

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ
Харьковский национальный университет радиоэлектроники

На правах рукописи

Евланов Максим Викторович

УДК 044.03; 681.518:061

**МОДЕЛИ, МЕТОДЫ И ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ
РАЗРАБОТКИ АРХИТЕКТУРЫ СЛОЖНЫХ ИНФОРМАЦИОННЫХ
СИСТЕМ НА ОСНОВЕ ФУНКЦИОНАЛЬНЫХ ТРЕБОВАНИЙ**

05.13.06 – информационные технологии

Диссертация на соискание ученой степени доктора технических наук

Научный консультант
Левыкин Виктор Макарович,
доктор технических наук, про-
фессор

Этот экземпляр диссертации идентичен по смыслу с другими, что
поданы в специализированный ученый совет Д 64.052.08

Ученый секретарь спецсовета

Плисс И.П.

Харьков – 2016

СОДЕРЖАНИЕ

Перечень условных обозначений, символов, единиц, сокращений и терминов.....	6
Введение.....	8
1. Анализ проблем формирования требований и создания описаний архитектуры сложных информационных систем	20
1.1. Анализ современного представления информационной системы...	20
1.2. Анализ современных представлений требований к информационной системе.....	27
1.2.1. Анализ определений понятия «требование к системе».....	27
1.2.2. Анализ существующих классификаций требований к информационной системе.....	29
1.2.3. Анализ методов сбора требований к информационной системе.....	30
1.2.4. Анализ моделей требований к информационной системе.....	32
1.2.5. Анализ подходов к управлению требованиями к информационной системе.....	35
1.3. Анализ современных способов описания архитектуры информационной системы.....	37
1.4. Анализ моделей и методов определения объема работ по созданию информационной системы.....	45
1.5. Анализ современных систем управления требованиями.....	51
1.6. Выводы из результатов проведенного анализа и постановка задач исследования.....	57
2. Методологические основы сервисного подхода к созданию информационной системы.....	69
2.1. Сервисный подход к созданию информационных систем:	

основные термины.....	69
2.2. Требования к информационной системе: определение и классификация.....	78
2.3. Концепция представления требований к информационной системе.....	83
2.4. Формализованное описание универсума требований к информационной системе.....	93
2.5. Обобщенное формализованное описание одноместного ковариантного функтора.....	101
2.6. Модель паттерна проектирования требований к информационной системе.....	107
2.7. Выводы ко второму разделу.....	115
3. Модели архитектурного фреймворка макропроектирования информационной системы.....	118
3.1. Обобщенная модель архитектурного фреймворка макропроектирования информационной системы.....	118
3.2. Разработка паттернов проектирования требований к информационной системе на уровне данных.....	127
3.3. Разработка паттернов проектирования требований к информационной системе на уровне информации.....	147
3.4. Разработка паттернов проектирования требований к информационной системе на уровне знаний.....	164
3.5. Выводы к третьему разделу.....	183
4. Модели и методы синтеза рациональной архитектуры информационной системы.....	185
4.1. Задача синтеза описания рациональной архитектуры информационной системы.....	185
4.2. Методы формирования представления сформулированного требования к информационной системе на уровне знаний.....	196

4.2.1. Метод формирования представления сформулированного требования на уровне информации.....	196
4.2.2. Метод формирования представления сформулированного требования на уровне знаний отдельного участника автоматизируемого процесса	198
4.2.3. Метод формирования представления сформулированного требования на уровне знаний Потребителя	200
4.2.4. Метод формирования представления сформулированного требования на уровне знаний Поставщика	203
4.2.5. Метод формирования общесистемного представления сформулированного требования на уровне знаний	207
4.3. Метод синтеза вариантов описания архитектуры создаваемой информационной системы	209
4.4. Модель и методы синтеза описания рациональной архитектуры информационной системы	215
4.5. Оценка объема работ по реализации описания рациональной архитектуры информационной системы	223
4.6. Методы анализа сформулированных функциональных требований к информационной системе.....	238
4.7. Выводы к четвертому разделу.....	245
5. Информационная технология формирования и анализа требований к информационной системе.....	247
5.1. Описание архитектуры и основных IT-услуг информационной технологии.....	247
5.2. Хранение и обработка представлений требований на уровне информации.....	259
5.3. Хранение и обработка представлений требований на уровне знаний.....	266
5.4. Хранение и обработка представлений требований на уровне	

данных.....	274
5.5. Выводы к пятому разделу.....	280
6. Программная реализация и апробация информационной технологии формирования и анализа требований к информационной системе.....	281
6.1. Реализация информационной технологии формирования и анализа требований к информационной системе.....	281
6.2. Описание требований и решений, ре-используемых в ходе апробации информационной технологии.....	291
6.3. Апробация информационной технологии на примере функционального модуля безопасности труда комплексной системы управления ПАО «Харьковгоргаз».....	302
6.4. Выводы к шестому разделу.....	325
Выводы.....	327
Перечень ссылок.....	331
Приложение А. Акты о внедрении результатов диссертации.....	357
Приложение Б. Исследование способов представления требований к информационной системе.....	367
Приложение В. Теоретико-категорные модели элементов универсума требований к информационной системе.....	382
Приложение Г. Модели паттернов проектирования версий требований... ..	384
Приложение Д. Примеры реализации результатов диссертационной работы.....	388
Приложение Е. Визуальные модели информационно-аналитической системы «Реестр Лиги украинских клубов интеллектуальных игр».....	397
Приложение Ж. Промежуточные результаты выполнения информационной технологии.....	419

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ, СИМВОЛОВ, ЕДИНИЦ,
СОКРАЩЕНИЙ И ТЕРМИНОВ

АС	– автоматизированная система;
АФ	– архитектурный фреймворк;
БД	– база данных;
БП	– бизнес-процесс;
БТ	– безопасность труда;
ВМ	– визуальная модель;
ВПФ	– вредный производственный фактор;
ВОО	– Всеукраинская общественная организация;
ДК	– диаграмма классов;
ЖЦ	– жизненный цикл;
ИАС	– информационно-аналитическая система;
ИО	– информационное обеспечение;
ИС	– информационная система;
ИТ	– информационные технологии;
КСА	– комплекс средств автоматизации;
ЛУК	– Лига украинских клубов интеллектуальных игр;
МП	– материализованное представление;
МТ	– модель требования;
ОА	– объект автоматизации;
ОКФ	– одноместный ковариантный функтор;
ООП	– объектно-ориентированное программирование;
ОРА	– описание рациональной архитектуры;
ОТ	– онтологическая точка;
ПО	– программное обеспечение;
ППТ	– паттерн проектирования требования;
ПрО	– предметная область;

СТ	– сформулированное требование;
СФТ	– сформулированное функциональное требование;
УАП	– участник автоматизируемого процесса;
УТ	– управление требованиями;
ФЗ	– функциональная задача;
ФМ	– функциональный модуль;
ФС	– функциональная структура;
ФТ	– функциональное требование;
AD	– architecture description;
BABOK	– Business Analysis Body of Knowledge;
CMMI DEV	– Capability Maturity Model Integration Development;
DIKW	– Data, Information, Knowledge and Wisdom;
GRASP	– General Responsibility Assignment Software Patterns;
MVC	– Model-View_Controller;
PMBOK	– Project Management Body of Knowledge;
REBOK	– Requirements Engineering Body of Knowledge;
SWEBOK	– Software Engineering Body of Knowledge.

ВВЕДЕНИЕ

Актуальность темы. В настоящее время результаты процессов формирования и анализа требований к информационным системам (ИС) и информационным технологиям (ИТ), предназначенным для автоматизации управления предприятиями и организациями, являются наиболее важным из факторов, определяющих успех или неудачу выполнения всех последующих работ по проектированию и разработке обеспечивающих частей, внедрению, эксплуатации и реинжинирингу данных ИС и ИТ. В основе этих процессов лежит формирование описания создаваемой ИС или ИТ как совокупности требований к системе с последующим преобразованием этих описаний вначале в описание архитектуры создаваемой ИС или ИТ, а впоследствии – в описания спецификаций на разработку элементов обеспечивающих комплексов создаваемой ИС или ИТ.

Современное представление требований к ИС и ИТ, а также методов формирования и анализа требований является результатом многочисленных научных работ таких ИТ-корпораций, как «Microsoft», «IBM», «Hewlett-Packard». Не менее актуальными в данной области являются работы отдельных зарубежных исследователей, среди которых можно выделить К.И. Виггера, А. Кобёрна, Д. Леффингуэлла и Д. Уидрига. В Украине проблемы, связанные с созданием ИС и ИТ различного назначения, являются областью исследований таких отечественных ученых, как А.А. Павлов, В.И. Гриценко, С.Ф. Теленик, О.Е. Федорович, В.В. Казимир, Н.В. Ткачук, И.П. Гамаюн, С.Г. Антощук, А.В. Горбенко, М.Ф. Бондаренко, В.А. Филатов.

Однако существующие в настоящее время модели требований к ИС или к ИТ, методы формирования и анализа этих требований обладают серьезными недостатками, приводящими к снижению эффективности и повышению затрат на выполнение процессов проектирования, разработки, внедрения, эксплуатации и реинжиниринга ИС или ИТ. Среди причин возникновения

этих недостатков следует особо выделить принятую в качестве основной точку зрения на требования к создаваемым ИС или ИТ как на уникальные артефакты, не связываемые друг с другом в единое представление создаваемой ИС или ИТ. Представление требований к ИС или ИТ как уникальных артефактов делает невозможным принятие решений о целесообразности использования опыта выполнения предыдущих ИТ-проектов в ходе инициации и планирования очередного ИТ-проекта создания ИС или ИТ. Кроме того, возникло противоречие между резко возрастающей сложностью архитектуры современных ИС и возможностью существующих инструментальных средств проектирования, которые учитывают широкий круг функциональных требований к ИС.

Эти и ряд других недостатков обусловили необходимость проведения научных исследований процессов формирования и обработки формальных описания требований к создаваемым ИС и проектирования архитектуры сложных ИС, что обеспечит возможность принятия решений о целесообразности ре-использования ранее реализованных требований для описания элементов создаваемой ИС и за счет этого сократить затраты на выполнение процессов микропроектирования ИС. Таким образом, создание методологических основ, моделей, методов и ИТ, которые позволяют сформировать рациональный вариант описания архитектуры создаваемой ИС с максимально возможным удовлетворением функциональных требований Поставщика и Потребителя и минимальными затратами на реализацию, является важной в концептуальном и прикладном аспектах научно-прикладной проблемой.

Связь работы с научными программами, планами, темами. Работа выполнялась автором на кафедре информационных управляющих систем Харьковского национального университета радиоэлектроники в рамках госбюджетных научно-исследовательских работ (НИР): «Исследование и разработка методологий, технологий проектирования информационных систем и их программно-аппаратных элементов» (№ ГР 0103U001559), «Разработка информационно-аналитической системы с распределенным искусственным интеллектом («Уни-

верситет»» (№ ГР 0101U001763), «Разработка типовой информационно-аналитической системы «Университет» (II очередь)» (№ ГР 0103U001560), «Методы и технологии создания информационно-образовательной среды с целью интеграции в общеевропейское пространство высшего образования» (№ ГР 0106U003152), «Методи, моделі та інформаційні технології розбудови соціально-економічної освітньо-наукової мережі з метою інтеграції у Європейський простір» (№ ГР 0109U002497), «Методи та моделі самоорганізації інфраструктури інтелектуального інформаційного середовища, що базується на використанні принципів хмарних обчислень» (№ ГР 0112U000207), «Теорія, методи і моделі управління життєвим циклом інтелектуальних інформаційних середовищ регіональних соціо-економічних об'єктів» (№ ГР 0115U002430), и таких хоздоговорных НИР: «Разработка информационной системы управления учебным процессом (I очередь)» (№ ГР 0106U008279); «Розробка специфікацій на інформаційну технологію інтеграції різнорідних сервісів до інформаційної системи управління підприємством» (№ ГР 0112U006065); «Розробка моделей, методів та технологій управління проектами впровадження інформаційних систем у страхових компаніях» (№ ГР 0115U001946).

Цель и задачи исследования. Целью диссертационной работы является решение актуальной научно-прикладной проблемы разработки методологических основ, моделей, методов и информационной технологии, позволяющих сформировать рациональный вариант описания архитектуры создаваемой информационной системы с максимально возможным удовлетворением функциональных требований Поставщика и Потребителя и минимальными затратами на реализацию.

Для достижения поставленной цели необходимо решить такие задачи:

- провести анализ существующих подходов к формированию требований и созданию описаний архитектуры сложных информационных систем управления организационно-техническими объектами и процессами;
- разработать методологические основы создания информационных систем, позволяющие установить четкую взаимосвязь между требованиями к системе,

описанием архитектуры системы и проектными решениями, реализующими требования в рамках предлагаемого описания архитектуры системы;

- разработать математические модели, описывающие множество требований к информационной системе на разных стадиях их формирования;

- разработать математическую модель паттерна проектирования требования к информационной системе;

- разработать математическую модель архитектурного фреймворка макропроектирования информационной системы и сформулированного функционального требования на уровне знаний как основного элемента архитектурного фреймворка;

- разработать математические модели, описывающие функциональные требования к информационной системе на уровнях информации и знаний;

- разработать методы формирования функциональных требований на уровнях информации и знаний;

- разработать метод синтеза вариантов описаний архитектуры информационной системы;

- разработать математическую модель синтеза описания рациональной архитектуры создаваемой информационной системы;

- разработать математическую модель унифицированного элемента информационной системы и метод формирования его описания, позволяющие оценить объем работ по созданию информационной системы;

- разработать методы анализа функциональных требований к информационной системе;

- разработать информационную технологию формирования описания архитектуры информационных систем, основанную на предлагаемых методологических основах, моделях и методах;

- разработать программный комплекс для реализации информационной технологии формирования описания архитектуры информационных систем в виде инструментального средства управления требованиями;

- провести апробацию результатов исследований в макропроектирова-

нии информационной системы управления организационно-техническим процессом.

Объект исследования – процесс макропроектирования информационных систем, к которому принадлежат процессы определения требований правообладателей, анализа требований, а также проектирования архитектуры системы.

Предмет исследования – методологические основы, модели, методы и информационная технология, обеспечивающие реализацию архитектурного фреймворка макропроектирования информационных систем.

Методы исследования. Диссертационное исследование базируется на системном анализе результатов исследований современных теоретических и прикладных разработок отечественных и зарубежных ученых в области моделей, методов и ИТ формирования и анализа требований к ИС. Для решения поставленных задач использованы методы системного анализа и общей теории систем – при модификации сервисного подхода к созданию ИС; математический аппарат теории категорий – при разработке моделей множества требований к ИС на разных стадиях их формирования, паттерна проектирования требования (ППТ), а также модели архитектурного фреймворка (АФ) макропроектирования ИС; математический аппарат теории множеств – при разработке моделей представлений требований к ИС на уровнях информации и знаний, модели сформулированного функционального требования (СФТ) к ИС на уровне знаний, методов формирования представлений функционального требования (ФТ) к ИС на уровнях информации и знаний, модели онтологических точек (ОТ) и метода формирования описаний ОТ; методы теории кластеризации – при разработке метода синтеза вариантов описаний архитектуры создаваемой ИС; модели и методы теории игр – при разработке модели синтеза описания рациональной архитектуры создаваемой ИС.

Научная новизна полученных результатов. Основной научный результат диссертации заключается в решении важно научно-прикладной проблемы создания методологических основ, моделей, методов и информационной технологии,

позволяющих сформировать рациональный вариант описания архитектуры создаваемой информационной системы с максимально возможным удовлетворением функциональных требований Поставщика и Потребителя и минимальными затратами на реализацию. .

В рамках выполненных исследований получены такие научные результаты:

1 Впервые разработаны:

1.1 Математическая модель паттерна проектирования требования к информационной системе, унифицирующая описание отдельного требования к информационной системе, что позволяет формализовать процесс реиспользования требований к информационной системе в ходе создания новых и реинжиниринга существующих систем.

1.2 Математические модели архитектурного фреймворка макропроектирования информационных систем и сформулированного функционального требования на уровне знаний как основного элемента данного фреймворка, основанные на формальном представлении фреймворка в виде множеств структурных и поведенческих паттернов проектирования требований, что позволяет установить общие особенности и ограничения и решить задачу синтеза описания рациональной архитектуры создаваемой информационной системы.

1.3 Модель онтологических точек, которая описывает создаваемую информационную систему как множество отдельных ветвей таксономий фреймов, и метод формирования описаний онтологических точек на основе данной модели, что позволяет автоматизировать процесс количественного оценивания объема работ по созданию информационной системы на основе разработанных моделей функциональных требований на уровне знаний.

1.4 Методы формирования представлений функциональных требований к информационной системе на уровнях информации и знаний, позволяющие формализовать процессы формирования описаний и анализа требований к информационной системе, что обеспечивает сокращение затрат времени на выполнение данных процессов и возможность автоматизированного проектирования архитектуры информационной системы.

2 Усовершенствованы:

2.1 Теоретико-множественные модели, описывающих синтаксис и семантику описаний требований к информационной системе на уровнях информации и знаний, которые, в отличие от существующих, основаны на модели паттерна проектирования требования к информационной системе, что позволяет унифицировать работы по формированию, хранению и обработке представлений требований к информационной системе на этих уровнях вне зависимости от конкретных проектов создания информационных систем, в которых требования были сформулированы.

2.2 Теоретико-игровая модель синтеза описания рациональной архитектуры создаваемой информационной системы, которая, в отличие от существующих, описывает процесс синтеза описания рациональной архитектуры как выбор варианта описания, в наибольшей степени соответствующего представлениям создаваемой системы Поставщика и Потребителя ИТ-услуг, что позволяет автоматизировать процесс проектирования архитектуры системы путем поиска точек равновесия по Нэшу в чистых и смешанных стратегиях.

2.3 Методы анализа сформулированных функциональных требований к создаваемой информационной системе путем выявления противоречивых и нелогичных требований в ходе решения задачи синтеза описания рациональной архитектуры информационной системы, что позволяет унифицировать и автоматизировать выполнение работ по анализу функциональных требований в рамках макропроектирования.

3 Получили дальнейшее развитие:

3.1 Сервисный подход к созданию информационных систем, который, в отличие от существующих, рассматривает информационную систему как совокупность ИТ-услуг и реализующих эти услуги ИТ-сервисов, формируемых на основе имеющихся проектных решений, удовлетворяющих множеству требований, выдвинутых к системе, что позволяет сократить затраты на создание информационной системы за счет ре-использования системных компонентов.

3.2 Математические модели, описывающие множество требований к ин-

формационной системе на разных стадиях их формирования, которые, в отличие от существующих, описывают связи между представлениями требований к информационной системе в ходе их формирования, что позволяет формально описать множество возможных архитектурных фреймворков создания информационных систем с помощью сервисного подхода.

3.3 Метод синтеза вариантов описаний архитектуры создаваемой информационной системы, который, в отличие от существующего, выделяет для дальнейшего исследования варианты описаний, незначительно хуже оптимального, что позволит в максимально возможной степени выполнить требования Поставщика и Потребителя IT-услуг в процессе формирования компромиссного описания рациональной архитектуры создаваемой информационной системы за счет увеличения количества исходных вариантов описаний архитектуры.

Практическое значение полученных результатов. Разработанные в диссертационной работе подход, модели и методы были использованы для создания IT формирования и анализа требований к ИС управления предприятиями и организациями. В работе предложена основная концепция, определены основные IT-услуги, разработаны решения по информационному (ИО) и программному (ПО) обеспечением данной IT. Разработанные методы, модели и IT прошли апробацию во время выполнения работ по разработке функционального модуля безопасности труда комплексной системы управления ПАО «Харьковгоргаз».

Полученные теоретические и практические результаты внедрены на предприятиях: ПАО «Харьковгоргаз» (акт внедрения от 16.06.2015 г.); ИТ СКБ «ПОЛИСВИТ» ГНПП «Объединение Коммунар» (акт внедрения от 27.10.2015 г.); ООО «ПрофITсофт» (акт внедрения от 15.12.2015 г.); ООО ТД «Технодар» (акт внедрения от 24.01.2016 г.). Акты о внедрении приведены в Приложении А.

Личный вклад соискателя. Все основные результаты получены автором лично и опубликованы в работах [118, 119, 128, 129, 134, 135, 140, 145-148, 157-161, 165-170, 180-185, 188-192, 197-199, 201-207, 209-215, 219, 222]. В работах,

опубликованных в соавторстве, соискателю принадлежат: [118] – описания основных ролей и формулировки глобальных целей деятельности Поставщика и Потребителя IT-услуг; [134] – основные положения сервисного подхода, концепция представления требований к ИС, теоретико-множественные модели описывающие множество требований к ИС на разных стадиях их формирования, математическая модель АФ макропроектирования ИС, математическая модель ППТ к ИС, теоретико-множественные модели, описывающие синтаксис и семантику описаний требований к ИС на уровнях данных, информации и знаний, обобщенная теоретико-игровая модель синтеза описания рациональной архитектуры создаваемой ИС, определение термина «онтологическая точка»; [135] – принципы формального описания представлений требований к ИС на уровне информации; [146] – модель одноместного ковариантного функтора; [147] – условия реализации модели одноместного ковариантного функтора в рамках CASE-системы разработки ИС; [157] – определение ППТ к ИС; [158] – определения и классификация ППТ к ИС; [160, 161] – математическая модель АФ макропроектирования ИС; [166] – математическая модель атрибута структурных и поведенческих ППТ к ИС; [167] – представление понятия «Требование к элементу ИС» в виде диаграммы классов; [168] – представление функций ИС как различных вариантов типового представления процесса обработки информации; [169] – модель представления сформулированных требований (СТ) к ИС на уровне информации, набор аксиоматических утверждений и следствий из них, принципы формального описания представлений требований к ИС на уровне информации, модели структурных и поведенческих ППТ к ИС на уровне информации; [170] – обобщенная модель представления требований к ИС на уровне знаний; [183] – метод формирования представления требований к ИС на уровне информации; [184] – теоретико-множественная модель СФТ к ИС на уровне знаний, метод синтеза вариантов описаний архитектуры создаваемой ИС, теоретико-игровая модель синтеза описания рациональной архитектуры создаваемой ИС, модель ОТ и метод формирования описаний ОТ; [188] – формальные описания законов композиции функциональной структуры ИС, определяющие возможные значения коэффици-

ента отталкивания; [189] – основные характеристики теоретико-игровой модели синтеза описания рациональной архитектуры создаваемой ИС; [190] – теоретико-игровая модель синтеза описания рациональной архитектуры ИС, функции и матрицы выигрышей Поставщика и Потребителя ИТ-услуг; [191] – теоретико-игровая модель принятия решения по выбору варианта конфигурации функциональной структуры ИС, чистые и смешанные стратегии участников игры; [192] – вариант теоретико-игровой модели синтеза описания рациональной архитектуры ИС; [198] – определение понятия «онтологическая точка», правила уточнения количественных оценок экранов, отчетов и модулей функции ИС; [198, 201] – правила модификации метода объектных точек для обработки представлений элементов создаваемой ИС в виде ОТ; [202] – IDEF3-диаграмма, описывающая работы по формированию и анализу функциональных требований к создаваемой ИС в соответствии с положениями сервисного подхода, и показатель, количественно характеризующий степень нелогичности ФТ к ИС; [203] – концепция представления требования к ИС и схема взаимодействия основных элементов ИТ формирования и анализа требований к ИС; [204] – основные положения создания ИТ формирования и анализа требований к ИС, содержание ее основных этапов; [205] – описания основных ИТ-услуг ИТ формирования и анализа требований к ИС; [206] – основные характеристики ИТ формирования и анализа требований к ИС как инновационного ИТ-проекта; [207] – описание концепции представления требований к ИС и подход к реализации ППТ к ИС в ИО ИТ формирования и анализа требований к ИС; [210] – схема базы данных (БД), используемой для хранения описаний ФТ и решений, реализующих эти требования; [211] – перечень стадий жизненного цикла (ЖЦ) ФТ к ИС; [212, 213] – концепция модификации шаблона проектирования ПО «Model – View – Controller» (MVC); [214] – метод формирования онтологических описаний предметной области (ПрО) на основе ФТ к ИС, выдвинутых по результатам обследования автоматизируемого объекта или процесса; [215] – последовательность работ ИТ формирования и анализа требований к ИС; [219] – описание концепции представления требований к ИС, описания ФТ к информационно-аналитической системе (ИАС) управ-

ления безопасностью труда, описания процессов и сущностей данных, формально описывающих выделенные ФТ; [222] – принципы формирования описания архитектуры ИАС управления безопасностью труда.

Апробация результатов диссертации. Основные теоретические и практические результаты диссертационной работы докладывались и получили одобрение на таких конференциях: Международной конференции «Теория и техника передачи, приема и обработки информации» (Харьков, 2003); IV Miedzynarodowej naukowii-praktycznej konferencji «Nowoczesnych naukowych osiagniec – 2008» (Przemysl, 2008); VI Miedzynarodowej naukowii-praktycznej konferencji «Dynamika naukowych badac – 2010» (Przemysl, 2010); VI Международной научно-практической конференции «Наука и социальные проблемы общества: информатизация и информационные технологии» (Харьков, 2011); IX mezinarodni vedecko – prakticka konference «Aplikovane vedecke novinky - 2013» (Praha, 2013); Науково-практичній конференції «Застосування інформаційних технологій у підготовці та діяльності сил охорони правопорядку» (Харків, 2013); IV Міжнародній науково-практичній конференції «Системний аналіз. Інформатика. Управління (САІУ-2013)» (Запорожжє, 2013); Научна конференция с международно участие «Хранителна наука, техника и технологии 2013» (Пловдив, 2013); XII конференции по физике высоких энергий, ядерной физике и ускорителям (Харьков, 2014); First International forum «IT-Trends: big data, artificial intelligence, social media» (Kremenchuk, 2014;); XI mazarodni vedecko – prakticka konference «Predni vedecke novinky – 2015» (Praha, 2015); IV Міжнародній науково-практичній конференції «Інформаційні управляючі системи та технології (ІУСТ-Одеса-2015)» (Одеса, 2015); Международной научно-технической конференции «Информационные системы и технологии» (Харьков, 2012, 2013, 2014, 2015).

Инновационный IT-проект, в основу которого положены основные теоретические и практические результаты диссертационной работы, был представлен на Конкурсе инновационных проектов в сфере информационно-коммуникационных технологий «IT-Kharkiv» (Харьков, 2013) и занял третье

место в номинации «Инструментальные средства разработки программного обеспечения и систем».

Публикации. По результатам диссертационных исследований опубликовано 50 научных работ, среди которых 1 монография, 10 статей в журналах и 15 статей в сборниках научных работ, включенных в перечень специализированных изданий Министерства образования Украины (из них 9 статей без соавторов, 10 статей опубликованы в изданиях, входящих в наукометрические базы, в частности 2 статьи без соавторов в изданиях, входящих в наукометрическую базу Scopus), 24 публикации в сборниках материалов научных конференций и форумов.

Структура и объем диссертации. Диссертация состоит из введения, шести разделов, выводов, перечня использованных источников из 222 наименований (на 26 стр.), семи приложений (на 71 стр.), содержит 48 иллюстраций (на 31 стр.), 22 таблицы (на 16 стр.). Общий объем работы – 429 страниц.

РАЗДЕЛ 1

АНАЛИЗ ПРОБЛЕМ ФОРМИРОВАНИЯ ТРЕБОВАНИЙ И СОЗДАНИЯ ОПИСАНИЙ АРХИТЕКТУРЫ СЛОЖНЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

1.1 Анализ современного представления информационной системы

В настоящее время термин «информационная система» является в значительной степени многозначным. Вместо единой точки зрения на семантику данного термина существует множество различных его определений, отличия которых обусловлены целями формулирования этих определений. Поэтому возникает необходимость анализа особенностей современного представления данного термина, результаты которого позволят выявить возможные направления развития теории и практики создания и эксплуатации ИС различного назначения.

В настоящее время основой для проведения подобного анализа следует считать сложившийся в мире к 2000-м гг. подход, обобщающий множество разнообразных концепций, теоретических разработок, прикладных работ и успешных практик выполнения IT-проектов в рамках системы стандартов. В настоящее время в Украине сложились три группы стандартов в сфере деятельности по созданию и эксплуатации ИС и ИТ.

Группа 1. Группа украинских стандартов, являющихся переводами международных стандартов ISO, ISO/IEC и им подобных.

Группа 2. Группа советских стандартов группы 34 «Информационные технологии», содержащих описания основ методологии и практические рекомендации по выполнению отдельных видов работ по разработке ИС и ИТ различного назначения.

Группа 3. Группа ДСТУ, фактически дублирующих содержание стандартов группы 2.

Каждая группа стандартов трактует термин «ИС» по-своему. Так, в соответствии с положениями ГОСТ 34.003-90 «Автоматизированные системы.

Термины и определения», термин «Автоматизированная система» (АС) определяется как система, состоящая из персонала и комплекса средств автоматизации (КСА) его деятельности, реализующая ИТ выполнения установленных функций. Под термином «ИТ» в этом стандарте понимаются приемы, способы и методы применения средств вычислительной техники при выполнении функций сбора, хранения, обработки, передачи и использования данных. Термином «функция АС» обозначается совокупность действий АС, направленная на достижение определенной цели [1].

Согласно действующему в Украине стандарту ДСТУ 2226-93 «Автоматизовані системи. Терміни і визначення», относящемуся ко второй группе стандартов, под термином «АС» понимается организационно-техническая система, состоящая из средств автоматизации определенного вида или нескольких видов деятельности людей и персонала, осуществляющего эту деятельность [2]. В то же время в стандарте ДСТУ 2941-93 «Розроблення систем. Терміни та визначення» под термином «АС» понимается система, реализующая ИТ выполнения установленных функций при помощи персонала и КСА [3], что в целом повторяет определения стандарта ГОСТ 34.003-90.

Вместе с тем, за время, прошедшее с начала 1990-х гг., в мире сложился целый ряд определений термина «ИС» как самостоятельно существующего термина. Отличительной чертой этих и аналогичных им определений данного термина является признание организации процессов хранения и обработки хранимой информации наиболее важной характеристикой, определяющей создание эффективных и качественных ИС [4-6]. Данный подход присущ не только научным исследованиям, но и прикладным разработкам в области методологий и методов создания ИС. Так, например, методология структурного анализа и проектирования ИС (Structured System Analysis and Design Method, SSADM), принятая Великобританией в качестве государственного стандарта, представляет завершение разработки ИС как создание и, при необходимости, оптимизацию ее физических моделей БД. Проблемы разработки ПО и технического обеспечения, реализующих ИТ сбора, обработки и отображения дан-

ных, выходят за пределы данной методологии [7].

Ситуация изменилась после того, как в середине 2000-х гг. в Украине был принят целый ряд стандартов ISO, относящихся к третьей группе, которые определяют базовые принципы, процессы, методологии и модели, используемые на различных стадиях жизненного цикла ИС. В этих стандартах основным термином является термин «система», под которым понимается комбинация взаимодействующих элементов, организованных для достижения одной или нескольких поставленных целей [8]. При этом отмечается, что на практике интерпретация данного термина зачастую уточняется с помощью ассоциативного существительного или может заменяться контекстным синонимом. Такое уточнение позволяет при описании конкретных видов систем и работ, выполняемых в рамках ЖЦ систем, использовать более специализированные термины. К таким терминам следует, в частности, отнести термин «программное средство», под которым понимается единая часть общей системы, выполняющая в данной системе определенные функции, что осуществляется посредством выделения из требований к системе требований к программным средствам, проектирования программных средств, их производства и объединения в систему [9].

Анализ изменений терминологической базы показывает, что в целом принятие Украиной стандартов третьей группы не противоречит ранее сложившейся терминологической базе стандартов ГОСТ и ДСТУ в сфере ИС и ИТ. Взаимосвязь основных концептов совокупности стандартов Украины в данной сфере показана на рис. 1.1.

Результаты анализа позволяют утверждать, что основным термином большинства украинских стандартов по-прежнему является термин «АС». Учитывая, что в любую АС входят ИТ, реализация которых позволяет достичь поставленные перед системой цели, любую АС можно определять как ИС. Однако данное утверждение условно и не находит подтверждения в конкретных стандартах ГОСТ и ДСТУ. В то же время наблюдаются существенные изменения семантики отдельных терминов, что приводит к необходимости изменений в методологиях, моделях и методах создания ИС. В качестве

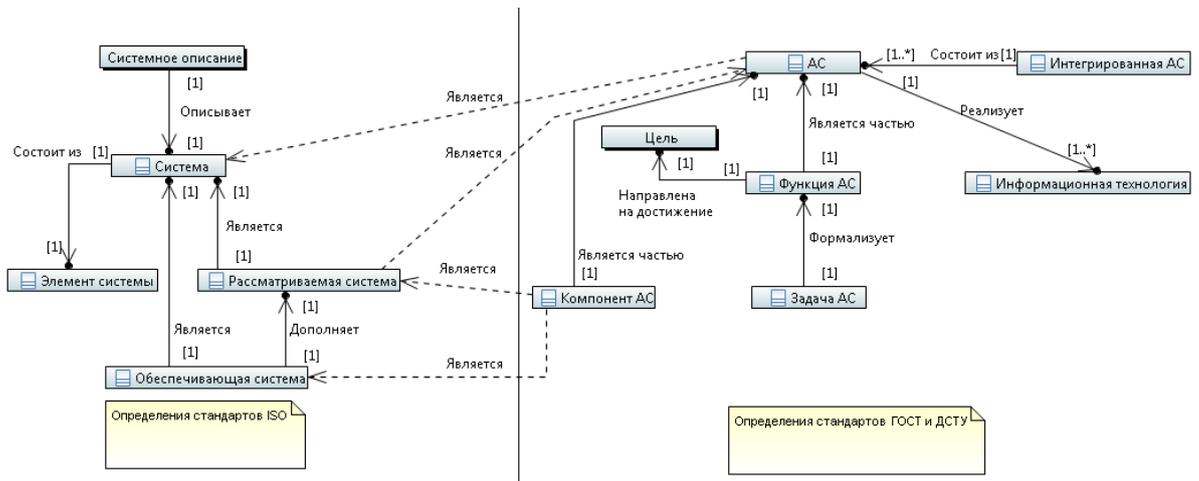


Рисунок 1.1 – Схема взаимосвязи основных концептов совокупности стандартов Украины в области информационных систем и технологий

основных примеров этих изменений необходимо отметить:

а) выделение в качестве основного термина «система», что позволяет сосредоточить основное внимание на тех аспектах существования, разработки, эксплуатации и модернизации систем, которые наиболее подвержены изменениям;

б) отказ от специального деления системы на функции и компоненты, обеспечивающие реализацию этих функций, и рассмотрение создаваемой системы как совокупности систем различной природы, обеспечивающих деятельность друг друга в рамках создаваемой системы;

в) отказ от использования понятия «цель» для характеристики системы и ее функций и представление целей системы как эффективного и качественного выполнения требований, выдвинутых к создаваемой системе;

г) отказ от особой роли человека как компонента системы и вынесение всех заинтересованных лиц за пределы создаваемой, внедряемой или эксплуатируемой системы;

д) отказ от представления видов обеспечений как компонентов системы и представление их как систем, которые в различных ситуациях могут быть представлены либо понятием «рассматриваемая система», либо понятием «обеспечивающая система»;

е) появление понятия «системное описание», которое определяет множество способов восприятия системы заинтересованными лицами на различных фазах ее ЖЦ.

Помимо прочего, особое внимание следует обратить на выделение термина «ИС» в качестве самостоятельного, что указывает на переход от классификации систем по признаку участия в их деятельности отдельной группы системных элементов (персонала) к классификации систем по особенностям природы интеграции всех групп элементов в систему. Однако действующие стандарты по-прежнему не уделяют проблеме системной интеграции должного внимания.

Как показано выше, системное описание ИС всеми заинтересованными лицами во многом определяется той фазой ЖЦ, на которой система находится в каждый конкретный момент времени. Поэтому необходимо проанализировать основные особенности системных описаний ИС на ранних стадиях ее ЖЦ.

Современное представление ЖЦ ИС определяется стандартом ISO/IEC 15288 «System Engineering – System life cycle processes» на основе возможного использования абстрактной функциональной модели, представляющей концептуализацию потребности в системе, ее реализации, применения, развития и ликвидации [8]. При построении такой модели предполагается, что система развивается на протяжении ЖЦ в результате действий, осуществляемых и управляемых людьми, работающими в организациях и использующими определенные процессы в своей деятельности. Детали модели ЖЦ выражаются в терминах этих процессов, названных процессами ЖЦ, при помощи которых может быть смоделирован ЖЦ системы, их результатов, взаимосвязи и возникновения.

На ранних стадиях ЖЦ системы выполняются процессы определения требований правообладателей, анализа требований и проектирования архитектуры системы [8]. Взаимодействие данных процессов в ходе макропроектирования ИС показано в виде IDEF3-модели на рис. 1.2.

В результате анализа процессов ЖЦ ИС, показанных на рис. 1.2, можно сделать вывод о том, что первым системным описанием, которое должно формировать представление о создаваемой ИС в целом, является системное описа-



Рисунок 1.2 – IDEF3-модель взаимосвязи процессов жизненного цикла информационной системы, выполняемых в ходе ее макропроектирования

ние архитектуры создаваемой системы. В [8] рекомендуется представлять данное описание в виде итогового проекта архитектуры системы, формируемого на основе множества приемлемых архитектур. В этом описании каждой функции системы ставится в соответствие множество элементов архитектуры системы, сопровождаемое множеством соответствующих производных требований.

Описание архитектуры создаваемой системы формируется в ходе выполнения процессов ЖЦ ИС на основе описаний множества отдельных требований правообладателей и системных требований. Данное описание связывает описания отдельных требований к системе с решениями по отдельным элементам системы и решениями по обеспечивающим системам [8].

Множество требований правообладателей является хронологически первым системным описанием создаваемой ИС и представляет систему не как единый целостный продукт, а как набор не связанных друг с другом описаний отдельных элементов системы. Данное множество включает в себя различные требования, на основе которых формируется множество сценариев, описывающих отдельную функциональную возможность создаваемой систе-

мы. Требования правообладателей и сценарии, идентифицирующие функциональные возможности системы, выражаются в терминах моделей (текстовых или формальных), ориентированных на цели и поведение системы и описывающих систему в контексте среды и условий функционирования.

На основе множества требований правообладателей и сценариев, идентифицирующих функциональные возможности системы, в ходе выполнения процесса анализа требований формируется системное описание, представляющее собой множество системных требований. Среди этих требований можно выделить [8]:

- а) ФТ;
- б) эксплуатационные требования;
- в) требования, связанные с такими свойствами системы, как здоровье, безопасность, защищенность, безотказность, готовность;
- г) требования, связанные со свойствами обеспечивающих систем;
- д) технические показатели и показатели качества при использовании.

Кроме того, множество системных требований дополняется множеством ограничений, среди которых особо следует выделить множество функциональных границ системы в терминах ее поведения и свойств, которые должны быть обеспечены. Рекомендации по способу выражения системных требований и ограничений в [8] не приводятся.

Анализ принятых в современных стандартах системных описаний создаваемой ИС на ранних стадиях ее ЖЦ позволяет сделать следующие выводы:

- а) рекомендуемые [8] системные описания, формируемые на ранних стадиях ЖЦ ИС, ориентированы на представление создаваемой системы как множества отдельных требований, функций и архитектурных элементов, а не единого целостного проекта;
- б) в ходе выполнения процессов формирования требований правообладателей, анализа требований и проектирования архитектуры задачи системной интеграции не решаются и не ставятся;
- в) использование при формировании системных описаний опыта, накоп-

ленного в ходе выполнения предыдущих проектов создания аналогичных систем, не рассматривается;

г) решение о целесообразности начала IT-проекта создания ИС должно приниматься на основе множества требований правообладателей и не базируется ни на опыте создания ИС аналогичного назначения, ни на представлении ИС как единого целостного продукта, что серьезно искажает оценки объема работ по созданию ИС.

Данные выводы позволяют утверждать, что в ходе создания ИС существуют серьезные проблемы на ранних стадиях ее ЖЦ, значительно повышающие затраты на создание ИС и увеличивающие количество инцидентов, которые могут возникнуть в ходе разработки обеспечивающей части данной ИС. В то же время практика выполнения проектов (в том числе – IT-проектов) показывает [10], что изменение ИС под влиянием заинтересованных сторон наиболее эффективно и требует наименьших затрат преимущественно на ранних стадиях ее ЖЦ. Поэтому решение указанных проблем является одним из наиболее перспективных направлений научно-прикладных исследований в IT-сфере.

1.2 Анализ современных представлений требований к информационной системе

1.2.1 Анализ определений понятия «требование к системе»

Результаты анализа современного представления ИС (см. подразд. 1.1) показывают, что требования к ИС являются основной исходной информацией для проектирования архитектуры ИС, на основе которой выполняются работы по разработке обеспечивающих систем. Поэтому необходимо проанализировать современные представления требований к ИС с целью выделения их недостатков, приводящих к увеличению затрат на создание или модернизацию ИС в рамках ее ЖЦ.

В настоящее время базовым определением понятия «требование» в IT-сфере считается определение, данное в стандарте IEEE 610.12-1990. Согласно

этому определению под требованием понимается [11]:

а) условие или возможность, необходимые пользователю для решения проблемы или достижения цели;

б) условие или возможность, которыми должна обладать система или компонент системы, соответствующие договору, стандарту, спецификации или другому официальному документу;

в) документированное представление условия или возможности подобно описанному в первых двух определениях.

Большинство IT-специалистов считает, что определения а) и б) являются вполне приемлемыми общими определениями понятия «требование» [12-14]. Вместе с тем, в случае необходимости IT-специалистами предлагаются собственные определения понятия «требование», например:

а) требование – это нечто такое, что приводит к выбору дизайна [15];

б) требования – это спецификация того, что должно быть реализовано.

В них описано поведение системы, свойства системы или ее атрибуты. Они могут быть ограничены процессом разработки системы [16].

В методологиях разработки ИС определению понятия «требование к ИС» не уделяется должного внимания. Так, в методологии разработки АС, изложенной в ГОСТах группы 34, термин «требование» до сих пор не имеет специального определения, хотя и применяется для описания структур и содержания проектных документов трех первых стадий создания АС [17, 18]. В методологии создания ИС SSADM требование к ИС определяется как ответ на следующие вопросы [19, 20]:

- «Что требуется от системы?»;

- «Зачем это нужно?»;

- «Кому это необходимо?»;

- «Насколько это важно?»;

- «Чем можно измерить степень соблюдения требования?».

Рассматривая проблему определения понятия «требование» в современной IT-индустрии (создание и эксплуатация ИС, IT программных продуктов раз-

личного назначения), исследователи сходятся во мнении, что данная проблема является одной из наиболее серьезных и не имеет общего решения [21, 22].

1.2.2 Анализ существующих классификаций требований к информационной системе

Проблема определения понятий «требование» и «требование к ИС» заключается, прежде всего, в том, что эти понятия объединяют в себе целый ряд элементарных понятий, обозначающих различные группы требований, объединенные по какому-либо общему признаку. Поэтому необходимо проанализировать существующие способы классификации требований.

В ходе анализа рассмотрены классификации требований к ИС, содержащиеся в следующих методологиях и работах исследователей:

- а) стандарты группы 34 «Информационные технологии» как методология создания АС различного назначения [17, 18];
- б) методология SSADM [19, 20];
- в) исследования, выполненные Д. Леффингуэллом и Д. Уидригом [13], а также К.И. Виггерсом [21];
- г) методология ИТ-компании Telelogic, разрабатывающей пакеты управления требованиями Telelogic DOORS [23];
- д) модель классификации требований FURPS и ее современный вариант FURPS+ (компания «Hewlett-Packard») [24, 25];
- е) свод знаний по анализу бизнес-процессов (БП) современных предприятий (Business Analysis Body of Knowledge, BABOK) [26], в том числе – с целью определения целесообразности автоматизации данных БП.

Данные варианты классификаций рассмотрены в разд. Б.1 Приложения Б и описывают все основные точки зрения на требования к ИС как к техническому ИТ-продукту, необходимому для достижения экономических или тех-

нико-экономических целей.

Соответствие рассмотренных вариантов классификации требований техническим процессам ЖЦ системы показано в табл. Б.1 Приложения Б.

По результатам анализа можно сделать следующие выводы:

- а) единой системы классификации требований к ИС не существует;
- б) в настоящее время большинство вариантов классификации требований рассматривают требования к системе как промежуточное звено между бизнес-требованиями и требованиями к ПО;
- в) нефункциональные требования рассматриваются как способ уточнения других требований в ходе их анализа;
- г) в процессе проектирования архитектуры системы внимание сосредотачивается, главным образом, на проектировании архитектуры ПО.

1.2.3 Анализ методов сбора требований к информационной системе

Как показано выше, в настоящее время не существует общепризнанных определения понятия «требование к системе» и системы классификации требований. Поэтому способы формального описания требований к ИС во многом определяются теми методами сбора требований, которые применяются в ходе выполнения процесса определения требований правообладателей.

Сравнительное описание методов сбора требований к системе проведено в [27] на основе следующих современных стандартов, методологий и рекомендаций конкретных специалистов:

- BABOK [26];
- свод знаний по управлению проектами (Project Management Body of Knowledge, PMBOK) [10, 28];
- свод знаний по инженерии ПО (Software Engineering Body of Knowledge, SWEBOK) [29];

- свод знаний по инженерии требований (Requirements Engineering Body of Knowledge, REBOK) [30];
- стандарт ISO/IEC 29148 «Системная и программная инженерия. Процессы жизненного цикла. Разработка требований» [31];
- комплекс моделей Capability Maturity Model Integration Development (CMMI DEV) v1.3 [32];
- методология IBM OpenUP и модель FURPS+ [24, 25];
- библиотека показателей ITIL v2 [33];
- методология компании Telelogic, изложенная Э. Халл, К. Джексоном и Дж. Диком [23];
- методы сбора требований, изложенные в работах А. Кобёрна [34], К.И. Виггерса [21], Л.А. Мацяшека [35].

Анализ методов, описанных данными источниками, позволяет утверждать, что основная проблема заключается в отсутствии общей точки зрения на целесообразность применения конкретных методов для выявления требований в конкретных IT-проектах создания ИС. Вместе с тем, с точки зрения разных стандартов, методологий и отдельных специалистов, некоторые методы сбора требований к системе более предпочтительны. Результаты анализа частоты рекомендаций отдельных методов сбора требований к ИС приведены в табл. Б.2 Приложения Б.

По результатам анализа можно сделать следующие выводы:

- а) наиболее распространенными с точки зрения применения следует считать методы индивидуального сбора требований (интервьюирование, наблюдение, прототипирование);
- б) методы группового сбора требований и методы выявления требований путем анализа примерно равнозначны по рекомендациям к применению;
- в) методы выявления требований путем анализа направлены на выявление требований через анализ информации, уже аккумулирующей в себе коллективный опыт, в то время, как методы группового сбора требований позволяют сформировать требования путем выявления такого коллективного опыта в ходе диалога;
- г) применение других методов сбора требований в сильной степени за-

висит от особенностей выполняемого IT-проекта.

В целом же можно утверждать, что описания требований, формируемые с применением рассмотренных методов, обладают следующими особенностями:

- а) отсутствие формальной основы построения подобных описаний;
- б) отражение только отдельных элементов создаваемой ИС без возможности сформировать описание создаваемой ИС в целом;
- в) невозможность ре-использования требований, выдвинутых и реализованных в предыдущих IT-проектах, в проекте создания новой ИС.

1.2.4 Анализ моделей требований к информационной системе

Как показано выше, применение подавляющего большинства методов сбора требований не позволяет получить формального описания требования к ИС. Поэтому в настоящее время можно выделить следующие способы описаний требований:

- а) в виде текстов на естественном языке или структурированных документов;
- б) в виде структурных или объектно-ориентированных визуальных моделей (ВМ);
- в) в виде текстов на формальном языке (в том числе в виде одноразовых или многократных программных прототипов).

Как правило, в ходе создания ИС различного назначения обычно используются описания требований, сделанные, как минимум, двумя из рассмотренных способов. Это обусловлено сложившимся в 1990-х гг. убеждением, согласно которому никакое описание требований в одном виде не дает их полной картины [36].

Такой подход к представлению требований в ходе создания ИС позволил ввести термин «модель требований» (МТ), обозначающий комплекс описаний требований теми способами, которые позволяют на основе данных описаний выполнить необходимые работы в рамках ЖЦ ИС с применением конкретных методологий и ИТ [37]. МТ необходимо пересматривать в начале

каждой итерации процессов ЖЦ системы и не обязательно оформлять в подробностях до создания кода.

Примером классической МТ к ИС является МТ, разработанная в рамках методологии SSADM. В основу данной МТ положены описания требований по таким направлениям [19, 38]:

- а) функции ИС (точка зрения пользователя на функционирование системы);
- б) события (реальные события бизнеса или события, генерируемые самой ИС);
- в) данные, характеризующие информационную среду, окружающую ИС.

Основным описанием требования к ИС в рамках данной МТ следует считать документ «Каталог требований». Этот документ может уточняться в ходе формирования и анализа требований к ИС. Форма каталога требований построена так, что невозможность заполнить какой-либо раздел (т.е. ответить на какой-либо из перечисленных вопросов) означает потребность в дополнительной информации. Для обеспечения измеримости в состав требования к качеству выполнения функции следует включать желаемое (целевое) значение показателя качества и диапазон допустимых отклонений - это позволяет уменьшить неопределенность и проверить, насколько система удовлетворяет предъявленным требованиям, при ее отладке и приемо-сдаточных испытаниях. По мере развития проекта в каталог требований следует добавлять ссылки на вновь разработанные документы, в которых реализованы требования. Это позволяет проследить за их соблюдением, не дожидаясь испытаний законченной системы [19, 20, 39].

Таким образом, МТ к ИС методологии SSADM может быть охарактеризована как набор описаний, рассмотренный в табл. Б.3 Приложения Б.

Несмотря на то, что с 2000 г. методология SSADM является составной частью методологии «Business System Development», основная МТ к системе осталась неизменной. Лишь в последнее время исследователями формулируются задачи модификации и расширения данной МТ в связи с переходом от разработки ИС как самостоятельного продукта к разработке ИС как совокупности сервисов, обслуживающих конкретные БП объекта автоматизации (ОА) [40].

В настоящее время наиболее распространена МТ, примером которой является МТ компании «Microsoft». Она сводится к набору описаний требований в виде ВМ и сопровождающих эти модели текстовых документов. Этот набор описаний рассмотрен в табл. Б.4 Приложения Б [37].

С точки зрения компании «Microsoft», неважно, с какого именно представления начнется формирование представлений требований к системе. В то же время, при создании указанных в табл. Б.2 представлений рекомендуется перемещаться от одного представления к другому.

В качестве возможной замены моделей, описывающих отдельные аспекты требований к системе, компания «Microsoft» предлагает рассматривать прототипы отдельных функций системы, разрабатываемые в упрощенной форме на ранних стадиях ИТ-проекта [37].

Результаты сравнительного анализа данных МТ к ИС как наиболее типичных представителей подобных моделей приведены в табл. Б.5 Приложения Б. На основе этих результатов были сделаны следующие выводы:

а) наиболее распространенные МТ к ИС носят описательный характер и не являются формальными конструкциями, позволяющими сформулировать ответы на вопросы относительно создаваемой ИС с достаточной степенью точности;

б) в современных МТ, примером которых является МТ компании «Microsoft», основной упор сделан не на описание данных, а на описание процессов системы и взаимодействия этих процессов с пользователями ИС;

в) в отличие от МТ методологии SSADM, в МТ компании «Microsoft» появляются элементы, описывающие БП ОА;

г) в МТ компании «Microsoft» практически не уделяется внимания описанию взаимосвязей пользователей ИС и требований, которые они выдвигают к системе;

д) основными способами описания требований в современных МТ к ИС становятся ВМ (в том числе – объектно-ориентированные), дополняемые текстовыми документами или программными прототипами.

1.2.5 Анализ подходов к управлению требованиями к информационной системе

Рассмотрим современные представления об управлении требованиями (УТ) в ходе разработки систем.

В стандартах группы 34 проблема УТ не рассматривается, а рекомендации по организации и документированию работ по УТ к ИС отсутствуют [1, 17, 18].

Д. Леффингуэлл и Д. Уидриг рассматривают УТ как систематический подход к выявлению, организации и документированию требований к системе, а также процесс, в ходе которого вырабатывается и обеспечивается соглашение между заказчиком и выполняющей проект группой по поводу меняющихся требований к системе [13]. Подобную точку зрения разделяет и К.И. Виггерс, определяя понятие «управление требованиями» как выработку и поддержание взаимного согласия с заказчиками по поводу требований к разрабатываемому ПО [41], в ходе которых должны выполняться такие действия [21]:

- определение основной версии требований (моментальный срез требований для конкретной версии продукта);
- просмотр предлагаемых изменений требований и оценка вероятности воздействия каждого изменения до его принятия;
- включение одобренных изменений требований в проект установленным способом;
- согласование плана проекта с требованиями;
- обсуждение новых обязательств, основанных на оцененном влиянии изменения требований;
- отслеживание отдельных требований с момента их формулирования до разработки их дизайна, исходного кода и вариантов тестирования;
- отслеживание на протяжении всего проекта статуса требования и действий по изменению этого статуса.

Для выполнения последнего действия из этого списка в [21] предлагается использовать шаблон определения статуса требования (см. табл. Б.6 Приложения Б).

Согласно методологии Rational Unified Processes (RUP), УТ – это систематический подход к выявлению, организации и документированию требований к системе, а также установление и поддержание соглашения между клиентом и группой разработки по поводу изменений требований к системе. Данное соглашение, как и тексты исходных требований, подлежит документальному оформлению. Документирование всего ЖЦ требования является вариантом отслеживания требования.

Процесс УТ, согласно RUP, разделяется на следующие работы [42]:

- формирование плана УТ;
- сбор требований;
- разработка документа Концепции (Vision);
- создание сценариев использования (Use Cases);
- дополнительная спецификация;
- создание тестовых сценариев (Test Cases) из Use Cases;
- создание Test Cases из дополнительной спецификации;
- проектирование системы.

Способ представления отношений между требованиями различного уровня в системе называется «трассировка требований». Она помогает определить источник любого требования. В процессе трассировки выполняются следующие действия [42]:

- подтверждение того, что реализация удовлетворяет всем требованиям;
- подтверждение того, что приложение делает только то, что было заказано;
- анализ воздействия;
- помощь в управлении изменениями требований и продукта.

Анализ приведенных выше подходов к организации УТ позволяет выделить основной принцип УТ – постепенное преобразование множества начальных значений атрибутов, описывающих требование, в множество желаемых значений тех же атрибутов. Под желаемым значением следует понимать зна-

чение, которое приобретает атрибут при описании реализованного требования, проверенного соответствующими тестами. В соответствии с этим принципом организованы практически все современные подходы к УТ в ИТ-сфере.

1.3 Анализ современных способов описания архитектуры информационной системы

Как показано в подразд. 1.1, ключевым системным описанием, определяющим особенности анализа и синтеза ИС, является описание архитектуры создаваемой ИС. В связи с этим целесообразно рассмотреть развитие современных трактовок понятия «архитектура системы» и прикладные аспекты применения данного понятия в процессах ЖЦ ИС.

Первоначально большинство определений понятий «Архитектура ПО» и «Архитектура системы» были сосредоточены на структурном аспекте архитектуры по аналогии с идеями архитектуры в строительной отрасли [43, 44]. При этом большинство определений фокусировалось на артефактах, используемых для описания архитектуры [45]. Под архитектурой здесь обычно понимался «комплекс или тщательно разработанная структура чего-либо; концептуальная структура и логическая организация компьютера или компьютерной системы» [46]. В 2000-х гг. данная точка зрения на архитектуру нашла воплощение в работах специалистов компании Microsoft «Metropolis» и «Metropolis and SOA Governance», проводящих аналогии между эволюцией ИТ и процессами эволюции городов и промышленности [47, 48].

Одной из первых попыток узаконить максимально общее определение понятия «Архитектура» для ИТ-сферы является стандарт IEEE 1471:2000. В нем предлагается следующее определение: «Архитектура: фундаментальная организация системы, воплощенная в ее компонентах, их взаимоотношении друг с другом и с окружающей средой, а также принципы, определяющие про-

ектирование и эволюцию системы». Однако в последующих редакциях этого стандарта данное определение было серьезно изменено [49]. Кроме того, данное определение многими исследователями использовалось, расширялось и трактовалось в измененном виде в самых различных формах [50-61].

Современное представление понятия «архитектура системы» зафиксировано в версии стандарта ISO/IEC/IEEE 42010 «Systems and Software Engineering – Architecture Description», принятой в 2011 г. Согласно данному стандарту, архитектура системы – это фундаментальные понятия и свойства системы в окружающей ее среде, воплощенные в ее элементах, отношениях, а также в принципах ее проектирования и развития [45]. Это определение учитывает основные проблемы, возникающие при попытках прикладного использования понятия «Архитектура» стандарта IEEE 1471:2000, и, по замыслу создателей, является максимально общим определением, пригодным для описания архитектур практически любых систем.

Для описания взаимосвязей основных терминов и понятий системы и ее архитектуры в стандарте ISO/IEC/IEEE 42010 предлагается ряд концептуальных моделей, выполненных в нотации ДК UML. Концептуальная модель, описывающая основные взаимосвязи базовых понятий стандарта, приведена на рис. 1.3 [45].

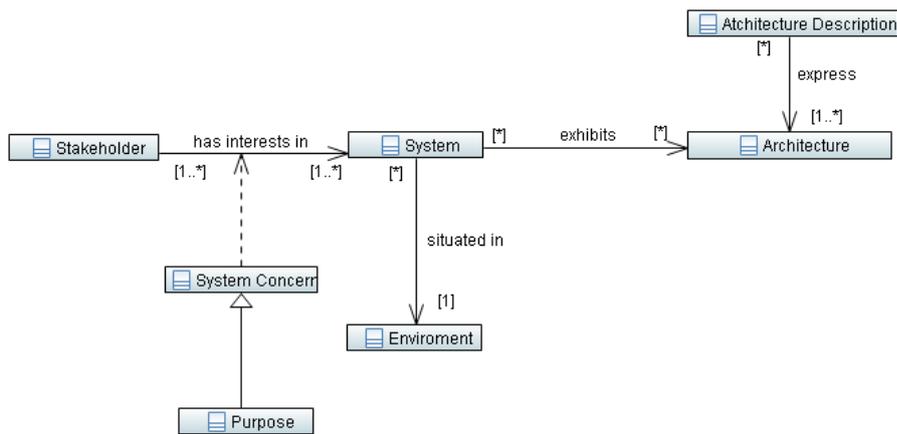


Рисунок 1.3 – Концептуальная модель взаимосвязей между базовыми понятиями стандарта ISO/IEC/IEEE 42010

В соответствии с данной моделью, каждая система (класс System) суще-

ствуется в своей окружающей среде (класс Environment), взаимодействуя с ней. Окружающая среда системы определяет диапазон воздействий на систему (в том числе проектные, операционные, технические, политические, нормативные и другие воздействия на архитектуру системы). Такие воздействия относятся к категории «Проблемы» (класс System Concern).

Система может быть представлена своей архитектурой (класс Architecture), для описания которой выделяется специальный артефакт – описание архитектуры (класс Architecture Description, AD). AD применяется участниками IT-проекта создания ИС для анализа архитектуры системы, сравнения возможных вариантов архитектуры, а также в качестве своего рода «чертежей» для проектирования и конструирования ИС. Концептуальная модель AD показана на следующей диаграмме (рис. 1.4) [45].

