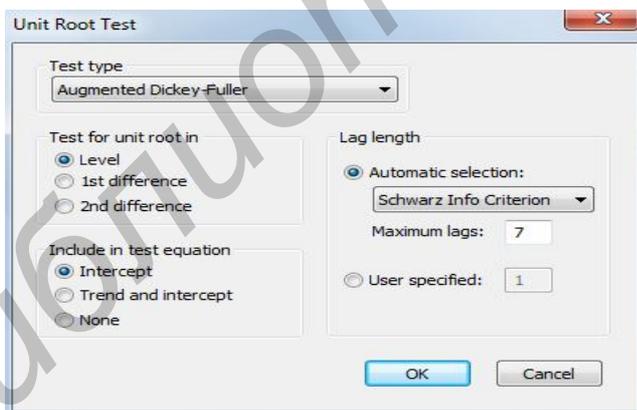


Рисунок 2.2 – Окно проверки временного ряда на стационарность.

При задании случаев стационарности (уровни, первые разности, вторые разности) дополнительно задаются следующие расширения для тестируемого временного ряда в порядке от более жесткого к более мягкому:

- дополнительное включение константы и линейного детерминированного тренда (соответствует Trendandintercept) является самым жестким,
- дополнительное включение только константы (соответствует Intercept) – более мягким,
- без дополнительного включения тренда и константы (соответствует None) – самым мягким.

Результат проверки на стационарность временного ряда прибыль (prbyl) при условии Trendandintercept приведен на рисунке 2.3.

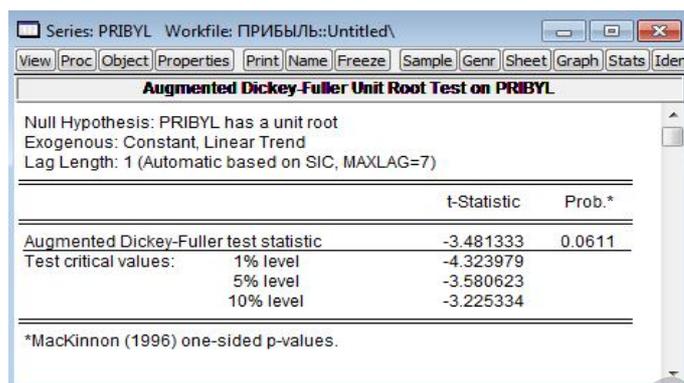


Рисунок 2.3 – Проверка на стационарность временного ряда прибыль (prbyl) в уровнях при расширении Trendandintercept.

Нулевая гипотеза ( $H_0$ ) для проверки временного ряда на стационарность заключается в следующем: «Временной ряд не стационарен».

Анализ полученных данных, приведенных на рисунке 2.3, показывает, что временной ряд прибыль (prbyl) при расширении Trendandintercept не стационарен в уровнях. Об этом свидетельствует расчетный вероятностный уровень  $p$  (probability) для  $t$ -статистики (t-Statistic, статистика Стьюдента), равный 0,0611, превышающий критический  $p$ -уровень 0,05 или 5%. Это в свою очередь свидетельствует о том, что  $H_0$  подтверждается.

Кроме того, о не стационарности временного ряда прибыль (prbyl) при условии Trendandintercept свидетельствует и сама расчетная  $t$ -статистика, равная -3,481333, которая сравнивается с критическим значением  $t$ -статистики 5%-го уровня, равным -3,580623. Так как  $t_{кр(0,05)} < t_{расч}$  это также свидетельствует о подтверждении нулевой гипотезы об отсутствии стационарности временного ряда.

Выполним еще раз *UnitRootTest* для временного ряда прибыль (prbyl) для дальнейшего исследования данного ряда на стационарность в уровнях с более мягким расширением Intercept. Результаты теста приведены на рисунке 2.4.

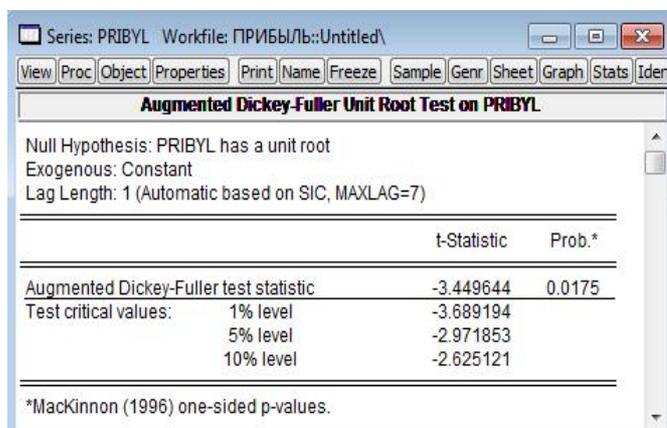


Рисунок 2.4 – Проверка на стационарность временного ряда прибыль (pribyl) в уровнях при расширении Intercept.

Анализ полученных данных, приведенных на рисунке 2.4, показывает следующее:

$$t_{кр(0,05)} = -2,97;$$

$$t_{расч} = -3,45;$$

$$p\text{-level критический для } t\text{-Statistic } (p_{кр(t)}) = 0,05;$$

$$p\text{-level расчетный для } t\text{-Statistic } (p_{расч(t)}) = 0,0175.$$

Так как  $t_{кр(0,05)} > t_{расч}$  и  $p_{кр(t)} > p_{расч(t)}$ , то  $H_0$  опровергается, действует альтернативная гипотеза  $H_1$ , которая свидетельствует о том, что временной ряд стационарен в уровнях. Следовательно, временной ряд прибыль (pribyl) можно отнести к временным рядам вида TS (стационарным в уровнях).

Далее необходимо:

- подобрать экзогенные переменные. При этом при моделировании прибыли в качестве факторов не могут быть использованы рентабельность и себестоимость, так как все эти показатели связаны между собой: например, рентабельность = прибыль / себестоимость.

Подбор экзогенных переменных проводится следующим образом. Выявив, что эндогенная переменная *pribyl* стационарна в уровнях подбираются факторы, стационарные в том же типе, что и зависимая переменная.

Исходя из данных, полученных при выполнении теста единичного корня, экзогенными переменными для моделирования прибыли могут быть следующие:

- matzatr (материальные затраты);
- summat (затраты на сырьё и материалы).

Результаты исследования данных временных рядов на стационарность приведены на рисунке 2.5.

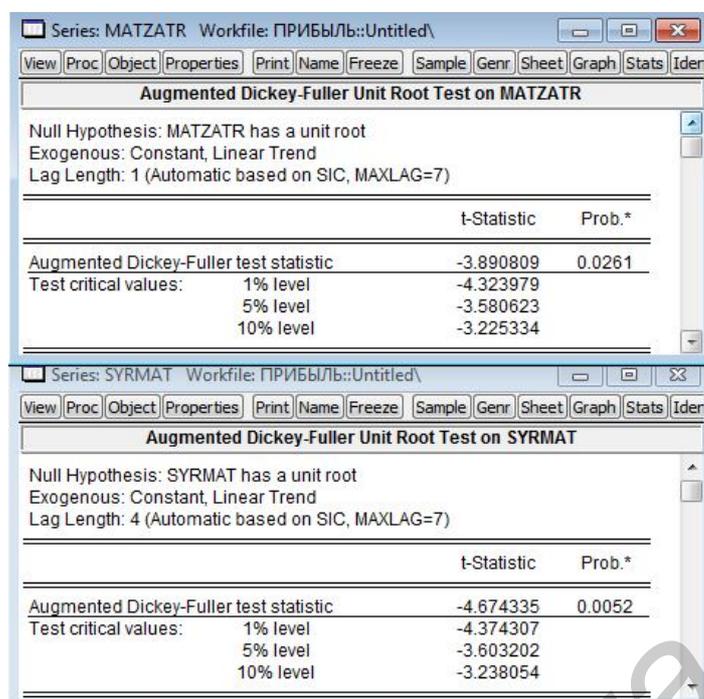


Рисунок 2.5 – Проверка на стационарность временных рядов matzatr и syrmat в уровнях.

Анализ полученных данных, приведенных на рисунке 2.5, показывает следующее:

– для временного ряда matzatr (материальные затраты):

$$t_{кр(0,05)} = -3,58;$$

$$t_{расч} = -3,89;$$

p-level критический для t-Statistic ( $p_{кр(t)} = 0,05$ );

p-level расчетный для t-Statistic ( $p_{расч(t)} = 0,0261$ ).

- для временного ряда syrmat (затраты на сырьё и материалы):

$$t_{кр(0,05)} = -3,60;$$

$$t_{расч} = -4,67;$$

p-level критический для t-Statistic ( $p_{кр(t)} = 0,05$ );

p-level расчетный для t-Statistic ( $p_{расч(t)} = 0,0052$ ).

Так как  $t_{кр(0,05)} > t_{расч}$  и  $p_{кр(t)} > p_{расч(t)}$ , то  $H_0$  опровергается, действует альтернативная гипотеза  $H_1$ , которая свидетельствует о том, что временные ряды matzatr и syrmat стационарны в уровнях. Следовательно, их также можно отнести к временным рядам вида TS (стационарным в уровнях).

Для исследования отобранных переменных на тесноту связи проводится корреляционный анализ. Тесноту связи изучаемых явлений оценивает линейный коэффициент парной корреляции  $r_{xy}$  для  $(-1 \leq r_{xy} \leq 1)$ . При этом учитывается следующее: чем ближе расчетное значение  $r_{xy}$  к 1 либо к -1, тем теснее связь между переменными, чем ближе расчетное значение  $r_{xy}$  к 0, тем слабее связь между переменными. Если расчетное значение  $r_{xy}$  имеет знак «-», то это означает, что между переменными существует обратный вид связи, который свидетельствует о том, что с ростом одного показателя наблюдается уменьшение второго показателя. Если расчетное значение  $r_{xy}$  имеет знак «+», то это означает, что между переменными существует прямой вид связи, который

свидетельствует о том, что с ростом одного показателя наблюдается увеличение второго показателя.

Для исследования эндогенной и экзогенных переменных на тесноту связи в пакете EViews их необходимо объединить в группу показателей, подлежащих исследованию с помощью корреляционного анализа. Это предполагает выполнения следующих действий:

– создать группу исследуемых показателей. Для этого выполняется следующий набор команд: Object→New Object→Group. Задается имя группы, например, gr1. Затем необходимо нажать ОК (приведено на рисунке 2.6).

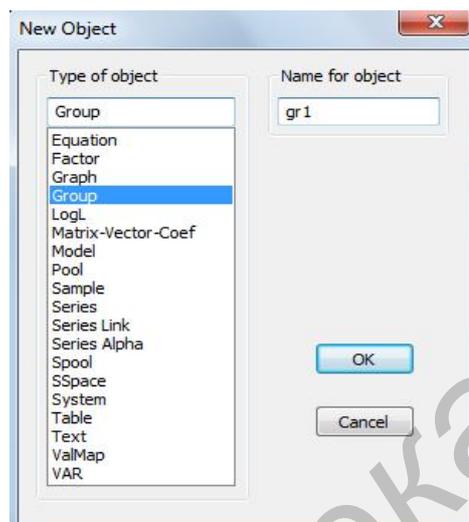


Рисунок 2.6 – Создание группы gr1.

– добавить в группу необходимые для исследования переменные. В рассматриваемом примере такими переменными будут prbyl, matzatr и syrmat.

Для этого в открытой группе gr1 выполняются команды: View→GroupMembers.

В появившееся окно вписываются исследуемые переменные через клавишу «ввод» (приведено на рисунке 2.7).

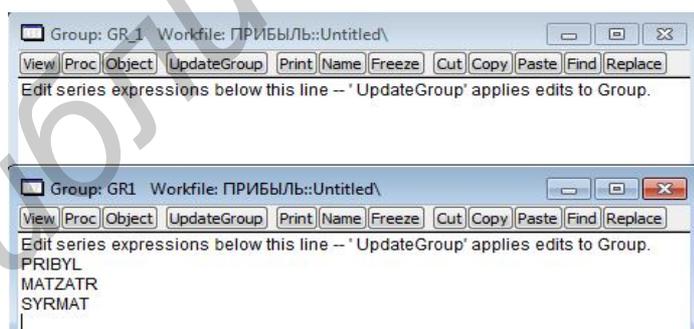


Рисунок 2.7 – Добавление в группу исследуемых переменных

Просмотр числовых значений добавленных переменных группы осуществляется через команды View→Spreadsheet(показано на рисунке 2.8).

obs	PRIBYL	MATZATR	SYRMAT
2003Q1	175397.6	1129080.	658545.8
2003Q2	210202.9	1145964.	710022.6
2003Q3	250402.9	1200477.	757630.6
2003Q4	222091.3	1268528.	789251.4
2004Q1	291776.5	1343452.	806608.0
2004Q2	338136.2	1410590.	895440.9
2004Q3	338060.1	1438739.	920768.2
2004Q4	296916.4	1476521.	921054.0
2005Q1	358543.9	1491426.	922002.8
2005Q2	374395.0	1524124.	992738.4
2005Q3	378068.3	1607206.	1066817.
2005Q4	322479.4	1655931.	1047318.
2006Q1	371978.7	1744267.	1103691.
2006Q2	402562.6	1744780.	1163314.
2006Q3	440271.4	1825932.	1218467.
2006Q4	392250.7	1796395.	1120893.
2007Q1	317687.0	1879564.	1151797.
2007Q2	400526.5	2048883.	1362420.
2007Q3	406052.2	2091614.	1386686.
2007Q4	390645.6	2169519.	1381198.
2008Q1	559317.7	2435235.	1566524.
2008Q2	671897.3	2549709.	1743953.
2008Q3	610157.0	2430413.	1631524.
2008Q4	185209.8	2154090.	1363446.
2009Q1	204149.2	1950078.	1161997.
2009Q2	293122.8	1968633.	1287854.
2009Q3	319389.6	1963865.	1296877.
2009Q4	297626.2	2101702.	1346686.
2010Q1	298976.4	2307371.	1399157.
2010Q2	379431.0	2549546.	1687871.

Рисунок 2.8 – Просмотр числовых значений добавленных переменных

– непосредственное выполнение корреляционного анализа. Осуществляется с помощью следующих команд: View→Covariance Analysis→Correlation (приведено на рисунках 2.9 – 2.10).

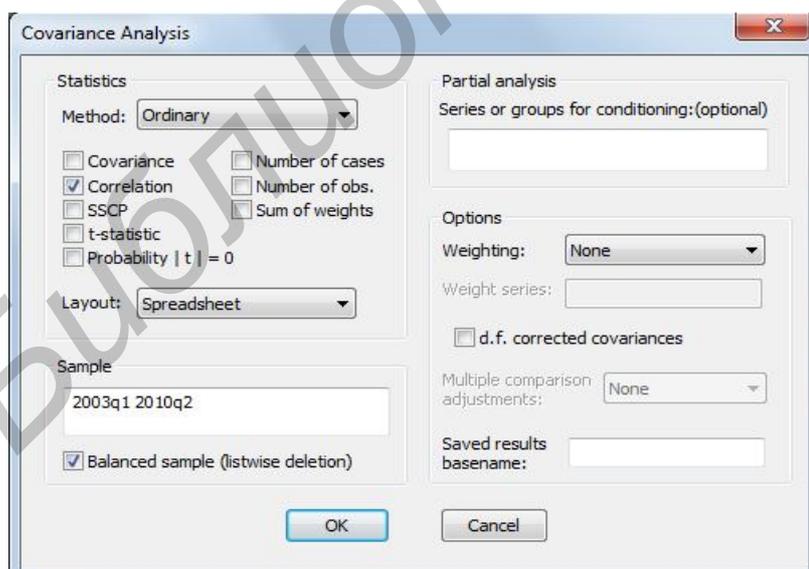


Рисунок 2.9 – Окно *CovarianceAnalysis*

Correlation			
	PRIBYL	MATZATR	SYRMAT
PRIBYL	1.000000	0.603306	0.667652
MATZATR	0.603306	1.000000	0.990143
SYRMAT	0.667652	0.990143	1.000000

Рисунок 2.10 – Окно *Correlation*

Анализ приведенной на рисунке 10 корреляционной матрицы показывает следующее:

Коэффициенты парной корреляции  $r_{xy}$  для *pribyl-matzatr* и *pribyl-syrmatr* равны 0,60 и 0,67 соответственно. Это означает, что каждый из факторов связан с эндогенной переменной умеренной корреляционной связью прямого типа и оказывает существенное влияние на зависимую переменную в сторону ее увеличения.

Коэффициент парной корреляции  $r_{xy}$  для *matzatr-syrmatr* равен 0,99, что свидетельствует о прямой тесной связи между факторами, то есть можно констатировать факт наличия мультиколлинеарности между переменными. Такое явление наблюдается потому, что материальные затраты включают в себя разные виды затрат, в том числе и затраты на сырьё и материалы, поэтому совместное включение в модель двух рассматриваемых факторов невозможно.

Для построения модели в дальнейшем будем использовать как фактор переменную *matzatr*.

Так как целью лабораторной работы является построение множественной линейной регрессионной модели, а на предыдущем этапе из двух возможных количественных переменных на основании корреляционного анализа осталась одна *matzatr*, представляется целесообразным ввести в будущую модель фиктивные переменные (переменные бинарного выбора), улучшающие качество модели, отвечающие за наличие либо отсутствие интересующего исследователя признака.

Фиктивные переменные в модель могут быть введены двумя способами:

- с помощью встроенных в EViews функций, которые записываются следующим образом: *@trend* (отражает динамику изменения временного ряда в сторону увеличения либо уменьшения в течение определенного периода времени), *@seas(1)*, *@seas(2)*, *@seas(3)*, *@seas(4)* (отражает сезонное изменение показателей, относящееся к 1, 2, 3 либо 4 кварталу соответственно), *c* (добавляет в модель свободный член), *ar(1)*, *ar(2)*, *ar(3)* (добавляют в модель авторегрессионный член 1, 2 и 3 порядка соответственно);

- с помощью смоделированных исследователем фиктивных переменных исходя из поведения зависимой переменной. Для данного способа выполняется графический анализ поведения эндогенной переменной, для чего строится линейный график.

Линейный график для переменной прибыль (*pribyl*) строится с помощью следующих команд: в открытом окне временного ряда *pribyl* выполняется *View*→*Graph*→*Line&Symbol*→*OK* (приведено на рисунке 2.11).

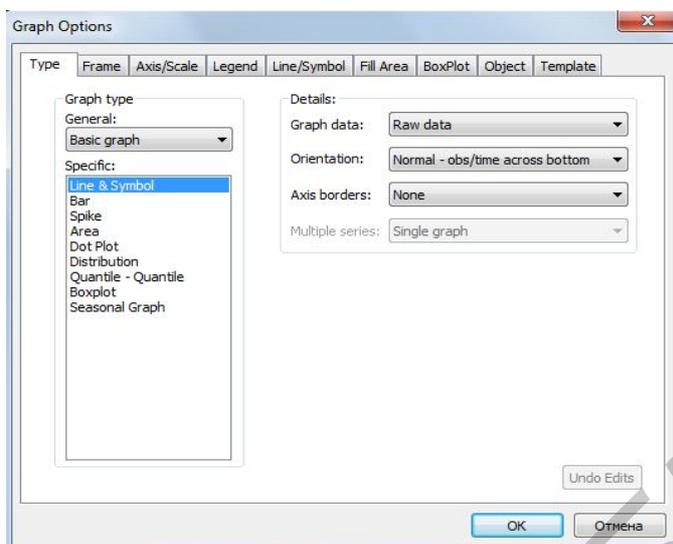


Рисунок 2.11 – Окно *View*→*Graph*

Сам линейный график для временного ряда прибыль (prbyl) приведен на рисунке 2.12.

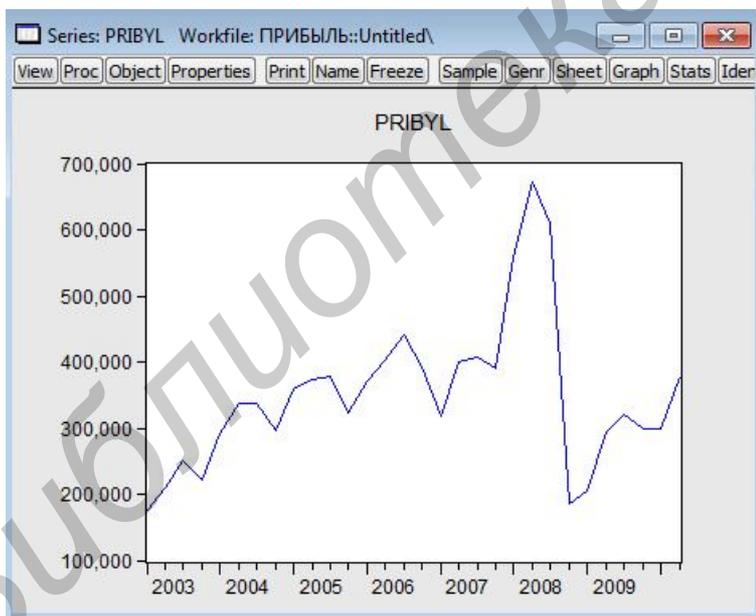


Рисунок 2.12 – Линейный график для временного ряда прибыль (prbyl)

Моделирование фиктивных переменных осуществляется с помощью следующего набора команд: *Object*→*NewObject*→*Series*. Задается имя временного ряда, например, sch1. Затем необходимо нажать ОК (приведено на рисунках 2.13-2.14).

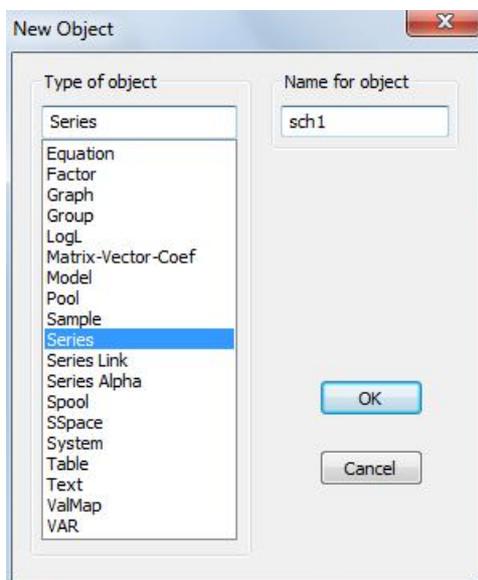


Рисунок 2.13 – Окно создания временных рядов (фиктивных переменных)

SCH_1	
Last updated: 01/10/15 - 15:51	
2003Q1	NA
2003Q2	NA
2003Q3	NA
2003Q4	NA
2004Q1	NA
2004Q2	NA
2004Q3	NA
2004Q4	NA
2005Q1	NA
2005Q2	NA
2005Q3	NA
2005Q4	NA
2006Q1	NA
2006Q2	NA
2006Q3	NA
2006Q4	NA
2007Q1	NA
2007Q2	NA
2007Q3	NA
2007Q4	NA
2008Q1	NA
2008Q2	NA
2008Q3	NA
2008Q4	NA
2009Q1	NA
2009Q2	NA
2009Q3	NA
2009Q4	NA
2010Q1	NA
2010Q2	NA

Рисунок 2.14 – Окно вновь созданного временного ряда

Заполнение данных временного ряда начинается с нажатия кнопки Edit+/. В случае, если моделируются сезонные изменения, то в наблюдения, являющиеся существенными для исследователя, проставляется 1, во все оставшиеся позиции 0. Если необходимо смоделировать трендовое изменение, то в ячейки, соответствующие существенным для исследователя наблюдениям, проставляются значения 1, 2, 3, 4, ... , n. При этом 1 проставляется в начало периода, соответствующего трендовым изменениям, а значение

проставляется в конец данного периода. Пример моделирования сезонных (sch\_1 и sch\_2) и трендовых (tch\_1 и tch\_2) фиктивных переменных приведен на рисунке 2.15.

The figure shows four spreadsheets side-by-side, each representing a different series: SCH\_1, SCH\_2, TCH\_1, and TCH\_2. Each spreadsheet has a grid with columns for 'View', 'Proc', 'Object', and 'Properties'. The data rows are organized by quarter and year, from 2003Q1 to 2010Q2. The values for SCH\_1 and SCH\_2 are binary (0.000000 or 1.000000), while TCH\_1 and TCH\_2 show a linear increasing trend over time.

Year	Quarter	SCH_1	SCH_2	TCH_1	TCH_2
2003	Q1	1.000000	0.000000	0.000000	0.000000
2003	Q2	0.000000	1.000000	0.000000	0.000000
2003	Q3	0.000000	0.000000	0.000000	0.000000
2003	Q4	0.000000	0.000000	0.000000	0.000000
2004	Q1	1.000000	0.000000	0.000000	0.000000
2004	Q2	0.000000	1.000000	0.000000	0.000000
2004	Q3	0.000000	0.000000	0.000000	0.000000
2004	Q4	0.000000	0.000000	0.000000	0.000000
2005	Q1	1.000000	0.000000	0.000000	0.000000
2005	Q2	0.000000	1.000000	0.000000	0.000000
2005	Q3	0.000000	0.000000	0.000000	0.000000
2005	Q4	0.000000	0.000000	0.000000	0.000000
2006	Q1	1.000000	0.000000	0.000000	0.000000
2006	Q2	0.000000	1.000000	0.000000	0.000000
2006	Q3	0.000000	0.000000	0.000000	0.000000
2006	Q4	0.000000	0.000000	0.000000	0.000000
2007	Q1	1.000000	0.000000	0.000000	0.000000
2007	Q2	0.000000	1.000000	0.000000	0.000000
2007	Q3	0.000000	0.000000	0.000000	0.000000
2007	Q4	0.000000	0.000000	0.000000	0.000000
2008	Q1	1.000000	0.000000	0.000000	0.000000
2008	Q2	0.000000	1.000000	0.000000	0.000000
2008	Q3	0.000000	0.000000	0.000000	0.000000
2008	Q4	0.000000	0.000000	0.000000	0.000000
2009	Q1	1.000000	0.000000	0.000000	0.000000
2009	Q2	0.000000	1.000000	0.000000	0.000000
2009	Q3	0.000000	0.000000	0.000000	0.000000
2009	Q4	0.000000	0.000000	0.000000	0.000000
2010	Q1	1.000000	0.000000	0.000000	0.000000
2010	Q2	0.000000	1.000000	0.000000	0.000000
2010	Q3	0.000000	0.000000	0.000000	0.000000
2010	Q4	0.000000	0.000000	0.000000	0.000000

Рисунок 2.15 – Пример моделирования фиктивных сезонных (sch\_1, sch\_2) и трендовых (tch\_1, tch\_2) переменных.

Для демонстрации настоящей лабораторной работы будем пользоваться встроенными в EViews функциями. Теперь можно переходить к следующему этапу.

**Этап построения регрессионной модели и оценка ее параметров.**

Построение регрессионной модели осуществляется следующим образом. Выполняется следующая последовательность команд: *Object*→*NewObject*→*Equation*. Задается имя модели, например, m2. Затем необходимо нажать ОК (приведено на рисунке 2.16).

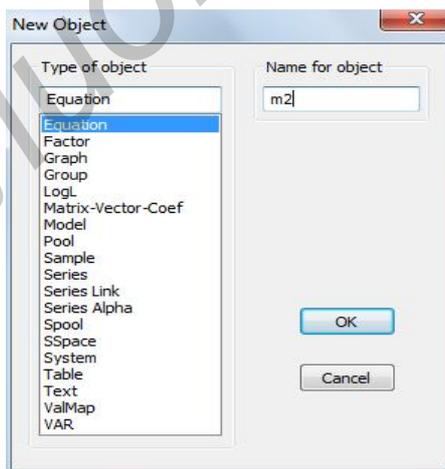


Рисунок 2.16 – Создание модели

В появившемся окне спецификации модели вводятся через пробел необходимые для построения модели переменные. При этом ввод переменных всегда начинается с эндогенной переменной, ввод факторов осуществляется после зависимой переменной в любом порядке (приведено на рисунке 2.17).

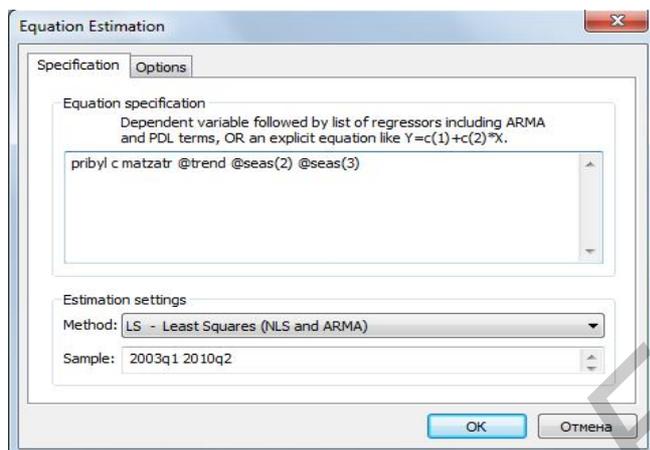


Рисунок 2.17 – Ввод членов модели и определение ее вида

Далее осуществляется следующий этап.

**Проверка статистической значимости коэффициентов регрессии построенной модели.** В качестве  $H_0$  берется предположение, что среди факторов, формирующих модель, есть хотя бы один статистически не значимый. Для ее проверки рассчитывается t-статистика:  $t_b = bsb$ . Если  $|t_{расч}| < t_{кр}$ , то гипотеза  $H_0$  не отвергается, т.е. выбранная переменная никакого влияния на эндогенную переменную не оказывает, соответственно экзогенная переменная признается не значимой. Об этом же свидетельствует и расчетный p-уровень для t-статистики, если он больше критического 0,05. В этом случае нулевая гипотеза также не отвергается.

Проведем анализ статистической значимости коэффициентов регрессии для построенной модели по данным, приведенным на рисунке 2.18.

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	-319413.8	78050.45	-4.092401	0.0004
MATZATR	0.491461	0.063632	7.723517	0.0000
@TREND	-17260.87	3027.694	-5.700997	0.0000
@SEAS(2)	44170.14	26359.79	1.675664	0.1063
@SEAS(3)	71138.76	27378.30	2.598363	0.0155

R-squared	0.765606	Mean dependent var	349924.1
Adjusted R-squared	0.728103	S.D. dependent var	114547.2
S.E. of regression	59729.26	Akaike info criterion	24.98404
Sum squared resid	8.92E+10	Schwarz criterion	25.21758
Log likelihood	-369.7607	Hannan-Quinn criter.	25.05875
F-statistic	20.41447	Durbin-Watson stat	1.189921
Prob(F-statistic)	0.000000		

Рисунок 2.18 – Множественная регрессионная модель

Полученные данные свидетельствуют о следующем. Среди факторов есть статистически не значимый показатель (соответствует @seas(2)), для которого подтверждается нулевая гипотеза.

Об этом свидетельствует значение по модулю  $|t_{расч}| = 1,676$ , которое является меньшим по отношению к  $t_{кр} = 1,708$  (критическое значение t-статистики находится по статистической таблице процентных точек распределения Стьюдента в зависимости от степеней свободы (в нашем случае оно 25) и уровня процентной точки  $\alpha = 0,05$ ). Кроме того, об этом свидетельствует и вероятностный уровень для t-статистики  $p_{расч} = 0,1063$ , который выше  $p_{кр} = 0,05$ .

Следовательно, модель придется строить заново, исключив при этом статистически незначимый показатель.

Для этого выполняется команда *Estimate*, в появившемся окне удаляется статистически незначимый показатель и нажимается кнопка ОК (показано на рисунке 2.19).

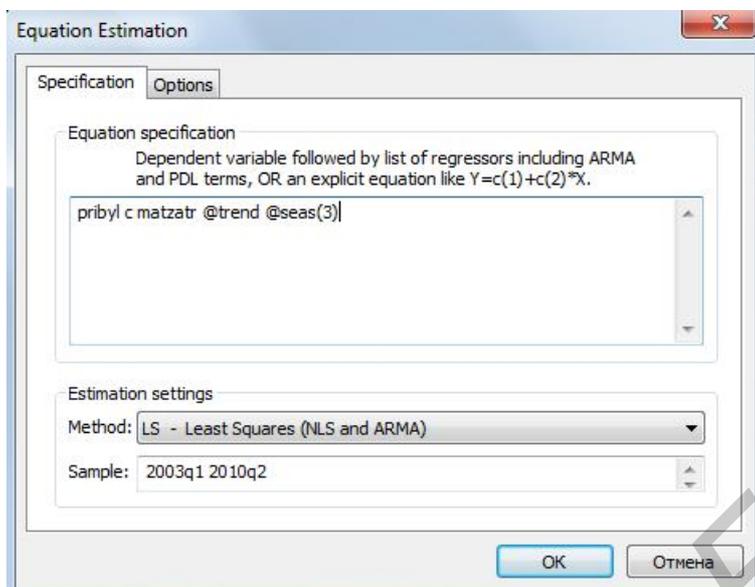


Рисунок 2.19 – Спецификация модели без статистически незначимого показателя @seas(2)

Результаты заново построенной регрессионной модели приведены на рисунке 2.20.

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	-320452.5	80715.84	-3.970131	0.0005
MATZATR	0.504630	0.065303	7.727518	0.0000
@TREND	-17776.64	3114.964	-5.706851	0.0000
@SEAS(3)	55772.43	26678.64	2.090528	0.0465

R-squared	0.739280	Mean dependent var	349924.1
Adjusted R-squared	0.709197	S.D. dependent var	114547.2
S.E. of regression	61770.93	Akaike info criterion	25.02382
Sum squared resid	9.92E+10	Schwarz criterion	25.21065
Log likelihood	-371.3573	Hannan-Quinn criter.	25.08359
F-statistic	24.57461	Durbin-Watson stat	1.324177
Prob(F-statistic)	0.000000		

Рисунок 2.20 – Новая регрессионная модель.

Анализ полученных данных позволяет сделать вывод о том, что среди них нет ни одного статистически незначимого, так как  $H_0$  опровергается. Об этом свидетельствуют значения по модулю  $|t_{расч}|$  для каждого из факторов модели, которые являются большими по отношению к  $t_{кр} = 1,708$ . Кроме того, об этом свидетельствуют и вероятностные уровни для t-статистики  $p_{расч(t)}$ , которые ниже  $p_{кр(t)} = 0,05$  для каждого из факторов. Далее выполняется следующий этап.