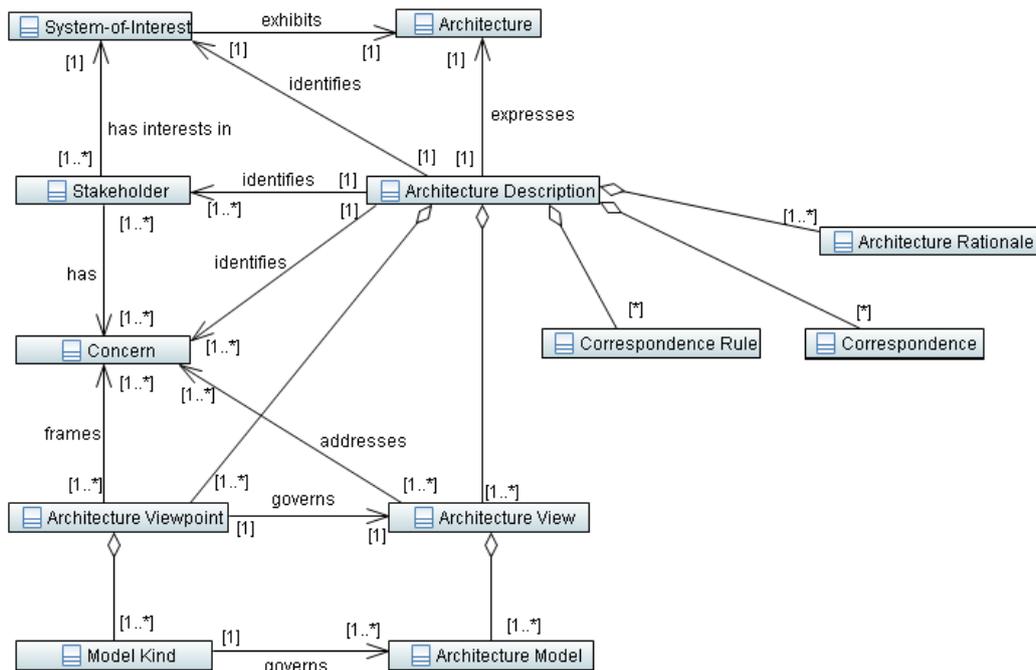


Рисунок 1.4 – Концептуальная модель понятия «Описание архитектуры системы»

Согласно данной модели, AD представляет собой рабочий продукт, который описывает одну из возможных архитектур проблемно-ориентированной

системы и может принимать форму документа, набора моделей или какую-либо другую форму (формат AD стандартом не определен).

Заинтересованная сторона (класс Stakeholder) – это отдельные лица, группы лиц или организации, имеющие проблемы в проблемно-ориентированных системах. Проблема (класс Concern) – это любой интерес в системе. Примерами проблем являются цели (системы), функции, структура, поведение, стоимость, возможности поддержки, безопасность, совместимость.

Архитектурная точка зрения (класс Architecture Viewpoint) представляет собой набор конвенций конструирования, интерпретации, использования и анализа конкретного типа взгляда на архитектуру. Эта точка зрения включает в себя виды моделей, точки зрения на языки и нотации, методы моделирования и аналитические методы в рамках определенного набора проблем. Взгляд на архитектуру (класс Architecture View) в AD отражает архитектуру проблемно-ориентированной системы с точки зрения одной или нескольких заинтересованных сторон для решения конкретных проблем, используя конвенции, устанавливающие данную точку зрения.

Взгляд на архитектуру может быть представлен набором архитектурных моделей (класс Architecture Model). Каждая модель строится в соответствии с конвенциями, учрежденными для данного вида модели, причем эти конвенции, как правило, определяются как часть архитектурной точки зрения. Модели обеспечивают средства для обмена деталями между взглядами на архитектуру и использования множественных обозначений в одном взгляде на архитектуру. Вид модели (класс Model Kind) определяет конвенции, задающие правила построения архитектурных моделей.

В настоящее время выделяется два механизма, используемые для работ с архитектурой и ее AD: АФ и языки описания архитектуры. Концептуальные модели каждого из этих механизмов формируются на основе рассмотренных выше концептуальных моделей архитектуры и AD.

В стандарте ISO/IEC/IEEE 42010 АФ определяется как конвенции, принципы и методы описания архитектуры, установленные в конкретной области

применения и/или сообществом заинтересованных сторон [45]. АФ соответствует международным стандартам IS, если он обозначает:

- информацию, идентифицирующую АФ;
- одну или несколько проблем;
- одну или несколько заинтересованных сторон, у которых есть указанные выше проблемы;
- одну или несколько архитектурных точек зрения (и их технические спецификации, соответствующие стандартам IS);
- правила соответствий, интегрирующие точки зрения;
- условия применимости;
- согласованность АФ с положениями концептуальной модели стандарта ISO/IEC/IEEE 42010.

В целом АФ устанавливает общие практики создания, интерпретации, анализа и использования АД в рамках конкретной области применения или сообщества заинтересованных сторон. Примерами современных АФ являются: MODAF, TOGAF, Kruchten's 4+1 View Model of Software Architecture, RM-ODP [62-65].

Языком описания архитектуры является любая форма выражения, которая может быть использована в АД. Язык описания архитектуры может быть основан на конкретном виде моделей, точке зрения, или на нескольких точках зрения одновременно. Примерами языков описания архитектуры являются языки Rapide, SysML, ArchiMate, ACME, xADL [66-69]. Также для описания архитектуры ПО часто используется язык UML [70-72].

По результатам проведенного выше анализа современного представления термина «архитектура системы» и его описаний можно сделать вывод о том, что языки описания архитектуры могут рассматриваться и как составляющие АФ. Поэтому основное внимание следует сосредоточить на выявлении особенностей современных АФ, используемых в ходе создания ИС, и анализе их достоинств и недостатков.

Для проведения анализа предлагается использовать:

- а) методологию SSADM как пример традиционного АФ разработки ИС

[7, 19, 38, 73];

б) АФ TOGAF 9.1 и RM-ODP как наиболее типичные современные АФ разработки и модернизации ИС [63, 65, 74-79].

Результаты сравнительного анализа данных АФ приведены в табл. 1.1.

На основании данных результатов сделаны следующие выводы:

а) в отличие от методологии SSADM, в современных АФ возможно либо уточнение базового определения архитектуры ИС, данного в стандарте ISO/IEC/IEEE 42010, либо применение собственного определения;

б) архитектура как способ представления ИС занимает промежуточное положение между архитектурой предприятия и его БП и архитектурой программных продуктов, составляющих ПО ИС;

в) архитектура ИС определяется, главным образом, совокупностью представлений обрабатываемых и хранимых данных;

г) ни в методологии SSADM, ни в современных АФ не рассматриваются модели и методы, позволяющие оптимизировать архитектуру системы в рамках ограничений IT-проектов создания ИС или других архитектур;

д) применение в методологии SSADM и АФ TOGAF 9.1 каскадной модели ЖЦ приводит к увеличению затрат времени на создание ИС, особенно в случае изменения требований, выдвигаемых к ИС в ходе ее проектирования;

е) в отличие от методологии SSADM, в современных АФ декларируется возможность ре-использования решений, накопленных в ходе выполнения предыдущих IT-проектов создания или модернизации ИС, однако не предлагаются конкретные методы, модели или инструменты определения допустимости ре-использования конкретных решений.

Сказанное выше определяет необходимость разработки новых или усовершенствования существующих АФ, целью которых является минимизация затрат основных ресурсов на создание, эксплуатацию или модернизацию ИС в рамках процессов ее ЖЦ.

Таблица 1.1 – Результаты сравнительного анализа описаний методологии SSADM, архитектурных фреймворков TOGAF 9.1 и RM-ODP

Параметр	SSADM	TOGAF 9.1	RM-ODP
1	2	3	4
Определение понятия «архитектура»	Явно не определено, но косвенно аналог данного понятия применяется при решении задачи выбора лучшего варианта бизнес-системы	АФ основан на определении стандарта ISO/IEC/IEEE 42010, но введено собственное дополнительное определение	Явного определения не существует, но введена трактовка архитектуры как результата действия структурирующих подходов
Варианты представления (точки зрения) архитектуры	<ul style="list-style-type: none"> - Представление функций системы; - представление событий, возникающих как в самой системе, так и в бизнес-процессах за ее границами; - представление данных системы 	<ul style="list-style-type: none"> - Бизнес-архитектура; - архитектура данных; - архитектура приложений; - технологическая архитектура 	<ul style="list-style-type: none"> - Предпринимательская точка зрения; - информационная точка зрения; - вычислительная точка зрения; - инженерная точка зрения; - технологическая точка зрения
Способ описания архитектуры системы	Совокупность ВМ, описывающих точки зрения на систему	Система архитектурного описания, состоящая из конечных результатов, артефактов и строительных блоков	Взаимосвязанная совокупность описаний системы на пяти языках точек зрения с учетом прозрачностей распределения

Продолжение табл. 1.1

1	2	3	4
Семантическая унификация описаний архитектуры	Устанавливается путем унификации используемых типов ВМ системы	С использованием хранилища объектов архитектуры	С использованием соглашения, устанавливающего архитектурную семантику открытой распределенной обработки данных для различных языков формальных спецификаций
Место требований к системе и управления ими в АФ	Являются инициаторами моделирования функций системы, в том числе – по результатам изменения отдельных требований	Являются связующим звеном между отдельными фазами и представлениями АФ	Роль требований к системе и процесса управления ими явно не определена
Наличие конкретной методологии	Методология SSADM	Методология Architecture Development Method	Конкретная методология проектирования системы не рассматривается
Модель ЖЦ	Вариант каскадной модели ЖЦ	Вариант каскадной модели ЖЦ	Вариант спиральной модели ЖЦ
Возможность ре-использования решений	Не рассматривается	Не рассматривается	Возможность задекларирована, но конкретные механизмы ее реализации не рассматриваются

1.4 Анализ моделей и методов определения объема работ по созданию информационной системы

Как показано на рис. 1.2, решение о заключении договора о поставке ИТ-услуг следует принимать по результатам выполнения процесса определения требований правообладателей. Однако принятие подобного решения в настоящее время чрезвычайно затруднено в силу отсутствия моделей и методов, позволяющих достоверно оценить затраты на выполнение ИТ-проекта создания ИС на основе множества СТ к ИС.

В настоящее время проблему повышения точности расчетов затрат, ведущихся на стадии инициации и в начале стадии планирования ИТ-проектов, можно считать практически нерешенной. Как показано в [80], сложившаяся практика оценивания затрат на выполнение ИТ-проектов считает приемлемым, если в результате оценивания размера ИТ-продукта величина затрат на выполнение ИТ-проекта по созданию данного продукта будет находиться в следующих диапазонах (см. рис. 1.5):

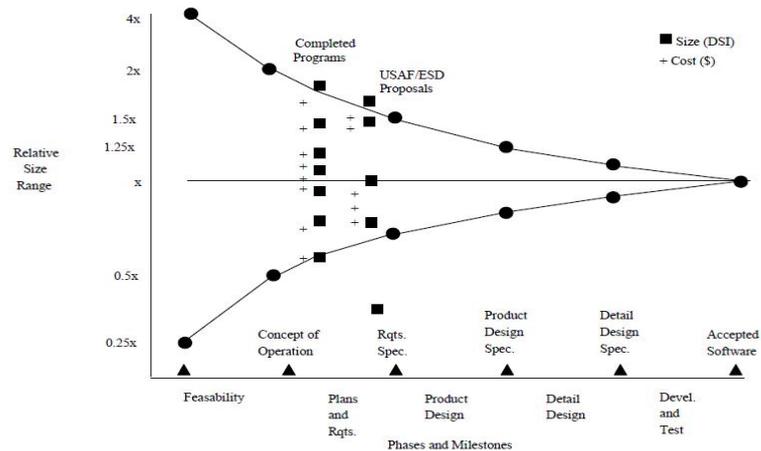


Рисунок 1.5 – Изменение точности оценок размера ИТ-продукта и затрат на выполнение ИТ-проекта по созданию данного продукта в зависимости от стадий ИТ-проекта

а) в начале стадии инициации (feasibility – оценка реализуемости ИТ-проекта)

– в диапазоне от 0,25 до 4 величин реальных значений бюджета проекта;

б) в конце стадии инициации – в диапазоне от 0,5 до 2 величин реальных значений бюджета проекта;

в) в начале стадии планирования (plans and requirements – планирование и сбор требований к ИТ-продукту) – в диапазоне от 0,5 до 2 величин реальных значений бюджета проекта;

г) в конце стадии планирования – в диапазоне от 0,7 до 1,7 величин реальных значений бюджета проекта.

В общем случае все множество моделей и методов, используемых для оценивания затрат на выполнение ИТ-проектов, может быть разделено на следующие подмножества [10]:

- а) экспертная оценка;
- б) оценка по аналогам;
- в) параметрическая оценка;
- г) оценки по трем точкам (PERT);
- д) анализ резервов.

По результатам анализа каждого из подмножеств моделей и методов оценивания затрат можно сделать вывод, что наиболее многочисленными являются модели и методы, формализованные в достаточно слабой степени. Такие модели и методы позволяют оценивать затраты на выполнение ИТ-проекта, исходя из собственного опыта ИТ-компаний. Это обеспечивает косвенное повышение точности расчетов за счет учета особенностей ИТ-компаний, не находящих формального отражения в используемых моделях и методах оценивания. Однако подобные методы и модели оценивания затрат не позволяют с желаемой точностью рассчитывать значения затрат на выполнение ИТ-проекта, основываясь только на множестве требований к ИС.

В настоящее время существует достаточно большое количество параметрических моделей и методов оценивания затрат на выполнение ИТ-проекта по созданию ИС, среди которых как основные можно выделить [11, 81]:

- а) метод функциональных точек;

- б) метод объектных точек;
- в) модели COSOMO и COSOMO II.

Модель COSOMO II представляет собой набор формул с параметрами, характеризующими конкретный IT-проект. Для оценивания трудозатрат в ходе инициации IT-проекта рекомендуется использовать модель COSOMO II Early Design, которая имеет следующий вид [80]:

$$PM = A \times [Size']^B \times \prod_{i=1}^7 EM_i + PM_M, \quad (1.1)$$

где PM - прогнозируемое значение трудозатрат на выполнение иницируемого IT-проекта;

A - поправочный коэффициент модели COSOMO II, $A = 2,5$;

$Size'$ - количество строк кода иницируемого IT-проекта;

B - коэффициент, отражающий влияние на трудозатраты иницируемого IT-проекта масштаба и экономичности этого проекта.

EM_i - драйвер затрат IT-проекта, $i = 1, \dots, 7$;

PM_M - значение трудозатрат на реинжиниринг и конверсию программного кода, которые предполагаются выполнять в рамках иницируемого IT-проекта.

Правила вычисления значений элементов модели (1.1) детально рассмотрены в [80].

Модель COSOMO II на данный момент является самой совершенной моделью расчета трудозатрат, времени и потребности в персонале на выполнение IT-проекта. Однако одним из наиболее серьезных недостатков модели является использование показателя «тысяча условных строк кода» (KLOC), который не в полной степени отражает особенности современных подходов к созданию ИС. Так, например, использование данной метрики породило следующий парадокс: несмотря на то, что в настоящее время высокоуровневые языки программирования применяются для разработки программных продуктов гораздо чаще, чем

низкоуровневые, метрики «строка кода» утверждают выгодность обратного [82].

В основу классического метода функциональных точек легло представление реализуемой системы, как множества элементов, которые принадлежат к двум основным подгруппам: данные и транзакции. Данные можно разделить на два класса: внутренние логические файлы (ILF) и файлы внешних интерфейсов (EIF). Транзакции делятся на следующие классы: внешние входы (EI), внешние выходы (EO) и внешние запросы (EQ). В классическом методе функциональных точек характеристики именно этих элементов позволяют определять количество нескорректированных функциональных точек (UFP), исходя из которого рассчитывается количество функциональных баллов для создаваемой системы. Расчет объема продукта в функциональных баллах по методу функциональных точек осуществляется по формуле [83]:

$$DFP = \left(\sum_{i=1}^{n_{ILF}} UFP_i + \sum_{j=1}^{n_{EIF}} UFP_j + \sum_{k=1}^{n_{EI}} UFP_k + \sum_{l=1}^{n_{EO}} UFP_l + \sum_{m=1}^{n_{EQ}} UFP_m + CFP \right) \times \left((0,01 \times \sum_{p=1}^{14} DI_p) + 0,65 \right), \quad (1.2)$$

где DFP - количество скорректированных функциональных точек, оценивающих объем ИТ-продукта, создаваемого в результате выполнения оцениваемого ИТ-проекта, с учетом влияния общесистемных характеристик этого продукта;

$\sum_{i=1}^{n_{ILF}} UFP_i$ - сумма нескорректированных функциональных точек, характеризующая сложность структур данных, используемых создаваемым

ИТ-продуктом в ходе выполнения своих функций;

n_{ILF} - количество структур данных, которые используются создаваемым ИТ-продуктом в ходе выполнения своих функций;

$\sum_{j=1}^{n_{EIF}} UFP_j$ - сумма нескорректированных функциональных точек, характери-

зующая сложность структур данных, поступающих из внешних по отношению к создаваемому IT-продукту файлов в ходе выполнения им своих функций;

n_{EIF} - количество данных, поступающих из внешних по отношению к создаваемому IT-продукту файлов;

$\sum_{k=1}^{n_{EI}} UFP_k$ - сумма нескорректированных функциональных точек, характери-

зующая сложность внешних интерфейсов для ввода данных, используемых создаваемым IT-продуктом для ввода данных от пользователя или устройства сбора информации в ходе выполнения своих функций;

n_{EI} - количество внешних интерфейсов для ввода данных;

$\sum_{l=1}^{n_{IEO}} UFP_l$ - сумма нескорректированных функциональных точек, характери-

зующая сложность внешних интерфейсов для вывода данных, используемых создаваемым IT-продуктом для вывода результатов выполнения своих функций;

n_{EO} - количество внешних интерфейсов для вывода данных;

$\sum_{m=1}^{n_{EQ}} UFP_m$ - сумма нескорректированных функциональных точек, характери-

зующая сложность внешних запросов, используемых создаваемым IT-продуктом для предоставления информации пользователю или другой системе в ответ на заданные условия поиска;

n_{EQ} - количество внешних запросов;

CFP - количество дополнительных функциональных точек, описывающих функции, необходимые в ходе внедрения создаваемого IT-продукта;

DI_p - системная характеристика создаваемого IT-продукта.

Правила расчета элементов выражения (1.2) подробно описаны в [83].

Результаты анализа классического метода функциональных точек поз-

воляют утверждать, что показатель «функциональные точки» также является условным. Причина условности заключается, главным образом, в использовании экспертных оценок для расчета значений элементов выражения (1.2). Кроме того, правила расчета нескорректированных функциональных точек не учитывают особенности разработки прикладного ПО для конкретной функции ИС. Поэтому данный показатель не может быть принят в качестве основного показателя, характеризующего объем работ по созданию ИС.

С помощью метода объектных точек рассчитывается оценка объема создаваемого программного кода в процессах инициации IT-проектов создания программных продуктов. Этот метод более прост, чем классический метод функциональных точек и дает более точные результаты. Каждая объектная точка может относиться к одной из следующих групп: экраны, отчеты и 3GL-модули создаваемого программного продукта [80, 84].

В общем случае расчет трудозатрат на создание программных продуктов методом объектных точек осуществляется по формуле

$$PM = \frac{NOP}{PROD}, \quad (1.3)$$

где NOP - количество новых объектных точек, вычисляемое следующим образом:

$$NOP = \frac{(OP) \times (100 - \%reuse)}{100}; \quad (1.4)$$

OP - количество объектных точек, определяемых в результате оценивания элементов разрабатываемого продукта (экранов, отчетов и 3GL-модулей);

$\%reuse$ - процент повторно используемых экранов, отчетов и 3GL-модулей ранее разработанного приложения;

$PROD$ - коэффициент производительности команды исполнителей проекта.

Правила расчета и определения элементов выражений (1.3) и (1.4) детально описаны в [80].

Несмотря на свою простоту и удобство, метод объектных точек также не лишен серьезных недостатков. Среди этих недостатков прежде всего следует отметить то, что в ходе инициации IT-проекта создания ИС информацию о программных элементах, которые предполагается использовать, получить затруднительно.

Следует отметить, что недостатки рассмотренных выше и других методов и моделей оценивания затрат на создание ИС обуславливают проведение работ по исследованию новых и усовершенствованию существующих методов и моделей данного назначения [85, 86]. Сложность выполнения данных работ усугубляется еще и тем, что рассмотренные выше модели и методы используют разные оценки объема работ по созданию ИС на ранних стадиях выполнения IT-проектов. Поэтому в ходе решения данной проблемы возникает необходимость решения хотя бы одной из следующих задач:

- а) задачи согласования различных моделей и методов определения объема работ по созданию ИС [86, 87];
- б) задачи разработки новых моделей и методов определения объема работ по созданию ИС.

Результаты решения этих задач должны обеспечить согласование точек зрения различных сторон на выполняемый IT-проект и ИС, которая будет создана в результате его выполнения.

1.5 Анализ современных систем управления требованиями

В настоящее время системы и ИТ, ориентированные на автоматизацию работ по формированию, анализу и УТ, достаточно распространены. Область применения таких систем и ИТ охватывает широкий диапазон IT-проектов – от сложных проектов с большим количеством участников до простейших проектов,

выполняемых одним человеком или же командой из нескольких исполнителей.

Классическим примером системы УТ, ориентированной на применение в ИТ-проектах создания систем, является пакет IBM Rational RequisitePro. Это инструмент, который способствует процессу УТ, позволяет вводить, обновлять, отслеживать и просматривать требования на протяжении всего ЖЦ проекта. Применяя комбинированный подход, основанный на использовании документов и БД, IBM Rational RequisitePro предоставляет мощную и удобную в использовании систему для УТ. Данная система позволяет создавать, организовывать, отслеживать требования и назначать им приоритеты, обеспечивает формирование документов, новых типов требований и атрибутивных МТ этих типов. Управление изменениями в IBM Rational RequisitePro поддерживается путем отслеживания связей между требованиями. Кроме того, система IBM Rational RequisitePro предоставляет условия для совместной работы команды проекта [88].

Однако система IBM Rational RequisitePro не лишена недостатков, затрудняющих ее применение в ходе планирования и выполнения ИТ-проектов. Среди этих недостатков особо отмечают [89]:

- а) очень плохо реализована интеграция файла MS Word, содержащего описание варианта использования, и описания требования в репозитории системы;
- б) существуют серьезные проблемы использования web-интерфейса для поддержки интеграции со средствами моделирования, в частности с IBM Rational XDE и Rational Rose;
- в) невозможно задание прав доступа различных пользователей к отдельным требованиям;
- г) необходимо формирование и ведение глоссария терминов ИТ-проекта, списка действующих лиц вариантов использования, а также реестра регламентирующих документов в репозитории требований, а не как вариантов специфических требований к системе;
- д) в качестве наиболее удобного варианта структурирования требований система предлагает только вариант, при котором требования группируются в оглавления (пакеты) и детализируются требованиями следующего уровня (связь предок

– потомок (parent – child)), что затрудняло назначение артефактов других типов;

е) трассировка между артефактами (в том числе требованиями) должна быть только одного типа, что ограничивает возможность контроля выполнения требований;

ж) отсутствует удобный механизм организации связи репозитория требований с внешними ресурсами, расположенными в Internet, а также с внутренними ресурсами.

В настоящее время компания IBM развивает решения по автоматизации работы с требованиями в рамках продуктов новой линейки – систем УТ IBM Rational DOORS. Последним на сегодняшний день продуктом данной линейки является система УТ IBM Rational DOORS Next Generation. Выполненная в виде единой платформы для совместной работы больших коллективов, система IBM Rational DOORS Next Generation помимо стандартных для IBM функций формирования, анализа и УТ, предоставляет такие возможности [90]:

- улучшенная среда совместной работы для УТ в ходе разработки комплексных, встроенных систем;

- web-базированный инструментарий для поддержки описаний требований в виде текстов, диаграмм ВМ и графических объектов;

- поддержка атрибутивных моделей ФТ как пользовательских историй и нефункциональных требований, а также возможность определения атрибутивных моделей других (в том числе пользовательских) типов требований к системе;

- автоматизация работы по внесению изменений не только в отдельные требования к системе, но и в требования и артефакты, связанные с изменяемым требованием;

- расширенные функции планирования и управления ЖЦ требований и продукта в соответствии с отраслевыми стандартами и нормативными требованиями;

- возможность интеграции системы как с продуктами компании IBM, так и с продуктами других разработчиков (например, JIRA).

- совместимость различных версий IBM Rational DOORS;

- интегрированные инструменты УТ для более удобного проектирования, разработки и тестирования ПО.

Система УТ 3SL Cradle применяется не только в IT-сфере, но в других отраслях промышленности, в которых осуществляется проектирование на

основе требований. Данная система выполняет следующие операции [91]:

- ввод требований в систему различными участниками проекта;
- выделение требований в ходе обработки документа «Техническое задание на систему» или аналогичных документов;
- трассировка требований и производных от них моделей и артефактов;
- управление изменениями требований;
- поиск и выявление ошибок в описаниях требований;
- возможность обработки больших объемов разнородной информации;
- генерация документов, основанных на проектных данных («Техническое задание на систему», устав проекта), публикация отчетов и статических сайтов;
- моделирование с применением различных нотаций ВМ;
- интеграция с системами управления проектами.

Система УТ Devprom Requirements представляет собой инструментальное средство, позволяющее решать такие задачи [92]:

- разработка требований с вовлечением всех участников команды, начиная с самых ранних этапов ЖЦ продукта;
- фиксация технических решений в форме фотографий, скриншотов, UML-моделей и формул с автоматической нотификацией об изменениях;
- подготовка постановки для разработчиков на релиз или итерацию, параллельная работа с требованиями для нескольких релизов, с использованием версий и бейзлайнов документов;
- контроль и поддержка целостности системных требований, тестовой и пользовательской документации, с использованием трассировок и матриц трассируемости;
- формирование библиотеки шаблонов и требований для реиспользования в проектах;
- выгрузка документов требований для подписания или согласования с внешними участниками проекта;
- поддержка Agile- и Lean-ориентированных процессов разработки ПО с максимальным вовлечением аналитиков.

Выполнение данных задач реализовано в системе Devprom Requirements

с применением web-технологий, позволяющих автоматизировать выполнение следующих операций [92]:

- совместное создание полноценных документов требований из браузера;
- обсуждение и рецензирование требований всей командой разработчиков;
- документирование UML-моделей, формул и алгоритмов;
- версионирование и трассировка требований на проектные артефакты;
- разработка, тестирование и документирование, основанные на требованиях;
- загрузка и выгрузка требований в формате MS Word;
- настройка процессов работы над требованиями;
- сбор и визуализация метрик для анализа проблем повышения продуктивности.

Система УТ am.Requirements ориентирована на одиночных исполнителей или небольшие команды IT-проектов, в которых УТ занимается один человек. Данная система позволяет автоматизировать выполнение следующих операций [93]:

- формирование описаний требований в виде дерева или в табличном виде;
- формирование и поддержка системы ссылок между требованиями;
- формирование собственных типов и атрибутивных описаний требований;
- связь с описанием требования графических файлов и документов;
- отображение описаний требований в виде HTML-документов.

В целом анализ особенностей систем УТ позволяет выделить следующие базовые особенности, характерные для IT-продуктов подобного назначения [94]:

- а) единицей управления в подобных системах является требование как элемент (точнее, как артефакт) описания создаваемой системы;
- б) требования как элементы описания создаваемой системы могут быть связаны друг с другом различными видами связей;
- в) требования и связи между ними могут быть представлены для пользователя различными способами в зависимости от задач, решаемых пользователем (формирование требований, анализ требований, трассировка требований и т.п.);
- г) механизм отслеживания изменений в отдельных требованиях основан на стандартных инструментах трассировки требований.

Реализация этих особенностей позволяет в ходе применения систем УТ

добиться следующих выгод [94]:

- а) возможность контролировать ошибки проектирования;
- б) сокращение времени и повышение точности формирования оценки изменений, вносимых в проект в ходе его планирования и выполнения.

В основе подавляющего большинства инструментальных средств УТ положена идея централизованного репозитория требований. Такой репозиторий чаще всего представляет собой иерархические структуры данных, в которых на верхнем уровне находятся требования к системе в целом. Эти требования (что должна делать система, как должна работать система) детализируются через формулирование более мелких требований, уточняющих отдельные аспекты требований к системе в целом. Аналогичным образом реализована и модель изменений требований [89].

Основным потребителем систем УТ по-прежнему являются компании, разрабатывающие крупные сложные системы. Однако в последнее время подобные системы начали активно применяться и в сравнительно небольших ИТ-компаниях. Особенно выгодным считается применение данных систем в проектах создания крупных или сложных систем с целью повышения вероятности обнаружения ошибок на ранних этапах создания системы.

В целом же в [94] отмечается, что в настоящее время наблюдается разделение продуктов, которые традиционно относятся к системам УТ, на два различных направления. Продукты первого направления работают с требованиями априорно заданных типов и не могут быть расширены или адаптированы под требования конкретных пользователей. Продукты второго направления основаны на принципе открытых моделей и позволяют описывать не только требования к создаваемой системе, но и другие артефакты, используемые для описания системы (например, риски, тесты, ошибки и т.п.). В общем случае продукты второго направления ориентированы на хранение следующих видов описания создаваемой системы:

- а) исходные требования (потребности);
- б) производные требования – текстовые или графические спецификации;

в) модели реализации системы и ее элементов.

Однако ни одна из существующих систем УТ не ориентирована на автоматизацию синтеза АД создаваемой системы. Кроме того, проблема принятия решений о ре-использовании ранее реализованных требований в новых ИТ-проектах по-прежнему далека в данных системах от эффективного решения. Решение этих проблем позволит значительно сократить затраты времени и других ресурсов за счет автоматизированного формирования оптимальных (или рациональных) вариантов АД создаваемой системы, в которых ранее реализованные требования ре-используются в максимально возможной для создаваемой системы степени.

1.6 Выводы по результатам проведенного анализа и постановка задач исследования

Результаты анализа современного состояния проблем формирования требований и создания описания архитектуры сложных ИС позволяют сделать следующие выводы.

Во-первых, современное представление ИС ориентировано на формирование описания системы как множества отдельных не связанных друг с другом требований; при этом задачи системной интеграции не решаются и не ставятся, а опыт предыдущих проектов создания аналогичных систем не рассматривается.

Во-вторых, в настоящее время не существует ни единого общепринятого определения понятия «требование к ИС», ни единой системы классификации этих требований, ни единой формальной модели требований. Существующие методы сбора требований к ИС в своем подавляющем большинстве не позволяют формировать формальные описания требований, пригодные для автоматизированной обработки или ре-использования в ИТ-проектах создания последующих ИС.

В-третьих, существующие МТ к ИС представляют собой не формальные

конструкции, а совокупности описаний отдельных аспектов требований, наличие которых обусловлено типовыми процессами ЖЦ ИС, выполняющимися в ходе IT-проекта создания ИС.

В-четвертых, современные подходы к УТ основаны не на формальных моделях или критериях, а на оценке участниками IT-проекта создания ИС степени соответствия значений атрибутивного описания каждого требования желаемым для участников значениям тех же атрибутов.

В-пятых, существующие определения понятия «архитектура системы» и описания этой архитектуры допускают одновременное представление системы и ее элементов через множество моделей, описывающих как структурный, так и поведенческий аспекты данной системы.

В-шестых, существующие методологии и АФ создания ИС, хотя и признают необходимость существования АД создаваемой ИС, однако не решают и не ставят на формальном уровне задачу оптимизации архитектуры ИС и ее АД, а также не позволяют решать проблему минимизации затрат на создание ИС в соответствии с ее АД, предлагаемым и одобренным участниками IT-проекта.

В-седьмых, существующие методы и модели определения объема работ по созданию ИС ориентированы на отображение точек зрения отдельных участников IT-проекта создания ИС и не позволяют сформировать объективную оценку объема работ на ранних стадиях создания ИС.

Данные выводы позволяют предположить, что требования к ИС в настоящее время рассматриваются как уникальные артефакты, не связанные друг с другом в единое представление создаваемой ИС. Данная точка зрения значительно затрудняет использование АФ или языков описания архитектуры в ходе проектирования архитектуры создаваемой ИС из-за дополнительных затрат на семантическое согласование отдельных требований и их описаний. Особенно отрицательно этот недостаток сказывается в ходе выполнения IT-проектов создания или адаптации ИС по заказам предприятий одной и той же сферы деятельности, когда априорная уникальность всех без исключения требований к ИС не позволяет эффективно использовать опыт, накопленный

в ходе выполнения предыдущих подобных IT-проектов.

Выделенные проблемы являются следствием недостаточных теоретических исследований в области формирования и анализа требований как основы для синтеза АД создаваемой ИС. Основные публикации зарубежных исследователей, среди которых следует выделить работы К.И. Виггера, А. Коёрна, Д. Леффингуэлла и Д. Уидрига, специалистов компаний «Microsoft», «Hewlett-Packard», «IBM», посвящены, главным образом, осмыслению практического опыта формирования и анализа требований к ИС [13, 14, 16, 21, 25, 34, 37]. В этих и целом ряде других публикаций практически не раскрыты теоретические основы, определяющие допустимые области применения освещаемых моделей, методов и технологий работы с требованиями к ИС.

В Украине основное внимание исследователей в области ИС и ИТ сосредоточено, главным образом, на решении проблем разработки и модернизации ИС различного назначения. Среди этих работ особо следует отметить исследования, выполненные под руководством А.А. Павлова и В.И. Гриценко. В этих работах рассматриваются теоретические основы и прикладные технологии моделирования и проектирования ИС различного назначения [95-100]. Особо стоит отметить работу С.Ф. Теленика [101], посвященную разработке концепции, моделей алгоритмов и средств технологии создания информационно-управляющих систем для организационно-технических объектов управления. Полученные в этой работе результаты охватывают практически весь жизненный цикл ИС.

Проблемам разработки ИС управления организационными и организационно-техническими объектами посвящены работы исследователей, выполненные под руководством В.М. Левыкина [102], О.Е. Федоровича [103]. Проблема управления предприятием как совокупностью интеллектуальных производственных систем рассмотрена в работе В.В. Казимира [104]. Созданию моделей, методов и ИТ адаптивной разработки и реинжиниринга информационно-управляющих систем для организационно-технических объектов посвящена работа Н.В. Ткачука [105]. Решению проблемы разработки формальной осно-

вы ИТ адаптивного синтеза оптимальной технологической схемы сборки системных технических объектов посвящено исследование И.П. Гамаюна [106]. Решению теоретико-прикладных вопросов создания гарантоспособных сервис-ориентированных Web-систем посвящена работа А.В. Горбенко [107].

Однако подавляющее большинство этих исследований не уделяет должного внимания формальному решению проблем формирования и анализа требований к ИС. Аналогичная ситуация наблюдается и в других странах. В настоящее время основное внимание исследователей сосредотачивается на разработке методов, позволяющих создавать и обрабатывать формальные МТ к ИС. Например, в [108] показано применение разработанного метода для анализа требований к ERP-ИС. В [109] рассматриваются вопросы создания и применения метода для сценарно-основанного анализа требований к системе. В [110] рассматривается применение процессной модели для выявления требований к специализированной ИС выявления и обработки знаний из бизнес-информации. Проблема автоматизации работ по формализации требований к ИС рассматривается в [111]. В то же время анализ этих исследований показывает, что проблема преобразования неформальных потребностей в формальные МТ решается в них преимущественно на концептуальном уровне [112].

В поисках способов формального описания требований в настоящее время изучается применение моделей и методов добычи знаний из описаний потребностей и требований к системе. Один из вариантов формализации ПрО корпоративных ИС путем семантического моделирования рассматривается в [113]. В [114] рассматривается вопрос применения методов добычи знаний для преобразования высказываемых потребностей в описания требований к ИС в здравоохранении. В Украине проблемы онтологического моделирования ИС и процессов их разработки рассмотрены в работах, выполненных под руководством А.В. Палагина. В этих работах рассматриваются вопросы, связанные с интеллектуализацией разработки и сопровождения ИС [115-117]. Разрабатываемые модели, методы и средства позволяют реализовать вопросы создания знание-ориентированных ИС, особенности которых определяются

онтологическими моделями ПрО и их изменениями. Однако подобные работы также отличаются минимальным уровнем формализации описываемых МТ к системе и методов формирования и анализа этих требований.

В целом же следует признать, что к настоящему времени проблема формального описания требований, методов их формирования и анализа – в том числе и на уровне добываемых из требований знаний – не имеет эффективного решения. Поэтому проведение научно-прикладных исследований, позволяющих сократить затраты на выполнение IT-проектов за счет формального анализа требований к ИС и синтеза на их основе АД создаваемой ИС, учитывающего накопленный в предыдущих IT-проектах опыт создания систем подобного назначения, является актуальным как с теоретической, так и с практической точек зрения.

Решение этой проблемы предлагается представлять как поиск компромиссного варианта АД ИС, удовлетворяющего основным целям и потребностям Поставщика и Потребителя IT-услуг (далее – Поставщика и Потребителя). В общем случае в ходе создания ИС Поставщик предоставляет Потребителю информацию о возможных вариантах конфигурации IT-услуг, образующих АД ИС. При этом каждая IT-услуга может быть описана как продукт, имеющий определенную стоимость. Эта стоимость в общем случае определяется:

- а) стоимостью создания IT-услуги, реализующей эту IT-услугу совокупности IT-сервисов и комплекта документов по каждому из этих IT-сервисов;
- б) стоимостью адаптации IT-услуги, реализующей эту IT-услугу совокупности IT-сервисов и комплекта документов по каждому из этих IT-сервисов к требованиям Потребителя.

Потребитель получает от Поставщика информацию о возможных вариантах конфигурации этих услуг в рамках ИС и стоимости этих вариантов для Потребителя. Каждый из предлагаемых Поставщиком вариантов конфигурации Потребитель оценивает с точки зрения эффекта от внедрения и эксплуатации этого варианта в своих БП. Затем Потребитель выбирает из совокупности вариантов конфигурации ИС тот, который обеспечивает максимальный эффект.

Такое представление позволяет определить основную концепцию управле-

ния взаимоотношениями Поставщика и Потребителя как проверенную на практике концепцию управления взаимоотношениями с клиентами (Client Relationship Management, CRM). В то же время необходимо учесть существование сложившегося представления работ по разработке, внедрению, сопровождению и модификации ИС как проекта, ориентированного на выполнение требований Потребителей при ограничениях на качество, стоимость создания или адаптации, время создания или адаптации и содержание работ по созданию или адаптации ИС как совокупности IT-услуг [8].

Сказанное выше позволяет описать глобальную цель деятельности Поставщика в ходе формирования требований к ИС, их анализа и построения на их основе АД создаваемой ИС следующим образом [118]: «Целью Поставщика является предоставление Потребителю такого набора IT-услуг, который наилучшим образом соответствует комплексу требований, определенных Потребителем и особенностями БП Потребителя, при ограничениях на стоимость, время выполнения и качество работ по предоставлению данного набора IT-услуг Потребителю». Глобальную цель деятельности Потребителя в ходе формирования требований к ИС, их анализа и построения на их основе АД создаваемой ИС можно описать следующим образом [118]: «Целью Потребителя является поиск и организация взаимодействия с таким Поставщиком, который предоставляет набор IT-услуг, наилучшим образом соответствующий комплексу требований, определенных Потребителем и особенностями БП Потребителя, при ограничениях на стоимость, время выполнения и качество работ по предоставлению данного набора IT-услуг, выбранных Поставщиком».

В качестве основного показателя, определяющего уровень достижимости глобальных целей Поставщика и Потребителя в ходе формирования требований к ИС, их анализа и построения на их основе АД создаваемой ИС, является показатель, характеризующий степень удовлетворения требований к ИС, выдвинутых Потребителем и принятых к исполнению Поставщиком. При этом будем исходить из того, что выдвигаемые Потребителем разнородные

требования будут являться элементами множества требований к конкретной ИС Tr_{IS} . В общем случае это множество будет иметь следующий вид:

$$Tr_{IS} = (tr_1, tr_2, \dots, tr_i, \dots, tr_n), \quad (1.5)$$

где tr_i – обобщенное описание i -го требования к ИС, отдельной ИТ-услуге ИС или же к отдельному ИТ-сервису ИТ-услуги ИС;

i – идентификатор обобщенного описания требования к ИС tr_i , $i = \overline{1, n}$;

n – количество требований, выдвинутых к ИС, ее ИТ-услугам и ИТ-сервисам этих услуг.

Тогда степень удовлетворения каждого требования к ИС tr_i можно в общем случае описать оператором $r(tr_i)$, который ставит в соответствие описанию требования tr_i число в диапазоне $[0...1]$. При этом ситуация $r(tr_i) = 0$ означает, что требование tr_i было выдвинуто Потребителем, но не было выполнено Поставщиком по тем или иным причинам. Ситуация $r(tr_i) = 1$ означает, что требование tr_i было выдвинуто Потребителем и выполнено Поставщиком полностью. Ситуация $0 < r(tr_i) < 1$ означает, что требование tr_i было выдвинуто Потребителем и выполнено Поставщиком лишь частично.

Для оценки стоимости выполнения требования введем оператор $pay(r(tr_i))$. В случае, если $r(tr_i) = 0$, данный оператор ставит в соответствие описанию требования tr_i нулевое значение. В случае, если $0 < r(tr_i) \leq 1$, данный оператор ставит в соответствие описанию требования tr_i положительное значение, характеризующее величину финансовых затрат $pay(r(tr_i))$ на выполнение требования tr_i со степенью удовлетворения $r(tr_i)$.

Для оценки времени выполнения требования введем оператор $t(r(tr_i))$. В случае, если $r(tr_i) = 0$, данный оператор ставит в соответствие описанию требования tr_i нулевое значение. В случае, если $0 < r(tr_i) \leq 1$, данный опера-

тор ставит в соответствие описанию требования tr_i положительное значение, характеризующее величину затрат времени $t(r(tr_i))$ на выполнение требования tr_i со степенью удовлетворения $r(tr_i)$.

Для оценки качества выполнения требования введем оператор $q(r(tr_i))$. В случае, если $r(tr_i) = 0$, данный оператор ставит в соответствие описанию требования tr_i нулевое значение. В случае, если $0 < r(tr_i) \leq 1$, данный оператор ставит в соответствие описанию требования tr_i положительное значение, характеризующее качество $q(r(tr_i))$ выполнения требования tr_i со степенью удовлетворения $r(tr_i)$.

Введенные обобщенные описания требований и операторы позволяют сформулировать обобщенное формализованное описание глобальной цели Поставщика в ходе формирования требований к ИС, их анализа и построения на их основе АД создаваемой ИС как задачу следующего вида:

$$F_{Pr} = \sum_{i=1}^n r^{Pr}(tr_i) \rightarrow \max, \quad (1.6)$$

$$\begin{cases} \sum_{i=1}^n \alpha_i^{Pr} \text{pay}(r^{Pr}(tr_i)) \geq \text{pay}^* \left(\sum_{i=1}^n r^{Pr}(tr_i) \right); \\ \sum_{i=1}^n \beta_i^{Pr} t(r^{Pr}(tr_i)) \leq t^* \left(\sum_{i=1}^n r^{Pr}(tr_i) \right); \\ \sum_{i=1}^n \gamma_i^{Pr} q(r^{Pr}(tr_i)) \geq q^* \left(\sum_{i=1}^n r^{Pr}(tr_i) \right). \end{cases} \quad (1.7)$$

где Pr – обозначение Поставщика ИТ-услуг и соответствующих ИТ-сервисов;

$r^{Pr}(tr_i)$ – степень удовлетворения требования к ИС tr_i с точки зрения Поставщика;

α_i^{Pr} – нормативный коэффициент стоимости выполнения требования tr_i , учитывающий индивидуальные особенности Поставщика;

$pay^* \left(\sum_{i=1}^n r^{Pr}(tr_i) \right)$ – минимально допустимая для Поставщика величина стоимости выполнения множества требований к ИС;

мости выполнения множества требований к ИС;

β_i^{Pr} – нормативный коэффициент длительности выполнения требования tr_i , учитывающий индивидуальные особенности Поставщика;

$t^* \left(\sum_{i=1}^n r^{Pr}(tr_i) \right)$ – максимально допустимое для Поставщика время выполнения

множества требований к ИС;

γ_i^{Pr} – нормативный коэффициент качества выполнения требования tr_i , учитывающий индивидуальные особенности Поставщика;

$q^* \left(\sum_{i=1}^n r^{Pr}(tr_i) \right)$ – минимально допустимое для Поставщика качество выполнения

множества требований к ИС.

Обобщенное формализованное описание глобальной цели Потребителя в ходе формирования требований к ИС, их анализа и построения на их основе АД создаваемой ИС может быть представлено задачей следующего вида:

$$F_U = \sum_{i=1}^n r^U(tr_i) \rightarrow \max, \quad (1.8)$$

$$\left\{ \begin{array}{l} \sum_{i=1}^n \alpha_i^U pay(r^U(tr_i)) \leq pay^* \left(\sum_{i=1}^n r^U(tr_i) \right); \\ \sum_{i=1}^n \beta_i^U t(r^U(tr_i)) \leq t^* \left(\sum_{i=1}^n r^U(tr_i) \right); \\ \sum_{i=1}^n \gamma_i^U q(r^U(tr_i)) \geq q^* \left(\sum_{i=1}^n r^U(tr_i) \right), \end{array} \right. \quad (1.9)$$

где U – обозначение Потребителя;

$r^U(tr_i)$ – степень удовлетворения требования к ИС tr_i с точки зрения Потребителя;

бителя;

α_i^U – нормативный коэффициент стоимости выполнения требования tr_i , учитывающий индивидуальные особенности Потребителя;

$pay^* (\sum_{i=1}^n r^U (tr_i))$ – максимально допустимая для Потребителя величина стоимости выполнения множества требований к ИС;

β_i^U – нормативный коэффициент длительности выполнения требования tr_i , учитывающий индивидуальные особенности Потребителя;

$t^* (\sum_{i=1}^n r^U (tr_i))$ – максимально допустимое для Потребителя время выполнения множества требований к ИС;

γ_i^U – нормативный коэффициент качества выполнения требования tr_i , учитывающий индивидуальные особенности Потребителя;

$q^* (\sum_{i=1}^n r^U (tr_i))$ – минимально допустимое для Потребителя качество выполнения множества требований к ИС.

Однако такое формализованное представление целей Поставщика и Потребителя имеет излишне общий характер и усложняется следующими факторами:

а) ни один представитель Потребителя, как правило, не имеет представления обо всей последовательности операций всех ИТ-услуг ИС;

б) Потребитель принимает решение о целесообразности внедрения и эксплуатации ИС, прежде всего исходя из прагматической ценности ИТ-услуг предлагаемого Поставщиком варианта ИС (в данном случае прагматическая ценность определяется как степень соответствия ИТ-услуг ИС задачам управления БП Потребителя);

в) непосредственные исполнители работ по созданию ИС со стороны Поставщика не обязаны знать особенности БП конкретного Потребителя, непосредственно не влияющие на разработку соответствующих ИТ-услуг;

г) Поставщик принимает решение о целесообразности разработки варианта

ИС, исходя из своих возможностей формирования набора IT-услуг, выполняющих множество требований Потребителя, при условии выполнения системы ограничений (1.7) в целом с возможностью нарушения отдельных неравенств.

Эти факторы позволяют рассматривать целевые функции (1.6) и (1.8), определяющие решения Поставщика и Потребителя при выполнении процесса проектирования архитектуры ИС, как функции выбора оптимального набора IT-услуг ИС при соблюдении Поставщиком и Потребителем определенных ограничений. При этом выполнение Поставщиком остальных групп требований будет относиться к числу дополнительных ограничений, частично установленных Потребителем, а частично – самим Поставщиком в процессе разработки ИС.

Такое представление целей Поставщика и Потребителя определяет цель диссертационной работы как решение актуальной научно-прикладной проблемы разработки методологических основ, моделей, методов и информационной технологии, позволяющих сформировать рациональный вариант описания архитектуры создаваемой информационной системы с максимально возможным удовлетворением функциональных требований Поставщика и Потребителя и минимальными затратами на реализацию.

Для достижения поставленной цели необходимо решить такие задачи:

- провести анализ существующих подходов к формированию требований и созданию описаний архитектуры сложных информационных систем управления организационно-техническими объектами и процессами;
- разработать методологические основы создания информационных систем, позволяющие установить четкую взаимосвязь между требованиями к системе, описанием архитектуры системы и проектными решениями, реализующими требования в рамках предлагаемого описания архитектуры системы;
- разработать математические модели, описывающие множество требований к информационной системе на разных стадиях их формирования;
- разработать математическую модель паттерна проектирования требования к информационной системе;
- разработать математическую модель архитектурного фреймворка мак-

ропроектирования информационной системы и сформулированного функционального требования на уровне знаний как основного элемента архитектурного фреймворка;

- разработать математические модели, описывающие функциональные требования к информационной системе на уровнях информации и знаний;

- разработать методы формирования функциональных требований на уровнях информации и знаний;

- разработать метод синтеза вариантов описаний архитектуры информационной системы;

- разработать математическую модель синтеза описания рациональной архитектуры создаваемой информационной системы;

- разработать математическую модель унифицированного элемента информационной системы и метод формирования его описания, позволяющие оценить объем работ по созданию информационной системы;

- разработать методы анализа функциональных требований к информационной системе;

- разработать информационную технологию формирования описания архитектуры информационных систем, основанную на предлагаемых методологических основах, моделях и методах;

- разработать программный комплекс для реализации информационной технологии формирования описания архитектуры информационных систем в виде инструментального средства управления требованиями;

- провести апробацию результатов исследований в макропроектировании информационной системы управления организационно-техническим процессом.

РАЗДЕЛ 2

МЕТОДОЛОГИЧЕСКИЕ ОСНОВЫ СЕРВИСНОГО ПОДХОДА К СОЗДАНИЮ ИНФОРМАЦИОННОЙ СИСТЕМЫ

2.1 Сервисный подход к созданию информационных систем: основные термины

Как показано в разд. 1, с современной точки зрения ИС являются системами, образованными множеством взаимосвязанных и экономически целесообразных ИТ-услуг, своевременное предоставление и выполнение которых обеспечивает эффективную и качественную деятельность управляемого объекта и/или процесса. Подход, на основании которого сформировано данное представление ИС, здесь и в дальнейшем будем называть сервисным подходом. Данный подход позволяет установить соответствие экономически и технически обоснованных потребностей заинтересованных лиц в автоматизации управления объектами и/или процессами множеству конкретных ИТ-услуг и реализующих эти услуги ИТ-сервисов, позволяющих удовлетворить эти потребности.

Контекстная ДК, отражающая основные взаимосвязи базовых понятий сервисного подхода к созданию ИС как множества взаимосвязанных элементов, способных оказывать заинтересованным лицам услуги по выполнению необходимых для достижения главной цели деятельности ИС операций над данными, приведена на рис. 2.1. Каждая проблема заинтересованных лиц (класс «Concern») может быть выражена множеством потребностей различной природы (класс «Need»), высказываемых заинтересованными сторонами в различной форме. Каждая из этих потребностей может быть удовлетворена одной или несколькими ИТ-услугами (класс «IT-accommodation») или же одним или несколькими ИТ-сервисами (класс «IT-service»), реализующими соответствующие ИТ-услуги. Тогда любая ИС может быть представлена как система, состоящая из множества ИТ-услуг, а любая ИТ-услуга может быть

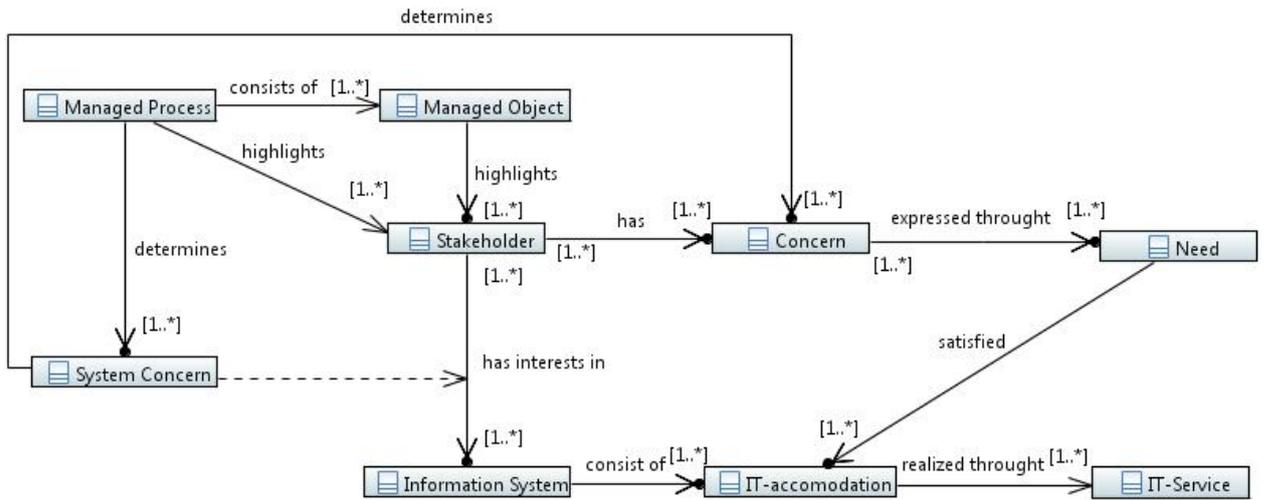


Рисунок 2.1 – Контекстная диаграмма классов, описывающая взаимосвязи базовых понятий сервисного подхода к созданию информационных систем

реализована множеством IT-сервисов [119].

Подобное представление ИС позволяет установить единую взаимосвязанную терминологическую базу для описания функциональной структуры (ФС) и обеспечивающей части ИС. Основным термином, определяющим главные особенности сервисного подхода, является термин «IT-услуга». Однако этот термин нуждается в уточнении, поскольку является, по сути, вариантом перевода оригинального термина «IT-service», который используется также, например, для обозначения элементов ПО ИС или ИТ, реализующих законченную функцию предоставления или обработки данных, переводя эти данные из одного целостного состояния в другое [120]. Поэтому во избежание терминологической путаницы понятия «IT-сервис» и «IT-услуга» предлагается разделить [118, 119].

Понятием «IT-сервис» в процессах разработки, внедрения, сопровождения и модернизации ИС следует описывать *совокупность различных средств КСА, реализующих законченную операцию предоставления или обработки данных, переводя эти данные из одного целостного состояния в другое, используя при этом стандартные платформу-независимые интерфейсы.*

Тогда понятие «IT-услуга» следует использовать для описания *взаимосвязанной совокупности IT-сервисов, которая предоставляется для выпол-*

нения отдельной работы процесса предприятия/организации или для управления этой работой [118, 119].

Данные определения позволяют рассматривать ИТ-услугу как аналог функциональной задачи (ФЗ) ИС, результат которой используется персоналом в ходе выполнения отдельной работы процесса предприятия/организации или в ходе управления этой работой. В то же время возможность представления процессов предприятия/организации как отдельных работ, которые, в свою очередь, могут быть разделены на более мелкие работы, позволяет трактовать ИТ-услугу и как аналог функции ИС, целью которой в общем случае является повышение эффективности и/или качества выполнения и/или управления соответствующим процессом предприятия/организации. Что же касается термина «ИТ-сервис», то под ним, в данном случае, следует понимать один из способов организации КСА, который обеспечивает выполнение функций и отдельных ФЗ ИС.

Такое установление соответствия позволяет по-иному сформулировать приведенное выше определение термина «ИТ-услуга». Согласно этой формулировке, *ИТ-услуга – это самостоятельная ФЗ ИС, использование которой для выполнения отдельной работы процесса предприятия/организации или для управления этой работой экономически и технически целесообразно*. Такое определение позволяет рассматривать главную цель деятельности ИС предприятия или организации как формирование и отображение единого целостного информационного представления процессов этого предприятия или организации в результате оказания совокупности ИТ-услуг [118, 119].

Сформулированные определения понятий «ИТ-услуга» и «ИТ-сервис» позволяют выделить и уточнить основные уровни представления понятия «ИС», определение которого приведено выше. Представление ИС как совокупности функциональной и обеспечивающей частей было выработано еще в 1970-1980-х гг. [121–124]. Однако применение сервисного подхода к созданию ИС заставляет пересмотреть данные уровни представления. Вместо уровней представления ИС, выделяющих функциональную и обеспечивающую части,

предлагается использовать следующие уровни представления создаваемой ИС:

- уровень управляемых объектов и/или процессов;
- общесистемный уровень;
- уровень IT-услуг;
- уровень IT-сервисов.

Уточнения определения понятия «ИС» на каждом из предлагаемых уровней представления создаваемой ИС приведены в табл. 2.1. Указанные в скобках традиционные уровни представления позволяют соотнести предлагаемые уточнения со сложившимися определениями функциональной и обеспечивающей части ИС [119].

В соответствии с уточнениями понятия «ИС», приведенными в табл. 2.1, можно уточнить понятия «архитектура» и «AD» в рамках сервисного подхода к созданию ИС. Следует отметить, что в настоящее время понятие «архитектура» может использоваться для описания фундаментальных понятий и свойств как ИС, так и функций ИС или же отдельных видов обеспечений ИС. В результате возникает терминологическая путаница, которая зачастую приводит к неоправданно узкому взгляду на разрабатываемую ИС (например, представление ИС как системы, архитектура которой определяется исключительно особенностями ПО этой системы). Кроме того, выделение и уточнение понятий «архитектура» и «AD» на уровнях IT-услуг и IT-сервисов может позволить выделить и формализовать основные принципы создания и развития комплексов IT-услуг и IT-сервисов ИС.

Контекстная ДК взаимосвязи базовых понятий, используемых для описания архитектуры информационной системы на основе сервисного подхода, приведена на рис. 2.2 [119].