**Проверка статистической значимости полученного уравнения регрессии.** На этом этапе производится оценка достаточно ли велик коэффициент детерминации  $R^2$ , чтобы говорить о существовании значимой связи между величинами  $x$  и  $y$ , а также рассчитывается значение  $F$ -критерия Фишера. В качестве нулевой гипотезы  $H_0$  берется предположение о статистической не значимости модели в целом. Если  $F_{расч} > F_{кр}$  и  $p$ -уровень для критерия Фишера расчетный меньше  $p$ -уровня критического 0,05, то  $H_0$  отвергается, иначе принимается.  $F_{кр}$  берется из статистической таблицы для заданного уровня значимости исходя из степеней свободы для числителя и знаменателя, в нашем случае это (4;25).

Анализ данных, приведенных на рисунке 2.1.20, показал, что построенная модель статистически значима в целом, так как  $F_{расч} > F_{кр}$ , то есть  $24,57 > 2,76$ , кроме того, об этом свидетельствует и  $p_{расч(F)} < p_{кр(F)}$ , то есть  $0 < 0,05$ . Значит, нулевая гипотеза о статистической не значимости модели в целом опровергается.

Кроме того, коэффициент детерминации  $R^2=0,74$  стремится к 1. Это указывает на довольно точную подгону факторов модели и свидетельствует о том, что модель объясняет на 74% долю дисперсии в общей дисперсии результативного признака прибыль (pribyl).

Посмотреть построенную модель регрессии и выписать уравнение регрессии можно с помощью следующих команд: *View*→*Representations*(результат приведен на рисунке 2.21).



Рисунок 2.21 – Спецификация регрессионной модели и уравнение регрессии

Таким образом, уравнение множественной регрессии имеет следующий вид:

$$PRIBYL = -320452.463618 + 0.504630355708 * MATZATR - 17776.6381798 * ds_1 + 55772.4307646 * ds_2 + \varepsilon_t, \text{ где}$$

$ds_1$  – @TREND – автоматически вызываемая программой функция, отражает тренд модели;

$ds_2$  – @SEAS(3) – автоматически вызываемая программой функция, отражает сезонные изменения модели, относящиеся к 3 кварталу каждого из наблюдаемых периодов.

Следующим этапом исследования регрессионной модели является изучение остатков модели.

**Исследование остатков для проверки модели на адекватность.** Предполагает проверку следующих предпосылок метода наименьших квадратов (МНК):

*Остатки подчиняются нормальному распределению.*

Для проверки остатков на нормальность распределения выполняется тест Жака-Бера. Для проверки по критерию Жака-Бера используется тот факт, что у нормального распределения коэффициент асимметрии равен нулю, а коэффициент эксцесса равен 3. Нулевая гипотеза для проверки остатков на нормальность распределения по тесту Жака-Бера заключается в том, что остатки распределены нормально. Сам критерий Жака-Бера сравнивается с критерием  $\chi^2$  (хи-квадрат) для 2 степеней свободы и уровня процентной точки  $\alpha=0,05$ . Если критерий  $JB < \chi^2_{(2;0,05)}$  и вероятностный уровень  $p_{расч(JB)}$  для критерия JB будет больше критического 0,05, то  $H_0$  подтверждается и можно утверждать, что остатки распределены нормально. Для проверки остатков модели на нормальность распределения выполняется следующий набор команд: *View*→*ResidualTests*→*Histogram – NormalityTest* (результаты представлены на рисунке 2.22).

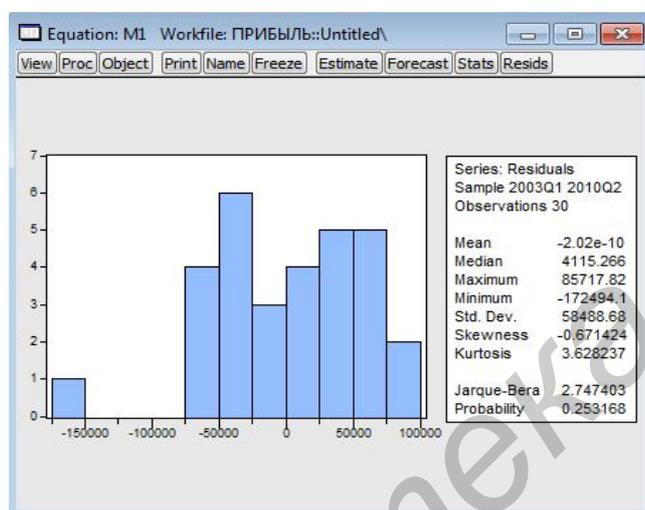


Рисунок 2.22 – Результаты проверки остатков на нормальное распределение

Анализ полученных данных свидетельствует о следующем: так как критерий  $JB=2,747 < \chi^2_{(2;0,05)}=5,99$  и  $p_{кр(JB)}=0,05 < p_{расч(JB)}=0,253$ , можно утверждать, что остатки распределены нормально (нулевая гипотеза подтверждается). Кроме того, расчетный коэффициент асимметрии, равный  $-0,67$ , стремится к 0, а коэффициент эксцесса, равный 3,63, стремится к 3, что в свою очередь также свидетельствует о нормальном распределении остатков.

*Проверка отсутствия автокорреляции остатков.* Это означает, что значения остатков распределены независимо друг от друга. Для обнаружения используется тест Бреуша — Годфри. Для проверки остатков на отсутствие автокорреляции необходимо выполнить следующие команды: *View*→*ResidualTests*→*SerialCorrelationLMTTest*. В появившемся окне проставляется значение *Lagstoinclude*: например 1 (смотри рисунок 2.23).

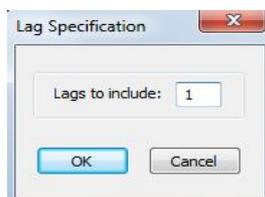


Рисунок 2.23 – *LagSpecification*

Лаг – это порядок коэффициента автокорреляции, т.е. число периодов, по которым рассчитывается коэффициент. Например, если лаг равен 1, то будет рассчитан коэффициент автокорреляции первого порядка, измеряющий зависимость между соседними остатками. Результаты анализа остатков на автокорреляцию представлены на рисунке 2.24.

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	12390.22	78312.83	0.158214	0.8756
MATZATR	-0.004899	0.063148	-0.077574	0.9388
@TREND	5.451858	3008.992	0.001812	0.9986
@SEAS(3)	-19036.22	28119.18	-0.676983	0.5046
RESID(-1)	0.360807	0.213214	1.692232	0.1030

R-squared	0.102774	Mean dependent var	-2.02E-10
Adjusted R-squared	-0.040783	S.D. dependent var	58488.68
S.E. of regression	59669.42	Akaike info criterion	24.98204
Sum squared resid	8.90E+10	Schwarz criterion	25.21557
Log likelihood	-389.7306	Hannan-Quinn criter.	25.05675
F-statistic	0.715912	Durbin-Watson stat	1.709332
Prob(F-statistic)	0.588978		

Рисунок 2.24 – Результаты анализа остатков на автокорреляцию

В соответствии с предпосылками МНК, остатки  $\varepsilon_t$  должны быть случайными. Однако при моделировании временных рядов нередко встречается ситуация, когда остатки содержат тенденцию или циклические колебания. Это свидетельствует о том, что каждое следующее значение остатков зависит от предшествующих.

Фактическое значение F-критерия Фишера –  $F_{расч} = 2,863649$ ,  $p = 0,1030$ . Сравним его с табличным значением F-критерия по заданному уровню значимости ( $\alpha = 0,05$ ) и числу степеней свободы ( $d.f.1 = 1$  и  $d.f.2 = 25$ )  $F_{кр} = 4,24$ .

$$F_{кр} = 4,24 > F_{расч} = 2,863649, p_{расч} = 0,1030 > p_{кр} = 0,05$$

Полученные значения свидетельствуют о том, что нулевая гипотеза об отсутствии автокорреляции подтверждается

Теперь рассмотрим критерий  $Obs * R^2$ , который ставняется с критерием хи-квадрат. Наблюдаемое значение критерия и соответствующий уровень значимости соответственно равны:

$$Obs * R^2 = 3,082310, \quad p = 0,0791$$

Проверим гипотезу об отсутствии автокорреляции при уровне значимости 0,05.

$$p = 0,05; k = 1$$

$$\chi^2_{(0,05;1)} = 3,84$$

$$\chi^2_{кр} = 3,84 > Obs * R^2 = 3,082310, p_{расч} = 0,0791 > p_{кр} = 0,05$$

Можно также сделать вывод о том, что в остатках отсутствует автокорреляция и нулевая гипотеза также подтверждается.

*Проверка гомоскедастичности остатков.* Это означает, что дисперсия каждого отклонения  $\varepsilon_i$  одинакова для всех значений  $x$ . Проверить ее можно с помощью теста Уайта. Для этого выполняется следующая последовательность команд: *View*→*ResidualTests*→*HeteroskedasticityTest* (приведено на рисунке 2.25).

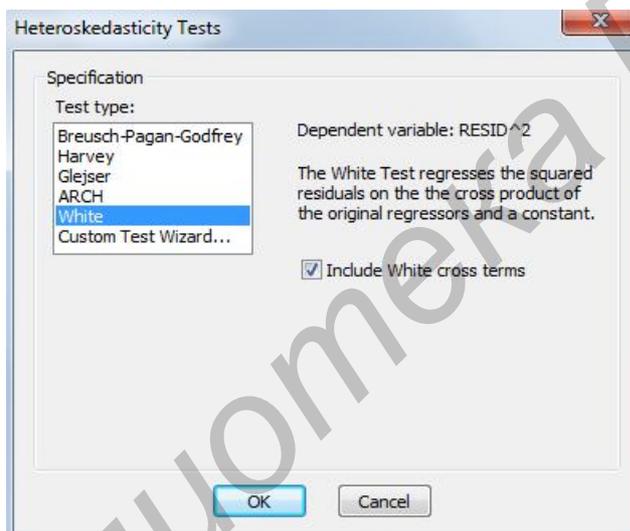


Рисунок 2.25 – Окно *HeteroskedasticityTest*

В появившемся окне *HeteroskedasticityTest* выбираем White и проводим данный тест с учетом попарного перемножения факторов (IncludeWhitexrossterms) и без такого перемножения (снимается флажок IncludeWhitexrossterms). Результаты теста Уайта с учетом и без учета попарного перемножения факторов приведены на рисунке 2.1.26.

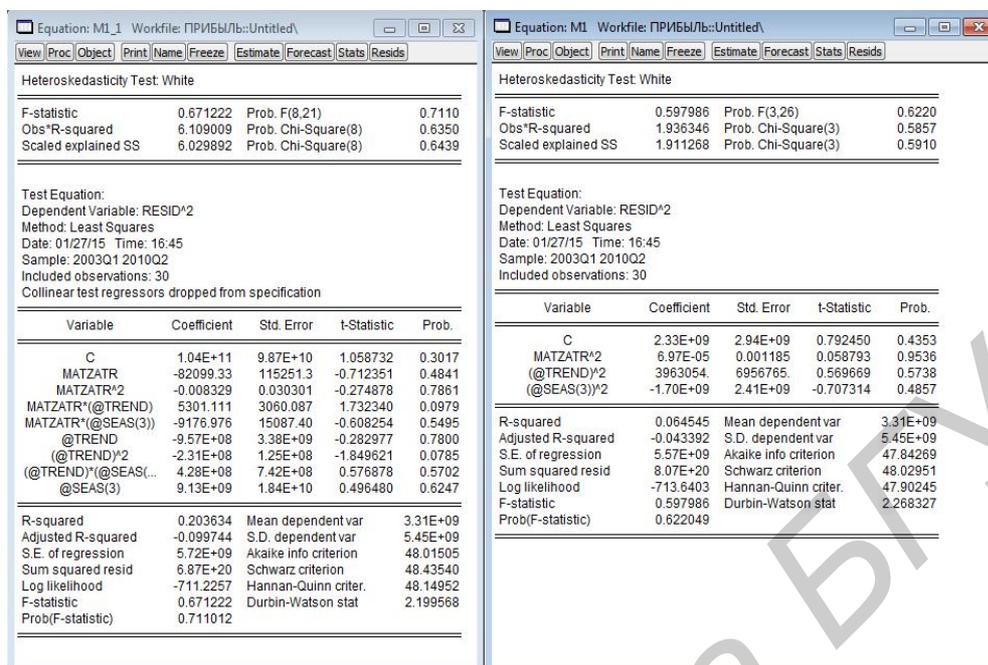


Рисунок 2.26 – Результаты теста Уайта с учетом и без учета попарного перемножения факторов

Сравнение теста Уайта с учетом и без учета попарного перемножения факторов позволяет сделать вывод о том, что модель с попарным перемножением лучше, чем без него, так как  $p$ -уровни для  $F$ -статистики и  $\chi^2$  выше. Это свидетельствует о том, что модель не аддитивная, а мультипликативная.

Далее произведем анализ лучшего результата.

Фактическое значение  $F$ -критерия Фишера –  $F_{расч} = 0,671$ ,  $p = 0,7110$ . Сравним его с табличным значением  $F$ -критерия по заданному уровню значимости ( $\alpha = 0,05$ ) и числу степеней свободы ( $d.f.1 = 8$  и  $d.f.2 = 21$ )  $F_{кр} = 2,42$ .

$$F_{кр} = 2,42 > F_{расч} = 0,671, p_{расч} = 0,7110 > p_{кр} = 0,05$$

Полученные значения свидетельствуют о том, что нулевая гипотеза о гомоскедастичности остатков подтверждается.

Теперь рассмотрим оба критерия ( $Obs * R^2$  и  $ScaledExplainedSS$ ), которые сравниваются с критерием хи-квадрат. Наблюдаемое значение критериев и соответствующий уровень значимости соответственно равны:

$$Obs * R^2 = 6,1090, \quad p_{расч} = 0,6350$$

$$ScaledExplainedSS = 6,0299, \quad p_{расч} = 0,6429$$

Проверим гипотезу о гомоскедастичности остатков при уровне значимости 0,05.

$$p = 0,05; k = 8$$

$$\chi^2_{0.05;8} = 15,51$$

$$\chi_{кр}^2 = 15,51 > Obs * R^2 = 6,1090, p_{расч}=0,6350 > p_{кр}=0,05$$
$$\chi_{кр}^2 = 15,51 > Scaled Explained SS = 6,0299, p_{расч}=0,6429 > p_{кр}=0,05$$

Можно также сделать вывод о том, что нулевая гипотеза о гомоскедастичности остатков подтверждается.

На следующем этапе делаются выводы относительно качества построенной модели и ее адекватности.

***Выводы о построенной модели.***

Исходя из полученных данных, можно сделать вывод о том, что была построена статистически значимая модель линейной множественной регрессии зависимости прибыли от факторов, влияющих на нее, которая содержит статистически значимые показатели и описывает выборку на 74%.

Построенная модель является адекватной, так как остатки имеют нормальное распределение, в них отсутствует автокорреляция и они гомоскедастичны.

В заключении отметим, что разработанные материалы рассчитаны и на студентов второй ступени высшего образования, которые уже обладают навыками эконометрического анализа и умеют работать со статистическими пакетами прикладных программ, такими как EViews, Statistica и т.д. Преимущества их использования заключается в том, что магистранты расширяют свои знания по теории применения моделей различного вида и приобретают практические навыки построения статистически значимых моделей (на примере моделей оценки риска банкротства) по актуальным данным и на перспективную для Республики Беларусь тему.

### **3 Исследование работоспособности организационных процессов в социально-экономических системах**

Операционный менеджмент занимается разработкой организационных систем, которые обеспечивают максимально эффективное использование материалов, человеческих ресурсов, оборудования и производственных помещений в процессе изготовления продукции или оказания услуг и управление ими. Концепции и методы операционного менеджмента широко применяются в управлении другими функциями бизнеса. Операционный менеджмент — это деятельность, связанная с разработкой, а также использованием и совершенствованием производственных систем, на основе которых производятся основная продукция или услуги компании.

Операционная стратегия заключается в разработке общей политики и планов использования ресурсов фирмы, нацеленных на максимально эффективную поддержку ее долговременной конкурентной стратегии. Различают несколько основных типов операционных приоритетов: издержки производства, качество и надежность продукции, срок выполнения заказа, надежность поставок, способность фирмы реагировать на изменение спроса, гибкость и скорость освоения новых товаров и специфические для каждого отдельного товара критерии.

Операционные услуги можно разделить на четыре большие категории: информационная поддержка, устранение проблем, поддержка при продаже и оперативное техническое обслуживание.

– Информационная поддержка обеспечивает предоставление исчерпывающих сведений относительно технических характеристик продукции, параметров ее обработки и себестоимости;

– Устранение проблем — это способность компании помочь группам внешних и внутренних потребителей в решении возможных проблем;

– Поддержка при продаже способствует повышению эффективности маркетинговых мероприятий компании и увеличению объемов продаж;

– Оперативное техническое обслуживание — это способность компании быстро заменять неисправные комплектующие.

Проблемы, стоящие перед специалистом в сфере операционного менеджмента, тесно коррелируют с проблемами, с которыми сталкиваются специалисты в области управления социально-экономическими системами. Как правило, речь идет о системах с несколькими уровнями работоспособности. Одно из направлений анализа надежности систем с несколькими уровнями работоспособности состоит в исследовании значимости или влияния отдельных элементов системы на различные уровни ее работоспособности. Математические методы логического дифференциального исчисления позволяют проанализировать изменение состояний работоспособности системы в зависимости от изменения состояний ее элементов.

Примечание [p1]: Для введения

#### **3.1 Применение многозначной логики в исследовании работоспособности социально-экономических систем**

В процессе системного анализа любой объект или процесс можно рассматривать как комплекс взаимосвязанных элементов. Обычно при анализе надежности или безопасности объекта и его элементов исследуются два возможных состояния: работоспособная система и неработоспособная система. Для математического описания таких объектов

используется математическая модель с двумя состояниями. Однако, в действительности состояний куда больше и порой для аналитика важно, не столько при каких условиях система будет работоспособна или нет, а – при каких условиях система будет частично работоспособна, какие меры необходимо предпринять для перевода системы из частично работоспособного состояния в полностью работоспособное. Для анализа таких промежуточных состояний работоспособности используются специальные математические модели, получившие название «системы с несколькими уровнями работоспособности». Использование таких моделей позволяет более детально исследовать надежность объекта, условия его функционирования и отказа. Одно из направлений анализа надежности систем с несколькими уровнями работоспособности состоит в исследовании значимости или влияния отдельных элементов системы на различные уровни ее работоспособности. В англоязычной литературе это направление получило название Importance Analysis [13], а математическая модель анализируемого объекта или процесса нескольких уровней работоспособности получила название Multi-State System (MSS) [14]. Эта модель позволяет представить исходный объект как систему с несколькими уровнями работоспособности. Например, это могут быть такие состояния работоспособности как отказ, минимальная, частичная и полная работоспособности системы. При этом традиционно используемые математические модели, в которых система имеет только два состояния – отказ и работоспособность, являются частным случаем MSS. Использование MSS в качестве математической модели для анализа надежности впервые было предложено в 70-е годы [15, 16]. В этих работах были определены основные теоретические понятия для анализа исходного объекта с помощью MSS. В частности, для MSS было введено понятие структурной функции (Structure Function). Эта функция устанавливает соответствие между состояниями отдельных элементов системы и ее надежностью.

Практическое использование математической модели MSS имеет ряд сложностей. К этим сложностям, во-первых, относят большую вычислительную нагрузку, связанную с размерностью структурной функции. Если размерность структурной функции BSS определяется как  $2^n$ , то для системы из  $n$  элементов с  $m$  состояниями работоспособности эта размерность соответствует  $m \times n$ . Снятие этого затруднения авторы видят в использовании новых алгоритмов обработки структурной функции, ориентированных на регулярные, параллельно-конвейерные вычисления.

Во-вторых, затруднения связаны с отсутствием однозначной и устоявшейся интерпретации элементарных MSS: параллельной, последовательной и  $k$ -из- $n$  ( $k$ -out-of- $n$ ).

В настоящее время для преодоления этих затруднений предложены эффективные методы решения. Так, представление структурной функции MSS в виде многозначных диаграмм решений (Multiple-Valued Decision Diagram, MDD) позволяет разрешить затруднения, связанные с большой размерностью, поскольку представление структурной функции в виде MDD требует меньше ресурсов, чем аналитическое [17, 18]. В последних работах по анализу надежности MSS предложен ряд алгоритмов для вычислений известных и новых оценок для анализа MSS заданных в виде MDD. Так, в работе [18] и ряде других рассмотрены алгоритмы вычисления частотных характеристик MSS. Для анализа динамического поведения MSS на основе ее структурной функции авторами предложено использовать математический аппарат логического дифференциального исчисления [19]. Этот математический аппарат позволяет исследовать зависимость

изменения значения логической функции от изменения значения ее одной или нескольких переменных. В случае, когда значения переменных структурной функции соответствуют состояниям работоспособности элементов системы, а значение функции – уровню работоспособности MSS, то математические методы логического дифференциального исчисления позволяют проанализировать изменение состояний работоспособности системы в зависимости от изменения состояний ее элементов. Анализ публикаций показывает, что, развитие этого направления в исследовании MSS представляется на сегодняшний день достаточно перспективным.

В заключении отметим, что в настоящее время математические модели MSS и алгоритмы их анализа успешно используются при оценке надежности и решении прикладных задач в области энергетики, обеспечения работы морского порта и морской транспортной системы. Известны примеры использования MSS при оценке надежности медицинских и логистических систем, газо- и нефтепроводов, программного обеспечения, в инженерной практике.

В настоящее время существуют различные подходы к оценке надежности систем. Многие из них пришли из теории надежности технических систем и нашли свое применение в оценке рисков управления экономическими объектами. Особое внимание привлекает теория многозначной логики, открывающие следующие перспективы в исследовании различных систем:

- использование лингвистических переменных
- построение модели с несколькими состояниями работоспособности
- анализа чувствительности надежности или работоспособности системы к изменениям уровня работоспособности ее отдельных элементов

Основными этапами исследования работоспособности социально-экономической системы являются:

- Анализ структуры выбранной системы, её декомпозиция.
- Определение числа уровней работоспособности системы;
- Построение корректной математической модели исходного объекта;
- Разработка оценок работоспособности и алгоритмов их вычисления.
- Анализ влияния отдельных элементов системы на различные уровни ее работоспособности.

Структура выбранной системы определяется, исходя из её особенностей, и включает в себя факторы, оказывающие влияние на работоспособность СЭС.

Математическая модель представляет собой структурную функцию, описывающую соответствие между состояниями отдельных элементов системы и ее работоспособностью.

Анализ влияния отдельных элементов проводится на основе направленных логических производных и путем построения DD-кубов.

Таким образом, обозначен алгоритм определения работоспособности СЭС и контроля изменений уровня работоспособности в зависимости от изменения значений отдельных факторов СЭС.

### **3.2 Методика построения структурной функции и исследование ее чувствительности**

Методика построения структурной функции работоспособности любого объекта социо-технической системы заключается в следующем:

- Анализ структуры социо-технической системы и определение факторов, влияющих на работоспособность системы в целом.
- Определение значений выбранных факторов
- Построение структурной функции

На начальных этапах создания структурной функции необходимо понять, как работает организация, которую собираются исследовать. Зачастую, никто в организации не знает, как она работает в той мере подробности, которая необходима для создания СФ. Руководитель хорошо знает работу в целом, но не в состоянии вникнуть в детали работы каждого рядового сотрудника. Рядовой сотрудник хорошо знает, что творится на его рабочем месте, но плохо знает, как работают коллеги. Поэтому для описания работы такой системы необходимо построить модель. Такая модель должна быть адекватна предметной области, следовательно, она должна содержать в себе знания всех участников бизнес-процессов организации.

Для анализа бизнес-процессов, происходящих в СЭС, удобно использовать методологию IDEF0. В IDEF0 система представляется как совокупность взаимодействующих работ или функций. Такая чисто функциональная ориентация является принципиальной - функции системы анализируются независимо от объектов, которыми они оперируют. Это позволяет более четко смоделировать логику и взаимодействие процессов организации.

В основе стандарта IDEF0 лежит понятие функции. Функция – это управляемое действие над входными данными, результатом которого являются выходные данные, при этом используется некий механизм, посредством которого осуществляется это действие.

IDEF0-диаграммы строятся при помощи блоков. Каждый блок описывает какое-либо законченное действие (функцию). Четыре стороны блока имеют разное назначение. Слева отображаются входные данные, справа – выходные данные, сверху – управление, снизу – механизм.

Входные данные – исходные ресурсы для описываемой блоком функции (исходная информация, материалы).

Выходные данные – результирующие ресурсы, полученные в результате выполнения описываемой блоком функции (выходная информация, обработанные исходные материалы).

Управление – это то, что воздействует на процесс выполнения описываемой блоком функции и позволяет влиять на результат выполнения действия (средства управления, датчики, люди).

Механизм – это то, посредством чего осуществляется данное действие (станки, приборы, людские ресурсы, программное обеспечение).

Одной из наиболее важных особенностей методологии SADT является постепенное введение все больших уровней детализации по мере создания диаграмм, отображающих модель[9].

В функциональной схеме необходимо добиться такой степени детализации, при которой каждый из функциональных блоков можно было бы представить в виде функции или группы функций. Блоки будут декомпозированы до тех пор, пока реализация функции, записанной в блоке, не будет простой и не потребует реализации других функций.

Таким образом, диаграммы IDEF0 способствуют сбору данных о системе и бизнес-процессах, происходящих в ней и дают возможность установить факторы, влияющие на работоспособность как одного бизнес-процесса системы, так и всей системы целиком.

Как пример разработки диаграммы IDEF0 можно привести диаграмму процесса принятия кредитного решения.

При разработке программного продукта исследовалась функция «Заклучить кредитный договор». Данная функция имеет входные данные «данные о заемщике», выходные данные – «Кредитный договор/ отказ в заключении кредитного договора», управляется внешним воздействием «Заявление-анкета кредитополучателя», «Черный список заемщиков», «Типовой договор кредитования», «Созданная скоринговая модель», «Методика расчета кредитоспособности заемщика» и реализуется механизмами «Оператор», «Сотрудник службы безопасности», «ПК» и «Кредитный офицер» (см. рисунок 3.1).

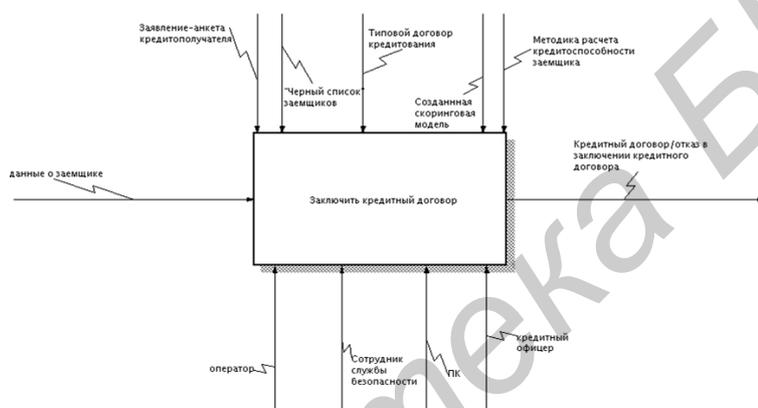


Рисунок 3.1 – Главная функция бизнес-процесса «Заклучить кредитный договор»

#### Модификация базы значений параметров.

Преобразование задаваемых действительными числами показателей в лингвистические предусматривает соотнесение одного, а чаще комбинации значений нескольких показателей, к  $m$ -уровневой шкале и называется дискретизацией. В общем случае дискретизация не приводит к снижению достоверности значений используемых в СППР атрибутов. Наоборот дискретизированные значения более адекватно оценивают экономические показатели и более устойчивы к методикам их измерения [16]. В результате дискретизации, например, показатель «ожидаемая прибыль проекта» будет принимать значения из множества конечных вариантов мощностью  $m$ : прибыль «очень высокая», «высокая», «средняя», «низкая» ( $m=4$ ).

Схематично иерархия действительных и лингвистических значений атрибутов представлена на рисунке 3.2 [17].

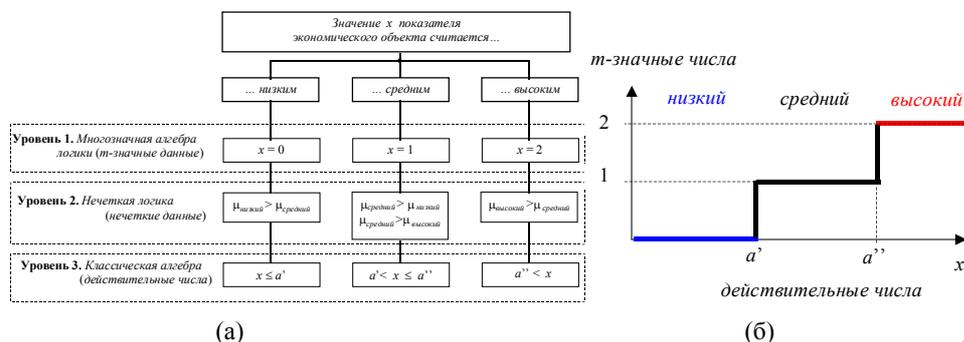


Рисунок 3.2 – Иерархия значений числовых и лингвистических переменных (а) и взаимосвязь действительных и лингвистических значений (б).

Анализ методов трансформации действительных значений в конечное число интервалов и соотнесение каждого интервала с дискретным значением достаточно хорошо исследована [20].

Особенно необходимо отметить тщательного выбора количества состояний работоспособности социо-технической системы. Данная результирующая переменная должна отражать потребность диагностики условий для безотказной либо для частично безотказной работы системы.

Одним из способов представления зависимости между переменными, описывающими социо-техническую систему, является использование структурной функции. В этом случае исходные показатели объекта интерпретируются как исходные атрибуты структурной функции, а результирующие – как ее результирующие атрибуты.

### Построение структурной функции

Структурная функция устанавливает однозначное соответствие между возможными состояниями работоспособности элементов и работоспособностью системы

$$\phi(x_1, x_2, \dots, x_n) = \phi(x): \{0, \dots, m-1\}^n \rightarrow \{0, \dots, m-1\}. \quad (3.1)$$

Пример структурной функции, описывающей оценку кредитоспособности заемщика, представлен на рисунке 3.3 [23], вектора значений данной структурной функции представлены в Приложении А.

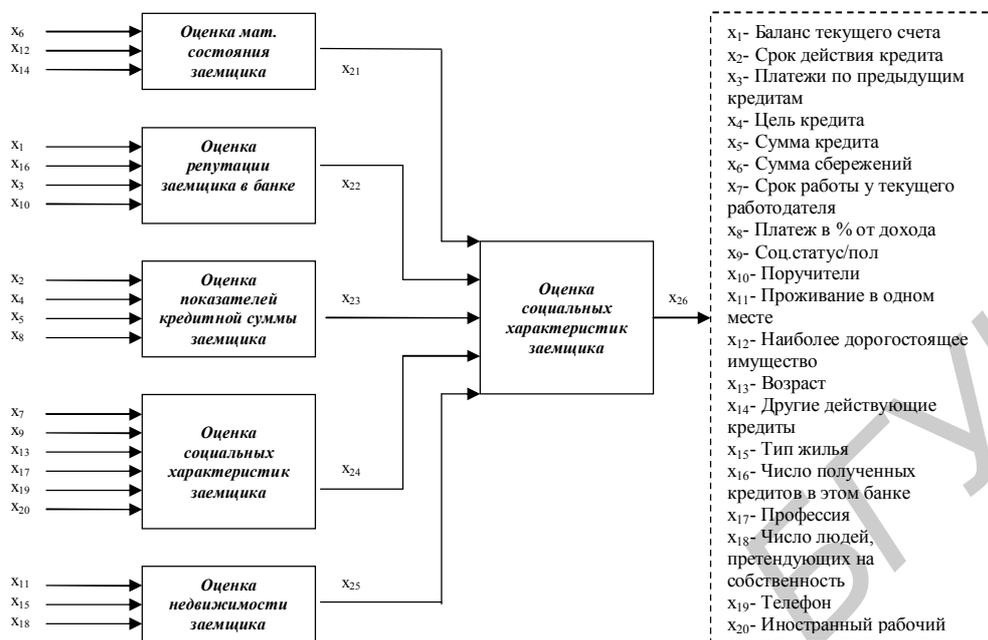


Рисунок 3.3 – Схематичное представление структурной функции, описывающей оценку кредитоспособности заемщика.

Для построения структурной функции используются база правил, разработанная экспертами. При исследовании экономического объекта, показатели которого представлены количественно и могут быть соотнесены с принятыми нормативами, помощь экспертов сводится к указанию порогового значения того или иного показателя. Другое дело, когда входящие переменные выражены качественно, и они не могут быть соотнесены с общепринятыми нормативами и нормами. Тогда необходимо участие экспертных групп для успешного, а главное корректного создания структурной функции.

Система, описывающая показатели реального экономического объекта достаточно громоздка, в результате структурная функция обладает значительной вычислительной сложностью. Одним из подходов к снижению этой сложности является декомпозиция структурной функции. Суть декомпозиции заключается в разбиении функции на подфункции с меньшим числом атрибутов.

Аргументы таких подфункций будем интерпретировать как промежуточные атрибуты, а сами подфункции – как компоненты структурной функции. В свою очередь компоненты этой функции так же являются структурной функцией меньшей сложности и в свою очередь могут быть декомпозируемы. Будем считать, что компонент структурной функции реализует функцию  $m$ -значной логики. Одним из способов представления функций многозначной логики является вектор значений, интерпретируемый как столбец таблицы истинности на упорядоченных наборах переменных.

Таким образом, методика построения структурной функции следующая:

- Анализ факторов, влияющих на корректную работу системы на основе диаграммы IEFD0.

- Выбор входных данных
- Определение значений независимых переменных
- Выбор исследуемого фактора и значений, которые он может принимать
- Переход к лингвистическим переменным
- Предварительная оценка вычислительной сложности структурной функции, просчет возможных вариантов декомпозиции структурной функции
- На основе экспертных правил расчёт столбцов таблицы истинности промежуточных и выходной переменных.