Основываясь на рассмотренном в подразд. 1.3 определении понятия «архитектура системы», понятие «архитектура IT-услуг» можно определить следующим образом: *«Архитектура IT-услуг – фундаментальные понятия и свойства комплекса IT-услуг, образующих ИС в окружающей её среде, воплощенные в*

Таблица 2.1 – Определения понятия «информационная система» на разных уровнях представления

Уровень представления	Определение понятия «информационная система»
1	2
Уровень управляемых объектов и/или процессов	ИС – один из механизмов управляемого объекта и/или процесса, формирующий и отображающий единое целостное информационное представление этого объекта и/или процесса в соответствии с поставленными перед данным механизмом целями
Общесистемный уровень	ИС – система, состоящая из персонала и КСА и направленная на формирование и отображение единого целостного информационного представления объекта или процесса в соответствии с поставленными перед системой целями
Уровень IT-услуг (функциональная часть ИС)	ИС – система IT-услуг, использование которой для объекта или процесса экономически или технически целесообразно, направленная на формирование и отображение единого целостного информационного представления этого объекта или процесса в соответствии с поставленными перед системой целями
Уровень IT-сервисов (обеспечивающая часть ИС)	ИС – система IT-сервисов, реализующих совокупность операций предоставления или обработки данных, направленная на формирование и отображение единого целостного информационного представления объекта или процесса в соответствии с поставленными перед системой целями

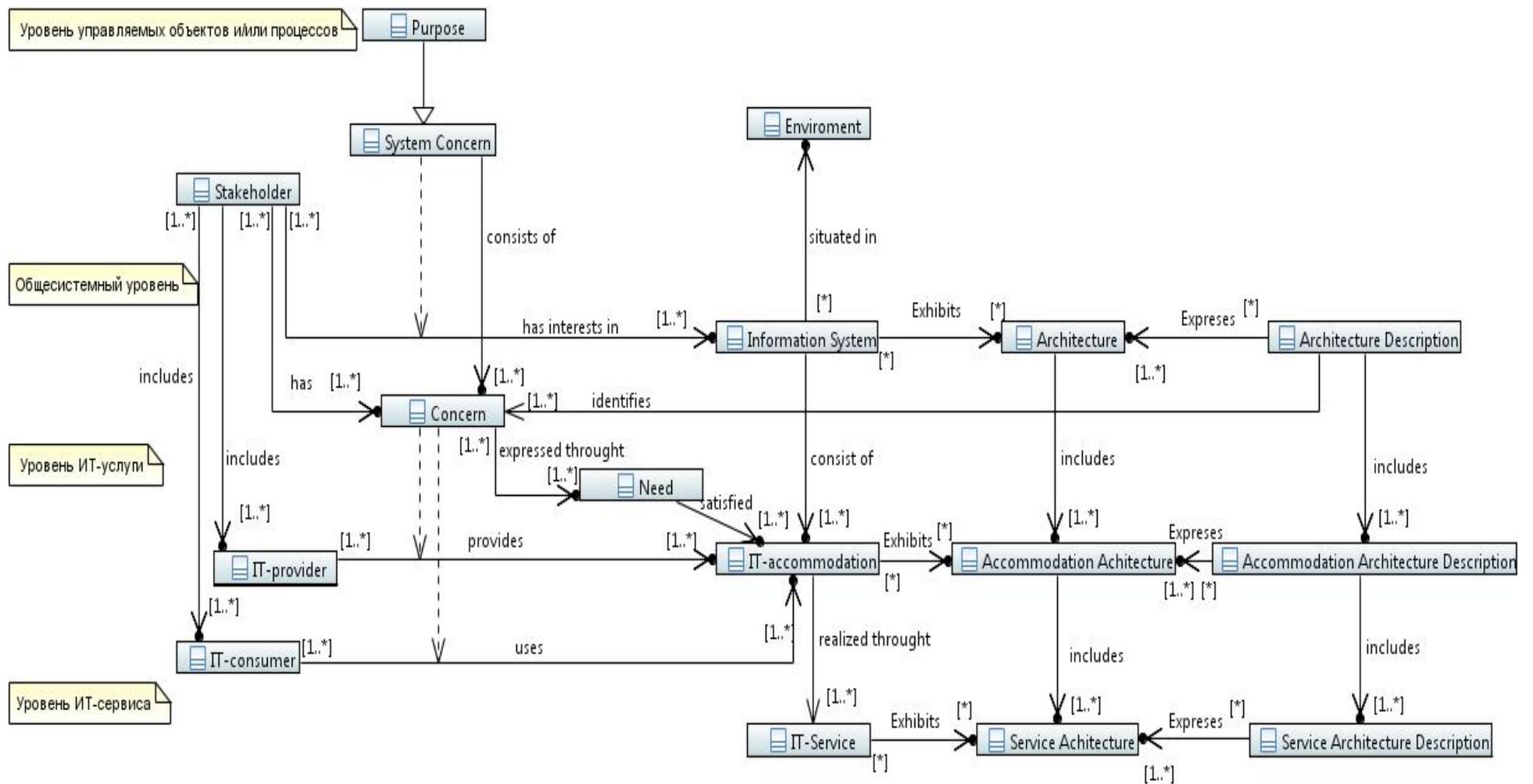


Рисунок 2.2 – Контекстная диаграмма классов взаимосвязи базовых понятий, используемых для описания архитектуры информационной системы на основе сервисного подхода

элементах и отношениях этого комплекса, а также в принципах его проектирования и развития» [119].

Тогда понятие «архитектура ИТ-сервисов» можно определить следующим образом: *«Архитектура ИТ-сервисов – фундаментальные понятия и свойства комплекса ИТ-сервисов и связывающих их интерфейсов, образующих ИС в окружающей её среде, воплощенные в элементах и отношениях этого комплекса, а также в принципах его проектирования и развития» [119].*

Не менее важным уточнением является уточнение понятия «заинтересованные стороны», которое на уровне управляемых объектов и/или процессов обозначает всех участников работ по созданию, внедрению, эксплуатации и модернизации ИС как совокупности ИТ-услуг. В настоящее время наиболее полный и точный перечень таких участников приведен в стандарте ГОСТ 34.601-90. Данный перечень не позволяет жестко разграничивать роли участников и допускает, в зависимости от условий создания АС, возможность различных совмещений функций заказчика, разработчика, поставщика и других организаций, участвующих в работах по созданию АС [125].

Однако предлагаемый ГОСТ 34.601-90 перечень участников ориентирован на вполне определенное представление ИС, сформировавшееся в 1980-х гг. Представление ИС как совокупности ИТ-услуг заставляет пересмотреть основные подходы к выделению ролей участников. С этой целью необходимо рассмотреть современные зарубежные стандарты, посвященные проблемам создания и эксплуатации ИС.

Одним из таких стандартов является международный стандарт ISO/IEC 20000. Это международный процессно-ориентированный стандарт управления ИТ-услугами. Цель его разработки заключалась в создании универсальных критериев, с помощью которых любая фирма или служба, предоставляющая ИТ-услуги, сможет оценивать их эффективность и выполнение требований заказчиков с учетом их бизнеса [126]. В стандарте ISO/IEC 20000 основным понятием, определяющим участников процессов оказания ИТ-услуг, является понятие «поставщик услуг». Это понятие определяет органи-

зацию, которая поставила перед собой цель соответствовать требованиям ISO/IEC 20000. В некоторых случаях поставщиком услуг может являться внутреннее подразделение организации, например, подразделение, занимающееся внедрением и эксплуатацией ИТ в организации [127].

Однако такое определение требует дополнительного уточнения ролей, в которых может выступать поставщик ИТ-услуг в процессах создания, внедрения, эксплуатации и модернизации ИС. Практика создания и эксплуатации ИС различного назначения показывает, что права на отдельные виды обеспечений ИС у различных участников процессов создания, внедрения, эксплуатации и модернизации ИТ-услуг могут серьезно различаться.

Поэтому предлагается в общем случае понятием «поставщик ИТ-услуг» описывать *организации и/или подразделения, которые основной целью своей деятельности считают эффективное и качественное выполнение работ по созданию, внедрению, эксплуатации и модернизации ИС, отдельных ИТ-услуг и реализующих эти услуги ИТ-сервисов*. В связи с этим целесообразно уточнить понятие «поставщик ИТ-услуг» для основных процессов ЖЦ ИС. Для такого уточнения лучше всего использовать рассмотренный выше перечень участников. На основе этого перечня в разных процессах ЖЦ ИС организации или подразделения-поставщики ИТ-услуг могут выступать в следующих ролях [118]:

а) разработчик – осуществляет работы по созданию ИТ-услуг, представляет заказчику совокупность научно-технических услуг на разных стадиях и этапах создания, а также разрабатывает и поставляет различные ИТ-сервисы и их элементы, обеспечивающие реализацию соответствующих ИТ-услуг;

б) поставщик – изготавливает и поставляет отдельные элементы ИТ-сервисов, обеспечивающих реализацию соответствующих ИТ-услуг, по заказу разработчика или заказчика;

в) дилер разработчика – занимается адаптацией предлагаемых разработчиком ИТ-услуг и реализующих эти услуги ИТ-сервисов к особенностям страны, региона и т.п.;

г) организация по внедрению ИТ-услуг – осуществляет работы по адап-

тации предлагаемых разработчиком ИТ-услуг и реализующих эти услуги ИТ-сервисов к особенностям конкретного заказчика и внедрению адаптированных ИТ-услуг и ИТ-сервисов в процессы заказчика;

д) ИТ-подразделение – осуществляет работы по непосредственному обслуживанию, настройке и администрированию ИТ-услуг и ИТ-сервисов, эксплуатируемых в процессах заказчика, работы по управлению этими ИТ-услугами и ИТ-сервисами, работы по выявлению отклонений фактических эксплуатационных характеристик ИТ-услуг от проектных значений, а также осуществление взаимодействия между заказчиком и другими участниками процессов создания, внедрения, эксплуатации и модернизации ИТ-услуг;

е) организация по сопровождению ИТ-услуг и ИТ-сервисов – осуществляет работы по сопровождению отдельных ИТ-услуг и реализующих их ИТ-сервисов, установлению причин возникновения отклонений фактических эксплуатационных характеристик ИТ-услуг от проектных значений, устранению выявленных причин и недостатков ИТ-услуг и ИТ-сервисов, а также работы по внесению необходимых изменений в документацию на ИТ-услуги и ИТ-сервисы.

Что касается понятия «потребитель ИТ-услуг», то оно в явном виде в стандарте ISO/IEC 20000 не определено. В общем случае понятием *«потребитель ИТ-услуг и реализующих эти услуги ИТ-сервисов»* следует описывать *организации и/или подразделения, которые нуждаются в этих услугах и сервисах и используют их в своих процессах для достижения целей, поставленных перед этими организациями и/или подразделениями* [118].

Уточним понятие «потребитель ИТ-услуг» для процессов ЖЦ ИС аналогично рассмотренному ранее уточнению понятия «поставщик ИТ-услуг». Тогда в разных процессах ЖЦ ИС организации или подразделения-потребители ИТ-услуг могут выступать в следующих ролях [118]:

а) заказчик ИС или отдельных ИТ-услуг – определяет основные требования к разрабатываемой ИС или же отдельным ИТ-услугам и обеспечивает финансирование работ по созданию, внедрению, эксплуатации и модернизации ИС и соответствующих ИТ-услуг, а также выполнение отдельных работ по со-

зданию, внедрению и модернизации ИС или отдельных ИТ-услуг;

б) пользователь ИТ-услуг и реализующих эти услуги ИТ-сервисов – обеспечивает приемку работ по созданию и эксплуатации ИТ-услуг (в том числе в рамках ИС) и реализующих эти услуги ИТ-сервисов, а также выполнение отдельных работ по созданию, внедрению и модернизации ИТ-услуг и реализующих эти услуги ИТ-сервисов.

Сформулированные выше определения основных понятий сервисного подхода к созданию ИС позволяют установить основные особенности представлений требований к ИС в процессах, формирующих архитектуру ИС как совокупности взаимосвязанных и экономически целесообразных ИТ-услуг, своевременное предоставление и выполнение которых обеспечивает эффективную и качественную деятельность управляемого объекта и/или процесса.

2.2 Требования к информационной системе: определение и классификация

Изложенные выше материалы позволяют утверждать, что требования к ИС, функции ИС, соответствующие этим требованиям, а также решения по видам обеспечений ИС, реализующие эти функции, являются откликами на высказанные заинтересованными сторонами проблемы, порожденные ПрО, и потребности в конкретных возможностях ИС, которая должна решить эти проблемы. Именно эти отклики формируют множество проектных решений, образующих проект создаваемой ИС. Таким образом, сервисный подход к созданию ИС формирует точку зрения на требования к ИС как на первоначальные описания проектных решений ИС, определяющие все последующие их описания.

Такое видение сервисного подхода приводит к необходимости уточнения определения понятия «требование к ИС» и системы классификации этих требований. Поскольку, как показано в п. 1.2.1, самостоятельно существующего

определения понятия «требование к ИС» не существует, в качестве базового определения будем использовать определение понятия «требование», сформулированное в стандарте IEEE 610.12-1990. Данное определение является наиболее общим, пригодным практически для любой отрасли компьютерных наук.

Основываясь на данном определении, понятие «*требование к ИС*» можно сформулировать следующим образом [128]:

а) условие или возможность, необходимые Потребителю ИТ-услуг для решения проблемы или достижения цели;

б) условие или возможность, которой должна обладать ИС или компонент ИС (ИТ-услуга, ИТ-сервис) с точки зрения Поставщика или Потребителя ИТ-услуг, соответствующие договору, стандарту, спецификации или другому официальному документу;

в) документированное представление условия или возможности, подобных описанным в первых двух определениях.

Предлагаемое определение является общим определением всех возможных групп требований к ИС, отдельным ИТ-услугам и отдельным ИТ-сервисам, реализующим эти ИТ-услуги. Для использования в конкретных проектах создания ИС это определение нуждается в уточнениях и дополнениях, для чего предлагается ввести иерархию следующих групп требований [128]:

а) бизнес-требование – условие или возможность, необходимые Потребителю ИТ-услуг для достижения своих бизнес-целей в результате выполнения соответствующих БП;

б) требование к ИС как аспекту бизнеса – условие или возможность, необходимые Потребителю ИТ-услуг для автоматизированного выполнения БП в соответствии с поставленными бизнес-целями;

в) требование к ИС в целом – условие или возможность, которой должна обладать ИС с точки зрения Поставщика или Потребителя ИТ-услуг, соответствующие договору, стандарту, спецификации или другому официальному документу;

г) ФТ к ИТ-услуге – возможность, которой должна обладать ИТ-услуга с точки зрения Поставщика или Потребителя ИТ-услуг, соответствующая дого-

вору, стандарту, спецификации или другому официальному документу;

д) нефункциональное требование к ИТ-услуге – условие, которому должна отвечать ИТ-услуга с точки зрения Поставщика или Потребителя ИТ-услуг, соответствующее договору, стандарту, спецификации или другому официальному документу;

е) ФТ к ИТ-сервису – возможность, которой должен обладать ИТ-сервис с точки зрения Поставщика или Потребителя ИТ-услуг, соответствующая договору, стандарту, спецификации или другому официальному документу;

ж) нефункциональное требование к ИТ-сервису – условие, которому должен отвечать ИТ-сервис с точки зрения Поставщика или Потребителя ИТ-услуг, соответствующее договору, стандарту, спецификации или другому официальному документу.

Схема иерархии данных групп требований приведена на рис. 2.3.

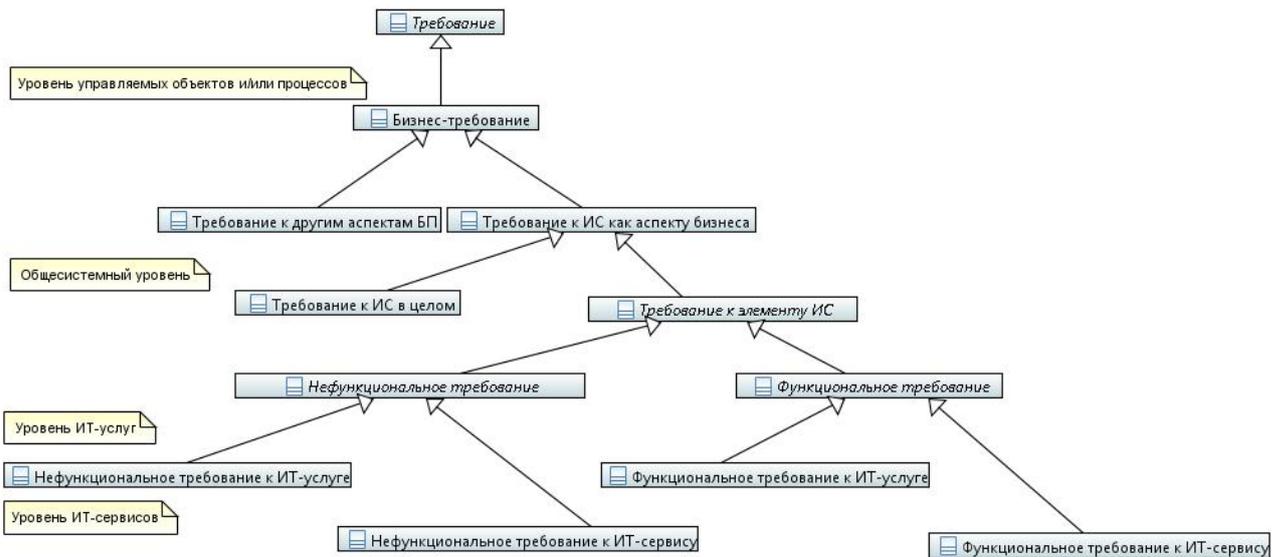


Рисунок 2.3 – Схема иерархии предлагаемых групп требований к информационной системе

В общем случае множество всех требований к ИС Tr_{IS} можно формально описать следующим образом:

$$Tr_{IS} = (tr_1, tr_2, \dots, tr_i, \dots, tr_n), \tag{2.1}$$

где tr_i – обобщенное описание i -го требования к ИС, отдельной ИТ-услуге ИС
или же к отдельному ИТ-сервису ИТ-услуги ИС;

i – идентификатор обобщенного описания требования к ИС tr_i , $i = \overline{1, n}$;

n – количество требований, выдвинутых к ИС, ее ИТ-услугам и ИТ-сервисам этих услуг.

Предложенная иерархия групп требований к ИС позволяет представить множество Tr_{IS} следующим образом [129]:

$$Tr_{IS} = Tr_{IS}^B \cup Tr_{IS}^{IB} \cup Tr_{IS}^S \cup Tr_{IS}^f \cup Tr_{IS}^{nf} \cup Tr_{IS}^{fw} \cup Tr_{IS}^{nfw}, \quad (2.2)$$

где Tr_{IS}^B – подмножество бизнес-требований, в общем случае имеющее вид

$$Tr_{IS}^B = (tr_1, tr_2, \dots, tr_i, \dots, tr_a), a < n; \quad (2.3)$$

Tr_{IS}^{IB} – подмножество требований к ИС как аспекту бизнеса, в общем случае имеющее вид

$$Tr_{IS}^{IB} = (tr_{a+1}, tr_{a+2}, \dots, tr_i, \dots, tr_b), b < n; \quad (2.4)$$

Tr_{IS}^S – подмножество требований к ИС в целом, в общем случае имеющее вид

$$Tr_{IS}^S = (tr_{b+1}, tr_{b+2}, \dots, tr_i, \dots, tr_c), c < n; \quad (2.5)$$

Tr_{IS}^f – подмножество ФТ к ИТ-услугам, в общем случае имеющее вид

$$Tr_{IS}^f = (tr_{c+1}, tr_{c+2}, \dots, tr_i, \dots, tr_e), e < n; \quad (2.6)$$

Tr_{IS}^{nf} – подмножество нефункциональных требований к ИТ-услугам, в общем случае имеющее вид

$$Tr_{IS}^{nf} = (tr_{e+1}, tr_{e+2}, \dots, tr_i, \dots, tr_g), g < n; \quad (2.7)$$

Tr_{IS}^{fw} – подмножество ФТ к ИТ-сервисам, в общем случае имеющее вид

$$Tr_{IS}^{fw} = (tr_{g+1}, tr_{g+2}, \dots, tr_i, \dots, tr_k), k < n; \quad (2.8)$$

Tr_{IS}^{nfw} – подмножество нефункциональных требований к ИТ-сервисам, в общем случае имеющее вид

$$Tr_{IS}^{nfw} = (tr_{k+1}, tr_{k+2}, \dots, tr_i, \dots, tr_n). \quad (2.9)$$

Количество бизнес-требований равно a . Количество требований к ИС как к аспекту БП равно $(b - a)$. Количество требований к ИС в целом равно $(c - b)$. Количество ФТ к ИТ-услугам равно $(e - c)$. Количество нефункциональных требований к ИТ-услугам равно $(g - e)$. Количество ФТ к ИТ-сервисам равно $(k - g)$. Количество нефункциональных требований к ИТ-сервисам равно $(n - k)$.

Для любых двух подмножеств Tr_{IS}^x и Tr_{IS}^y , выделенных в множестве требований к ИС (2.2), должно выполняться следующее условие:

$$Tr_{IS}^x \cap Tr_{IS}^y = \emptyset. \quad (2.10)$$

Это условие подразумевает невозможность рассмотрения описания любого из требований к ИС как принадлежащего одновременно к двум или более

группам требований. Иными словами, ни одно требование не может быть, например, описано как функциональное и нефункциональное одновременно.

Сопоставление предлагаемых определения понятия «требование к ИС» и классификации групп требований к ИС с описаниями групп требований к ПО позволяет утверждать, что ИС, с точки зрения программной инженерии, представляет собой прикладное ПО, которое предназначено для формирования и отображения единого целостного информационного представления управляемого объекта или процесса. Таким образом, предлагаемые определение понятия «требование к ИС» и классификация групп требований соответствуют принятому в международных стандартах определению понятия «требование» и сложившимся представлениям групп требований к ПО как к более общему классу систем и определяют особенности представления ИС как совокупности ИТ-услуг, реализуемых соответствующими ИТ-сервисами.

2.3 Концепция представления требований к информационной системе

Предложенные в подразд. 2.2 формализованные описания множества требований к ИС и подмножеств отдельных групп требований являются обобщенными, не раскрывающими особенности выполнения процессов, непосредственно работающих с требованиями и процесса проектирования архитектуры ИС. Поэтому возникает необходимость определить основную концепцию представления требований к ИС, устанавливающую особенности их детализированного формализованного описания.

Одной из наиболее важных проблем, возникающих при попытках описать требования к ИС формальными способами, является изначальная неполнота требований. Эта неполнота вызвана, прежде всего, стремлением Потребителя получить только те ИТ-услуги, эксплуатация которых обеспечит скорейшее появление прибыли от автоматизируемого процесса. При этом Потребитель не задумывает-

ся о необходимости дополнительных IT-услуг, необходимых для обеспечения эффективной согласованной работы основных IT-услуг. Не стоит сбрасывать со счетов и «забывчивость» Потребителя, возможность слабой ориентации Потребителя в предложениях Поставщика, а также плохое знание Поставщиком особенностей ПрО Потребителя. Кроме того, следует помнить о возможности изменений автоматизируемого процесса с течением времени (например, в результате реинжиниринга или постепенного улучшения).

Решение этой проблемы становится возможным в том случае, если предположить, что Поставщик и Потребитель в каждом конкретном проекте создания ИС имеют дело не с простым множеством требований к ИС, полностью или частично упорядоченным определенным образом, а с более общей совокупностью требований. При этом в такую совокупность должны входить не только требования к ИС, выдвинутые Потребителем и принятые к исполнению Поставщиком, но и так называемые «забытые» (невьявленные или несвоевременно выявленные) требования, требования, которые Поставщик может выдвинуть к ИС, исходя из своего видения ПрО и т.п. Все эти требования могут раньше или позже стать элементами множества требований к ИС (2.2) в ходе очередной итерации процессов, непосредственно работающих с требованиями.

Для обозначения и дальнейшего формализованного описания такой расширенной совокупности требований к ИС предлагается использовать существующее понятие «универсум». В современном научном мышлении это понятие часто обозначает фиксированную систему (или системы) объектов, к которым относятся утверждения (высказывания) какой-либо теории. Другой аспект анализа идей, связанных с универсумом, – различение стандартного и нестандартного, конструктивного и неконструктивного универсума, а также онтологического и гносеологического универсума рассуждения (отнесение последнего непосредственно к теории, а первого – к возможным моделям этой теории) [130].

Однако подобные представления оставляют в стороне самого исследователя, который своими действиями (или бездействием) формирует универсум с использованием конкретных АФ, методов и методик. В то же время опыт

разработки многих программных систем и, в частности, ИС, показывает, что выбор тех или иных инструментальных средств выполнения этих проектов в существенной степени определяется предпочтениями конкретных разработчиков. Аналогичная ситуация возникает, например, при выборе моделей и методов анализа и проектирования программных систем [131].

Исходя из этого, предлагается интерпретировать понятие «универсум» применительно к потребностям разработки ИС следующим образом: «универсум – это «множество всех возможных систем», из которых лишь одна – исследуемая система – реальна, а все остальные (в том числе и проектируемую ИС) возможно осмыслить только логическим путем, то есть непротиворечивым образом представляя возможные факты или связи исследуемой системы». Развивая данную интерпретацию, понятие «универсум» в теории и практике создания, внедрения, эксплуатации и модернизации ИС можно определить следующим образом: *«Универсум – совокупность данных, информации и знаний об исследуемой системе, объекте или процессе, как известных, так и неизвестных исследователю, в распоряжении которого имеется конечное множество методов получения и обработки этих данных, информации и знаний»* [129].

Данное определение порождает целый ряд следствий, из которых для формализованного описания требований к ИС и процессов проектирования архитектуры важны, прежде всего [129]:

а) *следствие 1* из данного определения: точность описания исследуемой системы, объекта или процесса стремится к максимуму в том случае, если объем неизвестных исследователю данных, информации и знаний об этой системе, объекте или процессе стремится к минимуму;

б) *следствие 2* из данного определения: исследователь всегда должен допускать, что совокупность неизвестных ему данных, информации и знаний об исследуемой системе, объекте или процессе не является пустой;

в) *следствие 3* из данного определения и следствий 1 и 2: для практического применения потребителем универсум должен обладать не максимальной точностью, а такой точностью, которая позволит на основании совокупности

известных данных, информации и знаний принимать решения с желаемыми для потребителя универсума характеристиками эффективности и качества.

Формально предлагаемое определение универсума U может быть описано следующим образом [129]:

$$U = (\langle D, I, K \rangle, \langle UnD, UnI, UnK \rangle, (F) \rangle), \quad (2.11)$$

где U – обозначение универсума;

D – множество данных, известных исследователю благодаря применению имеющихся в его распоряжении методов формирования и обработки данных;

I – множество информации, известной исследователю благодаря применению имеющихся в его распоряжении методов формирования и обработки информации;

K – множество знаний, известных исследователю благодаря применению имеющихся в его распоряжении методов формирования и обработки знаний;

UnD – множество данных неизвестных исследователю;

UnI – множество информации, неизвестной исследователю;

UnK – множество знаний, неизвестных исследователю;

$(F) : \langle UnD, UnI, UnK \rangle \rightarrow \langle D, I, K \rangle$ – множество отображений, описывающих множество методов, преобразующих неизвестные данные, информацию и знания в известные.

Рассмотрим описание совокупности требований к ИС с использованием универсума (2.11). Согласно определению универсума, множество требований к ИС (2.2) становится известным тогда, когда требования, составляющие это множество, выдвинуты Потребителем, приняты к исполнению Поставщиком и согласованы между Поставщиком и Потребителем. В дальнейшем множество (2.2), используемое для описания известных требований к ИС, будем называть множеством СТ к ИС. Каждое из подмножеств множества СТ к ИС в этом случае может быть определено следующим образом [129]:

$$\begin{aligned}
Tr_{IS}^B &= \langle D_{IS}^B, I_{IS}^B, K_{IS}^B \rangle; Tr_{IS}^{IB} = \langle D_{IS}^{IB}, I_{IS}^{IB}, K_{IS}^{IB} \rangle; Tr_{IS}^S = \langle D_{IS}^S, I_{IS}^S, K_{IS}^S \rangle; \\
Tr_{IS}^f &= \langle D_{IS}^f, I_{IS}^f, K_{IS}^f \rangle; Tr_{IS}^{nf} = \langle D_{IS}^{nf}, I_{IS}^{nf}, K_{IS}^{nf} \rangle; \\
Tr_{IS}^{fw} &= \langle D_{IS}^{fw}, I_{IS}^{fw}, K_{IS}^{fw} \rangle; Tr_{IS}^{IB} = \langle D_{IS}^{nfw}, I_{IS}^{nfw}, K_{IS}^{nfw} \rangle.
\end{aligned} \tag{2.12}$$

В этих описаниях каждый из элементов кортежей представляет собой описание данных, информации и знаний, определяющих СТ к ИС.

Исходя из представлений групп СТ (2.12), представление множества СТ к ИС Tr_{IS} в универсуме требований будет иметь вид [129]:

$$\begin{aligned}
Tr_{IS} = \langle D_{IS}, I_{IS}, K_{IS} \rangle &= \langle D_{IS}^B, I_{IS}^B, K_{IS}^B \rangle \cup \langle D_{IS}^{IB}, I_{IS}^{IB}, K_{IS}^{IB} \rangle \cup \\
&\cup \langle D_{IS}^S, I_{IS}^S, K_{IS}^S \rangle \cup \langle D_{IS}^f, I_{IS}^f, K_{IS}^f \rangle \cup \langle D_{IS}^{nf}, I_{IS}^{nf}, K_{IS}^{nf} \rangle \cup \\
&\cup \langle D_{IS}^{fw}, I_{IS}^{fw}, K_{IS}^{fw} \rangle \cup \langle D_{IS}^{nfw}, I_{IS}^{nfw}, K_{IS}^{nfw} \rangle.
\end{aligned} \tag{2.13}$$

Что касается данных, информации и знаний, неизвестных одному или всем участникам проекта создания ИС, то они могут быть описаны следующим образом [129]:

$$\begin{aligned}
\langle UnD, UnI, UnK \rangle &= \langle UnD^{Sp}, UnI^{Sp}, UnK^{Sp} \rangle \cup \\
&\cup \langle UnD^{Cs}, UnI^{Cs}, UnK^{Cs} \rangle \cup \langle UnD^{Sp\&Cs}, UnI^{Sp\&Cs}, UnK^{Sp\&Cs} \rangle,
\end{aligned} \tag{2.14}$$

где $\langle UnD^{Sp}, UnI^{Sp}, UnK^{Sp} \rangle$ – множество данных, информации и знаний, неизвестных Поставщику;

$\langle UnD^{Cs}, UnI^{Cs}, UnK^{Cs} \rangle$ – множество данных, информации и знаний, неизвестных Потребителю;

$\langle UnD^{Sp\&Cs}, UnI^{Sp\&Cs}, UnK^{Sp\&Cs} \rangle$ – множество данных, информации и знаний, неизвестных Поставщику и Потребителю.

Множество методов добычи и обработки для универсума требований можно в этом случае разделить на следующие подмножества:

а) подмножество методов выявления и анализа требований Потребителем, имеющее в общем случае следующий вид:

$$\begin{aligned} (F_{Sp}^{Sp\&Cs}) : < UnD^{Sp\&Cs}, UnI^{Sp\&Cs}, UnK^{Sp\&Cs} > \rightarrow ; \\ & \rightarrow < UnD^{Sp}, UnI^{Sp}, UnK^{Sp} > \end{aligned} \quad (2.15)$$

б) подмножество методов выявления и анализа требований Поставщиком, имеющее в общем случае следующий вид [129]:

$$\begin{aligned} (F_{Cs}^{Sp\&Cs}) : < UnD^{Sp\&Cs}, UnI^{Sp\&Cs}, UnK^{Sp\&Cs} > \rightarrow ; \\ & \rightarrow < UnD^{Cs}, UnI^{Cs}, UnK^{Cs} > \end{aligned} \quad (2.16)$$

в) подмножество методов согласования требований, выявленных Потребителем и предъявленных Поставщику, в общем случае имеющее вид [129]:

$$(F^{Sp}) : < UnD^{Sp}, UnI^{Sp}, UnK^{Sp} > \rightarrow < D, I, K >; \quad (2.17)$$

г) подмножество методов согласования требований, выявленных Поставщиком и предъявленных Потребителю, в общем случае имеющее вид [129]:

$$(F^{Cs}) : < UnD^{Cs}, UnI^{Cs}, UnK^{Cs} > \rightarrow < D, I, K >. \quad (2.18)$$

Тогда универсум требований к ИС в общем случае будет иметь вид [129]:

$$\begin{aligned} U_{TrIS} = (< D_{IS}, I_{IS}, K_{IS} >, < UnD^{Sp}, UnI^{Sp}, UnK^{Sp} >, \\ < UnD^{Cs}, UnI^{Cs}, UnK^{Cs} >, < UnD^{Sp\&Cs}, UnI^{Sp\&Cs}, UnK^{Sp\&Cs} >, \\ (F_{Sp}^{Sp\&Cs}), (F_{Cs}^{Sp\&Cs}), (F^{Sp}), (F^{Cs})). \end{aligned} \quad (2.19)$$

Данное описание основано на иерархии DIKW (Data, Information, Knowledge and Wisdom), предложенной для управления знаниями [132]. Использование этой иерархии для описания процессов, непосредственно работающих с требованиями, а также процессов проектирования архитектуры обусловлено сложившимся в 1990-х гг. представлением [133], согласно которому никакое представление требований в одном виде не дает их полной картины. Сравнение представлений требований, полученных различными специалистами в ходе разнообразных исследований, помогает выявить несоответствия, неясности, допущения и упущения, которые трудно обнаружить, когда требования представлены в одном формате.

Следует отметить, что классическая модель DIKW предполагает четкую направленность преобразований данных в информацию, информации – в знания, а знаний – в мудрость. Однако в процессах, непосредственно работающих с требованиями, часто основой для моделирования требований к ИС является представление этих требований на уровне информации, выраженное в виде их текстовых описаний или же ВМ (см. рис. 2.4). При этом процесс проектирования архитектуры системы предполагает возможное уточнение ВМ требований описанием этих требований на уровне данных в виде той или иной атрибутивной модели [134, 135].

В то же время использование атрибутивных моделей, рассмотренных в Приложении Б, или аналогичных им, для представления требований к ИС на уровне данных затруднительно. Это затруднение вызвано, прежде всего, стремлением объединить в рамках одной атрибутивной модели требования к ИС наборы атрибутов, используемых в принципиально разных целях.

Поэтому возникает необходимость рассмотрения каждого требования к ИС как изначального многообразия представлений этого требования в виде данных, информации и знаний. Существование такого многообразия обусловлено следующими причинами [129]:

а) представление требований к ИС на уровне информации предназначено для описания различными способами элементов автоматизируемых объектов или процессов, элементов разрабатываемой ИС или ИС в целом с целью обеспечения возможности выполнения проекта создания ИС, к которой выдвинуто требование,

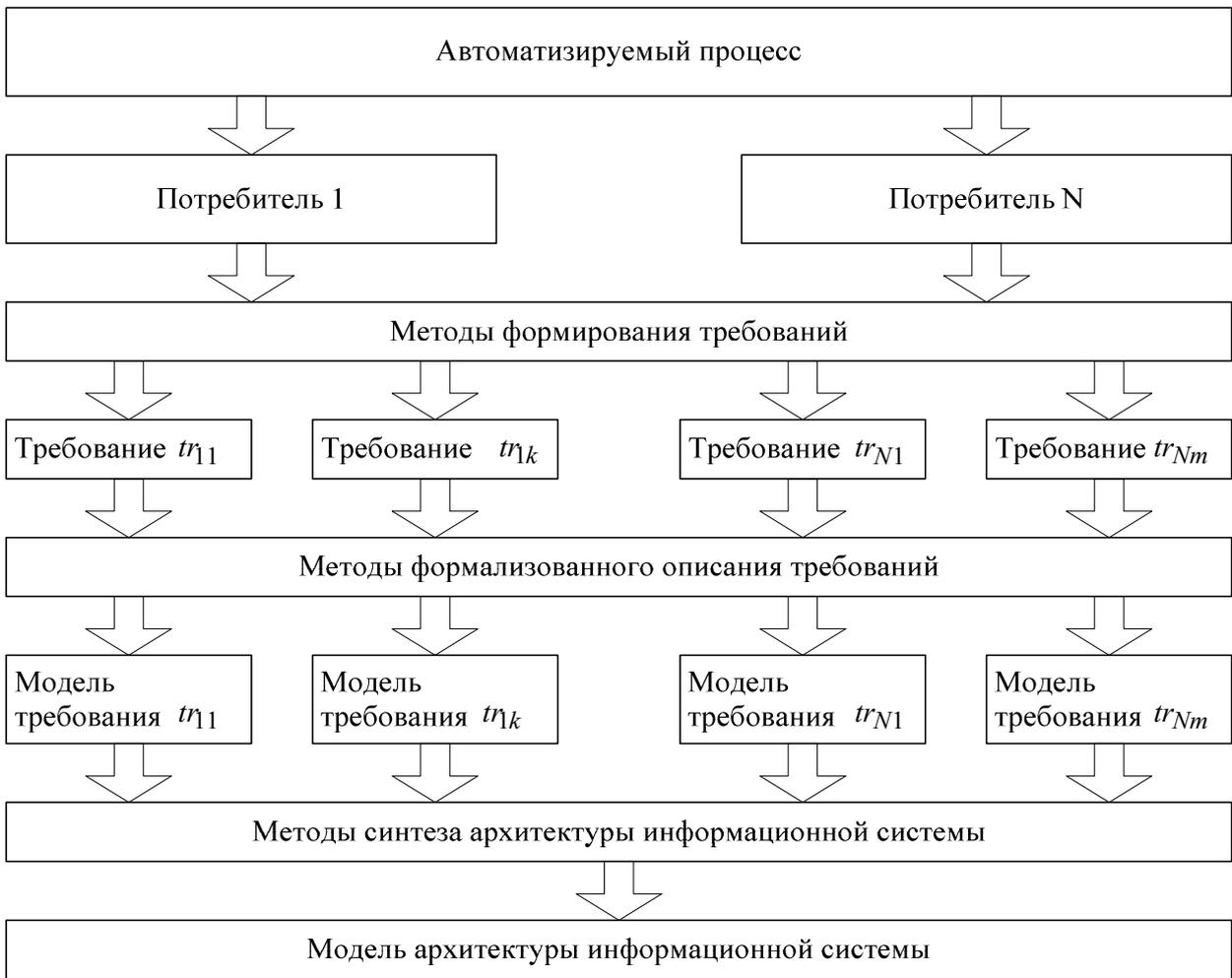


Рисунок 2.4 – Схема последовательности преобразований описаний требований к информационной системе при проектировании архитектуры системы

путем выполнения последовательности преобразований «неформализованное описание требования на естественной языке – частично формализованные описания требований – формальное описание требования в виде набора целевых показателей, значения которых характеризуют степень удовлетворения требования»;

б) представление требований на уровне данных предназначено для формирования описаний требования к ИС и его отдельных представлений, взаимно согласованных друг с другом, с целью обеспечения возможности осуществления стандартных операций ввода, отображения, редактирования и удаления требования к ИС и его отдельных представлений в рамках ИТ фор-

мирования и анализа требований к ИС, а также управления отдельными требованиями в ходе выполнения проекта создания ИС;

в) представление требования к ИС на уровне знаний предназначено для выявления знаний об автоматизируемых объектах или процессах, разрабатываемых элементах ИС или ИС в целом с целью обеспечения возможности реиспользования этих знаний в проектах создания других ИС.

Принятие рассмотрения каждого требования к ИС как изначального многообразия его представлений в качестве аксиомы формирования универсума требований (2.11) приводит к тому, что начальным описанием любого возможного требования к ИС, как известного, так и неизвестного, может быть любой из следующих вариантов [129]:

а) описание требования к ИС в виде данных (например, фрагмент атрибутивной модели требования к ИС, однозначно определяющей идентификаторы реиспользуемого представления требования к ИС или всего требования в целом);

б) описание требования к ИС в виде информации (например, результат интервьюирования Потребителя ИТ-услуг или же разработанная Поставщиком ИТ-услуг модель потоков данных);

в) описание требования к ИС в виде знаний (например, онтология ПрО, сформулированная Поставщиком на основе своего опыта автоматизации конкретного типа БП или же Потребителем на основе изучения своих БП).

Подобное представление универсума требований к ИС позволяет пересмотреть процесс проектирования архитектуры системы. Суть такого пересмотра заключается в том, что особенности моделей архитектуры системы (например, моделей ФС ИС) должны отражать особенности моделей автоматизируемых БП предприятия. Следовательно, и модели требований как исходные данные для построения модели архитектуры системы, и формулировки требований, выдвинутых Потребителями и принятых к исполнению Поставщиком, должны отражать особенности процессного подхода к описанию объекта управления, системы управления (а в нашем случае – еще и ИС как элемента этой системы управления).

С учетом предложенного в стандарте ISO/IEC 24774:2007 подхода к описанию процесса минимальная процессная атрибутивная модель требования к ИС должна включать в себя следующие атрибуты:

- а) название (наименование) процесса, к которому выдвигается требование;
- б) ожидаемые результаты выполнения процесса, к которому выдвигается требование (выходы требования);
- в) виды деятельности, выполняемые в рамках процесса, к которому выдвигается требование.

Опыт моделирования процессов [136–139] показывает, что в дополнение к указанным атрибутам минимальная процессная атрибутивная модель требования к ИС может также включать следующие атрибуты:

- а) цели, достижение которых свидетельствует о выполнении требования;
- б) ресурсы, обрабатываемые процессом, к которому выдвигается требование (входы требования);
- в) неизменяемые ресурсы, используемые процессом, к которому выдвигается требование (механизмы выполнения требования);
- г) перечень Потребителей и Поставщиков, которые выдвигают и принимают к исполнению требование (источники требования).

Указанное выше определяет концепцию представления требований к ИС как набор следующих положений [129, 134, 140]:

- а) отказ от рассмотрения только множества СТ к ИС и изначальное представление требований к ИС как элементов универсума, включающего в себя как известные, так и неизвестные Поставщику, Потребителю или им обоим требования к ИС, а также методы формирования этих требований;
- б) изначальное многообразие представлений требований к ИС в виде данных, информации и знаний;
- в) процессный подход к описанию требований, определяющий минимальную процессную атрибутивную модель требования к ИС;
- г) подход к управлению требованиями к ИС, основанный на изложенном в п. 1.2.5 основном принципе управления требованиями.

Изложенная концепция определяет основные особенности формализованного описания групп требований к ИС и отдельных требований, их моделей различного вида, методов и средств формирования, анализа и управления этими требованиями, а также возможных моделей и методов синтеза описания проектируемой архитектуры ИС на различных уровнях представления.

2.4 Формализованное описание универсума требований к информационной системе

Сформулированные в подразд. 2.3 основные положения концепции представления требований к ИС определяют общие формализованные описания требований. Учитывая положения а) и б) данной концепции, общие формализованные описания требований к ИС должны включать в себя:

- а) формализованные описания требований к ИС (как известных, так и неизвестных Поставщику, Потребителю или им обоим);
- б) формализованные описания методов формирования требований к ИС;
- в) формализованные описания представлений требований к ИС на уровнях данных, информации и знаний.

Кроме того, с учетом предложенного в подразд. 2.3 представления универсума требований к ИС в виде выражения (2.19), общие формализованные описания требований к ИС дополнительно должны включать в себя:

- а) формализованные описания методов, приемов и способов преобразования представлений различных требований, выполненных на одном и том же уровне, самих в себя и друг в друга, что позволит установить и отображать взаимосвязи между отдельными требованиями группы, обусловленные особенностями автоматизируемого процесса;
- б) формализованные описания методов, приемов и способов преобразования представлений требований к ИС, выполненных на одном уровне, в

представления требований, выполненных на другом уровне.

Рассмотренные условия создания формализованных описаний требований к ИС заставляют отказаться от классических теоретико-множественных описаний требований. Математический аппарат, на основе которого следует разрабатывать формализованные описания групп требований к ИС, должен позволять:

- рассматривать требования к ИС как единую целостную систему – прообраз разрабатываемой ИС – и одновременно как множества связанных друг с другом элементов, преобразуемые одно в другое в соответствии с методологией и технологиями формирования и анализа требований;

- учитывать влияние на представления требований к ИС конкретных методов, приемов и способов формирования и преобразования представлений этих требований.

Данным условиям лучше всего отвечает математический аппарат теории категорий [141-144].

Рассмотрим категорные модели элементов универсума требований. Так модель множества данных, информации и знаний, неизвестных Поставщику и Потребителю $L_{Pr\&U}$, будет иметь вид:

$$\begin{aligned}
 L_{Pr\&U} = [& UnD^{Pr\&U}, UnI^{Pr\&U}, UnK^{Pr\&U}, H(UnD^{Pr\&U}), H(UnI^{Pr\&U}), \\
 & H(UnK^{Pr\&U}), H(UnD^{Pr\&U}, UnI^{Pr\&U}), H(UnI^{Pr\&U}, UnD^{Pr\&U}), \\
 & H(UnD^{Pr\&U}, UnK^{Pr\&U}), H(UnK^{Pr\&U}, UnD^{Pr\&U}), H(UnI^{Pr\&U}, \\
 & UnK^{Pr\&U}), H(UnK^{Pr\&U}, UnI^{Pr\&U})], \quad (2.20)
 \end{aligned}$$

где $UnD^{Pr\&U}$ – подкласс представлений требований, неизвестных Поставщику и Потребителю, в виде данных;

$UnI^{Pr\&U}$ – подкласс представлений требований, неизвестных Поставщику и Потребителю, в виде информации;

$UnK^{Pr\&U}$ – подкласс представлений требований, неизвестных Поставщику и Потребителю, в виде знаний;

$H(UnD^{Pr\&U})$ – подмножество морфизмов, описывающих преобразования представлений требований, неизвестных Поставщику и Потребителю, в виде данных самих в себя и друг в друга, имеющее следующий вид:

$$\forall X, Y \in UnD^{Pr\&U} \quad H(UnD^{Pr\&U}) = H(1_X) \cup H(X, Y) \cup H(1_Y); \quad (2.21)$$

$1_X, 1_Y$ – единичные морфизмы, условия существования которых описаны в [141];

$H(UnI^{Pr\&U})$ – подмножество морфизмов, описывающих преобразования представлений требований, неизвестных Поставщику и Потребителю, в виде информации самих в себя и друг в друга, имеющее следующий вид:

$$\forall X, Y \in UnI^{Pr\&U} \quad H(UnI^{Pr\&U}) = H(1_X) \cup H(X, Y) \cup H(1_Y); \quad (2.22)$$

$H(UnK^{Pr\&U})$ – подмножество морфизмов, описывающих преобразования представлений требований, неизвестных Поставщику и Потребителю, в виде знаний самих в себя и друг в друга, имеющее следующий вид:

$$\forall X, Y \in UnK^{Pr\&U} \quad H(UnK^{Pr\&U}) = H(1_X) \cup H(X, Y) \cup H(1_Y); \quad (2.23)$$

$H(UnD^{Pr\&U}, UnI^{Pr\&U})$ – подмножество морфизмов, описывающих методы, приемы и способы преобразования представлений требований, неизвестных Поставщику и Потребителю, в виде данных в представления требований, неизвестных Поставщику и Потребителю, в виде информации;

$H(UnI^{Pr\&U}, UnD^{Pr\&U})$ – подмножество морфизмов, описывающих методы, приемы и способы преобразования представлений требований, неизвестных Поставщику и Потребителю, в виде информации в представления требований, неизвестных Поставщику и Потребителю, в виде данных;

$H(UnD^{Pr\&U}, UnK^{Pr\&U})$ – подмножество морфизмов, описывающих методы, приемы и способы преобразования представлений требований, неизвестных Поставщику и Потребителю, в виде данных в представления требований, неизвестных Поставщику и Потребителю, в виде знаний;

$H(UnK^{Pr\&U}, UnD^{Pr\&U})$ – подмножество морфизмов, описывающих методы, приемы и способы преобразования представлений требований, неизвестных Поставщику и Потребителю, в виде знаний в представления требований, неизвестных Поставщику и Потребителю, в виде данных;

$H(UnI^{Pr\&U}, UnK^{Pr\&U})$ – подмножество морфизмов, описывающих методы, приемы и способы преобразования представлений требований, неизвестных Поставщику и Потребителю, в виде информации в представления требований, неизвестных Поставщику и Потребителю, в виде знаний;

$H(UnK^{Pr\&U}, UnI^{Pr\&U})$ – подмножество морфизмов, описывающих методы, приемы и способы преобразования представлений требований, неизвестных Поставщику и Потребителю, в виде знаний в представления требований, неизвестных Поставщику и Потребителю, в виде информации.

Категорные модели, описывающие кортежи $\langle UnD^{Pr}, UnI^{Pr}, UnK^{Pr} \rangle$ и $\langle UnD^U, UnI^U, UnK^U \rangle$, имеют аналогичный вид и представлены в Приложении В (выражения (В.1) и (В.2)).

Для описания множества СТ к ИС (2.2), представленного в универсуме требований к ИС в виде выражения (2.13), предлагается каждую группу требований рассматривать как отдельную категорию. Так модель сформулированных бизнес-требований к ИС L_{IS}^B будет иметь следующий вид:

$$L_{IS}^B = [D_{IS}^B, I_{IS}^B, K_{IS}^B, H(D_{IS}^B), H(I_{IS}^B), H(K_{IS}^B), H(D_{IS}^B, I_{IS}^B), H(I_{IS}^B, D_{IS}^B), H(D_{IS}^B, K_{IS}^B), H(K_{IS}^B, D_{IS}^B), H(I_{IS}^B, K_{IS}^B), H(K_{IS}^B, I_{IS}^B)], \quad (2.24)$$

где D_{IS}^B – подкласс представлений сформулированных бизнес-требований к ИС в виде данных;

I_{IS}^B – подкласс представлений сформулированных бизнес-требований к ИС в виде информации;

K_{IS}^B – подкласс представлений сформулированных бизнес-требований к ИС в виде знаний;

$H(D_{IS}^B)$ – подмножество морфизмов, описывающих преобразования представлений сформулированных бизнес-требований к ИС в виде данных самих в себя и друг в друга, имеющее следующий вид:

$$\forall X, Y \in D_{IS}^B \quad H(D_{IS}^B) = H(1_X) \cup H(X, Y) \cup H(1_Y); \quad (2.25)$$

$1_X, 1_Y$ – единичные морфизмы, условия существования которых описаны в [141];

$H(I_{IS}^B)$ – подмножество морфизмов, описывающих преобразования представлений сформулированных бизнес-требований к ИС в виде информации самих в себя и друг в друга, имеющее следующий вид:

$$\forall X, Y \in I_{IS}^B \quad H(I_{IS}^B) = H(1_X) \cup H(X, Y) \cup H(1_Y); \quad (2.26)$$

$H(K_{IS}^B)$ – подмножество морфизмов, описывающих преобразования представлений сформулированных бизнес-требований к ИС в виде знаний самих в себя и друг в друга, имеющее следующий вид:

$$\forall X, Y \in K_{IS}^B \quad H(K_{IS}^B) = H(1_X) \cup H(X, Y) \cup H(1_Y); \quad (2.27)$$

$H(D_{IS}^B, I_{IS}^B)$ – подмножество морфизмов, описывающих методы, приемы и способы преобразования представлений сформулированных бизнес-требований к ИС в виде данных в представления сформулированных бизнес-требований к ИС в виде информации;

$H(I_{IS}^B, D_{IS}^B)$ – подмножество морфизмов, описывающих методы, приемы и способы преобразования представлений сформулированных бизнес-требований к ИС в виде информации в представления сформулированных бизнес-требований к ИС в виде данных;

$H(D_{IS}^B, K_{IS}^B)$ – подмножество морфизмов, описывающих методы, приемы и способы преобразования представлений сформулированных бизнес-требований к ИС в виде данных в представления сформулированных бизнес-требований к ИС в виде знаний;

$H(K_{IS}^B, D_{IS}^B)$ – подмножество морфизмов, описывающих методы, приемы и способы преобразования представлений сформулированных бизнес-требований к ИС в виде знаний в представления сформулированных бизнес-требований к ИС в виде данных;

$H(I_{IS}^B, K_{IS}^B)$ – подмножество морфизмов, описывающих методы, приемы и способы преобразования представлений сформулированных бизнес-требований к ИС в виде информации в представления

сформулированных бизнес-требований к ИС в виде знаний;

$H(K_{IS}^B, I_{IS}^B)$ – подмножество морфизмов, описывающих методы, приемы и способы преобразования представлений сформулированных бизнес-требований к ИС в виде знаний в представления сформулированных бизнес-требований к ИС в виде информации.

Рассмотренные особенности модели (2.24) определяют особенности моделей, описывающих остальные группы требований к ИС и представленных в Приложении В (выражения (В.3) – (В.8)).

Основываясь на моделях (2.24), (В.3)-(В.8), можно представить формализованное описание множества СТ к ИС (2.13) моделью $M_{Tr_{IS}}$, имеющей следующий вид:

$$M_{Tr_{IS}} = [L_{IS}^B, L_{IS}^{IB}, L_{IS}^S, L_{IS}^f, L_{IS}^{nf}, L_{IS}^{fw}, L_{IS}^{nfw}, \Phi_{IB}^B, \Phi_s^{IB}, \Phi_f^S, \Phi_{nf}^S, \Phi_{nf}^f, \Phi_{fw}^f, \Phi_{nfw}^{nf}, \Phi_{nfw}^{fw}], \quad (2.29)$$

где Φ_{IB}^B – одноместный ковариантный функтор (ОКФ) [114], устанавливающий связь между категорией L_{IS}^B (начало функтора) и категорией L_{IS}^{IB} (конец функтора);

Φ_s^{IB} – ОКФ, устанавливающий связь между категорией L_{IS}^{IB} (начало функтора) и категорией L_{IS}^S (конец функтора);

Φ_f^S – ОКФ, устанавливающий связь между категорией L_{IS}^S (начало функтора) и категорией L_{IS}^f (конец функтора);

Φ_{nf}^S – ОКФ, устанавливающий связь между категорией L_{IS}^S (начало функтора) и категорией L_{IS}^{nf} (конец функтора);

Φ_{nf}^f – ОКФ, устанавливающий связь между категорией L_{IS}^f (начало функтора)

и категорией L_{IS}^{nf} (конец функтора);

Φ_{fw}^f – ОКФ, устанавливающий связь между категорией L_{IS}^f (начало функтора) и категорией L_{IS}^{fw} (конец функтора);

Φ_{nfw}^{nf} – ОКФ, устанавливающий связь между категорией L_{IS}^{nf} (начало функтора) и категорией L_{IS}^{nfw} (конец функтора);

Φ_{nfw}^{fw} – ОКФ, устанавливающий связь между категорией L_{IS}^{fw} (начало функтора) и категорией L_{IS}^{nfw} (конец функтора).

Тогда весь универсум требований к ИС может быть формализованно описан моделью $M_{U_{Tr}}$, имеющей следующий вид [145]:

$$M_{U_{Tr}} = [L_{Tr_{IS}}, L_{Pr}, L_U, L_{Pr\&U}, \Phi_{Pr}^{Pr\&U}, \Phi_U^{Pr\&U}, \Phi_{Tr_{IS}}^{Pr}, \Phi_{Tr_{IS}}^U], \quad (2.30)$$

где $\Phi_{Pr}^{Pr\&U}$ – ОКФ, устанавливающий связь между категорией $L_{Pr\&U}$ (начало функтора) и категорией L_U (конец функтора);

$\Phi_U^{Pr\&U}$ – ОКФ, устанавливающий связь между категорией $L_{Pr\&U}$ (начало функтора) и категорией L_U (конец функтора);

$\Phi_{Tr_{IS}}^{Pr}$ – ОКФ, устанавливающий связь между категорией L_{Pr} (начало функтора) и категорией $L_{Tr_{IS}}$ (конец функтора);

$\Phi_{Tr_{IS}}^U$ – ОКФ, устанавливающий связь между категорией L_U (начало функтора) и категорией $L_{Tr_{IS}}$ (конец функтора).

Ковариантность функторов, присутствующих в моделях (2.29) и (2.30), призвана обеспечить в категории-конец функтора сохранение направленности морфизмов, связывающих объекты категории-начала функтора [141]. Данное

свойство функторов с прикладной точки зрения обеспечивает сохранность особенностей связей между отдельными требованиями в группе требований к ИС, обусловленными особенностями автоматизируемого процесса.

С прикладной точки зрения, модель универсума требований (2.30) является формализованным описанием множества допустимых для реализации АФ, в котором рассматриваются возможные модели и методы формирования и анализа требований к ИС. Под словами «допустимые для реализации» следует понимать то множество АФ, которое может быть разработано на основе рассмотренной выше концепции представления требований к ИС. Тогда модель сформулированных требований (2.29), с прикладной точки зрения является формализованным описанием набора допустимых для реализации ИТ формирования и анализа требований к ИС. В соответствии с приведенным в подразд. 1.1 определением понятия «ИТ», данная модель содержит выполненные в виде функторов описания методов, приемов и способов применения КСА или его отдельных элементов, используемых для преобразования представлений сформулированных групп требований к ИС в группы ФТ и нефункциональных требований к ИТ-услугам или в группы ФТ и нефункциональных требований к ИТ-сервисам, реализующим эти ИТ-услуги. Формирование в рамках данной модели групп ФТ и нефункциональных требований к ИТ-услугам предоставит исходную информацию для выполнения процесса проектирования архитектуры ИС.

2.5 Обобщенное формализованное описание одноместного ковариантного функтора

Использование функторов в моделях множества СТ (2.29) и универсума требований (2.30) ставит вопрос о возможности разработки единого подхода к их формализованному описанию. Такой подход важен не только с теоретической, но и с прикладной точки зрения, поскольку его существование в теории ставит

перед специалистами проблему стандартизации и унификации методов, приемов и способов применения КСА или его отдельных элементов, используемых для преобразования представлений сформулированных групп требований к ИС в группы ФТ и нефункциональных требований к ИТ-услугам или в группы ФТ и нефункциональных требований к ИТ-сервисам, реализующим эти ИТ-услуги.

Для разработки общего формализованного описания ОКФ, связывающих две модели в рамках моделей (2.29) и (2.30), обозначим исходную модель как категорию A , а конечную категорную модель – как категорию B . Тогда отображение, переводящее исходную категорию A в конечную категорию B , можно будет рассматривать в виде обобщенного ОКФ Φ_B^A – частного случая отображения категории A в категорию B при выполнении условий [141]:

$$\forall a \in Ob^A \quad \exists \Phi_B^A(a) \in Ob^B ; \quad (2.31)$$

$$\forall \alpha \in H_A(a_i, a_j) \subseteq Mor^A \quad \exists \Phi_B^A(\alpha) \in H_B(\Phi_B^A(a_i), \Phi_B^A(a_j)) \in Mor^B ; \quad (2.32)$$

$$\forall 1_a \in Mor^A \quad \exists \Phi_B^A(1_a) = 1_{\Phi_B^A(a)} \in Mor^B ; \quad (2.33)$$

$$\begin{aligned} \forall \alpha \in H_A(a_i, a_j) \in Mor^A, \beta \in H_A(a_j, a_k) \in Mor^A \\ \exists \Phi_B^A(\alpha\beta) = \Phi_B^A(\alpha)\Phi_B^A(\beta) \in Mor^B, \end{aligned} \quad (2.34)$$

где Ob^A – класс объектов исходной категории A , описывающих элементы класса объектов исходной категорной модели;

a – любой объект, относящийся к классу объектов Ob^A исходной категории A ;

Ob^B – класс объектов конечной категории B , описывающих элементы класса объектов конечной категорной модели;

Mor^A – класс морфизмов исходной категории A , описывающих связи эле-

ментов класса объектов исходной категорной модели;

$H_A(a_i, a_j)$ – множество морфизмов, определенных в исходной категории A для объекта a_i как начала морфизма и объекта a_j как конца морфизма, при этом $i \neq j$ являются идентификаторами объектов исходной категории A ;

α – морфизм, являющийся элементом множества $H_A(a_i, a_j)$;

Mor^B – класс морфизмов конечной категории B , описывающих связи элементов класса объектов конечной категорной модели;

$H_B(\Phi_B^A(a_i), \Phi_B^A(a_j))$ – множество морфизмов, определенных в конечной категории B для объекта $\Phi_B^A(a_i)$ как начала морфизма и объекта $\Phi_B^A(a_j)$ как конца морфизма;

1_a – единичный морфизм, определенный для объекта a исходной категории A ;

$1_{\Phi_B^A(a)}$ – единичный морфизм, определенный для соответствующего объекта конечной категории B ;

$\alpha\beta$ – частичная бинарная операция умножения морфизмов исходной категории A .

Условия (2.31)-(2.34) позволяют утверждать, что определение обобщенного

ОКФ Φ_B^A целесообразно разделить на две связанные между собой задачи [146]:

- определение системы правил трансформации объектов исходной категории A в объекты конечной категории B ;
- определение системы правил трансформации морфизмов исходной категории A в морфизмы конечной категории B .

Эти задачи в работе [147] решены для обобщенного ОКФ, описывающего процесс проектирования распределенных БД ИС, причем решение приведено в виде обобщенного алгоритма. Однако этот алгоритм сформулирован с учетом единой природы категорных представлений ИС и ее БД. В общем же случае для проектирования ИС одновременно могут использоваться описания требований, в основе которых находятся как структурный, так и объектно-ориентированный подходы.

Описать функтор – это значит:

- описать классы объектов и морфизмов исходной категории A ;
- описать классы объектов и морфизмов конечной категории B ;
- описать совокупность правил отображения объектов исходной категории A в объекты конечной категории B , удовлетворяющих условию (2.31);
- описать совокупность правил отображения морфизмов исходной категории A в морфизмы конечной категории B , удовлетворяющих условиям (2.32)-(2.34).