Полученная структурная функция может быть использована для определения работоспособности исследуемого процесса при заданных входных параметрах. Кроме того, она является основой для проведения анализа влияния изменения значений исходных атрибутов на результирующий атрибут.

### **Анализ чувствительности структурных функций**

Анализ чувствительности (sensitivity analysis) предполагает исследование изменения значения результирующих атрибутов при изменении значения одного или нескольких исходных атрибутов. Актуальность этого анализа объясняется тем, что его результаты позволяют:

- определить влияние значений отдельных показателей на поведение объекта;
- вычислить устойчивость к изменениям отдельных показателей;
- обнаружить начальные условия, обеспечивающие развитие или ухудшение результирующих показателей экономического объекта и ориентированных на получения оптимальной стратегии развития;
- определить при проектировании оптимальную структуру объекта с учетом разнообразных критериев.

Данная методика основывается на использовании логических производных функций  $m$ -значной логики. Однако, прямое использование полученных логических производных ограничивается задачами исследования чувствительности функции только относительно исходных атрибутов и требует значительных вычислительных затрат.

В работе [19] задачу исследования чувствительности структурной функции предлагается решать с помощью направленных логических производных. Суть этого метода заключается в том, что для функции  $m$ -значной логики вычисляется направленная логическая производная  $\partial f(j \rightarrow k) / \partial x_i(a \rightarrow b)$ . Данная производная фиксирует условия, при которых изменение значения аргумента  $x_i$  с  $a$  на  $b$  вызывает изменение значений функции с  $j$  на  $k$ . Из совокупности этих условий для определенных исходных данных рассчитываются условия влияния однократного изменения значения атрибута  $x_i$  на значение функции. Следует подчеркнуть, что вычислительные процессы построены на символических преобразованиях логических функций.

### **DD-кубы для $m$ -уровневых компонент и алгоритм их формирования**

Введем ряд обозначений. Изменение значения  $m$ -уровневого атрибута с  $p$  на  $q$  будем записывать в виде  $D_{p \rightarrow q}$  ( $p, q \in \{0, \dots, m-1\}$ ,  $p \neq q$ ). Например, запись  $D_{1 \rightarrow 2}$  фиксирует факт изменения значения атрибута с уровня логической 1 на уровень логической 2.

Теоретико-множественную операцию разности множеств  $M \setminus Y$ , где  $M = \{0, \dots, m-1\}$  и  $Y \subseteq M$  будем обозначать в виде  $U_Y$ . Например, запись  $U_{0,1,4}$  определяет множество  $\{2, 3, 5, \dots, m-1\}$ . Для обозначения множества  $\{0, \dots, m-1\}$  используется символ  $U$ .

Направленным D-кубом (DD-кубом) называется способ описания структурной функции в виде зависимости изменения значения результирующего атрибута от изменения значений его исходных атрибутов (одного или нескольких) [21].

Без потери общности рассмотрим структурную функцию с  $n$  исходными и одним результирующим атрибутами, принимающую  $m$  различных состояний. В этом случае DD-куб для компонента с  $n$  исходными атрибутами  $x_1, \dots, x_i, \dots, x_n$  и реализующего логическую функцию  $f(x_1, \dots, x_i, \dots, x_n)$ , представляет на рисунке 3.4

Исходные атрибуты компонента					Результирующий атрибут компонента
$x_1$	...	$x_i$	...	$x_n$	$f(x_1, \dots, x_i, \dots, x_n)$
$U_{Y_1}$	...	$D_{a \rightarrow b}$	...	$U_{Y_n}$	$D_{j \rightarrow k}$

Рисунок 3.4 – Схематичное представление DD-куба

На рисунке 3.4 символы  $U_{Y_1}, \dots, U_{Y_n}$  эквиваленты записи  $U_Y$  для атрибутов  $x_1, \dots, x_n$ .

Покажем связь направленной логической производной функции  $m$ -значной логики и DD-куба для  $m$ -уровневого компонента, реализующего эту функцию.

**Теорема 1.** Направленная логическая производная  $\partial X(j \rightarrow k) / \partial x_i(a \rightarrow b)$  функции  $m$ -значной логики определяет DD-кубы.

*Доказательство.* Направленная логическая производная  $\partial X(j \rightarrow k) / \partial x_i(a \rightarrow b)$  функции  $m$ -значной логики определяет, при каких значениях  $t_1, \dots, t_{i-1}, t_{i+1}, \dots, t_n$  исходных атрибутов  $x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n$  изменение значения  $i$ -го исходного атрибута  $x_i$  с  $a$  на  $b$  приводит к изменению значения результирующего атрибута с  $j$  на  $k$ . С другой стороны, в записи DD-куба для компонента присутствуют значения исходных атрибутов  $x_1, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_n$  этого компонента, при которых изменение исходного атрибута  $x_i$  (изменение  $D_{a \rightarrow b}$ ) приводит к изменению значения результирующего атрибута (изменение  $D_{j \rightarrow k}$ ). Следовательно, по результату вычисления направленной логической производной может быть построен DD-куб.

Теорема 1 является основой для построения алгоритма формирования DD-куба по результату вычисления направленной логической производной.

Алгоритм формирования DD-куба для  $m$ -уровневого компонента:

а) Вычислить отличную от нуля направленную логическую производную функции  $m$ -значной логики  $\partial X(j \rightarrow k) / \partial x_i(a \rightarrow b)$  для  $j, k, a, b = 0, \dots, m-1; a \neq b; i = 1, \dots, n$  [9].

б) Определить номера  $t$  компонент вектора  $\partial X(j \rightarrow k) / \partial x_i(a \rightarrow b)$ , равных  $(m-1)$ .

в) Записать DD-куб, используя параметры изменения  $j, k, a, b$  и  $m$ -значные представления номеров  $t = (t_1, \dots, t_{i-1}, t_{i+1}, \dots, t_n)_m$  по правилу:

1)  $i$ -му компоненту DD-куба соответствует запись  $D_{a \rightarrow b}$ , а последнему,  $(n+1)$ -му, -  $D_{j \rightarrow k}$ ;

2) поставить в соответствие  $l$ -му ( $l = 1, \dots, n, l \neq i$ ) компоненту DD-куба множество  $Y_l = \{0, \dots, m-1\} \setminus t_l$ ;

3) представить  $l$ -й компонент DD-куба символом  $U_{Y_l}$

Теорема 2. DD-куб, полученный при одновременной замене компонент  $D_{a \rightarrow b}$  на  $D_{b \rightarrow a}$  и  $D_{j \rightarrow k}$  на  $D_{k \rightarrow j}$ , эквивалентен исходному DD-кубу.

*Доказательство* следует из свойства симметрии направленной логической производной  $\partial X(j \rightarrow k) / \partial x_i(a \rightarrow b) = \partial X(k \rightarrow j) / \partial x_i(b \rightarrow a)$ .

*Пересечением* DD-кубов называется операция над DD-кубами для соединенных между собой компонент, при условии совпадения изменения значения результирующего атрибута одного из них и исходного атрибута другого.

Установим соответствие между приведенными определениями DD-кубов и исследуемыми изменениями для структурных функций, реализованных на основе  $m$ -уровневых компонент. Если требуется исследовать изменение атрибута, меняющее его значение с  $a$  на  $b$ , то необходимо: (а) определить исходное изменяемое значение атрибута - параметр  $a$ ; (б) задать изменение значения этого атрибута  $D_{a \rightarrow b}$ .

#### **Анализ чувствительности структурной функции, реализованной на $m$ -уровневых компонентах**

Изложим методику применения предложенного метода для анализа чувствительности структурной функции, реализованной на  $m$ -уровневых компонентах. При этом выделим следующие важные для практики аспекты: (а) применение метода для исследования чувствительности функций, реализованных в различном базисе  $m$ -уровневых компонент; (б) особенности применения метода при анализе чувствительности функций, принимающих  $m$ -устойчивых состояний.

*Направленной D-последовательностью* (DD-последовательностью) будем называть способ описания структурной функции с заданным в  $s$ -м атрибуте изменением. В этом описании используются DD-кубы для компонентов структурной функции, изменение и пути его транспортировки до результирующего атрибута.

Таким образом, алгоритм (модифицированный метод Спиллмана) для анализа чувствительности структурной функции, реализованной на  $m$ -уровневых компонентах состоит в следующем:

- Определить для исследуемого  $s$ -го атрибута изменения значений (начальная DD-последовательность).
- Определить  $m$ -уровневые компоненты, находящиеся на пути транспортировки изменения от места возникновения до результирующего атрибута
- Вычислить DD-кубы для этих компонентов.
- Выполнить операции пересечения начальной DD-последовательности с DD-кубами для компонентов, находящимися на пути транспортировки изменения (прямая фаза). Результатом является DD-последовательность.
- Доопределить DD-последовательности исходными значениями компонентов, не лежащих на пути активизации (обратная фаза).

Предложенная модификация известного метода Спиллмана и Су для анализа чувствительности структурной функции, реализованной на  $m$ -уровневых компонентах, дает возможность получить качественно новые функциональные характеристики метода:

- отсутствие ограничений на допустимый базис и значность  $m$ -уровневых компонент структурной функции;
- исследование чувствительности структурной функции к изменению значений исходных и промежуточных атрибутов;
- организация эффективной поддержки вычислений.

Таблица 3.2– DD-кубы для компонента X

Отличная от нуля направленная логическая производная	DD-куб			
	Исходные атрибуты			Результирующий
$x_{17} = f_X(x_{12}, x_{13}, x_{16}), X = [001\ 011\ 112\ 001\ 112\ 112\ 112\ 112\ 222]$				
$\partial X(0 \rightarrow 1) / \partial x_{12}(0 \rightarrow 1)$	$D_{0 \rightarrow 1}$	1	0	$D_{0 \rightarrow 1}$
$\partial X(1 \rightarrow 2) / \partial x_{12}(0 \rightarrow 1)$	$D_{0 \rightarrow 1}$	1	2	$D_{1 \rightarrow 2}$
$\partial X(0 \rightarrow 1) / \partial x_{12}(0 \rightarrow 2)$	$D_{0 \rightarrow 2}$	0	$U_1$	$D_{0 \rightarrow 1}$
	$D_{0 \rightarrow 2}$	1	0	$D_{0 \rightarrow 1}$
$\partial X(1 \rightarrow 2) / \partial x_{12}(0 \rightarrow 2)$	$D_{0 \rightarrow 2}$	$U_1$	2	$D_{1 \rightarrow 2}$
	$D_{0 \rightarrow 2}$	2	1	$D_{1 \rightarrow 2}$
$\partial X(0 \rightarrow 1) / \partial x_{13}(0 \rightarrow 1)$	$U_1$	$D_{0 \rightarrow 1}$	1	$D_{0 \rightarrow 1}$
$\partial X(1 \rightarrow 2) / \partial x_{13}(0 \rightarrow 1)$	1	$D_{0 \rightarrow 1}$	2	$D_{1 \rightarrow 2}$
$\partial X(0 \rightarrow 1) / \partial x_{13}(0 \rightarrow 2)$	0	$D_{0 \rightarrow 2}$	$U_1$	$D_{0 \rightarrow 2}$
	1	$D_{0 \rightarrow 2}$	$U_1$	$D_{0 \rightarrow 1}$
$\partial X(1 \rightarrow 2) / \partial x_{13}(0 \rightarrow 2)$	$U$	$D_{0 \rightarrow 2}$	2	$D_{1 \rightarrow 2}$
	2	$D_{0 \rightarrow 2}$	2	$D_{1 \rightarrow 2}$
$\partial X(0 \rightarrow 1) / \partial x_{13}(1 \rightarrow 2)$	0	$D_{1 \rightarrow 2}$	$U_0$	$D_{0 \rightarrow 1}$
$\partial X(1 \rightarrow 2) / \partial x_{13}(1 \rightarrow 2)$	$U_0$	$D_{1 \rightarrow 2}$	2	$D_{1 \rightarrow 2}$
	2	$D_{1 \rightarrow 2}$	$U_1$	$D_{1 \rightarrow 2}$
$\partial X(0 \rightarrow 1) / \partial x_{16}(0 \rightarrow 1)$	0	1	$D_{0 \rightarrow 1}$	$D_{0 \rightarrow 1}$
$\partial X(0 \rightarrow 1) / \partial x_{16}(1 \rightarrow 2)$	0	$U_2$	$D_{1 \rightarrow 2}$	$D_{0 \rightarrow 1}$
	1	0	$D_{1 \rightarrow 2}$	$D_{0 \rightarrow 1}$
$\partial X(1 \rightarrow 2) / \partial x_{16}(1 \rightarrow 2)$	$U_1$	2	$D_{1 \rightarrow 2}$	$D_{1 \rightarrow 2}$
	1	1	$D_{1 \rightarrow 2}$	$D_{1 \rightarrow 2}$
	2	$U_1$	$D_{1 \rightarrow 2}$	$D_{1 \rightarrow 2}$

Активизируем путь транспортировки  $D_{0 \rightarrow 1}$  изменения атрибута  $x_{13}$  к результирующему атрибуту и сформируем DD-последовательность (таблица 3.3).

Таблица 3.3– Построение DD-последовательности для атрибута  $x_{13}$  (оценка качества ПО) структурной схемы

Комп онент	Атрибуты структурной функции																Результ п. X
	Исходные											Промежуточные					
	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$	$x_9$	$x_{10}$	$x_{11}$	$x_{12}$	$x_{13}$	$x_{14}$	$x_{15}$		
*												1	$D_{0 \rightarrow 1}$			2	$D_{1 \rightarrow 2}$
A	$U_2$	$U_2$	0									1					
B				0	$U_1$												
C						$U$	2	$U$						2			
D									1	$U_1$	$U$					$U_1$	
E														2	$U_1$	2	
DD-последовательность																	
	$U_2$	$U_2$	0	0	$U_1$	$U$	2	$U$	1	$U_1$	$U$	1	$D_{0 \rightarrow 1}$	2	$U_1$	2	$D_{1 \rightarrow 2}$

Таким образом, условия для обнаружения изменения  $D_{0 \rightarrow 1}$  атрибута  $x_{13}$  структурной функции (рисунок 3.7) имеют вид:

$$U_2 U_2 0 0 U_1 U 2 U 1 U_1 U 1 U 1 U 1.$$

Дадим экономическую интерпретацию полученного результата.

Изменение качества программного кода ( $D_{0 \rightarrow 1}$ ) в априори заданных условиях: число отключений электроэнергии планируется равным 0 или 1, качество программного кода высокое, а проведенная подготовительная работа организаторов записи на ЦТ оценивается средне, приведет к ухудшению работоспособности системы при условии:

- качество связи будет находиться на среднем или низком уровне,
- качество используемого оборудования на среднем или низком уровне.
- качество БД и запросов к ней среднее,
- низкой квалификации ответственных,
- не зависит от организации формирования аудиторного фонда,
- время отклика службы поддержки на запросы не менее 4 часов.

При выполнении данных условий возможно изменение работоспособности всего процесса регистрации на ЦТ с частично работоспособного до неработоспособного.

Аналогично анализируется чувствительность результирующего атрибута (работоспособность системы) от изменения значений других исходных атрибутов.

### 3.3 Исследование работоспособности процесса организации олимпиады «Созвездие талантов»

On-line олимпиады «Созвездие талантов» проводится ежегодно зимой по следующим предметам: химия, биология и математика. Информирование об олимпиаде осуществляется с помощью веб-ресурса и рассылки. После on-line регистрации участников, в системе дистанционного обучения Moodle создаются курсы по соответствующим предметам, вводятся новые пользователи и тестовые задания, подготовленные преподавателями заранее. Затем всем участникам рассылается информация, необходимая для выполнения тестовых заданий. В назначенное время участники заходят в систему Moodle и выполняют тестовые задания. Далее проводится

подсчет результатов, анализ ответов, информирование участников о набранных баллах. Проблемы, с которыми столкнулись организаторы данной олимпиады, приведены ниже:

- Не все участники знали о сроках проведения олимпиады, а затем, не получив рассылки с информацией об авторизации, не приняли в олимпиаде участия.
- Не все участники получили письма с информацией, необходимой для авторизации и соответственно, не смогли принять участие в олимпиаде.
- Были неточности в ответах на тестовые задания.
- Из-за обрыва связи некоторые участники не смогли выполнить тестовые задания до конца, и их результаты оказались неверными.
- Плохо работала обратная связь с администраторами.

Данная олимпиада проводится для выявления талантливых учащихся старших классов с целью их дальнейшего участия в международном фестивале педагогов и учащихся «Созвездие талантов». Поэтому особенно важно, чтобы все зарегистрировавшиеся приняли в ней участие. Задачей данного исследования является выявление условий влияния факторов, характеризующих процесс организации и проведения олимпиады «Созвездие талантов».

### Структура процесса проведения on-line олимпиады «Созвездие талантов»

Для определения факторов, влияющих на работоспособность данного исследуемого процесса важно осознать структуру организационной деятельности по проведению on-line олимпиады «Созвездие талантов». Для этого необходимо представить бизнес-процессы, протекающие при организации данного мероприятия в форме диаграммы IDEF0. На рисунке 3.12 приведена главная диаграмма процесса проведения олимпиады.

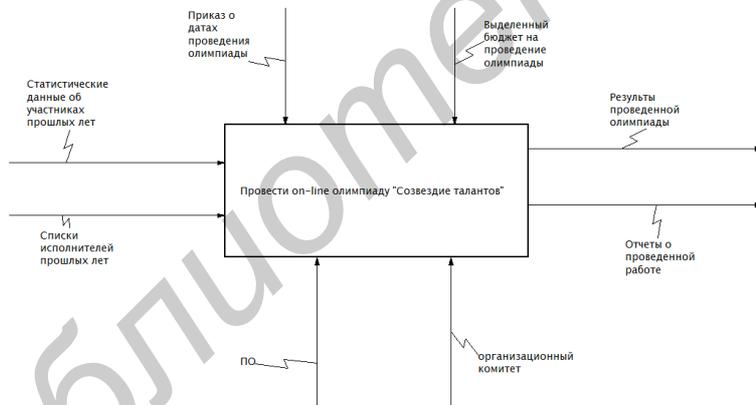


Рисунок 3.5 – Главная диаграмма процесса проведения олимпиады

Дальнейшее исследование структуры организационного процесса проводится за счет детализации главной диаграммы (рисунки 3.5, 3.6, 3.7).

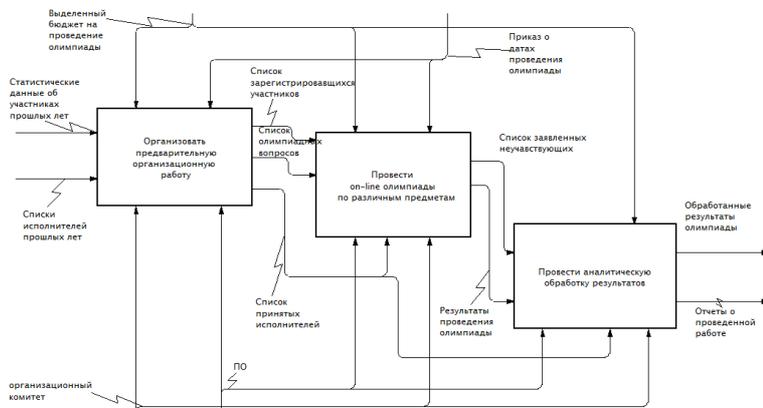


Рисунок 3.5 – Детализация диаграммы «Провести on-line олимпиаду»

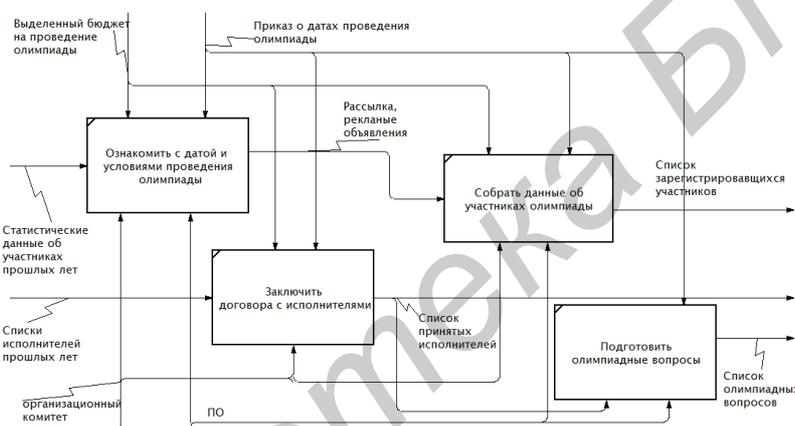


Рисунок 3.6 – Детализация диаграммы «Организовать предварительную подготовительную работу»

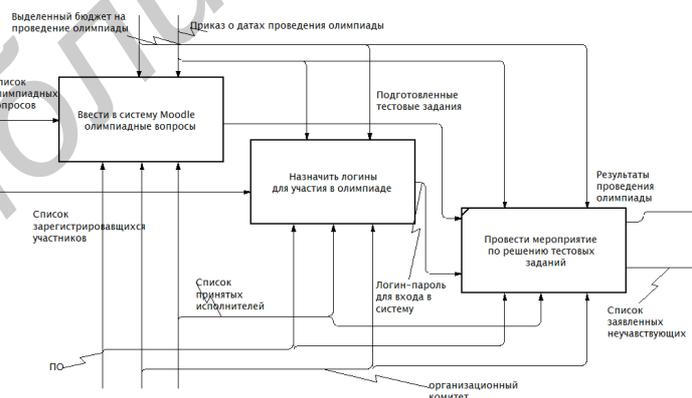


Рисунок 3.7 – Детализация диаграммы «Провести on-line олимпиады по различным предметам»

В ходе исследования структуры процесса проведения on-line олимпиады были выявлены факторы, влияющие на организацию и проведение олимпиады «Созвездие талантов» (таблица 3.4).

Таблица 3.4– Входные переменные процесса проведения олимпиады «Созвездие талантов»

Наименование атрибута	Значение атрибута и его описание		
	0	1	1
Информированность участников об олимпиаде ( $x_1$ )	Низкая	Средняя	Высокая
Качество обратной связи ( $x_2$ )	Низкое (связь отсутствует)	Среднее	Высокое
Корректность вводимой информации об участниках ( $x_3$ )	Информация некорректна	Информация корректна, но есть небольшие ошибки	Информация корректна
Корректность тестовых заданий ( $x_4$ )	Тестовые задания некорректны	Тестовые задания корректны, но имеются неточности	Тестовые задания корректны
Квалификация операторов ( $x_5$ )	Низкая	Средняя	Высокая
Количество сбоев электроэнергии ( $x_6$ )	4 и более	1-3	0
Количество сбоев сети ( $x_7$ )	5 и более	1-5	0
Количество сбоев аппаратного обеспечения ( $x_8$ )	3 и более	1-2	0
Количество сбоев ПО ( $x_9$ )	5 и более	1-5	0

#### **Разработка структурной функции процесса проведения олимпиады «Созвездие талантов»**

Основными этапами построения структурной функции работоспособности процесса организации и проведения олимпиады «Созвездие талантов» являются:

- Определение переменных, включаемых в математическую модель.
- Группировка входящих переменных и нахождение промежуточных переменных на основе экспертной базы правил.
- На основе полученных промежуточных переменных расчет структурной функции.

На рисунке 3.8 представлена графическая интерпретация структурной функции проведения олимпиады «Созвездие талантов».

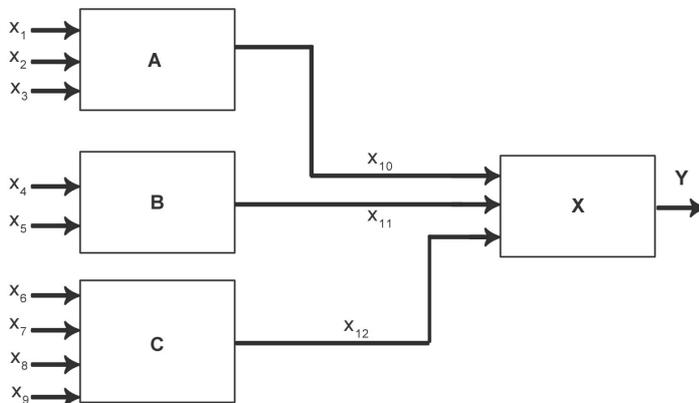


Рисунок 3.8 – Схема структурной функции процесса проведения on-line олимпиады «Созвездие талантов»

Группируются переменные по признакам принадлежности к классам: оборудование, ПО, организация. Это проводится для того, чтобы уменьшить вычислительную сложность структурной функции. После разбиения переменных на группы, производится расчет промежуточных переменных на основе экспертной базы правил. Результаты расчетов представлены на рисунке 3.9.

$x_1$	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	
$x_2$	0	0	0	1	2	1	1	2	2	0	0	0	1	1	1	2	2	2	0	0	0	1	1	1	2	2	2
$x_3$	0	1	2	0	0	1	2	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2
$X_A=x_{12}$	0	0	1	0	1	1	1	1	1	0	0	1	1	1	1	1	1	0	1	2	1	1	2	1	2	2	
$x_4$	0	0	0	1	1	1	2	2	2																		
$x_5$	0	1	2	1	1	2	0	1	2																		
$X_B=x_{13}$	0	1	2	1	1	2	2	2	2																		
$x_6$	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	
$x_7$	0	0	0	1	2	1	1	2	2	0	0	0	1	1	1	2	2	2	0	0	0	1	1	1	2	2	
$x_8$	0	1	2	0	0	1	2	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2
$X_C=x_{14}$	0	0	1	0	1	1	1	1	1	0	0	1	1	1	1	1	1	0	1	2	1	1	2	1	2	2	

Рисунок 3.9 – Расчет промежуточных переменных

На базе найденных промежуточных переменных рассчитывается вектор-значений X. Результаты расчетов приведены на рисунке 3.10.

$x_{10}$	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	
$x_{11}$	0	0	0	1	2	1	1	2	2	0	0	0	1	1	1	2	2	2	0	0	0	1	1	1	2	2	
$x_{12}$	0	1	2	0	0	1	2	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2
X	0	0	1	0	1	1	1	1	2	0	0	1	1	1	2	1	1	2	1	1	2	1	1	2	2	2	

Рисунок 3.10 – Расчет результирующей переменной работоспособности организации олимпиады «Созвездие талантов»

Таким образом, результатом построения структурной функции организации олимпиады «Созвездие талантов» стал вектор-столбец X, интерпретируемый как столбец

таблицы истинности на упорядоченных наборах переменных.  
 $X=[001\ 011\ 112\ 001\ 112\ 112\ 112\ 222]$ .

Разработанная структурная функция может быть использована для определения работоспособности исследуемого процесса при заданных входных параметрах. Например, когда известно, что информированность участников олимпиады низкая при остальных высоких значениях входных данных. С помощью структурной функции можно обосновать выделения дополнительных средств на рекламу и дополнительную рассылку информации школам и гимназиям РБ.

Библиотека БГУИР

## **4 Представление геометрических объектов с помощью булевых формул в задачах проектирования топологии СБИС**

### **4.1 Введение и формулировка решаемых задач**

В современных условиях уровень промышленного развития любой страны определяется степенью развития наукоемких, передовых в технологическом плане отраслей, в частности, электронной промышленности и ее базовой отрасли – микроэлектроники.

Ускоренный рост развития электроники в большей степени определяется развитием ведущего направления этой отрасли – производства сверхбольших интегральных схем (СБИС), совмещающих тысячи и сотни тысяч электронных и оптических компонентов, формирующих суперскоростные устройства цифровой обработки информации. Все это требует ультратонких технологий создания в полупроводниковых материалах компонентов субмикронных размеров, отвечающих строгим расчетам физиков, химиков, схемотехников, образующих в итоге электронную схему. Расчет, проектирование и производство интегральных схем (ИС) являются чрезвычайно сложными процессами, для проведения которых необходимо специальное производственное оборудование, реализующее порядка 1000 физико-химических операций внутри полупроводниковой пластины и на ее поверхности с использованием практически всех известных науке элементов. Современные заводы по производству полупроводниковых компонентов способны обрабатывать пластины диаметром 300 мм, использовать технологию КМОП и БиКМОП с размером компонент 0,13 микрон и менее.

В современных условиях сложнейшие наукоемкие технологии производства микроэлектронных схем и устройств способны обеспечивать эффективность бизнеса, его конкурентоспособность и привлекательность, если эти технологии будут интегрированы в рамках так называемых CALS-технологий. CALS (Continuous Acquisition and Life-cycle Support – непрерывная информационная поддержка жизненного цикла продукции) – это идеология создания единой информационной среды для процессов проектирования, производства, испытаний, поставки и эксплуатации продукции. Системность информационного подхода заключается в охвате всех стадий жизненного цикла продукции от замысла до утилизации на основе стандартизации методов представления данных на каждой из этих стадий. Применение CALS-технологий позволяет эффективно, в едином ключе решать проблемы обеспечения качества выпускаемой продукции и ее конкурентоспособности, поскольку используемое электронное описание процессов разработки, производства, монтажа и т.д. полностью соответствует требованиям международных стандартов серии ИСО 9000, реализация которых гарантирует выпуск высококачественной продукции.

В микроэлектронике одной из важных составных частей технологического процесса изготовления СБИС с использованием фотошаблонов является подготовка с помощью САПР СБИС топологии послойных масок для изготовления кристаллов интегральных схем [22]. Разработанная топология используется в фотонаборных установках микроэлектроники для создания изображений, которые проецируются на некоторый материал (на пленку или стекло для получения фотошаблонов, на полупроводниковую пластину для получения ИС). Под генерацией изображения понимается процесс создания изображения посредством генератора изображения, являющегося составной частью фотонаборной установки. Исходная топология, полученная с помощью САПР СБИС, представлена в векторной форме. В процессе генерации для каждого объекта топологии

создается изображение, которое получается путем пошагового экспонирования отдельных областей объекта в виде прямоугольников, объединение которых дает изображение объекта. Для получения изображения прямоугольника на экспонируемой поверхности используется диафрагма прямоугольной формы с регулируемыми размерами и ориентацией. Перемещение изображения прямоугольника по экспонируемой поверхности осуществляется с помощью координатного стола.

В настоящем разделе приводятся результаты решения некоторых частных задач, связанных с подготовкой топологической информации для микрофотонаборных генераторов изображений, а именно – описаны методы представления топологических объектов (многоугольников) в виде булевых формул. Ранее для описания геометрических объектов в системах автоматизированного проектирования топологии ИС [22,23] использовались методы аналитической геометрии, векторной алгебры, теории матриц [22-26]. Альтернативные способы описания, появившиеся в последнее время, основаны на использовании булевых формул в скобочной [27] и в дизъюнктивной нормальной [28-31] формах. Это направление исследований открывает новые возможности для решения широкого круга оптимизационных задач топологического проектирования ИС путем использования развитого аппарата булевой алгебры.

## 4.2 Основные понятия и определения

### 4.2.1 Граница многоугольника, ориентация сторон, предикатные переменные

Многоугольник на плоскости задается своей *границей* – замкнутой не пересекающейся ломаной линией, состоящей из отрезков прямых или *сторон* многоугольника. Эту границу можно определить последовательностью *угловых точек* или *вершин* многоугольника  $p_1, p_2, \dots, p_n$ , получаемых при последовательном обходе его вдоль границы (рисунок 4.1, где  $n = 11$ ).

Так как каждая пара соседних угловых точек ограничивает соответствующую сторону многоугольника, то его границу можно задать также последовательностью сторон многоугольника:  $s_1, s_2, \dots, s_n$ , где  $s_1 = (p_1, p_2)$ ,  $s_2 = (p_2, p_3)$ , ...,  $s_n = (p_n, p_1)$ .

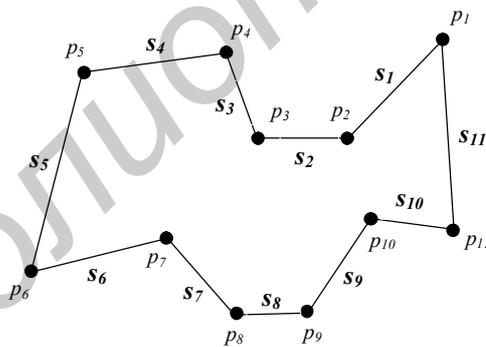


Рисунок 4.1 – Вершины и стороны многоугольника

Вершина  $p_1$ , которая служит начальной угловой точкой для последовательного обозначения отрезков, образующих границу многоугольника, называется *начальной*. В качестве начальной будем выбирать вершину, наиболее удаленную от начала координат.

Каждой стороне  $s_i$  многоугольника поставим в соответствие *ориентированную* прямую  $v_i$ , содержащую точки  $p_i$  и  $p_{i+1}$ . Будем считать, что она ориентирована от  $p_i$  к  $p_{i+1}$ .

Рассмотрим некоторую произвольную точку плоскости  $p$ , заданную парой декартовых координат  $(x, y)$ . Будем считать, что точка  $p$  расположена *слева от прямой*  $v_i$ , если она принадлежит полуплоскости, расположенной слева от ориентированной прямой  $v_i$ , или лежит на прямой  $v_i$ . Все возможные варианты левостороннего расположения точки  $p$  относительно ориентированной прямой  $v_i$  представлены на рисунке 4.2 (последние два варианта соответствуют случаю, когда прямая  $v_i$  параллельна координатной оси  $X$ ).

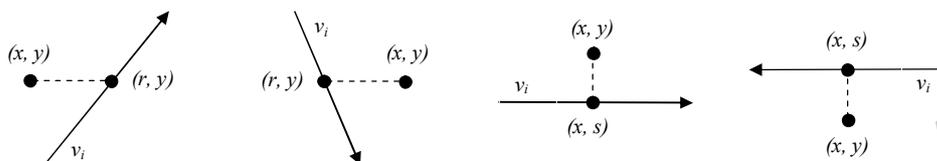


Рисунок 4.2– Варианты левостороннего расположения точки относительно ориентированной прямой

Как и в работе [4.6], будем обозначать отрезки ломаной буквами  $a, b, c, \dots$ , границу многоугольника как  $abc\dots$ , а полуплоскости, расположенные слева от соответствующих ориентированных прямых – буквами  $A, B, C, \dots$  (считая, что каждая из этих полуплоскостей включает в себя еще и все точки порождающей ее ориентированной прямой). Введем также предикаты  $a, b, c, \dots$  для описания положения некоторой точки  $p$  на плоскости, полагая, что  $a(p) = 1$ , если и только если  $p \in A$ .