Тогда, в общем случае, любой ОКФ можно описать выражением вида [146, 148]

$$\Phi_B^A = (Ob^A, Ob^B, Mor^A, Mor^B, \Phi_{Ob^A}^{Ob^B}, \Phi_{Mor^A}^{Mor^B}), \quad (2.35)$$

где $\Phi_{Ob^A}^{Ob^B}$ – система правил трансформации элементов класса объектов исходной

категории A в элементы класса объектов конечной категории B ;

$\Phi_{Mor^A}^{Mor^B}$ – система правил трансформации элементов класса морфизмов исходной

категории A в элементы класса морфизмов конечной категории B .

Подобное формализованное описание ОКФ определяет следующее условие его реализуемости [146]:

$$Ob^A \subseteq \Phi_{Ob^B}^{Ob^A}(Ob^A) \subseteq Ob^B, \quad Mor^A \subseteq \Phi_{Mor^B}^{Mor^A}(Mor^A) \subseteq Mor^B. \quad (2.36)$$

Данное условие для МТ к ИС можно интерпретировать следующим образом: фрагмент тезауруса, образующий исходную модель, не должен превышать по размерам и сложности фрагмент тезауруса, образующий конечную модель. Иными словами, при формулировании требований к ИС все многообразие синтаксических конструкций и возможных синонимов, используемых Потребителем для описания некоего понятия ПрО, должно сводиться Поставщиком к единому понятному для него представлению этого понятия в

виде данных, информации или знаний. При этом размеры указанных фрагментов тезауруса определяются количеством типов объектов соответствующих категорий. Сложность фрагментов тезауруса определяется количеством типов морфизмов соответствующих категорий.

Подобная интерпретация будет справедливой только для тех случаев, когда уровни представления требований к ИС исходной и конечной категорий будут адекватны друг другу. На практике чаще всего встречается иная ситуация. Так от моделей, описывающих БП предприятия, разработчики переходят к моделям, описывающим ИС управления этим предприятием, а от них, в свою очередь, – к моделям, описывающим отдельные виды обеспечений ИС [136, 137]. Возможны и обратные переходы – например, в ходе реверс-инжиниринга (обратного проектирования) БД ИС [149]. Поэтому существование систем правил $\Phi_{Ob^A}^{Ob^B}$ и $\Phi_{Mor^A}^{Mor^B}$ возможно только при выполнении одного из переходов, показанных коммутативными диаграммами

$$\begin{array}{ccc} Tez(Ob^A) \rightarrow Tez(Ob^B) & ; & Tez(Mor^A) \rightarrow Tez(Mor^B) \\ \downarrow & & \downarrow \\ MTez(Ob^A) \rightarrow MTez(Ob^B) & ; & MTez(Mor^A) \rightarrow MTez(Mor^B) \end{array} ; \quad (2.37)$$

$$\begin{array}{ccc} mTez(Ob^A) \rightarrow mTez(Ob^B) & ; & mTez(Mor^A) \rightarrow mTez(Mor^B) \\ \downarrow & & \downarrow \\ Tez(Ob^A) \rightarrow Tez(Ob^B) & ; & Tez(Mor^A) \rightarrow Tez(Mor^B) \end{array} , \quad (2.38)$$

где $Tez(\bullet)$ – текущий уровень представления фрагмента тезауруса, используемого моделью;

$MTez(\bullet)$ – макроуровень представления фрагмента тезауруса, используемого моделью, для которого $Tez(\bullet)$ является частным случаем;

$mTez(\bullet)$ – микроуровень представления фрагмента тезауруса, используемого моделью, который является частным случаем $Tez(\bullet)$.

Тогда ОКФ (2.35) можно представить следующим образом [146, 148]:

$$\Phi_B^A = (Sc_{Tez(A)}, Ob^A, Mor^A, Sc_{Tez(B)}, Ob^B, Mor^B, \Phi_{Ob^A}^{Ob^B}, \Phi_{Mor^A}^{Mor^B}), \quad (2.39)$$

где $Sc_{Tez(A)}$ – уровень представления фрагмента тезауруса, используемого исходной моделью требований к ИС, которая описана категорией A ;

$Sc_{Tez(B)}$ – уровень представления тезауруса, используемого итоговой моделью требований к ИС, которая описана категорией B .

Выражение (2.39) позволяет сделать вывод о существовании двух основных способов реализации предлагаемого формализованного описания ОКФ. Первый способ предполагает поиск таких систем правил, которые однозначно определяли бы соответствие заранее заданных элементов исходной и конечной категорий. Второй способ предполагает формирование таких объектов и морфизмов категории B , которые удовлетворяли бы заранее заданным системам правил трансформации, объектам и морфизмам исходной категории A , а также учитывали бы разность уровней представления тезаурусов моделей $Sc_{Tez(A)}$ и $Sc_{Tez(B)}$.

С прикладной точки зрения, первый способ реализации ОКФ предполагает формирование и реализацию некоего общего алгоритма или базы правил преобразования исходного представления требования к ИС в конечное представление этого же требования или множества других требований. При этом предполагается, что текстовые конструкции или элементы ВМ, описывающие эти требования, заранее известны алгоритму или базе правил и практически не меняются с течением времени. Поэтому данный способ наиболее подходит для таких описаний, которые сводят к минимуму количество элементов алфавита языка описания или моделирования требований к ИС. К подобным описаниям можно отнести большинство распространенных ВМ ИС и ее компонентов (диаграммы потоков данных, диаграммы «сущность – связь», ДК и т.п.).

Второй способ реализации ОКФ предполагает создание и постоянное раз-

витие некоего множества алгоритмов или некоей системы правил преобразования исходного представления требования к ИС в конечное представление этого же требования или множества других требований. При этом темпы изменения этого множества алгоритмов или систем правил определяются темпами появления новых или изменения существующих понятий тезауруса исходного представления требования к ИС. Появление каждого нового, ранее неизвестного понятия в таком представлении требования к ИС (в виде данных, информации или знаний) требует дополнения существующего варианта реализации ОКФ новыми алгоритмами или правилами, устанавливающими способ и форму описания такого нового понятия в конечном представлении этого требования.

2.6 Модель паттерна проектирования требований к информационной системе

Как следует из моделей сформулированных требований к ИС (2.29) и универсума требований к ИС (2.30), процессы разработки ИС, непосредственно работающие с требованиями, могут рассматриваться как последовательные преобразования представлений требований в виде данных, информации и знаний в ходе выявления и формулирования требований к ИС в соответствии с выбранной методологией разработки ИС. Такие преобразования в предложенных моделях показаны в виде различных вариантов ОКФ (2.39). Однако эта модель оставляет открытым вопрос о виде представлений отдельных требований к ИС, методах и способах формирования этих представлений и преобразований этих представлений самих в себя и друг в друга.

В общем случае формализованные описания представлений требований к ИС определяются сформулированными в подразд. 2.4 основными положениями концепции представления требований к ИС. Однако кроме указанных положений, на подход к формализованным описаниям представлений требований к ИС

в значительной степени влияет стремление Поставщика ре-использовать требования, сформулированные в ходе создания конкретной ИС, в последующих проектах аналогичных ИС. Подобное стремление приводит к необходимости рассматривать требования к конкретной ИС как частные случаи неких общих шаблонов, описывающих автоматизируемый процесс и разрабатываемую ИС на более высоких уровнях абстракции.

Наиболее близкой к такому взгляду на требования к ИС является идея паттернов проектирования как шаблонов, определяющих решение отдельных задач, часто повторяющихся в различных проектах программных систем [150]. Результаты анализа опыта применения паттернов проектирования, рассмотренные в [151], позволяют сделать следующие выводы:

а) паттерны выгодны в случае, если создаваемая программная система должна быть гибкой и предполагает постоянные изменения своей структуры и содержания (добавление или изменение функций и т.п.);

б) паттерны выгодны в случае проведения рефакторинга или других процедур, направленных на совершенствование организации программной системы или ее отдельных элементов [152-155];

в) паттерны невыгодны в случаях, когда затраты ресурсов на создание и сопровождение программной системы, обусловленные сложностью этой системы, превышают эффект от гибкости этой системы.

Основываясь на этих выводах, можно утверждать, что современные паттерны являются естественным обобщением результатов процесса проектирования программных систем. При этом применение паттернов проектирования должно являться не самоцелью, а необходимостью, вызванной особенностями конкретного проекта (или портфеля проектов) [151]. Данная точка зрения позволяет, в свою очередь, предположить, что идея использования паттернов в процессах проектирования программных систем является стремлением формализовать и использовать с прикладными целями знания разработчиков таких систем о методах, способах и стилях программирования тех или иных проблем практического характера. Такое предположение хорошо иллюстриру-

ется примером паттернов GRASP (General Responsibility Assignment Software Patterns – общие образцы распределения обязанностей) которые используются в объектно-ориентированном проектировании для решения общих задач по назначению обязанностей классам и объектам. Можно сказать, что GRASP-паттерны – это хорошо документированные, стандартизированные и проверенные временем принципы объектно-ориентированного анализа и проектирования, а не попытка привнести что-то принципиально новое [156].

Однако рассмотренная концепция паттернов проектирования ориентирована только на разработку ПО систем. Использование этой концепции в других процессах ЖЦ систем (согласно ISO/IEC 15288:2002) без серьезных изменений затруднительно. Особенно это касается процессов разработки ИС как разновидности программных систем. Так, в соответствии с приведенным в подразд. 1.1 описанием главной цели деятельности ИС, одной из основных проблем разработки ИС является проблема отображения желаемого для Потребителя единого целостного информационного представления объекта или процесса как на уровне ИТ-услуг (ФС ИС), так и на уровне отдельных ИТ-сервисов (обеспечивающая часть ИС). Для решения данной проблемы предлагается из всего множества паттернов проектирования в соответствии с предложенной категорной моделью универсума требований к ИС (2.30) выделить такую разновидность паттернов, как ППТ к ИС, используя следующее определение.

ППТ к ИС – это результат выделения и ре-использования с прикладными целями Поставщиком ИТ-услуг следующих видов знаний [157-159]:

а) об условии или возможности, которые необходимы Потребителю ИТ-услуг для решения стоящей перед ним проблемы или достижения поставленной перед ним цели;

б) об условии или возможности, которой должна обладать ИС или компонент ИС (ИТ-услуга, ИТ-сервис) с точки зрения Поставщика или Потребителя ИТ-услуг, соответствующих договору, стандарту, спецификации или другому официальному документу;

в) о документированном (с использованием естественного или фор-

мального языка) представлении условия или возможности, подобно описанным в первых двух определениях.

В соответствии с предлагаемым определением, ППТ к ИС можно классифицировать следующим образом [158, 159]:

а) по назначению – структурные и поведенческие паттерны;

б) по уровню применения – паттерны представления требований к ИС в виде знаний, паттерны представления требований к ИС в виде информации, паттерны представления требований к ИС в виде данных.

Предложенное определение ППТ к ИС позволяет утверждать, что рассмотренные выше модели (2.24), (В.3)-(В.8) являются частными случаями математической модели паттерна проектирования сформулированных требований к ИС M_{IS}^{Pt} , которая имеет следующий вид [159]:

$$M_{IS}^{Pt} = [D_{IS}^{Pt}, I_{IS}^{Pt}, K_{IS}^{Pt}, H(D_{IS}^{Pt}), H(I_{IS}^{Pt}), H(K_{IS}^{Pt}), H(D_{IS}^{Pt}, I_{IS}^{Pt}), H(I_{IS}^{Pt}, D_{IS}^{Pt}), H(D_{IS}^{Pt}, K_{IS}^{Pt}), H(K_{IS}^{Pt}, D_{IS}^{Pt}), H(I_{IS}^{Pt}, K_{IS}^{Pt}), H(K_{IS}^{Pt}, I_{IS}^{Pt})], \quad (2.40)$$

где D_{IS}^{Pt} – подкласс структурных и поведенческих паттернов представлений требований к ИС в виде данных;

I_{IS}^{Pt} – подкласс структурных и поведенческих паттернов представлений требований к ИС в виде информации;

K_{IS}^{Pt} – подкласс структурных и поведенческих паттернов представлений требований к ИС в виде знаний;

$H(D_{IS}^{Pt})$ – подмножество морфизмов, описывающих преобразования структурных и поведенческих паттернов представлений требований к ИС в виде данных самих в себя и друг в друга, имеющее следующий вид:

$$\forall X, Y \in D_{IS}^{Pt} \quad H(D_{IS}^{Pt}) = H(1_X) \cup H(X, Y) \cup H(1_Y); \quad (2.41)$$

$1_X, 1_Y$ – единичные морфизмы, условия существования которых описаны в [141];
 $H(I_{IS}^{Pt})$ – подмножество морфизмов, описывающих преобразования структурных и поведенческих паттернов представлений требований к ИС в виде информации самих в себя и друг в друга, имеющее следующий вид:

$$\forall X, Y \in I_{IS}^{Pt} \quad H(I_{IS}^{Pt}) = H(1_X) \cup H(X, Y) \cup H(1_Y); \quad (2.42)$$

$H(K_{IS}^{Pt})$ – подмножество морфизмов, описывающих преобразования структурных и поведенческих паттернов представлений требований к ИС в виде знаний самих в себя и друг в друга, имеющее следующий вид:

$$\forall X, Y \in K_{IS}^{Pt} \quad H(K_{IS}^{Pt}) = H(1_X) \cup H(X, Y) \cup H(1_Y); \quad (2.43)$$

$H(D_{IS}^{Pt}, I_{IS}^{Pt})$ – подмножество морфизмов, описывающих преобразования структурных и поведенческих паттернов представлений требований к ИС в виде данных в структурные и поведенческие паттерны представлений требований к ИС в виде информации;

$H(I_{IS}^{Pt}, D_{IS}^{Pt})$ – подмножество морфизмов, описывающих преобразования структурных и поведенческих паттернов представлений требований к ИС в виде информации в структурные и поведенческие паттерны представлений требований к ИС в виде данных;

$H(D_{IS}^{Pt}, K_{IS}^{Pt})$ – подмножество морфизмов, описывающих преобразования структурных и поведенческих паттернов представлений требований к ИС в виде данных в структурные и поведенческие паттерны представлений требований к ИС в виде знаний;

$H(K_{IS}^{Pt}, D_{IS}^{Pt})$ – подмножество морфизмов, описывающих преобразования структурных и поведенческих паттернов представлений тре-

бований к ИС в виде знаний в структурные и поведенческие паттерны представлений требований к ИС в виде данных;

$H(I_{IS}^{Pt}, K_{IS}^{Pt})$ – подмножество морфизмов, описывающих преобразования представлений структурных и поведенческих паттернов требований к ИС в виде информации в структурные и поведенческие паттерны представлений требований к ИС в виде знаний;

$H(K_{IS}^{Pt}, I_{IS}^{Pt})$ – подмножество морфизмов, описывающих преобразования структурных и поведенческих паттернов представлений требований к ИС в виде знаний в структурные и поведенческие паттерны представлений требований к ИС в виде информации.

По аналогии с понятием фактор-объекта категории [141], модель L_{IS}^{Pt} будет являться фактор-категорией для рассмотренных ранее моделей (2.20), (B.1), (B.2), (2.24) и (B.3)-(B.8). Это означает, что данная категория является обобщенным описанием различных групп требований к ИС, устанавливая для этих групп единый базис знаний о требованиях к ИС, их представлениях и способах преобразования этих представлений самих в себя и друг в друга.

Связи между моделью M_{IS}^{Pt} и моделями (2.20), (B.1), (B.2), (2.24) и (B.3)-(B.8) можно описать следующим набором ОКФ [159]:

$$\begin{aligned}
& \Phi_{M_{IS}^{Pt}}^{L_{IS}^B} : L_{IS}^B \rightarrow M_{IS}^{Pt}; \Phi_{M_{IS}^{Pt}}^{L_{IS}^{IB}} : L_{IS}^{IB} \rightarrow M_{IS}^{Pt}; \Phi_{M_{IS}^{Pt}}^{L_{IS}^S} : L_{IS}^S \rightarrow M_{IS}^{Pt}; \\
& \Phi_{M_{IS}^{Pt}}^{L_{IS}^f} : L_{IS}^f \rightarrow M_{IS}^{Pt}; \Phi_{M_{IS}^{Pt}}^{L_{IS}^{nf}} : L_{IS}^{nf} \rightarrow M_{IS}^{Pt}; \Phi_{M_{IS}^{Pt}}^{L_{IS}^{fw}} : L_{IS}^{fw} \rightarrow M_{IS}^{Pt}; \\
& \Phi_{M_{IS}^{Pt}}^{L_{IS}^{nfw}} : L_{IS}^{nfw} \rightarrow M_{IS}^{Pt}; \Phi_{L_{IS}^B}^{M_{IS}^{Pt}} : M_{IS}^{Pt} \rightarrow L_{IS}^B; \Phi_{L_{IS}^{IB}}^{M_{IS}^{Pt}} : M_{IS}^{Pt} \rightarrow L_{IS}^{IB}; \\
& \Phi_{L_{IS}^S}^{M_{IS}^{Pt}} : M_{IS}^{Pt} \rightarrow L_{IS}^S; \Phi_{L_{IS}^f}^{M_{IS}^{Pt}} : M_{IS}^{Pt} \rightarrow L_{IS}^f; \Phi_{L_{IS}^{nf}}^{M_{IS}^{Pt}} : M_{IS}^{Pt} \rightarrow L_{IS}^{nf}; \\
& \Phi_{L_{IS}^{fw}}^{M_{IS}^{Pt}} : M_{IS}^{Pt} \rightarrow L_{IS}^{fw}; \Phi_{L_{IS}^{nfw}}^{M_{IS}^{Pt}} : M_{IS}^{Pt} \rightarrow L_{IS}^{nfw}.
\end{aligned} \tag{2.44}$$

Здесь функторы $\Phi_{M_{IS}^{Pt}}^{L_{IS}^B}, \Phi_{M_{IS}^{Pt}}^{L_{IS}^{IB}}, \Phi_{M_{IS}^{Pt}}^{L_{IS}^S}, \Phi_{M_{IS}^{Pt}}^{L_{IS}^f}, \Phi_{M_{IS}^{Pt}}^{L_{IS}^{nf}}, \Phi_{M_{IS}^{Pt}}^{L_{IS}^{fw}}, \Phi_{M_{IS}^{Pt}}^{L_{IS}^{nfw}}$ описывают методы, приемы и способы формирования ППТ к ИС как выделения знаний на основе представлений ранее сформулированных групп требований к ИС. Функторы $\Phi_{L_{IS}^B}^{M_{IS}^{Pt}}, \Phi_{L_{IS}^{IB}}^{M_{IS}^{Pt}}, \Phi_{L_{IS}^S}^{M_{IS}^{Pt}}, \Phi_{L_{IS}^f}^{M_{IS}^{Pt}}, \Phi_{L_{IS}^{nf}}^{M_{IS}^{Pt}}, \Phi_{L_{IS}^{fw}}^{M_{IS}^{Pt}}, \Phi_{L_{IS}^{nfw}}^{M_{IS}^{Pt}}$ описывают методы, приемы и способы ре-использования ППТ к ИС в ходе формирования представлений сформулированных групп требований к ИС.

С прикладной точки зрения, модель ППТ к ИС (2.40) определяет общую структуру фрагмента ИТ формирования и анализа требований к ИС, обеспечивающего решение задачи ре-использования требований в новых проектах ИС. Совокупность ОКФ (2.44) устанавливает возможные методы, приемы и способы формирования базы знаний о требованиях к ИС и использования этой базы знаний в ходе выполнения проектов создания новых ИС.

Добавление в формализованное описание множества ИТ формирования и анализа требований к ИС, основанное на модели (2.29), модели паттерна проектирования сформулированных требований к ИС (2.40) и функторов (2.44), устанавливающих связь между моделью (2.40) и другими элементами модели (2.29), позволяет сформировать модель M_{TrIS}^{Pt} подмножества ИТ формирования и анализа требований к ИС, которые могут быть реализованы на основе концепции выявления и использования знаний о Про и ИС в виде паттернов. Эта модель будет иметь следующий вид [159]:

$$\begin{aligned}
 M_{TrIS}^{Pt} = [& M_{IS}^{Pt}, L_{IS}^B, L_{IS}^{IB}, L_{IS}^S, L_{IS}^f, L_{IS}^{nf}, L_{IS}^{fw}, L_{IS}^{nfw}, \Phi_{M_{IS}^{Pt}}^{L_{IS}^B}, \Phi_{M_{IS}^{Pt}}^{L_{IS}^{IB}}, \Phi_{M_{IS}^{Pt}}^{L_{IS}^S}, \\
 & \Phi_{M_{IS}^{Pt}}^{L_{IS}^f}, \Phi_{M_{IS}^{Pt}}^{L_{IS}^{nf}}, \Phi_{M_{IS}^{Pt}}^{L_{IS}^{fw}}, \Phi_{M_{IS}^{Pt}}^{L_{IS}^{nfw}}, \Phi_{L_{IS}^B}^{M_{IS}^{Pt}}, \Phi_{L_{IS}^{IB}}^{M_{IS}^{Pt}}, \Phi_{L_{IS}^S}^{M_{IS}^{Pt}}, \Phi_{L_{IS}^f}^{M_{IS}^{Pt}}, \Phi_{L_{IS}^{nf}}^{M_{IS}^{Pt}}, \\
 & \Phi_{L_{IS}^{fw}}^{M_{IS}^{Pt}}, \Phi_{L_{IS}^{nfw}}^{M_{IS}^{Pt}}, \Phi_{L_{IS}^B}^{L_{IS}^B}, \Phi_{L_{IS}^{IB}}^{L_{IS}^{IB}}, \Phi_{L_{IS}^S}^{L_{IS}^S}, \Phi_{L_{IS}^f}^{L_{IS}^f}, \Phi_{L_{IS}^{nf}}^{L_{IS}^{nf}}, \Phi_{L_{IS}^{fw}}^{L_{IS}^{fw}}, \Phi_{L_{IS}^{nfw}}^{L_{IS}^{nfw}}].
 \end{aligned} \quad (2.45)$$

В соответствии с приведенным в подразд. 1.1 определением понятия «информационная технология», модель (2.45) содержит выполненные в виде факторов описания методов, приемов и способов применения КСА или его отдельных элементов, используемых для следующих видов преобразований [157]:

а) преобразования представлений различных групп требований друг в друга, позволяющие сформировать ФТ и нефункциональные требования к ИТ-услугам ИС и ИТ-сервисам, реализующим эти ИТ-услуги, с учетом главной цели деятельности ИС и выдвинутой в подразд. 2.3 концепции представления требований к ИС;

б) преобразования представлений различных групп требований к ИС в представления ППТ к ИС, позволяющие выявлять новые знания об автоматизируемых объектах и процессах, а также знания о новых подходах к формированию требований и способов их представления;

в) преобразования представлений ППТ к ИС в представления различных групп требований к ИС, позволяющие использовать знания об автоматизируемых объектах и процессах, а также знания о новых подходах к формированию требований и способов их представления в ходе формирования и уточнения отдельных требований или групп требований, выдвигаемых к конкретной разрабатываемой ИС.

Применение ИТ формирования и анализа требований к ИС, в основу которой положена концепция выявления и использования знаний о Про и ИС в виде паттернов, позволит решить задачу ре-использования требований к ИС за счет формирования и постоянного обновления базы знаний об автоматизируемых объектах и процессах, а также знаний о новых подходах к формированию требований и способов их представления.

2.7 Выводы ко второму разделу

1. На основе приведенных в разд. 1 результатов анализа подходов к представлению ИС предложено уточнение определений основных понятий сервисного подхода к созданию ИС. Разработаны ВМ взаимосвязи основных понятий сервисного подхода к созданию ИС и описания архитектуры ИС на основе сервисного подхода (рис. 2.1 и рис. 2.2 соответственно). Полученные результаты, в отличие от существующих, рассматривают ИС как совокупность ИТ-услуг и реализующих эти услуги ИТ-сервисов, формируемых на основе имеющихся проектных решений, удовлетворяющих множеству требований, выдвинутых к системе. Это позволяет сократить затраты на создание ИС за счет ре-использования системных компонентов.

2. Исходя из предложенных в подразд. 2.1 определений основных понятий сервисного подхода к созданию ИС, было уточнено основное определение понятия «требование к ИС», предложены классификация групп требований к ИС, основные определения групп требований к ИС, а также формальные теоретико-множественные описания множества требований к ИС и отдельных групп требований к ИС (выражения (2.1)-(2.9)).

3. Для устранения проблемы изначальной неполноты множества СТ к ИС введено понятие «универсум требований к ИС» и предложено теоретико-множественное описание этого понятия (выражение (2.19)). Данное понятие, в отличие от существующих, позволяет учитывать влияние на множество СТ к ИС методов и ИТ, используемых Поставщиком и Потребителем для выявления и анализа этих требований.

4. Разработана концепция представления требований к ИС как набор следующих положений:

а) отказ от рассмотрения только множества СТ к ИС и изначальное представление требований к ИС как элементов универсума, включающего в себя как известные, так и неизвестные Поставщику, Потребителю или им

обоим требования к ИС, а также методы формирования этих требований;

б) изначальное многообразие представлений требований к ИС в виде данных, информации и знаний;

в) процессный подход к описанию требований, определяющий минимальную процессную атрибутивную модель требования к ИС;

г) подход к управлению требованиями к ИС, основанный на изложенном в п. 1.2.5 основном принципе УТ.

5. Изложенная концепция представления требований к ИС определяет основные особенности теоретико-категорных моделей невыявленных требований (2.20), частично выявленных требований (В.1) и (В.2), а также групп требований к ИС (2.24), (В.3)-(В.8). Эти модели позволяют установить связи между различными представлениями отдельных требований к ИС в пределах соответствующих множеств требований к ИС и подмножеств отдельных групп требований.

6. Разработаны математические модели (2.29) и (2.30), описывающие множество требований к информационной системе на разных стадиях их формирования. Данные модели, в отличие от существующих, описывают связи между представлениями требований к информационной системе в ходе их формирования, что позволяет формально описать множество возможных АФ создания ИС с помощью сервисного подхода.

7. Для формализованного описания преобразований представлений групп требований друг в друга разработано формализованное описание ОКФ (выражение (2.39)), позволяющее установить два основных способа реализации подобного функтора в рамках АФ создания ИС и ИТ формирования и анализа требований к ИС:

а) формирование и реализация некоего общего алгоритма или базы правил преобразования исходного представления требования к ИС в конечное представление этого же требования или множества других требований

б) создание и постоянное развитие некоего множества алгоритмов или некоей системы правил преобразования исходного представления требования к ИС в конечное представление этого же требования или множества других требований.

8. Для решения проблемы ре-использования требований к ИС в ходе со-

здания новых и развития существующих систем разработано определение ППТ к ИС и предложена математическая модель, описывающая подобные паттерны (выражение (2.40)). Эта модель унифицирует описание отдельного требования к ИС, что позволяет формализовать процесс ре-использования требований к информационной системе в ходе создания новых и реинжиниринга существующих систем. В качестве результата данной формализации предложена модель подмножества ИТ формирования и анализа требований к ИС, которые могут быть реализованы на основе выявления и использования знаний о ПрО и ИС в виде паттернов (выражение (2.45)). Применение подобной ИТ позволяет решить задачу ре-использования требований к ИС за счет формирования и постоянного обновления базы знаний об автоматизируемых объектах и процессах, а также знаний о новых подходах к формированию требований и способов их представления.

Основные результаты данного раздела изложены в работах [118, 119, 128, 129, 134, 135, 140, 145-148, 157-159].

РАЗДЕЛ 3

МОДЕЛИ АРХИТЕКТУРНОГО ФРЕЙМВОРКА

МАКРОПРОЕКТИРОВАНИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ

3.1 Обобщенная модель архитектурного фреймворка
макропроектирования информационной системы

Изложенная в разд. 2 концепция представления требований к ИС, в частности, модель подмножества ИТ формирования и анализа требований к ИС (2.45) не зависит от вида архитектуры конкретной ИС. Поэтому до начала программной реализации любой из ИТ данного подмножества следует решить задачу адаптации модели (2.45) к особенностям конкретной архитектуры ИС.

Решением этой задачи в общем случае является АФ, определяющий особенности описания ИС, ее элементов и архитектуры, а также особенности решения задач интеграции ИС из отдельных элементов на различных уровнях представления. Модель такого АФ должна отражать тот набор методологических положений, знаний, видов моделей и методов, правил и практик, которые могут быть использованы в процессах создания ИС. Место АФ в процессах создания конкретной ИС показано на рис. 3.1 [160, 161].

Из рис. 3.1 следует, что основная точка зрения участников процессов создания ИС на АФ представляет его как своеобразный фильтр, выделяющий и формирующий из всего множества концептуальных положений, видов моделей, методов, знаний, правил и практик создания ИС внутренне непротиворечивое подмножество, элементы которого необходимы для создания конкретной ИС. Под внутренней непротиворечивостью здесь следует понимать существование отображений, связывающих концептуальные положения, виды моделей, методы, знания, правила и практики создания ИС в единую целостную систему.

Основываясь на данном представлении, любой АФ можно разделить на две части. К первой части – формальной – относятся поддающиеся формаль-



Рисунок 3.1 – Место архитектурного фреймворка в процессах создания информационной системы

ному описанию виды моделей, методы, декларативные знания, процедурные правила и алгоритмы, которые могут быть использованы при решении задач, возникающих в процессах создания ИС. Ко второй части – неформальной – относятся неформальные или слабо формализуемые концептуальные положения, стандарты, знания, правила и практики, которые могут быть использованы конкретными представителями Поставщика и Потребителя для успешного выполнения процессов создания ИС. Такое разделение позволяет рассматривать первую часть АФ в виде своеобразного банка видов моделей, методов, знаний, правил и алгоритмов, элементы которого используются для описаний функторов (2.44) любой из возможных ИТ формирования и анализа требований к ИС, описанных моделью (2.45). Целесообразно описать формальную часть АФ в виде следующей модели [160, 161]:

$$M_{AF} = [L_{Md}, L_{Mt}, L_{Kn}, L_{Al}, \Phi_{L_{Mt}}^{L_{Md}}, \Phi_{L_{Kn}}^{L_{Md}}, \Phi_{L_{Al}}^{L_{Md}}, \Phi_{L_{Kn}}^{L_{Mt}}, \Phi_{L_{Al}}^{L_{Mt}}, \Phi_{L_{Al}}^{L_{Kn}}], \quad (3.1)$$

где M_{AF} – обобщенная модель формальных элементов АФ;

L_{Md} – модель набора видов моделей, используемых АФ;

L_{Mt} – модель набора методов, используемых АФ;

L_{Kn} – модель набора декларативных знаний, используемых АФ;

L_{Al} – модель набора процедурных правил и алгоритмов, используемых АФ;

$\Phi_{L_{Mt}}^{L_{Md}}$ – ОКФ, устанавливающий связь между моделями L_{Md} и L_{Mt} ;

$\Phi_{L_{Kn}}^{L_{Md}}$ – ОКФ, устанавливающий связь между моделями L_{Md} и L_{Kn} ;

$\Phi_{L_{Al}}^{L_{Md}}$ – ОКФ, устанавливающий связь между моделями L_{Md} и L_{Al} ;

$\Phi_{L_{Kn}}^{L_{Mt}}$ – ОКФ, устанавливающий связь между моделями L_{Mt} и L_{Kn} ;

$\Phi_{L_{Al}}^{L_{Mt}}$ – ОКФ, устанавливающий связь между моделями L_{Mt} и L_{Al} ;

$\Phi_{L_{Al}}^{L_{Kn}}$ – ОКФ, устанавливающий связь между моделями L_{Kn} и L_{Al} .

Из модели (3.1) следует, что основным элементом любого АФ создаваемой ИС следует считать набор видов моделей, используемых конкретным АФ в процессах создания ИС с соответствующей архитектурой. Поэтому основное внимание следует сосредоточить на детализации описаний элементов модели L_{Md} и особенностях практической реализации этих описаний.

Использование изложенной в разд. 2 концепции представления требований к ИС кардинально изменяет модели, методы, знания, правила и алгоритмы, используемые в ходе выполнения процессов, непосредственно работающих с требованиями, а также процесса проектирования архитектуры системы. Это дает возможность говорить о существовании АФ, основанного на положениях сервисного подхода и предложенной концепции представления требований к ИС. Такой АФ в дальнейшем будем называть АФ макропроектирования ИС. В соответствии с описанием понятия «архитектурный фреймворк», приведенным в подразд. 1.3, модель АФ макропроектирования ИС должна уста-

новить общие особенности и ограничения создания, анализа, интерпретации и использования АД ИС на основе требований к ИС, применяемых в указанных выше процессах создания ИС.

Использование концепции представления требований к ИС, предложенной в подразд. 2.3, позволяет уточнить взаимосвязи основных элементов понятия «АД системы на основе системного подхода» (см. рис. 2.2). Контекстная ДК понятия «АД системы на основе сервисного подхода с точки зрения требований к системе» показана на рис. 3.2 [160, 161].

Создание ИС с применением АФ макропроектирования ИС, основанное на предложенной контекстной ДК, подразумевает разработку и интеграцию АД на трех основных уровнях представления: общесистемном уровне, уровне ИТ-услуг и уровне ИТ-сервисов. При этом процесс проектирования архитектуры системы, который может осуществляться как сверху вниз, так и снизу вверх, предполагает обязательную интеграцию описаний элементов ИС, выполненных на разных уровнях представления. Так, например, АД ИТ-сервиса должно включать: описание взаимосвязанных функциональных операций, в совокупности обеспечивающих физическую реализацию функциональности сервиса; АД ИТ-услуги – описание ИТ-сервисов, в совокупности обеспечивающих реализацию данной ИТ-услуги; АД ИС – описания взаимосвязанных ИТ-услуг. Иными словами, архитектура ИС, как и АД ИС, на всех уровнях представления рассматривается как набор автономных модулей, обеспечивающих реализацию конкретного варианта конфигурации ФС ИС. Каждый такой модуль любого уровня представления характеризуется набором входных данных, необходимых для получения информации извне, и выходных данных, необходимых для отображения результатов выполнения ИТ-услуги ИС, а также интерфейсов, обеспечивающих взаимодействие модуля с внешней средой. При этом внешней средой могут являться другие модули этой же ИС, внешние системы, или пользователи ИС, взаимодействующие с модулем через пользовательский интерфейс.

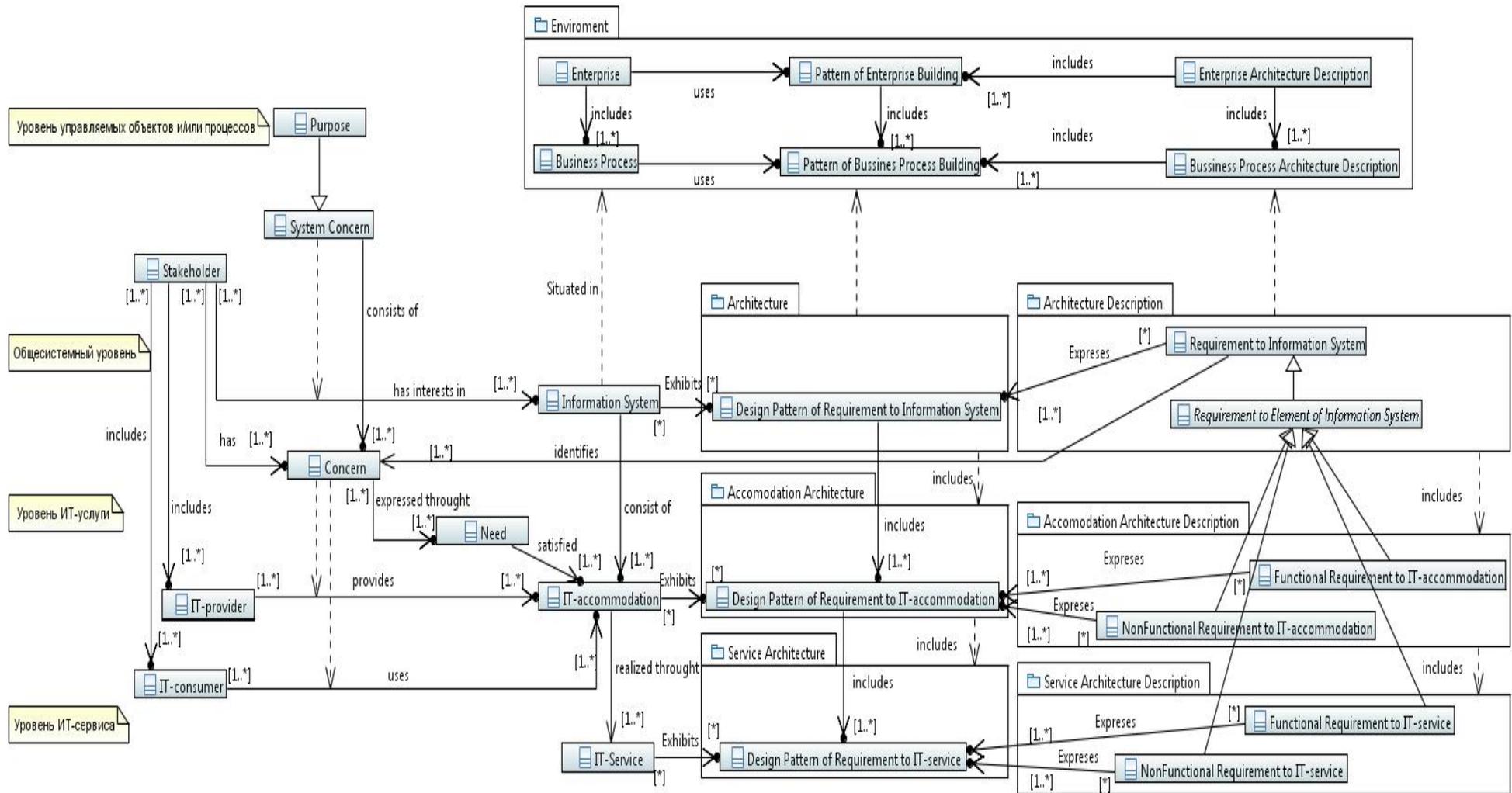


Рисунок 3.2 – Контекстная диаграмма классов понятия «описание архитектуры системы на основе сервисного подхода с точки зрения требований к системе»

Такая точка зрения на АФ макропроектирования ИС определяет как наиболее предпочтительный способ проектирования архитектуры ИС методом «снизу вверх». Это обусловлено наличием библиотеки готовых элементов ИС (ИТ-услуг и отдельных ИТ-сервисов), эффект от ре-использования которых увеличивается при оперировании описанием элементов ИС, выполненным на уровне отдельных ИТ-услуг и особенно отдельных ИТ-сервисов. Данный уровень представления позволяет обеспечить ре-использование ИТ-сервисов, даже в том случае, когда не востребована полная функциональность созданных ранее ИТ-услуг, реализацию которых обеспечивают данные ИТ-сервисы. В этом случае такие ИТ-сервисы становятся элементами новой ИТ-услуги.

Поэтому процесс проектирования архитектуры системы в рамках АФ макропроектирования ИС следует рассматривать как процесс синтеза варианта конфигурации ФС ИС из отдельных ИТ-услуг. При этом предполагается, что ре-использование требований к ИС позволит идентифицировать те ИТ-услуги, которые находятся в библиотеке ранее реализованных требований, доступной Поставщику в ходе процессов создания конкретной ИС. Процессы, непосредственно работающие с требованиями к ИС, в этом случае целесообразно рассматривать как процессы идентификации требований к системе в целом, а также ФТ и нефункциональных требований к ИТ-услугам, образующим ФС ИС.

Сказанное выше позволяет рассматривать модель АФ макропроектирования ИС как частный случай модели (3.1), в которой компонент L_{Md} определяется как совокупность ППТ к ИС, в общем случае описываемых моделью (2.40). Тогда модель АФ макропроектирования ИС будет иметь вид [134, 160, 161]:

$$M_{AF} = [M_{str}^{Pt}, M_{bhv}^{Pt}, L_{Kn}, L_{Al}, \Phi_{M_{bhv}^{Pt}}^{M_{str}^{Pt}}, \Phi_{L_{Kn}}^{M_{str}^{Pt}}, \Phi_{L_{Al}}^{M_{str}^{Pt}}, \Phi_{L_{Kn}}^{M_{bhv}^{Pt}}, \Phi_{L_{Al}}^{L_{bhv}^{Pt}}, \Phi_{L_{Al}}^{L_{Kn}}], \quad (3.2)$$

где M_{str}^{Pt} – модель структурных ППТ к ИС, являющаяся частным случаем модели (2.40);

M_{bhv}^{Pt} – модель поведенческих ППТ к ИС, являющаяся частным случаем модели (2.40);

$\Phi_{M_{bhv}^{Pt}}^{M_{str}^{Pt}}$ – ОКФ, устанавливающий связь между моделями M_{str}^{Pt} и M_{bhv}^{Pt} ;

$\Phi_{L_{Kn}}^{M_{str}^{Pt}}$ – ОКФ, устанавливающий связь между моделями M_{str}^{Pt} и L_{Kn} ;

$\Phi_{L_{Al}}^{M_{str}^{Pt}}$ – ОКФ, устанавливающий связь между моделями M_{str}^{Pt} и L_{Al} ;

$\Phi_{L_{Kn}}^{M_{bhv}^{Pt}}$ – ОКФ, устанавливающий связь между моделями M_{bhv}^{Pt} и L_{Kn} ;

$\Phi_{L_{Al}}^{M_{bhv}^{Pt}}$ – ОКФ, устанавливающий связь между моделями M_{bhv}^{Pt} и L_{Al} .

Следует отметить, что, как и для модели (3.1), основным элементом АФ макропроектирования ИС следует считать набор структурных ППТ, устанавливающих допустимые для использования виды моделей и формализованных описаний различных групп требований к ИС. Для их успешной реализации в рамках АФ макропроектирования ИС необходимо выделить основные способы представления различных групп требований к ИС. В общем случае, как следует из изложенного выше материала, представления отдельных требований к ИС определяются на основе:

а) определения главной цели деятельности ИС как формирования и отображения единого целостного информационного представления объекта или процесса в соответствии с поставленными перед системой целями;

б) сформулированной в подразд. 3.1. концепции представления требований к ИС;

в) категорных моделей отдельных групп требований к ИС (3.16), (3.20)-(3.25), категорной модели СТ к ИС (3.26) и категорной модели универсума требований к ИС (3.27);

г) атрибутивных моделей описания различных групп требований к ИС.

Исходя из этих условий, предлагаются следующие способы представления отдельных требований к ИС на уровне данных, информации и знаний (табл. 3.1) [134].

Таблица 3.1 – Описание способов представлений отдельных требований к информационной системе на различных уровнях

Виды представлений требования к ИС	Способ представления требования к ИС
Представление требования к ИС на уровне данных	Атрибутивная модель требования, используемая для идентификации и управления требованием к ИС
Представление требования к ИС на уровне информации	Описание требования к ИС в виде текстов на естественном языке, множестве формальных языков (например, в виде ВМ) или же в виде набора целевых показателей, характеризующих степень удовлетворения требования к ИС
Представление требования к ИС на уровне знаний	Онтология элемента управляемого объекта или процесса, элемента ИС или ИС в целом, к которым выдвигается требование

Предложенные в табл. 3.1 способы представлений отдельных требований к ИС определяют дополнительное условие, дополняющее рассмотренные выше условия для случая формирования представлений ППТ к ИС. Это условие заключается в следующем: различные представления одного и того же требования к ИС не должны исказить заложенную в это требование Поставщиком и Потребителем семантику автоматизируемого объекта, процесса, разрабатываемой ИС или ее элемента. Для выполнения этого условия необходимо выбрать подход к решению проблемы формирования единых описаний автоматизируемого объекта или процесса, а также разрабатываемой ИС.

На сегодняшний день имеется несколько равноправных методов решения данной проблемы. Одним из таких методов является создание специализированных описаний и формализованных представлений, которые опреде-

ляют синтаксис и семантику конкретных реализаций ИС и ее компонентов. Такие описания и формализованные представления М. Фаулер предложил называть метамоделями. Хотя термин «метамодель» в понимании Фаулера приобрел более узкий смысл, суть его осталась той же, что и в традиционной теории систем, где существование метаописания и метамодели является и необходимым, и конструктивным [162].

Некоторые исследователи-практики особое внимание обращают не на то, каким образом метамодель можно формализовать и поместить в текст, именуемый «общая теория систем», а на то, что метамодель присутствует в конструкции систем как принцип устройства этой конструкции и одновременно как механизм, обеспечивающий конструирование, существование и взаимодействие множеств моделей [163]. При этом сам механизм реализуется не с помощью специально выделенных структур данных или индексации структур – это очевидно вытекает из факта существования Геделевской схемы нумерации. Так А.О. Поляков в [164] указывает, что все естественные открытые системы имеют не гомеостатический, а гомеокинетический характер поведения и устройства. Представляется разумным предположить, что реализация метамодели ИС как открытой системы включает в себя не только определенные типы структур данных и взаимодействие этих структур, но и определенные типы и способы организации этих взаимодействий, то есть имеет характер сложно организованной, но целостной системы взаимодействий и способов организации этих взаимодействий. Наиболее приемлемым способом организации метамодели А.О. Поляков видит реализацию некоторого механизма, обеспечивающего отдельную организацию работы механизма логического вывода и механизма интерпретации результатов вывода [164].

Использование метамodelей для описания и принципов устройства моделей представлений требований к ИС на различных уровнях и механизма, обеспечивающего конструирование, существование и взаимодействие множеств моделей представлений требований к ИС на различных уровнях, позволяет определить способы представления ППТ к ИС следующим образом (табл. 3.2) [134].

Таблица 3.2 – Описание способов представления паттерна проектирования требований к информационной системе на различных уровнях

Виды представления паттерна проектирования требования к ИС	Способ представления паттерна проектирования требования к ИС
Представление паттерна проектирования требования к ИС на уровне данных	Метамоделю, определяющая синтаксис и семантику построения атрибутивной модели требования, принадлежащего конкретной группе требований к ИС
Представление паттерна проектирования требования к ИС на уровне информации	Метамоделю, определяющая синтаксис и семантику описаний требования, принадлежащего конкретной группе требований к ИС, в виде текстов на естественном языке или же на множестве формальных языков (например, в виде визуальных моделей)
Представление паттерна проектирования требования к ИС на уровне знаний	Описание базовых онтологий Про или ИС, а также метамоделю, определяющая синтаксис и семантику онтологии требования, принадлежащего конкретной группе требований к ИС, на основе выделенных базовых онтологий

3.2 Разработка паттернов проектирования требований к информационной системе на уровне данных

Как показано в подразд. 3.1, представлением требования к ИС на уровне данных предложено считать атрибутивную модель этого требования. Данная МТ применяется для выполнения операции хранения требования в каталоге требований к ИС, а также для управления этим требованием. Представлением ППТ к ИС на уровне данных предложено считать метамоделю, определя-

ющую синтаксис и семантику построения атрибутивной МТ, принадлежащего конкретной группе требований к ИС.

Однако, атрибутивные МТ к ИС могут серьезно различаться при описании требований, принадлежащих к разным группам требований к ИС. Кроме того, не существует научно обоснованных гарантий адекватности каких бы то ни было атрибутивных МТ процессам разработки ИС. В то же время можно выделить ряд групп показателей, которые необходимы для выполнения таких стандартных формальных операций, как ввод нового требования, отображение введенного требования, редактирование ранее введенного требования, удаление ранее введенного требования, а также для осуществления операций УТ к ИС. Эти показатели должны присутствовать во всех атрибутивных МТ всех групп требований к ИС.

В дополнение к сказанному следует отметить, что разрабатываемая модель представления требований на уровне данных и метамодель, определяющая ее синтаксис и семантику, должны быть в максимальной степени независимы от особенностей их технологической реализации. В частности, эти модель и метамодель должны быть в максимальной степени независимы от особенностей формального описания конкретных моделей данных, которые, в свою очередь, определяют особенности выполнения операций хранения и управления представлениями требований к ИС. Это условие позволит обеспечить возможность использования разрабатываемой модели и метамодели даже в случае кардинального изменения ИТ хранения данных.

Для разработки модели представления требований к ИС на уровне данных введем по аналогии с моделями отдельных групп требований (2.24), (B.3)-(B.8) обобщенную модель СТ к ИС M_{IS}^{tr} вида [134, 165]:

$$M_{IS}^{tr} = [D_{IS}^{tr}, I_{IS}^{tr}, K_{IS}^{tr}, H(D_{IS}^{tr}), H(I_{IS}^{tr}), H(K_{IS}^{tr}), H(D_{IS}^{tr}, I_{IS}^{tr}), H(I_{IS}^{tr}, D_{IS}^{tr}), H(D_{IS}^{tr}, K_{IS}^{tr}), H(K_{IS}^{tr}, D_{IS}^{tr}), H(I_{IS}^{tr}, K_{IS}^{tr}), H(K_{IS}^{tr}, I_{IS}^{tr})], \quad (3.3)$$

где D_{IS}^{tr} – подкласс представлений СТ к ИС на уровне данных;

I_{IS}^{tr} – подкласс представлений СТ к ИС на уровне информации;

K_{IS}^{tr} – подкласс представлений СТ к ИС на уровне знаний;

$H(D_{IS}^{tr})$ – подмножество морфизмов, описывающих преобразования представлений СТ к ИС на уровне данных самих в себя и друг в друга, имеющее следующий вид:

$$\forall X, Y \in D_{IS}^{tr} \quad H(D_{IS}^{tr}) = H(1_X) \cup H(X, Y) \cup H(1_Y); \quad (3.4)$$

$1_X, 1_Y$ – единичные морфизмы, условия существования которых описаны в [141];

$H(I_{IS}^{tr})$ – подмножество морфизмов, описывающих преобразования представлений СТ к ИС на уровне информации самих в себя и друг в друга, имеющее следующий вид:

$$\forall X, Y \in I_{IS}^{tr} \quad H(I_{IS}^{tr}) = H(1_X) \cup H(X, Y) \cup H(1_Y); \quad (3.5)$$

$H(K_{IS}^{tr})$ – подмножество морфизмов, описывающих преобразования представлений СТ к ИС на уровне знаний самих в себя и друг в друга, имеющее следующий вид:

$$\forall X, Y \in K_{IS}^{tr} \quad H(K_{IS}^{tr}) = H(1_X) \cup H(X, Y) \cup H(1_Y); \quad (3.6)$$

$H(D_{IS}^{tr}, I_{IS}^{tr})$ – подмножество морфизмов, описывающих методы, приемы и способы преобразования представлений СТ к ИС на уровне данных в представления СТ к ИС на уровне информации;

$H(I_{IS}^{tr}, D_{IS}^{tr})$ – подмножество морфизмов, описывающих методы, приемы и способы преобразования представлений СТ к ИС на уровне информации в представления СТ к ИС на уровне данных;

$H(D_{IS}^{tr}, K_{IS}^{tr})$ – подмножество морфизмов, описывающих методы, приемы и способы преобразования представлений СТ к ИС на уровне данных в представления СТ к ИС на уровне знаний;

$H(K_{IS}^{tr}, D_{IS}^{tr})$ – подмножество морфизмов, описывающих методы, приемы и способы преобразования представлений СТ к ИС на уровне знаний в представления СТ к ИС на уровне данных;

$H(I_{IS}^{tr}, K_{IS}^{tr})$ – подмножество морфизмов, описывающих методы, приемы и способы преобразования представлений СТ к ИС на уровне информации в представления СТ к ИС на уровне знаний;

$H(K_{IS}^{tr}, I_{IS}^{tr})$ – подмножество морфизмов, описывающих методы, приемы и способы преобразования представлений СТ к ИС на уровне знаний в представления СТ к ИС на уровне информации.

Данную модель целесообразно использовать для выделения тех элементов моделей представлений требований к ИС на различных уровнях, которые должны быть едиными для различных групп требований и определяются паттернами проектирования требования к ИС.

Тогда модель представления СТ к ИС на уровне данных следует представлять как формализованное описание каждого элемента подкласса D_{IS}^{tr} с последующим уточнением этого описания после отнесения данного элемента к одному из следующих подклассов: D_{IS}^B , D_{IS}^{IB} , D_{IS}^s , D_{IS}^f , D_{IS}^{nf} , D_{IS}^{fw} , D_{IS}^{nfw} . Данная точка зрения позволяет представить модель подкласса представлений СТ к ИС на уровне данных как кортеж атрибутов, который структурно разделен на такие части [134, 165]:

$$M_{D_{IS}^{tr}} = \left\langle \langle M_D^{AtPt} \rangle, \langle M_D^{Atgr} \rangle \right\rangle, \quad (3.7)$$

где $M_{D_{IS}^{tr}}$ – модель подкласса представлений СТ к ИС на уровне данных;

$\langle M_D^{AtPt} \rangle$ – кортеж элементов атрибутивной МТ к ИС, которые определяются ППТ к ИС на уровне данных и являются обязательными для требований любой группы;

$\langle M_D^{Atgr} \rangle$ – кортеж элементов атрибутивной МТ к ИС, которые определяются, исходя из индивидуальных особенностей выполнения Поставщиком и Потребителем процессов, непосредственно работающих с представлениями требований на уровне данных конкретной группы.

В общем случае конкретный вид элемента $\langle M_D^{AtPt} \rangle$ модели (3.7) определяется совокупностью структурных и поведенческих ППТ к ИС. Однако, как было показано выше, поведенческие ППТ определяются особенностями структурных ППТ к ИС. На практике это означает, что конкретные методы и алгоритмы выполнения основных операций над требованиями к ИС на уровне данных могут быть выделены из множества общих методов на основе как общей специфики процессов создания ИС, так и конкретных особенностей структурных ППТ к ИС. Поэтому основное внимание в данном разделе уделяется разработке структурных ППТ к ИС, определяющих семантику общих представлений требований различных групп на уровне данных в ходе выполнения основных операций над требованиями. В качестве таких операций следует выделить:

- а) идентификацию требования к ИС;
- б) УТ к ИС;
- в) операции создания, отображения, редактирования и удаления описаний отдельных требований, версий этих требований, а также представлений требований и их версий.

Рассмотрим структурный ППТ к ИС на уровне данных для операции идентификации требований различных групп. В данном случае под термином «идентификация требования» понимается установление следующих характеристик требования к системе:

- а) уникальное описание каждого требования;
- б) описание взаимосвязей между требованиями и правообладателями;

в) уникальное описание правообладателей требований;

г) описание взаимосвязей требований друг с другом для прослеживания каждого требования до первоначальных требований правообладателей.

В дополнение к данным характеристикам при идентификации требований к ИС, которые могут относиться к разным группам требований, необходимо установление таких характеристик:

а) уникальное описание каждой группы требований к ИС;

б) уникальное описание каждой версии требования к ИС;

в) уникальное описание сотрудников Поставщика, которые либо являются авторами требования к ИС или версии этого требования, либо выполняют работу по формулированию требования к ИС или версии этого требования, высказанного представителем (или представителями) Потребителя;

г) уникальное описание проекта создания ИС, в ходе которого было сформулировано требование к ИС или версия этого требования;

д) факт использования требования к ИС или версии этого требования в проекте создания ИС.

При этом стоит учесть, что для ре-использования требований к ИС или версий этих требований в проектах создания различных ИС необходимо, чтобы описание требования к ИС было первичным по отношению к описанию проекта создания конкретной ИС.

На основании этих показателей предлагается структурный паттерн идентификации требования к ИС, который имеет следующий вид [134, 165]:

$$Pt_{Id} = \langle At_{rg}, At_r, At_{asp}, At_{acs}, At_{proj}, \langle at_{rg}, at_r, at_{asp}, at_{acs}, at_{proj} \rangle \rangle, \quad (3.8)$$

где Pt_{Id} – модель структурного паттерна идентификации требования к ИС, определяющего семантику моделей требований к ИС различных групп;

At_{rg} – кортеж атрибутов, описывающих группы требований к ИС;

At_r – кортеж атрибутов, описывающих отдельные СТ к ИС;

At_{asp} – кортеж атрибутов, описывающих сотрудников Поставщика как возможных авторов требований к ИС;

At_{acs} – кортеж атрибутов, описывающих сотрудников Потребителя как возможные источники требований к ИС;

At_{proj} – кортеж атрибутов, описывающих проекты создания ИС, для которых формулируются требования;

at_{rg} – атрибут, идентифицирующий группу требований к ИС;

at_r – атрибут, идентифицирующий отдельное требование к ИС;

at_{asp} – атрибут, идентифицирующий сотрудника Поставщика как автора требований к ИС;

at_{acs} – атрибут, идентифицирующий сотрудника Потребителя как автора требований к ИС;

at_{proj} – атрибут, идентифицирующий проект создания ИС, для которого было сформулировано требование;

$\langle at_{rg}, at_r, (at_{apr}), (at_{au}), at_{proj} \rangle$ – кортеж атрибутов, описывающий факт идентификации конкретного требования к ИС, сформулированного конкретными авторами в проекте создания конкретной ИС.

Интерпретацией модели (3.8) в ходе разработки ИТ формирования и анализа требований к ИС является схема одной или нескольких витрин данных, предназначенных для хранения следующей исторической информации:

а) о каталоге СТ, допустимых для ре-использования в проектах создания различных ИС;

б) о каталоге СТ, выдвинутых в ходе выполнения проекта создания конкретной ИС.

В случае использования модели (3.8) для идентификации версии требования к ИС, данная модель принимает вид, описанный выражением (Г.1) Приложения Г.

Как было показано в п. 1.2.5, управление отдельными требованиями основано на постепенном преобразовании множества начальных значений атрибутов, описы-

вающих требование, в множество желаемых значений тех же атрибутов. Под желаемым значением следует понимать значение, которое приобретает атрибут при описании реализованного требования, проверенного соответствующими тестами.

Вместо анализа совокупности атрибутов, определяющих состояние требования к ИС, часто предлагается использовать анализ статуса требования (см. п. 1.2.5). Такой подход позволяет организовать УТ к ИС как анализ, проводимый лицами, принимающими решение (среди которых могут быть как представители Поставщика, так и представители Потребителя). Анализу подвергаются возможности изменения статуса требования и, в случае принятия положительного решения, изменения текущего значения статуса анализируемого требования.

Данная процедура определяет описание процесса УТ к ИС, выполняемого в течение всего ЖЦ требования, следующим образом (табл. 3.3) [134].

Предложенное описание процесса УТ, выполненное в соответствии с процессной моделью, рассмотренной в подразд. 3.1, определяет необходимость установления следующих характеристик:

- а) уникальное описание управляемого требования;
- б) описание текущего статуса управляемого требования;
- в) уникальное описание лиц, принимающих решение об управлении требованием со стороны Поставщика;
- г) уникальное описание лиц, принимающих решение об управлении требованием со стороны Потребителя;
- д) дата и время принятия управленческого решения об изменении текущего статуса требования;
- е) описание причины принятия управленческого решения об изменении статуса требования;
- ж) дата и время изменения текущего статуса требования.

На основании этих показателей предлагается структурный ППТ к ИС на уровне данных для УТ, который имеет следующий вид [134, 165]:

Таблица 3.3 – Описание процесса управления требованием
к информационной системе

№ п/п	Элемент процессной модели	Значение элемента процессной модели
1	Название (наименование) процесса	Процесс УТ к ИС
2	Ожидаемые результаты выполнения процесса	Установление нового текущего статуса требования к ИС
3	Виды деятельности, выполняемые в рамках процесса	Анализ текущего статуса требования к ИС, принятие решения об изменении текущего статуса требования к ИС, изменение текущего статуса требования к ИС
4	Цели, достижение которых свидетельствует о выполнении процесса	Изменение значения текущего статуса требования к ИС на значение, желательное для Поставщика и Потребителя
5	Ресурсы, обрабатываемые процессом	Требования к ИС
6	Механизмы процесса	Лица, принимающие решение со стороны Поставщика и со стороны Потребителя
7	Управляющие воздействия процесса	Множество состояний требования к ИС, элементы которого могут назначаться текущими статусами требования к ИС

$$Pt_{Ctrl} = \langle At_r, At_{cst}, At_{dsp}, At_{dcs}, \langle at_r, at_{cst}, at_{dsp}, at_{dcs}, at_{ddt}, at_{cause}, at_{cdt} \rangle \rangle, \quad (3.9)$$

где Pt_{Ctrl} – модель структурного ППТ к ИС, определяющего семантику моделей требований к ИС различных групп;

At_{cst} – кортеж атрибутов, описывающих все возможные статусы требования к ИС;

- At_{dsp} – кортеж атрибутов, описывающих лиц, принимающих решение об изменении текущего статуса требования к ИС со стороны Поставщика;
- At_{dcs} – кортеж атрибутов, описывающих лиц, принимающих решение об изменении текущего статуса требования к ИС со стороны Потребителя;
- at_{cst} – атрибут, идентифицирующий текущее состояние требования к ИС;
- at_{dsp} – атрибут, идентифицирующий лицо, принимающее решение об изменении текущего статуса требования к ИС со стороны Поставщика;
- at_{dcs} – атрибут, идентифицирующий лицо, принимающее решение об изменении текущего статуса требования к ИС со стороны Потребителя;
- at_{ddt} – атрибут, описывающий дату принятия решения об изменении текущего статуса требования к ИС;
- at_{cause} – атрибут, описывающий причину изменения текущего статуса требования к ИС;
- at_{cdt} – атрибут, описывающий дату перевода требования к ИС в текущее состояние;
- $\langle at_r, at_{cst}, at_{dsp}, at_{dcs}, at_{ddt}, at_{cause}, at_{cdt} \rangle$ – кортеж атрибутов, устанавливающий факт управляющего воздействия на требование к ИС.

Интерпретацией модели (3.9) в ходе разработки ИТ формирования и анализа требований к ИС является схема одной или нескольких витрин данных, предназначенных для хранения исторической информации о целенаправленных изменениях текущих статусов требований к ИС, в результате которых каждое требование либо выполняется с желаемой степенью удовлетворения, либо отклоняется и удаляется из дальнейшего рассмотрения в процессе проектирования архитектуры ИС.

Перечень показателей, подлежащих учету в случае необходимости осуществления управления отдельными версиями требования, приведен в подразд. Г.1 Приложения Г.

В случае использования модели (3.9) для описания процесса управления версией требования к ИС, данная модель принимает вид, описанный выраже-

нием (Г.2) Приложения Г.

В ходе выполнения работ по формулированию требований к ИС, результатом которых является полная совокупность СТ к системе, подвергаемая в дальнейшем анализу, возникает необходимость в выполнении операций создания, отображения, редактирования и удаления описаний отдельных требований, версий этих требований, а также представлений требований и их версий. Выполнение этих операций может вызвать такие последствия:

а) изменение результата идентификации отдельного требования к ИС или его версии;

б) инициация процесса УТ к ИС или его версией как изменения текущего статуса этого требования или его версии.

Поскольку эти последствия могут иметь необратимый характер и влияют не только на отдельный проект создания конкретной ИС, но и на последующие проекты создания различных ИС, необходимо использовать фрагмент атрибутивной МТ к ИС для учета операций, выполняемых над требованием.

Для учета фактов выполнения операций над требованиями к ИС необходимо установление таких характеристик:

а) уникальное описание операций, выполняемых над отдельным требованием к ИС;

б) уникальное описание сотрудников Поставщика, участвующих в выполнении конкретной операции над отдельным требованием к ИС;

в) уникальное описание сотрудников Потребителя, участвующих в выполнении конкретной операции над отдельным требованием к ИС;

г) дата и время выполнения операции над требованием к ИС.

Тогда, с учетом результатов разработки моделей ППТ (3.8), (3.9), (Г1) и (Г.2), предлагается модель структурного ППТ к ИС на уровне данных для учета результатов выполнения операций над требованиями к ИС, которая имеет следующий вид [134, 165]:

$$Pt_{Op} = \langle \langle At_r, At_{opr}, At_{aopsp}, At_{aopcs}, \langle at_r, at_{opr}, at_{aopsp}, at_{aopcs}, at_{dopr} \rangle \rangle \rangle, \quad (3.10)$$

где Pt_{Op} – модель структурного паттерна учета результатов выполнения операций над требованием к ИС, определяющего семантику МТ к ИС различных групп;

At_{opr} – кортеж атрибутов, описывающих операции, выполняемые над требованием к ИС;

At_{aopsp} – кортеж атрибутов, описывающих сотрудников Поставщика, участвующих в выполнении операции над требованием к ИС;

At_{aopcs} – кортеж атрибутов, описывающих сотрудников Потребителя, участвующих в выполнении операции над требованием к ИС;

at_{opr} – атрибут, идентифицирующий операцию, выполняемую над требованием к ИС;

at_{aopsp} – атрибут, идентифицирующий сотрудника Поставщика, участвующего в выполнении операции над требованием к ИС;

at_{aopcs} – атрибут, идентифицирующий сотрудника Потребителя, участвующего в выполнении операции над требованием к ИС;

at_{dopr} – атрибут, описывающий дату и время выполнения операции над требованием к ИС;

$\langle at_r, at_{opr}, at_{aopsp}, at_{aopcs}, at_{dopr} \rangle$ – кортеж атрибутов, описывающий факт выполнения конкретной операции над конкретным сформулированным требованием к ИС.

Интерпретацией модели (3.10) в ходе разработки ИТ формирования и анализа требований к ИС является схема одной или нескольких витрин данных, предназначенных для хранения исторической информации о последовательности выполненных операций над сформулированными требованиями к ИС.

В случае использования модели (3.10) для учета результатов выполнения операций над версиями требований к ИС, данная модель принимает вид, описанный выражением (Г.3) Приложения Г.

Таким образом, для случая использования в качестве элементарных описаний создаваемой системы отдельных требований к ИС подкласс объектов D_{IS}^{Pt} модели (2.40) следует рассматривать как множество структурных ППТ к ИС на уровне данных, устанавливающих конкретный вид элемента $\langle M_D^{AtPt} \rangle$ модели (3.7). Это множество ППТ будет в общем случае иметь следующий вид [134, 165]:

$$\begin{aligned}
 D_{IS}^{Pt} = \{ Pt_{Id}, Pt_{Ctrl}, Pt_{Op} \} = \{ & \langle At_{rg}, At_r, At_{asp}, At_{acs}, At_{proj}, \\
 & \langle at_{rg}, at_r, at_{asp}, at_{acs}, at_{proj} \rangle \rangle, \langle At_r, At_{cst}, At_{dsp}, At_{dcs}, \\
 & \langle at_r, at_{cst}, at_{dsp}, at_{dcs}, at_{ddt}, at_{cause}, at_{cdt} \rangle \rangle, \\
 & \langle At_r, At_{opr}, At_{aopsp}, At_{aopcs}, \langle at_r, at_{opr}, at_{aopsp}, at_{aopcs}, at_{dopr} \rangle \rangle \}.
 \end{aligned} \tag{3.11}$$

Для случая использования отдельных версий требований к ИС в качестве элементарных описаний создаваемой системы подкласс объектов D_{IS}^{Pt} модели (3.39) следует рассматривать как множество структурных паттернов проектирования версий требований к ИС, имеющее вид, описанный выражением (Г.4) Приложения Г.

Подкласс морфизмов $H(D_{IS}^{Pt})$ представляет собой совокупность моделей преобразований структурных и поведенческих ППТ к ИС на уровне данных самих в себя и друг в друга. В соответствии с выражением (2.41), данный подкласс по отношению к подклассу объектов D_{IS}^{Pt} , представленному выражением (3.11), состоит из следующих видов морфизмов:

а) единичные морфизмы $1_{Pt_{Id}}$, $1_{Pt_{Ctrl}}$, $1_{Pt_{Op}}$, отображающие структурные ППТ Pt_{Id} , Pt_{Ctrl} и Pt_{Op} соответственно сами в себя;

б) морфизмы $H(Pt_{Id}, Pt_{Ctrl})$, $H(Pt_{Id}, Pt_{Op})$, $H(Pt_{Ctrl}, Pt_{Op})$, $H(Pt_{Op}, Pt_{Ctrl})$, $H(Pt_{Op}, Pt_{Id})$ и $H(Pt_{Ctrl}, Pt_{Id})$, устанавливающие связи между структурными ППТ Pt_{Id} , Pt_{Ctrl} и Pt_{Op} .

Единичные морфизмы структурных ППТ к ИС на уровне данных описы-

вают поведенческий паттерн модификации. Суть данного поведенческого ППТ заключается в выполнении операций следующего вида:

а) добавление новых атрибутов в модель существующего структурного ППТ к ИС;

б) изменение существующих описаний атрибутов в модели существующего структурного ППТ к ИС;

в) удаление неиспользуемых атрибутов из модели существующего структурного ППТ к ИС.

Исходя из сказанного, существование единичного морфизма $1_{Pt_{Id}}^{add} \in 1_{Pt_{Id}}$, описывающего операцию добавления новых атрибутов в модель ППТ Pt_{Id} , определяется следующим условием [134, 165]:

$$\forall At_x \in AT^* \quad \exists 1_{Pt_{Id}}^{add} \in 1_{Pt_{Id}} : Pt_{Id} \rightarrow Pt_{Id} \cup At_x, \quad (3.12)$$

где At_x – подмножество атрибутов, по которым принято решение о целесообразности их добавления в модель паттерна Pt_{Id} ;

AT^* – множество атрибутов, не нарушающих целостность модели паттерна Pt_{Id} ;

$1_{Pt_{Id}}^{add}$ – единичный морфизм, описывающий операцию добавления атрибутов подмножества At_x в модель паттерна Pt_{Id} ;

$1_{Pt_{Id}}$ – подкласс единичных морфизмов модели паттерна Pt_{Id} .

Совокупность условий проверки целостности моделей структурных ППТ в общем случае определяется моделями представления знаний, на основе которых формируются модели паттернов, и в данной работе не рассматривается.

Тогда любой единичный морфизм $1_{Pt_{Id}}^{add}$, реализующий поведенческий паттерн модификации модели паттерна Pt_{Id} , может быть описан операцией вида [134, 165]:

$$1_{Pt_{Id}}^{add} : [At_{rg} \oplus At_x] \oplus [At_r \oplus At_x] \oplus [At_{asp} \oplus at_x] \oplus [At_{acs} \oplus At_x] \oplus [At_{proj} \oplus At_x] \oplus [\emptyset \oplus At_x], \quad (3.13)$$

где \emptyset – пустое множество, представляющее собой не существовавший ранее кортеж атрибутов, который описывает новый аспект идентифицируемого требования.

Использование операции «сложение по модулю 2» позволяет отказаться от добавления тех атрибутов, которые по каким-либо причинам ранее уже были включены в кортежные модели соответствующих аспектов требований к ИС. В то же время данная операция может использоваться для включения в кортежную модель какого-либо аспекта требования к ИС атрибутов других кортежных моделей, что позволяет установить взаимосвязи между отдельными аспектами требования.

Для формализованного описания единичного морфизма $1_{Pt_{Id}}^{upd} \in 1_{Pt_{Id}}$, описывающего операцию изменения существующего описания атрибута модели паттерна Pt_{Id} , введем предположение о способе описания подобных атрибутов. Согласно этому предположению, любой атрибут любого структурного или поведенческого ППТ может быть описан выражением вида [166]:

$$at = \langle n_{at}, T_{at}, D_{at} \rangle, \quad (3.14)$$

где at – обозначение любого возможного атрибута, используемого в моделях любых возможных структурных и поведенческих ППТ;

n_{at} – имя атрибута at ;

T_{at} – тип атрибута at ;

D_{at} – домен (множество допустимых значений) атрибута at .

Данное выражение является минимально необходимой моделью атрибутов, используемых в моделях паттернов, и может быть расширено при необходимости более детального описания понятия «атрибут».

Любое изменение атрибута $at_x \in At_x$ модели структурного ППТ приведет к появлению в этой модели атрибута $at'_x \in At'_x$, для которого выполняется следующее условие [134, 165]:

$$\forall at_x, \forall at'_x \quad at_x \setminus at'_x \neq \emptyset. \quad (3.15)$$

Тогда существование единичного морфизма $1_{Pt_{Id}}^{upd} \in 1_{Pt_{Id}}$, описывающего операцию изменения существующего описания атрибута модели ППТ Pt_{Id} , определяется условием следующего вида [134, 165]:

$$\begin{aligned} \forall At_x, \forall At'_x \in AT^* \quad \exists 1_{Pt_{Id}}^{upd} \in 1_{Pt_{Id}} : Pt_{Id} \rightarrow Pt'_{Id}, \\ At_x \in Pt_{Id}, At_x \notin Pt'_{Id}, At'_x \notin Pt_{Id}, At'_x \in Pt'_{Id}. \end{aligned} \quad (3.16)$$

Любой единичный морфизм $1_{Pt_{Id}}^{upd}$, реализующий поведенческий паттерн модификации модели ППТ Pt_{Id} , может быть описан операцией вида [134, 165]:

$$\begin{aligned} 1_{Pt_{Id}}^{upd} : [(At_{rg} \setminus At_x) \cup At'_x] \oplus [(At_r \setminus At_x) \cup At'_x] \oplus \\ \oplus [(At_{asp} \setminus At_x) \cup At'_x] \oplus [(At_{acs} \setminus At_x) \cup At'_x] \oplus [(At_{proj} \setminus At_x) \cup At'_x]. \end{aligned} \quad (3.17)$$

Существование единичного морфизма $1_{Pt_{Id}}^{del} \in 1_{Pt_{Id}}$, описывающего операцию удаления неиспользуемых атрибутов из модели паттерна Pt_{Id} , определяется условием [134, 165]

$$\forall At_x \in AT^* \quad \exists 1_{Pt_{Id}}^{del} \in 1_{Pt_{Id}} : Pt_{Id} \rightarrow Pt''_{Id}, At_x \in Pt_{Id}, At_x \notin Pt''_{Id}, \quad (3.18)$$

где Pt''_{Id} – модель структурного паттерна идентификации требования к ИС,

не содержащая подмножество атрибутов At_x , введенных, но не используемых в модели Pt_{Id} .

Тогда любой единичный морфизм $1_{Pt_{Id}}^{del}$, реализующий поведенческий паттерн модификации модели ППТ Pt_{Id} , может быть описан операцией вида [134, 165]:

$$1_{Pt_{Id}}^{del} : [At_{rg} \setminus At_x] \oplus [At_r \setminus At_x] \oplus [At_{asp} \setminus At_x] \oplus [At_{acs} \setminus At_x] \oplus [At_{proj} \setminus At_x]. \quad (3.19)$$

Для модели ППТ Pt_{Ctrl} условия существования единичных морфизмов аналогичны условиям (3.12), (3.16) и (3.18), а модели этих морфизмов будут иметь, соответственно, следующий вид [134, 165]:

$$1_{Pt_{Ctrl}}^{add} : [At_{cst} \oplus At_x] \oplus [At_{dsp} \oplus At_x] \oplus [At_{dcs} \oplus At_x] \oplus [\emptyset \oplus At_x], \quad (3.20)$$

$$1_{Pt_{Ctrl}}^{upd} : [(At_{cst} \setminus At_x) \cup At'_x] \oplus [(At_{dsp} \setminus At_x) \cup At'_x] \oplus [(At_{dcs} \setminus At_x) \cup At'_x], \quad (3.21)$$

$$1_{Pt_{Ctrl}}^{del} : [At_{cst} \setminus At_x] \oplus [At_{dsp} \setminus At_x] \oplus [At_{dcs} \setminus At_x]. \quad (3.22)$$

Для модели ППТ Pt_{Op} условия существования единичных морфизмов аналогичны условиям (3.12), (3.16) и (3.18), а модели этих морфизмов будут иметь, соответственно, следующий вид [134, 165]:

$$1_{Pt_{Op}}^{add} : [At_{opr} \oplus At_x] \oplus [At_{aopsp} \oplus At_x] \oplus [At_{aopcs} \oplus At_x] \oplus [\emptyset \oplus At_x], \quad (3.23)$$

$$1_{PtOp}^{upd} : [(At_{opr} \setminus At_x) \cup At'_x] \oplus [(At_{aopsp} \setminus At_x) \cup At'_x] \oplus \oplus [(At_{aopcs} \setminus At_x) \cup At'_x], \quad (3.24)$$

$$1_{PtOp}^{del} : [At_{opr} \setminus At_x] \oplus [At_{aopsp} \setminus At_x] \oplus [At_{aopcs} \setminus At_x]. \quad (3.25)$$

Морфизмы, устанавливающие связи между моделями структурных ППТ, реализуют поведенческий паттерн интеграции описаний. Суть данного поведенческого паттерна заключается в установлении отображений, которые призваны обеспечить возможность для представителей Поставщика и Потребителя, а также для ИТ формирования и анализа требований к ИС рассматривать представления ППТ к ИС на уровне данных в виде единой атрибутивной МТ. С другой стороны, эти отображения должны минимизировать степень связности моделей отдельных структурных ППТ, чтобы снизить сложность представления ППТ к ИС на уровне данных в ходе выполнения различных операций ИТ формирования и анализа требований к ИС.

Из всех возможных морфизмов для моделей структурных ППТ Pt_{Id} , Pt_{Ctrl} и Pt_{Op} могут существовать только морфизмы $H(Pt_{Id}, Pt_{Ctrl})$, $H(Pt_{Id}, Pt_{Op})$, $H(Pt_{Op}, Pt_{Ctrl})$ и $H(Pt_{Op}, Pt_{Id})$. Существование данных морфизмов обусловлено, как показано выше, следующими условиями:

- невозможно осуществлять управление неидентифицированным требованием к ИС и осуществлять операции над таким требованием;
- в ходе выполнения отдельных операций над требованиями к ИС возможна инициация как процесса УТ, так и процесса идентификации нового требования.

Морфизм $H(Pt_{Id}, Pt_{Ctrl})$ в общем случае имеет следующий вид [134, 165]:

$$H(Pt_{Id}, Pt_{Ctrl}) : \{ At_r^{Id}, At_{asp}^{Id}, At_{acs}^{Id} \} \rightarrow \{ At_r^{Ctrl}, At_{dsp}^{Ctrl}, At_{dcs}^{Ctrl} \}, \quad (3.26)$$

где At_r^{Id} – подмножество атрибутов, идентифицирующих требование к ИС, присутствующее в модели структурного ППТ Pt_{Id} ;

At_{asp}^{Id} – подмножество атрибутов, описывающих сотрудников Поставщика как возможных авторов требований к ИС, присутствующее в модели структурного ППТ Pt_{Id} ;

At_{acs}^{Id} – подмножество атрибутов, описывающих сотрудников Потребителя как возможных авторов требований к ИС, присутствующее в модели структурного ППТ Pt_{Id} ;

At_r^{Ctrl} – подмножество атрибутов, идентифицирующих требование к ИС, присутствующее в модели структурного ППТ Pt_{Ctrl} ;

At_{dsp}^{Ctrl} – подмножество атрибутов, идентифицирующих лиц, принимающих решение об изменении текущего статуса требования к ИС со стороны Поставщика, присутствующее в модели структурного ППТ Pt_{Ctrl} ;

At_{dcs}^{Ctrl} – подмножество атрибутов, идентифицирующих лиц, принимающих решение об изменении текущего статуса требования к ИС со стороны Потребителя, присутствующее в модели структурного ППТ Pt_{Ctrl} .

Морфизм $H(Pt_{Id}, Pt_{Op})$ в общем случае имеет следующий вид [134, 165]:

$$H(Pt_{Id}, Pt_{Op}) : \{ At_r^{Id}, At_{asp}^{Id}, At_{acs}^{Id} \} \rightarrow \{ At_r^{Op}, At_{aopsp}^{Op}, At_{aopcs}^{Op} \}, \quad (3.26)$$

где At_r^{Op} – подмножество атрибутов, идентифицирующих требование к ИС, присутствующее в модели структурного ППТ Pt_{Op} ;

At_{aopsp}^{Op} – подмножество атрибутов, описывающих сотрудников Поставщика, участвующих в выполнении операции над требованием к ИС, присутствующее в модели структурного ППТ Pt_{Op} ;

At_{aopcs}^{Op} – подмножество атрибутов, описывающих сотрудников Потребителя, участвующих в выполнении операции над требованием к ИС, присутствующее в модели структурного ППТ Pt_{Op} .

Морфизм $H(Pt_{Op}, Pt_{Id})$ в общем случае имеет следующий вид [134, 165]:

$$H(Pt_{Op}, Pt_{Id}) : \{ At_{aopsp}^{Op}, At_{aopcs}^{Op} \} \rightarrow \{ At_{asp}^{Id}, At_{acs}^{Id} \}. \quad (3.28)$$

Морфизм $H(Pt_{Op}, Pt_{Ctrl})$ в общем случае имеет следующий вид:

$$H(Pt_{Op}, Pt_{Ctrl}) : \{ At_r^{Op}, At_{aopsp}^{Op}, At_{aopcs}^{Op} \} \rightarrow \{ At_r^{Ctrl}, At_{dsp}^{Ctrl}, At_{dcs}^{Ctrl} \}. \quad (3.29)$$

Все морфизмы (3.25)-(3.29) являются инъективными.

Необходимо отметить, что для случая структурных ППТ к ИС, описывающих отдельные версии требований, морфизмы, составляющие подкласс $H(D_{IS}^{Pt})$, будут иметь аналогичный вид.

Таким образом, предлагаемые в данном подразделе модели структурных ППТ (4.8)-(4.10) и (Г.1)-(Г.3) как объектов подкласса D_{IS}^{Pt} и поведенческих ППТ (3.13), (3.17), (3.19)-(3.29) как морфизмов подкласса $H(D_{IS}^{Pt})$ определяют синтаксис и семантику атрибутивных МТ к ИС на уровне данных. С точки зрения практической реализации, разработанные модели задают структуру схемы фрагмента хранилища данных, обеспечивающего хранение представлений требований к ИС и их версий на уровне данных вне зависимости от конкретных проектов создания ИС, в которых были сформулированы эти требования.

3.3 Разработка паттернов проектирования требований к информационной системе на уровне информации

Как показано в подразд. 3.1, представлением требования к ИС на уровне информации предложено считать множество возможных описаний требования в виде текстов на естественных или формальных языках, либо в виде набора целевых показателей, характеризующих степень удовлетворения требования к ИС. Однако, подобные тексты чаще всего представляют описания требований, формулируемых в ходе инициации, планирования и исполнения проекта создания конкретной ИС. Ре-использование подобных текстов чрезвычайно затруднено. Поэтому представления требований к ИС на уровне информации следует рассматривать как набор описаний этих требований, выполненных с различных точек зрения различными участниками проекта создания ИС с целью фиксации знаний этих участников о ПрО и разрабатываемой ИС.

Данное определение понятия «представление требования к ИС на уровне информации» позволяет рассматривать любое из возможных описаний требования к ИС как частный случай публикации этого требования. *Под термином «публикация требования к ИС» здесь и в дальнейшем следует понимать [134, 167, 168]:*

а) описание условия или возможности, необходимых Потребителю ИТ-услуг для решения проблемы или достижения цели, выполненное одним из допустимых в рамках методологии разработки ИС способов;

б) описание условия или возможности, которой должна обладать ИС или компонент ИС (ИТ-услуга, ИТ-сервис) с точки зрения Поставщика или Потребителя ИТ-услуг, соответствующих договору, стандарту, спецификации или другому официальному документу, выполненное одним из допустимых в рамках методологии разработки ИС способов.

Данное определение термина «публикация требования к ИС» является частным случаем пункта «в)» определения понятия «требование к ИС», данного в п. 1.2.1. Этот частный случай применяется для описания требований к

ИС, выдвигаемых в ходе выполнения текущего проекта создания ИС. Документирование требований к ИС, ре-используемых в рамках нового проекта создания ИС, может выполняться другими способами, неприменимыми для описания представлений требований к ИС на уровне информации.

Существующие методологии разработки ИС позволяют выделить следующие способы описания требований к ИС [167, 168]:

а) тексты на естественном языке, являющиеся результатами интервьюирования или анкетирования сотрудников Потребителя;

б) тексты на языке программирования, являющиеся результатами прототипирования компонентов ИС представителями Поставщика;

в) структурные ВМ, являющиеся результатами описания управляемого процесса, разрабатываемой ИС или отдельных компонентов ИС (например, модели, выполненные в нотациях IDEF0, IDEF3, DFD, ERD);

г) объектно-ориентированные ВМ, являющиеся результатами описания управляемого процесса или отдельных компонентов ИС (например, модели, выполненные в нотациях языка UML).

Два последних способа описания требований к ИС в настоящее время занимают промежуточное место между описаниями требований на естественном языке и описаниями требований в виде программно реализуемых прототипов.

В то же время рассмотрение представлений требования к ИС на уровне информации только как множества публикаций, выполненных одним или несколькими из перечисленных выше способов, является недостаточным для использования этих требований в проекте создания ИС и особенно для ре-использования этих требований в других проектах создания ИС. Для выделения из этих описаний знаний об управляемом объекте или процессе, а также о разрабатываемой ИС или ее компоненте необходим специальный механизм трансформации описаний требований к ИС к виду, пригодному для извлечения знаний.

Используем для разработки модели представления требования к ИС на уровне информации предложенную в подразд. 3.2 обобщенную модель СТ к ИС L_{IS}^{tr} . Тогда модель представления СТ к ИС на уровне информации будет

представлять собой кортеж атрибутов, который, по аналогии с моделью (3.7), также структурно разделен на две части [134, 169]:

$$M_{IS}^{tr} = \left\langle \langle M_I^{AtPt} \rangle, \langle M_I^{Atgr} \rangle \right\rangle, \quad (3.30)$$

где M_{IS}^{tr} – модель подкласса представлений СТ к ИС на уровне информации;

$\langle M_I^{AtPt} \rangle$ – кортеж элементов атрибутивной МТ к ИС, которые определяются ППТ к ИС на уровне информации и являются обязательными для требований любой группы;

$\langle M_I^{Atgr} \rangle$ – кортеж элементов атрибутивной МТ к ИС, которые определяются, исходя из индивидуальных особенностей выполнения Поставщиком и Потребителем процессов, непосредственно работающих с представлениями требований конкретной группы на уровне информации.

Как уже было отмечено выше, элементы кортежа $\langle M_I^{AtPt} \rangle$ в общем случае определяются двумя основными структурными ППТ:

- структурным паттерном публикаций требований к ИС Pt_r_publ ;
- структурным паттерном представления публикаций требований к ИС в виде, пригодном для извлечения знаний Pt_r_kn .

Модель, описывающая структурный паттерн публикаций требований к ИС Pt_r_publ , в общем случае будет иметь следующий вид [134, 169]:

$$Pt_r_publ = \langle At_r, At_{pub_t}, At_{pub}, At_{files}, At_{proj}, \langle at_r, at_{pub_t}, at_{pub}, at_{files}, at_{proj} \rangle \rangle, \quad (3.31)$$

где Pt_r_publ – модель структурного паттерна публикации требования к ИС

на уровне информации, определяющего семантику описания публикаций требований к ИС различных групп;

At_{pub_t} – подмножество атрибутов, описывающих типы публикаций требований к ИС;

At_{pub} – подмножество атрибутов, описывающих публикации требований к ИС;

At_{files} – подмножество атрибутов, описывающих файлы, в которых хранятся публикации требований к ИС;

at_{pub_st} – атрибут, идентифицирующий тип публикации конкретного сформулированного требования к ИС;

at_{pub} – атрибут, идентифицирующий публикацию конкретного сформулированного требования к ИС;

at_{files} – атрибут, идентифицирующий файл, в котором хранится публикация конкретного сформулированного требования к ИС;

$\langle at_r, at_{pub_t}, at_{pub}, at_{files}, at_{proj} \rangle$ – кортеж атрибутов, описывающий факт создания публикации конкретного сформулированного требования к ИС в проекте создания конкретной ИС.

Интерпретацией модели (3.31) в ходе разработки ИТ формирования и анализа требований к ИС является схема одной или нескольких витрин данных, предназначенных для хранения исторической информации о публикациях требований, выполненных в рамках проектов создания различных ИС.

В случае использования модели (3.31) для идентификации версии требования к ИС данная модель принимает вид, описанный выражением (Г.5) Приложения Г.

Для описания представления публикации требования к ИС в виде, пригодном для извлечения знаний, следует, прежде всего, определить основной подход к организации этих знаний. Для этого необходимо установить возможные виды представлений требований к ИС на уровне информации.

Как показано в подразд. 2.2, формирование требований к ИС Поставщиком

и Потребителем завершается созданием групп ФТ и нефункциональных требований к ИТ-услугам (на стадии макропроектирования ИС) или групп ФТ и нефункциональных требований к ИТ-сервисам (на стадии микропроектирования ИС). Поэтому общий подход к рассмотрению описаний требований и ППТ должен определяться необходимостью использования унифицированных представлений ФТ к ИТ-услугам и реализующим эти услуги ИТ-сервисам для решения задач синтеза архитектур ИС и ее обеспечивающей части.

В качестве такого общего подхода рассмотренная в подразд. 2.3 концепция представления требований к ИС использует процессный подход. Данный подход позволяет рассматривать любое (функциональное или нефункциональное) требование к любому элементу ИС как часть описания процесса, визуальная модель которого в нотации Гейна-Сарсона показана на рис. 3.3 [169].

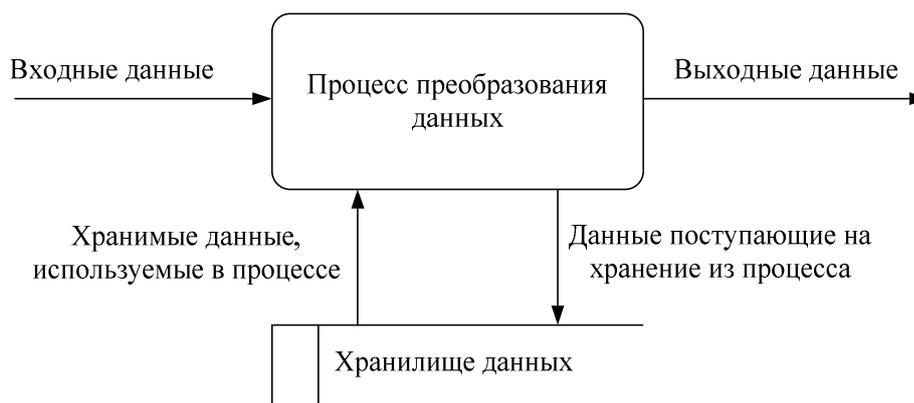


Рисунок 3.3 – Визуальная модель процесса преобразования данных как типового элемента информационной системы

Предлагаемое обобщенное представление элемента ИС позволяет уточнить онтологию понятия «требование к элементу ИС» путем формулирования следующих аксиоматических утверждений и следствий из них [134, 135, 167-170].

Утверждение 1. Каждое требование к процессу описывает только один требующий автоматизации процесс предприятия или только одну разрабатываемую функцию ИС. По мере необходимости это описание может быть детализировано совокупностью других требований к процессам.

Утверждение 2. Требования к границам системы описывают пределы ИС, за которые нельзя выходить при выявлении требований к процессам.

Следствие из Утверждения 2. Каждое требование к процессу будет представлять собой требование к границам системы для совокупности требований к процессам, детализирующим описание этого требования.

Утверждение 3. Требования к входным, хранимым и выходным данным описывают структуры данных с устанавливаемой разработчиком степенью детализации и позволяют уточнить взаимодействие процессов, требующих автоматизации.

Следствие из Утверждения 3. Требования к входным, выходным и хранимым данным являются звеньями, согласующими описания различных требований к процессам.

Данные утверждения позволяют рассматривать понятие «требование к элементу ИС» как категориальное, включающее в себя следующие классы требований [169]:

а) «Требование к входным данным», устанавливающий особенности представления потоков, инициирующих выполнение процесса;

б) «Требование к выходным данным», устанавливающий особенности представления результатов выполнения процесса;

в) «Требование к хранимым данным», устанавливающий особенности представления потоков, доступных для использования в ходе периодического выполнения процесса;

г) «Требование к процессу преобразования данных», который устанавливает особенности преобразований требований к входным данным в требования к хранимым данным, требований к входным данным в требования к выходным данным, а также требований к хранимым данным в требования к выходным данным в ходе выполнения процесса.

В свою очередь, класс «Требование к хранимым данным» можно разделить на два следующих подкласса:

а) «Требование к хранимым данным, используемым в процессе»;

б) «Требование к данным, поступающим на хранение из процесса».

Контекстная ДК, описывающая понятие «Требование к элементу ИС»,

приведена на рис. 3.4 [134, 169].

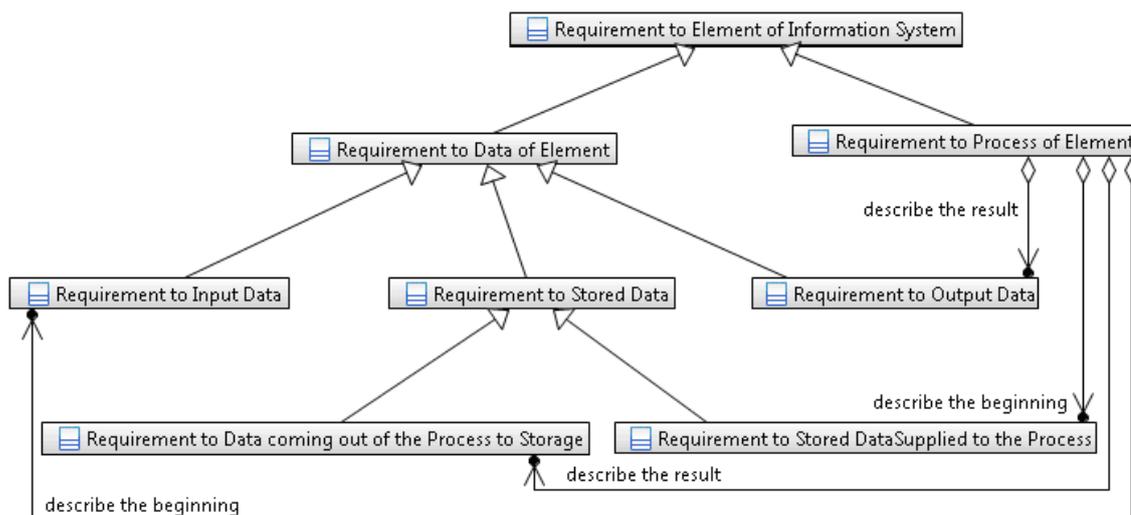


Рисунок 3.4 – Контекстная диаграмма классов, описывающая понятие «требование к элементу информационной системы»

Предлагаемые классы, описывающие требования к элементам ИС, полностью соответствуют как главной цели деятельности ИС (см. табл. 2.1), так и процессному подходу к описанию требований к ИС.

Рассмотренный подход к определению понятия «требование к элементу ИС» позволяет рассматривать модель структурного паттерна представления публикации требований к ИС в виде, пригодном для извлечения знаний, Pt_r_kn как множество, состоящее из следующих элементов [169]:

а) модель структурного паттерна представления структур данных, присутствующих в публикации требования к ИС, $Pt_r_kn_dst$;

б) модель структурного паттерна представления внешних по отношению к процессу объектов, которые непосредственно участвуют в выполнении процесса и присутствуют в публикации требования к ИС, $Pt_r_kn_eobj$;

в) модель структурного паттерна представления процессов, присутствующих в публикации требования к ИС, $Pt_r_kn_proc$.

Модель структурного паттерна представления структур данных, присут-

ствующих в публикации требования к ИС, $Pt_{r_kn_dst}$ в общем случае будет иметь следующий вид [134, 169]:

$$Pt_{r_kn_dst} = \langle At_{pub_t}, At_{pub}, At_{str_t}, At_{str}, At_{str_m_t}, At_{str_m}, At_{attr_t}, At_{attr}, \langle at_{pub_t}, at_{pub}, at_{str_t}, at_{ztr}, at_{str_m_t}, at_{str_m}, at_{attr_t}, at_{attr} \rangle \rangle, \quad (3.32)$$

где At_{str_t} – подмножество атрибутов, описывающих типы структур данных (сущности, классы), присутствующих в публикации требования к ИС;

At_{str} – подмножество атрибутов, описывающих структуру данных, присутствующую в публикации требования к ИС;

$At_{str_m_t}$ – подмножество атрибутов, описывающих типы элементов структур данных, присутствующих в публикации требования к ИС;

At_{str_m} – подмножество атрибутов, описывающих элемент структуры данных, присутствующий в публикации требования к ИС;

At_{attr_t} – подмножество атрибутов, описывающих типы атрибутов структур данных, присутствующих в публикации требования к ИС;

At_{attr} – подмножество атрибутов, описывающих атрибут структуры данных, присутствующий в публикации требования к ИС;

at_{str_t} – атрибут, идентифицирующий тип структуры данных, присутствующей в публикации конкретного СТ к ИС;

at_{str} – атрибут, идентифицирующий структуру данных, присутствующую в публикации конкретного СТ к ИС;

$at_{str_m_t}$ – атрибут, идентифицирующий тип элемента структуры данных, присутствующего в публикации конкретного СТ к ИС;

at_{str_m} – атрибут, идентифицирующий элемент структуры данных, присутствующий в публикации конкретного СТ к ИС;

at_{attr_t} – атрибут, идентифицирующий тип атрибута структуры данных, присутствующего в публикации конкретного СТ к ИС;

at_{attr} – атрибут, идентифицирующий атрибут структуры данных, присутствующий в публикации конкретного СТ к ИС.

Интерпретацией модели (3.32) в ходе разработки ИТ формирования и анализа требований к ИС является схема одной или нескольких витрин данных, предназначенных для хранения исторической информации о структурах данных, присутствовавших в публикациях требований, выполненных в рамках проектов создания различных ИС.

Модель структурного паттерна представления внешних по отношению к процессу объектов, которые непосредственно участвуют в выполнении процесса и присутствуют в публикации требования к ИС $Pt_{r_kn_eobj}$, в общем случае будет иметь следующий вид [134, 169]:

$$Pt_{r_kn_eobj} = \langle At_{pub_t}, At_{pub}, At_{eobj_t}, At_{eobj}, \langle at_{pub_t}, at_{pub}, at_{eobj_t}, at_{eobj} \rangle \rangle, \quad (3.33)$$

где At_{eobj_t} – подмножество атрибутов, описывающих типы внешних по отношению к процессу объектов, которые непосредственно участвуют в выполнении процесса и присутствуют в публикации требования к ИС;

At_{eobj} – подмножество атрибутов, описывающих внешний по отношению к процессу объект, который непосредственно участвует в выполнении процесса и присутствует в публикации требования к ИС;

at_{eobj_t} – атрибут, идентифицирующий тип внешнего по отношению к процессу объекта, который непосредственно участвует в выполнении процесса и присутствует в публикации конкретного СТ к ИС;

at_{eobj} – атрибут, идентифицирующий внешний по отношению к процессу объект, который непосредственно участвует в выполнении процесса

и присутствует в публикации конкретного СТ к ИС.

Интерпретацией модели (3.33) в ходе разработки ИТ формирования и анализа требований к ИС является схема одной или нескольких витрин данных, предназначенных для хранения исторической информации о внешних по отношению к процессу объектах, которые непосредственно участвуют в выполнении процесса и присутствуют в публикациях требований, выполненных в рамках проектов создания различных ИС.

Модель структурного паттерна представления процессов, присутствующих в публикации требования к ИС $P_{tr_kn_proc}$, в общем случае будет иметь следующий вид [134, 169]:

$$P_{tr_kn_proc} = \langle At_{pub_t}, At_{pub}, At_{proc}, At_{inst_t}, At_{inst}, \langle at_{pub_t}, at_{pub}, at_{proc}, at_{inst_t}, at_{inst} \rangle \rangle, \quad (3.34)$$

где At_{proc} – подмножество атрибутов, описывающих процесс, присутствующих в публикации требования к ИС;

At_{inst_t} – подмножество атрибутов, описывающих типы элементов (внешних по отношению к процессу объектов или структур данных), связанных с процессом в публикации требования к ИС;

At_{eobj} – подмножество атрибутов, описывающих элементы, связанные с процессом в публикации требования к ИС;

at_{proc} – атрибут, идентифицирующий процесс, присутствующий в публикации конкретного СТ к ИС;

at_{inst_t} – атрибут, идентифицирующий тип элемента, связанного с процессом в публикации конкретного СТ к ИС;

at_{inst} – атрибут, идентифицирующий элемент, связанный с процессом в публикации конкретного СТ к ИС.

Интерпретацией модели (3.34) в ходе разработки ИТ формирования и анализа

требований к ИС является схема одной или нескольких витрин данных, предназначенных для хранения исторической информации о процессах, присутствующих в публикациях требований, выполненных в рамках проектов создания различных ИС.

Использование моделей (3.32)-(3.34) для описания представления публикации версий требований к ИС в виде, пригодном для извлечения знаний, не требует изменения вида этих моделей.

Таким образом, для случая использования в качестве элементарных описаний создаваемой системы отдельных требований к ИС подкласс объектов I_{IS}^{Pt} модели (2.40) следует рассматривать как множество структурных ППТ к ИС на уровне информации, устанавливающих конкретный вид элемента $\langle M_I^{AtPt} \rangle$ модели (3.30). Это множество паттернов в общем случае будет иметь следующий вид [134, 169]:

$$\begin{aligned}
 I_{IS}^{Pt} &= \{ Pt_r_publ, Pt_r_kn_dst, Pt_r_kn_eobj, Pt_r_kn_proc \} = \\
 &= \{ \langle At_r, At_pub_t, At_pub, At_files, At_proj, \langle at_r, at_pub_t, at_pub, at_files, \\
 &\quad at_proj \rangle \rangle, \langle At_pub_t, At_pub, At_str_t, At_str, At_str_m_t, At_str_m, \\
 &\quad At_attr_t, At_attr, \langle at_pub_t, at_pub, at_str_t, at_ztr, at_str_m_t, at_str_m, \\
 &\quad at_attr_t, at_attr \rangle \rangle, \langle At_pub_t, At_pub, At_eobj_t, At_eobj, \langle at_pub_t, at_pub, \\
 &\quad at_eobj_t, at_eobj \rangle \rangle, \langle At_pub_t, At_pub, At_proc, At_inst_t, At_inst, \\
 &\quad \langle at_pub_t, at_pub, at_proc, at_inst_t, at_inst \rangle \rangle \}.
 \end{aligned} \tag{3.35}$$

Для случая использования в качестве элементарных описаний создаваемой системы отдельных версий требований к ИС подкласс объектов I_{IS}^{Pt} модели (3.39) следует рассматривать как выражение (Г.6) Приложения Г.

Подкласс морфизмов $H(I_{IS}^{Pt})$, по аналогии с рассмотренным в подразд. 3.2 подклассом морфизмов $H(D_{IS}^{Pt})$, состоит из следующих видов морфизмов:

а) единичные морфизмы $1_{Pt_r_publ}$, $1_{Pt_r_kn_dst}$, $1_{Pt_r_kn_eobj}$, $1_{Pt_r_kn_proc}$, отображающие структурные ППТ Pt_r_publ , $Pt_r_kn_dst$, $Pt_r_kn_eobj$ и

$Pt_r_kn_proc$, соответственно, самих в себя;

б) морфизмы, устанавливающие связи между структурными ППТ Pt_r_publ , $Pt_r_kn_dst$, $Pt_r_kn_eobj$ и $Pt_r_kn_proc$.

Для модели ППТ Pt_r_publ условия существования единичных морфизмов аналогичны условиям (3.12), (3.16) и (3.18), а модели этих морфизмов будут иметь, соответственно, следующий вид [134, 169]:

$$\begin{aligned} 1_{Pt_r_publ}^{add} : [At_r \oplus At_x] \oplus [At_{pub_t} \oplus At_x] \oplus [At_{pub} \oplus at_x] \oplus \\ \oplus [At_{files} \oplus At_x] \oplus [At_{proj} \oplus At_x] \oplus [\emptyset \oplus At_x], \end{aligned} \quad (3.26)$$

$$\begin{aligned} 1_{Pt_r_publ}^{upd} : [(At_r \setminus At_x) \cup At'_x] \oplus [(At_{pub_t} \setminus at_x) \cup At'_x] \oplus \\ \oplus [(At_{pub} \setminus At_x) \cup At'_x] \oplus [(At_{files} \setminus At_x) \cup At'_x] \oplus [(At_{proj} \setminus At_x) \cup At'_x], \end{aligned} \quad (3.37)$$

$$\begin{aligned} 1_{Pt_r_publ}^{del} : [At_r \setminus At_x] \oplus [At_{pub_t} \setminus At_x] \oplus [At_{pub} \setminus At_x] \oplus \\ \oplus [At_{files} \setminus At_x] \oplus [At_{proj} \setminus At_x]. \end{aligned} \quad (3.38)$$

Для модели ППТ $Pt_r_kn_dst$ условия существования единичных морфизмов аналогичны условиям (3.12), (3.16) и (3.18), а модели этих морфизмов будут иметь, соответственно, следующий вид [134, 169]:

$$\begin{aligned} 1_{Pt_r_kn_dst}^{add} : [At_{pub_t} \oplus At_x] \oplus [At_{pub} \oplus At_x] \oplus [At_{str_t} \oplus At_x] \oplus \\ \oplus [At_{str} \oplus At_x] \oplus [At_{str_m_t} \oplus At_x] \oplus [At_{str_m} \oplus At_x] \oplus \\ \oplus [At_{attr_t} \oplus At_x] \oplus [At_{attr} \oplus At_x] \oplus [\emptyset \oplus At_x], \end{aligned} \quad (3.39)$$

$$\begin{aligned}
1_{Pt_r_kn_dst}^{upd} : & [(At_{pub_t} \setminus At_x) \cup At'_x] \oplus [(At_{pub} \setminus at_x) \cup At'_x] \oplus \\
& \oplus [(At_{str_t} \setminus At_x) \cup At'_x] \oplus [(At_{str} \setminus At_x) \cup At'_x] \oplus \\
& \oplus [(At_{str_m_t} \setminus At_x) \cup At'_x] \oplus [(At_{str_m} \setminus At_x) \cup At'_x] \oplus \\
& \oplus [(At_{attr_t} \setminus At_x) \cup At'_x] \oplus [(At_{attr} \setminus At_x) \cup At'_x],
\end{aligned} \tag{3.40}$$

$$\begin{aligned}
1_{Pt_r_kn_dst}^{del} : & [At_{pub_t} \setminus At_x] \oplus [At_{pub} \setminus At_x] \oplus [At_{str_t} \setminus At_x] \oplus \\
& \oplus [At_{str} \setminus At_x] \oplus [At_{str_m_t} \setminus At_x] \oplus [At_{str_m} \setminus At_x] \oplus \\
& \oplus [At_{attr_t} \setminus At_x] \oplus [At_{attr} \setminus At_x].
\end{aligned} \tag{3.41}$$

Для модели ППТ $Pt_r_kn_eobj$ условия существования единичных морфизмов аналогичны условиям (3.12), (3.16) и (3.18), а модели этих морфизмов будут иметь, соответственно, следующий вид [134, 169]:

$$\begin{aligned}
1_{Pt_r_kn_eobj}^{add} : & [At_{pub_t} \oplus At_x] \oplus [At_{pub} \oplus At_x] \oplus [At_{eobj_t} \oplus At_x] \oplus \\
& \oplus [At_{eobj} \oplus At_x] \oplus [\emptyset \oplus At_x],
\end{aligned} \tag{3.42}$$

$$\begin{aligned}
1_{Pt_r_kn_eobj}^{upd} : & [(At_{pub_t} \setminus At_x) \cup At'_x] \oplus [(At_{pub} \setminus at_x) \cup At'_x] \oplus \\
& \oplus [(At_{eobj_t} \setminus At_x) \cup At'_x] \oplus [(At_{eobj} \setminus At_x) \cup At'_x],
\end{aligned} \tag{3.43}$$

$$\begin{aligned}
1_{Pt_r_kn_eobj}^{del} : & [At_{pub_t} \setminus At_x] \oplus [At_{pub} \setminus At_x] \oplus [At_{eobj_t} \setminus At_x] \oplus \\
& \oplus [At_{eobj} \setminus At_x].
\end{aligned} \tag{3.44}$$

Для модели ППТ $Pt_r_kn_proc$ условия существования единичных морфизмов аналогичны условиям (3.12), (3.16) и (3.18), а модели этих морфизмов будут иметь, соответственно, следующий вид [134, 169]:

$$\begin{aligned}
1_{Pt_r_kn_proc}^{add} : [At_{pub_t} \oplus At_x] \oplus [At_{pub} \oplus At_x] \oplus [At_{proc} \oplus At_x] \\
\oplus [At_{inst_t} \oplus At_x] \oplus [At_{inst} \oplus At_x] \oplus [\emptyset \oplus At_x],
\end{aligned} \tag{3.45}$$

$$\begin{aligned}
1_{Pt_r_kn_proc}^{upd} : [(At_{pub_t} \setminus At_x) \cup At'_x] \oplus [(At_{pub} \setminus At_x) \cup At'_x] \oplus \\
\oplus [(At_{proc} \setminus At_x) \cup At'_x] \oplus [(At_{inst_t} \setminus At_x) \cup At'_x] \oplus \\
\oplus [(At_{inst} \setminus At_x) \cup At'_x],
\end{aligned} \tag{3.46}$$

$$\begin{aligned}
1_{Pt_r_kn_proc}^{del} : [At_{pub_t} \setminus At_x] \oplus [At_{pub} \setminus At_x] \oplus [At_{proc} \setminus At_x] \oplus \\
\oplus [At_{inst_t} \setminus At_x] \oplus [At_{inst} \setminus At_x].
\end{aligned} \tag{3.47}$$

Из всех возможных морфизмов, устанавливающих связи между моделями структурных паттернов Pt_r_publ , $Pt_r_kn_dst$, $Pt_r_kn_eobj$ и $Pt_r_kn_proc$, могут существовать только морфизмы $H(Pt_r_publ, Pt_r_kn_dst)$, $H(Pt_r_publ, Pt_r_kn_eobj)$, $H(Pt_r_publ, Pt_r_kn_proc)$, $H(Pt_r_kn_dst, Pt_r_kn_proc)$, $H(Pt_r_kn_eobj, Pt_r_kn_proc)$. Существование данных морфизмов обусловлено следующими ограничениями:

- невозможно появление описаний ранее неизвестных в рамках ИТ формирования и анализа требований к ИС структур данных, внешних объектов и процессов без предварительного появления публикации требования к ИС, содержащей эти структуры данных, внешние объекты и процессы;

- невозможно подробное описание конкретного процесса без предварительного описания структур данных и внешних объектов, характерных только для данного процесса.

Отдельным вопросом, требующим дальнейшего исследования, является вопрос существования морфизмов $H(Pt_r_kn_proc, Pt_r_kn_dst)$, $H(Pt_r_kn_proc, Pt_r_kn_eobj)$. Существование этих морфизмов может быть объяснено наличием какого-либо метода автоматического формирования опи-

саний структур данных и внешних объектов на основе описания процесса, присутствующего в публикациях требований, сформулированных или реиспользуемых в ходе создания конкретной ИС. Последний случай особенно интересен для решения проблемы адаптации типовой ИС для различных объектов или процессов автоматизации, имеющих общее назначение.

Морфизм $H(Pt_r_publ, Pt_r_kn_dst)$ в общем случае имеет вид [134, 169]:

$$\begin{aligned} H(Pt_r_publ, Pt_r_kn_dst) : \{ At_{pub_t}^{r_publ}, At_{pub}^{r_publ} \} \rightarrow \\ \rightarrow \{ At_{pub_t}^{r_kn_dst}, At_{pub}^{r_kn_dst} \}, \end{aligned} \quad (3.48)$$

где $At_{pub_t}^{r_publ}$ – подмножество атрибутов, описывающих тип публикации требования к ИС в модели структурного ППТ Pt_r_publ ;

$At_{pub}^{r_publ}$ – подмножество атрибутов, описывающих публикацию требования к ИС в модели структурного ППТ Pt_r_publ ;

$At_{pub_t}^{r_kn_dst}$ – подмножество атрибутов, описывающих тип публикации требования к ИС в модели структурного ППТ $Pt_r_kn_dst$;

$At_{pub}^{r_kn_dst}$ – подмножество атрибутов, описывающих публикацию требования к ИС в модели структурного ППТ $Pt_r_kn_dst$.

Морфизм $H(Pt_r_publ, Pt_r_kn_eobj)$ в общем случае имеет вид [134, 169]:

$$\begin{aligned} H(Pt_r_publ, Pt_r_kn_eobj) : \{ At_{pub_t}^{r_publ}, At_{pub}^{r_publ} \} \rightarrow \\ \rightarrow \{ At_{pub_t}^{r_kn_eobj}, At_{pub}^{r_kn_eobj} \}, \end{aligned} \quad (3.49)$$

где $At_{pub_t}^{r_kn_eobj}$ – подмножество атрибутов, описывающих тип публикации тре-

бования к ИС в модели структурного ППТ $Pt_r_kn_eobj$;

$At_{pub}^{r_kn_eobj}$ – подмножество атрибутов, описывающих публикацию требования к ИС в модели структурного ППТ $Pt_r_kn_eobj$.

Морфизм $H(Pt_r_publ, Pt_r_kn_proc)$ в общем случае имеет вид [134, 169]:

$$\begin{aligned} H(Pt_r_publ, Pt_r_kn_proc) : \{ At_{pub_t}^{r_publ}, At_{pub}^{r_publ} \} \rightarrow \\ \rightarrow \{ At_{pub_t}^{r_kn_proc}, At_{pub}^{r_kn_proc} \}, \end{aligned} \quad (3.50)$$

где $At_{pub_t}^{r_kn_proc}$ – подмножество атрибутов, описывающих тип публикации требования к ИС в модели структурного ППТ $Pt_r_kn_proc$;

$At_{pub}^{r_kn_proc}$ – подмножество атрибутов, описывающих публикацию требования к ИС в модели структурного ППТ $Pt_r_kn_proc$.

Морфизм $H(Pt_r_kn_dst, Pt_r_kn_proc)$ в общем случае имеет вид [134, 169]:

$$\begin{aligned} H(Pt_r_kn_dst, Pt_r_kn_proc) : \\ : \left\{ \begin{aligned} \{ At_{str_t}^{r_kn_dst}, At_{str_m_t}^{r_kn_dst}, At_{attr_t}^{r_kn_dst} \} \rightarrow At_{inst_t}^{r_kn_proc} \\ \{ At_{str}^{r_kn_dst}, At_{str_m}^{r_kn_dst}, At_{attr}^{r_kn_dst} \} \rightarrow At_{inst}^{r_kn_proc} \end{aligned} \right. \end{aligned} \quad (3.51)$$

где $At_{str_t}^{r_kn_dst}$ – подмножество атрибутов, описывающих тип структуры данных в модели структурного ППТ $Pt_r_kn_dst$;

$At_{str_m_t}^{r_kn_dst}$ – подмножество атрибутов, описывающих тип элемента структуры данных в модели структурного ППТ $Pt_r_kn_dst$;

$At_{attr_t}^{r_kn_dst}$ – подмножество атрибутов, описывающих тип атрибута структу-

ры данных в модели структурного ППТ $Pt_r_kn_dst$;

$At_{inst_t}^{r_kn_proc}$ – подмножество атрибутов, описывающих тип элемента, связанного с процессом, в модели структурного ППТ $Pt_r_kn_proc$;

$At_{str}^{r_kn_dst}$ – подмножество атрибутов, описывающих структуру данных в модели структурного ППТ $Pt_r_kn_dst$;

$At_{str_m}^{r_kn_dst}$ – подмножество атрибутов, описывающих элемент структуры данных в модели структурного ППТ $Pt_r_kn_dst$;

$At_{attr}^{r_kn_dst}$ – подмножество атрибутов, описывающих атрибут структуры данных в модели структурного ППТ $Pt_r_kn_dst$;

$At_{inst}^{r_kn_proc}$ – подмножество атрибутов, описывающих элемент, связанный с процессом, в модели структурного ППТ $Pt_r_kn_proc$.

Морфизм $H(Pt_r_kn_eobj, Pt_r_kn_proc)$ в общем случае имеет вид [134, 169]:

$$H(Pt_r_kn_dst, Pt_r_kn_proc) : \begin{cases} At_{eobj_t}^{r_kn_eobj} \rightarrow At_{inst_t}^{r_kn_proc} \\ At_{eobj}^{r_kn_eobj} \rightarrow At_{inst}^{r_kn_proc} \end{cases} \quad (3.52)$$

где $At_{eobj_t}^{r_kn_eobj}$ – подмножество атрибутов, описывающих тип внешнего по отношению к процессу объекта в модели структурного ППТ $Pt_r_kn_eobj$;

$At_{eobj}^{r_kn_eobj}$ – подмножество атрибутов, описывающих внешний по отношению к процессу объект в модели структурного ППТ $Pt_r_kn_eobj$.

Морфизмы (3.48)-(3.50) являются инъективными.

Необходимо отметить, что для случая структурных ППТ к ИС, описывающих отдельные версии требований, морфизмы, составляющие подкласс

$H(I_{IS}^{Pt})$, будут иметь аналогичный вид.

Таким образом, предлагаемые в данном подразделе модели структурных ППТ (3.31)-(3.34), (Г.5) как объектов подкласса I_{IS}^{Pt} и поведенческих ППТ (3.36)-(3.52) как морфизмов подкласса $H(I_{IS}^{Pt})$ определяет синтаксис и семантику атрибутивных МТ к ИС на уровне информации. С точки зрения практической реализации, разработанные модели задают структуру схемы фрагмента хранилища данных, обеспечивающего хранение представлений требований к ИС и их версий на уровне информации вне зависимости от конкретных проектов создания ИС, в которых были сформулированы эти требования.

3.4 Разработка паттернов проектирования требований к информационной системе на уровне знаний

В подразд. 3.1 представлением требования к ИС на уровне знаний предлагается считать онтологии элемента управляемого объекта или процесса, элемента ИС или ИС в целом, к которым выдвигается требование. В то же время стремление ре-использования требований к ИС обуславливает необходимость дополнительного выделения онтологий элементов ИС или ИС в целом, реализованных в выполненных ранее проектах создания ИС. В целом предлагается рассматривать следующие виды онтологий:

а) онтологии ПрО, которые представляют собой знания об автоматизируемых объектах или БП, добытые в ходе выявления и анализа требований к ИС;

б) онтологии реализованных требований к ИС, которые представляют собой знания о структурах данных и процессах ИС, ИТ-услуг и ИТ-сервисов, созданных в рамках предыдущих проектов;

в) онтологии требований к создаваемой ИС, которые представляют собой знания о структурах данных и процессах ИС, ИТ-услуг и ИТ-сервисов, выделяемые в хо-

де формирования и анализа требований в рамках текущего проекта создания ИС.

Формирование этих онтологий предполагается проводить на основе фреймовой модели знаний. Применение данной модели обусловлено следующими соображениями:

- использование фреймовой модели знаний позволяет применять единый математический аппарат для описания знаний о ПрО для формального представления требований к ИС, а также для описания знаний, реализованных в элементах ИС в виде моделей ПО данной системы;

- использование фреймовой модели позволяет реализовать взаимно-однозначное отображение представлений требований к создаваемой ИС на уровне знаний в элементы ПО этой ИС.

Поскольку большинство современных СУБД основано на реляционной модели данных, применение фреймовой модели знаний позволяет осуществить в результате решения задачи объектно-реляционного отображения последующее взаимно-однозначное отображение требований к ПрО и элементов ПО в элементы ИО создаваемой ИС [171, 172].

Фрейм является структурой данных для представления стереотипной ситуации. В методологии объектно-ориентированного программирования (ООП) данное понятие соответствует классу [173].

Как правило, фреймовые модели ПрО представляются в виде сети фреймов, состоящей из узлов и различных связей между ними [134, 173]:

$$M = \langle FR, C, G \rangle, \quad (3.52)$$

где $FR = \{fr_1, \dots, fr_h\}$ – множество информационных единиц (фреймов);

$C = \{C_1, C_2, \dots, C_n\}$ – множество связей и отношений между информационными единицами (иерархические, ссылочные и т.д.);

G – множество отображений, которые задают связи из заданного множества $\{C_1, C_2, \dots, C_n\}$ (как иерархические связи наследования, так и горизонтальные связи ассоциации фреймов) между информационными едини-

цами, входящими во множество FR ; $G_i \in G = \langle fr_{i1}, fr_{i2}, C_i \rangle$ [174].