#### 4.2.2 Выпуклые и вогнутые углы многоугольника

Рассмотрим пару соседних отрезков  $a$  и  $b$ , задающую некоторый угол многоугольника. Если этот угол меньше 180 градусов, будем называть его *выпуклым*, иначе – *вогнутым*.

Пара ориентированных прямых, соответствующая отрезкам  $a$  и  $b$ , ограничивает участок плоскости, который можно представить формулой  $A \cap B$  в первом случае и формулой  $A \cup B$  – во втором случае (рисунок 4.3).

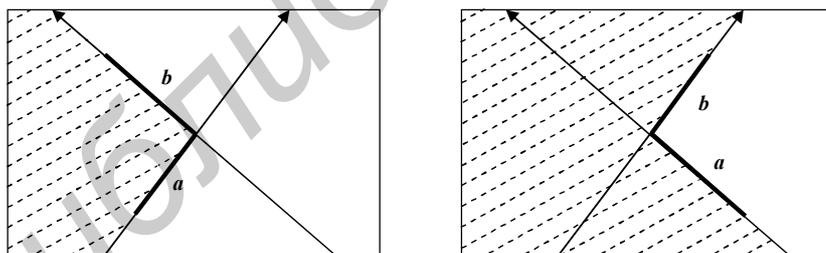


Рисунок 4.3 – Участки плоскости, ограниченные выпуклым (слева) и вогнутым (справа) углами

Если все углы многоугольника будут выпуклыми, то такой многоугольник называется *выпуклым*. Например, выпуклый многоугольник с границей  $abcde$  (рис. 4) будет занимать участок плоскости:  $A \cap B \cap C \cap D \cap E$ , а его булева формула будет иметь вид:  $F = abcde$ .

Каждой стороне многоугольника припишем *ранг*, равный числу тех смежных с ней углов, которые являются вогнутыми.

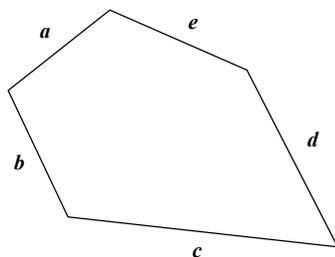


Рисунок 4.4 – Выпуклый многоугольник

В общем случае многоугольник может содержать как выпуклые, так и вогнутые углы. Последовательность следующих друг за другом выпуклых углов образует *выпуклый фрагмент* границы многоугольника, последовательность вогнутых углов – *вогнутый фрагмент*. При этом выпуклый фрагмент ограничивает участок плоскости, который представляется *пересечением* полуплоскостей, ограниченных отдельными ориентированными прямыми (рисунок 4.5, а). Аналогично, вогнутый фрагмент представляется *объединением* таких полуплоскостей (рисунок 4.5, б). Выпуклый и вогнутый фрагменты называются *полными*, если их уже нельзя расширить добавлением соседних элементов границы.

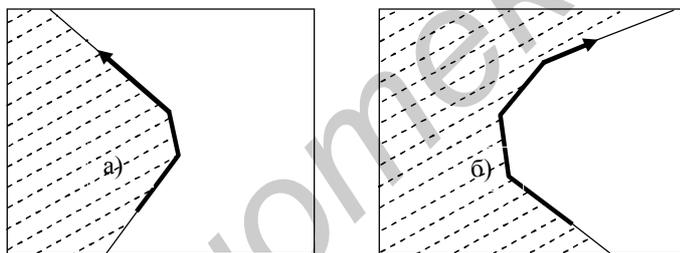


Рисунок 4.5 – Фрагменты границы многоугольника: а) выпуклый, б) вогнутый

Таким образом, граница многоугольника будет состоять из чередующихся выпуклых и вогнутых фрагментов. Граница выпуклого многоугольника представляет собой замкнутый выпуклый фрагмент.

### 4.3 Способ поиска множества крайних вершин многоугольника

#### 4.3.1 Каноническая булева формула многоугольника

Вершина многоугольника называется *крайней*, если через нее можно провести прямую, не пересекающуюся ни с одним из отрезков границы за исключением двух соседних отрезков, на стыке которых эта вершина располагается. Например, на рисунке 4.6 таких вершин восемь, и они отмечены жирными кружками. Начальная вершина  $p_1$ , как следует из указанного выше способа ее выбора, всегда принадлежит множеству крайних вершин многоугольника.

Основываясь на понятиях предикатной переменной и крайней вершины, в работе [4.6] описан метод построения канонической булевой формулы  $F$ , представляющей многоугольник и обладающей следующим свойством: если выполнить подстановку предикатных координат произвольной точки плоскости, то формула  $F$  примет значение 1 в случае, когда точка принадлежит данному многоугольнику, и значение 0 – в противном случае.

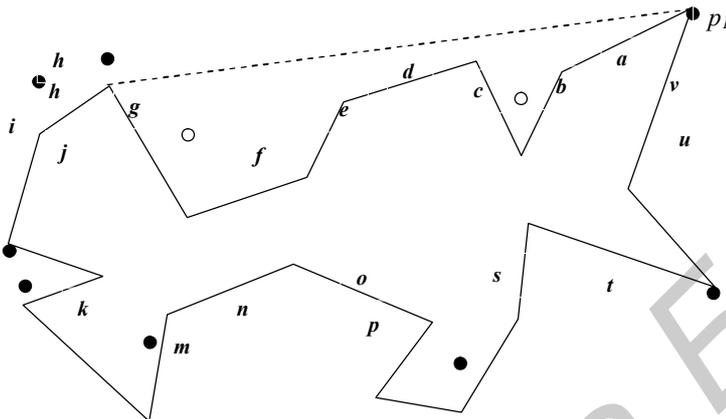


Рисунок 4.6 – Крайние вершины многоугольника

Необходимо, однако, отметить, что в этом методе построения формулы  $F$  многократно используется операция поиска множества крайних вершин (сначала для исходного многоугольника, затем – для более простых многоугольников, получаемых в процессе формирования формулы  $F$ ). Поскольку указанная задача является нетривиальной, однако, как частная задача, не рассматривается в работе [27], то ниже предлагается способ ее решения, описанный в [32]. Тем самым метод построения канонической булевой формулы  $F$  многоугольника может быть легко доведен до формы алгоритма и, далее, переведен в форму программ на каком-либо языке программирования.

Метод нахождения булевой формулы для любого многоугольника, изложенный в работе [27], проиллюстрируем на примере многоугольника, изображенного на рисунке 4.6. Метод включает в себя следующие действия.

- В многоугольнике находятся все крайние вершины (на рисунке они отмечены жирными кружками).

- Граница многоугольника  $abcdefghijklmnopqrstuv$  разбивается крайними вершинами на фрагменты первого ранга, обозначаемые последовательностями образующих их отрезков:  $abcdefg$ ,  $h$ ,  $i$ ,  $jk$ ,  $l$ ,  $mnpq$ ,  $rst$ ,  $uv$ .

- Искомая формула  $F$  строится как конъюнкция формул, описывающих фрагменты первого ранга:

$$F = F(abcdefg) \wedge F(h) \wedge F(i) \wedge F(jk) \wedge F(l) \wedge F(mnpq) \wedge F(rst) \wedge F(uv).$$

После замены фрагментов, содержащих лишь один отрезок, символом соответствующего предиката, получаем:

$$F = F(abcdefg) \wedge h \wedge i \wedge F(jk) \wedge l \wedge F(mnpq) \wedge F(rst) \wedge F(uv).$$

а) Фрагменты первого ранга, содержащие более одного отрезка, пересекаются на фрагменты второго ранга следующей процедурой, иллюстрируемой на примере фрагмента  $abcdefg$ :

1) две концевые точки фрагмента соединяются условным отрезком (показанным на рисунке пунктирной линией), в результате чего получается многоугольник, называемый *замкнутым фрагментом*;

2) в замкнутом фрагменте отыскиваются крайние вершины (к ним не относятся концевые точки). Они отмечены на рисунке простыми кружками и делят рассматриваемый фрагмент на фрагменты второго ранга  $ab$ ,  $cdef$  и  $g$ . Дизъюнкция формул этих фрагментов представляет формулу фрагмента в целом:

$$F(abcdefg) = F(ab) \vee F(cdef) \vee F(g).$$

Аналогично делятся фрагменты первого ранга  $jk$ ,  $mnpq$ ,  $rst$  и  $uv$ . В результате этого шага получаем:

$$F = (F(ab) \vee F(cdef) \vee g) \wedge h \wedge i \wedge (j \vee k) \wedge l \wedge (m \vee n \vee o \vee F(pq)) \wedge (F(rs) \vee t) \wedge (u \vee v).$$

Полученные фрагменты второго ранга, содержащие более одного отрезка, разбиваются на фрагменты *третьего ранга*, и т. д.

б) Эта итеративная процедура повторяется, пока все фрагменты не разобьются на отдельные отрезки. При этом все фрагменты нечетного ранга являются вогнутыми, а фрагменты четного ранга – выпуклыми. Каждый раз формула фрагмента нечетного ранга представляется дизъюнкцией формул выпуклых фрагментов, получаемых при разбиении, а формула фрагмента четного ранга – конъюнкцией формул соответствующих вогнутым фрагментам.

Для рассматриваемого многоугольника получаем в конечном результате следующую каноническую булеву формулу:

$$F = (ab \vee cd \vee ef) \vee g \wedge h \wedge i \wedge (j \vee k) \wedge l \wedge (m \vee n \vee o \vee pq) \wedge (rs \vee t) \wedge (u \vee v).$$

#### 4.3.2 Поиск множества крайних вершин многоугольника

Суть предлагаемого способа поиска множества крайних вершин многоугольника заключается в том, что крайние вершины отыскиваются последовательно, одна за другой, в том порядке, в котором производится обход границы многоугольника. При этом, чтобы найти очередную крайнюю вершину  $p_j$ , необходимо лишь знать, какая крайняя вершина  $p_i$  была найдена непосредственно перед этим (в самом начале в качестве  $p_i$  берется начальная вершина  $p_1$ , с которой начинается нумерация всех вершин многоугольника).

Процедура поиска очередной крайней вершины  $p_j$  заключается в следующем. Строим ориентированную прямую  $L_{i,i+1}$ , содержащую вершины  $p_i$  и  $p_{i+1}$ . Прямую ориентируем в направлении от  $p_i$  к  $p_{i+1}$ . Полу плоскость, расположенную слева от прямой  $L_{i,i+1}$ , обозначим через  $H_{i,i+1}$ . Проверяем, все ли вершины многоугольника содержат полу плоскость  $H_{i,i+1}$ . Если условие выполняется, то вершина  $p_{i+1}$  является крайней вершиной многоугольника и поиск заканчивается.

В противном случае строим новую ориентированную прямую  $L_{i,i+2}$ , проходящую через вершины  $p_i$  и  $p_{i+2}$ , и проверяем, все ли вершины многоугольника содержат полу плоскость  $H_{i,i+2}$ . Если условие выполняется, то вершина  $p_{i+2}$  является крайней вершиной многоугольника и поиск заканчивается. Иначе строим следующую ориентированную прямую  $L_{i,i+3}$  и так далее до тех пор, пока очередная полу плоскость  $H_{i,i+k}$  не будет содержать в себе все вершины многоугольника. В результате будет найдена очередная (в порядке обхода границы) крайняя вершина  $p_j$ , где  $j = i + k$ .

После нахождения крайней вершины  $p_j$ , эта вершина используется аналогичным образом для поиска следующей по порядку крайней вершины и так далее. Процесс поиска крайних вершин заканчивается тогда, когда очередной найденной крайней вершиной оказывается начальная вершина  $p_1$ , основываясь на которой поиск и был начат.

Предложенный способ поиска множества крайних вершин проиллюстрируем на примере многоугольника, изображенного на рисунке 4.7.

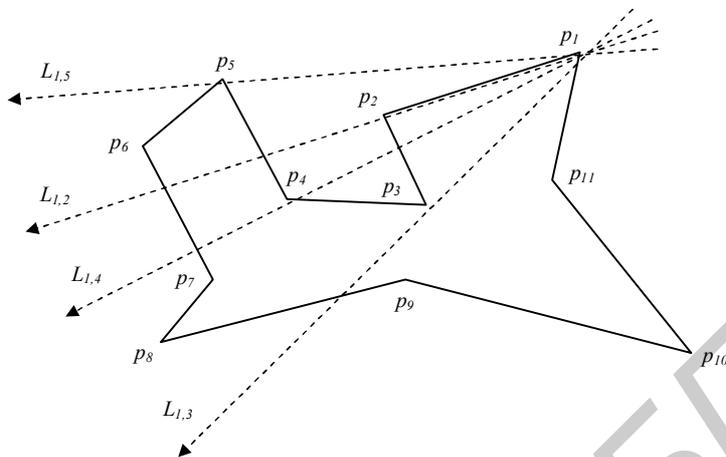


Рисунок 4.7 – Ориентированные прямые, необходимые для нахождения крайней вершины многоугольника

Поиск второй крайней вершины (первой крайней вершиной является начальная вершина  $p_1$ ) начинается с рассмотрения вершины  $p_2$  и проведения через  $p_1$  и  $p_2$  ориентированной прямой  $L_{1,2}$ . Полуплоскость  $H_{1,2}$ , расположенная слева от прямой  $L_{1,2}$ , содержит не все вершины многоугольника (что видно из рисунка), поэтому рассматривается следующая (в порядке обхода границы) вершина  $p_3$  и через  $p_1$  и  $p_3$  проводится ориентированная прямая  $L_{1,3}$ . Полуплоскость  $H_{1,3}$ , расположенная слева от прямой  $L_{1,3}$ , также содержит не все вершины многоугольника, поэтому рассматривается следующая вершина  $p_4$  и проводится ориентированная прямая  $L_{1,4}$ . Но и полуплоскости  $H_{1,4}$  принадлежат не все вершины многоугольника, поэтому рассматривается вершина  $p_5$  и проводится ориентированная прямая  $L_{1,5}$ . Так как полуплоскость  $H_{1,5}$  содержит все вершины многоугольника, то вершина  $p_5$  есть искомая крайняя вершина.

Далее аналогичным образом отыскивается третья крайняя вершина (причем для ее поиска проводятся ориентированные прямые, проходящие через вершину  $p_5$ ) и так далее. В результате будут найдены все крайние вершины многоугольника:  $p_1, p_5, p_6, p_8, p_{10}$ .

#### 4.4 Простой метод нахождения булевой формулы многоугольника в ДНФ

В работах [4.6,4.7] показано, что булеву формулу любого многоугольника можно представить в ДНФ, которая соответствует покрытию многоугольника его выпуклыми компонентами.

Однако, для алгоритма из [28], как и для многих алгоритмов вычислительной геометрии, характерно наличие проблемы вычислительной точности. Она заключается в том, что хотя теоретически строго обоснована правильность работы алгоритма, однако на практике встречаются задачи, для которых алгоритм не работает или работает некорректно в силу ограниченной точности представления вещественных чисел в памяти компьютера и потери точности в промежуточных вычислениях. Практика разработки САПР в области проектирования топологии интегральных схем показала, что такой

эффект возникает, в частности, если в алгоритме используется операция разбиения стороны топологического объекта на отрезки, которые далее используются в качестве операндов.

Ниже предлагается достаточно простой и приемлемый на практике метод нахождения булевой формулы многоугольника в ДНФ, который основан на использовании двух простых операций: вычисления угла между прямыми и проверки факта принадлежности вершин многоугольника его выпуклой компоненте, что снимает указанную выше проблему вычислительной точности. Суть метода заключается в последовательном отсечении от него выпуклых компонент до тех пор, пока непокрытый выпуклыми компонентами остаток многоугольника не станет выпуклой компонентой. После этого все полученные выпуклые компоненты заменяются представляющими их конъюнкциями соответствующих предикатных переменных, а дизъюнкция этих конъюнкций даст искомую булеву формулу многоугольника в ДНФ.

Нахождение очередной выпуклой компоненты включает в себя следующие действия. Нахождение очередной выпуклой компоненты включает в себя следующие действия.

- Отыскивается начальная вершина непокрытого остатка многоугольника.
- Находится полный выпуклый фрагмент границы непокрытого остатка, включающий в себя начальную вершину.
- Из непокрытого остатка выделяется компонента, представляющая собой многоугольник, ограниченный найденным выпуклым фрагментом и отрезком прямой, соединяющим последнюю и первую вершины этого фрагмента.
- Выполняется проверка, являются ли выпуклыми углы, прилегающие к первой и последней вершинам выпуклого фрагмента, на основе которого построена компонента. Если оба этих угла оказываются выпуклыми (т.е. компонента представляет собой выпуклый многоугольник), то выполняется переход к следующему пункту, иначе операция выделения компоненты отменяется, а выпуклый фрагмент корректируется путем удаления последнего элемента его границы. Затем из непокрытого остатка вновь выделяется компонента, ограниченная скорректированным выпуклым фрагментом и отрезком прямой, соединяющим последнюю и первую вершины этого фрагмента. Далее выполняется переход к началу данного пункта.
- Выполняется проверка, содержится ли внутри выпуклой компоненты хотя бы одна из тех вершин исходного многоугольника, которые не принадлежат этой компоненте. Если таких вершин нет, то выполняется переход к следующему пункту, иначе операция выделения компоненты отменяется, а лежащий в основе ее построения выпуклый фрагмент корректируется путем удаления последнего элемента его границы. Затем из непокрытого остатка вновь выделяется выпуклая компонента, ограниченная скорректированным выпуклым фрагментом и отрезком прямой, соединяющим последнюю и первую вершины этого фрагмента. Далее выполняется переход к началу данного пункта.
- Выполняется альтернативный расчет путем повторения всех описанных выше операций с тем лишь отличием, что при корректировке выпуклого фрагмента удаляется не последний, а первый элемент его границы.
- Из двух альтернативных решений выбирается то, которому соответствует выпуклый компонент с большим числом сторон.
- Непокрытый остаток корректируется путем отсечения от него части, покрываемой найденным выпуклым компонентом.

*Замечание.* Иногда может возникнуть ситуация, когда из непокрытого остатка не удастся выделить выпуклую компоненту, удовлетворяющую описанным выше критериям. В этом случае расчет нужно повторить с тем лишь отличием, что вместо начальной вершины непокрытого остатка используется любая из его крайних вершин.

Изложенный метод проиллюстрируем на примере многоугольника, изображенного на рисунке 4.8, а.

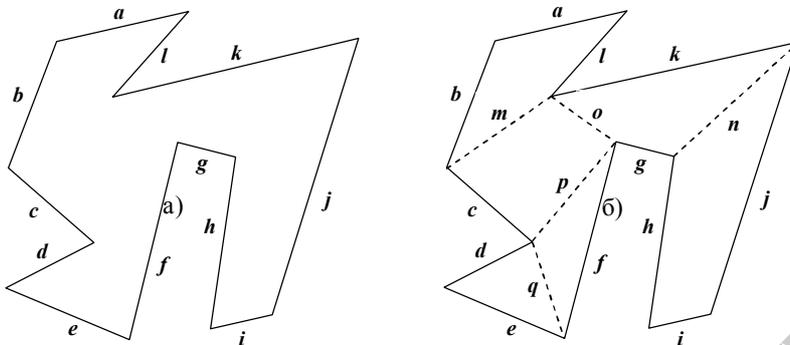


Рисунок 4.8 – а) многоугольник, б) его выпуклые компоненты

Поиск первой выпуклой компоненты начинается с нахождения начальной вершины многоугольника. Ей будет вершина, лежащая на стыке сторон  $a$  и  $l$ . Полным выпуклым фрагментом, включающим в себя эту вершину, будет фрагмент  $labc$ . Альтернативные решения порождаются его подмножествами  $lab$  и  $bc$ . В итоге отыскивается выпуклый компонент с границей  $labm$  (рисунок 4.9, б), который будет первым элементом покрытия исходного многоугольника выпуклыми компонентами.

Далее те же действия выполняются применительно к непокрытому остатку с границей  $kmcdefghij$ . Его начальной вершиной будет вершина, лежащая на стыке сторон  $k$  и  $j$ , а полным выпуклым фрагментом – фрагмент  $hijkmc$ . Альтернативные решения порождаются его подмножествами  $hij$  и  $cm$ . В итоге отыскивается выпуклый компонент с границей  $nhij$ , который будет вторым элементом искомого покрытия.

Снова выполняются аналогичные действия по отношению к непокрытому остатку с границей  $kmcdefgn$ . В итоге отыскивается третий элемент покрытия с границей  $kogn$ .

Четвертым элементом покрытия будет элемент с границей  $mcpo$ , пятым – элемент с границей  $deq$ .

Так как оставшийся непокрытый остаток с границей  $pqf$  представляет собой выпуклый многоугольник, то он становится последним, шестым элементом искомого покрытия.

В результате булева формула многоугольника в ДНФ будет иметь следующий вид:

$$F = abml\backslash nhij\vee kogn\vee mcp\bar{o}\vee deq\vee p\bar{q}f.$$

#### 4.5 Метод нахождения булевой формулы многоугольника в ДНФ без использования дополнительных предикатных переменных (базовый метод)

Недостаток приведенного выше простого метода нахождения булевой формулы многоугольника заключается в том, что в найденную им формулу наряду с предикатными переменными, которые связаны с полуплоскостями, порождаемыми сторонами многоугольника, входят еще и дополнительные предикатные переменные, а также инверсии некоторых из них. Ниже предлагается метод построения ДНФ многоугольника, свободный от этих недостатков, и позволяющий в большинстве случаев получить более качественное решение, однако характеризующийся и более высокой вычислительной сложностью.

#### 4.5.1 Элементы покрытия, порожденные сторонами многоугольника

Будем говорить, что многоугольник  $M^*$  является элементом покрытия многоугольника  $M$ , если все точки плоскости, которые принадлежат многоугольнику  $M^*$ , принадлежат также и многоугольнику  $M$ . Кроме того, будем говорить, что многоугольник  $M$  поглощает любой свой элемент покрытия  $M^*$ .

Опишем способ нахождения элементов покрытия многоугольника  $M$ , основанный на последовательном анализе его сторон.

Сначала будем рассматривать те стороны многоугольника, которые ограничены с обеих сторон выпуклыми углами, т. е. имеют ранг 0. Такой стороной в многоугольнике  $M$ , показанном на рисунке 4.9, является, например, сторона  $a$ .

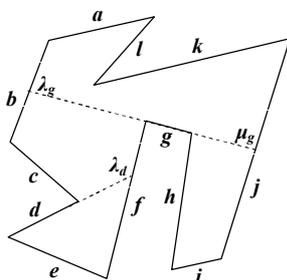


Рисунок 4.9 – Многоугольник  $M$

Начиная со стороны  $a$ , выполняем обход границы многоугольника  $M$  в прямом направлении (т. е. в направлении, задаваемом ориентированными прямыми, соответствующими сторонам многоугольника), придерживаясь следующих правил.

Если при обходе встречается выпуклый угол, то выполняется переход по этому углу на очередную сторону многоугольника.

Если же встречается вогнутый угол, то текущая сторона продлевается дальше до первого пересечения с другой стороной многоугольника, после чего выполняется переход на эту сторону и движение по ней в прямом направлении.

Такой процесс обхода продолжается до тех пор, пока мы не окажемся на стороне многоугольника, которая уже была пройдена раньше. В результате граница обхода будет представлять собой контур или будет включать в себя контур. Так как направление движения изменялось только выпуклыми углами, то фигура, ограниченная найденным контуром, будет представлять собой выпуклый многоугольник и являться элементом покрытия многоугольника  $M$ .

На рисунке 4.9 для стороны  $a$  граница обхода будет  $abc^+f^+k^+$  (индексом «+» помечена удлиненная сторона многоугольника, индексом «±» помечена сторона, которая продлена с одной стороны и укорочена – с другой). Найденный при этом контур задает элемент покрытия  $S_l$ , имеет границу  $b^-c^+f^+k^+$  (индексом «-» помечена укороченная сторона многоугольника) и выделен серым цветом на рисунке 4, а.

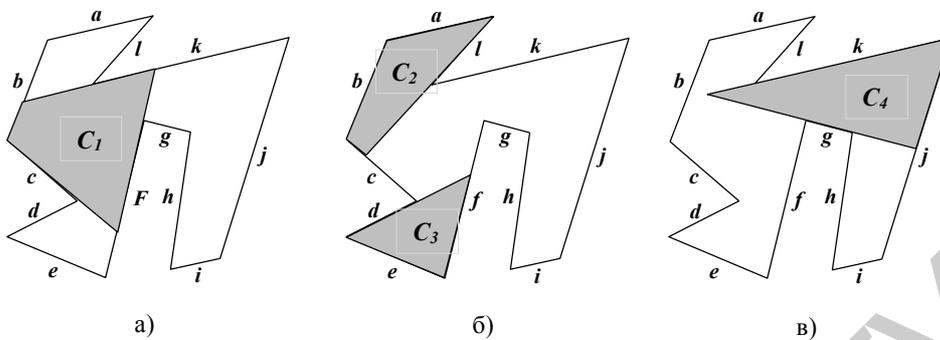


Рисунок 4.10 – Элементы покрытия: а)  $C_1$ , б)  $C_2$  и  $C_3$ , в)  $C_4$

Однако обход границы при поиске элементов покрытия можно вести не только в прямом, но и в обратном направлении (т. е. в направлении, противоположном тому, которое задается ориентированными прямыми, соответствующими сторонам многоугольника).

Например, сторона  $a$  при обходе в обратном направлении порождает элемент покрытия  $C_2$  (рис. 4, б) с границей  $a^+c^-b$ .

Теперь рассмотрим способ порождения элементов покрытия теми сторонами многоугольника, которые ограничены с одной стороны выпуклым, а с другой – вогнутым углами, т. е. имеют ранг 1. В этом случае необходимо придерживаться следующих правил.

Если обход границы должен начинаться со стороны выпуклого угла, то способ порождения элемента покрытия остается тем же, что был описан выше. Например, сторона  $c$  будет порождать элемент покрытия  $C_1$  с границей  $c^+f^+k^+b^-$ .

Если же обход границы должен начинаться со стороны вогнутого угла, то сначала нужно продлить рассматриваемую сторону многоугольника в направлении от вогнутого угла и найти первую точку пересечения с другой стороной многоугольника. Именно от этой точки пересечения и необходимо начинать обход границы. Например, сторону  $d$  многоугольника (рис. 3) нужно сначала продлить до пересечения со стороной  $f$  в точке  $\lambda_d$ , после чего, начиная с этой точки, выполнить обход по границе  $d^+ef$ . В результате будет найден элемент покрытия  $C_3$  (рис. 4, б).

Перейдем к рассмотрению способа порождения элементов покрытия теми сторонами многоугольника, которые ограничены с обеих сторон вогнутыми углами, т. е. имеют ранг 2. В многоугольнике *М* единственной такой стороной является сторона  $g$ . Эту сторону нужно предварительно продлить в обоих направлениях до первого пересечения с другими сторонами многоугольника (соответствующие точки пересечения  $\lambda_g$  и  $\mu_g$  показаны на рисунке 3). С одной из этих точек (в прямом направлении – с точки  $\lambda_g$ , в обратном направлении – с точки  $\mu_g$ ) и необходимо начинать обход границы при поиске элементов покрытия.

Таким образом, сторона  $g$  порождает элемент покрытия  $C_4$  (рис. 4, в) с границей  $g^+j^-k^+$  и началом обхода в точке  $\lambda_g$ , и элемент покрытия  $C_5$  (рис. 5) с границей  $g^-b^-a^+$  (индексом « $\sim$ » помечен отрезок прямой, лежащий на линии, являющейся продолжением стороны многоугольника) и началом обхода в точке  $\mu_g$ .

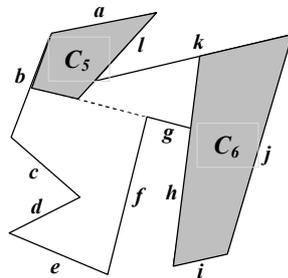


Рисунок 4.11 – Элементы покрытия  $C_5$  и  $C_6$

#### 4.5.2 Базовый метод нахождения элементов покрытия многоугольника

В работах [27,28] показано, что булеву формулу любого многоугольника можно представить в ДНФ, которая соответствует покрытию многоугольника его выпуклыми компонентами.

В работе [30] предложен метод покрытия многоугольника выпуклыми компонентами, который далее будем называть базовым, суть которого заключается в следующем:

- последовательно перебираются все стороны  $s_1, s_2, \dots, s_n$  многоугольника  $M$ , в результате чего находятся все порождаемые ими элементы покрытия, которые в совокупности образуют множество  $C = \{C_1, C_2, \dots, C_p\}$ ;
- просматриваются элементы множества  $C$ , и если очередной элемент поглощается каким-либо другим элементом этого же множества, то он удаляется.

После этого все оставшиеся элементы множества  $C$  заменяются представляющими их конъюнкциями соответствующих предикатных переменных. Тогда дизъюнкция этих конъюнкций и даст искомую булеву формулу многоугольника в ДНФ.

Базовый метод нахождения элементов покрытия проиллюстрируем на примере многоугольника  $M$ , изображенного на рисунке 3.

Если обход границы вести в прямом направлении, то сторона  $a$ , как было показано выше, порождает элемент покрытия  $C_1$  с границей  $b^-c^+f^+k^+$ . Тот же самый элемент покрытия порождают стороны  $b, c, e, f, i, j$ . Сторона  $d$  порождает элемент покрытия  $C_3$  с границей  $d^+ef^-$ , а сторона  $g$  – элемент покрытия  $C_4$  с границей  $g^+j^-k^+$ . Сторона  $h$  порождает элемент покрытия  $C_6$  с границей  $h^+ijk^-$ , а сторона  $l$  – элемент покрытия  $C_2$  с границей  $l^+abc^-$ .

Если обход границы вести в обратном направлении, то сторона  $a$  порождает элемент покрытия  $C_2$  с границей  $al^+c^-b$ . Тот же самый элемент покрытия порождают стороны  $b, c, l$ . Сторона  $d$  порождает элемент покрытия  $C_3$  с границей  $d^+f^-e$ . Тот же элемент покрытия  $C_3$  порождают стороны  $e$  и  $f$ . Сторона  $g$  порождает элемент покрытия  $C_5$  с границей  $g^-b^-al^+$ . Сторона  $h$  порождает элемент покрытия  $C_6$  с границей  $h^+k^-ji$ . Тот же элемент покрытия  $C_6$  порождают стороны  $i$  и  $j$ .

Тем самым будет найдено множество  $C = \{C_1, C_2, \dots, C_6\}$ , содержащее все возможные элементы покрытия, порождаемые сторонами многоугольника  $M$ . Так как элемент  $C_5$  поглощается элементом  $C_2$ , то он удаляется из множества  $C$ .

В результате булева формула многоугольника  $M$  в ДНФ будет иметь следующий вид:

$$F = bcfk \vee alcb \vee def \vee gjk \vee hijk.$$

#### 4.5.3 Усовершенствованный метод нахождения булевой формулы многоугольника

Однако, базовый метод нахождения элементов покрытия не всегда приводит к получению точного решения для рассматриваемой задачи. Применив его, например, к многоугольнику, показанному на рисунке 4.12, а, мы получим точное решение. Однако, чуть видоизменив этот многоугольник путем увеличения длины стороны  $b$  (рисунок 4.12, б), мы получим уже приближенное решение (непокрытый выпуклыми компонентами остаток показан на рисунке серым цветом).

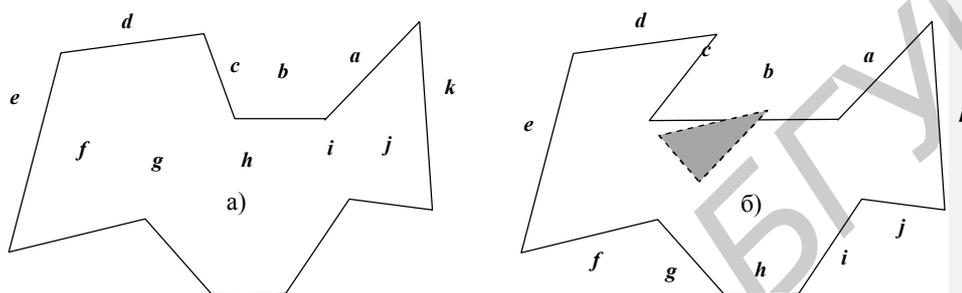


Рисунок 4.12 – Многоугольники: а) покрываемый выпуклыми компонентами, б) с непокрытым остатком

Усовершенствованный метод [31], изложенный ниже, всегда позволяет находить точное решение. Это означает, что для любой задачи элементы решения, интерпретируемые как выпуклые компоненты, будут покрывать в совокупности всю область плоскости, занимаемую многоугольником.

Эмпирически было установлено, что:

- базовый метод нахождения элементов покрытия многоугольника всегда позволяет получить точное решение, если граница многоугольника не содержит сторон с рангом 2 (т.е. сторон, у которых оба смежных угла являются вогнутыми);
- если в результате продления стороны с рангом 2 до первого пересечения с другой стороной многоугольника будет отсекается выпуклый многоугольник, то любая точка этого многоугольника не будет принадлежать непокрытому остатку, полученному после применения базового метода.

Эти факты позволяют усовершенствовать базовый метод нахождения элементов покрытия так, что для любого многоугольника будет найдено точное решение. Усовершенствованный метод включает в себя следующие действия.

- К многоугольнику  $M$  применяется базовый метод нахождения элементов покрытия, т.е. последовательно перебираются все стороны  $s_1, s_2, \dots, s_n$  многоугольника, в результате чего находятся все порождаемые ими элементы покрытия, которые после удаления поглощаемых элементов образуют множество  $S$ . При этом можно сократить перебор, если выбросить из рассмотрения стороны с рангом, равным нулю. Это объясняется тем, что стороны с нулевым рангом не дают никаких новых элементов покрытия в сравнении с элементами покрытия, которые порождаются сторонами с рангом 1 и рангом 2.

Если многоугольник  $M$  не содержит сторон с рангом 2, то выполняется переход к пункту б.

– Многоугольник  $M$  заменяется двумя смежными (имеющими общий отрезок внутри своих границ) компонентами  $M^1$  и  $M^2$ , получаемыми путем разбиения  $M$  на две части. Для этого сторону с рангом 2 достаточно продлить в прямом или в обратном направлении до первого пересечения с другой стороной многоугольника.

Если сторон с рангом 2 несколько, например  $m$ , то из  $2 \times m$  вариантов продления выбирается вариант, позволяющий отсечь выпуклый многоугольник. Если такой вариант отсутствует, то выбирается тот вариант, который позволяет преобразовать большее число сторон с рангом 2 в стороны с рангом 1.

Каждый из вновь полученных многоугольников  $M^1$  и  $M^2$  подвергается анализу на наличие в нем сторон с рангом 2. Если такой многоугольник имеется, то он заменяется парой смежных компонент тем же способом, что описан в этом пункте выше. Такой процесс анализа и преобразования порождаемых компонент-многоугольников продолжается до тех пор, пока каждый из них не будет содержать только стороны с рангом 0 и 1.

Из найденного множества многоугольников удаляются все те, которые являются выпуклыми. В итоге будут найдены многоугольники, которые обозначим:  $M^i, M^j, \dots, M^k$ .

– К каждому многоугольнику  $M_i, M_j, \dots, M_k$  последовательно применяется базовый метод нахождения элементов покрытия. В результате будут найдены множества  $C_i, C_j, \dots, C_k$ , содержащие элементы покрытия многоугольников  $M_i, M_j, \dots, M_k$  соответственно.

– Последовательно перебираются элементы, входящие в множества  $C_i, C_j, \dots, C_k$ , и для каждого из них проверяется, поглощается ли этот элемент каким-либо элементом множества  $C$ . Если поглощение отсутствует, то такой элемент добавляется во вспомогательное множество  $C^*$ , первоначально пустое.

– Если  $C^* \neq \emptyset$ , то элементы множества  $C^*$  добавляются в множество  $C$ .

– Все элементы множества  $C$  заменяются представляющими их конъюнкциями соответствующих предикатных переменных. Тогда дизъюнкция этих конъюнкций и будет представлять собой искомую булеву формулу многоугольника в ДНФ.

Изложенный метод проиллюстрируем на примере многоугольника  $M$ , изображенного на рисунке 6, б.

Применив к многоугольнику  $M$  базовый метод, находим множество  $C = \{c_1, c_2, \dots, c_5\}$  его элементов покрытия с границами  $a^+g^-hi^+k, b^+e^-f^+k, c^+def^-j^+ka^+, b^+k^-j^+e^-$  соответственно.

Так как многоугольник  $M$  содержит всего одну сторону с рангом 2, то имеется только два варианта его замены смежными компонентами  $M^1$  и  $M^2$ . Эти варианты, показанные на рисунке 4.13, равноценны, поскольку оба отсекают выпуклые компоненты и преобразуют сторону с рангом 2 в сторону с рангом 1.

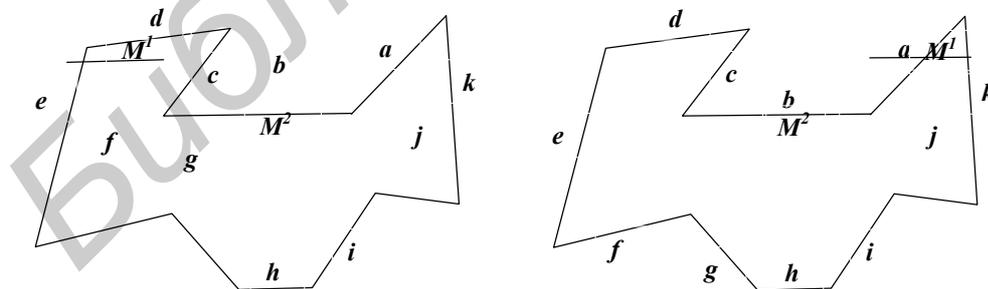


Рисунок 4.13 – Два варианта замены многоугольника  $M$  компонентами  $M^1$  и  $M^2$

Рассмотрим, например, вариант 2. Исключив из рассмотрения выпуклый многоугольник  $M^2$ , применим базовый метод к многоугольнику  $M^1$  и найдем множество  $C^1$ , содержащее элементы покрытия  $M^1$ .

Последовательно перебираем элементы, входящие в множество  $C^1$ , и для каждого из них проверяем, поглощается ли этот элемент каким-либо элементом множества  $C$ . В результате оказывается, что только один элемент покрытия, имеющий границу  $g^+h^+b^+$ , не поглощается ни одним элементом множества  $C$ , и, поэтому, будет помещен во вспомогательное множество  $C^*$ , а затем – и в множество  $C$ . В итоге искомая булева формула многоугольника  $M$  в ДНФ будет иметь следующий вид:

$$F = aghik \vee befkv \vee cdef \vee jka \vee bkje \vee ghib.$$

#### 4.6 Комплекс программ минимизации систем частичных булевых функций

Задача минимизации булевых функций в классе дизъюнктивных нормальных форм (задача минимизации ДНФ) появилась в связи с задачами синтеза комбинационных схем при проектировании логики цифровых устройств из элементов И-НЕ, И-ИЛИ-НЕ и др.

Минимизация ДНФ оказывается полезной и в других случаях, как предварительный этап синтеза, значительно упрощающий реализацию последующих этапов, например, при синтезе схем из программируемых логических матриц (ПЛИМ) или на программируемых логических интегральных схемах (ПЛИС).

Методам минимизации ДНФ посвящены уже сотни работ, которые восходят к классическим работам Квайна-МакКласки и Блека-Порецкого [33,34].

Ниже рассматриваются методы минимизации систем в общем случае не полностью определенных булевых функций, опирающиеся и обобщающие ряд оригинальных результатов, приведенных в [35–41], а также описывается комплекс программ, реализующих точные и приближенные алгоритмы минимизации.

##### 4.6.1 Булевы функции

Булевой называется функция  $y = f(x_1, x_2, \dots, x_n)$ , аргументы которой  $x_1, x_2, \dots, x_n$  принимают значения из множества  $\{0,1\}$  и сама она принимает значения из этого же множества.

Упорядоченная совокупность переменных  $(x_1, x_2, \dots, x_n)$  образует  $n$ -компонентный вектор  $x$ , значением которого являются булевы векторы, т.е. векторы, составленные из 0 и 1.

Все  $2^n$  различных булевых векторов длины  $n$  образуют в совокупности булево пространство  $M$  размерности  $n$ .

Можно считать, что функция  $f$  задает разбиение булева пространства  $M$  на два непересекающихся подмножества  $M^1$  и  $M^0$ , на элементах которого она принимает соответственно единичное и нулевое значения.

Пусть, например,  $n=3$  и булева функция  $f(x_1, x_2, x_3)$  задает следующее разбиение булева пространства  $M$ :

$$M^1 = (000, 010, 011, 110, 111), \quad M^0 = \{001, 100, 101\}.$$

Это означает, что она принимает значение 1 на наборе 000 ( $x_1=0, x_2=0, x_3=0$ ), значение 0 на наборе 001 ( $x_1=0, x_2=0, x_3=1$ ) и т.д.

Можно ограничиться заданием только одного из множеств  $M^1, M^0$ . Обычно указывают множество  $M^1$ , называя его характеристическим множеством булевой функции.

$M^1$  удобно задавать как множество строк некоторой булевой матрицы (т.е. матрицы, состоящей из 0 и 1), принимающей для рассматриваемого случая следующий вид:

$$\begin{array}{c} x_1x_2x_3 \\ 1 \begin{bmatrix} 0 & 0 & 0 \\ 2 & 0 & 1 & 0 \\ 3 & 0 & 1 & 1 \\ 4 & 1 & 1 & 0 \\ 5 & 1 & 1 & 1 \end{bmatrix} \end{array}$$

Компактность представления характеристического множества  $M^1$  можно повысить, если использовать троичные векторы, компоненты которых могут принимать в качестве своих значений не только символы 0 и 1, но и символ “-“. Будем при этом интерпретировать троичный вектор как совокупность всех таких булевых векторов, которые получаются из него всевозможными подстановками значений 0 и 1 вместо значения “-“. Если в троичном векторе существует  $k$  значений “-“, то он порождает таким образом  $2^k$  булевых векторов, образующих в булевом пространстве  $M$  объект, называемый интервалом.

Например, троичный вектор 1-0- при такой интерпретации рассматривается как интервал пространства четырех булевых переменных, образованный элементами 1000, 1001, 1100, 1101.

Введем в рассмотрение троичные матрицы, строками которых служат троичные векторы. Положим, что троичная матрица эквивалентна булевой матрице, получаемой из нее заменой каждой троичной строки на порождаемую ею совокупность булевых строк и последующим “приведением подобных“, при котором устраняются повторения среди строк. Будем считать, что они представляют одну и ту же булеву функцию. Например, приведенная выше булева матрица оказывается эквивалентной троичной матрице

$$\begin{array}{c} x_1x_2x_3 \\ 1 \begin{bmatrix} 0 & - & 0 \\ 2 & - & - \end{bmatrix}, \end{array}$$

позволяющей, как видно, представить рассматриваемую булеву функцию  $f$ , т.е. ее характеристическое множество  $M^1$ , значительно более компактно.

Две троичные матрицы эквивалентны, если они эквивалентны одной и той же булевой матрице, т.е. если они представляют одну и ту же булеву функцию.

Нетрудно видеть, что от булевой матрицы

$$\begin{array}{c} x_1x_2x_3 \\ 1 \begin{bmatrix} 0 & 0 & 0 \\ 2 & 0 & 1 & 0 \\ 3 & 0 & 1 & 1 \\ 4 & 1 & 1 & 0 \\ 5 & 1 & 1 & 1 \end{bmatrix}, \end{array}$$

задающей булеву функцию переменных  $x_1, x_2, x_3$ , легко совершается переход к соответствующей совершенной дизъюнктивной нормальной форме (СДНФ):

$$\bar{x}_1\bar{x}_2\bar{x}_3 \vee \bar{x}_1x_2\bar{x}_3 \vee \bar{x}_1x_2x_3 \vee x_1x_2\bar{x}_3 \vee x_1x_2x_3,$$

а от эквивалентной троичной матрицы

$$\begin{array}{c} x_1x_2x_3 \\ 1 \begin{bmatrix} 0 & - & 0 \end{bmatrix} \\ 2 \begin{bmatrix} - & 1 & - \end{bmatrix} \end{array}$$

– к дизъюнктивной нормальной форме (ДНФ)  $\bar{x}_1\bar{x}_3 \vee x_2$ . Как видно, последняя формула значительно проще и, следовательно, удобнее.

Получение простейших ДНФ для заданной булевой функции известно как задача минимизации ДНФ. За критерий простоты при этом может быть принято общее число букв (литералов) в формуле и в этом случае говорят о поиске минимальной ДНФ. На практике чаще используется другой критерий, а именно число элементарных конъюнкций в ДНФ. Дизъюнктивная нормальная форма, простейшая в этом смысле, называется кратчайшей. Задача нахождения кратчайшей ДНФ совпадает с задачей получения троичной матрицы, эквивалентной заданной и содержащей минимальное число строк.

#### 4.6.2 Частичные булевы функции

Если значения некоторой булевой функции заданы на всех наборах значений аргументов, то она называется полностью определенной. Наряду с такими функциями практический интерес представляют и не полностью определенные булевы функции, называемые также частичными булевыми функциями. Их значения определены лишь на некоторых наборах. Будем считать, что на остальных наборах такая функция принимает неопределенное значение (неизвестно 0 или 1), обозначаемое символом “–”.

Частичная булева функция  $f$  может быть задана разбиением булева пространства  $M$  на три подмножества  $M^1, M^0, M$ , образуемые наборами, на которых функция получает соответственно значение 1, 0 или “–”. Очевидно, чтобы определить такую функцию, достаточно задать лишь два из этих подмножеств. Обычно задают множества  $M^1$  и  $M^0$ .

#### 4.6.3 Системы булевых функций

Часто требуется минимизировать не одну булеву функцию, а систему булевых функций

$$F = \{f_1(x), f_2(x), \dots, f_m(x)\},$$

которые в общем случае являются частичными и зависят от одного и того же множества аргументов  $x_1, x_2, \dots, x_n$ .

Такая задача возникает как один из этапов синтеза при проектировании многовыходных комбинационных схем и заключается не столько в минимизации каждой из функций системы, но в таком их преобразовании, при котором наряду с упрощением формул выявлялись бы одинаковые части, реализуемые на общем оборудовании.

Поскольку булевы функции, входящие в систему  $F$ , представляются в одном и том же булевом пространстве  $M$ , возникает необходимость введения некоторых характеристик общности объектов (элементов, интервалов) булева пространства для системы  $F$ .

Обозначим через  $M_i^1$  и  $M_i^0$  множества элементов булева пространства  $M$ , на котором функция  $f_i$  принимает соответственно единичное и нулевое значения. Интервал  $u$  булева пространства  $M$  называется полезным для функции  $f_i \in F$ , если выполняются следующие два условия:

$$\begin{aligned} u \cap M_i^0 &= \emptyset; \\ u \cap M_i^1 &\neq \emptyset. \end{aligned}$$

Полезный для функции  $f_i \in F$  интервал  $u \subseteq M$  называется максимальным полезным интервалом, если для этой функции нельзя построить другой полезный интервал  $u'$ , такой, что  $u' \supset u$ .

При совместной минимизации функций в системе необходимо каждый объект булева пространства рассматривать не по отношению к отдельной функции, а по отношению к некоторому подмножеству функций в системе. В связи с этим введем понятие  $F$ -интервала, представляющего собой пару  $(u, h)$ , состоящую из интервала  $u \subseteq M$  и некоторого подмножества функций  $h \subseteq F$ , называемого характеристикой интервала.

$F$ -интервал  $(u, h)$  называется полезным, если интервал  $u$  является полезным для каждой из функций, входящих в характеристику  $h$ .

Полезный  $F$ -интервал  $(u, h)$  называется максимальным полезным интервалом, если нельзя построить другой полезный  $F$ -интервал  $(u', h)$ , такой, чтобы  $u' \supset u$ .

Интервал  $u$  и его характеристику  $h$  будем называть также левой и правой компонентами  $F$ -интервала  $(u, h)$ .

В алгебраической форме  $F$ -интервала  $(u, h)$  записывается в виде конъюнкции, соответствующей интервалу  $u$ , и ее характеристики, представленной в виде перечня номеров функций. Например,  $x_1 \bar{x}_3 \{1, 3, 4\}$  означает, что конъюнкция  $x_1 \bar{x}_3$  относится к функциям  $f_1, f_3, f_4$ .

Ранг  $F$ -интервала  $(u, h)$  состоит из двух компонент  $r(u)$  и  $|h|$ , где  $r(u)$  – ранг интервала  $u$ , равный числу литералов в соответствующей ему конъюнкции, а  $|h|$  – ранг характеристики, равный мощности множества  $h$ .

Если ранг характеристики  $h \subseteq F$  равен единице, то такой  $F$ -интервал будем называть  $f$ -интервалом. Нетрудно видеть, что каждый  $F$ -интервал можно разложить на составляющие его  $f$ -интервалы, число которых равно рангу характеристики  $h$ .

Элемент булева пространства  $e \in M$  также обобщим, рассматривая его не по отношению к отдельной булевой функции, а по отношению к некоторому подмножеству функций в системе. В этом случае пара  $(e, h)$  называется  $F$ -элементом. Если ранг характеристики  $h$  равен единице, то такой  $F$ -элемент будем называть  $f$ -элементом. Последний уже нельзя разложить на какие-либо составные части.

Дальнейшие рассуждения проиллюстрируем на примере системы шести частичных булевых функций, зависящих от семи переменных. Множества  $M_i^1, M_i^0$  ( $i = 1, 2, \dots, 6$ ) представим следующими булевыми матрицами:

$$x_1 x_2 x_3 x_4 x_5 x_6 x_7 \quad x_1 x_2 x_3 x_4 x_5 x_6 x_7$$

$$M_1^1 = \frac{1}{2} \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}, \quad M_1^0 = \frac{1}{3} \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix},$$

$$M_2^1 = \frac{1}{1} \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}, \quad M_2^0 = \frac{1}{2} \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 \end{bmatrix},$$



В множествах  $P$  и  $Q$  каждую группу  $f$ -элементов, имеющих одинаковые левые компоненты, можно заменить одним  $F$ -элементом с той же левой компонентой и с правой компонентой, получаемой объединением правых компонент  $f$ -элементов группы. В результате множества  $P$  и  $Q$  преобразуются в более компактные множества  $P_F$  и  $Q_F$ :

$$\begin{array}{l}
 x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ x_7 \\
 P_F = \begin{array}{l} 1 \left[ \begin{array}{ccccccc} 0 & 0 & 1 & 0 & 0 & 1 & 0 \end{array} \right] \{1,6\} \\ 2 \left[ \begin{array}{ccccccc} 0 & 1 & 1 & 1 & 0 & 1 & 0 \end{array} \right] \{1\} \\ 3 \left[ \begin{array}{ccccccc} 1 & 0 & 0 & 1 & 1 & 1 & 0 \end{array} \right] \{2\} \\ 4 \left[ \begin{array}{ccccccc} 0 & 1 & 0 & 1 & 0 & 1 & 1 \end{array} \right] \{3\} \\ 5 \left[ \begin{array}{ccccccc} 0 & 1 & 0 & 1 & 1 & 0 & 1 \end{array} \right] \{3\} \\ 6 \left[ \begin{array}{ccccccc} 0 & 0 & 1 & 0 & 1 & 0 & 1 \end{array} \right] \{4,6\} \\ 7 \left[ \begin{array}{ccccccc} 1 & 1 & 1 & 0 & 1 & 1 & 1 \end{array} \right] \{5\} \end{array} \\
 Q_F = \begin{array}{l} 1 \left[ \begin{array}{ccccccc} 0 & 0 & 1 & 0 & 1 & 0 & 1 \end{array} \right] \{1\} \\ 2 \left[ \begin{array}{ccccccc} 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{array} \right] \{1,2,3,5\} \\ 3 \left[ \begin{array}{ccccccc} 1 & 0 & 0 & 1 & 1 & 1 & 0 \end{array} \right] \{1,4,5\} \\ 4 \left[ \begin{array}{ccccccc} 1 & 1 & 1 & 0 & 1 & 1 & 1 \end{array} \right] \{2\} \\ 5 \left[ \begin{array}{ccccccc} 0 & 0 & 1 & 0 & 0 & 1 & 0 \end{array} \right] \{3\} \\ 6 \left[ \begin{array}{ccccccc} 0 & 1 & 1 & 1 & 0 & 1 & 0 \end{array} \right] \{4\} \\ 7 \left[ \begin{array}{ccccccc} 0 & 1 & 0 & 1 & 0 & 1 & 1 \end{array} \right] \{5\} \\ 8 \left[ \begin{array}{ccccccc} 0 & 1 & 0 & 1 & 1 & 0 & 1 \end{array} \right] \{5,6\} \end{array}
 \end{array}$$

Будем говорить, что  $F$ -интервал  $(u, h)$  поглощает  $F$ -интервал  $(v, g)$ , если  $u \supseteq v$  и  $h \supseteq g$ .

Множество  $WF$ -интервалов, поглощающих в совокупности все  $f$ -элементы множества  $P$  и не поглощающих ни одного  $f$ -элемента множества  $Q$ , называется множеством, реализующим систему  $F$ . Такое множество  $F$ -интервалов, записанных в алгебраической форме, называется дизъюнктивной нормальной формой (ДНФ) системы  $F$ . Очевидно, что из ДНФ системы можно получить ДНФ любой функции  $f_i \in F$  путем объединения знаками дизъюнкции тех конъюнкций, характеристики которых содержат номер функции  $f_i$ .

Таким образом, для нахождения кратчайшей ДНФ системы функций  $F$  необходимо получить множество  $F$ -интервалов, реализующее систему  $F$  и обладающее минимальной мощностью.

#### 4.6.4 Поиск кратчайшей ДНФ системы функций

Рассмотрим множество  $Pf$ -элементов, задающих области единичных значений функций системы  $F$ . Подмножество множества  $P$  назовем совместимым, если существует полезный  $F$ -интервал, поглощающий каждый из  $f$ -элементов этого подмножества. Совместимое подмножество будем называть максимальным, если в него уже невозможно добавить ни одного нового  $f$ -элемента из  $P$  без потери свойства совместимости.

Классический метод решения задачи поиска кратчайшей ДНФ системы функций заключается в сведении ее к задаче о кратчайшем покрытии. Для этого в множестве  $P$  следует отыскать все максимальные совместимые подмножества и для каждого из них сформировать полезный  $F$ -интервал, поглощающий все элементы данного подмножества, после чего построить матрицу бинарного отношения поглощения  $f$ -элементов множества  $P$  сформированными  $F$ -интервалами. Кратчайшее покрытие этой булевой матрицы и определит множество  $R$  полезных  $F$ -интервалов, реализующее систему  $F$  и обладающее наименьшей мощностью по сравнению со всеми другими множествами, реализующими систему  $F$ .

Для нахождения всех максимальных совместимых подмножеств можно использовать метод, основанный на сокращенном переборе подмножеств множества  $P$  и проверке, существует ли полезный  $F$ -интервал, поглощающий элементы очередного подмножества. Сокращение перебора основано на том факте, что если какое-либо

подмножество не является совместимым, то и любое его расширение, получаемое путем добавления в него новых элементов, также будет несовместимым.

Процесс перебора можно отобразить деревом поиска, в котором каждое ребро соответствует некоторому  $f$ -элементу из множества  $P$ , а каждая вершина – совместимому подмножеству, образованному  $f$ -элементами-ребрами, связывающими данную вершину с корнем дерева.

Метод обхода дерева поиска реализует лексикографический перебор вариантов, порождаемых последовательным расщеплением ситуаций: продукты расщепления каждой ситуации нумеруются и перебираются затем в соответствующем порядке. При методе обхода достаточно сохранять в памяти информацию лишь о тех ситуациях, которые соответствуют вершинам, образующим цепь (путь) от рассматриваемой вершины к корню дерева. Объем этой информации, как правило, невелик по сравнению с по ярусным построением дерева или с методом ветвей и границ. После того, как дерево поиска, отображающее процесс сокращенного перебора подмножеств множества  $P$  будет построено, его концевые вершины (листья дерева) будут соответствовать максимальным совместимым подмножествам (сами подмножества отображаются цепями, связывающими листья с корнем дерева).

Если продолжить рассмотрение примера системы из шести булевых функций, то дерево поиска всех максимальных совместимых подмножеств будет иметь вид, представленный на рисунке 4.6.1 (для простоты ребра отмечены не  $f$ -элементами, а номерами соответствующих строк матрицы  $P$ ).

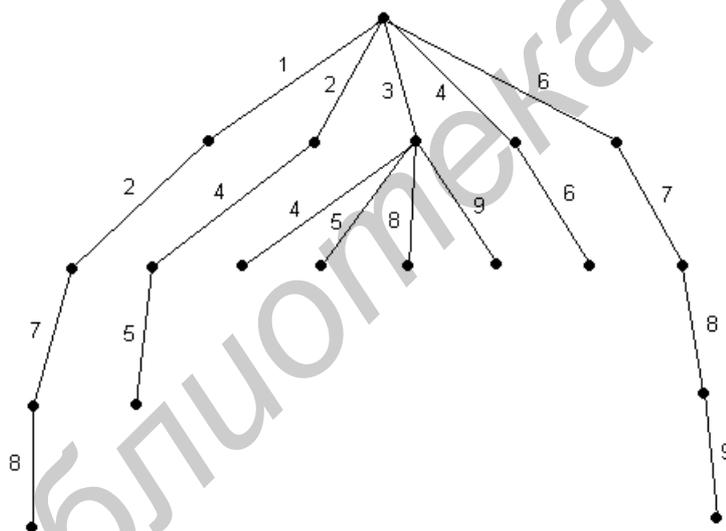


Рисунок 4.14 – Дерево поиска максимальных совместимых подмножеств множества  $P$

Теперь можно перечислить все максимальные совместимые подмножества множества  $P$ , изобразив их для удобства с помощью секционированной взвешенной матрицы:

		$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	
1	1	0	0	1	0	0	1	0	{1}
	2	0	1	1	1	0	1	0	{1}
	7	1	1	1	0	1	1	1	{5}
	8	0	0	1	0	0	1	0	{6}
2	2	0	1	1	1	0	1	0	{1}
	4	0	1	0	1	0	1	1	{3}
	5	0	1	0	1	1	0	1	{3}
3	3	1	0	0	1	1	1	0	{2}
	4	0	1	0	1	0	1	1	{3}
4	3	1	0	0	1	1	1	0	{2}
	5	0	1	0	1	1	0	1	{3}
5	3	1	0	0	1	1	1	0	{2}
	8	0	0	1	0	0	1	0	{6}
6	3	1	0	0	1	1	1	0	{2}
	9	0	0	1	0	1	0	1	{6}
7	4	0	1	0	1	0	1	1	{3}
	6	0	0	1	0	1	0	1	{4}
8	6	0	0	1	0	1	0	1	{4}
	7	1	1	1	0	1	1	1	{5}
	8	0	0	1	0	0	1	0	{6}
	9	0	0	1	0	1	0	1	{6}

Здесь каждая секция представляет элементы одного максимального совместимого подмножества.

Для каждого из найденных максимальных совместимых подмножеств сформируем полезный  $F$ -интервал по следующему правилу: левая компонента  $F$ -интервала есть минимальный поглощающий интервал для левых компонент  $f$ -элементов секции, правая компонента образуется объединением их правых компонент. В результате получим множество  $WF$ -интервалов, которое выглядит следующим образом (строки обозначим не цифрами, а буквами  $a, b, \dots, h$ ):

$$W = \begin{array}{l} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \end{array} \begin{array}{l} x_1 x_2 x_3 x_4 x_5 x_6 x_7 \\ \left[ \begin{array}{ccccccc} - & - & 1 & - & - & 1 & - \\ 0 & 1 & - & 1 & - & - & - \\ - & - & 0 & 1 & - & 1 & - \\ - & - & 0 & 1 & 1 & - & - \\ - & 0 & - & - & - & 1 & 0 \\ - & 0 & - & - & 1 & - & - \\ 0 & - & - & - & - & - & 1 \\ - & - & 1 & 0 & - & - & - \end{array} \right] \end{array} \begin{array}{l} \{1,5,6\} \\ \{1,3\} \\ \{2,3\} \\ \{2,3\} \\ \{2,6\} \\ \{2,6\} \\ \{3,4\} \\ \{4,5,6\} \end{array}$$

Теперь можно построить матрицу  $B$  бинарного отношения поглощения  $f$ -элементов множества  $PF$ -интервалами множества  $W$ , которая примет следующий вид:

$abcdefgh$

$$B = \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{matrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

Простейший способ нахождения кратчайшего покрытия матрицы  $B$  заключается в переборе всех возможных подмножеств столбцов, отборе из них тех подмножеств, которые содержат сообща не менее одной единицы в каждой строке, и в выборе среди последних подмножества, содержащего наименьшее количество столбцов.

Перебор можно сократить основываясь на том, что множество решений задачи покрытия – подмножеств столбцов с указанным выше свойством – является выпуклым по отношению включения, и это означает, что если некоторое подмножество является решением, то и любое его расширение, получаемое добавлением в него некоторых новых элементов, также будет представлять собой решение. В таком случае нахождение множества всех решений сводится к нахождению его нижней границы, что может иногда существенно сократить объем производимых при поиске вычислений.

Процесс сокращенного перебора подмножеств строк при поиске кратчайшего покрытия также можно отобразить деревом поиска, в котором каждое ребро ставится в соответствие некоторому столбцу матрицы  $B$ , а каждая вершина – подмножеству строк, образованному столбцами-ребрами, связывающими данную вершину с корнем дерева. Прямой шаг выполняется лишь в том случае, если подмножество столбцов, соответствующее текущей вершине, не является покрытием и, следовательно, граница выпуклого множества решений еще не достигнута.

Еще одна возможность сокращения перебора основана на том, что если при обходе дерева поиска будет найдено покрытие, содержащее  $m$  столбцов, то в дальнейшем можно ограничиться рассмотрением цепей длины не более, чем  $m$  (при поиске всех кратчайших покрытий), либо не более, чем  $m-1$  (при поиске одного кратчайшего покрытия).

По окончании перебора самая короткая цепь в дереве поиска, достигающая нижней границы выпуклого множества решений, определит своими столбцами-ребрами кратчайшее покрытие матрицы  $B$ . Если ищутся все кратчайшие покрытия, то таких цепей может быть несколько.

Продолжая рассмотрение примера, построим дерево поиска одного кратчайшего покрытия матрицы  $B$  (рисунок 4.15), отмечая пунктиром нижнюю границу выпуклого множества решений и плюсом – вершины, пути из которых представляют решение задачи покрытия.

На рисунке видно, что кратчайшее покрытие матрицы  $B$  образует подмножество столбцов  $\{a, f, g\}$ . Отсюда следует, что множество

$$R = \begin{matrix} x_1 x_2 x_3 x_4 x_5 x_6 x_7 \\ a \\ f \\ g \end{matrix} \begin{bmatrix} - & - & 1 & - & - & 1 & - \\ - & 0 & - & - & 1 & - & - \\ 0 & - & - & - & - & - & 1 \end{bmatrix} \begin{matrix} \{1,5,6\} \\ \{2,6\} \\ \{3,4\} \end{matrix}$$

реализует систему функций  $F$  и обладает наименьшей мощностью.

Рассмотренный метод поиска кратчайшего покрытия матрицы  $B$  характеризуется громоздкими вычислениями. Если же нужно найти только одно из кратчайших покрытий, целесообразно использовать следующий более простой метод. Он основан на использовании формулируемых ниже правил редуцирования (упрощения) переменной матрицы  $X$ , характеризующей текущую ситуацию и первоначально равной матрице  $B$ .

*Правило удаления строки.* Из матрицы  $X$  удаляется строка  $x_i$ , если она поглощает некоторую другую строку  $x_j$  (булев вектор  $c$  поглощает булев вектор  $d$ , если каждая пара одноименных компонент  $c_i$  и  $d_i$  удовлетворяет условию:  $c_i \geq d_i$ ).

*Правило удаления столбца.* Из матрицы  $X$  удаляется столбец  $x^j$ , если он поглощается некоторым другим столбцом  $x^k$ .

*Правило приращения покрытия.* Если некоторая строка  $x_i$  матрицы  $X$  имеет лишь один элемент  $x_i^j$  со значением 1, то столбец  $x^j$  включается в конструируемое покрытие.

Включаемые в покрытие столбцы удаляются из матрицы  $X$  вместе со всеми покрываемыми строками, в которых они содержат единицы.

Если в очередном миноре нет ни поглощающих строк, ни поглощаемых столбцов, следует применить правило расщепления, представляющее собой обобщение правила приращения покрытия.

*Правило расщепления.* В матрице  $X$  отыскивается минимальная (по числу единиц) строка и поочередно рассматриваются варианты, связанные с включением в конструируемое покрытие столбцов, имеющих в этой строке единицу. Дополнительное сокращение перебора обеспечивается следующим правилом ограничения, или правилом возврата.

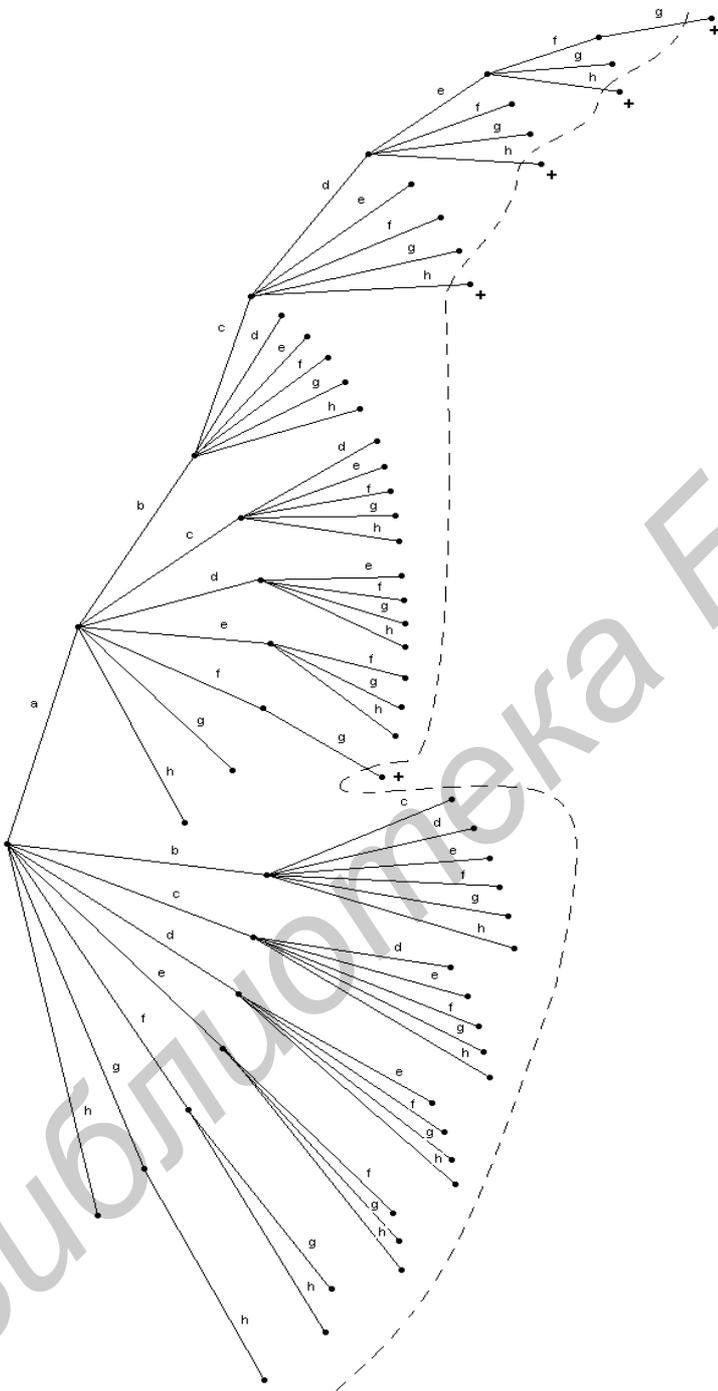


Рисунок 4.15 – Дерево поиска одного кратчайшего покрытия матрицы  $B$

*Правило возврата.* Если в процессе поиска решения удалось найти некоторое покрытие, последующий перебор ограничивается рассмотрением таких подмножеств столбцов, мощность которых меньше мощности найденного покрытия.