Фреймы подразделяются на фреймы-прототипы (им соответствуют классы в ООП и таблицы в БД) и фреймы-экземпляры (им соответствуют экземпляры классов – объекты и записи в таблицах БД). По характеру отношений фреймы классифицируются следующим образом: «субфреймы, фреймы и суперфреймы – это иерархически упорядоченные элементы, образующие системы фреймов». В технологии ООП существуют также иерархические и ссылочные связи, а понятиям «субфрейм» и «суперфрейм» соответствуют субклассы (классы-потомки) и суперклассы (классы-родители). Аналогично фреймовой модели, в виде сети можно представить иерархию классов (ДК в UML), в которой классы-родители определяют набор полей и функциональности, свойственный всем классам-потомкам [35, 173, 175-177].

Фрейм $fr \in FR$ можно описать структурированным множеством, имеющим следующий вид [134, 175]:

$$fr = \{n, [(ns_1, vs_1, ps_1), (ns_2, vs_2, ps_2), \dots, (ns_k, vs_k, ps_k)]\}, \quad (3.54)$$

где n – имя фрейма;

(ns, vs, ps) – слот фрейма;

k – количество слотов фрейма;

ns_i – имя слота, $i = \overline{1, k}$;

vs_i – значение слота, $i = \overline{1, k}$;

ps_i – имя присоединенной процедуры, $i = \overline{1, k}$.

В качестве значения слота «имя присоединенной процедуры» в фреймах используется подпрограмма процедурного типа. Присоединённым процедурам в ООП соответствуют методы классов, в реляционных БД им соответствуют связанные с таблицами триггеры, процедуры и функции [178].

Однако фреймовая модель в классическом виде, представленном выражением (3.54), не вполне соответствует особенностям представления знаний

о ПрО, и особенно о создаваемой ИС и ее элементах. Подобные особенности определяются, главным образом, теми парадигмами проектирования систем, которые помещаются Поставщиком в основу подавляющего большинства проектных решений ИС в целом, отдельных ИТ-услуг и ИТ-сервисов. В качестве таких парадигм обычно указываются структурный и объектно-ориентированный подходы. Данные подходы, в свою очередь, определяют необходимость модернизации фреймовой модели представления знаний о ПрО, ИС, ИТ-услугах и ИТ-сервисах таким образом, чтобы эти знания можно было впоследствии преобразовать в паттерны и конкретные проектные решения по видам обеспечений создаваемой ИС.

Прежде всего, предлагается расширить базовое понятие «фрейм» путем выделения как отдельного элемента фрейма совокупности всех методов (присоединенных процедур) $Mt = \{mt_1, \dots, mt_z\}$, связанных с фреймом в целом, а не с конкретными слотами. Кроме того, предлагается расширить базовое понятие «фрейм» введением дополнительного понятия «интерфейс фрейма», соответствующего понятию «интерфейс класса» в методологии ООП [134, 178].

Следует отметить, что в некоторых языках программирования (например, в языке Object Pascal) поддерживается возможность определения свойств в интерфейсах классов. В других языках программирования (например, в языках Java, C++ и ряде других) свойства могут быть объявлены путем задания методов *GetPropertyName* и *SetPropertyName*, осуществляющих доступ к полям класса. Интерфейсы, как и классы, могут содержать ссылочные свойства и наследоваться, образуя иерархии [35, 134, 177, 179]. В реляционных БД нет понятия, полностью соответствующего термину «интерфейс класса». Наиболее близким к нему с точки зрения участия в организации взаимодействия между различными компонентами ИС является термин «представление».

Сказанное выше позволяет определить термин «интерфейс фрейма» как декларативное объявление множества свойств и методов без их подробного описания (в том числе, без подробного описания присоединенных процедур). Предполагается, что каждый интерфейс фрейма описывает отдельную точку

зрения на фрейм или их подмножество, причем данная точка зрения может не воспринимать эти фрейм или подмножество фреймов полностью. Данное определение согласуется с определением понятия «архитектурная точка зрения», приведенным в подразд. 1.3.

Согласно предложенному определению, формализованное описание интерфейса фрейма if будет представлять собой структурированное множество, имеющее следующий вид [134, 178]:

$$if = \langle g, \{ns_{if_1}, \dots, ns_{if_n}\}, \{nm_1, \dots, nm_s\} \rangle, \quad (3.55)$$

где g – глобально уникальный идентификатор (Globally Unique Identifier – GUID);

$\{ns_{if_1}, \dots, ns_{if_n}\}$ – множество декларативных объявлений слотов в интерфейсе фрейма;

$\{nm_1, \dots, nm_s\}$ – множество декларативных объявлений методов в интерфейсе фрейма.

Следует отметить, что любому декларативно объявленному имени слота интерфейса фрейма можно поставить в соответствие множество значений как отдельного фрейма $fr \in Fr$, так и отдельного слота $ns_i \in fr$. В дальнейшем фрейм, участвующий в формировании интерфейса if , будем называть порождающим фреймом.

Процесс формирования интерфейса if из подмножества порождающих фреймов $FR_{par} \subset FR$ можно описать отображением $F_{FR_{par}}^{if} : FR_{par} \rightarrow if$.

Данное отображение биективно, поскольку каждый конкретный элемент интерфейса if может иметь в качестве родителя только один фрейм или слот фрейма из подмножества FR_{par} .

Тогда формализованное описание понятия «фрейма» (3.54) с учетом предлагаемых модификаций примет следующий вид [134, 178]:

$$fr = \{n, [(ns_1, vs_1, ps_1), \dots, (ns_k, vs_k, ps_k)], \{if_1, \dots, if_n\}, \{mt_1, \dots, mt_z\}\}, \quad (3.56)$$

где $\{if_1, \dots, if_n\}$ – множество интерфейсов, используемых фреймом fr (может быть пустым).

Использование модифицированного формализованного описания фрейма (3.56) позволяет утверждать, что сеть фреймов (3.53) будет включать не только знания о структурах данных, представляемые в виде фреймов, но и знания о процессах, в ходе которых будет осуществляться взаимодействие этих структур. Такие знания представляются в сети в виде интерфейсов и методов фреймов. Кроме того, использование модифицированного описания фрейма позволяет формализовать не только описания знаний о ПрО, ИО и ПО создаваемой ИС, но и описания их взаимно-однозначных отображений.

Используем для разработки модели сети фреймов как представления требования к ИС на уровне знаний предложенную в подразд. 3.2 обобщенную модель СТ к ИС L_{IS}^{tr} . Тогда модель представления требований к ИС на уровне знаний будет представлять собой кортеж атрибутов, который, по аналогии с моделью (3.7), также структурно разделен на две части [134]:

$$M_{K_{IS}^{tr}} = \left\langle \langle M_K^{AtPt} \rangle, \langle M_K^{Atgr} \rangle \right\rangle, \quad (3.57)$$

где $M_{K_{IS}^{tr}}$ – модель подкласса представлений СТ к ИС на уровне знаний;

$\langle M_K^{AtPt} \rangle$ – кортеж элементов атрибутивной МТ к ИС, которые определяются ППТ к ИС на уровне знаний и являются обязательными для требований любой группы;

$\langle M_K^{Atgr} \rangle$ – кортеж элементов атрибутивной МТ к ИС, которые определяются, исходя из индивидуальных особенностей выполнения Поставщиком и Потребителем процессов, непосредственно работающих с представлениями требований конкретной группы на уровне знаний.

Решение о формировании рассмотренных выше видов онтологий на основе мо-

дифицированной фреймовой модели знаний (3.56) как сети фреймов (3.53) определяет необходимость выделения следующих структурных ППТ к ИС на уровне знаний:

- а) структурный паттерн проектирования фрейма;
- б) структурный паттерн проектирования интерфейса фрейма;
- в) структурный паттерн проектирования связей между узлами сети фреймов;
- г) обобщенный структурный паттерн проектирования сети фреймов.

Модель, описывающая структурный паттерн проектирования фрейма, Pt_{fr_str} , в общем случае будет иметь следующий вид [134]:

$$Pt_{fr_str} = \langle At_n, At_{el_fr}, At_{el_fr_t}, \langle at_n, at_{el_fr}, at_{el_fr_t} \rangle \rangle, \quad (3.58)$$

где At_n – кортеж атрибутов, описывающих имя фрейма;

At_{el_fr} – кортеж атрибутов, описывающих элемент фрейма (слот, интерфейс, метод);

$At_{el_fr_t}$ – кортеж атрибутов, описывающих тип элемента фрейма;

at_n – атрибут, идентифицирующий имя фрейма;

at_{el_fr} – атрибут, идентифицирующий элемент фрейма;

$at_{el_fr_t}$ – атрибут, идентифицирующий тип элемента фрейма.

Интерпретацией модели (3.58) в ходе разработки ИТ формирования и анализа требований к ИС является схема одной или нескольких витрин данных, предназначенных для хранения исторической информации о фреймах как узлах сети фреймов, описывающей Про, реализованные требования к ИС или требования к создаваемой ИС.

Модель, описывающая структурный паттерн проектирования интерфейса фрейма Pt_{if} , в общем случае будет иметь следующий вид [134]:

$$Pt_{if} = \langle At_g, At_{el_if}, At_{el_if_t}, \langle at_g, at_{el_if}, at_{el_if_t} \rangle \rangle, \quad (3.59)$$

где At_g – кортеж атрибутов, описывающих глобально уникальный иденти-

фикатор интерфейса фрейма;

At_{el_if} – кортеж атрибутов, описывающих элемент интерфейса фрейма (слот, метод);

$At_{el_if_t}$ – кортеж атрибутов, описывающих тип элемента интерфейса фрейма;

at_g – атрибут, идентифицирующий глобально уникальный идентификатор интерфейса фрейма;

at_{el_if} – атрибут, идентифицирующий элемент интерфейса фрейма;

$at_{el_fr_t}$ – атрибут, идентифицирующий тип элемента интерфейса фрейма.

Интерпретацией модели (3.59) в ходе разработки ИТ формирования и анализа требований к ИС является схема одной или нескольких витрин данных, предназначенных для хранения исторической информации об интерфейсах фреймов как узлах сети фреймов, описывающей ПрО, реализованные требования к ИС или требования к создаваемой ИС.

Модель, описывающая структурный паттерн проектирования связей между узлами сети фреймов, Pt_{fr_rel} в общем случае будет иметь вид [134]:

$$Pt_{fr_rel} = \langle At_{fr_rel_n}, At_{el_fr_rel}, At_{el_fr_rel_t}, \langle at_{fr_rel_n}, at_{el_fr_rel}, at_{el_fr_rel_t} \rangle \rangle, \quad (3.60)$$

где $At_{fr_rel_n}$ – кортеж атрибутов, описывающих имя связи;

$At_{el_fr_rel}$ – кортеж атрибутов, описывающих элемент описания связи;

$At_{el_fr_rel_t}$ – кортеж атрибутов, описывающих тип элемента описания связи;

$at_{fr_rel_n}$ – атрибут, идентифицирующий имя связи;

$at_{el_fr_rel}$ – атрибут, идентифицирующий элемент описания связи;

$at_{el_fr_rel_t}$ – атрибут, идентифицирующий тип элемента описания связи.

Описания возможных типов связей между объектными и структурными моделями сущностей приведены в [166].

Интерпретацией модели (3.60) в ходе разработки ИТ формирования и

анализа требований к ИС является схема одной или нескольких витрин данных, предназначенных для хранения исторической информации о связях между узлах сетей фреймов, описывающих ПрО, реализованные требования к ИС или требования к создаваемой ИС.

Основываясь на приведенных моделях структурных паттернов проектирования элементов сети фреймов, можно сделать вывод, что модель, описывающая структурный паттерн Pt_{net_fr} проектирования сети фреймов (3.53), в общем случае будет иметь следующий вид [134]:

$$Pt_{net_fr} = \langle Pt_{fr_str}, Pt_{if}, Pt_{fr_rel}, \langle at_n^1, at_n^2, at_{if}, at_{fr_rel_n} \rangle \rangle, \quad (3.61)$$

где at_n^1 – атрибут, идентифицирующий имя первого фрейма, который может принимать участие в образовании связи (может быть не определен);

at_n^2 – атрибут, идентифицирующий имя второго фрейма, который может принимать участие в образовании связи (может быть не определен).

Атрибут at_{if} также может быть не определен, поскольку каждая конкретная связь может существовать только между двумя фреймами, или между одним фреймом и интерфейсом. Условие, ограничивающее существование связей между узлами сети фреймов, будет иметь следующий вид [134]:

$$\begin{aligned} & \forall (at_n^1, at_n^2, at_{if}) \in \langle at_n^1, at_n^2, at_{if}, at_{fr_rel_n} \rangle \\ & (at_n^1 \cap at_n^2) \oplus (at_n^1 \cap at_{if}) \oplus (at_n^2 \cap at_{if}) \oplus (at_{if} \cap at_n^1) \oplus (at_{if} \cap at_n^2) = 1. \end{aligned} \quad (3.62)$$

Интерпретацией модели (3.61) в ходе разработки ИТ формирования и анализа требований к ИС является схема витрины данных, предназначенной для хранения информации о сети фреймов, описывающей ПрО, реализованные требования к ИС или требования к создаваемой ИС.

Таким образом, подкласс объектов K_{IS}^{Pt} модели (2.40) следует рассматривать как множество структурных ППТ к ИС на уровне знаний, устанавливающих конкретный вид элемента $\langle M_K^{AtPt} \rangle$ модели (3.57). Это множество паттернов в общем случае будет иметь следующий вид [134]:

$$\begin{aligned}
K_{IS}^{Pt} &= \{ Pt_{fr_str}, Pt_{if}, Pt_{fr_rel}, Pt_{net_fr} \} = \\
&= \{ \langle At_n, At_{el_fr}, At_{el_fr_t}, \langle at_n, at_{el_fr}, at_{el_fr_t} \rangle \rangle, \\
&\langle At_g, At_{el_if}, At_{el_if_t}, \langle at_g, at_{el_if}, at_{el_if_t} \rangle \rangle, \langle At_{fr_rel_n}, \\
&At_{el_fr_rel}, At_{el_fr_rel_t}, \langle at_{fr_rel_n}, at_{el_fr_rel}, \\
&at_{el_fr_rel_t} \rangle \rangle, \langle at_n^1, at_n^2, at_{if}, at_{fr_rel_n} \rangle \}.
\end{aligned} \tag{3.63}$$

Подкласс морфизмов $H(K_{IS}^{Pt})$, по аналогии с рассмотренными в подразд. 3.2 и 3.3 подклассами морфизмов $H(D_{IS}^{Pt})$ и $H(I_{IS}^{Pt})$, состоит из следующих видов морфизмов:

а) единичные морфизмы $1_{Pt_{fr_str}}, 1_{Pt_{if}}, 1_{Pt_{fr_rel}}$, отображающие структурные ППТ Pt_{fr_str}, Pt_{if} и Pt_{fr_rel} , соответственно, самих в себя;

б) морфизмы, устанавливающие связи между структурными ППТ $Pt_{fr_str}, Pt_{if}, Pt_{fr_rel}$ и Pt_{net_fr} .

Для структурного ППТ Pt_{net_fr} единичные морфизмы не существуют, поскольку данный ППТ наследует особенности структурных ППТ Pt_{fr_str}, Pt_{if} и Pt_{fr_rel} .

Для модели ППТ Pt_{fr_str} условия существования единичных морфизмов аналогичны условиям (3.12), (3.16) и (3.18), а модели этих морфизмов будут иметь, соответственно, следующий вид [134]:

$$\begin{aligned}
1_{Pt_{fr_str}}^{add} &: [At_n \oplus At_x] \oplus [At_{el_fr} \oplus At_x] \oplus [At_{el_fr_t} \oplus At_x] \oplus \\
&\oplus [\emptyset \oplus At_x],
\end{aligned} \tag{3.64}$$

$$\begin{aligned}
1_{Pt_{fr_str}}^{upd} : & [(At_n \setminus At_x) \cup At'_x] \oplus [(At_{el_fr} \setminus At_x) \cup At'_x] \oplus \\
& \oplus [(At_{el_fr_t} \setminus At_x) \cup At'_x],
\end{aligned} \tag{3.65}$$

$$1_{Pt_{fr_str}}^{del} : [At_n \setminus At_x] \oplus [At_{el_fr} \setminus At_x] \oplus [At_{el_fr_t} \setminus At_x]. \tag{3.66}$$

Для модели ППТ Pt_{if} условия существования единичных морфизмов аналогичны условиям (3.12), (3.16) и (3.18), а модели этих морфизмов будут иметь, соответственно, следующий вид [134]:

$$\begin{aligned}
1_{Pt_{if}}^{add} : & [At_g \oplus At_x] \oplus [At_{el_if} \oplus At_x] \oplus [At_{el_if_t} \oplus At_x] \oplus \\
& \oplus [\emptyset \oplus At_x],
\end{aligned} \tag{3.67}$$

$$\begin{aligned}
1_{Pt_{if}}^{upd} : & [(At_g \setminus At_x) \cup At'_x] \oplus [(At_{el_if} \setminus At_x) \cup At'_x] \oplus \\
& \oplus [(At_{el_if_t} \setminus At_x) \cup At'_x],
\end{aligned} \tag{3.68}$$

$$1_{Pt_{if}}^{del} : [At_g \setminus At_x] \oplus [At_{el_if} \setminus At_x] \oplus [At_{el_if_t} \setminus At_x]. \tag{3.69}$$

Для модели ППТ Pt_{fr_rel} условия существования единичных морфизмов аналогичны условиям (3.12), (3.16) и (3.18), а модели этих морфизмов будут иметь, соответственно, следующий вид [134]:

$$\begin{aligned}
1_{Pt_{fr_rel}}^{add} : & [At_{fr_rel_n} \oplus At_x] \oplus [At_{el_fr_rel} \oplus At_x] \oplus \\
& \oplus [At_{el_fr_rel_t} \oplus At_x] \oplus [\emptyset \oplus At_x],
\end{aligned} \tag{3.70}$$

$$1_{Pt_{fr_rel}}^{upd} : [(At_{fr_rel_n} \setminus At_x) \cup At'_x] \oplus [(At_{el_fr_rel} \setminus At_x) \cup At'_x] \oplus \oplus [(At_{el_fr_rel_t} \setminus At_x) \cup At'_x], \quad (3.71)$$

$$1_{Pt_{fr_rel}}^{del} : [At_{fr_rel} \setminus At_x] \oplus [At_{el_fr_rel} \setminus At_x] \oplus [At_{el_fr_rel_t} \setminus At_x]. \quad (3.72)$$

Из всех морфизмов, устанавливающих связи между моделями структурных ППТ Pt_{fr_str} , Pt_{if} , Pt_{fr_rel} и Pt_{net_fr} , могут существовать только морфизмы $H(Pt_{fr_str}, Pt_{if})$, $H(Pt_{fr_str}, Pt_{fr_rel})$, $H(Pt_{fr_str}, Pt_{net_fr})$, $H(Pt_{if}, Pt_{fr_str})$, $H(Pt_{if}, Pt_{fr_rel})$, $H(Pt_{if}, Pt_{net_fr})$, $H(Pt_{fr_rel}, Pt_{net_fr})$. Существование данных морфизмов обусловлено невозможностью появления описаний связей между узлами сети фреймов без предварительного появления описаний отдельных фреймов и интерфейсов, являющихся узлами этой сети.

Отдельным вопросом, требующим дальнейшего исследования, является вопрос существования морфизмов $H(Pt_{net_fr}, Pt_{fr_str})$, $H(Pt_{net_fr}, Pt_{if})$ и $H(Pt_{net_fr}, Pt_{fr_rel})$. Существование данных морфизмов может быть объяснено наличием каких-либо методов автоматической добычи знаний на основе результатов анализа сформированной сети фреймов, описывающей требования к создаваемой ИС. Кроме того, не менее важным вопросом является вопрос существования морфизмов типа $H(Pt_{net_fr}, Pt'_{net_fr})$ и $H(Pt'_{net_fr}, Pt_{net_fr})$, где Pt_{net_fr} – структурный паттерн проектирования сети фреймов, описывающей СТ к создаваемой ИС, а Pt'_{net_fr} – структурный паттерн проектирования сети фреймов, описывающей ПрО или реализованные требования к ИС. Существование данных морфизмов может быть объяснено наличием методов автоматического формирования или модификации сети фреймов. Данные методы могут использоваться для решения следующих задач:

а) формирование или модификация сети фреймов, описывающей СТ к создаваемой ИС, по результатам решения задач анализа соответствия конкретных СТ к создаваемой ИС реализованным требованиям к ранее разработанным ИС;

б) формирование или модификация сетей фреймов, описывающих ПрО или реализованные требования к ИС по результатам успешного завершения проекта создания ИС, знания о СТ и реализованных требованиях к которой описаны в виде соответствующей сети фреймов.

Решение задачи первого типа позволяет автоматизировать формирование сети фреймов, описывающей СТ к создаваемой ИС, с учетом решений о целесообразности ре-использования реализованных требований к ранее разработанным ИС и ИТ-услуг или ИТ-сервисов, реализующих эти требования. Решение задачи второго типа позволяет автоматизировать процессы расширения или модификации сетей фреймов, описывающих ПрО и реализованные требования к ИС, с учетом знаний, добытых из требований реализованного проекта создания ИС.

Морфизм $H(Pt_{fr_str}, Pt_{if})$ в общем случае имеет следующий вид [134]:

$$\begin{aligned} H(Pt_{fr_str}, Pt_{if}) : \{ At_n^{fr_str}, At_{el_fr}^{fr_str}, At_{el_fr_t}^{fr_str} \} \rightarrow \\ \rightarrow \{ At_{el_if}^{if}, At_{el_if_t}^{if} \} \end{aligned} \quad (3.73)$$

и существует только при выполнении условия

$$\begin{aligned} \forall \{ At_n^{fr_str}, At_{el_fr}^{fr_str}, At_{el_fr_t}^{fr_str} \} \\ [\{ At_n^{fr_str} \} \rightarrow \{ At_{el_if}^{if} \}] \oplus [\{ At_{el_fr}^{fr_str}, At_{el_fr_t}^{fr_str} \} \rightarrow \\ \rightarrow \{ At_{el_if}^{if}, At_{el_if_t}^{if} \}] = 1, \end{aligned} \quad (3.74)$$

где $At_n^{fr_str}$ – подмножество атрибутов, описывающих имя фрейма в модели структурного ППТ Pt_{fr_str} ;

$At_{el_fr}^{fr_str}$ – подмножество атрибутов, описывающих элемент фрейма в модели структурного ППТ Pt_{fr_str} ;

$At_{el_fr_t}^{fr_str}$ – подмножество атрибутов, описывающих тип элемента фрейма в модели структурного ППТ Pt_{fr_str} ;

$At_{el_if}^{if}$ – подмножество атрибутов, описывающих элемент интерфейса фрейма в модели структурного ППТ Pt_{if} ;

$At_{el_if_t}^{if}$ – подмножество атрибутов, описывающих тип элемента интерфейса фрейма в модели структурного ППТ Pt_{if} .

Морфизм $H(Pt_{fr_str}, Pt_{fr_rel})$ в общем случае имеет вид [134]:

$$\begin{aligned} H(Pt_{fr_str}, Pt_{fr_rel}) : \{ At_n^{fr_str}, At_{el_fr}^{fr_str}, At_{el_fr_t}^{fr_str} \} \rightarrow \\ \rightarrow \{ At_{el_fr_rel}^{fr_rel}, At_{el_fr_rel_t}^{fr_rel} \}, \end{aligned} \quad (3.75)$$

где $At_{el_fr_rel}^{fr_rel}$ – подмножество атрибутов, описывающих элемент описания связи в модели структурного ППТ Pt_{fr_rel} ;

$At_{el_fr_rel_t}^{fr_rel}$ – подмножество атрибутов, описывающих тип элемента описания связи в модели структурного ППТ Pt_{fr_rel} .

Морфизм $H(Pt_{fr_str}, Pt_{net_fr})$ в общем случае имеет вид [134]:

$$\begin{aligned} H(Pt_{fr_str}, Pt_{net_fr}) : \{ At_n^{fr_str}, At_{el_fr}^{fr_str}, At_{el_fr_t}^{fr_str} \} \rightarrow \\ \rightarrow \{ At_n^{net_fr}, At_{el_fr}^{net_fr}, At_{el_fr_t}^{net_fr} \}, \end{aligned} \quad (3.76)$$

где $At_n^{net_fr}$ – подмножество атрибутов, описывающих имя фрейма в модели структурного ППТ Pt_{net_fr} ;

$At_{el_fr}^{net_fr}$ – подмножество атрибутов, описывающих элемент фрейма в модели структурного ППТ Pt_{net_fr} ;

$At_{el_fr_t}^{net_fr}$ – подмножество атрибутов, описывающих тип элемента фрейма в модели структурного ППТ Pt_{net_fr} .

Морфизм $H(Pt_{if}, Pt_{fr_str})$ в общем случае имеет вид [134]:

$$H(Pt_{if}, Pt_{fr_str}) : \{ At_g^{if} \} \rightarrow \{ At_{el_fr_str}^{fr} \}, \quad (3.77)$$

где At_g^{if} – подмножество атрибутов, описывающих глобально уникальный идентификатор интерфейса фрейма в модели структурного ППТ Pt_{if} .

Морфизм $H(Pt_{if}, Pt_{fr_rel})$ в общем случае имеет вид [134]:

$$\begin{aligned} H(Pt_{if}, Pt_{fr_rel}) : \{ At_g^{if}, At_{el_if}^{if}, At_{el_if_t}^{if} \} \rightarrow \\ \rightarrow \{ At_{el_fr_rel}^{fr_rel}, At_{el_fr_rel_t}^{fr_rel} \}. \end{aligned} \quad (3.78)$$

Морфизм $H(Pt_{if}, Pt_{net_fr})$ в общем случае имеет вид [134]:

$$\begin{aligned} H(Pt_{if}, Pt_{net_fr}) : \{ At_g^{if}, At_{el_if}^{if}, At_{el_if_t}^{if} \} \rightarrow \\ \rightarrow \{ At_g^{net_fr}, At_{el_if}^{net_fr}, At_{el_if_t}^{net_fr} \}, \end{aligned} \quad (3.79)$$

где $At_g^{net_fr}$ – подмножество атрибутов, описывающих глобально уникальный идентификатор интерфейса фрейма в модели структурного ППТ Pt_{net_fr} ;

$At_{el_if}^{net_fr}$ – подмножество атрибутов, описывающих элемент интерфейса

фрейма в модели структурного ППТ Pt_{net_fr} ;

$At_{el_if_t}^{net_fr}$ – подмножество атрибутов, описывающих тип элемента интерфейса фрейма в модели структурного ППТ Pt_{net_fr} .

Морфизм $H(Pt_{fr_rel}, Pt_{net_fr})$ в общем случае имеет вид [134]:

$$H(Pt_{fr_rel}, Pt_{net_fr}) : \{ At_{fr_rel_n}^{fr_rel}, At_{el_fr_rel}^{fr_rel}, At_{el_fr_rel_t}^{fr_rel} \} \rightarrow \{ At_{fr_rel_n}^{net_fr}, At_{el_fr_rel}^{net_fr}, At_{el_fr_rel_t}^{net_fr} \}, \quad (3.80)$$

где $At_{fr_rel_n}^{net_fr}$ – подмножество атрибутов, описывающих имя связи в модели структурного ППТ Pt_{net_fr} ;

$At_{el_fr_rel}^{net_fr}$ – подмножество атрибутов, описывающих элемент описания связи в модели структурного ППТ Pt_{net_fr} ;

$At_{el_fr_rel_t}^{net_fr}$ – подмножество атрибутов, описывающих тип элемента описания связи в модели структурного ППТ Pt_{net_fr} .

Необходимо отметить, что для случая использования отдельных версий требований к ИС морфизмы, составляющие подкласс $H(K_{IS}^{Pt})$, будут иметь аналогичный вид.

Таким образом, предлагаемые модели структурных ППТ (3.58)-(3.61) как объектов подкласса K_{IS}^{Pt} и поведенческих ППТ (3.64)-(3.80) как морфизмов подкласса $H(K_{IS}^{Pt})$ определяют синтаксис и семантику описаний знаний, добытых из публикаций требований к ИС. С точки зрения практической реализации, разработанные модели задают структуру схемы фрагмента хранилища данных, обеспечивающего хранение представлений требований к ИС и их версий на уровне знаний вне зависимости от конкретных проектов создания ИС, в которых были сформулированы эти требования.

Результат применения рассмотренной выше группы структурных и пове-

денческих ППТ требований к ИС на уровне знаний (3.58)-(3.61) будет представлять собой модель представления i -го ФТ к ИС tr_i^f на уровне знаний. Данная модель является основным элементом предлагаемого АФ макропроектирования ИС и представляет собой множество кортежей вида [180]:

$$K_i^f = \{ \langle d_n^{ij}, \{ \langle d_{el_fr}^{ij}, d_{el_fr_t}^{ij} \rangle \} \rangle, \langle d_g^{ij}, \{ \langle d_{el_if}^{ij}, d_{el_if_t}^{ij} \rangle \} \rangle, \langle d_{fr_rel_n}^{ij}, \{ \langle d_{el_fr_rel}^{ij}, d_{el_fr_rel_t}^{ij} \rangle \} \rangle \}, \quad (3.81)$$

где d_n^{ij} - описание наименования фрейма;

$d_{el_fr}^{ij}$ - описание элемента фрейма;

$d_{el_fr_t}^{ij}$ - описание типа элемента фрейма;

d_g^{ij} - описание наименования интерфейса;

$d_{el_if}^{ij}$ - описание элемента интерфейса;

$d_{el_if_t}^{ij}$ - описание типа элемента интерфейса;

$d_{fr_rel_n}^{ij}$ - описание наименования связи между интерфейсами и/или фреймами;

$d_{el_fr_rel}^{ij}$ - описание элемента связи;

$d_{el_fr_rel_t}^{ij}$ - описание типа элемента связи.

При этом между элементами представления (3.81) существуют следующие отношения принадлежности [180]:

а) отношение принадлежности элементов фрейма конкретному фрейму, имеющее вид

$$F_n^{d_n^{ij}} \left\{ \langle d_{el_fr}^{ij}, d_{el_fr_t}^{ij} \rangle \right\} : \begin{cases} D_n^i \rightarrow 2^{\langle d_{el_fr}^{ij}, d_{el_fr_t}^{ij} \rangle} \\ d_n^{ij} \rightarrow \{ \langle d_{el_fr}^{ij}, d_{el_fr_t}^{ij} \rangle \} \end{cases}, \quad (3.82)$$

где D_n^i - множество описаний наименований фреймов, характеризующих концепты ПрО с точки зрения отдельного участника автоматизируемого процесса (УАП) st_k ;

$2^{\langle d_{el_fr}^{ij}, d_{el_fr_t}^{ij} \rangle}$ - булеан (множество всех подмножеств) подмножества $\{ \langle d_{el_fr}^{ij}, d_{el_fr_t}^{ij} \rangle \}$, описывающий возможное содержание фреймов с множеством описаний наименований D_n^i ;

$\{ \langle d_{el_fr}^{ij}, d_{el_fr_t}^{ij} \rangle \}$ - подмножество описаний элементов и их типов фрейма с наименованием d_n^{ij} ;

б) отношение принадлежности элементов интерфейса конкретному интерфейсу, имеющее вид

$$F_g^{d_g^{ij}} \left\{ \langle d_{el_if}^{ij}, d_{el_if_t}^{ij} \rangle \right\} : \begin{cases} D_g^i \rightarrow 2^{\langle d_{el_if}^{ij}, d_{el_if_t}^{ij} \rangle} \\ d_g^{ij} \rightarrow \{ \langle d_{el_if}^{ij}, d_{el_if_t}^{ij} \rangle \} \end{cases}, \quad (3.83)$$

где D_g^i - множество описаний наименований интерфейсов, характеризующих концепты ПрО с точки зрения отдельного УАП st_k ;

$2^{\langle d_{el_if}^{ij}, d_{el_if_t}^{ij} \rangle}$ - булеан подмножества $\{ \langle d_{el_if}^{ij}, d_{el_if_t}^{ij} \rangle \}$, описывающий возможное содержание интерфейсов с множеством описаний наименований D_g^i ;

$\{ \langle d_{el_if}^{ij}, d_{el_if_t}^{ij} \rangle \}$ - подмножество описаний элементов и их типов ин-

терфейса с наименованием d_g^{ij} ;

в) отношение принадлежности элементов связи конкретной связи, имеющее вид

$$F^{d_{fr_rel_n}^{ij}} \{ \langle d_{el_fr_rel}^{ij}, d_{el_fr_rel_t}^{ij} \rangle \} : \left\{ \begin{array}{l} D_{fr_rel_n}^i \rightarrow 2^{\{ \langle d_{el_fr_rel}^{ij}, d_{el_fr_rel_t}^{ij} \rangle \}}; \\ d_{fr_rel_n}^{ij} \rightarrow \{ \langle d_{el_fr_rel}^{ij}, d_{el_fr_rel_t}^{ij} \rangle \}, \end{array} \right. \quad (3.84)$$

где $D_{fr_rel_n}^i$ - множество описаний наименований связей между интерфейсами и/или фреймами, характеризующими концепты Про с точки зрения отдельного УАП st_k ;

$2^{\{ \langle d_{el_fr_rel}^{ij}, d_{el_fr_rel_t}^{ij} \rangle \}}$ - булеан подмножества $\{ \langle d_{el_fr_rel}^{ij}, d_{el_fr_rel_t}^{ij} \rangle \}$ описывающий возможное содержание описаний связей с множеством описаний наименований $D_{fr_rel_n}^i$;

$\{ \langle d_{el_fr_rel}^{ij}, d_{el_fr_rel_t}^{ij} \rangle \}$ - подмножество описаний элементов и их типов связи с наименованием $d_{fr_rel_n}^{ij}$.

Разработанная модель позволяет описать представление ФТ к ИС на уровне знаний. Использование данной модели позволяет упростить автоматизацию операций над представлениями ФТ к ИС на уровне знаний за счет унификации их формального описания. Это, в свою очередь, позволяет установить общие особенности и ограничения и решить задачу синтеза описания рациональной архитектуры создаваемой информационной системы.

3.5 Выводы к третьему разделу

1. Основываясь на изложенной в подразд. 2.3 концепции представления требований к ИС и модели (2.40), разработана математическая модель АФ макропроектирования ИС (3.2), основанная на формальном представлении данного АФ в виде множеств структурных и поведенческих ППТ к ИС. Эта модель позволяет установить общие особенности и ограничения практик создания, интерпретации, анализа и использования АД ИС на уровне требований, выдвигаемых к данной ИС, в процессах макропроектирования и синтеза единого целостного АД ИС.

2. Разработаны теоретико-множественные модели структурных ППТ (3.8), (3.9), (Г.1) и (Г.2) как объектов подкласса D_{IS}^{Pt} и поведенческих ППТ (3.13), (3.17), (3.19)-(3.29) как морфизмов подкласса $H(D_{IS}^{Pt})$, определяющие синтаксис и семантику описаний требований к ИС на уровне данных, что позволяет унифицировать работы по формированию, хранению и обработке представлений требований к ИС на уровне данных вне зависимости от конкретных проектов создания ИС, в которых требования были сформулированы, и адаптировать АФ макропроектирования ИС к особенностям атрибутивных МТ, применяемых конкретными Поставщиками.

3. Разработаны теоретико-множественные модели структурных ППТ (3.31)-(3.34), (Г.5) как объектов подкласса I_{IS}^{Pt} и поведенческих ППТ (3.36)-(3.52) как морфизмов подкласса $H(I_{IS}^{Pt})$, определяющие синтаксис и семантику описаний требований к ИС на уровне информации, что позволяет унифицировать работы по формированию, хранению и обработке представлений требований к ИС на уровне информации вне зависимости от конкретных проектов создания ИС, в которых требования были сформулированы.

4. Разработаны теоретико-множественные модели структурных ППТ

(3.58)-(3.61) как объектов подкласса K_{IS}^{Pt} и поведенческих ППТ (3.64)-(3.80) как морфизмов подкласса $H(K_{IS}^{Pt})$, определяющие синтаксис и семантику описаний требований к ИС на уровне знаний, что позволяет унифицировать работы по формированию, хранению и обработке представлений требований к ИС на уровне знаний вне зависимости от конкретных проектов создания ИС, в которых требования были сформулированы.

5. С использованием моделей структурных ППТ к ИС (3.58)-(3.61) разработана модель СФТ на уровне знаний как основного элемента предложенного в работе АФ макропроектирования ИС, что позволяет установить общие особенности и ограничения и решить задачу синтеза описания рациональной архитектуры создаваемой ИС.

Основные результаты раздела изложены в работах [134, 160, 161, 165-170, 180].

РАЗДЕЛ 4

МОДЕЛИ И МЕТОДЫ СИНТЕЗА РАЦИОНАЛЬНОЙ АРХИТЕКТУРЫ
ИНФОРМАЦИОННОЙ СИСТЕМЫ4.1 Задача синтеза описания рациональной архитектуры
информационной системы

В соответствии с предложенной в подразд. 2.4 моделью универсума требований к ИС M_{UTr} (2.30), множество СТ к ИС может быть получено различными способами, которые могут быть в общем случае сведены к следующим вариантам:

а) формирование требований к ИС Потребителем и последующее согласование этих требований с Поставщиком;

б) формирование требований к ИС Поставщиком и последующее согласование этих требований с Потребителем;

в) параллельное формирование требований к ИС Поставщиком и Потребителем с последующим согласованием этих требований заинтересованными сторонами.

IDEF3-диаграмма, иллюстрирующая перечисленные выше варианты, приведена на рис. 4.1.

Применение модели ППТ к ИС M_{IS}^{Pt} (2.40) не только к множеству СТ к ИС (2.29), но и ко всему универсуму требований к ИС (2.30) позволяет утверждать, что в рамках процессов непосредственной работы с требованиями к ИС и Поставщик, и Потребитель используют один и тот же аппарат формального представления требований к ИС на уровнях данных, информации и знаний. Это, в свою очередь, позволяет представить работы по формированию множества СТ к ИС IDEF3-диаграммой, показанной на рис. 4.2.

Такое представление работ по созданию множества СТ к ИС позволяет выделить следующие виды представлений отдельного требования к ИС на уровне знаний [180]:

а) представление требования к ИС на уровне знаний с точки зрения Потребителя K_i^{trU} как объединение версий данного требования, отражающих точки

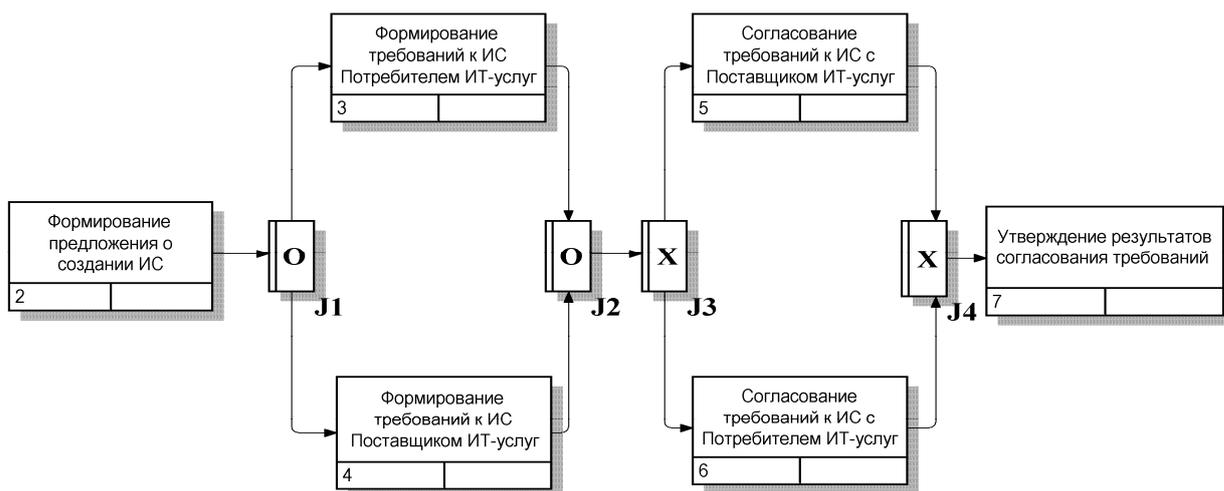


Рис. 4.1 – IDEF3-диаграмма, иллюстрирующая работы по созданию множества сформулированных требований к информационной системе

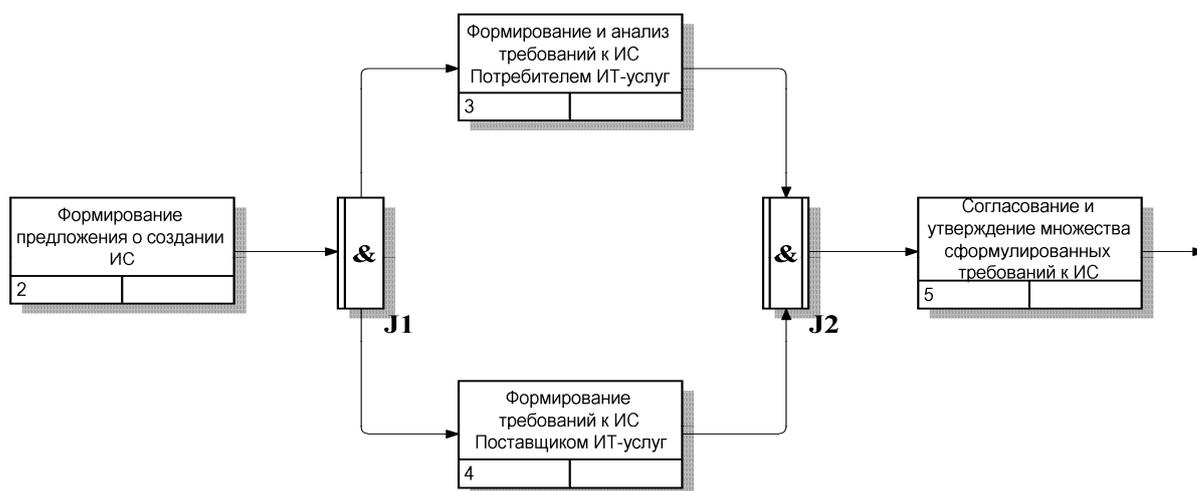


Рис. 4.2 – IDEF3-диаграмма, иллюстрирующая работы по созданию множества сформулированных требований к информационной системе на основе общего для Поставщика и Потребителя аппарата формального представления требований

зрения различных УАП $K_i^{tr}(U_{st_k})$, заинтересованных в создании ИС ;

б) представление требования к ИС на уровне знаний с точки зрения Поставщика K_i^{trPr} как совокупность знаний о возможности ре-использования в данной ИС ИТ-сервисов, реализующих аналогичные требования в ранее выполненных ИТ-проектах;

в) представление требования к ИС на уровне знаний $K_i^{tr} \in K_{IS}^{tr}$, $K_{IS}^{tr} = [K_1^{tr}, K_2^{tr}, \dots, K_i^{tr}, \dots, K_n^{tr}]$, сформированное в результате согласования точек зрения Потребителя и Поставщика.

Данные представления позволяют рассматривать стандартный процесс проектирования архитектуры создаваемой ИС [8] как процесс синтеза описания этой архитектуры путем согласования знаниеориентированных МТ к данной системе с точки зрения Поставщика и Потребителя и выработка по результатам данного согласования единого представления ИС как совокупности ИТ-услуг, реализующих СТ к ИС. Однако, как показано в подразд. 1.6, Поставщик и Потребитель в ходе выполнения процесса синтеза архитектуры ИС преследуют различные цели, что серьезно затрудняет выработку единого представления ИС на уровне знаний.

Поэтому возникает задача синтеза такого АД создаваемой ИС на уровне знаний, которое в максимальной степени обеспечивало бы достижение целей Поставщика и Потребителя. Такое описание здесь и в дальнейшем будем называть описанием рациональной архитектуры (ОРА) создаваемой ИС.

Предложенная в подразд. 2.3 концепция представления требований к ИС и разработанные на ее основе модели представлений требований к ИС на уровнях данных, информации и знаний позволяют детализировать формализованные описания целей Поставщика и Потребителя ИТ-услуг, представленные выражениями (1.3)-(1.6). С учетом предложенной в подразд. 2.2 классификации требований к ИС, целевые функции (1.3) и (1.5) примут, соответственно, следующий вид [134, 181]:

$$F_{Pr} = \sum_{i=1}^a r^{Pr}(tr_i^B) + \sum_{i=a+1}^b r^{Pr}(tr_i^{IB}) + \sum_{i=b+1}^c r^{Pr}(tr_i^S) + \sum_{i=c+1}^e r^{Pr}(tr_i^f) + \sum_{i=e+1}^g r^{Pr}(tr_i^{nf}) + \sum_{i=g+1}^k r^{Pr}(tr_i^{fw}) + \sum_{i=k+1}^n r^{Pr}(tr_i^{nfw}) \rightarrow \max, \quad (4.1)$$

$$\begin{aligned}
F_U = & \sum_{i=1}^a r^U (tr_i^B) + \sum_{i=a+1}^b r^U (tr_i^{IB}) + \sum_{i=b+1}^c r^U (tr_i^S) + \sum_{i=c+1}^e r^U (tr_i^f) + \\
& + \sum_{i=e+1}^g r^U (tr_i^{nf}) + \sum_{i=g+1}^k r^U (tr_i^{fw}) + \sum_{i=k+1}^n r^U (tr_i^{nfw}) \rightarrow \max
\end{aligned} \quad (4.2)$$

Подобное представление целевых функций можно свести к функциям выбора оптимального набора ИТ-услуг ИС при соблюдении Поставщиком и Потребителем определенных ограничений. Тогда описание глобальной цели Поставщика в процессе проектирования архитектуры ИС можно представить как задачу максимального удовлетворения требований, выдвинутых Поставщиком к создаваемой ИС, при соблюдении ограничений реализации всех выделенных выше групп требований [181, 182]. При этом в ходе описания соответствующих ограничений следует учитывать как особенности формирования и анализа требований соответствующих групп, так и индивидуальные особенности Поставщика. Следует также помнить, что Поставщик формирует свое представление об ИТ-услугах создаваемой ИС и, соответственно, о ФТ к этим ИТ-услугам, исходя из тех требований к ИС как аспекту БП, которые предоставил ему Потребитель создаваемой ИС.

Формализованно данную задачу можно представить [181]:

$$F_{Pr} = \sum_{i=c+1}^e r^{Pr} (tr_i^f) \rightarrow \max, \quad (4.3)$$

$$\left\{ \begin{array}{l} \sum_{i=c+1}^e \alpha_i^{Pr} \text{pay}(r^{Pr}(tr_i^f)) \geq \text{pay}^* \left(\sum_{i=c+1}^e r^{Pr}(tr_i^f) \right); \\ \sum_{i=c+1}^e \beta_i^{Pr} t(r^{Pr}(tr_i^f)) \leq t^* \left(\sum_{i=c+1}^e r^{Pr}(tr_i^f) \right); \\ \sum_{i=c+1}^e \gamma_i^{Pr} q(r^{Pr}(tr_i^f)) \geq q^* \left(\sum_{i=c+1}^e r^{Pr}(tr_i^f) \right); \end{array} \right. , \left\{ \begin{array}{l} \sum_{i=1}^a \mu_i^{Pr} r^{Pr}(tr_i^B) \geq \rho_{Pr}^B; \\ \sum_{i=a+1}^b \nu_i^{Pr} r^{Pr}(tr_i^{IB}) \geq \rho_{Pr}^{IB}; \\ \sum_{i=b+1}^c \chi_i^{Pr} r^{Pr}(tr_i^S) \geq \rho_{Pr}^S; \\ \sum_{i=e+1}^g \eta_i^{Pr} r^{Pr}(tr_i^{nf}) \geq \rho_{Pr}^{nf}; \\ \sum_{i=g+1}^k \zeta_i^{Pr} r^{Pr}(tr_i^{fW}) \geq \rho_{Pr}^{fW}; \\ \sum_{i=k+1}^n \omega_i^{Pr} r^{Pr}(tr_i^{nfW}) \geq \rho_{Pr}^{nfW}; \end{array} \right. \quad (4.4)$$

где μ_i^{Pr} – нормативный коэффициент полноты реализации бизнес-требования

tr_i^B , учитывающий индивидуальные особенности Поставщика;

ρ_{Pr}^B – минимально допустимый для Поставщика уровень полноты реализации подмножества бизнес-требований;

ν_i^{Pr} – нормативный коэффициент полноты реализации требования к ИС как к аспекту

бизнеса tr_i^{IB} , учитывающий индивидуальные особенности Поставщика;

ρ_{Pr}^{IB} – минимально допустимый для Поставщика уровень полноты реализации подмножества требований к ИС как к аспекту БП;

χ_i^{Pr} – нормативный коэффициент полноты реализации требования к ИС в

целом tr_i^S , учитывающий индивидуальные особенности Поставщика;

ρ_{Pr}^S – минимально допустимый для Поставщика уровень полноты реализации подмножества требований к ИС в целом;

η_i^{Pr} – нормативный коэффициент полноты реализации нефункционального требования

к IT-услуге tr_i^{nf} , учитывающий индивидуальные особенности Поставщика;

ρ_{Pr}^{nf} – минимально допустимый для Поставщика уровень полноты реализации подмножества нефункциональных требований к ИТ-услугам;

ζ_i^{Pr} – нормативный коэффициент полноты реализации ФТ к ИТ-сервису tr_i^{fw} , учитывающий индивидуальные особенности Поставщика;

ρ_{Pr}^{fw} – минимально допустимый для Поставщика уровень полноты реализации подмножества ФТ к ИТ-сервисам;

ω_i^{Pr} – нормативный коэффициент полноты реализации нефункционального требования к ИТ-сервису tr_i^{nfw} , учитывающий индивидуальные особенности Поставщика;

ρ_{Pr}^{nfw} – минимально допустимый для Поставщика уровень полноты реализации подмножества нефункциональных требований к ИТ-сервисам.

Описание глобальной цели Потребителя в процессе проектирования архитектуры ИС можно представить как задачу максимального удовлетворения требований, выдвинутых Потребителем к создаваемой ИС, при соблюдении ограничений реализации всех выделенных выше групп требований [181, 182]. При этом в ходе описания соответствующих ограничений следует учитывать как особенности формирования и анализа требований соответствующих групп, так и индивидуальные особенности Потребителя.

Формализованно данную задачу можно представить [181]:

$$F_U = \sum_{i=c+1}^e r^U(tr_i^f) \rightarrow \max, \quad (4.5)$$

$$\left\{ \begin{array}{l} \sum_{i=c+1}^e \alpha_i^U \text{pay}(r^U(tr_i^f)) \leq \text{pay}^* \left(\sum_{i=c+1}^e r^U(tr_i^f) \right); \\ \sum_{i=c+1}^e \beta_i^U t(r^U(tr_i^f)) \leq t^* \left(\sum_{i=c+1}^e r^U(tr_i^f) \right); \\ \sum_{i=c+1}^e \gamma_i^U q(r^U(tr_i^f)) \geq q^* \left(\sum_{i=c+1}^e r^U(tr_i^f) \right); \end{array} \right. , \left\{ \begin{array}{l} \sum_{i=1}^a \mu_i^U r^U(tr_i^B) \geq \rho_U^B; \\ \sum_{i=a+1}^b v_i^{Pr} r^{Pr}(tr_i^{IB}) \geq \rho_{Pr}^{IB}; \\ \sum_{i=b+1}^c \chi_i^U r^U(tr_i^S) \geq \rho_U^S; \\ \sum_{i=e+1}^g \eta_i^U r^U(tr_i^{nf}) \geq \rho_U^{nf}; \\ \sum_{i=g+1}^k \zeta_i^U r^U(tr_i^{fw}) \geq \rho_U^{fw}; \\ \sum_{i=k+1}^n \omega_i^U r^U(tr_i^{nfw}) \geq \rho_U^{nfw}, \end{array} \right. \quad (4.6)$$

где μ_i^U – нормативный коэффициент полноты реализации бизнес-требования

tr_i^B , учитывающий индивидуальные особенности Потребителя;

ρ_U^B – минимально допустимый для Потребителя уровень полноты реализации подмножества бизнес-требований;

v_i^U – нормативный коэффициент полноты реализации требования к ИС как к аспекту бизнеса tr_i^{IB} , учитывающий индивидуальные особенности Потребителя;

ρ_U^{IB} – минимально допустимый для Потребителя уровень полноты реализации подмножества требований к ИС как к аспекту БП;

χ_i^U – нормативный коэффициент полноты реализации требования к ИС в целом tr_i^S , учитывающий индивидуальные особенности Потребителя;

ρ_U^S – минимально допустимый для Потребителя уровень полноты реализации подмножества требований к ИС в целом;

η_i^U – нормативный коэффициент полноты реализации нефункционального требования к ИТ-услуге tr_i^{nf} , учитывающий индивидуальные особенности Потребителя;

ρ_U^{nf} – минимально допустимый для Потребителя уровень полноты реализации подмножества нефункциональных требований к ИТ-услугам;

ζ_i^U – нормативный коэффициент полноты реализации ФТ к ИТ-сервису tr_i^{fw} , учитывающий индивидуальные особенности Потребителя;

ρ_U^{fw} – минимально допустимый для Потребителя уровень полноты реализации подмножества ФТ к ИТ-сервисам;

ω_i^U – нормативный коэффициент полноты реализации нефункционального требования к ИТ-сервису tr_i^{nfw} , учитывающий индивидуальные особенности Потребителя;

ρ_U^{nfw} – минимально допустимый для Потребителя уровень полноты реализации подмножества нефункциональных требований к ИТ-сервисам.

Разделение систем ограничений (4.4) и (4.6) на две части обусловлено различными уровнями представлений данных ограничений с точки зрения стандарта ISO 15288 [8], в котором процессы проекта и технические процессы создания ИС разделены на две самостоятельные группы.

С учетом выделенных выше представлений требований к ИС на уровне знаний, выражения (4.3) и (4.4) примут вид:

$$F_{Pr} = \sum_{i=c+1}^e r^{Pr} (K_i^{f_{Pr}}) \rightarrow \max, \quad (4.7)$$

$$\begin{cases} \sum_{i=c+1}^e \alpha_i^{Pr} \text{pay}(r^{Pr} (K_i^{f_{Pr}})) \geq \text{pay}^* \left(\sum_{i=c+1}^e r^{Pr} (K_i^{f_{Pr}}) \right); \\ \sum_{i=c+1}^e \beta_i^{Pr} t(r^{Pr} (K_i^{f_{Pr}})) \leq t^* \left(\sum_{i=c+1}^e r^{Pr} (K_i^{f_{Pr}}) \right); \\ \sum_{i=c+1}^e \gamma_i^{Pr} q(r^{Pr} (K_i^{f_{Pr}})) \geq q^* \left(\sum_{i=c+1}^e r^{Pr} (K_i^{f_{Pr}}) \right), \end{cases} \quad (4.8)$$

$$\left\{ \begin{array}{l} \sum_{i=1}^a \mu_i^{Pr, Pr} (K_i^{B_{Pr}}) \geq \rho_{Pr}^B; \\ \sum_{i=a+1}^b \nu_i^{Pr, Pr} (K_i^{IB_{Pr}}) \geq \rho_{Pr}^{IB}; \\ \sum_{i=b+1}^c \chi_i^{Pr, Pr} (K_i^{S_{Pr}}) \geq \rho_{Pr}^S; \\ \sum_{i=e+1}^g \eta_i^{Pr, Pr} (K_i^{nf_{Pr}}) \geq \rho_{Pr}^{nf}; \\ \sum_{i=g+1}^k \zeta_i^{Pr, Pr} (K_i^{fw_{Pr}}) \geq \rho_{Pr}^{fw}; \\ \sum_{i=k+1}^n \omega_i^{Pr, Pr} (K_i^{nfw_{Pr}}) \geq \rho_{Pr}^{nfw}, \end{array} \right. \quad (4.9)$$

где $K_i^{f_{Pr}}$ - представление на уровне знаний ФТ tr_i^f с точки зрения Поставщика;

$K_i^{B_{Pr}}$ - представление на уровне знаний бизнес-требования tr_i^B с точки зрения Поставщика;

$K_i^{IB_{Pr}}$ - представление на уровне знаний требования к ИС как к аспекту бизнеса tr_i^{IB} с точки зрения Поставщика;

$K_i^{S_{Pr}}$ - представление на уровне знаний требования к ИС в целом tr_i^S с точки зрения Поставщика;

$K_i^{nf_{Pr}}$ - представление на уровне знаний нефункционального требования к ИТ-услуге tr_i^{nf} с точки зрения Поставщика;

$K_i^{fw_{Pr}}$ - представление на уровне знаний ФТ к ИТ-сервису tr_i^{fw} с точки зрения Поставщика;

$K_i^{nfw_{Pr}}$ - представление на уровне знаний нефункционального требования к ИТ-сервису tr_i^{nfw} с точки зрения Поставщика.

Задача максимального удовлетворения требований, выдвинутых Потребителем к создаваемой ИС, при соблюдении ограничений реализации всех выделенных выше групп требований примет вид:

$$F_U = \sum_{i=c+1}^e r^U(K_i^{fU}) \rightarrow \max, \quad (4.10)$$

$$\left\{ \begin{array}{l} \sum_{i=c+1}^e \alpha_i^U \text{pay}(r^U(K_i^{fU})) \leq \text{pay}^* \left(\sum_{i=c+1}^e r^U(K_i^{fU}) \right); \\ \sum_{i=c+1}^e \beta_i^U t(r^U(K_i^{fU})) \leq t^* \left(\sum_{i=c+1}^e r^U(K_i^{fU}) \right); \\ \sum_{i=c+1}^e \gamma_i^U q(r^U(K_i^{fU})) \geq q^* \left(\sum_{i=c+1}^e r^U(K_i^{fU}) \right), \end{array} \right. \quad (4.11)$$

$$\left\{ \begin{array}{l} \sum_{i=1}^a \mu_i^U r^U(K_i^{BU}) \geq \rho_U^B; \\ \sum_{i=a+1}^b \nu_i^{\text{Pr}} r^{\text{Pr}}(K_i^{IBU}) \geq \rho_{\text{Pr}}^{IB}; \\ \sum_{i=b+1}^c \chi_i^U r^U(K_i^{SU}) \geq \rho_U^S; \\ \sum_{i=e+1}^g \eta_i^U r^U(K_i^{nfU}) \geq \rho_U^{nf}; \\ \sum_{i=g+1}^k \zeta_i^U r^U(K_i^{fwU}) \geq \rho_U^{fw}; \\ \sum_{i=k+1}^n \omega_i^U r^U(K_i^{nfwU}) \geq \rho_U^{nfw}. \end{array} \right. \quad (4.12)$$

где K_i^{fU} - представление на уровне знаний ФТ tr_i^f с точки зрения Потребителя;

K_i^{BU} - представление на уровне знаний бизнес-требования tr_i^B с точки зрения Потребителя;

K_i^{IBU} - представление на уровне знаний требования к ИС как к аспекту бизнеса tr_i^{IB} с точки зрения Потребителя;

K_i^{SU} - представление на уровне знаний требования к ИС в целом tr_i^S с точки зрения Потребителя;

K_i^{nfU} - представление на уровне знаний нефункционального требования к ИТ-услуге tr_i^{nf} с точки зрения Потребителя;

K_i^{fwU} - представление на уровне знаний ФТ к ИТ-сервису tr_i^{fw} с точки зрения Потребителя;

K_i^{nfwU} - представление на уровне знаний нефункционального требования к ИТ-сервису tr_i^{nfw} с точки зрения Потребителя.

Таким образом, в ходе выполнения процесса синтеза архитектуры создаваемой ИС требуется решить задачу синтеза ОРА на уровне знаний, которое будет в максимальной степени удовлетворять целям Поставщика и Потребителя, заданным выражениями (4.7)-(4.12). Для решения этой задачи необходимо разработать:

- а) методы синтеза представлений ФТ к ИС на уровне знаний;
- б) метод синтеза вариантов АД создаваемой ИС;
- в) модели и метод решения задачи синтеза ОРА ИС;
- г) модели и метод оценивания затрат на создание ИС с синтезированной архитектурой.

4.2 Методы формирования представлений сформулированного требования к информационной системе на уровнях информации и знаний

4.2.1 Метод формирования представления сформулированного требования на уровне информации

Основой для формирования представлений требований к ИС на уровне знаний являются представления данных требований на уровне информации. Эти представления, как показано в подразд. 3.3, формируются на основе публикаций требований к ИС. Метод формирования представлений требований к ИС на уровне информации в общем случае предполагает выполнение следующих этапов [180, 183].

Этап 1. Формирование публикации i -го требования к ИС (или его версии).

Этап 2. Формирование описания I_{ij}^{r-publ} публикации i -го требования к ИС (или его версии), структура которого определяется паттерном (3.31).

Этап 3. Формирование множества описаний $\{I_{ij}^{r-kn-st}\}$ представителя заинтересованной стороны (Потребителя или Поставщика) как источника i -го требования к ИС (или его версии), структура которого определяется паттерном (3.33).

Этап 4. Формирование множества описаний $\{I_{ij}^{r-kn-eobj}\}$ внешних по отношению к процессам i -го требования к ИС (или его версии) объектов, непосредственно участвующих в выполнении процессов и присутствующих в публикации этого требования к ИС (или его версии), структура которых определяется паттерном (3.33).

Этап 5. Формирование множества описаний $\{I_{ij}^{r-kn-dst}\}$ структур данных, присутствующих в публикации i -го требования к ИС (или его версии), структура которых определяется паттерном (3.32).

Этап 6. Формирование множества описаний $\{I_{ij}^{r-kn-proc}\}$ процессов, присутствующих в публикации i -го требования к ИС (или его версии), структура которых определяется паттерном (3.34).

Этап 7. Формирование общего представления i -го требования к ИС (или его версии) на уровне информации

$$I_i^{tr} = I_{ij}^{r-publ} \cup \{I_{ij}^{r-kn-st}\} \cup \{I_{ij}^{r-kn-eobj}\} \cup \{I_{ij}^{r-kn-dst}\} \cup \{I_{ij}^{r-kn-proc}\}. \quad (4.13)$$

Завершение применения метода.

Данный метод универсален и может применяться для формирования представлений требований к ИС с точки зрения отдельного УАП как представителя Потребителя, а также с обобщенной точки зрения Поставщика.

Результаты выполнения метода формирования представлений требований к ИС на уровне информации являются исходными данными для группы методов формирования представлений требования к ИС на уровне знаний. Данная группа состоит из следующих методов [180]:

- а) метод формирования представления i -го ФТ к ИС на уровне знаний $K_i^f(U_{st_k})$ отдельного УАП st_k как представителя Потребителя;
- б) метод формирования представления i -го ФТ к ИС на уровне знаний K_i^{fU} Потребителя;
- в) метод формирования представления i -го ФТ к ИС на уровне знаний K_i^{fPr} Поставщика;
- г) метод формирования общесистемного представления i -го ФТ к ИС на уровне знаний K_i^{fIS} .

4.2.2 Метод формирования представления сформулированного требования на уровне знаний отдельного участника автоматизируемого процесса

На основании модели (3.81) был разработан метод формирования представления i -го ФТ к ИС на уровне знаний $K_i^f(U_{st_k})$ отдельного УАП st_k . Данный метод предполагает выполнение следующих этапов [180].

Этап 1. Выделить из множества $\{I_{ij}^{r-kn-dst}\}$ ранее не рассматривавшийся элемент $I_{ij}^{r-kn-dst}$.

Этап 2. Выполнить операцию формирования множества фреймов, описывающих выделенную структуру данных

$$d_n^{ij}(U_{st_k}) = d_{str}^{ij}, \quad (4.14)$$

где d_{str}^{ij} - значение атрибута, идентифицирующего j -ю структуру данных i -го ФТ к ИС.

Этап 3. Для каждого элемента подмножества $\{<d_{attr}^{ij}, d_{attr_t}^{ij}>\}$, для которого выполняется условие

$$\{<d_{attr}^{ij}, d_{attr_t}^{ij}>\} \in I_{ij}^{r-kn-dst} \neq \emptyset, \quad (4.15)$$

где d_{attr}^{ij} - значение атрибута, идентифицирующего атрибут j -й структуры данных i -го ФТ к ИС;

$d_{attr_t}^{ij}$ - значение атрибута, идентифицирующего тип атрибута j -й структуры данных i -го ФТ к ИС,

выполнить операцию формирования описания атрибутов фреймов

$$\{ \langle d_{el_fr}^{ij}(U_{st_k}), d_{el_fr_t}^{ij}(U_{st_k}) \rangle \} = \{ \langle d_{attr}^{ij}, d_{attr_t}^{ij} \rangle \} \quad (4.16)$$

и установить для подмножества $\{ \langle d_{el_fr}^{ij}(U_{st_k}), d_{el_fr_t}^{ij}(U_{st_k}) \rangle \}$ отношение принадлежности (3.82), после чего перейти к Этапу 5. В противном случае перейти к Этапу 4.

Этап 4. В диалоговом режиме для каждого $d_n^{ij}(U_{st_k})$ сформировать подмножество $\{ \langle d_{el_fr}^{ij}(U_{st_k}), d_{el_fr_t}^{ij}(U_{st_k}) \rangle \}$ и установить для этого подмножества отношение принадлежности (3.82).

Этап 5. Исключить элемент $I_{ij}^{r-kn-dst}$ из множества $\{ I_{ij}^{r-kn-dst} \}$. Если выполняется условие $\{ I_{ij}^{r-kn-dst} \} \neq \emptyset$, то перейти к Этапу 1. В противном случае перейти к Этапу 6.

Этап 6. В диалоговом режиме для каждого фрейма $\langle d_n^{ij}(U_{st_k}), \{ \langle d_{el_fr}^{ij}(U_{st_k}), d_{el_fr_t}^{ij}(U_{st_k}) \rangle \} \rangle$ установить наличие или отсутствие связей с другими фреймами, сформированными в результате выполнения операций (4.14) и (4.16). Если наличие связи подтверждается, то сформировать описание связи $\langle d_{fr_rel_n}^{ij}(U_{st_k}), \{ \langle d_{el_fr_rel}^{ij}(U_{st_k}), d_{el_fr_rel_t}^{ij}(U_{st_k}) \rangle \} \rangle$ и установить для этого описания отношение принадлежности (3.84).

Этап 7. Сформировать k -е представление i -го ФТ к ИС на уровне знаний $K_i^f(U_{st_k})$ в результате выполнения операции

$$K_i^f(U_{st_k}) = \sum_{j=1}^{r_{fr}} \langle d_n^{ij}(U_{st_k}), \{ \langle d_{el_fr}^{ij}(U_{st_k}), d_{el_fr_t}^{ij}(U_{st_k}) \rangle \} \rangle + \sum_{j=1}^{r_{rel}} \langle d_{fr_rel_n}^{ij}(U_{st_k}), \{ \langle d_{el_fr_rel}^{ij}(U_{st_k}), d_{el_fr_rel_t}^{ij}(U_{st_k}) \rangle \} \rangle, \quad (4.17)$$

где r_{fr} - количество фреймов, сформированных в результате выполнения
Этапов 2-5;

r_{rel} - количество связей, сформированных в результате выполнения Этапа 6,
после чего завершить применение метода.

Результатом применения данного метода является представление i -го
ФТ к ИС на уровне знаний $K_i^f(U_{st_k})$, отражающее локальную точку зрения
заинтересованного представителя Потребителя на сформулированное i -е ФТ.

4.2.3 Метод формирования представления сформулированного требования на уровне знаний Потребителя

Метод формирования представления i -го ФТ к ИС на уровне знаний K_i^{trU}
Потребителя выполняется после формирования множества представлений дан-
ного требования $K_i^f(U_{st_k})$ всех УАП, выдвинувших свои версии данного ФТ,
и предполагает выполнение следующих этапов [180].

Этап 1. Выделить множество представлений i -го ФТ к ИС на уровне зна-
ний отдельных УАП $\{K_i^f(U_{st_k})\}$.

Этап 2. Для каждого фрейма $\langle d_n^{ia}(U_{st_k}), \{ \langle d_{el_fr}^{ia}(U_{st_k}),$
 $d_{el_fr_t}^{ia}(U_{st_k}) \rangle \rangle$, $k = 1, \dots, st_{max}$, где st_{max} - количество УАП, выдвинув-
ших свои версии i -го ФТ, выполнить операцию

$$\begin{aligned}
& \exists (\langle d_n^{ia}(U_{st_k}), \{ \langle d_{el_fr}^{ia}(U_{st_k}), d_{el_fr_t}^{ia}(U_{st_k}) \rangle \} \rangle = \\
& = \langle d_n^{ib}(U_{st_m}), \{ \langle d_{el_fr}^{ib}(U_{st_m}), d_{el_fr_t}^{ib}(U_{st_m}) \rangle \} \rangle \Rightarrow \\
& \Rightarrow \langle d_n^{ju}, \{ \langle d_{el_fr}^{ju}, d_{el_fr_t}^{ju} \rangle \} \rangle = \langle d_n^{ia}(U_{st_k}), \{ \langle d_{el_fr}^{ia}(U_{st_k}), \\
& d_{el_fr_t}^{ia}(U_{st_k}) \rangle \} \rangle, a = 1, \dots, r_{fr}(U_{st_k}), b = 1, \dots, r_{fr}(U_{st_m}), \\
& a \neq b, m = 1, \dots, st_{max}, k \neq m, j = 1, 2, \dots,
\end{aligned} \tag{4.18}$$

где $r_{fr}(U_{st_k})$ - количество фреймов, присутствующих в представлении $K_i^f(U_{st_k})$;

$r_{fr}(U_{st_m})$ - количество фреймов, присутствующих в представлении $K_i^f(U_{st_m})$;

$\langle d_n^{ju}, \{ \langle d_{el_fr}^{ju}, d_{el_fr_t}^{ju} \rangle \} \rangle$ - описание фрейма, характеризующего j -й концепт ПрО с точки зрения Потребителя,

после чего исключить фреймы, для которых операция (4.18) была выполнена из соответствующих представлений $K_i^f(U_{st_k})$ и $K_i^f(U_{st_m})$.

Этап 3. Для каждого фрейма $\langle d_n^{ia}(U_{st_k}), \{ \langle d_{el_fr}^{ia}(U_{st_k}), d_{el_fr_t}^{ia}(U_{st_k}) \rangle \} \rangle$, $k = 1, \dots, st_{max}$, выполнить операцию

$$\begin{aligned}
& \exists (\langle d_n^{ia}(U_{st_k}), \{ \langle d_{el_fr}^{ia}(U_{st_k}), d_{el_fr_t}^{ia}(U_{st_k}) \rangle \} \rangle \subset \\
& \subset \langle d_n^{ju}, \{ \langle d_{el_fr}^{ju}, d_{el_fr_t}^{ju} \rangle \} \rangle \Rightarrow \langle d_g^{ju}, \{ \langle d_{el_if}^{ju}, d_{el_if_t}^{ju} \rangle \} \rangle = \\
& = \langle d_n^{ia}(U_{st_k}), \{ \langle d_{el_fr}^{ia}(U_{st_k}), d_{el_fr_t}^{ia}(U_{st_k}) \rangle \} \rangle, a = 1, \dots, r_{fr}(U_{st_k}), \\
& b = 1, \dots, r_{fr}(U_{st_m}), a \neq b, m = 1, \dots, st_{max}, k \neq m,
\end{aligned} \tag{4.19}$$

где $\langle d_g^{ju}, \{ \langle d_{el_if}^{ju}, d_{el_if_t}^{ju} \rangle \} \rangle$ - описание интерфейса, характеризующего

j -й концепт ПрО с точки зрения k -го

участника автоматизируемого процесса,

после чего исключить фреймы, для которых операция (4.19) была выполнена, из

представлений $K_i^f(U_{st_k})$.

Этап 4. Если подмножество $\{ \langle d_n^{ia}(U_{st_k}), \{ \langle d_{el_fr}^{ia}(U_{st_k}), d_{el_fr_t}^{ia}(U_{st_k}) \rangle \} \rangle \neq \emptyset$, то выполнить дополнительную операцию формирования описаний фреймов

$$\begin{aligned} & \langle d_g^{iju}, \{ \langle d_{el_if}^{iju}, d_{el_if_t}^{iju} \rangle \} \rangle = \langle d_n^{ia}(U_{st_k}), \{ \langle d_{el_fr}^{ia}(U_{st_k}), \\ & d_{el_fr_t}^{ia}(U_{st_k}) \rangle \} \rangle, a = 1, \dots, r_{fr}(U_{st_k}) \end{aligned} \quad (4.20)$$

для каждого элемента данного подмножества. В противном случае – перейти к Этапу 5.

Этап 5. Для каждого описания связи $\langle d_{fr_rel_n}^{ia}(U_{st_k}), \{ \langle d_{el_fr_rel}^{ia}(U_{st_k}), d_{el_fr_rel_t}^{ia}(U_{st_k}) \rangle \} \rangle$, где $a = 1, \dots, r_{rel}(U_{st_k})$, $k = 1, \dots, st_{max}$, выполнить операцию формирования описания связи

$$\begin{aligned} & \langle d_{fr_rel_n}^{iju}, \{ \langle d_{el_fr_rel}^{iju}, d_{el_fr_rel_t}^{iju} \rangle \} \rangle = \\ & = \langle d_{fr_rel_n}^{ia}(U_{st_k}), \{ \langle d_{el_fr_rel}^{ia}(U_{st_k}), d_{el_fr_rel_t}^{ia}(U_{st_k}) \rangle \} \rangle, \end{aligned} \quad (4.21)$$

с учетом результатов выполнения Этапов 2-4.

Этап 6. Сформировать представление i -го ФТ к ИС K_i^{trU} на уровне знаний Потребителя как результат выполнения операции

$$\begin{aligned}
K_i^{trU} = & \sum_{j=1}^{r_{fr}} \langle d_n^{iju}, \{ \langle d_{el_fr}^{iju}, d_{el_fr_t}^{iju} \rangle \} \rangle + \\
& + \sum_{j=1}^{r_{if}} \langle d_g^{iju}, \{ \langle d_{el_if}^{iju}, d_{el_if_t}^{iju} \rangle \} \rangle + \\
& + \sum_{j=1}^{r_{rel}} \langle d_{fr_rel_n}^{iju}, \{ \langle d_{el_fr_rel}^{iju}, d_{el_fr_rel_t}^{iju} \rangle \} \rangle,
\end{aligned} \tag{4.22}$$

где r_{fr} - количество описаний фреймов, сформированных в результате выполнения Этапов 2 и 4;

r_{if} - количество описаний интерфейсов, сформированных в результате выполнения Этапа 3;

r_{rel} - количество описаний связей, сформированных в результате выполнения Этапа 5,

после чего завершить применение метода.

Результатом применения данного метода является представление i -го ФТ к ИС на уровне знаний K_i^{trU} , отражающее согласованную точку зрения Потребителя на сформулированное i -е ФТ, выработанную на основе локальных точек зрения отдельных УАП.

4.2.4 Метод формирования представления сформулированного требования на уровне знаний Поставщика

Как было отмечено в подразд. 1.6, главной целью Поставщика является максимизация эффекта от ре-использования ранее разработанных ИТ-сервисов. Поэтому формирование и анализ представления i -го ФТ к ИС K_i^{fPr} на уровне знаний Поставщика предлагается проводить на основе библиотеки ранее реали-

зованных требований к ИТ-услугам созданных ИС и соответствующих этим требованиям ИТ-сервисах. Структура описания данной библиотеки также будет определяться ППТ к ИС. Поэтому представление некоего ранее реализованного i -го ФТ к ИС на уровне знаний $K_i^{f_{lib}}$ в данной библиотеке будет иметь вид, аналогичный (3.81) с учетом действующих отношений принадлежности, аналогичных (3.82) – (3.84).

Тогда формирование представления i -го ФТ к ИС $K_i^{f_{Pr}}$ на уровне знаний Поставщика следует рассматривать как процесс принятия решений о возможности реализации данного представления на основе элементов библиотеки ранее реализованных требований. Исходя из этого, метод формирования представления i -го ФТ к ИС $K_i^{f_{Pr}}$ на уровне знаний Поставщика предполагает выполнение следующих этапов [180].

Этап 1. Выделить из множества $\{I_{ij}^{r-kn-dst}\}$ ранее не рассматривавшийся элемент $I_{ij}^{r-kn-dst}$.

Этап 2. Для каждого элемента $I_{ij}^{r-kn-dst} \in \{I_{ij}^{r-kn-dst}\}$ выделить описания фреймов ранее реализованных ФТ к ИС, допустимые для ре-использования в представлении i -го ФТ к создаваемой ИС, путем выполнения операции

$$\begin{aligned} \exists (d_n^{ab_{lib}} = d_{str}^{ij}) \cup (\{ \langle d_{el_fr}^{ab_{lib}}, d_{el_fr_t}^{ab_{lib}} \rangle = \{ \langle d_{attr}^{ij}, d_{attr_t}^{ij} \rangle \}) \Rightarrow \\ \Rightarrow K_i^{f_{Pr}} = K_i^{f_{Pr}} + \langle d_n^{ab_{lib}}, \{ \langle d_{el_fr}^{ab_{lib}}, d_{el_fr_t}^{ab_{lib}} \rangle \} \rangle, \quad (4.23) \\ a = 1, \dots, r_{tr_{lib}}, b = 1, \dots, r_{if_{lib}}, \end{aligned}$$

где $\langle d_n^{ab_{lib}}, \{ \langle d_{el_fr}^{ab_{lib}}, d_{el_fr_t}^{ab_{lib}} \rangle \} \rangle$ - описание b -го фрейма в a -ом ранее реализованном ФТ, представление которого на уровне знаний присутствует в библиотеке;

r_{trlib} - количество представлений на уровне знаний ранее реализованных требований, присутствующих в библиотеке;

r_{iflib} - количество фреймов всех представлений на уровне знаний ранее реализованных требований, присутствующих в библиотеке,

после чего исключить элемент $I_{ij}^{r-kn-dst}$, для которого была выполнена операция (4.23), из рассмотрения в множестве $\{I_{ij}^{r-kn-dst}\}$.

Этап 3. Для каждого элемента $I_{ij}^{r-kn-dst} \in \{I_{ij}^{r-kn-dst}\}$ выделить описания интерфейсов ранее реализованных ФТ к ИС, допустимых для ре-использования в представлении i -го ФТ к создаваемой ИС, путем выполнения операции

$$\begin{aligned} \exists (d_g^{ablib} = d_{str}^{ij}) \cup (\{ \langle d_{el_if}^{ablib}, d_{el_if_t}^{ablib} \rangle = \{ \langle d_{attr}^{ij}, d_{attr_t}^{ij} \rangle \} \}) \Rightarrow \\ \Rightarrow K_i^{fPr} = K_i^{fPr} + \langle d_g^{ablib}, \{ \langle d_{el_if}^{ablib}, d_{el_if_t}^{ablib} \rangle \} \rangle, \quad (4.24) \\ a = 1, \dots, r_{trlib}, b = 1, \dots, r_{iflib}, \end{aligned}$$

где $\langle d_g^{ablib}, \{ \langle d_{el_if}^{ablib}, d_{el_if_t}^{ablib} \rangle \} \rangle$ - описание b -го интерфейса в a -ом ранее реализованном ФТ, представление которого на уровне знаний присутствует в библиотеке;

r_{iflib} - количество интерфейсов всех представлений на уровне знаний ранее реализованных требований, присутствующих в библиотеке,

после чего исключить элемент $I_{ij}^{r-kn-dst}$, для которого была выполнена операция (4.24), из рассмотрения в множестве $\{I_{ij}^{r-kn-dst}\}$.

Этап 4. Для каждого элемента $I_{ij}^{r-kn-dst} \in \{I_{ij}^{r-kn-dst}\}$ выделить избыточное описание фрейма ранее реализованного ФТ к ИС, допустимое для ре-использования в представлении i -го ФТ к создаваемой ИС, путем выполнения операции

$$\begin{aligned}
& \exists (\{ \langle d_{el_fr}^{ablib}, d_{el_fr_t}^{ablib} \rangle \supset \{ \langle d_{attr}^{ij}, d_{attr_t}^{ij} \rangle \}) \cap \\
& \cap (\{ \langle d_{el_fr}^{ablib}, d_{el_fr_t}^{ablib} \rangle / \{ \langle d_{attr}^{ij}, d_{attr_t}^{ij} \rangle \} \rightarrow \emptyset) \Rightarrow \\
& \Rightarrow K_i^{fPr} = K_i^{fPr} + \langle d_n^{ablib}, \{ \langle d_{el_fr}^{ablib}, d_{el_fr_t}^{ablib} \rangle \} \rangle, \\
& \quad a = 1, \dots, r_{trlib}, b = 1, \dots, r_{frlib},
\end{aligned} \tag{4.25}$$

после чего исключить элемент $I_{ij}^{r-kn-dst}$, для которого была выполнена операция (4.25), из рассмотрения в множестве $\{ I_{ij}^{r-kn-dst} \}$.

Этап 5. Для каждого фрейма $\langle d_n^{ijlib}, \{ \langle d_{el_fr}^{ijlib}, d_{el_fr_t}^{ijlib} \rangle \} \rangle \in K_i^{fPr}$ и каждого интерфейса $\langle d_g^{ijlib}, \{ \langle d_{el_if}^{ijlib}, d_{el_if_t}^{ijlib} \rangle \} \rangle \in K_i^{fPr}$ проверить существование в библиотеке описаний связи генерализации (наследования), связывающей данный фрейм или интерфейс с фреймом $\langle d_n^{ablib}, \{ \langle d_{el_fr}^{ablib}, d_{el_fr_t}^{ablib} \rangle \} \rangle \notin K_i^{fPr}$. Если описание такой связи в библиотеке существует, то выполнить операцию

$$K_i^{fPr} = K_i^{fPr} + \langle d_{fr_rel_n}^{ablib}, \{ \langle d_{el_fr_rel}^{ablib}, d_{el_fr_rel_t}^{ablib} \rangle \} \rangle, \tag{4.26}$$

после чего повторять Этап 5 до тех пор, пока в представлении K_i^{fPr} не останутся нерассмотренных описаний фреймов и интерфейсов.

Этап 6. Сформировать представление i -го ФТ к создаваемой ИС K_i^{fPr} путем включения в него множества описаний связей между ре-используемыми фреймами и/или интерфейсами посредством выполнения для каждого из описаний фреймов и/или интерфейсов представления K_i^{fPr} операции

$$\begin{aligned}
& \exists(((d_n^{xylib} = d_n^{iaPr}) \cap (d_n^{xzlib} = d_n^{ibPr})) \cup ((d_n^{xylib} = d_n^{iaPr}) \cap \\
& \quad \cap (d_g^{xqlib} = d_g^{ibPr})) \cap (d_{fr_rel_n}^{xclib} \in K_x^{flib})) \Rightarrow \\
& \Rightarrow K_i^{fPr} = K_i^{fPr} + \langle d_{fr_rel_n}^{xclib}, \{ \langle d_{el_fr_rel}^{xclib}, d_{el_fr_rel_t}^{xclib} \rangle \} \rangle, \\
& \quad x = 1, \dots, r_{trlib}, y, z = 1, \dots, r_{frlib}, y \neq z, q = 1, \dots, r_{iflib},
\end{aligned} \tag{4.27}$$

где $\langle d_{fr_rel_n}^{xclib}, \{ \langle d_{el_fr_rel}^{xclib}, d_{el_fr_rel_t}^{xclib} \rangle \} \rangle$ - описание c -ой связи в x -ом ранее реализованном ФТ, представление которого на уровне знаний присутствует в библиотеке,

после чего завершить применение метода.

Результатом применения данного метода является представление i -го ФТ к ИС на уровне знаний K_i^{fPr} , отражающее точку зрения Поставщика на сформулированное i -е ФТ, выработанную на основе опыта Поставщика по реализации схожих ФТ в ранее выполненных ИТ-проектах создания ИС.

4.2.5 Метод формирования общесистемного представления сформулированного требования на уровне знаний

Общесистемное представление i -го ФТ к ИС на уровне знаний K_i^{fIS} предназначено для согласования точек зрения Поставщика и Потребителя на i -ю ИТ-услугу, к которой выдвигается требование. Поэтому метод формирования общесистемного представления i -го ФТ к ИС на уровне знаний K_i^{fIS} предполагает выполнение следующих этапов [180].

Этап 1. Выделить представления K_i^{fU} и K_i^{fPr} , отражающие точки зрения

на i -е ФТ Потребителя и Поставщика соответственно.

Этап 2. Включить в представление K_i^{fIS} описания фреймов, интерфейсов и связей, выделенных Потребителем, которые могут быть повторно реализованы Поставщиком, путем выполнения операций

$$\begin{aligned} \exists(\{ \langle d_{el_fr}^{ijU}, d_{el_fr_t}^{ijU} \rangle \} \subseteq \{ \langle d_{el_fr}^{ibPr}, d_{el_fr_t}^{ibPr} \rangle \}) &\Rightarrow \\ \Rightarrow K_i^{fIS} = K_i^{fIS} + \langle d_n^{ibPr}, \{ \langle d_{el_fr}^{ibPr}, d_{el_fr_t}^{ibPr} \rangle \} \rangle; &\quad (4.28) \end{aligned}$$

$$\begin{aligned} \exists(\{ \langle d_{el_if}^{ijU}, d_{el_if_t}^{ijU} \rangle \} \subseteq \{ \langle d_{el_if}^{ibPr}, d_{el_if_t}^{ibPr} \rangle \}) &\Rightarrow \\ \Rightarrow K_i^{fIS} = K_i^{fIS} + \langle d_g^{ibPr}, \{ \langle d_{el_if}^{ibPr}, d_{el_if_t}^{ibPr} \rangle \} \rangle; &\quad (4.29) \end{aligned}$$

$$\begin{aligned} \exists(\{ \langle d_{el_fr_rel}^{ijU}, d_{el_fr_rel_t}^{ijU} \rangle \} = \{ \langle d_{el_fr_rel}^{ibPr}, d_{el_fr_rel_t}^{ibPr} \rangle \}) &\Rightarrow \\ \Rightarrow K_i^{fIS} = K_i^{fIS} + \langle d_{fr_rel_n}^{ibPr}, \{ \langle d_{el_fr_rel}^{ibPr}, d_{el_fr_rel_t}^{ibPr} \rangle \} \rangle. &\quad (4.30) \end{aligned}$$

Этап 3. Дополнить общесистемное представление K_i^{fIS} описаниями фреймов, интерфейсов и связей, которые необходимы с точки зрения Поставщика, путем выполнения операции

$$K_i^{fIS} = K_i^{fIS} \cup K_i^{fPr}. \quad (4.31)$$

Этап 4. Дополнить общесистемное представление K_i^{fIS} описаниями фреймов, интерфейсов и связей, которые необходимы с точки зрения Потребителя и не соответствуют условиям выполнения операций (4.28)-(4.30), путем выполнения операции

$$K_i^{fIS} = K_i^{fIS} \cup K_i^{fU}, \quad (4.32)$$

после чего завершить применение метода.

Разработанные методы позволяют автоматизировать процессы формирования и анализа множества СТ к ИС, что, в свою очередь, позволит значительно сократить затраты времени на выполнение предпроектного обследования ОА. Результаты выполнения данных методов – представления K_i^{fU} , K_i^{fPr} и K_i^{fIS} – являются основными описаниями ФТ к ИС, позволяющими формализовать и автоматизировать решение задачи синтеза ОРА создаваемой ИС.

4.3 Метод синтеза вариантов описаний архитектуры создаваемой информационной системы

Общесистемные представления ФТ к ИС на уровне знаний K_i^{fIS} , формируемые в результате выполнения рассмотренных в подразд. 4.1 методов, являются исходной информацией для синтеза АД создаваемой системы. Однако перед началом синтеза этих АД следует провести анализ СФТ с целью выявления их несоответствий или дублирования друг друга.

Последний недостаток становится особенно значимым в свете применения предлагаемого в подразд. 2.1 сервисного подхода к созданию ИС. В соответствии с данным подходом каждое ФТ описывает конкретную ИТ-услугу. Поэтому дублирование ФТ приводит к необходимости тратить время и ресурсы на исследование и разработку одной и той же ИТ-услуги как множества различных ИТ-услуг. Следствием этого является необходимость перепроектирования ИС с целью исправления допущенных ошибок и снижение эффекта от эксплуатации ИС из-за дублирования ИТ-услуг. В то же время определенная степень дублирования требований может быть необходимой с точки зрения БП Потребителя. Поэтому в

ходе осуществления синтеза АД создаваемой ИС возникает необходимость подвергнуть объективному анализу все приемлемые варианты АД. Под приемлемыми здесь и в дальнейшем следует понимать такие варианты АД, которые будут не хуже по одному или нескольким критериям исходного варианта АД.

Исходным вариантом АД будем называть сеть фреймов $Arch_{base}$, задаваемую паттерном (3.61) с учетом выполнения условия (3.62), и формируемую следующим образом [184, 185]:

$$Arch_{base} = \bigcup_{i=c+1}^e K_i^{fIS} . \quad (4.33)$$

Сеть фреймов $Arch_{base}$ описывает вариант АД создаваемой ИС, в котором каждое ФТ будет описывать отдельную ИТ-услугу (для данного варианта считается, что все ФТ сформулированы правильно).

Для выявления других возможных вариантов АД ИТ-услуг создаваемой ИС следует учитывать такие особенности формируемых сетей фреймов:

а) в соответствии с множеством паттернов (3.63) описания формируемых сетей фреймов предполагается сохранять в специализированном хранилище данных как совокупность транзакций, каждая из которых описывает отдельное представление K_i^{fIS} ;

б) K_i^{fIS} представляет собой набор категориальных данных (формализованные структурированные описания фреймов, интерфейсов и связей), числовые представления которых затруднено;

в) оценка степени дублирования представлений K_i^{fIS} должна основываться на оценке количества повторений описаний элементов сети фреймов в различных представлениях;

г) выявление дублирования ФТ должно приводить к синтезу новых вариантов АД, в которых описания новых ИТ-услуг формируются путем группировки

исходных представлений $K_i^{f_{IS}}$.

Исходя из отмеченных особенностей, задачу синтеза вариантов АД создаваемой ИС следует рассматривать как задачу кластеризации категориальных данных, алгоритм решения которой основан на оптимизации глобального критерия. В качестве такого алгоритма предлагается использовать алгоритм CLOPE [184, 185]. Применение этого алгоритма позволяет формализовать задачу синтеза вариантов АД создаваемой ИС следующим образом.

Пусть имеется сеть фреймов (4.33), состоящая из множества представлений $K_i^{f_{IS}}$. Каждое представление $K_i^{f_{IS}}$ является набором объектов ob (фреймов, интерфейсов и связей), описание которого аналогично (3.81). Множество искомых описаний ИТ-услуг IT_{act} есть разбиение сети фреймов (4.33), такое, что $\{IT_{act_1}, \dots, IT_{act_k}\} = Arch_{base}$ и $IT_{act_i} \neq \emptyset \wedge IT_{act_i} \cap IT_{act_j} = \emptyset$ для $1 \leq i, j \leq k$, где k - количество ИТ-услуг создаваемой ИС. Каждая ИТ-услуга IT_{act_j} имеет следующие характеристики:

а) $D|IT_{act}|$ - множество уникальных объектов;

б) $Occ(ob, IT_{act_j})$ - количество вхождений объекта ob в описание ИТ-

услуги IT_{act_j} ;

в) $S(IT_{act_j}) = \sum_{ob \in D(IT_{act_j})} Occ(ob, IT_{act_j})$;

г) $W(IT_{act_j}) = |D(IT_{act_j})|$;

д) $H(IT_{act_j}) = S(IT_{act_j}) / W(IT_{act_j})$.

Тогда расчет функции стоимости ИТ-услуги $Profit(IT_{act}, r)$ в соответствии с положениями алгоритма CLOPE будет проводиться по формуле [186, 187]:

$$Profit(IT_{act}, r) = \sum_{j=1}^k \frac{S(IT_{act_j})}{W(IT_{act_j})^r} \times |IT_{act_j}| \bigg/ \sum_{j=1}^k |IT_{act_j}|, \quad (4.34)$$

где $|IT_{actj}|$ - количество общесистемных представлений ФТ на уровне зна-
ний K_i^{fIS} , описывающих ИТ-услугу IT_{actj} .

Тогда постановку задачи синтеза вариантов АД создаваемой ИС можно сформулировать следующим образом: для заданных $D(Arch_{base})$ и r найти разбиение: $IT_{act} : Profit(IT_{act}, r) \rightarrow max$.

Однако не следует забывать, что множество СТ практически никогда не описывает создаваемую ИС полностью (см. подразд. 2.3). Поэтому постановку задачи синтеза вариантов АД создаваемой ИС следует модифицировать таким образом: для заданных $D(Arch_{base})$ и r найти разбиения: $IT_{act} : Profit(IT_{act}, r) \in [Profit_{max} - \varepsilon; Profit_{max}]$, где $Profit_{max}$ - максимальное значение функции (4.34), ε - величина допустимой погрешности. В случае, если величину ε не удастся определить с помощью методов экспертных оценок, рекомендуется считать $\varepsilon = 0,1 \times Profit_{max}$ [184, 185].

Точное определение значения коэффициента отталкивания r требует проведения специальных исследований успешных ИТ-проектов создания ИС. На основании результатов исследований законов композиции ФС ИС, рассмотренных в [188], предложено выбирать значение r , исходя из приведенных в табл. 4.1 данных [184, 185].

Таблица 4.1 – Зависимость значения коэффициента отталкивания r от вариантов организации архитектуры информационной системы, создаваемой как совокупность ИТ-услуг

Вариант организации архитектуры информационной системы	Значение коэффициента отталкивания r
Функциональная (подсистемы)	1,5
Модульная	2
Сервисная	≥ 3

Для решения поставленной задачи алгоритм CLOPE необходимо модифицировать. В частности, следует изменить действия по инициализации разбиения набора транзакций на кластеры, поскольку, как показано выше, исходный вариант AD создаваемой ИС и, соответственно, разделение представлений K_i^{fIS} между ИТ-услугами IT_{act_j} уже определены.

Результатом данной модификации является метод синтеза вариантов AD создаваемой ИС, предполагающий выполнение следующих этапов [184, 185].

Этап 1. Сформировать исходный вариант AD создаваемой ИС $Arch_{base}$.

Шаг 1.1. Принять $n = |K_i^{fIS}|$.

Шаг 1.2. Сформировать множество описаний ИТ-услуг IT_{act} путем выполнения операции $IT_{act_j} = K_i^{fIS}$.

Шаг 1.3. Для множества представлений $\{K_i^{fIS}\}$ выполнить операцию (4.33).

Этап 2. Установить значение коэффициента отталкивания r и рассчитать значение функции $Profit(IT_{act}, r)$ по выражению (4.34).

Этап 3. Провести синтез оптимальных и/или приемлемых вариантов AD создаваемой ИС.

Шаг 3.1. Принять $Profit_{max} = Profit(IT_{act}, r)$, $i = 1$, $j = 1$, $k = |IT_{act}|$.

Шаг 3.2. Выбрать представление K_i^{fIS} .

Шаг 3.3. Если $K_i^{fIS} \in IT_{act_j}$, то исключить K_i^{fIS} из IT_{act_j} . В противном случае перейти к Шагу 3.13.

Шаг 3.4. Принять $z = 0$, $m = 0$.

Шаг 3.5. Если $z = j$, то принять $z = z + 1$.

Шаг 3.6. Если $z > k$, то перейти к Шагу 3.12.

Шаг 3.7. Включить K_i^{fIS} в IT_{act_z} .

Шаг 3.8. Рассчитать значение $Profit(IT_{act}, r)$ по выражению (4.34).

Шаг 3.9. Если $Profit(IT_{acm}, r) > Profit_{max}$, то зафиксировать вариант АД создаваемой ИС как множество IT_{acm} , скорректированное с учетом выполнения Шага 3.3 и Шага 3.7, принять $m = 1$ и перейти к Шагу 3.12.

Шаг 3.10. Если $Profit(IT_{acm}, r) \in [Profit_{max} - \varepsilon; Profit_{max}]$, то зафиксировать вариант АД создаваемой ИС как множество IT_{acm} , скорректированное с учетом выполнения Шага 3.3 и Шага 3.7.

Шаг 3.11. Принять $z = z + 1$. Если $z \leq k$, то перейти к Шагу 3.5.

Шаг 3.12. Если $IT_{acm_j} = \emptyset$ и $m = 1$, то исключить IT_{acm_j} из множества IT_{acm} и принять $k = k - 1$.

Шаг 3.13. Принять $j = j + 1$. Если $j \leq k$, то перейти к Шагу 3.3.

Шаг 3.14. Принять $i = i + 1$. Если $i \leq n$, то перейти к Шагу 3.2.

Шаг 3.15. Если $m = 1$, то принять $i = 1$ и $j = 1$, после чего перейти к Шагу 3.2. В противном случае завершить выполнение этапа метода.

Этап 4. Исключить из рассмотрения все зафиксированные на Этапе 3 варианты АД создаваемой ИС IT_{acm} , для которых не выполняется условие $Profit(IT_{acm}, r) \in [Profit_{max} - \varepsilon; Profit_{max}]$. Завершить применение метода.

Результатом применения метода синтеза вариантов АД создаваемой ИС является множество приемлемых вариантов АД, в которых степень дублирования описаний ПрО в ФТ сведена до необходимого минимума. Каждый из этих вариантов АД представляет собой множество ИТ-услуг IT_{acm} . В свою очередь, каждая из ИТ-услуг IT_{acm_j} является результатом объединения, аналогичного (4.33), отдельных представлений K_i^{fIS} , включаемых в описание данной ИТ-услуги по результатам выполнения Этапа 3 разработанного метода.

Разработанный метод, в отличие от стандартного алгоритма CLOPE, позволяет выделять для последующего анализа варианты АД создаваемой ИС, которые лишь незначительно хуже оптимального, что позволяет значительно расширить область поиска компромиссного варианта АД создаваемой ИС, удовлетворяющего как Поставщика, так и Потребителя.

4.4 Модель и методы синтеза описания рациональной архитектуры информационной системы

Множество приемлемых вариантов АД создаваемой ИС, формируемые в результате выполнения метода синтеза вариантов АД, являются исходной информацией для решения сформулированной в подразд. 4.1 задачи синтеза ОРА ИС. Поэтому постановка данной задачи (4.7)-(4.12) нуждается в уточнении. Данное уточнение заключается в определении конкретного способа расчета степени удовлетворения требований Поставщика и Потребителя от реализации ИС по синтезированному ОРА.

Использование общесистемных представлений ФТ к ИС на уровне знаний K_i^{fIS} позволяет рассматривать операторы $r^{Pr}(K_i^{fPr})$ и $r^U(K_i^{fU})$ целевых функций (4.7) и (4.10) как функции, сравнивающие общесистемное представление i -го ФТ на уровне знаний и представление данного ФТ на уровне знаний Поставщика и Потребителя соответственно [184, 185, 189]. В общем случае такие функции будут иметь вид:

$$r^{Pr}(K_i^{fPr}) = f(K_i^{fIS}, K_i^{fPr}) = 1 - \left| K_i^{fIS} / K_i^{fPr} \right| \left| K_i^{fIS} \right|, \quad (4.35)$$

$$r^U(K_i^{fU}) = f(K_i^{fIS}, K_i^{fU}) = 1 - \left| K_i^{fIS} / K_i^{fU} \right| \left| K_i^{fIS} \right|. \quad (4.36)$$

Тогда целевая функция (4.7) примет вид [185, 190]

$$F_{Pr} = \sum_{i=c+1}^e 1 - \left| K_i^{fIS} / K_i^{fPr} \right| \left| K_i^{fIS} \right| \rightarrow \max, \quad (4.37)$$

а целевая функция (4.10) примет вид [185, 190]

$$F_U = \sum_{i=c+1}^e 1 - \left| K_i^{f_{IS}} / K_i^{f_U} \right| / \left| K_i^{f_{IS}} \right| \rightarrow \max. \quad (4.38)$$

Достижение цели (4.37) Поставщиком возможно в результате выполнения следующих действий [134]:

- разработка новых IT-услуг и IT-сервисов, удовлетворяющих требования, выдвигаемые Потребителями;
- адаптация разработанных ранее IT-услуг и IT-сервисов под особенности требований конкретных Потребителей.

Достижение цели (4.38) Потребителем возможно в результате выполнения следующих действий [134]:

- формулировка требований к ИС, IT-услугам и IT-сервисам, учитывающим общие и индивидуальные особенности БП ОА;
- анализ и выбор IT-услуг и IT-сервисов, предлагаемых Поставщиками и соответствующих СТ.

При этом между Поставщиком и Потребителем в процессе проектирования архитектуры ИС в общем случае практически отсутствуют какие-либо формы предварительной договоренности, ограничивающие выбор Потребителя или же разработки Поставщика. Это позволяет представить процесс проектирования архитектуры ИС как рационального набора IT-услуг в виде игры Поставщика и Потребителя [191, 192]. Эта игра будет иметь следующие отличительные особенности [185, 189-194]:

- по количеству игроков – игра двух лиц (Поставщика и Потребителя);
- по количеству стратегий – конечная игра, число чистых стратегий которой ограничено числом отдельных IT-услуг;
- по типу взаимоотношений игроков – бескоалиционная игра, до окончания которой Поставщик и Потребитель не могут действовать совместно (вступать в соглашения, передавать информацию, ресурсы и пр.);
- по характеру выигрышей – игра с нулевой суммой (сумма выигрышей Поставщика и Потребителя в каждой партии игры равняется нулю; общий их

капитал перераспределяется между ними зависимо от полученных результатов);

– по виду функции выигрыша – биматричная игра, поскольку выигрыши Поставщика и Потребителя различны и определяются, как следует из (4.37) и (4.38), различными функциями выигрыша.

В общем случае такая игра для Поставщика и Потребителя будет иметь вид [134, 181, 184, 185, 190, 191]:

$$\Gamma_{IS} = \langle \{Pr, U\}, \{X^{\{Pr, U\}}\}, \{f^{\{Pr, U\}}\} \rangle, \quad (4.39)$$

где Γ_{IS} – обозначение игры Поставщика и Потребителя;

$\{Pr, U\}$ – множество игроков, участвующих в игре Γ_{IS} (Поставщик и Потребитель);

$\{X^{\{Pr, U\}}\}$ – множество стратегий игры Γ_{IS} , формируемое на основе множества вариантов АД создаваемой ИС IT_{act} ;

$\{f^{\{Pr, U\}}\}$ – множество функций выигрыша игры Γ_{IS} , состоящее из выражений (4.37) и (4.38).

Физический смысл функции выигрыша Поставщика (4.37) заключается в максимизации ре-использования в IT-проекте создания ИС IT-сервисов, реализующих СФТ к этой системе. В этом случае основными видом IT-проектов создания ИС с точки зрения Поставщика будет адаптация типовых ИС, IT-услуг и реализующих эти услуги IT-сервисов к особенностям процессов Потребителя. Создание ИС «с нуля» будет рассматриваться Поставщиком как частный, наименее желаемый случай IT-проекта, который, однако, после успешного завершения может существенно расширить возможности Поставщика.

С точки зрения Потребителя, основной эффект от выполнения IT-проекта создания требуемой ИС и, следовательно, основной выигрыш, описываемый функцией (4.38), достигается в случае максимизации выполнения требований к ИС, выдвинутых Потребителем и принятых к реализации Поставщиком. При этом Потребителя интересует в первую очередь реализация описаний онтоло-

гий именно тех терминов ПрО, которые характерны для его процессов.

Тогда матрица выигрыша Поставщика будет иметь вид [185, 190]

$$Pr = \begin{pmatrix} 1 - \frac{|K_{l(c+1)}^{fIS} / K_{l(c+1)}^{fPr}|}{|K_{l(c+1)}^{fIS}|} & \dots & 1 - \frac{|K_{li}^{fIS} / K_{li}^{fPr}|}{|K_{li}^{fIS}|} & \dots & 1 - \frac{|K_{le}^{fIS} / K_{le}^{fPr}|}{|K_{le}^{fIS}|} \\ \dots & \dots & \dots & \dots & \dots \\ 1 - \frac{|K_{j(c+1)}^{fIS} / K_{j(c+1)}^{fPr}|}{|K_{j(c+1)}^{fIS}|} & \dots & 1 - \frac{|K_{ji}^{fIS} / K_{ji}^{fPr}|}{|K_{ji}^{fIS}|} & \dots & 1 - \frac{|K_{je}^{fIS} / K_{je}^{fPr}|}{|K_{je}^{fIS}|} \\ \dots & \dots & \dots & \dots & \dots \\ 1 - \frac{|K_{k(c+1)}^{fIS} / K_{k(c+1)}^{fPr}|}{|K_{k(c+1)}^{fIS}|} & \dots & 1 - \frac{|K_{ki}^{fIS} / K_{ki}^{fPr}|}{|K_{ki}^{fIS}|} & \dots & 1 - \frac{|K_{ke}^{fIS} / K_{ke}^{fPr}|}{|K_{ke}^{fIS}|} \end{pmatrix}. \quad (4.40)$$

Матрица выигрыша Потребителя будет иметь вид [185, 190]:

$$U = \begin{pmatrix} 1 - \frac{|K_{l(c+1)}^{fIS} / K_{l(c+1)}^{fU}|}{|K_{l(c+1)}^{fIS}|} & \dots & 1 - \frac{|K_{li}^{fIS} / K_{li}^{fU}|}{|K_{li}^{fIS}|} & \dots & 1 - \frac{|K_{le}^{fIS} / K_{le}^{fU}|}{|K_{le}^{fIS}|} \\ \dots & \dots & \dots & \dots & \dots \\ 1 - \frac{|K_{j(c+1)}^{fIS} / K_{j(c+1)}^{fU}|}{|K_{j(c+1)}^{fIS}|} & \dots & 1 - \frac{|K_{ji}^{fIS} / K_{ji}^{fU}|}{|K_{ji}^{fIS}|} & \dots & 1 - \frac{|K_{je}^{fIS} / K_{je}^{fU}|}{|K_{je}^{fIS}|} \\ \dots & \dots & \dots & \dots & \dots \\ 1 - \frac{|K_{k(c+1)}^{fIS} / K_{k(c+1)}^{fU}|}{|K_{k(c+1)}^{fIS}|} & \dots & 1 - \frac{|K_{ki}^{fIS} / K_{ki}^{fU}|}{|K_{ki}^{fIS}|} & \dots & 1 - \frac{|K_{ke}^{fIS} / K_{ke}^{fU}|}{|K_{ke}^{fIS}|} \end{pmatrix}. \quad (4.41)$$

Здесь k - количество вариантов АД создаваемой ИС, сформированных в результате применения метода синтеза вариантов АД создаваемой ИС.

Строки матрицы выигрыша Поставщика (4.40) определяют значения функции удовлетворения Поставщика от реализации совокупности ИТ-услуг j -го варианта АД ИС. Столбцы матрицы выигрыша Поставщика (4.40) определяют зна-

чения функции удовлетворения Поставщика от реализации i -го ФТ к ИС в различных вариантах АД данной ИС. Строки матрицы выигрыша Потребителя (4.41) определяют значения функции удовлетворения Потребителя от реализации совокупности ИТ-услуг j -го варианта АД ИС. Столбцы матрицы выигрыша Потребителя (4.41) определяют значения функции удовлетворения Потребителя от реализации i -го ФТ к ИС в различных вариантах АД данной ИС [185, 190].

В случае, если все описания фреймов, интерфейсов и связей, характеризующие ПрО в i -ом ФТ к ИС и используемые для формирования представления K_{ji}^{fIS} , будут взяты из библиотеки ранее реализованных требований к ИТ-услугам ИС, выигрыш Поставщика от реализации i -го ФТ в j -ом варианте АД ИС будет равен 1. В случае, если все описания фреймов, интерфейсов и связей, характеризующие ПрО в i -ом ФТ к ИС и используемые для формирования представления K_{ji}^{fIS} , являются для Поставщика новыми, то выигрыш Поставщика от реализации i -го ФТ в j -ом варианте АД ИС будет равен 0. В случае, если новой для Поставщика является только часть описаний этих фреймов, интерфейсов или связей, либо новые описания фреймов и интерфейсов базируются на присутствующих в библиотеке описаниях фреймов, интерфейсов или связей, характеризующих абстрактные термины ПрО, выигрыш Поставщика от реализации i -го ФТ в j -ом варианте АД ИС будет находиться в промежутке между 0 и 1.

В случае, если все описания фреймов, интерфейсов и связей, характеризующие ПрО в i -м ФТ к ИС и используемые для формирования представления K_{ji}^{fIS} , будут отражать только ПрО Потребителя, выигрыш Потребителя от реализации i -го ФТ в j -ом варианте АД ИС будет равен 1. Если все описания фреймов, интерфейсов и связей, характеризующие ПрО в i -м ФТ к ИС и используемые для формирования представления K_{ji}^{fIS} , являются для Потребителя новыми, то выигрыш Потребителя от реализации i -го ФТ в j -ом варианте АД ИС будет равен 0. Если же новой для Потребителя является только часть

описаний этих фреймов, интерфейсов или связей, либо новые описания фреймов и интерфейсов абстрагируют сформулированные Потребителем фреймы, интерфейсы или связи, выигрыш Потребителя от реализации i -го ФТ в j -ом варианте АД ИС будет находиться в промежутке между 0 и 1.

Результатом игры (4.39) будет являться ОРА ИС как набора ИТ-услуг, сформированное в результате нахождения исхода данной игры.

Для нахождения исхода игры (4.39) предлагается использовать метод поиска равновесий Нэша для биматричной игры в чистых стратегиях. Данный метод направлен на поиск ситуации равновесия по Нэшу – таких стратегий, уклоняясь от которых ни Поставщик, ни Потребитель не смогут увеличить значение своих функций выигрыша.

Для определения ситуации равновесия по Нэшу в ходе синтеза ОРА ИС уточним описания стратегий игры (4.39). Как следует из описания целевых функций (4.37) и (4.38), в общем случае данные стратегии определяются множеством общесистемных представлений ФТ к ИС на уровне знаний, формирующих варианты АД создаваемой ИС. Следовательно, любую стратегию $X_j^{\{Pr,U\}} \in \{X\}^{\{Pr,U\}}$ можно представить как сеть фреймов, представляющую собой результат выполнения операции (4.33) над множеством общесистемных представлений $\{K_{ji}^{f_{IS}}\}_{i=(c+1),\dots,e}^{\{Pr,U\}}$.

Тогда ситуация равновесия по Нэшу в игре (4.39) будет представлять собой один или несколько вариантов АД создаваемой ИС $\{K_{ji}^{f_{IS}}\}_{i=(c+1),\dots,e}^{\{Pr,U\}}$, для которых выполняется условие [184, 185]

$$\begin{aligned} & \{K_{ji}^{f_{IS}}\}_i^{\{Pr,U\}} \in \\ & \in \operatorname{argmax}_{\{K_{ji}^{f_{IS}}\}_i^{\{Pr,U\}} \in \{\{K_{li}^{f_{IS}}\}, \dots, \{K_{li}^{f_{IS}}\}\}_i^{\{Pr,U\}}} f^{\{Pr,U\}}(\{K_{-ji}^{f_{IS}}\}_i^{\{Pr,U\}} \parallel \{K_{ji}^{f_{IS}}\}_i^{\{Pr,U\}}), \end{aligned} \quad (4.42)$$

где $j = 1, \dots, k$, $i = (c + 1), \dots, e$.

Исходя из данного условия, метод поиска равновесий Нэша в чистых стратегиях

биматричной игры (4.39) предполагает выполнение следующих этапов [184, 185].

Этап 1. Сформировать матрицу выигрышей Потребителя (4.41) и матрицу выигрышей Поставщика (4.40).

Этап 2. Отметить максимальные элементы в каждом столбце матрицы (4.41). Если в некотором столбце данной матрицы несколько максимальных элементов, то необходимо отметить все такие элементы.

Этап 3. Отметить максимальные элементы в каждой строке матрицы (4.40). Если в некоторой строке данной матрицы несколько максимальных элементов, то необходимо отметить все такие элементы.

Этап 4. Провести поиск пересечения результатов Этапа 2 и Этапа 3 (то есть максимальных элементов, которые находятся на одних и тех же позициях в каждой из матриц).

Этап 5. Если результат выполнения Этапа 4 не является пустым множеством, зафиксировать соответствующие варианты АД создаваемой ИС как ОРА данной системы $\{K_{ji}^{f_{IS}}\}_{i=(c+1), \dots, e}^{\{Pr, U\}}$. В противном случае признать, что ОРА ИС на чистых стратегиях не существует. Завершить применение метода.

Ситуация, когда множество элементов, отмеченных в матрице (4.41) (максимальных по столбцам), и множество элементов, отмеченных в матрице (4.40) (максимальных по строкам), не пересекаются, означает, что равновесий в чистых стратегиях для данного варианта игры не существует. В этом случае предлагается рассматривать множество смешанных стратегий, поскольку равновесие Нэша существует для смешанного расширения любой конечной биматричной игры (4.39). В смешанном расширении биматричной игры всегда существует хотя бы одно равновесие по Нэшу. В ходе игры Поставщик и Потребитель должны использовать с положительными вероятностями только чистые стратегии, что позволяет свести поиск равновесий в смешанных стратегиях к решению систем линейных уравнений и неравенств. Для этого целесообразно использовать стандартные методы и алгоритмы, в частности – алгоритм Лемке – Хаусона [195].

Недостатком применения метода поиска равновесий Нэша в чистых стратегиях биматричной игры (4.39) следует считать увеличение количества рисков принятия решения. Однако для рассмотренной задачи синтеза ОРА ИС как совокупности ИТ-услуг данный недостаток не является критичным, поскольку процесс синтеза ОРА предполагает координацию действий Поставщика и Потребителя после исхода игры с целью согласования и утверждения документов ИТ-проекта на создание ИС.

Использование рассмотренных в данном подразделе теоретико-игровых моделей, методов и алгоритмов поиска выигрышных чистых и смешанных стратегий должно обеспечить Поставщику и Потребителю в ходе игры выбор такого варианта конфигурации ИТ-услуг, который:

- а) соответствует выдвинутым Потребителем требованиям в максимальной степени;
- б) приносит Поставщику наибольший из возможных доходов от использования ИТ-услуг и соответствующих ИТ-сервисов;
- в) обеспечивает реализацию непрерывных технологических цепочек сбора, хранения и обработки данных в рамках единой целостной ИС;
- г) гарантирует отсутствие ИТ-услуг и ИТ-сервисов, которые не являются обязательными в технологических цепочках сбора, хранения и обработки данных и не соответствуют выдвинутым требованиям.

Таким образом, использование теоретико-игрового представления взаимодействия Поставщика и Потребителя позволяет формализовать процедуры формирования и корректировки такой архитектуры ИС как совокупности ИТ-услуг, которая обеспечивает приемлемую для каждой стороны степень достижения глобальных целей.

4.5 Оценка объема работ по реализации описания рациональной архитектуры информационной системы

В соответствии с постановкой задачи синтеза ОРА создаваемой ИС (см. подразд. 4.1), результат игры (4.39) следует проверить:

- а) на выполнение систем ограничений (4.8) и (4.9) целевой функции Поставщика;
- б) на выполнение систем ограничений (4.11) и (4.12) целевой функции Потребителя.

Однако системы ограничений (4.9) и (4.12) могут быть определены далеко не всегда из-за отсутствия публикаций требований соответствующих групп. Кроме того, данные системы ограничений характеризуют создаваемую ИС на нижнем (в соответствии с положениями стандарта ISO 15288:2002) – техническом – уровне представления. Поэтому проверка выполнимости данных систем ограничений будет осуществляться только при соблюдении двух условий:

- а) выполнение систем ограничений (4.8) и (4.11), характеризующих создаваемую ИС на более высоком – проектном – уровне;
- б) в случае определения требований соответствующих групп.

Поэтому основное внимание следует уделить решению задачи проверки выполнения систем ограничений (4.8) и (4.11), которое следует проводить вне зависимости от особенностей конкретной создаваемой ИС.

Как показано в подразд. 1.4, основой для расчета стоимости, времени и качества выполнения IT-проекта является оценка объема работ по созданию ИС. Поэтому системы ограничений (4.8) и (4.11) примут, соответственно, вид:

$$\left\{ \begin{array}{l} \sum_{i=c+1}^e \alpha_i^{\text{Pr}} \text{pay}(PM(K_i^{fIS}, K_i^{fPr})) \geq \text{pay}^*(PM(K_i^{fIS}, K_i^{fPr})); \\ \sum_{i=c+1}^e \beta_i^{\text{Pr}} t(P(K_i^{fIS}, K_i^{fPr})) \leq t^*(PM(K_i^{fIS}, K_i^{fPr})); \\ \sum_{i=c+1}^e \gamma_i^{\text{Pr}} q(PM(K_i^{fIS}, K_i^{fPr})) \geq q^*(PM(K_i^{fIS}, K_i^{fPr})), \end{array} \right. \quad (4.43)$$

$$\left\{ \begin{array}{l} \sum_{i=c+1}^e \alpha_i^U \text{pay}(PM(K_i^{fIS}, K_i^{fU})) \leq \text{pay}^*(PM(K_i^{fIS}, K_i^{fU})); \\ \sum_{i=c+1}^e \beta_i^U t(PM(K_i^{fIS}, K_i^{fU})) \leq t^*(PM(K_i^{fIS}, K_i^{fU})); \\ \sum_{i=c+1}^e \gamma_i^U q(PM(K_i^{fIS}, K_i^{fU})) \geq q^*(PM(K_i^{fIS}, K_i^{fU})), \end{array} \right. \quad (4.44)$$

где $PM(K_i^{fIS}, K_i^{fPr})$ - функция расчета объема работ по созданию ИС с точки зрения Поставщика;

$PM(K_i^{fIS}, K_i^{fU})$ - функция расчета объема работ по созданию ИС с точки зрения Потребителя.

Физический смысл систем ограничений (4.43) и (4.44) заключается в проверке следующих условий эффективного выполнения IT-проекта создания ИС на основе ее ОРА:

а) стоимость IT-проекта создания данной ИС должна быть не ниже величины, минимально приемлемой для Поставщика, и не выше величины, максимально приемлемой для Потребителя;

б) время выполнения IT-проекта создания данной ИС должно быть не больше максимально допустимых сроков с точки зрения как Поставщика, так и Потребителя;

в) оценка качества выполнения IT-проекта создания данной ИС должна быть не меньше минимально приемлемых для Поставщика и Потребителя значений.

Тогда правые части систем (4.43) и (4.44) будут представлять собой условия, предлагаемые Поставщиком и Потребителем в ходе формирования и заключения

договора на создание ИС. Левые части систем (4.43) и (4.44) в этом случае будут представлять собой результаты расчета условий эффективного выполнения ИТ-проекта создания ИС на основе ее ОРА.

Для определения конкретного вида функций $P(K_i^{fIS}, K_i^{fPr})$ и $P(K_i^{fIS}, K_i^{fU})$ следует учесть, что основной объем работ по созданию ИТ-сервисов, реализующих ИТ-услуги ИС на основе ее ОРА, образуют работы по созданию ИО и ПО данных сервисов. Поэтому использование для оценивания объема подобных работ множеств представлений ФТ к ИС $\{K_i^{fIS}\}$, $\{K_i^{fPr}\}$ и $\{K_i^{fU}\}$ позволяет достичь следующих преимуществ:

а) любой из потоков данных ИТ-услуги можно одновременно описать и как набор сущностей, состоящих из отдельных атрибутов (по классическому методу функциональных точек), и как набор классов, формирующих такие объекты, как экраны или отчеты (по методу объектных точек, согласно которому каждый отдельный поток может быть представлен как экран или как отчет);

б) любой процесс преобразования данных ИТ-услуги можно таким образом одновременно описать и как набор внутренних логических файлов (ILF по классическому методу функциональных точек), и как набор 3GL или аналогичных им программных модулей (по методу объектных точек).