Рассмотрим работу сформулированных правил на том же самом примере. В исходной ситуации принимается, что

$$X_0 = B = \begin{matrix} & abcdefgh \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{matrix} & \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix} \end{matrix},$$

а конструируемое покрытие пока пустое. В матрице  $X_0$  удаляются строки 2, 7 и 8, поскольку каждая из них поглощает строку 1. В полученном миноре выбрасывается столбец  $b$ , так как он поглощается столбцом  $g$ , и столбец  $e$ , поглощаемый столбцом  $c$ . Получается не упрощаемый далее минор

$$X_1 = \begin{matrix} & acdfgh \\ \begin{matrix} 1 \\ 3 \\ 4 \\ 5 \\ 6 \\ 9 \end{matrix} & \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix} \end{matrix}$$

Столбец  $a$  включается в конструируемое покрытие, поскольку строка 1 содержит единственную единицу в этом столбце, после чего мы переходим к рассмотрению матрицы

$$X_2 = \begin{matrix} & cdfgh \\ \begin{matrix} 3 \\ 4 \\ 5 \\ 6 \\ 9 \end{matrix} & \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix} \end{matrix}$$

Выберем теперь строку 4, первую из минимальных по числу единиц. Она содержит единицы в столбцах  $c$  и  $d$ , в соответствии с чем предстоит рассмотреть два варианта

расширения конструируемого множества: включение в него столбца  $c$  или отказ от этого и, следовательно, включение в него столбца  $g$ . Анализируя первый вариант, мы удаляем как столбец  $c$ , так и покрываемые им строки 3 и 4 и рассматриваем далее остаток матрицы

$$X_3 = \begin{matrix} & d & f & g & h \\ \begin{matrix} 5 \\ 6 \\ 9 \end{matrix} & \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \end{matrix}.$$

Затем выбрасываются поглощаемые столбцы  $d, f$  и, далее, поглощающая строка 6, в результате чего получаем

$$X_4 = \begin{matrix} & g & h \\ \begin{matrix} 5 \\ 9 \end{matrix} & \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \end{matrix}$$

Становится очевидным, что в решение нужно включить столбцы  $g$  и  $h$ , поскольку строки 5 и 9 содержат по одной единице. В результате получаем покрытие  $\{a, c, g, h\}$ , содержащее четыре столбца.

В дальнейшем имеет смысл продолжить поиск покрытий только среди таких подмножеств, которые состоят не более, чем из трех элементов. Возвращаясь к точке ветвления вариантов мы вновь рассматриваем минор  $X_2$  и выбираем теперь оставшийся вариант покрытия строки 4, по которой производилось расщепление, т.е. выбираем столбец  $g$ . Выбрасывая его и покрываемые им строки 4, 5 и 6, переходим к рассмотрению матрицы

$$X_5 = \begin{matrix} & c & d & f & h \\ \begin{matrix} 3 \\ 9 \end{matrix} & \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \end{matrix}$$

Далее выбрасываются поглощаемые столбцы  $c, d$  и  $h$ . Очередной минор элементарен:

$$X_6 = \begin{matrix} & f \\ \begin{matrix} 3 \\ 9 \end{matrix} & \begin{bmatrix} 1 \\ 1 \end{bmatrix} \end{matrix}$$

В решение включается единственный оставшийся столбец  $f$ . В результате получаем покрытие  $\{a, f, g\}$ , содержащее три столбца и являющееся кратчайшим. Этому покрытию соответствует уже знакомое нам множество.

$$R = \begin{matrix} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \\ \begin{matrix} a \\ f \\ g \end{matrix} & \begin{bmatrix} - & - & 1 & - & - & 1 & - \\ - & 0 & - & - & 1 & - & - \\ 0 & - & - & - & - & - & 1 \end{bmatrix} \end{matrix} \begin{matrix} \{1,5,6\} \\ \{2,6\} \\ \{3,4\} \end{matrix},$$

реализующее систему  $F$  и обладающее наименьшей мощностью.

Рассмотренный пример оказался, таким образом, довольно простым с одной точкой ветвления вариантов. Однако число точек ветвления и, соответственно, объем вычислений быстро возрастают с увеличением размеров исходной матрицы  $B$ .

Если сравнивать два рассмотренных выше метода поиска кратчайшего покрытия, то метод, основанный на сокращенном переборе подмножеств множества  $W$ , характеризуется большим объемом вычислений, однако позволяет находить все кратчайшие покрытия (из них можно выбрать покрытие, которому соответствует ДНФ системы с наименьшим суммарным рангом конъюнкций). Метод, основанный на использовании правил редуцирования матрицы  $B$ , более быстродействующий, но ограничивается поиском только одного кратчайшего покрытия.

Необходимо отметить, что полезные  $F$ -интервалы, входящие в решение  $R$ , в общем случае не являются максимальными и их левые компоненты можно попытаться расширить за счет областей неопределенности функций, входящих в правые компоненты. Для этого перебираются элементы множества  $R$  и для очередного  $F$ -интервала  $(u, h)$  выполняются следующие действия.

В векторе  $u$  случайным образом выбираем компоненту, принимающую определенное значение (0 или 1) и это значение заменяем на “-”, получая в результате вектор  $u'$ . Если  $F$ -интервал  $(u', h)$  оказывается полезным, то интервал  $u$  заменяем на  $u'$  и описанную процедуру замены определенных значений на неопределенные повторяем снова и т.д. до тех пор, пока эти замены возможны. В результате будет сформирован максимальный полезный  $F$ -интервал, поглощающий исходный  $F$ -интервал  $(u, h)$ .

Случайный выбор в векторе  $u$  компонент для замены их значений объясняется тем, что для  $F$ -интервала  $(u, h)$  существует, вообще говоря, много поглощающих его максимальных полезных  $F$ -интервалов, имеющих различные ранги. Мы же ограничиваемся приближенным решением.

Что касается рассматриваемого нами примера, то попытка устранить избыточность рассмотренным выше способом к успеху не приводит, поскольку все  $F$ -интервалы, вошедшие в решение  $R$ , оказались максимальными полезными  $F$ -интервалами.

#### 4.6.5 Приближенные методы решения задачи минимизации ДНФ

Точное решение задачи поиска кратчайшей ДНФ системы функций, как видно из предыдущего материала, сопряжено с громоздкими вычислениями, объем которых резко возрастает с ростом размерности решаемых задач. По этой причине на практике имеет смысл использовать приближенные методы, позволяющие находить решения, близкие к оптимальным. Иллюстрацией к сказанному может служить следующий простейший алгоритм решения задачи покрытия матрицы  $B$ , дающий достаточно хорошее приближение к кратчайшему покрытию: найти минимальную (по числу единиц) строку, выбрать максимальный среди столбцов, содержащих в ней единицу, и включить его в конструируемое подмножество (вначале оно пустое), затем удалить из матрицы выбранный столбец и все строки, содержащие в нем единицу, и повторить эту процедуру уже для остатка матрицы, и т.д. до тех пор, пока растущее подмножество не станет покрытием. Оно и будет представлять приближенное решение задачи покрытия.

В рассматриваемом нами примере этот алгоритм приводит к начальному выбору в матрице  $B$  строки 1 и столбца  $a$ , после чего из матрицы удаляется столбец  $a$  и строки 1, 2, 7, 8 и она сокращается до

$$\begin{array}{c}
 bcdefgh \\
 3 \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \end{bmatrix} \\
 4 \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \\
 5 \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix} \\
 6 \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \\
 9 \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}
 \end{array}$$

Затем выбираются строка 6 и столбец  $g$  и матрица сокращается до

$$\begin{array}{c}
 bcdefh \\
 3 \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} \\
 9 \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}
 \end{array}$$

Следующий выбор строки 9 и столбца  $f$  приводит к получению покрытия  $\{1, 6, 7\}$ . В данном случае оно оказывается кратчайшим.

Если же рассматривать задачу минимизации ДНФ системы функций в целом, то ее решение связано не только с отысканием покрытия матрицы  $B$ , но и с достаточно трудоемким процессом ее построения, из-за чего решение задачи практической сложности оказывается невозможным.

Рассмотрим в связи с этим приближенный метод, позволяющий избежать построения промежуточной матрицы  $B$  и поиска ее кратчайшего покрытия.

Известно, что самыми быстродействующими являются цепные алгоритмы, без возвратных движений по дереву поиска. Их реализация разбивается на этапы, и на каждом из этих этапов выбирается лишь один вариант его прохождения, причем оптимальность его условна и определяется некоторым эвристическим критерием, составляющим суть метода. Для решения рассматриваемой задачи мы будем использовать двухступенчатое разбиение решающего процесса на этапы, при котором каждое звено цепного алгоритма реализуется в свою очередь своей цепью, состоящей из более мелких звеньев.

Пусть результатом прохождения крупного этапа является включение в решение  $R$  некоторого полезного  $F$ -интервала, а результатом прохождения мелкого этапа – минимальное расширение очередного конструируемого  $F$ -интервала, т.е. минимальное изменение рангов его компонент, при котором он станет поглощать некоторый новый  $f$ -элемент множества  $P$ .

В качестве критерия оптимальности выбора очередного минимального расширения для конструируемого  $F$ -интервала примем ранг его левой компоненты. Оптимальным будем считать такое расширение, которому соответствует максимальное значение этого критерия; в этом случае поглощается наиболее “близкий” к нему  $f$ -элемент и остается максимум возможностей для дальнейшего расширения. При этом выбор ограничивается теми минимальными расширениями, которые не выводят конструируемый  $F$ -интервал из класса полезных  $F$ -интервалов. Если же таких расширений нет, то конструирование очередного  $F$ -интервала считается завершенным и он заносится в решение  $R$ . При наличии нескольких конкурирующих вариантов расширения с максимальным значением данного критерия будем выбирать тот из них, который характеризуется минимальным рангом характеристики. Если же и таких вариантов оказывается несколько, то выбирается первый из них, минимальный по номеру.

Те  $f$ -элементы из  $P$ , которые не поглощаются ни одним из  $F$ -интервалов множества  $R$ , образуют не поглощаемый остаток  $S$ , и конструирование нового  $F$ -интервала

начинается с выбора одного из элементов этого остатка. В качестве критерия оптимальности такого выбора примем число элементов из  $S$ , совместимых с выбранным  $f$ -элементом, полагая, что чем меньше это число, тем лучше, т.к. в этом случае выбранный  $f$ -элемент будет входить, как правило, в меньшее число полезных  $F$ -интервалов, поглощающих элементы множества  $S$ .

Приняв такой  $f$ -элемент в качестве нового  $F$ -интервала, мы уменьшаем число возможных ошибок, которые можно сделать при последующем его расширении. При наличии нескольких конкурирующих вариантов с минимальным значением этого критерия выбирается первый из них.

Назовем первый из описанных критериев критерием  $A$ , а второй – критерием  $B$  и применим данный алгоритм к уже известной нам задаче минимизации системы из шести булевых функций, заданных множествами  $P$  и  $Q$ .

Согласно изложенному алгоритму конструирование первого  $F$ -интервала начинается с выбора из множества  $P$ , где

$$P = \begin{matrix} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{matrix} & \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} & \begin{matrix} \{1\} \\ \{1\} \\ \{2\} \\ \{3\} \\ \{3\} \\ \{4\} \\ \{5\} \\ \{6\} \\ \{6\} \end{matrix} \end{matrix},$$

$f$ -элемента, представленного строкой 3, оптимального по критерию  $B$ . Совместимыми с ним будут  $f$ -элементы, представленные строками 4, 5, 8, 9, поэтому имеется четыре варианта минимального расширения конструируемого  $F$ -интервала. Оптимальным по критерию  $A$  будет расширение, поглощающее  $f$ -элемент, представленный строкой 8. В результате будет сформирован  $F$ -интервал  $-0---10\{2,6\}$ . Так как последний оказывается несовместимым со всеми непоглощаемыми им  $f$ -элементами множества  $P$ , то первый этап завершается и полученный  $F$ -интервал  $-0---10\{2,6\}$  заносится в решение  $R$ .

Аналогично находятся  $F$ -интервалы  $--1--1-\{1,5\}$ ,  $0010101\{4,6\}$ ,  $0101--1\{3\}$ , после чего не поглощаемый остаток  $S$  станет равным пустому множеству.

После устранения избыточности, связанной с тем, что как правило не все элементы множества  $R$  являются максимальными полезными интервалами, будет получено следующее окончательное решение:

$$R = \begin{matrix} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} - & - & - & - & - & - & 0 \\ - & - & 1 & - & - & 1 & - \\ - & - & - & 0 & - & - & - \\ 0 & - & 0 & - & - & - & - \end{bmatrix} & \begin{matrix} \{2,6\} \\ \{1,5\} \\ \{4,6\} \\ \{3\} \end{matrix} \end{matrix}$$

#### 4.6.6 Программная реализация методов минимизации ДНФ системы функций

На основе рассмотренных методов поиска кратчайшей ДНФ системы частных булевых функций разработан комплекс программ на языке С++ (среда программирования С++ Builder), реализующих точные и приближенные алгоритмы минимизации. Комплекс включает в себя программы минимизации: ExactMinSys1, ExactMinSys2, ApproxMinSys1, ApproxMinSys2, ApproxMinSys3, ApproxMinSys4, и две утилиты: SourceAnalyzer и ResultAnalyzer.

Программы используют другую, более удобную для машинной обработки, форму представления исходных данных с помощью булевой  $X$  и троичной  $Y$  матриц. Обратимся к уже известной нам системе из шести булевых функций, которую мы задавали с помощью множеств  $P$  и  $Q$  или, что то же самое, с помощью более компактных множеств  $P_F$  и  $Q_F$ :

$$\begin{array}{c}
 x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ x_7 \\
 P_F = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 \end{bmatrix} \begin{array}{l} \{1,6\} \\ \{1\} \\ \{2\} \\ \{3\} \\ \{3\} \\ \{4,6\} \\ \{5\} \end{array} \\
 Q_F = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} \begin{array}{l} \{1\} \\ \{1,2,3,5\} \\ \{1,4,5\} \\ \{2\} \\ \{3\} \\ \{4\} \\ \{5\} \\ \{5,6\} \end{array}
 \end{array}$$

Однако ту же самую систему функций можно задать в следующей матричной форме:

$$\begin{array}{c}
 x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ x_7 \ f_1 \ f_2 \ f_3 \ f_4 \ f_5 \ f_6 \\
 X = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \\
 Y = \begin{bmatrix} 1 & - & 0 & - & - & 1 \\ 1 & - & - & 0 & - & - \\ 0 & 1 & - & 0 & 0 & - \\ - & - & 1 & - & 0 & - \\ - & - & 1 & - & 0 & 0 \\ 0 & - & - & 1 & - & 1 \\ - & 0 & - & - & 1 & - \\ 0 & 0 & 0 & - & 0 & - \end{bmatrix}
 \end{array}$$

Здесь строки матрицы  $X$  представляют элементы булева пространства  $M$ , а сама матрица  $X$  представляет объединение областей определения всех функций, входящих в систему  $F$ . Строки матрицы  $Y$  задают значения, которые принимают функции, которыми отмечены столбцы матрицы, на элементах булева пространства, перечисленных строками матрицы  $X$  (т.е. на наборе 0010010 функции  $f_1$  и  $f_6$  принимают значение 1, функция  $f_3$  – значение 0, значение остальных функций не определено и т.д.).

Результат минимизации также представляется в матричной форме с помощью троичной матрицы  $X_R$  и булевой матрицы  $Y_R$ :

$$\begin{array}{cccccccc}
 & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & f_1 & f_2 & f_3 & f_4 & f_5 & f_6 \\
 X_R = & \begin{bmatrix} - & - & 1 & - & - & 1 & - \\ - & 0 & - & - & 1 & - & - \\ 0 & - & - & - & - & - & 1 \end{bmatrix} & Y_R = & \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}
 \end{array}$$

Пара одноименных строк в этих матрицах задает один из  $F$ -интервалов, входящих в решение  $R$  (строка матрицы  $X_R$  представляет его левую компоненту, а строка матрицы  $Y_R$  единицами указывает функции, входящие в его правую компоненту).

Разработанные программы минимизации были опробованы на ряде примеров систем функций различной размерности и различной степени неопределенности. При этом установлено, что время решения зависит от ряда параметров, характеризующих исходные данные, среди которых основными являются

- число  $n$  столбцов в матрице  $X$ ;
- число  $l$  единиц в матрице  $Y$ ;

причем эту зависимость можно приближенно определить как экспоненциальную от параметра  $l$  и мультипликативную от параметра  $n$ . Кроме того, время решения зависит от алгоритма, реализуемого программой, и выбранных эвристических критериев  $A$  и  $B$ .

Программы ExactMinSys1 и ExactMinSys2 предназначены для нахождения точного решения задачи минимизации системы функций путем нахождения всех максимальных совместимых подмножеств множества  $P$ , построения булевой матрицы  $B$  и отыскания ее кратчайшего покрытия. Эти программы реализуют различные алгоритмы поиска кратчайшего покрытия. ExactMinSys1 использует алгоритм сокращенного перебора столбцов матрицы  $B$  по правилам обхода дерева поиска и позволяет находить все кратчайшие ДНФ системы функций. ExactMinSys2 использует алгоритм, основанный на применении правил редукционирования матрицы  $B$  и позволяет найти лишь одну из кратчайших ДНФ системы. Однако размерность задач, которые могут быть решены с помощью этих программ за приемлемое на практике время, весьма невелика и определяется значением  $l$ , не превышающим 15 – 20 (при этом программа ExactMinSys2 является несколько более быстродействующей).

С помощью программ ApproxMinSys1, ApproxMinSys2, ApproxMinSys3 и ApproxMinSys4 находится приближенное решение задачи минимизации. Программы реализуют приближенный алгоритм, основанный на двухступенчатом цепном алгоритме, использующем эвристические критерии выбора  $A$  и  $B$ .

Для программы ApproxMinSys1 критерий  $A$  определяется рангом левой компоненты конструируемого  $F$ -интервала. Оптимальным считается такое расширение  $F$ -интервала, которому соответствует максимальное значение этого критерия.

В программе ApproxMinSys2 критерием  $A$  служит ранг правой компоненты конструируемого  $F$ -интервала. Оптимальным считается такое расширение  $F$ -интервала, которому соответствует минимальное значение этого критерия.

Для программы ApproxMinSys3 критерий  $A$  можно сформулировать следующим образом: оптимальным считается такое расширение  $F$ -интервала, для которого число  $f$ -элементов не поглощаемого остатка, перестающих быть с ним совместимыми, минимально.

Программа ApproxMinSys4 реализует несколько иной алгоритм, использующий лишь критерий  $B$  для выбора  $f$ -элемента, с которого начинается конструирование очередного  $F$ -интервала. После того, как такой выбор сделан, начинается перебор всех максимальных совместимых подмножеств не поглощаемого остатка, содержащих выбранный  $f$ -элемент. Из этих подмножеств выбирается подмножество, обладающее максимальной мощностью, которое и определит искомый  $F$ -интервал.

Все четыре программы приближенной минимизации могут быть настроены на любой из следующих трех вариантов критерия  $B$ , обеспечивающего оптимальность выбора  $f$ -элемента, на основе которого будет конструироваться очередной  $F$ -интервал:

- выбирается первый по порядку  $f$ -элемент не поглощаемого остатка;
- выбирается  $f$ -элемент, для которого число совместимых с ним элементов не поглощаемого остатка минимально;
- каждый  $f$ -элемент сравнивается со всеми совместимыми с ним элементами не поглощаемого остатка и при этом подсчитывается суммарное число совпадающих литералов в соответствующих конъюнкциях; выбирается  $f$ -элемент для которого эта сумма минимальна.

Программы, реализующие приближенные алгоритмы минимизации, пригодны для решения задач практической сложности, которые характеризуются значением  $l$  порядка нескольких сотен и значением  $n$  порядка нескольких десятков (кроме программы `ApproxMinSys4`, у которой эти показатели значительно ниже). Время решения на персональном компьютере при значениях  $n$  и  $l$ , приближающихся к максимальным, составляет несколько часов. Самыми быстродействующими оказываются программы `ApproxMinSys1` и `ApproxMinSys2`, настроенные на первый из вариантов критерия  $B$ .

Качество решения, получаемого программами приближенной минимизации, можно было оценить только на задачах небольшой размерности, для которых решение находили и точные алгоритмы. Для этих задач приближение достаточно хорошее. В ряде задач приближенными алгоритмами было получено точное решение.

На задачах практической размерности можно было сравнивать между собой только программы приближенной минимизации. При большой связности функций в системе наиболее предпочтительной является программа `ApproxMinSys1`, способствующая получению меньшего числа различных  $F$ -интервалов в решении  $R$  с большими мощностями характеристик. При малой связности функций в системе более предпочтительной является программа `ApproxMinSys2`, полнее использующая возможности минимизации отдельных функций, что приводит к получению  $F$ -интервалов с меньшими рангами.

Все программы приближенной минимизации допускают сжатую форму представления исходных данных, при которой матрица  $X$  является троичной и представляет как элементы, так и интервалы (либо только интервалы) булева пространства входных переменных. Однако полученное при этом решение  $R$  будет хуже по качеству, чем решение, полученное по этому же алгоритму, но примененному к развернутой форме представления, при которой матрица  $X$  является булевой. Это объясняется особенностью алгоритмов, которая заключается в следующем. Для любого  $f$ -элемента множества  $P$  в решении  $R$  обязательно найдется поглощающий его  $F$ -интервал. Иными словами, для каждого  $f$ -элемента из  $P$  в решении  $R$  обязательно должен присутствовать  $F$ -интервал, “полностью поглощающий” этот  $f$ -элемент. Это накладывает ограничение на возможные варианты покрытий, допускающиеся в случае развернутой формы задания.

Программа-утилита `SourceAnalyzer` является вспомогательной и предназначена для проверки исходных данных, если они представлены в сжатой форме. Проверка основана на том, что ни одна из функций системы не должна принимать различных значений на пересекающихся, т.е. содержащих некоторый общий элемент, интервалах, поскольку это противоречило бы определению понятия функции.

Программа-утилита `ResultAnalyzer` предназначена для анализа результата, полученного с помощью той или иной программы минимизации. Анализ заключается в проверке, реализуют ли  $F$ -интервалы, вошедшие в решение  $R$ , исходную систему булевых функций  $F$ .

Все программы работают с указателями, обеспечивающими создание динамических структур данных, которые позволяют обрабатывать большие объемы информации. Исходные данные и результат минимизации представляются в виде текстовых файлов.

Библиотека БГУИР

## **5 Моделирование процессов усвоения знаний и методов извлечения знаний для управления качеством программных средств**

### **5.1 Обобщенная характеристика проводимых исследований**

Разработка обобщенных интеллектуальных методов и моделей для оценивания качества программ и формирование общих принципов управления качеством на этапах создания программных средств. предполагает решение следующих задач

–Разработка структуры и методов адаптации обобщенной модели программных средств к задаче управления качеством.

–Систематизация и обобщение основных подходов к описанию качества программных средств.

–Разработка структуры и методов формализованного описания качества программных средств.

–Исследование наиболее распространенных принципов оценивания качества программных средств.

–Анализ и систематизация существующих подходов к оптимизации структурных и поведенческих свойств программных средств.

–Разработка общих принципов управления качеством на основе методов и моделей извлечения знаний из хранилищ данных.

–Разработка методики оценки качества программных средств на основе механизмов интеллектуального анализа данных.

–Разработка инструментальных средств интеллектуальной поддержки управленческих решений при управлении качеством программных средств.

*Методы исследования.* При выполнении исследований используются: общие методы системного анализа, общие методы теории программирования, методы теории автоматического управления, методы функционального анализа.

Ключевым фактором обеспечения эффективного применения программных продуктов является тщательное оценивание и достижение высоких значений показателей качества. Постоянный рост сложности функций, реализованных программами в информационных системах, непосредственно приводит к увеличению их объема и трудоемкости их создания. Согласно изменениям сложности программ возрастает количество найденных и оставленных в них дефектов и ошибок, что отражается на их качестве. Программа объемом в миллионы строк кода в принципе не может быть безошибочной. Проблема нахождения и устранения ошибок обостряется по мере увеличения сложности задач и программ, которые их решают, и угрожает катастрофами в областях, которые используют программное обеспечение.

Проблема заключается в том, чтобы обеспечить нужное качество функционирования ПО с учетом того, что некоторое неизвестное количество ошибок и дефектов всегда остается в сложных комплексах программ, и должно быть блокировано или сокращено их негативное воздействие до допустимого уровня. В связи с этим стратегическая задача в жизненном цикле современного ПО - обеспечение качества программных продуктов.

Качество ПО определяется качеством методов и инструментальных средств, которые применялись для обеспечения всего их жизненного цикла. На практике важно

оценивать качество программ не только в завершенном виде, но и в процессе их проектирования и разработки.

Сегодня ПО является определяющей составляющей многих систем, среди которых системы критического применения, встроенные и специализированные системы различного назначения. Для указанных систем наличие ошибок и низкое качество ПО грозят катастрофами, которые приводят к человеческим жертвам, экологическим катаклизмам, экономическим потерям. Развитие современных технологий разработки ПО требует динамичного развития средств оценки качества ПО, причем уже на этапе проектирования (с точки зрения экономической и временной целесообразности).

Современная индустрия ПО характеризуется высокой конкуренцией. Для успешной работы на этом рынке программная компания должна разрабатывать, внедрять и сопровождать ПО быстро, укладываясь в сроки, и с удовлетворительным качеством. Поэтому многие программные компании вкладывают средства в модернизацию технологий разработки ПО, помня о том, что такое вложение средств обязательно окупается.

*В процессе работы получены следующие результаты.*

- Общие принципы и алгоритм управления качеством на этапах создания программных средств.
- Многоцелевая математическая модель программных средств и способы ее адаптации к задаче управления качеством.
- Метод формализованного описания качества программных средств.
- Метод оценки степени соответствия программного средства типовым программным решениям.
- Концептуальная архитектура информационно-аналитической подсистемы поддержки принятия решений в области управления качеством программных средств.
- Принципы, основные модели и предложения по архитектуре хранилища данных.

## **5.2 Моделирование процессов усвоения знаний и методов извлечения знаний для управления качеством программных средств**

Интеллектуальный метод оценивания результатов проектирования и прогнозирования характеристик качества программного обеспечения (ИМОП). Принимая во внимание результаты анализа методов и средств оценки сложности и качества ПО, стало очевидным, что необходимо разработать методы и средства оценивания результатов проектирования и прогнозирования характеристик качества ПО. Перспективным направлением исследований является разработка интеллектуальных методов и систем, которые:

- будут вычислять расчетными и экспертными методами точные или прогнозируемые значения метрик программного обеспечения уже на этапе проектирования;
- на основе полученных значений метрик будут предоставлять рекомендации и выводы о разрабатываемом программном обеспечении, а также будут анализировать и обрабатывать результаты метрических оценок.

Исходя из анализа метрик, значения которых доступны на этапе проектирования, можно получить оценки, характеризующие этап проектирования, который осуществляется конкретной программной компанией, и получить прогнозируемые оценки качества разрабатываемого ПО по результатам этапа проектирования. Полученные оценки

результатов проектирования характеризуют уровень софтверной компании и серьезности отношения компании к данному заказу, а также предоставляют данные заказчику для выбора лучшей софтверной компании для разработки необходимого проекта и ПО. Прогнозируемые оценки характеристик качества разрабатываемого ПО предоставляют прогноз относительно качества реализации конкретной версии проекта и позволяют сравнить между собой различные версии проекта с такой точки зрения.

Суть интеллектуального метода оценки результатов проектирования и прогнозирования характеристик качества программного обеспечения заключается в оценивании результатов проектирования и прогнозирования характеристик качества ПО на основе метрического анализа. Для оценивания и прогнозирования сложности и качества ПО на основе метрического анализа следует решить задачу определения взаимосвязи между значениями метрик и такими характеристиками, как качество и сложность проекта ПО и уже разработанного ПО. Одним из средств, которое позволяет обобщить информацию и выявить зависимости между входными и результирующими данными, являются искусственные нейронные сети.

ИМОП использует искусственную нейронную сеть (ИНС), которая осуществляет аппроксимацию метрик, характеризующих ПО на этапе проектирования, в результате чего получаем оценку качества этапа проектирования и прогноз характеристик качества разрабатываемого ПО, т.е. ИМОП позволяет оценить проект и спрогнозировать характеристики сложности и качества разрабатываемого ПО на основе точных или прогнозируемых значений метрик сложности и качества ПО этапа проектирования.

Для формирования входных данных ИНС необходимо: множество метрик этапа проектирования с точными значениями (множество ТМР =  $\{tmp_i | i = 1..9\}$ ) и множество метрик этапа проектирования с прогнозируемыми значениями (множество РМР =  $\{rmp_j | j=1..15\}$ ).

Из полученных множеств ТМР, РМР формируются векторы, которые подаются на вход искусственной нейронной сети (ИНС).

Результатом работы ИНС является оценка сложности и качества проекта на основе точных метрик этапа проектирования и прогноз сложности и качества будущего программного обеспечения на основе прогнозируемых метрик этапа проектирования (рис.1).

ИНС для обработки метрик этапа проектирования ПО имеет 9 входов  $x'$  и 15 входов  $x$ , поскольку на входы  $x'$  подаются количественные значения метрик этапа проектирования с точными значениями, которых в предварительном анализе обнаружено 9, а на входы  $x$  подаются количественные значения метрик этапа проектирования с прогнозируемыми значениями, которых в предыдущем анализе обнаружено 15.

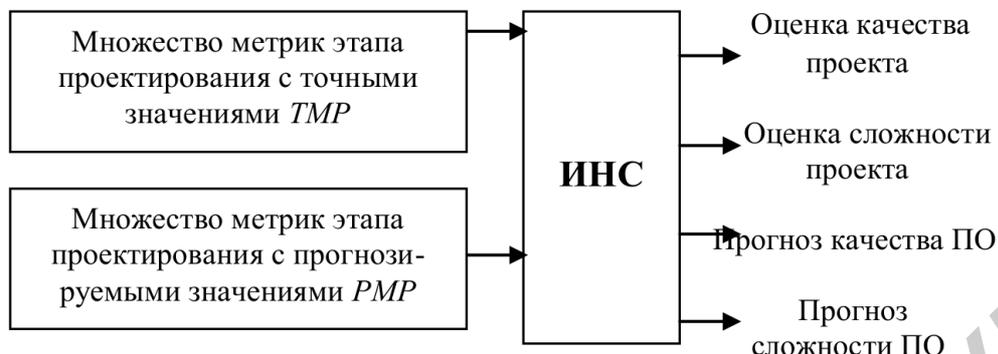


Рисунок 5.1 – Нейросетевая составляющая ИМОП

Входные данные для ИНС подаются в виде следующих множеств:

- множество метрик сложности этапа проектирования с точными значениями  $SMEV = \{smev_i \mid i = 1..4\}$ , где  $smev_i$  – количественное значение  $i$ -й метрики сложности с точным значением, если эта метрика определялась, иначе (-1);

- множество метрик качества этапа проектирования с точными значениями  $QMEV = \{qmev_j \mid j = 1..5\}$ , где  $qmev_j$  – количественное значение  $j$ -й метрики качества с точным значением, если эта метрика определялась, иначе (-1).

- множество метрик сложности этапа проектирования с прогнозируемыми значениями  $SMPV = \{smpv_k \mid k = 1..6\}$ , где  $smpv_k$  – количественное значение  $k$ -й метрики сложности с прогнозируемым значением, если эта метрика определялась, иначе (-1);

- множество метрик качества этапа проектирования с прогнозируемыми значениями  $QMPV = \{qmpv_n \mid n = 1..9\}$ , где  $qmpv_n$  – количественное значение  $n$ -й метрики качества с прогнозируемым значением, если эта метрика определялась, иначе (-1).

Результатами обработки этих множеств являются:

- оценка сложности проекта  $PCE$  в диапазоне  $[0, 1]$ , где 0 – метрики сложности с точными значениями не определялись, около 0 – проект сложный для реализации и предполагает высокую стоимость реализации, а 1 – проект прост для реализации;

- оценка качества проекта  $PQE$  в диапазоне  $[0, 1]$ , где 0 – метрики качества с точными значениями не определялись, около 0 – проект не качественный, а 1 – проект удовлетворяет требования заказчика по качеству;

- прогноз сложности разрабатываемого программного обеспечения  $SCP$  в диапазоне  $[0, 1]$ , где 0 – метрики сложности с прогнозируемыми значениями не определялись, около 0 – будущее ПО будет иметь существенную сложность, а 1 – будущее ПО ожидается простым;

- прогноз качества разрабатываемого программного обеспечения  $SQP$  в диапазоне  $[0, 1]$ , где 0 – метрики качества с прогнозируемыми значениями не определялись, около 0 – будущее ПО будет некачественным, а 1 – будущее ПО ожидается высокого качества.

На основе анализа 4-х полученных результатов делается вывод о качестве и сложности проекта и ожидаемых качестве и сложности разрабатываемого по проекту программного обеспечения.

Далее предстоит работа по разработке и программной реализации базовых модулей информационного и функционального обеспечения хранилища данных и моделей прогнозирования развития основных процессов управления качеством программных средств на основе «добычи» знаний (извлечение и анализ типовых «траекторий» управления качеством, разработка и обоснование пессимистических и оптимистических прогнозов с целью реализации предупреждающих действий в процессах).

Дополнительным достоинством системы будет являться использование в ней для извлечения данных технологии нейросетей, деревьев решений, кластеризации и др. методов Data Mining, позволяющей производить интеллектуальный анализ информации, а средства визуализации его результатов обеспечат эффективный инструмент для поддержки принятия решений по управлению качеством программных средств.

## **6 Программная поддержка управления исполнением процессов на основе сценариев**

### **6.1 Особенности управленческой деятельности в процессе обучения**

Применительно к учебному процессу управление представляет собой целенаправленное, систематическое воздействие преподавателя на коллектив учащихся и отдельного учащегося для достижения заданных результатов обучения. Но управлять – не значит подавлять или навязывать определенному процессу ход, противоречащий его природе. Наоборот, это значит максимально учитывать природу процесса, согласовывать каждое воздействие на процесс с его логикой.

Управление учебно-познавательной деятельностью учащихся в процессе обучения иногда понимается узко: как управление усвоением в рамках отдельных познавательных задач, т. е. на уровне оперативного управления (например, процессом открытия неизвестного в различных типах проблемных ситуаций). Средствами управления в этом случае являются «наводящие» задачи, подсказки разной интенсивности, переформулировка задачи. В широком смысле управление учебным познанием трактуется как процесс предъявления учащимся такой системы учебных задач, которая предусматривает в ходе их решения постепенное и последовательное продвижение обучаемых по ступеням познания – от низкого уровня проблемности заданий и познавательной самостоятельности к творческой, исследовательской деятельности. Тем самым проектируется определенный уровень сформированности свойств, качеств знаний (системность, динамичность, обобщенность и т. п.). Деятельность преподавателя направлена на выявление условий организации учебной работы, соблюдение которых позволит обучаемому сознательно ориентироваться в предмете, актуализировать полученные знания и умения, осуществлять самоконтроль.

Важнейшим условием организации процесса управления учебной деятельностью выступает система типовых учебных задач, применение которой способствует целенаправленному формированию у учащихся умения и потребности систематически пользоваться обязательным минимумом знаний как инструментом открытия новых знаний, новых связей и закономерностей. Через систему учебных задач, соответствующих конкретным целям обучения, преподаватель создает благоприятные условия для принятия таких задач учащимися, инструктирует их о способах предстоящей деятельности, оказывает им своевременную помощь, побуждает у них любознательность, чувство долга и ответственность.

При разработке программы обучения преподаватель должен оценить, какие знания, с какой целью и в какой степени он предполагает сформировать у учеников в результате

изучения ими конкретного материала. Для этого необходимо учесть особенности отдельных видов деятельности обучаемых. Существенное значение имеют также последовательность действий обучающихся, операциональный состав этих действий (исполнительских, оценочных и ориентировочных), нахождение способов мотивации обучаемых к участию в познавательной деятельности. В определении данных параметров учебной деятельности и заключается первая задача педагогического управления в структуре обучения.

Вторая задача сводится к осуществлению принципа активности и самоуправления в познавательной деятельности обучаемых. Она реализуется в такой организации учебных занятий, при которой преподаватель с помощью программ обеспечения и организации учебно-познавательной деятельности направлял бы и интенсифицировал процесс активной, самостоятельной и результативной работы каждого обучаемого по овладению основами теории и методами ее применения при решении учебно-познавательных задач [42].

## 6.2 Автоматизация процессов управления в образовательной сфере

Вычислительная техника является частью любого процесса обучения. Применяется преподавателями и учащимися для подготовки заданий, лекций, лабораторных работ, отчетов и других материалов. Элементы электронного обучения постепенно внедряются в вузах в процессе обучения студентов, автоматизируется процесс обучения. Управление процессом обучения при его информатизации остается как правило не затронутым или касается вспомогательных подразделений. Объединение автоматизации процесса обучения и его управления позволит, имея единую информационную систему, упростить ведение образовательной деятельности и его контроль [43].

На смену традиционному обучению приходит электронное и смешанное обучение. Электронное обучение (e-learning) широко применяется при организации дистанционного обучения. Смешанное обучение (blended learning) сочетает преимущества и недостатки электронного и традиционного обучения [44]. Достоинством традиционного обучения является очное общение, к недостаткам можно отнести:

- ограниченный объем доступных материалов;
- фиксированное расписание занятий и учебный план;
- низкая самостоятельность и ответственность студента.

К недостаткам электронного обучения можно отнести отсутствие гарантии при идентификации студента. Достоинств больше:

- дистанционное общение, объективная оценка и тестирование, система самоконтроля;
- удаленное обучение;
- большое число учебных материалов;
- гибкость, скорость, темп работы и индивидуализация;
- самостоятельная работа студента, выработка навыков самообучения и поиска информации.

Простой алгоритм работы преподавателя с учащимся включает следующие шаги:

- формирование преподавателем задания для учащегося;
- получение учащимся задания;
- выполнение задания;
- сдача отчета преподавателю.

Реализация этой последовательности действий через интернет является элементом e-learning. Введение информационной системы с функционалом оповещения преподавателей, учащихся и других сотрудников о необходимости выполнить некоторую задачу, автоматической рассылкой уведомлений, сбора и анализа отчетов позволит автоматизировать процесс обучения.

Автоматизация контроля и управления образовательным процессом развита слабо. Реализация информационной системы управления учебным процессом позволит снизить нагрузку на сотрудников вуза, переложив часть обязанностей на информационную систему управления, сократить время, затрачиваемое на выполнение отдельных задач, повысить точность и эффективность работ. Администрация сможет контролировать и отслеживать исполнение приказов, распоряжений, поручений, получать сводные данные и анализировать их.

Реализация информационной системы (рисунок 6.1), автоматизирующей процесс обучения и его управление, обеспечит работу сотрудников в едином информационном пространстве. Это позволит избежать проблем интеграции разного программного обеспечения, уменьшит затраты на внедрение и обучение работы с системой. Элементы e-learning, электронного документооборота, системы оповещений и напоминаний, автоматической рассылки, генерации отчетов, сбора и анализа информации будут взаимосвязано функционировать в одной информационной системе, с возможностью разделенного доступа к необходимым данным для всех сотрудников [45].



Рисунок 6.1 – Управление как часть системы

Таким образом можно сделать вывод, что образовательным процессом как и любым другим можно и нужно управлять. Но существующие формы традиционного управления обучением сегодня недостаточно эффективны, т.к. имеют ряд серьезных недостатков: контроль носит нерегулярный, по существу, эпизодический характер; действующие методики не ориентированы на перманентность применения и поэтому не вскрывается имеющая место динамика изменений; основная направленность на итоги обучения, поэтому сам процесс обучения и непрерывное управление им, формирующие уверенные знания, умения и навыки, а следовательно, и определяющие итоги обучения, остаются вне поля зрения.

### 6.3 Место сценарного подхода в управлении

Сценарный подход используется в стратегическом управлении и планировании. В стратегическом менеджменте под сценарием часто понимают план развития ситуации, однако зачастую теория стратегического планирования не учитывает актуальность анализа всех важных состояний системы, иными словами вариантов использования

потенциала. В отличие от стратегического плана, сценарий должен содержать описание тенденций развития, набор связей «что - если» между траекториями, событиями (ситуациями) деятельности, методами управления.

Таким образом, применение метода сценариев сопряжено с определенными трудностями: проработкой возможных состояний системы, определением связей между ними, а также обоснованием связей между формированием сценариев, управленческими решениями и развитием.

Безусловно, что прикладные особенности применения сценарного подхода к управлению определяются исходной предметной областью. Д. Новиков подчеркивает, что качественные методы (в частности, сценарии) позволяют исследовать траектории динамики системы [6.5]. Проведение анализа устойчивости решений, строящихся на основе сценариев, дает возможность выявить объективные системные свойства (качественные изменения, скачки развития), а также их изменения из-за неадекватности моделей. Стратегическое управление дополняет сценарный и ситуационный подход наличием цели.

Сочетание моделей управления, подходов и инструментов, используемых в сценарном подходе, позволяет определить его место в стратегическом и ситуационном управлении (рисунок 6.2).

Отметим, что сценарный подход в управлении ориентируется на вариативность ресурсов и формализацию будущих альтернатив использования потенциала. Такая формализация осуществляется с учетом опыта экспертов путем проведения экспертной оценки.

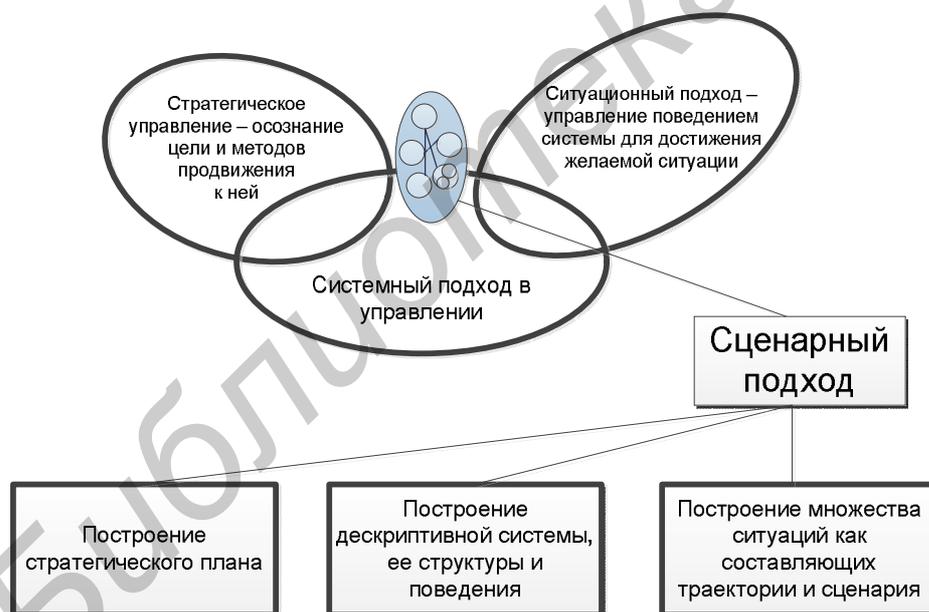


Рисунок 6.2 – Место сценарного подхода в управлении

Привлечение специалистов обеспечивает связь стратегии и сценариев, поскольку становится возможным выявить закономерности развития, проанализировать внешние и

внутренние факторы, влияющие на развитие системы и формулирование целей, провести анализ, создать вспомогательные информационные фонды.

Сценарий требует наличия обратной связи с определенными будущими ситуациями системы, значит, предусматривает внедрение системы мониторинга реализации сценариев и анализа ситуаций, с целью предупреждения и уменьшения негативных последствий неоднозначной интерпретации сценариев и их результатов. Представим сценарный подход к управлению схематично (рисунок 6.3). Выражение формулой:

$$AK = (AK1; AK2; AK3) \quad (6.1)$$

где  $A$  – начальная точка реализации сценариев;

$K_1, K_2, K_3$  – конечные цели для каждого сценария, формирующие пространство будущего развития событий и состояний социально-экономической системы.

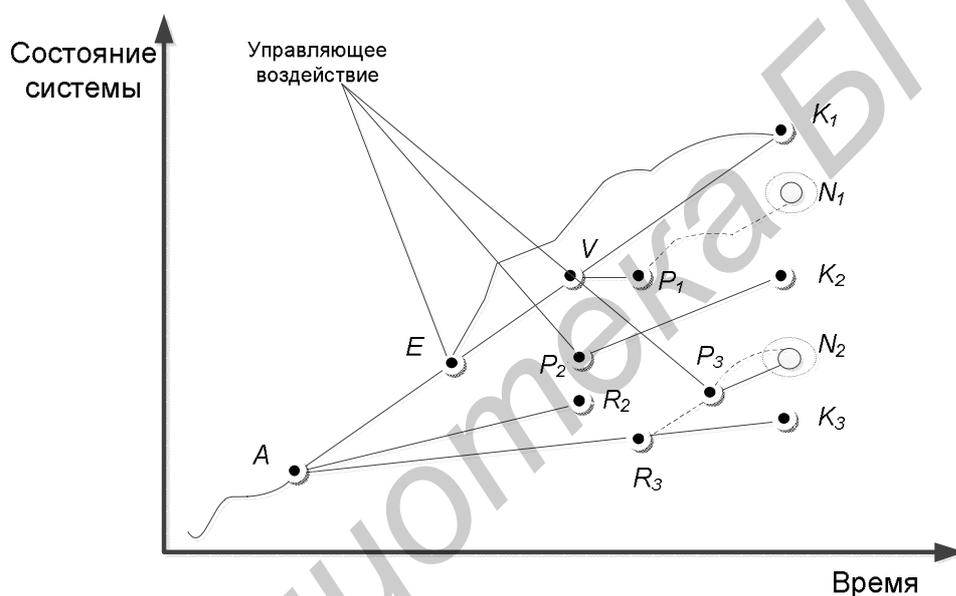


Рисунок 6.3 – Модель сценарного подхода в управлении

Сценарии управления строятся соответственно конечным ориентирам, следовательно, выделяются типы управления: составом, структурой, целевыми функциями, допустимыми множествами, информированностью, для которых выделены параметры системы, момент и время воздействия.

Конечные ориентиры формируются исходя из, во-первых, структурных преобразований. Во-вторых, конечным ориентиром, определяющим сценарий, является уровень зависимости от других процессов. В-третьих, сценарием можно считать интенсивность динамики развития. Этот сценарий является достаточно обширным и традиционно под ним понимаются количественные показатели. Вместе с тем, в этом сценарии важными и латентными являются движущие силы динамики. Четвертый ориентир, определяющий сценарий, имеет социальную направленность – достижение определенного уровня мотивации.

Таким образом, основные сценарии должны быть направлены на формирование единой системы целей, обеспечивающей:

- устойчивость элементов;
- надежность элементов, направленную на формирование системных и межсистемных образований;
- развитие элементов в виде устойчивой динамики;
- мотивацию элементов как стимул к деятельности.

На основании вышесказанного можно сделать следующие выводы:

- сценарный подход к управлению – это анализ и синтез вероятных будущих состояний системы с учетом поставленной стратегической цели;
- для сценария (или совокупности сценариев) существует точка отсчета. Сценарии отличаются друг от друга конечной целью. Учитывая влияние времени и необратимость развития, сценарием развития является ориентированный мультиграф траекторий развития, в вершинах которого осуществляет управление лицо, принимающее решения;
- определенные состояния или события – ситуации формируют множество траекторий развития управления, формирования и использования потенциала с целью достижения стратегической цели сценария;
- сценарии ориентированы на формирование системы конечных целей, включающей обеспечение устойчивости и надежности элементов, их развитие, а также поддержание и повышение мотивации. Сценарии не являются независимыми, возможно их пересечение путем координации, корректировки и внутренних процессов изменений структуры;
- траектория – это реализация процесса развития экономической системы с конкретизацией его формы. Траектория с формой развития, которая реализуется через реформы, является траекторией изменения сценария и осознания новой цели.

#### **6.4 Методы и способы повышения эффективности управления образовательным процессом компании**

Успешность проведения обучения компании зависимая величина. На рисунке 6.4 представлены факторы, влияющие на успех обучения персонала [47]. Немаловажную роль в этом играет поддержка руководства, ведь именно оно принимает решение о том быть образовательному процессу в компании или не быть. В большинстве случаев, организация обучения начинается «с легкой руки» руководства компании, реже – по инициативе самих корпоративных студентов (рисунок 6.4).



Рисунок 6.4 – Факторы, влияющие на успех в обучении

Для повышения эффективности использования трудовых ресурсов и сокращения времени разработки проектов компании требуется разработать политику, направленную на создание единого образовательного процесса, а так же деятельность по управлению им. Это позволит ускорить и облегчить доступ к необходимым обучающим ресурсам и материалам. Внедрение образовательного процесса в деятельность компании позволит систематизировать и структурировать обучение сотрудников, сделав его более качественным и эффективным, что в свою очередь сократит время реализации проектов и вместе с тем повысит качество разрабатываемого продукта.

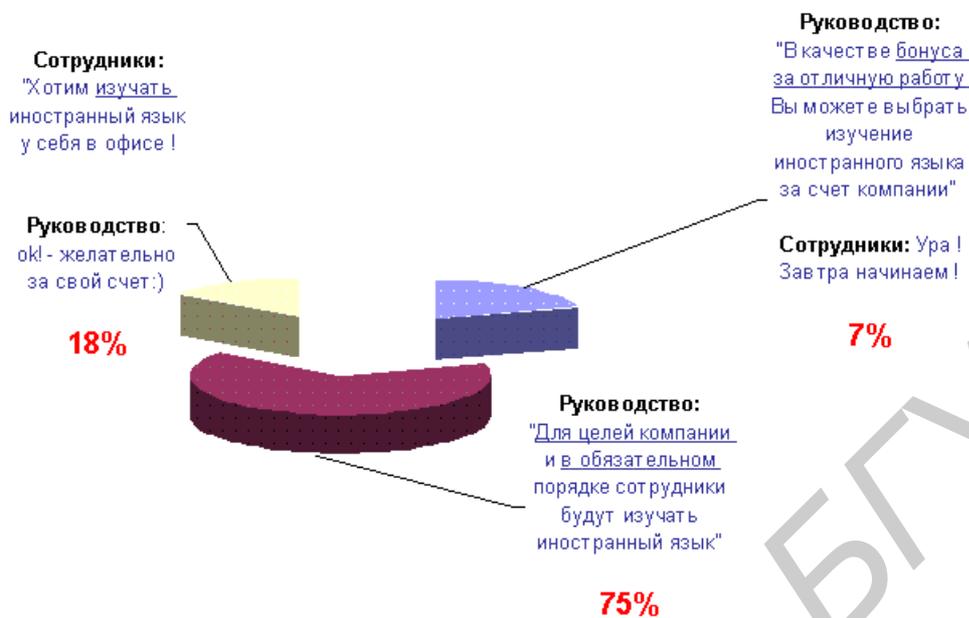


Рисунок 6.5 – Инициаторы изучения иностранных языков в компаниях.

Наиболее очевидным и приемлемым решением является использование автоматизированной системы обучения, а так же системы по управлению образовательным процессом на основе сценарного подхода.

Прежде всего, это обусловлено особенностями работы в ИТ-сфере. Как правило, график работы гибкий и зависит от заказчика проекта. Поэтому работая в одной компании, но на разных проектах с заказчиками из различных стран, сотрудники могут иметь разный график работы. Исходя из этого очевидно, что обучение сотрудников должно быть дистанционным. Это позволит сотрудникам самим выбирать удобное время для обучения не в ущерб основной деятельности и без надобности подстраиваться под кого-либо.

В последние десятилетия стал активно использоваться в управленческой психологии и стратегическом менеджменте «сценарный подход» для составления прогнозов развития тех или иных проектов. В основу сценарного подхода заложена идея о возможности и целесообразности выделения из всего многообразия развития событий наиболее вероятных/встречающихся вариантов, имеющих критически важные сходные характеристики. При этом в качестве сценариев развития событий, как правило, используются «линейные» схемы, предполагающие наличие устойчивых причинно-следственных связей. Это позволяет использовать уже зарекомендованные себя методические разработки и понятийный аппарат для понимания психологических закономерностей, включая закономерности профессионально-личностного становления ИТ-специалистов [48].

Применение сценарного подхода к исследованию профессионального становления будущих ИТ-специалистов позволяет выделять наиболее устойчивые, а значит, предсказуемые линии поведения и реакции обучаемых, возникающие под влиянием ограничений, требований и стимулов.

Эффективность использования сценарного подхода для управления образовательным процессом компании заключается в том, что он предоставляет возможность однажды настроить систему управления путем создания сценария или

последовательности прохождения обучаемым тем и выставления минимального процента усвоения предыдущей темы для перехода к следующей. Созданный сценарий позволяет контролировать процесс обучения и управлять им без вмешательства преподавателя. Обучаемый будет автоматически получать задания и требования к ним при успешном выполнении предыдущих заданий. Или же система будет требовать должного процента усвоения текущего материала для перехода к следующему.

### 6.5 Постановка задачи и обзор методов её решения

Работа над данным проектом направлена на разработку программного средства, которое позволит управлять процессом обучения сотрудников компании, добавлять дополнительные дисциплины для изучения, а так же создавать для них сценарии прохождения. Для повышения эффективности проведения процесса обучения в компании программное средство должно предоставлять следующие функциональные возможности преподавателю-эксперту:

- Добавление новых дисциплин для изучения.
- Добавление списка тем для дисциплин.
- Подключение к серверам, содержащим контент изучаемых тем.
- Создание для каждой дисциплины своего сценария прохождения за счет проверки доступа к теме.

Функциональные возможности системы так же должны предоставлять доступ обучаемым к темам для изучения и запрашивать данные об успеваемости у системы мониторинга.

Функциональный состав системы определяется соответствующими бизнес-требованиями и должен включать следующие особенности:

- удобный и понятный интерфейс для облегчения работы программой, в частности, для создания сценариев;
- возможность работы с данными с использованием локальной базы данных и директории с xml-данными.

Для реализации системы подходят технологии ASP.NET MVC 4 (для представления пользовательского интерфейса) и WCF, где будут размещаться классы бизнес-логики.

Windows Communication Foundation (WCF) это платформа для разработки и развертывании сети распределенной системы. WCF является сервис-ориентированным приложением. Использование WCF позволяет настраивать безопасные, надежные, транзакционные решения, которые являются также кроссплатформенными [49].

WCF позволяет отправлять данные в виде асинхронных сообщений от одной конечной точки службы в другую. Конечная точка службы может быть частью постоянно доступных услуг размещенных на IIS, или это может быть служба, размещенная в приложении. Конечная точка может быть клиентом сервиса, который запрашивает данные из конечной точки службы. Отправляемые сообщения могут быть простыми, как одним символом или словом как XML, или же сложным, как поток двоичных данных [50].

Сценарий для прохождения дисциплины или курса может быть разработан с использованием конечного автомата.

Конечный автомат — абстрактный автомат без выходного потока, число возможных состояний которого конечно. Результат работы автомата определяется по его конечному состоянию.

Все состояния, кроме конечного, должны иметь по крайней мере один переход. Конечное состояние не должно иметь переходов.

Можно выделить 2 способа описания конечного автомата.

Диаграмма состояний (или иногда граф переходов) — графическое представление множества состояний и функции переходов. Представляет собой размеченный ориентированный граф, вершины которого — состояния конечного автомата, дуги — переходы из одного состояния в другое, а метки дуг — символы, по которым осуществляется переход из одного состояния в другое. Если переход из состояния  $q_1$  в  $q_2$  может быть осуществлен по одному из нескольких символов, то все они должны быть надписаны над дугой диаграммы.

Таблица переходов — табличное представление функции  $\delta$ . Обычно в такой таблице каждой строке соответствует одно состояние, а столбцу — один допустимый входной символ. В ячейке на пересечении строки и столбца записывается состояние, в которое должен перейти автомат, если в данном состоянии он считал данный входной символ [51].

Для задания сценария обучения по специальности будем использовать таблицу переходов.

Таблица переходов строится самой системой на основании данных настройки сценария преподавателя-эксперта (рисунок 6.5). Для строки – видно, какие темы доступны после данной темы. Для столбца – видно, какие темы нужно изучить для доступа к данной. Функция перехода представлена, заданными изначально, значениями минимального процента усвоения.

	Тема 1	Тема 2	Тема 3	Тема 4	Тема 5	Тема 6
Тема 1	90%	✓	✓			
Тема 2		86%		✓	✓	
Тема 3			70%	✓	✓	
Тема 4				95%	✓	✓
Тема 5					83%	✓
Тема 6						67%

Рисунок 6.6 – Таблица переходов внутри дисциплины

## 6.6 Спецификация системы

Рассмотрим UML-диаграммы, построенные в Enterprise Architect. На рисунке 6.7 представлена диаграмма вариантов использования автоматизированной системы управления процессом на основе сценарного подхода.

На диаграмме вариантов использования представлены основные функциональные возможности разрабатываемого программного средства. Функционал системы доступен для пользователей двух ролей «Преподаватель-эксперт» и «Обучаемый».

«Преподаватель-эксперт» имеет доступ к функционалу непосредственно по настройке системы и созданию необходимых объектов системы для её функционирования. Рассмотрим варианты использования подробнее.

Вариант использования «Авторизоваться» предоставляет доступ к системе только преподавателю-эксперту, который управляет системой. Логин и пароль один для всех, хранится в конфигурационном файле.

Вариант использования «Просмотреть список дисциплин» позволяет просмотреть все доступные дисциплины, их темы, а так же расширится возможностью редактировать или удалять дисциплину или темы.

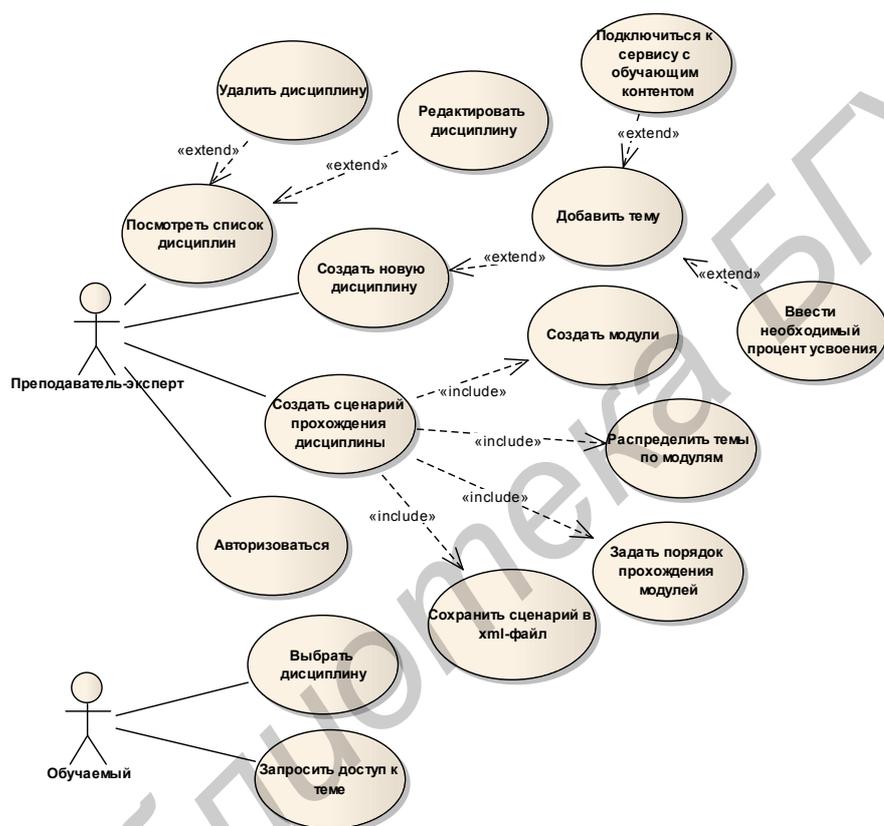


Рисунок 6.7 – Диаграмма вариантов использования автоматизированной системы управления процессом на основе сценарного подхода

Вариант использования «Создать новую дисциплину» обеспечивает набор возможностей по созданию новой дисциплины и созданию списка тем. При создании новой темы необходимо ввести её название, необходимый процент усвоения и подключиться к сервису с контентом, т.е. ввести адрес сервиса для получения самой темы к изучению. Дисциплина сохраняется как в базу данных, так и в xml-файл.

Вариант использования «Создать сценарий прохождения дисциплины» предусматривает операции по созданию сценария, которые включают в себя следующие варианты использования «Создать модули», «Распределить темы по модулям», «Задать порядок прохождения модулей», «Сохранить сценарий в xml-файл».

«Обучаемый» используя автоматизированную систему управления процессом на основе сценарного подхода имеет возможность выбрать дисциплину для изучения и получить доступ к изучению темы.

Вариант использования «Получить доступ» подразумевает проверку системой успеваемости обучаемого, его процента усвоения текущей темы, на основании которого делается вывод о предоставлении доступа к следующей или запрашиваемой теме.

### 6.7 Модели представления системы и их описание

Для моделирования данного проекта были построены следующие диаграммы: диаграмма последовательности, диаграмма состояний, диаграмма классов, диаграмма развертывания, диаграмма компонентов.

На рисунке 6.8 представлена диаграмма состояний темы, которая составляет часть обучающего контента дисциплины.

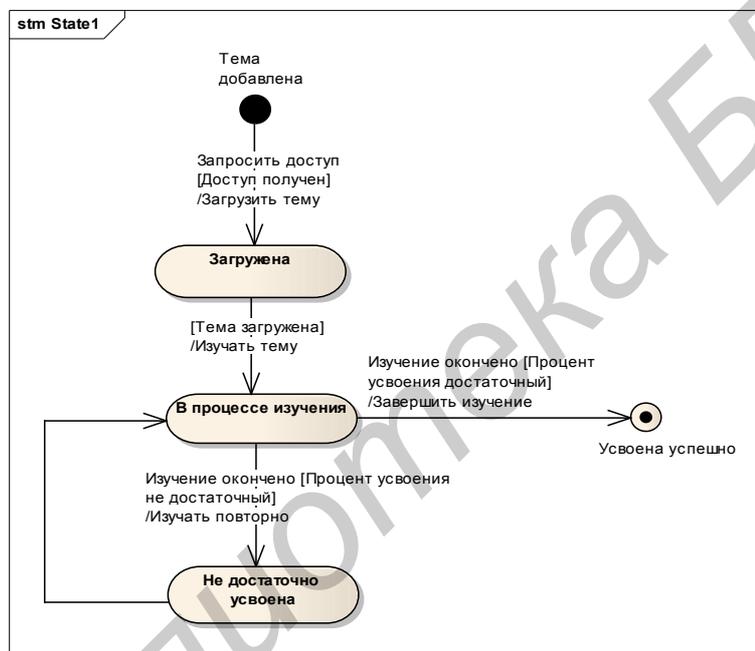


Рисунок 6.8 – Диаграмма состояний объекта «Тема»

После добавления новой темы она переходит в состояние «Добавлена», т.е. теперь можно делать запрос на её изучение. Если право доступа предоставлено, тема приобретает состояние «Загружена» и она готова к изучению, в противном случае нет доступа и тема не переходит ни в какое состояние. После загрузки тема идет на изучение и переходит в состояние «В процессе изучения». После завершения изучения есть два возможных состояния перехода в зависимости от её усвоения. Если процент усвоения больше либо равен необходимому, то тема переходит в конечное состояние «Усвоена успешно», иначе переходит в состояние «Не достаточно усвоена» и идет на процесс повторного изучения, после завершения которого переходит в состояние «Повторно изучена».

На рисунке 6.9 представлена диаграмма классов бизнес-логики системы.

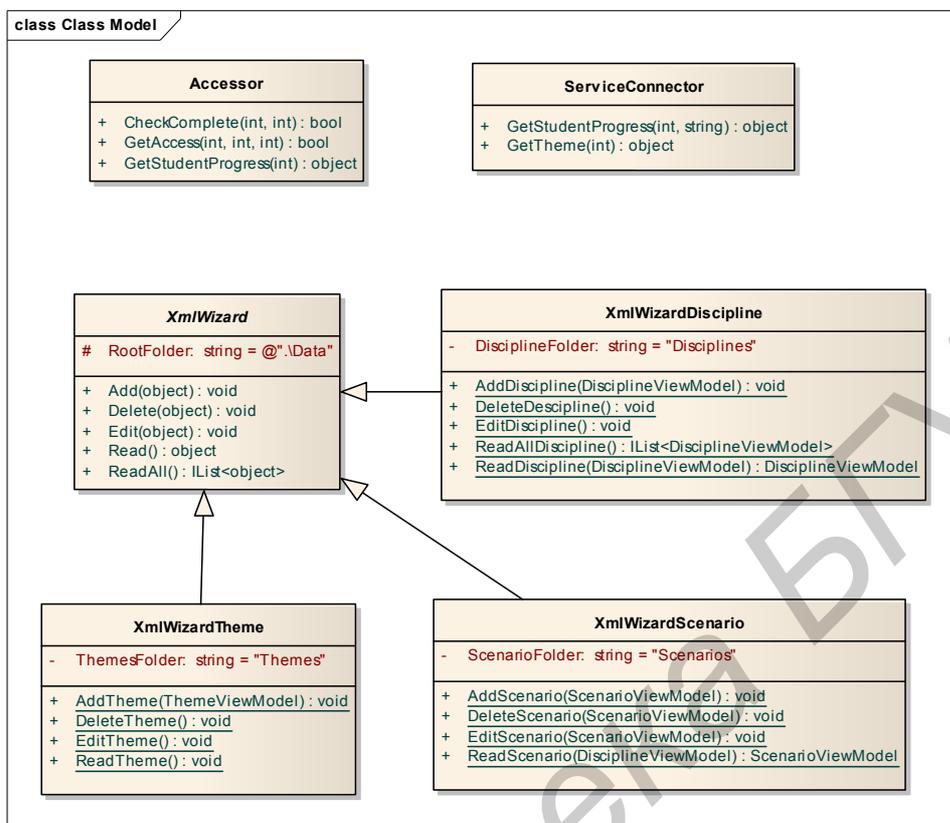


Рисунок 6.9 – Диаграмма классов бизнес-логики системы

На диаграмме классов изображены основные классы бизнес-логики системы, которые позволяют формировать xml-данные для записи в файл. А так же классы для соединения с сервисами и предоставления доступа к данным.

На рисунке 6.10 представлена диаграмма компонентов системы, отображающая взаимодействие основных модулей системы. На диаграмме компонентов показано взаимодействие основных модулей шаблона MVC с остальными программными модулями системы.

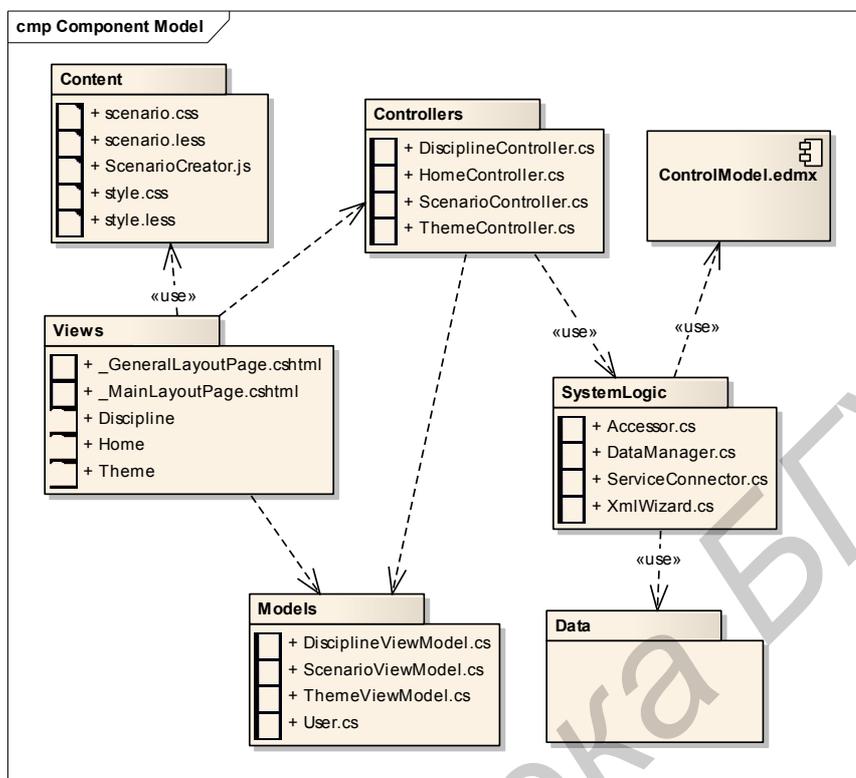


Рисунок 6.10 – Диаграмма компонентов системы

На рисунке 6.11 представлена диаграмма развертывания системы, на которой представлена конфигурация обрабатывающих узлов системы и размещенных в них компонентов.

Приложение выполняется на компьютере с ОС Windows 7 Ultimate, где должен быть установлен .NET Framework версии 4.5. База данных хранится на Microsoft SQL Server 2008. Веб-приложение и WCF-сервис разворачиваются на Internet Information Server (IIS).

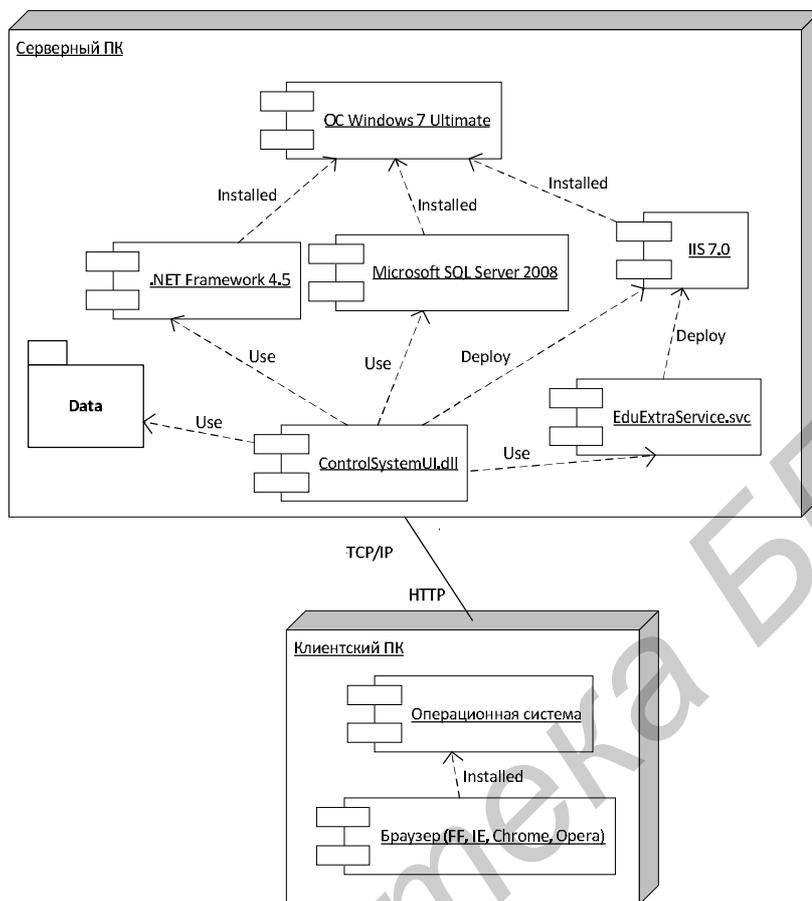


Рисунок 6.11 – Диаграмма развертывания системы

На рисунке 6.12 отображена диаграмма последовательности добавления сценария для существующей дисциплины. Для входа в систему необходимо авторизоваться, данные, введенные преподавателем-экспертом, проверяются путем сравнения с логином и паролем, записанными в конфигурационном файле. При успешной авторизации система возвращает окно стартовой страницы. На данной странице будут представлены основные действия, которые можно выполнить над дисциплинами, а так же списки всех ранее созданных дисциплин, хранящихся в базе.

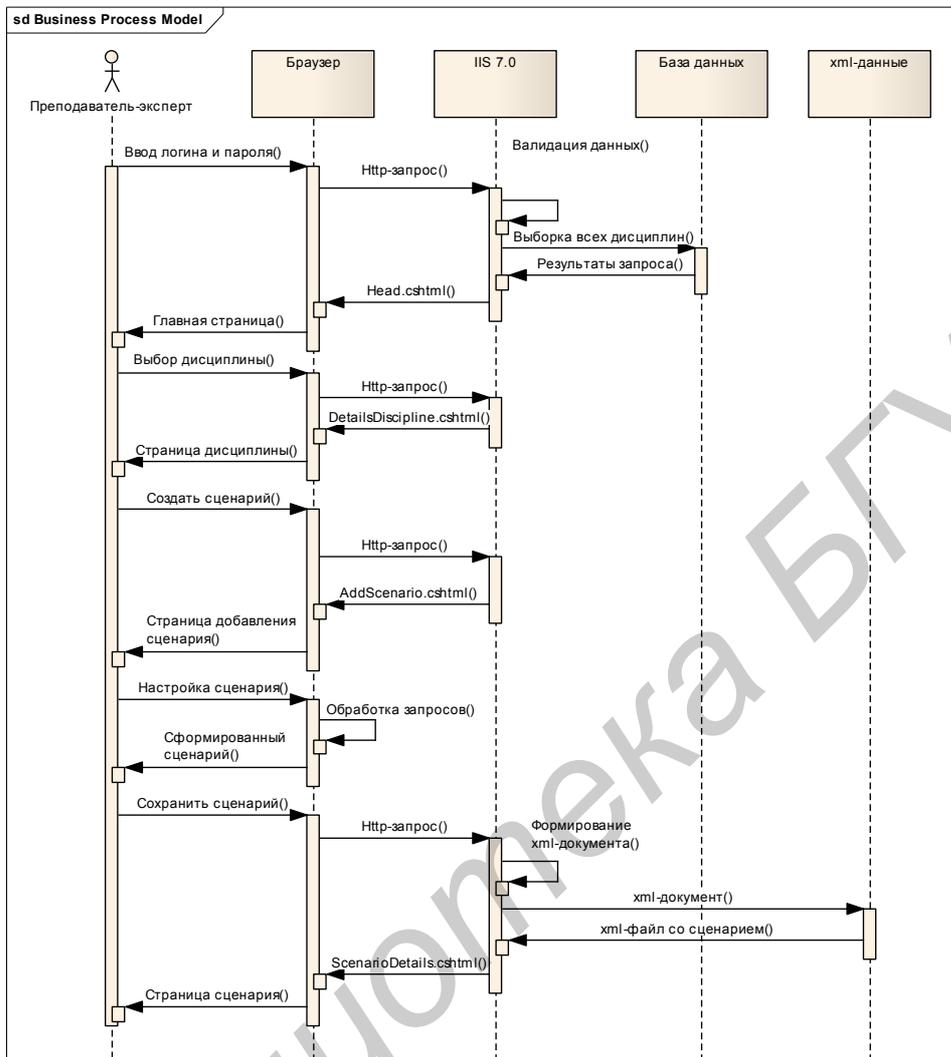


Рисунок 6.12 – Диаграмма последовательности добавления сценария

Для работы с какой-либо дисциплиной, необходимо выбрать её из списка и нажать кнопку “Править”. После чего браузер сформирует http-запрос, содержащий объект дисциплины и отправит его к IIS, который обработает его и вернет страницу с данными о дисциплине, где нужно выбрать действие по созданию сценария, IIS сформирует страницу для настройки сценария. Пользователь настраивает сценарий нужным образом, после чего сохраняет. Объект сценария обрабатывается WCF-сервисом, который так же развернут на IIS, и сохраняет xml-документ в директорию с xml-данными. После сохранения пользователю отображается страница с сохраненным сценарием.

## 6.8 Информационная модель системы и описание

Информационная модель системы состоит из структуры базы данной и xsd-схемы файла хранения сценария.

Физическая модель базы данных представлена на рисунке 6.13.

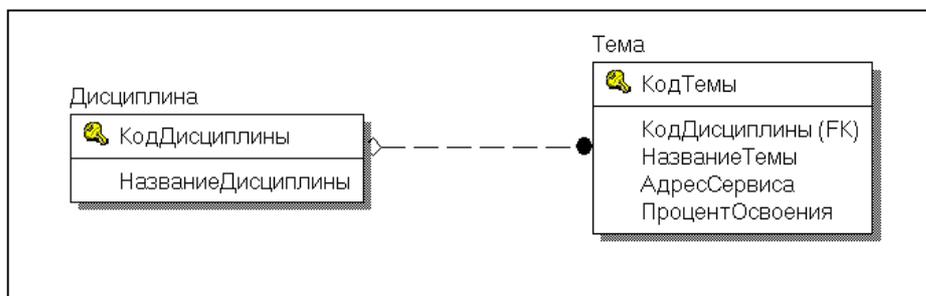


Рисунок 6.13– Физическая модель базы данных

База данных хранит в себе информацию о созданных дисциплинах и темах, связи между таблицами поддерживаются на уровне бизнес-логики. Таблица «Дисциплины» содержит в себе идентификатор и название дисциплины. Таблица «Темы» содержит идентификатор темы, название, адрес к сервису с контентом, необходимый процент усвоения и идентификатор дисциплины, к которой принадлежит тема.

Диаграмма схемы xml-файла, используемая для записи сценария прохождения специальности представлена на рисунке 3.10.

Корневой элемент «scenario» содержит в себе название дисциплины и список модулей. Каждый модуль также является контейнером, который хранит в себе элементы тем и свой номер. Контейнер темы соответствует объекту темы и хранит данные о названии, адресе сервиса и необходимом проценте завершения.

## 6.9 Руководство пользователя

Система состоит из веб-приложения и сервиса. Все операции выполняются на сервисе и если обучаемому необходимо будет получить доступ к теме, то его запрос непосредственно будет направлен на сервис. Веб-приложение является интерфейсом для настройки сервиса: выполнения операций над дисциплинами и темами, настройкой сценария. Такой доступ предоставляется только преподавателю-эксперту и поэтому первоначально он должен пройти авторизацию. Окно авторизации показано на рисунке 6.14.

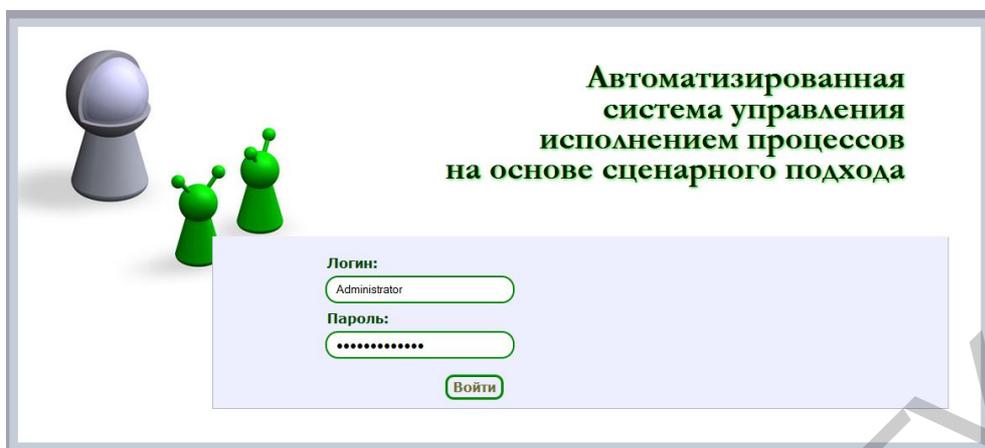


Рисунок 6.14 – Страница авторизации пользователя

После авторизации преподаватель попадает на главную страницу, которая представляет собой список уже имеющихся дисциплин в системе (рисунок 6.15). Он может либо работать с дисциплиной из списка или же добавить новую дисциплину.

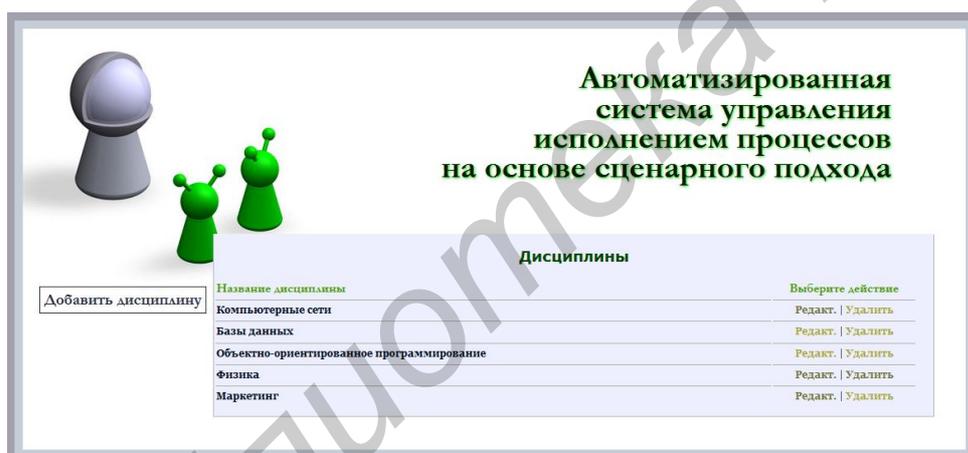


Рисунок 6.15 – Стартовая страница системы

Если преподавателю необходимо добавить новую дисциплину, он выбирает действие «Добавить дисциплину» и попадает на страницу добавления дисциплины, где необходимо ввести название дисциплины и нажать кнопку сохранения (рисунок 6.16).

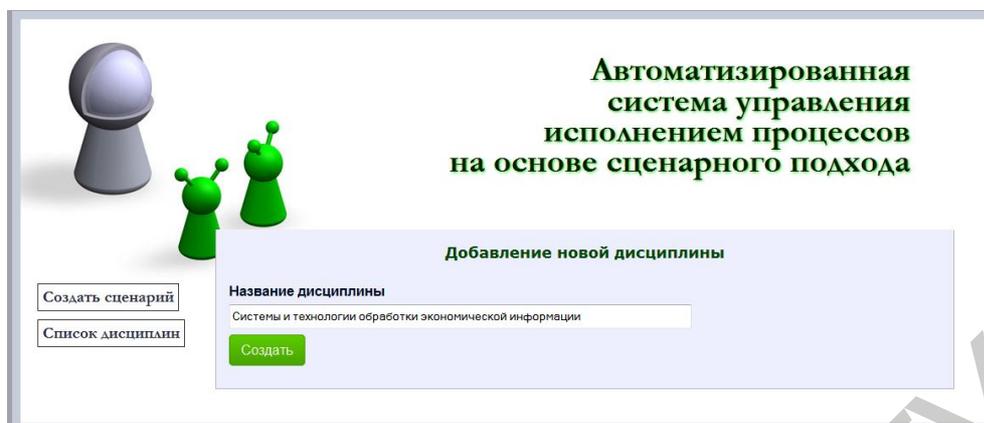


Рисунок 6.16 – Страница добавления дисциплины

После добавления дисциплины преподаватель перенаправляется на страницу просмотра с возможностью выбора дальнейшего действия над дисциплиной в меню слева (рисунок 6.17).

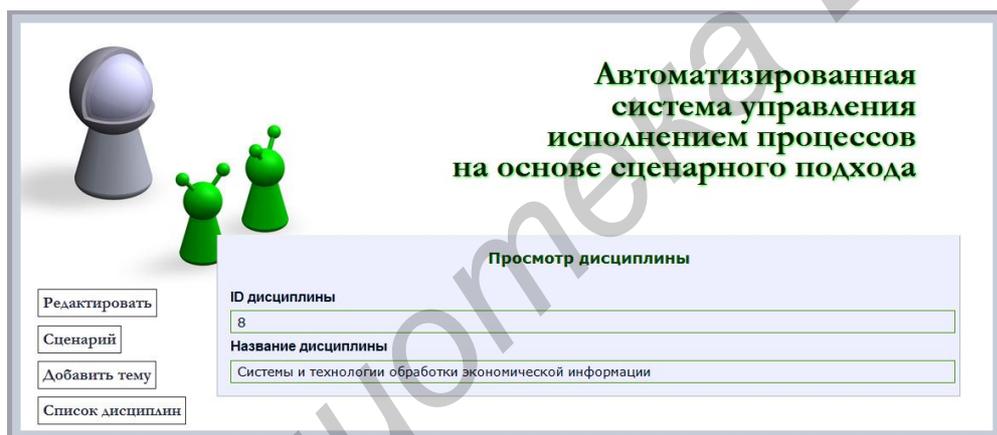


Рисунок 6.17 – Страница просмотра добавленной дисциплины

Далее необходимо добавить темы этой дисциплины. Выбираем действие «Добавить тему» и переходим на страницу добавления темы (рисунок 6.18). Здесь необходимо ввести название темы, адрес сервиса, который содержит эту тему и обязательный процент усвоения.

Рисунок 6.18– Страница добавления темы

После того как преподаватель добавил весь список тем дисциплины, страница просмотра дисциплин выглядит как показано на рисунке 6.19

Название темы	Адрес сервиса	Обязательный процент усвоения	Выберите действие
Процессный подход. Типы процессов.	http://localhost:8010/EducationLibrary/STOEI/ProcessApproach.html	87	Редакт.   Удалить
Методы сбора информации и выявления и получения требований.	http://localhost:8010/EducationLibrary/STOEI/MethodsCollectionsInform.html	78	Редакт.   Удалить
Процесс разработки требований	http://localhost:8010/EducationLibrary/STOEI/RequirementsProcess.html	95	Редакт.   Удалить
Способы описания бизнес-процессов	http://localhost:8010/EducationLibrary/STOEI/WaysOfDescribingBusinessProcesses.html	93	Редакт.   Удалить
Суть, понятие и формы представления требований.	http://localhost:8010/EducationLibrary/STOEI/PresentationRequirements.html	74	Редакт.   Удалить
Документирование бизнес-правил. Атрибуты качества.	http://localhost:8010/EducationLibrary/STOEI/DocumentingBusinessRules.html	86	Редакт.   Удалить
User Stories как требования к ПО.	http://localhost:8010/EducationLibrary/STOEI/UserStoriesAsRequirements.html	91	Редакт.   Удалить

Рисунок 6.19 – Просмотр дисциплины после добавления списка тем

Если дисциплина имеет свой список тем можно задавать сценарий их прохождения. Выбираем в левом меню пункт «Сценарий» и попадаем на страницу создания сценария (рисунок 6.20). На странице слева расположен столбик со списком доступных дисциплин, справа область разбита на шесть модулей. Для добавления дисциплины в какой-либо модуль используется технология «Drag-and-Drop». Преподаватель по своему усмотрению растаскивает темы по модулям так, что любую тему в модуле можно изучать независимо от остальных, но чтобы перейти к следующему модулю необходимо усвоить все темы из предыдущего модуля.

### Создание сценария

Доступные темы	Модуль 1	Модуль 2	Модуль 3
Использует ли требования к ИО.	Процессный подход. Типы процессов.	Способы описания бизнес-процессов	Образ и границы продукта в контексте требований.
Связь. Вариантов использования и требований.	Суть, понятие и формы представления требований.	Процесс разработки требований	Управление требованиями и их изменениями.
Итерационная и итеративная модели разработки ИО.	Методы сбора информации и выявления и получения требований.	Техника структуры спецификации для ведения требований.	Модели разработки ИО.
Основные подходы к оценке объема работ.	Документирование бизнес-правил. Атрибуты качества.	Анализ и переработка требований.	
Основные этапы процесса функционального тестирования.			
Тестовые сценарии и подходы к их созданию.			
Требования к команде Agile.			
	<b>Модуль 4</b>	<b>Модуль 5</b>	<b>Модуль 6</b>
	Методология XP.	Взаимодействие с заказчиком в процессе выполнения проекта по Agile.	
	RUP, основные фазы, работы, артефакты.	Организация процесса тестирования в Agile.	
	Базовые принципы методологии Agile.		

[Вернуться к дискуссии](#) Сохранить

Рисунок 6.20– Страница создания сценария

## ЗАКЛЮЧЕНИЕ

В результате проведенных исследований были подготовлены и изданы за период с 2012 г. по 2014 г. 2 монографии, 21 учебных и учебно-методических пособий, 36 статей, 69 материалов конференций, 10 тезисов докладов на научных конференциях, около 30 переработанных и внедренных в учебный процесс электронных учебно-методических комплексов дисциплин.

Проведенный анализ современных тенденций развития методов и инструментальных средств моделирования и разработки информационных, социально-экономических и технических систем, а также ряд исследований смежных предметных областей и работа с синтезированными экспериментальными моделями позволяют сформулировать следующие достигнутые результаты и выводы:

1) Выделены и формализованы в виде абстрактных схем основные процессы деятельности кафедры. Выделены основные типы процессов, определены единицы функциональной нагрузки процессов, определена оценка процессов, представлены основные результаты деятельности по системным процессам, предложены методики и модели разработки курсового и дипломного проектирования, выделены этапы выполнения и сформулированы цели и задачи по каждому из них, а также методика оценки результатов работы студента по каждому из выполняемых этапов, формализованы диаграммы выполнения и защиты курсового и дипломного проектирования. Представлена модель программного комплекса поддержки виртуальной обучающей среды, разработана модель и основные алгоритмы виртуальной обучающей среды на основе Петри. Представлено руководство пользователя по данной инструментальной среде.

2) Проведен анализ перспективных методик преподавания экономических дисциплин специальности, в частности, перспективная методика преподавания дисциплины «Эконометрика». Рассмотрены вопросы адаптации процесса обучения к индивидуальным особенностям обучаемого. Разработана и представлена методика лабораторных работ по дисциплине «Эконометрика» для студентов первой и второй ступени высшего образования.

3) Проведены исследования организационных процессов в социально-экономических системах. Выделены и проведен анализ четыре категории операционных услуг, определены основные этапы исследований работоспособности социально-экономической системы. Проведены исследования эффективных методик построения моделей работоспособности структурных функций, в частности разработки структурной функции описания кредитоспособности заемщика на основе выбранной системы показателей и декомпозиции исходной функции. Выполнен анализ чувствительности структурных функций с помощью направленных логических производных. Показана связь направленной логической производной функции  $m$ -значной логики и DD-куба для  $m$ -уровневого компонента, реализующего эту функцию. Определена методика анализа чувствительности структурной функции, реализованной на  $m$ -уровневых компонентах., представлен пример использования теоретических выкладок на практике разработки структурной функции процесса проведения олимпиады «Созвездие талантов»

4) Представлен ряд результатов для решения частных задач, связанных с подготовкой топологической информации для микрофотонаборных генераторов изображений, а именно – описаны методы представления топологических объектов (многоугольников) в виде булевых формул. Для оптимально раскрытия, рассмотрен ряд методов поиска крайних вершин многоугольника, подготовки топологической информации для микрофотонаборных генераторов изображений, а именно – описаны методы представления топологических объектов (многоугольников) в виде булевых формул. Представлены и проведен анализ способов поиска множества крайних вершин, описана формализованная процедура поиска, определен метод нахождения булевой

формулы многоугольника в ДНФ, а также метод базовый метод нахождения булевой формулы многоугольника в ДНФ без использования дополнительных предикатных переменных. Разработан модифицированный, усовершенствованный метод нахождения булевой формулы многоугольника. Разработан комплекс программ минимизации систем частичных булевых функций и приведены доказательства эффективности его применения.

5) Представлены результаты разработки и применения моделей процессов усвоения и методов извлечения знаний для управления качеством программных средств. Для этого разработан ряд обобщенных интеллектуальных методов для определения решений задач, оценивания качества программ и формирование общих принципов управления качеством на этапах создания программных средств. Предложен ряд интеллектуальных методов и средства оценивания результатов проектирования и прогнозирования характеристик качества ПО.

6) Предложен метод программная поддержка управления исполнением процессов обучения на основе сценариев. Рассмотрен процесс, выявлены недостатки традиционного управления процессом обучения в образовательной сфере, определено и обосновано место сценарного подхода в управлении. Определены факторы, влияющие на успех учебного процесса и усвоения знаний. Определена спецификация, выполнен процесс проектирования и программной реализации Автоматизированной системы управления исполнением процессов на основе сценарного подхода. Разработано и представлено руководство пользователя, для внедрения программного продукта в учебный процесс.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. БГУИР – Кафедра экономической информатики. Учебная работа кафедры. – [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://www.bsuir.by/online/showpage.jsp?PageID=76358&resID=100229&lang=ru&menuItemID=102765>
2. БГУИР – Кафедра экономической информатики. Научная работа. – [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://www.bsuir.by/online/showpage.jsp?PageID=76357&resID=100229&lang=ru&menuItemID=102765>
3. Репин В.В., Елиферов В.Г. Процессный подход к управлению. Моделирование бизнес-процессов
4. Бизнес-процессы: основные понятия. [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://www.elitarium.ru/>
5. Бизнес-процессы. Подходы к оптимизации, моделирование и реинжиниринг. [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://www.informicus.ru/default.aspx?SECTION=4&id=92>
6. Процессы: ключевые или сквозные? [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://www.management.com.ua/cm/cm141.html>
7. ARIS Toolset/BPwin: выбор за аналитиком. [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://compress.ru/article.aspx?id=9480&part=tab51ext1>
8. Котов В.Е. Сети Петри.– М.: Наука. Главная редакция физико-математической литературы, 1984.– 160 с.
9. Музыкачина Т. М. Описание бизнес-процессов организации сетями Петри при внедрении и сертификации QM-систем // "НИРС-2003": VIII Республиканская научно-техническая конференция студентов и аспирантов, 9–10 декабря 2003г., г. Минск.– Мн.: БНТУ, 2003 г.– С. 79.;
10. Музыкачина Т. М. Планирование множества допустимых сетевых планов системы бизнес-процессов предприятия после проведения реинжиниринга // Известия белорусской инженерной академии. № 1 (17)/3' 2004.– С. 171–174.
11. Унучек Т.М., Унучек Е.Н. Моделирование поведения бизнес-процессов при проведении реинжиниринга с применением аппарата сетей Петри и методологии SADT// Региональная экономическая политика: материалы Междунар. науч.-практ. конф., Гродно, 4–5 мая 2006 г.: в 2 ч. Ч.1 / Гродн. гос. ун-т; редколлегия: Ли Чон Ку [и др.]. – Гродно: ГрГУ, 2007. – 321 с.
12. Унучек Т.М. Моделирование состояний бизнес-процессов при проведении реинжиниринга на основе аппарата сетей Петри // Управление в социальных и экономических системах: Материалы XV международной научно-практической конференции (6 июня 2006 г., г. Минск)/ Редкол.: Н.В. Суша (пред.) и др.; Минский ин-т управления.– Мн. Изд-во МИУ, 2006.– 304 с.
13. Левашенко В., Зайцева Е., Ковалик Ш. Принятие решений на основе нечетких данных. Основы, методы и алгоритмы / В. Левашенко, Е. Зайцева, Ш. Ковалик – LAMBERT Academic Publishing, 2014 – 369 p.
14. Barlow R., Wu A. Coherent system with multistate elements / R. Barlow, A. Wu // Mathematics of Operations Research. – 1978. – Vol.3. – № 11. – P.275-281.
15. Murchland, J.D. Fundamental Concepts and relations for Reliability Analysis of Multistate System, Reliability and Fault Tree Analysis / J.D. Murchland // Theoretical and Applied Aspects of System Reliability. – SIAM, 1975. – P.581-618.
16. Zaitseva E., Levashenko V. Decision Diagrams for Reliability Analysis of Multi-State System / E. Zaitseva, V. Levashenko // Proc. of the Int. Conf. on Dependability of Computer System (DepCoS-RELCOMEX'08), 26-28 Jun. – Szklarska Poreba, Poland, 2008. – P.55-62.

17. Xing L., Dai Y. A new decision diagram based method for efficient analysis on multi-state systems / L. Xing, Y. Dai // IEEE Trans. on Dependable and Secure Computing, 2009. – Vol.6. – №3. – P.161–174.
18. Liu H., Hussain F., Lim Tan C., Dash M. Discretization: An Enabling Technique / H. Liu, F. Hussain, C. Lim Tan, M. Dash // Data Mining and Knowledge Discovery, Kluwer Academic Publishers. – 2002 – №6 – pp. 393–423.
19. Левашенко В.Г., Козлова И.К., Поттосина С.А. Использование многозначной логики для анализа данных в системах поддержки принятия решений / В.Г. Левашенко, И.К. Козлова, С.А. Поттосина // Доклады БГУИР – 2008 – №1 (31) – С.115-123.
20. Van de Merckt T. Decision trees in numerical attribute spaces. / T. Van de Merckt // Machine Learning – 1990 – pp.1016–1021.
21. Koga M., Uchiyama A., and Sampei M. A transformation of discrete-time systems into continuous-time systems using the general solution of a matrix equation. // A Publication of Elect. Infor. and Systems Society – 1999 - 119-C(12): 1561
22. Фейнберг В.З. Геометрические задачи машинной графики больших интегральных схем // М.: Радио и связь 1987.-178 с.
23. Шестаков Е.А., Бутов А.А., Орлова Т.Л., Воронов А.А. Автоматизированная система подготовки информации для формирования фотошаблонов. Искусственный интеллект, Украина, Донецк, № 4, с. 200-207, 2008.
24. Ласло М. Вычислительная геометрия и компьютерная графика на C++ / Пер. с англ. М.: БИНОМ, 1997. 304 с.
25. Препарата Ф., Шеймос М. Вычислительная геометрия: Введение / Пер. с англ. М.: Мир, 1989. 478 с.
26. Никулин Е.А. Компьютерная геометрия и алгоритмы машинной графики – СПб.: БКХ-Петербург, 2005. – 576 с.
27. Закревский А.Д. Канонические булевы формулы многоугольников // Информатика. – 2009. – № 2. – С. 93-101.
28. Поттосин Ю.В., Шестаков Е.А. Использование булевых функций для представления многоугольников // Вестник Томского Государственного университета. Управление, вычислительная техника и информатика. – 2008. – № 2(3). – С. 106-115.
29. Бутов А.А. Простой метод нахождения булевой формулы многоугольника в дизъюнктивной нормальной форме // Вестник Брестского Государственного Технического Университета. Физика, Математика, Информатика. – 2011. – № 5. – С. 35-38.
30. Бутов А.А. Метод нахождения булевой формулы многоугольника в дизъюнктивной нормальной форме без использования дополнительных предикатных переменных // Вестник Брестского Государственного Технического Университета. Физика, Математика, Информатика. – 2012. – № 5. – С. 48-51.
31. Бутов А.А. Метод построения булевой формулы многоугольника в дизъюнктивной нормальной форме // Информатика. – 2014. – № 4. – С. 117-124.
32. Бутов А.А. К задаче нахождения канонической булевой формулы многоугольника // Вестник Брестского Государственного Технического Университета. Физика, Математика, Информатика. – 2010. – № 5. – С. 29-31.
33. Quine W.V. The problem of simplyfying truth functions. – American Mathematical Monthly, 1952, v.59, No.6.
34. Mc Cluskey E. Minimization of Boolean function. – The Bell System Technical Journal, 1956, No.6.
35. Закревский А.Д. О приближенных методах решения логических задач. В кн.: Проблемы синтеза цифровых автоматов, М.: Наука, 1967.
36. Закревский А.Д. Алгоритмы синтеза дискретных автоматов. – М.: Наука, 1971.
37. Поспелов Д.А. Логические методы анализа и синтеза схем. – М.: Энергия, 1974.
38. Синтез асинхронных автоматов на ЭВМ. Под общ. ред. А.Д. Закревского. – Мн.: Наука и техника, 1975.

39. Погарцев А.Г. Новые алгоритмы совместной минимизации булевых функций. – Автоматика и вычислительная техника, 1980, № 1.
40. Агибалов Г.П., Беляев В.А. Технология решения комбинаторно-логических задач методом сокращенного обхода дерева поиска. – Томск, 1981.
41. Закревский А.Д. Логический синтез каскадных схем. – М.: Наук
42. Большая онлайн библиотека [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://www.e-reading.co.uk/chapter.php/98177/27/>
43. Серина М.С. Автоматизированная система управления кафедрой. / Серина М.С. // Материалы XX Международной конференции «Применение новых технологий в образовании», 26 – 27 июня 2009г., г.Троицк, Московской области - ГОУ ДПО "Центр новых педагогических технологий" Московской области, МОО Фонд новых технологий в образовании «Байтик». С. 513-514.
44. Модели смешанного обучения. Режим доступа: <http://www.distance-learning.ru/db/el/F7DC8A6A9C0F4F70C3257216003F76D0/doc.html>
45. Управление информационной инфраструктурой образовательного процесса. Режим доступа: [http://www.masu-inform.ru/datanews/economy\\_magazine/page27.htm](http://www.masu-inform.ru/datanews/economy_magazine/page27.htm)
46. Новиков Д.А. Математические модели формирования и функционирования команд / Д.А. Новиков. - М.: Издательство физико-математической литературы, 2008. – 184 с.
47. Трифильцева Н. Мы все учились понемногу... / Н. Трифильцева // Управление персоналом. – 2002. – N 8. – С. 38-39.
48. Малюченко Г.Н., Плоткина Л.Н. Применение сценарного подхода к исследованию закономерностей профессионально-личностного становления будущих ИТ-специалистов. Журнал: Вестник Самарской гуманитарной академии. Серия: Психология. Выпуск №2/2010
49. Пабло Сибраро, Курт Клайс, Фабио Косолино, Йохан Грабнер WCF 4: Windows Communication Foundation и .NET 4 для профессионалов = Professional WCF 4: Windows Communication Foundation with .NET 4. — М.: «Диалектика», 2011. — С. 464.
50. Web Code Expert [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://www.webcodeexpert.com/2013/04/how-to-create-and-consume-wcf-services.html>
51. Дехтярь М. И. Введение в схемы, автоматы и алгоритмы. – Москва, 2010.