Таким образом, использование множеств представлений $\{K_i^{fIS}\}$, $\{K_i^{fPr}\}$ и $\{K_i^{fU}\}$ и формируемого на их основе ОРА ИС $\{K_{ji}^{fIS}\}_{i=(c+1), \dots, e}$ позволяет использовать это описание и множество $\{K_i^{fIS}\}$ как единую основу для формирования спецификаций на разработку ИО и ПО создаваемой ИС. Однако это преимущество является и дополнительным условием, ограничивающим использование данных представлений и ОРА как основы для оценки объема работ по созданию ИС. Рассмотрим это условие подробнее.

С точки зрения Потребителя, ИТ-услуга создаваемой ИС должна содержать информационную и программную реализации только тех концептов Про,

фреймы и интерфейсы которых присутствуют в представлениях ФТ к данной ИТ-услуге на уровне знаний. Однако такая точка зрения неудобна для Поставщика, поскольку приводит того к необходимости создания по требованиям различных Потребителей большого количества уникальных ИС, реализующих одни и те же ИТ-услуги в различных вариантах.

С точки зрения Поставщика, заявленные Потребителем концепты и описывающие их фреймы являются частными случаями семантической сети фреймов, заданной на иерархиях абстрактных концептов (таксономиях), детализирующих максимально абстрактные концепты, используемые в разных ИТ-услугах типовой ИС, до максимально конкретных концептов, используемых в конкретной ИТ-услуге данной ИС. Такая точка зрения во многом определена концепцией паттернов проектирования, применяемой при разработке ПО типовых ИС [150, 151]. Что касается разработки ИО типовой ИС, то данная точка зрения будет базироваться на предположении о возможности представления схемы данных создаваемой ИС как реализации многомерной модели данных [196], для которой реляционная модель данных будет частным случаем.

Сравнительный анализ точек зрения Поставщика и Потребителя позволяет выделить в качестве основной компромиссной условной единицы для оценивания объема работ по созданию ИС отдельную ветвь таксономий, образующих семантическую сеть фреймов ОРА создаваемой ИС. Каждая такая ветвь будет состоять из отдельных фреймов и интерфейсов, описывающих концепты различных уровней абстракции ПрО в онтологии ФТ к ИС и объединенных отношениями вложения (или обобщения (наследования)). В ходе создания ИТ-сервисов ИС такой ветви будут соответствовать [134, 197-199]:

а) взаимосвязанная совокупность таблиц нормализованных схем данных, схем данных типа «звезда» или «снежинка» ИО соответствующей ИТ-услуги ИС, обеспечивающая хранение значений данных концептов ПрО, формируемых в ходе эксплуатации ИТ-услуги;

б) совокупность классов ПО соответствующей функции ИС, реализующая бизнес-логику, экранные формы и отчеты, необходимые для ввода, обработки и вывода значе-

ний данных концептов Про, формируемых в ходе эксплуатации IT-услуги.

Тогда, по аналогии с функциональными и объектными точками как наименованиями условных единиц объема работ по созданию ИС, предлагается именовать каждую ветвь таксономии фреймов, присутствующей в описании рациональной архитектуры создаваемой ИС, термином «онтологическая точка» (ОТ). Показатель, характеризующий объем работ по созданию ИС как количество ОТ, позволит оценить как объем работ по созданию ИО ИС (как разновидность программирования на языке SQL или другом языке аналогичного назначения), так и объем работ по созданию ПО этой же ИС [184, 199].

В общем случае любую связь между фреймами или между фреймом и интерфейсом можно описать выражением вида [166, 184, 199]:

$$C_{fr_b}^{fr_a} \in C = \langle n_C, A_{fr_a}^C, A_{fr_b}^C, Pow_{fr_a}^C, Pow_{fr_b}^C, S_{fr_a}^C, S_{fr_b}^C \rangle, \quad (4.45)$$

где $C_{fr_b}^{fr_a}$ - элемент множества видов связей C ;

n_C - наименование вида связи $C_{fr_b}^{fr_a}$;

$A_{fr_a}^C$ - подмножество атрибутов фрейма fr_a , который участвует в образовании связи $C_{fr_b}^{fr_a}$, $A_{fr_a}^C \subseteq (ns_{a1}, \dots, ns_{ak})$;

$A_{fr_b}^C$ - подмножество атрибутов фрейма fr_b , который участвует в образовании связи $C_{fr_b}^{fr_a}$, $A_{fr_b}^C \subseteq (ns_{b1}, \dots, ns_{bk})$;

$Pow_{fr_a}^C$ - мощность связи $C_{fr_b}^{fr_a}$ для фрейма fr_a ;

$Pow_{fr_b}^C$ - мощность связи $C_{fr_b}^{fr_a}$ для фрейма fr_b ;

$S_{fr_a}^C$ - степень участия значений фрейма fr_a в образовании связи $C_{fr_b}^{fr_a}$;

$S_{fr_b}^C$ - степень участия значений фрейма fr_b в образовании связи $C_{fr_b}^{fr_a}$.

Мощность связи для фрейма fr_k определяется как количество элементов множества значений этого фрейма O_{fr_k} , которые могут участвовать в образовании связи.

Для выделения таксономии фреймов введем формализованное описание связи вида «генерализация». Исходя из выражения (4.45), данная связь может быть описана следующим образом [166, 184, 199]:

$$C^{gen} = \langle n_C, A_{fr_a}^C, A_{fr_b}^C, 1, 1, S_{fr_a}^C, S_{fr_b}^C \rangle \quad (4.46)$$

при выполнении для каждого дочернего фрейма fr_b условия [166, 184]

$$\left\{ \begin{array}{l} \exists fr_a = \{n, [(ns_1, vs_1, ps_1), \dots, (ns_k, vs_k, ps_k)], \{if_1, \dots, if_n\}, \{mt_1, \dots, mt_z\}\} \\ o_{fr_b}^i = \langle v_{fr_a}^{ij} \rangle \cup \langle v_{fr_b}^{ij} \rangle \\ F(o_{fr_a}^i) = F(o_{fr_b}^i) \\ F(o_{fr_b}^i) \neq F(o_{fr_a}^i) \end{array} \right. , \quad (4.47)$$

где $o_{fr_b}^i$ - множество значений фрейма fr_b ;

$v_{fr_a}^{ij}$ - значение j-го атрибута i-го значения фрейма fr_a ;

$v_{fr_b}^{ij}$ - значение j-го атрибута i-го значения фрейма fr_b ;

F - совокупность операций над значениями фреймов fr_a и fr_b , причем операции совокупности F не обязательно принадлежат данным классам.

Тогда введенный термин «ОТ», определяемый как отдельная ветвь таксономии фреймов, присутствующей в ОРА создаваемой ИС как сети фреймов, может быть формализованно описан выражением вида [184, 199]:

$$\begin{aligned}
OntPD =< FR_{OntPD} = (fr_i, \dots, fr_k, \dots, fr_j), C^{gen} , G_{OntPD} = \\
=< (< fr_i, fr_{i+1}, C^{gen} >, \dots, < fr_{k-1}, fr_k, C^{gen} >, < fr_k, fr_{k+1}, C^{gen} >), \dots, \\
< fr_{j-1}, fr_j, C^{gen} > >
\end{aligned} \quad (4.48)$$

при выполнении условия

$$\forall fr_k \in FR_{OntPD} \exists < fr_{k-1}, fr_k, C^{gen} > \cap < fr_k, fr_{k+1}, C^{gen} > \in G_{OntPD}, \quad (4.49) \\
i < k < j,$$

где $OntPD$ - формализованное описание ОТ;

FR_{OntPD} - подмножество фреймов, образующих ОТ, $FR_{OntPD} \subseteq FR$;

G_{OntPD} - множество отображений, которые задают связи генерализации между фреймами, входящими во множество FR_{OntPD} ;

i - идентификатор корневого фрейма ОТ;

j - идентификатор фрейма-листа ОТ.

Для формирования описаний ОТ на основе множества общесистемных представлений $\{ K_i^{fIS} \}$, образующих ОРА создаваемой ИС, предлагается метод, предусматривающий выполнение следующих этапов [184].

Этап 1. Выбрать нерассмотренное ранее представление $K_i^{fIS} \in \{ K_i^{fIS} \}$, образующее ОРА создаваемой ИС.

Этап 2. Зафиксировать нерассмотренный ранее фрейм fr^{ij} из множества фреймов представления K_i^{fIS} .

Этап 3. Выделить подмножество связей $G_{OntPD}^P \in C^{gen}$, для которых зафиксированный фрейм является дочерним.

Этап 4. Если $G_{OntPD}^P = \emptyset$, то перейти к Этапу 6.

Этап 5. Зафиксировать одну из связей подмножества G_{OntPD}^P и фрейм fr^{ik} , ко-

торый для зафиксированной связи является родительским. Затем перейти к Этапу 3.

Этап 6. Выделить подмножество связей $G_{OntPD}^{ch} \in C^{gen}$, для которых зафиксированный фрейм является родительским.

Этап 7. Если $G_{OntPD}^{ch} = \emptyset$, то перейти к Этапу 9.

Этап 8. Зафиксировать одну из связей подмножества G_{OntPD}^{ch} и фрейм fr^{ik} , который для зафиксированной связи является дочерним. Затем перейти к Этапу 6.

Этап 9. Сформировать ранее не существовавшее описание ОТ $OntPD_m, m = 1, 2, 3, \dots$ из зафиксированных в ходе выполнения Этапов 2-8 фреймов и связей и исключить зафиксированные связи из рассмотрения. Если для фрейма fr^{ij} выполняется условие $G_{OntPD}^p = \emptyset \cap G_{OntPD}^{ch} = \emptyset$, то перейти к Этапу 10. Если для фрейма fr^{ij} выполняется условие $G_{OntPD}^p = \emptyset$, то перейти к Этапу 8. В противном случае перейти к Этапу 5.

Этап 10. Исключить фрейм fr^{ij} из рассмотрения. Если множество фреймов представления K_i^{fIS} пусто, то исключить представление K_i^{fIS} из рассмотрения и перейти к Этапу 11. В противном случае перейти к Этапу 2.

Этап 11. Если выполняется условие $\{K_i^{fIS}\} = \emptyset$, то завершить применение метода. В противном случае перейти к Этапу 1.

Разработку метода оценки объема работ по реализации ОРА ИС с использованием ОТ предлагается осуществить путем модификации известного и апробированного на практике метода объектных точек [184, 200, 201]. Данный выбор обусловлен сравнительной простотой метода объектных точек и возможностью получения с его помощью достаточно точных оценок объема работ по созданию ПО ИС на ранних стадиях соответствующих ИТ-проектов. Для устранения невозможности оценки методом объектных точек объема работ по созданию ИО ИС предлагается рассматривать данный вид работ как программирование на языке PL/SQL (или аналогичному ему) БД со структурой, определяемой описанием рациональной архитектурой создаваемой ИС.

Тогда для определения объема работ по созданию ПО ИС необходимо установить вид функции оценки уровней сложности работ по созданию экранов, отчетов и модулей, необходимых для реализации каждой ИТ-услуги создаваемой ИС.

Использование процессного представления ФТ к ИС (см. подразд. 3.3) позволяет определить термин «экран» метода объектных точек как совокупность структур данных, необходимых для выполнения следующих операций [184]:

- а) ввод данных, инициирующих выполнение процесса;
- б) вывод результатов выполнения процесса на экран дисплея;
- в) извлечение из БД используемых в процессе данных;
- б) сохранение поступающих из процесса данных.

Термин «отчет» метода объектных точек в этом случае можно определить как совокупность структур данных, необходимых для выполнения операции вывода результатов выполнения процесса в виде электронного или бумажного документа [184].

Термин «3GL-модуль» в этом случае следует заменить термином «модуль», обозначающим совокупность структур данных, необходимых для обеспечения нормального функционирования всех экранов и отчетов, запланированных к реализации в рамках ИТ-услуги [184].

Поскольку подавляющее большинство архитектур современных ИС не предполагают хранение данных на компьютерах операторов и пользователей ИС, предлагаемая модификация метода объектных точек предполагает только оперирование суммарными показателями количества используемых таблиц (без их классификации на клиентские и серверные). В то же время сохранение значений и физического смысла базовых критериев позволяет применять проверенные на практике таблицы показателей метода объектных точек [134, 184, 201].

Формальное описание экранов, отчетов и модулей каждой конкретной ИТ-услуги будет определяться способом публикации ФТ к данной ИТ-услуге. В случае, если требование представляется в виде DFD или сводимых к ней вариантов публикаций, отчетами будут являться только выходные потоки данных, которые можно достаточно легко выделить в ходе формирования

публикации требования. Все остальные потоки данных будут являться экранами. Тогда уровень сложности работ по созданию экрана $Screen_{ij}$ или отчета $Report_{ik}$ можно будет оценить на основе количества ОТ, описывающих данный экран или отчет, и количества фреймов в этих ОТ [184, 199]:

$$Screen_{ij} = f(|\{OntPD_{ijm}\}|, |\{FR_{OntPD_{ijm}}\}|), \quad (4.50)$$

$$Report_{ik} = f(|\{OntPD_{ikm}\}|, |\{FR_{OntPD_{ikm}}\}|), \quad (4.51)$$

где $FR_{OntPD_{ijm}}$ - множество фреймов m -ой ОТ, характеризующей j -й экран i -й ИТ-услуги.

$FR_{OntPD_{ikm}}$ - множество фреймов m -ой ОТ, характеризующей k -й отчет i -й ИТ-услуги.

$j = 1, \dots, n$ - идентификатор экрана $Screen_{ij}$;

$k = 1, \dots, p$ - идентификатор отчета $Report_{ik}$.

Значения функции (4.50) приведены в табл. 4.2. Значения функции (4.51) приведены в табл. 4.3 [184, 199].

Для повышения точности оценки объема работ используем подход, принятый в классическом методе функциональных точек [83]. Согласно данному подходу, объем работ по реализации программного модуля обработки данных оценивается на основе совокупности структур данных, обрабатываемых в оцениваемом модуле. Поэтому для оценки уровня сложности работ по созданию модуля $Module_i$ и фрагмента БД ИС, с которым данный модуль взаимодействует в ходе эксплуатации ИС, можно использовать количество фреймов ОТ, характеризующих i -ю ИТ-услугу, и количество атрибутов данных фреймов [184, 199]:

$$Module_i = f(|\{FR_{OntPD_{im}}\}|, |\{ATTR_FR_{OntPD_{im}}\}|). \quad (4.52)$$

Значения функции (4.52) приведены в табл. 4.4 [199].

В случае, если требование представляется в виде диаграммы вариантов использования языка UML или сводимых к ней вариантов публикаций, каждая отдельная ОТ соответствующей ИТ-услуги будет требовать отдельных экранов для ввода и вывода данных. Функция (4.50) для таких экранов будет иметь вид [184, 199]

$$Screen_{ij} = f(1, \{FR_{OntPD_{ijm}}\}), \quad (4.53)$$

а ее значения будут определяться верхней строкой значений табл. 4.2. Кроме того, для обмена данными между модулем и БД необходим экран, характеризующийся количеством всех ОТ данной ИТ-услуги и оцениваемый функцией (4.50). Наличие отчетов в этом случае должно быть особо оговорено в публикации требования.

Тогда, по аналогии с методом объектных точек, объем работ по созданию ИС на основе ОРА можно оценить как объем работ по реализации новых ОТ следующим образом [184, 199]:

$$PM = \sum_{i=(c+1)}^e PM_i, \quad PM_i = \frac{NOntP_i}{PROD}, \quad (4.54)$$

где $PROD$ - коэффициент производительности исполнителей ИТ-проекта создания ИС (см. табл. 4.5);

$NOntP_i$ - оценка новых ОТ, определяемая по формуле

$$NOntP_i = \frac{NOntP_i(100 - \%reuse)}{100}; \quad (4.55)$$

$OntP_i$ - оценка ОТ ИТ-услуги, определяемая по формуле

Таблица 4.2 – Значения функции оценки уровня сложности работ по созданию экрана программного модуля ИТ-услуги

	$ \{FR_{OntPD_{ijm}}\} < 4$	$4 \leq \{FR_{OntPD_{ijm}}\} < 8$	$8 \leq \{FR_{OntPD_{ijm}}\} $
$ \{OntPD_{ijm}\} < 3$	$f = 1$	$f = 1$	$f = 2$
$3 \leq \{OntPD_{ijm}\} \leq 7$	$f = 1$	$f = 2$	$f = 3$
$8 \leq \{OntPD_{ijm}\} $	$f = 2$	$f = 3$	$f = 3$

Таблица 4.3 – Значения функции оценки уровня сложности работ по созданию отчета программного модуля ИТ-услуги

	$ \{FR_{OntPD_{ikm}}\} < 4$	$4 \leq \{FR_{OntPD_{ikm}}\} < 8$	$8 \leq \{FR_{OntPD_{ikm}}\} $
$ \{OntPD_{ikm}\} < 3$	$f = 2$	$f = 2$	$f = 5$
$3 \leq \{OntPD_{ikm}\} \leq 7$	$f = 2$	$f = 5$	$f = 8$
$8 \leq \{OntPD_{ikm}\} $	$f = 5$	$f = 8$	$f = 8$

Таблица 4.4 – Значения функции оценки уровня сложности работ по созданию программного модуля IT-услуги

	$ \{ ATTR_FR_{OntPD_{im}} \} \leq 19$	$20 \leq \{ ATTR_FR_{OntPD_{im}} \} \leq 50$	$50 < \{ ATTR_FR_{OntPD_{im}} \} $
$ \{ FR_{OntPD_{im}} \} = 1$	$f = 7$	$f = 7$	$f = 10$
$2 \leq \{ FR_{OntPD_{im}} \} \leq 5$	$f = 7$	$f = 10$	$f = 15$
$6 \leq \{ FR_{OntPD_{im}} \} $	$f = 10$	$f = 15$	$f = 15$

Таблица 4.5 – Значения коэффициента производительности исполнителей IT-проекта создания информационной системы

Опыт и способности разработчиков, зрелость и возможности CASE	Очень низкий	Низкий	Нормальный	Высокий	Очень высокий
PROD	4	7	13	25	50

$$OntP_i = \sum_{j=1}^n Screen_{ij} + \sum_{k=1}^p Report_{ik} + Module_i ; \quad (4.56)$$

$\%reuse$ - процент повторного использования фреймов ОТ, определяемый по формуле

$$\%reuse = 100 * (1 - (|FR_{OntPD_i}| - |FR_{K_i^{fPr}}|)); \quad (4.57)$$

$OntPD_i$ - множество описаний ОТ, формируемое на основе общесистемного представления K_i^{fIS} рассмотренным выше методом;

FR_{OntPD_i} - множество описаний фреймов из множества описаний ОТ $OntPD_i$;

$FR_{K_i^{fPr}}$ - множество описаний фреймов представления K_i^{fIS} .

Для расчета времени выполнения IT-проекта создания ИС на основе ее ОРА можно в этом случае использовать формулу расчета затрат времени модели СОСОМО II, которая имеет вид [184, 199, 200]:

$$TDEV = \left[3 \times \sqrt{PM^{(0,33+0,2 \times 0,01 \times B)}} \right] \times \frac{SCED\%}{100}, \quad (4.58)$$

где $TDEV$ - время выполнения IT-проекта создания ИС, месяцы;

B - оценка совокупности драйверов масштаба и экономичности IT-проекта создания ИС, которая определяется по формуле

$$B = PREC + FLEX + RESL + TEAM + PMAT ; \quad (4.59)$$

$PREC$ - показатель беспрецедентности создаваемой ИС;

$FLEX$ - показатель гибкости создаваемой ИС;

$RESL$ - показатель предпочтения архитектурных или рискованных решений;

TEAM - показатель сплоченности команды исполнителей IT-проекта создания ИС;

PMAT - показатель зрелости процессов разработки ИС;

SCED% - показатель сжатия расписания проекта по времени.

Значения драйверов масштаба и экономичности IT-проекта создания ИС приведены в табл. 4.6. Значения показателя *SCED%* приведены в табл. 4.7 [200].

Таблица 4.6 – Значения коэффициента производительности исполнителей IT-проекта создания информационной системы

Драйвер	Качественное значение драйвера					
	Очень низкий	Низкий	Нормальный	Высокий	Очень высокий	Слишком высокий
<i>PREC</i>	6,20	4,96	3,72	2,48	1,24	0
<i>FLEX</i>	5,07	4,05	3,04	2,03	1,01	0
<i>RESL</i>	7,07	5,65	4,24	2,83	1,41	0
<i>TEAM</i>	5,48	4,38	3,29	2,19	1,10	0
<i>PMAT</i>	7,80	6,24	4,68	3,12	1,56	0

Таблица 4.7 – Значения показателя *SCED%*

Показатель	Качественное значение показателя					
	Очень низкий	Низкий	Нормальный	Высокий	Очень высокий	Слишком высокий
<i>SCED%</i>	130	114	100	100	100	100

Тогда потребность *SS* в персонале для IT-проекта создания ИС на основе ее ОРА можно оценить по формуле [184, 200]

$$SS = \frac{PM}{TDEV}. \quad (4.60)$$

Результат расчета оценки потребности в персонале по выражению (4.60)

является основой для дальнейшего определения стоимости IT-проекта создания ИС на основе ее ОРА с учетом нормативных величин оплаты труда исполнителей работ IT-проекта создания данной ИС.

Предлагаемый подход к оцениванию объема работ на ранних стадиях IT-проекта создания ИС на основе ее ОРА позволяет повысить точность оценивания объема работ, затрат времени и потребности в персонале в ходе формирования и анализа требований к данной ИС, а также в ходе синтеза архитектуры создаваемой ИС.

4.6 Методы анализа сформулированных функциональных требований к информационной системе

Результаты, полученные в разд. 3 и разд. 4 данной работы, направлены на автоматизацию работ по формированию ФТ и синтезу на основе СТ ОРА создаваемой ИС. Однако ожидать приемлемые результаты выполнения данных работ невозможно без проведения анализа СФТ к создаваемой ИС.

В соответствии с положениями стандарта ISO 15288:2002, анализу рекомендуется подвергать полную совокупность выявленных требований [8]. При этом в анализ требований рекомендуется включать идентификацию противоречивых, пропущенных, неполных, неоднозначных, нелогичных или непроверяемых требований, а также расстановку приоритетов выполнения этих требований. В ходе анализа также следует разрешать проблемы, возникающие в связи с определением требований. Эти проблемы связаны с требованиями, которые не могут быть реализованы или которые нецелесообразно реализовывать.

Организация процессов определения требований правообладателей и анализа этих требований в соответствии со стандартом ISO 15288:2002 требует выполнения анализа ФТ после выполнения основных работ по формированию данных требований. Однако результаты разработки сервисного под-

хода, моделей формального описания требований на уровнях данных, информации и знаний, а также методов формирования представлений ФТ, синтеза и выбора ОРА создаваемой ИС позволяют организовать проведение отдельных видов анализа ФТ параллельно основным работам. Описание такой организации работ приведено в виде IDEF3-диаграммы на рис. 4.3 [202].

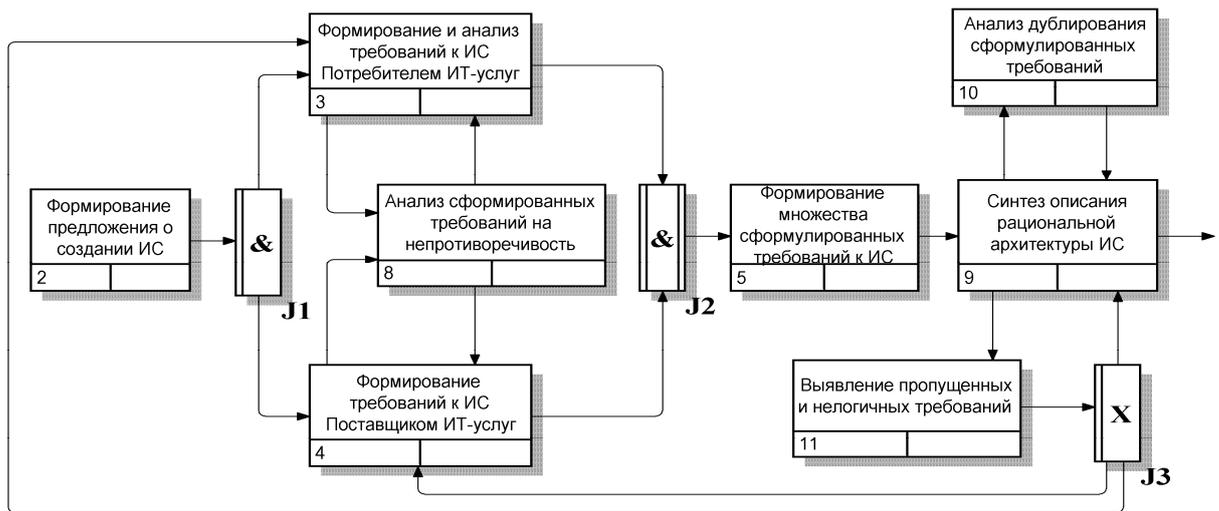


Рисунок 4.3 – IDEF3-диаграмма, описывающая работы по формированию и анализу функциональных требований к создаваемой информационной системе в соответствии с положениями сервисного подхода

Согласно предлагаемой диаграмме, организации работ анализ в рамках предлагаемых решений следует проводить:

- а) до начала синтеза ОРА создаваемой ИС на взаимную непротиворечивость отдельных ФТ;
- б) в ходе синтеза вариантов ОРА создаваемой ИС на неполноту СФТ путем выявления представлений требований, частично дублирующих друг друга;
- в) после выбора ОРА создаваемой ИС путем выявления пропущенных или нелогичных ФТ.

Основной предпосылкой проведения анализа ФТ на непротиворечивость является использование единого формального описания представлений требо-

ваний к ИС на уровне знаний. Тогда задача анализа ФТ на непротиворечивость может быть представлена как задача поиска представлений двух или более ФТ, в которых существует хотя бы одна из следующих ситуаций [202]:

а) если фреймы или интерфейсы с одинаковым или схожим названием содержат непересекающиеся множества элементов;

б) если между двумя или более парами фреймов или между фреймом и интерфейсом с одинаковыми или схожими описаниями существуют различные связи.

При этом данная задача не должна зависеть от того, какие именно представления требований на уровне знаний подвергаются анализу – с точки зрения Поставщика, с точки зрения Потребителя или общесистемные.

Проведение анализа требований на непротиворечивость в ходе формирования отдельного i -го ФТ позволяет выделить множество ранее сформированных представлений ФТ на уровне знаний $\{K_j^f\}$, то есть представлений требований, которые были сформулированы ранее i -го ФТ. Тогда метод анализа отдельных фреймов представления K_i^f на непротиворечивость предполагает выполнение следующих этапов.

Этап 1. Выбрать ранее не рассмотренный фрейм $fr^{ia} \in K_i^f$.

Этап 2. Выбрать ранее не рассмотренный фрейм $fr^{jb} \in K_j^f$, $K_j^f \in \{K_j^f\}$.

Этап 3. Если выполняется условие

$$(d_n^{ia} = d_n^{jb}) \wedge ((\{ \langle d_{el_fr}^{ia}, d_{el_fr_t}^{ia} \rangle \} / \{ \langle d_{el_fr}^{jb}, d_{el_fr_t}^{jb} \rangle \}) = \emptyset), \quad (4.61)$$

то признать наличие противоречия между i -ым и j -ым ФТ в описаниях фреймов fr^{ia} и fr^{jb} , после чего перейти к Этапу 5.

Этап 4. Если выполняется условие

$$\begin{aligned} & ((d_n^{ia} \subseteq d_n^{jb}) \vee (d_n^{ia} \supseteq d_n^{jb})) \wedge \\ & ((\{ \langle d_{el_fr}^{ia}, d_{el_fr_t}^{ia} \rangle \} \cap \{ \langle d_{el_fr}^{jb}, d_{el_fr_t}^{jb} \rangle \}) = \emptyset) \end{aligned} \quad (4.62)$$

то признать наличие противоречия между i -ым и j -ым ФТ в описаниях фреймов fr^{ia} и fr^{jb} .

Этап 5. Исключить фрейм fr^{jb} из дальнейшего рассмотрения. Если не рассмотрены все фреймы $fr^{jb} \in K_j^f$, то перейти к Этапу 2.

Этап 6. Исключить представление K_j^f из дальнейшего рассмотрения. Если не рассмотрены все представления множества $\{K_j^f\}$, то выбрать ранее не рассмотренное представление $K_j^f \in \{K_j^f\}$ и перейти к Этапу 2.

Этап 7. Исключить фрейм fr^{ia} из дальнейшего рассмотрения. Если не рассмотрены все фреймы $fr^{ia} \in K_i^f$, то перейти к Этапу 1, в противном случае – завершить применение метода.

Метод анализа отдельных интерфейсов представления K_i^f на непротиворечивость будет аналогичен рассмотренному выше методу анализа отдельных фреймов.

Для анализа второй возможной ситуации предлагается метод анализа отдельных связей представления K_i^f на непротиворечивость, предполагающий выполнение следующих этапов.

Этап 1. Выбрать ранее не рассмотренное представление $K_j^f \in \{K_j^f\}$.

Этап 2. Выбрать ранее не рассмотренную связь $rel^{jm} \in K_j^f$, а также фреймы fr^{jc} и fr^{jd} , образующие связь rel^{jm} .

Этап 3. Найти в представлении K_i^f фреймы fr^{ia} и fr^{ib} , для которых выполняется условие $(fr^{jc} = fr^{ia}) \wedge (fr^{jd} = fr^{ib})$. Если таких фреймов не

существует, то перейти к Этапу 7.

Этап 4. Проверить существование связи $rel^{ik} \in K_i^f$, определенной на фреймах fr^{ia} и fr^{ib} . Если такая связь не существует, то признать наличие противоречия между i -ым и j -ым ФТ в описании связи $rel^{jm} \in K_j^f$ и отсутствием аналогичного описания $rel^{ik} \in K_i^f$.

Этап 5. Проверить выполнение условия $rel^{jm} = rel^{ik}$. Если условие не выполняется, то признать наличие противоречия между i -ым и j -ым ФТ в описаниях связей $rel^{jm} \in K_j^f$ и $rel^{ik} \in K_i^f$.

Этап 6. Исключить связь rel^{jm} из дальнейшего рассмотрения. Если не рассмотрены все связи $rel^{jm} \in K_j^f$, то перейти к Этапу 2.

Этап 7. Исключить представление K_j^f из дальнейшего рассмотрения. Если не рассмотрены все представления множества $\{K_j^f\}$, то выбрать ранее не рассмотренное представление $K_j^f \in \{K_j^f\}$ и перейти к Этапу 2. В противном случае – завершить применение метода.

Результатами применения данных методов будут списки противоречий между отдельными ФТ, которые выявляются в ходе формирования отдельных ФТ к создаваемой ИС. Выявление данных противоречий дает возможность корректировать публикации ФТ в ходе их формирования, что позволит сократить время проведения анализа СТ.

Применение разработанного в подразд. 4.3 метода синтеза вариантов АД создаваемой ИС позволяет отказаться от проведения отдельного анализа СФТ на неполноту. Выявление частично дублирующих друг друга ФТ осуществляется непосредственно в процессе выполнения Этапа 3 метода синтеза вариантов АД создаваемой ИС [202].

Для выявления пропущенных и нелогичных ФТ введем следующие определе-

ния. Нелогичным будем называть ФТ к ИТ-услуге, которая не связана потоками данных с любой из других ИТ-услуг. Пропущенными будем называть одно или несколько ФТ, которые могли бы превратить нелогичное требование в логичное [202].

Для выявления нелогичных и пропущенных требований предлагается использовать введенный в подразд. 4.5 термин «ОТ». Данный термин характеризует не только все i -е ФТ к ИС, но и каждый поток данных ИТ-услуги, к которой выдвигается это требование.

Тогда i -е ФТ можно считать нелогичным, если все его ОТ присутствуют только в представлении K_i^{fIS} . Для характеристики степени нелогичности i -го ФТ предлагается использовать показатель Irr_i , значение которого определяется по формуле [202]

$$Irr_i = \frac{|\{OntPD_{im}^{irr}\}|}{|\{OntPD_{im}\}|} \times 100\%, \quad (4.63)$$

где $OntPD_{im}^{irr}$ - описание m -ой ОТ, присутствующей только в представлении K_i^{fIS} ;

$OntPD_{im}$ - описание m -ой ОТ, присутствующей в представлении K_i^{fIS} .

Тогда для выявления нелогичных ФТ предлагается использовать метод, предполагающий выполнение следующих этапов.

Этап 1. Выбрать нерассмотренное ранее представление K_i^{fIS} из множества представлений $\{K_i^{fIS}\}$, образующих ОРА создаваемой ИС.

Этап 2. Выделить множество описаний ОТ $\{OntPD_{im}\}$ представления K_i^{fIS} . Принять множество описаний нелогичных ОТ $\{OntPD_{im}^{irr}\} = \{OntPD_{im}\}$.

Этап 3. Выбрать нерассмотренное ранее представление K_j^{fIS} , $j \neq i$ из множества представлений, образующих ОРА создаваемой ИС. Принять $\{K_j^{fIS}\} = \{K_i^{fIS}\} / K_i^{fIS}$.

Этап 4. Сформировать множество описаний ОТ $\{OntPD_{jm}\}$ представления K_j^{fIS} .

Этап 5. Принять множество описаний нелогичных ОТ $\{OntPD_{im}^{irr}\} = \{OntPD_{im}^{irr}\} / \{OntPD_{jm}\}$. Если $\{OntPD_{im}^{irr}\} = \emptyset$, то перейти к Этапу 8.

Этап 6. Исключить представление K_j^{fIS} из дальнейшего рассмотрения. Если $\{K_j^{fIS}\} \neq \emptyset$, то перейти к Этапу 3.

Этап 7. Рассчитать значение показателя Irr_i по формуле (4.63) и признать нелогичность i -го ФТ с выделением нелогичных ОТ.

Этап 8. Исключить представление K_i^{fIS} из дальнейшего рассмотрения. Если $\{K_i^{fIS}\} \neq \emptyset$, то перейти к Этапу 1. В противном случае – завершить применение метода.

Предлагаемый метод позволяет значительно упростить процедуру выявления нелогичных ФТ за счет автоматизации операций по выявлению нелогичных объектов ПрО исследуемого требования к ИТ-услуге, данные о которых не предполагается использовать ни в одной другой ИТ-услуге.

Таким образом, предлагаемые методы анализа СФТ позволяют автоматизировать процессы выявления противоречивых, неполных и нелогичных ФТ за счет использования сформулированной в подразд. 3.3 концепции представления требований к ИС и разработанных на ее основе моделей представлений ФТ к ИС на уровне знаний.

4.7 Выводы к четвертому разделу

1. На основе результатов, полученных в разд. 2 и разд. 3, была сформулирована задача синтеза ОРА ИС как задача выбора совокупности представлений ФТ к ИС на уровне знаний. Целью решения данной задачи является удовлетворение в максимальной степени целей Поставщика (выражения (4.7)-(4.9)) и Потребителя (выражения (4.10)-(4.12)) в процессе проектирования архитектуры системы.

2. Разработаны методы формирования представлений ФТ к ИС на уровнях информации и знаний, позволяющие формализовать процессы формирования описаний и анализа требований к ИС, что обеспечивает сокращение затрат времени на выполнение данных процессов и возможность автоматизированного проектирования архитектуры ИС.

3. Разработан метод синтеза вариантов АД создаваемой ИС, в основу которого положен получивший дальнейшее развитие алгоритм решения задачи кластеризации CLOPE. В отличие от существующих версий алгоритма, предлагаемый метод выделяет для дальнейшего исследования варианты описаний, незначительно хуже оптимального, что позволит в максимально возможной степени выполнить требования Поставщика и Потребителя ИТ-услуг в процессе формирования компромиссного ОРА создаваемой ИС за счет увеличения количества исходных вариантов АД.

4. Разработана теоретико-игровая модель синтеза ОРА создаваемой ИС, которая, в отличие от существующих, описывает процесс синтеза ОРА как выбор варианта описания, в наибольшей степени соответствующего представлениям создаваемой системы Поставщика и Потребителя ИТ-услуг, что позволяет автоматизировать процесс проектирования архитектуры системы путем поиска точек равновесия по Нэшу в чистых и смешанных стратегиях.

5. Разработаны модель ОТ, которая описывает создаваемую ИС как множество отдельных ветвей таксономий фреймов, и метод формирования описаний ОТ на основе данной модели. Данная модель и метод позволяют авто-

матизировать процесс количественного оценивания объема работ по созданию ИС на основе разработанных моделей ФТ на уровне знаний.

6. Разработаны методы анализа СФТ к создаваемой ИС путем выявления противоречивых и нелогичных требований в ходе решения задачи синтеза ОРА ИС, что позволяет унифицировать и автоматизировать выполнение работ по анализу ФИ в рамках макропроектирования.

Основные результаты данного раздела изложены в работах [134, 180-185, 188-192, 197-199, 201, 202].

РАЗДЕЛ 5

ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ ФОРМИРОВАНИЯ И АНАЛИЗА
ТРЕБОВАНИЙ К ИНФОРМАЦИОННОЙ СИСТЕМЕ5.1 Описание архитектуры и основных IT-услуг информационной
технологии

Приведенное в конце подразд. 1.1 определение понятия «ИТ» является результатом уточнения определения, изложенного в стандарте ГОСТ 34.003-90 [1]. Однако данный стандарт не учитывает особенности процессного представления ЖЦ ИС и сервисного подхода к созданию ИС. Поэтому, прежде чем перейти к описанию особенностей практической реализации ИТ формирования и анализа требований к ИС, возможное подмножество которых описано моделью (2.45), следует вначале уточнить определение понятия «ИТ». Данное уточнение связано с общностью точек зрения на АФ и ИТ как на совокупность методов, приемов и способов. Разница этих точек зрения заключается в следующем аспекте [134]:

а) АФ, как показано в подразд. 1.3 и 3.1, представляет собой совокупность методов, приемов и способов, используемых для выполнения создания ИС в целом на основе конкретного подхода;

б) ИТ представляет собой совокупность унифицированных методов, приемов и способов применения КСА или его отдельных элементов, которая применяется для формальной обработки данных в ходе выполнения работ одного или нескольких процессов создания ИС в соответствии с АФ, принятым Поставщиком и Потребителем.

Иными словами, ИТ следует рассматривать как формальную часть АФ, описываемую моделью (3.1). При этом каждый АФ может быть реализован множеством различных ИТ, не противоречащих базовым положениям этого АФ. Следовательно, любую ИТ, используемую в ходе создания ИС, следует рассматривать как совокупность унифицированных методов, приемов и способов применения КСА или его отдельных элементов, используемую для вы-

полнения одной или нескольких работ в рамках процессов создания ИС.

Термин «ИТ формирования и анализа требований к ИС» был введен в подразд. 2.6 для описания любой возможной ИТ, в основе которой находится модель (2.45). Исходя из сказанного выше, данный термин следует трактовать как совокупность унифицированных и основанных на знаниях методов, приемов и способов применения КСА или его отдельных элементов, используемую для выполнения одной или нескольких работ в рамках процессов создания ИС, непосредственно работающих с требованиями к данной системе.

Применительно к ИТ формирования и анализа требований к ИС, цель ИТ, назначение и состав ИТ-услуг будут определяться следующими процессами создания ИС:

- а) процесс определения требований правообладателей;
- б) процесс анализа требований;
- в) процесс проектирования архитектуры.

Тогда для реализации ИТ формирования и анализа требований к ИС на основе разработанных в разд. 3 и 4 моделей и методов следует в общем случае разработать *совокупность унифицированных в рамках АФ макропроектирования ИС и основанных на знаниях методов, приемов и способов применения КСА или его отдельных элементов, используемую для выполнения одной или нескольких работ в рамках процессов создания ИС, непосредственно работающих с требованиями к данной системе.*

Концепция ИТ формирования и анализа требований к ИС определяется результатами разработки элементов АФ макропроектирования ИС, рассмотренными в разд. 3 и 4. В общем случае эти результаты позволяют определить основные виды взаимодействия между представлениями ФТ к ИС на уровне информации, данных и знаний. Схема таких взаимодействий будет иметь вид, показанный на рис. 5.1 [203].

Основываясь на данной концепции, представим общую технологию формирования и анализа требований к ИС как совокупность работ, выполняемых в рамках следующих стадий.

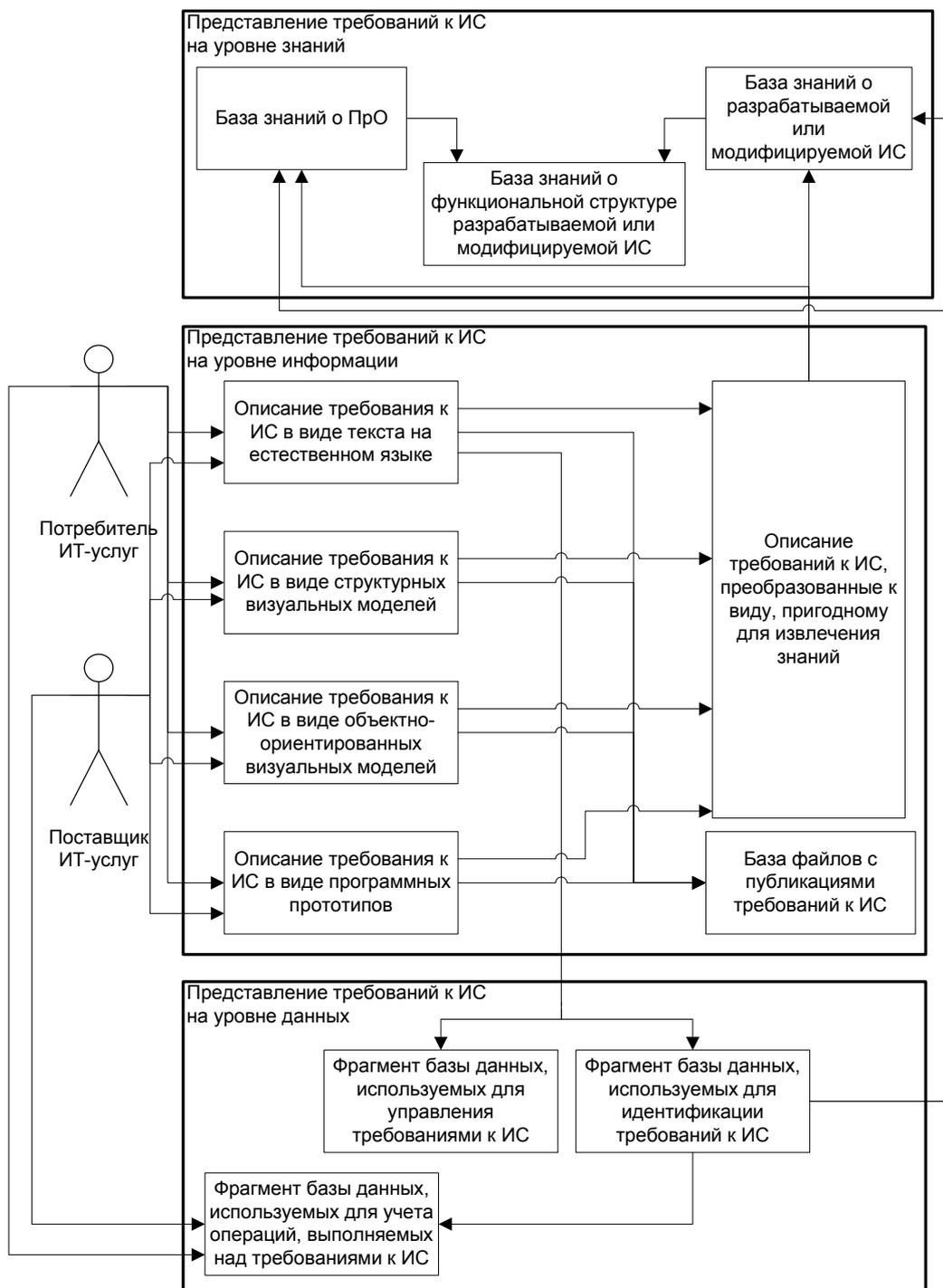


Рисунок 5.1 – Схема взаимодействия представлений требований к ИС на уровнях информации, данных и знаний в рамках информационной технологии формирования и анализа требований к информационной системе

Стадия 1 «Выявление и публикация ФТ к ИС» состоит из следующих работ:

а) выявление ФТ к ИС, неизвестного Потребителю;

- б) выявление ФТ к ИС, неизвестного Поставщику;
- в) публикация СФТ к ИС;
- г) принятие СФТ к ИС;
- д) формирование представления ФТ к ИС на уровне информации.

Стадия 2 «Формирование и анализ представлений ФТ к ИС на уровне знаний» состоит из следующих работ:

- а) формирование представления ФТ к ИС на уровне знаний с точки зрения УАП;
- б) формирование представления ФТ к ИС на уровне знаний с точки зрения Потребителя;
- в) формирование представления ФТ к ИС на уровне знаний с точки зрения Поставщика;
- г) формирование общесистемного представления ФТ к ИС на уровне знаний;
- д) выявление противоречивых ФТ к ИС.

Стадия 3 «Синтез ОРА ИС на основе СТ» состоит из следующих работ:

- а) синтез вариантов АД ИС;
- б) выбор ОРА ИС.

Стадия 4 «Оценивание основных проектных ограничений на создание ИС» состоит из следующих работ:

- а) формирование описаний ОТ ИТ-услуг создаваемой ИС;
- б) оценивание объема работ по созданию ИС;
- в) определение основных проектных ограничений ИТ-проекта создания ИС.

Стадия 5 «Формирование спецификаций на разработку ИО и ПО ИС» состоит из следующих работ:

- а) формирование схем данных ИО ИТ-сервисов создаваемой ИС;
- б) формирование ДК ПО ИТ-сервисов создаваемой ИС.

IDEF3-диаграмма, описывающая основные взаимосвязи стадий обобщенной технологии формирования и анализа требований к ИС, приведена на рис. 5.2.

Таким образом, разрабатываемая ИТ формирования и анализа требований к ИС должна обеспечить реализацию совокупности методов, приемов и

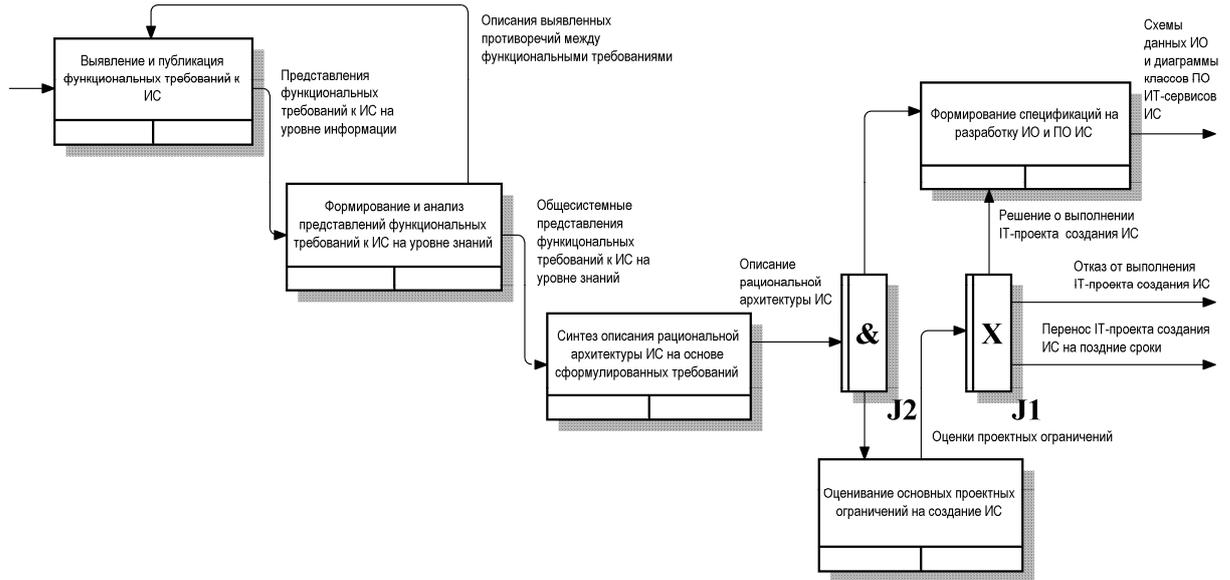


Рисунок 5.2 – IDEF3-диаграмма, описывающая основные взаимосвязи стадий обобщенной технологии формирования и анализа требований к информационной системе

способов применения КСА или его отдельных элементов, которая используется для выполнения одной или нескольких работ в рамках обобщенной технологии формирования и анализа требований к ИС. Однако, как следует из (3.7), (3.30) и (3.57), совокупность структурных и поведенческих ППТ к ИС, определяющих рассмотренную выше концепцию обобщенной технологии, может изменяться в зависимости от потребностей Поставщика и Потребителя. Поэтому возникает необходимость рассматривать ИТ формирования и анализа требований к ИС как самоописываемую ИТ. Это означает, что разработанные в разд. 3 и 4 МТ, их представлений на уровнях данных, информации и знаний, а также ППТ должны рассматриваться как описания ПрО, знания о которой следует использовать в ходе разработки данной ИТ.

Для реализации самоописания ИТ формирования и анализа требований к ИС примем предположение о возможности рассмотрения каждого ППТ к ИС как представления функционального требования к данной ИТ на уровне знаний. Это предположение не только позволит упростить разработку ИТ, но и

обеспечит уменьшение затрат на сопровождение и модернизацию данной ИТ за счет представления ее как очередной ИС, знания о которой хранятся в библиотеке ранее реализованных требований. Кроме того, подобное представление ППТ делает возможным ре-использование этих паттернов как первоначального наполнения данной библиотеки.

С учетом дополнений, предложенных в ходе разработки ИТ формирования и анализа требований к ИС и основанных на опыте создания ИС и ИТ различного назначения, было сформировано ОРА данной ИТ. Для ее визуализации могут применяться как ДК языка UML, так и ER-диаграммы. Каждая из этих диаграмм позволяет настраивать уровень детализации классов и таблиц (например, только названия и связи или названия, связи и атрибуты и т.д.). Критерием выбора конкретного типа диаграмм и уровня их детализации является определение оптимального уровня сложности диаграмм, который является компромиссным решением двух противоречивых задач:

- а) формирование максимально ёмкого и подробного описания объектов ПрО и их атрибутов;
- б) минимизация уровня сложности визуальных диаграмм с целью обеспечения возможности их восприятия человеком и удобства их доработки.

UML-диаграмма, отражающая «низкорослый лес» ОРА данной ИТ и содержащая только названия элементов данного описания, приведена на рис. 5.3. Данный пример наглядно иллюстрирует, что при расширении библиотеки ранее реализованных требований к ИС или в ходе создания ИС с достаточно большим количеством ИТ-услуг ДК, описывающие соответствующие архитектуры, будут очень громоздкими и вытянутыми в горизонтальном направлении. Кроме того, для подобных диаграмм будет характерным наличие большого количества очень длинных горизонтальных связей, устанавливающих ассоциации между элементами разных ОТ, что затруднит восприятие подобных диаграмм. В то же время такой вид диаграмм наиболее естественным и полным образом отражает структуру ОТ АД создаваемой ИС.

На рис. 5.4 приведена ER-диаграмма, описывающая архитектуру данной ИТ в

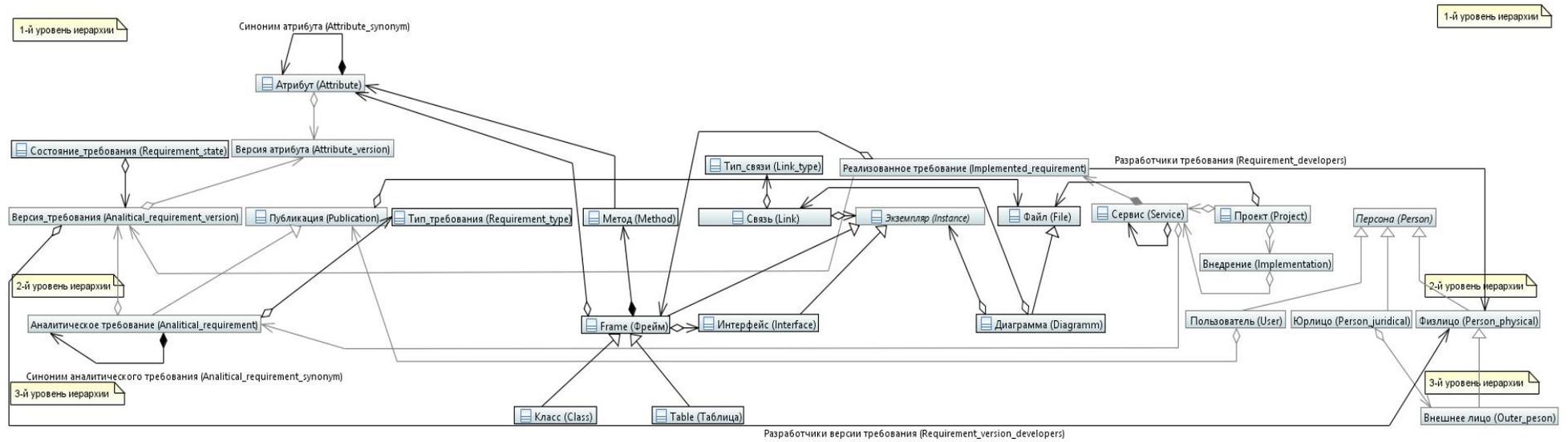


Рисунок 5.3 – Описание архитектуры информационной технологии формирования и анализа требований к информационной системе в виде диаграммы классов

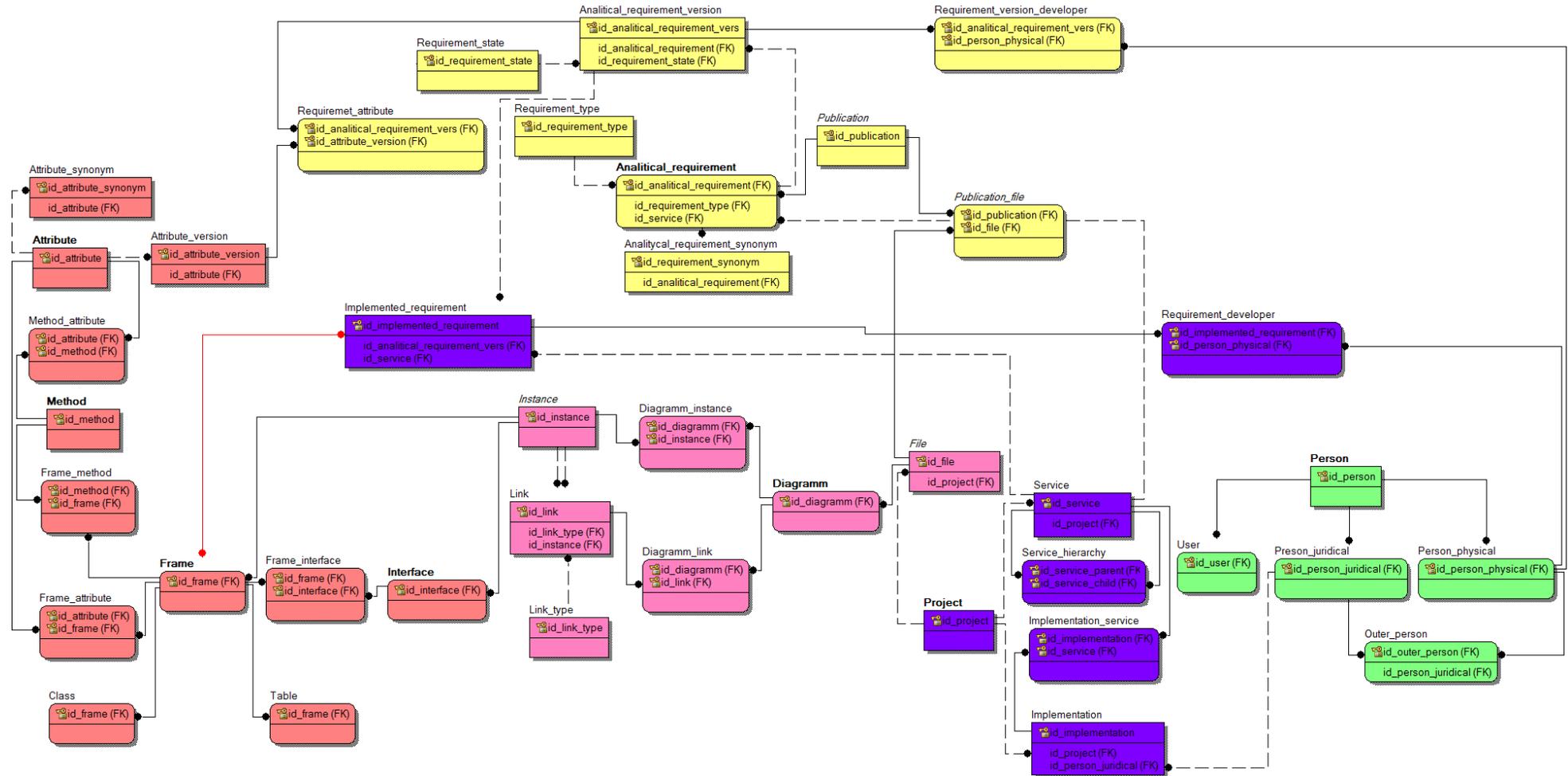


Рисунок 5.4 – Описание архитектуры информационной технологии формирования и анализа требований к информационной системе в виде ER-диаграммы

виде набора схем данных типа «звезда» и «снежинка». Таблицы, участвующие в формировании общей витрины данных, выделены на рис. 5.4 одним и тем же цветом.

Необходимо отметить, что приведенные на рис. 5.3 и рис. 5.4. AD ИТ формирования и анализа требований к ИС не отражают ряд фреймов, описывающих стандартные (то есть существующие в ИТ аналогичного назначения) возможности данной ИТ.

Сравнительный анализ диаграмм, приведенных на рис. 5.3 и рис. 5.4, показывает, что информативность ER-диаграммы в данном случае аналогична информативности ДК. В то же время сложность восприятия ER-диаграммы гораздо ниже за счёт того, что отдельные схемы данных группируют в себе множества таблиц, что позволяет визуально показать, как связи между отдельными таблицами отражают связи между целыми витринами данных.

Схемы типа «звезда» или «снежинка», соответствующие отдельным ОТ, могут располагаться друг относительно друга в произвольном порядке, что снижает громоздкость ER-диаграмм за счёт потери части наглядности ДК UML. Снижение сложности восприятия ER-диаграммы позволяет аналитикам оперировать с её расширенной версией, в которую, в зависимости от количества отображаемых схем данных, могут включаться также ключевые атрибуты.

Разработанное AD определяет следующие основные ИТ-услуги ИТ формирования и анализа требований к ИС [204-206]:

- а) «Выявление требований к ИС»;
- б) «Формирование иерархий терминов Про создаваемой ИС»;
- в) «Сравнение сформированных иерархий терминов Про ИС с разработанными ранее иерархиями терминов Про»;
- г) «Выделение и уточнение паттернов требований к ИС»;
- д) «Синтез архитектуры ИС на основе выделенных паттернов требований к ИС»;
- е) «Осуществление взаимно-однозначного отображения паттернов ИС в сущности БД и классы ПО разрабатываемой ИС».

Схема потоков данных, описывающая основные взаимосвязи указанных ИТ-услуг ИТ формирования и анализа требований к ИС, приведена на рис.

5.5. Следует отметить, что на данной схеме не показаны ИТ-услуги, автоматизирующие управление ФТ и обработку представлений ФТ на уровне данных, поскольку они не отличаются от функций существующих ИТ УТ.

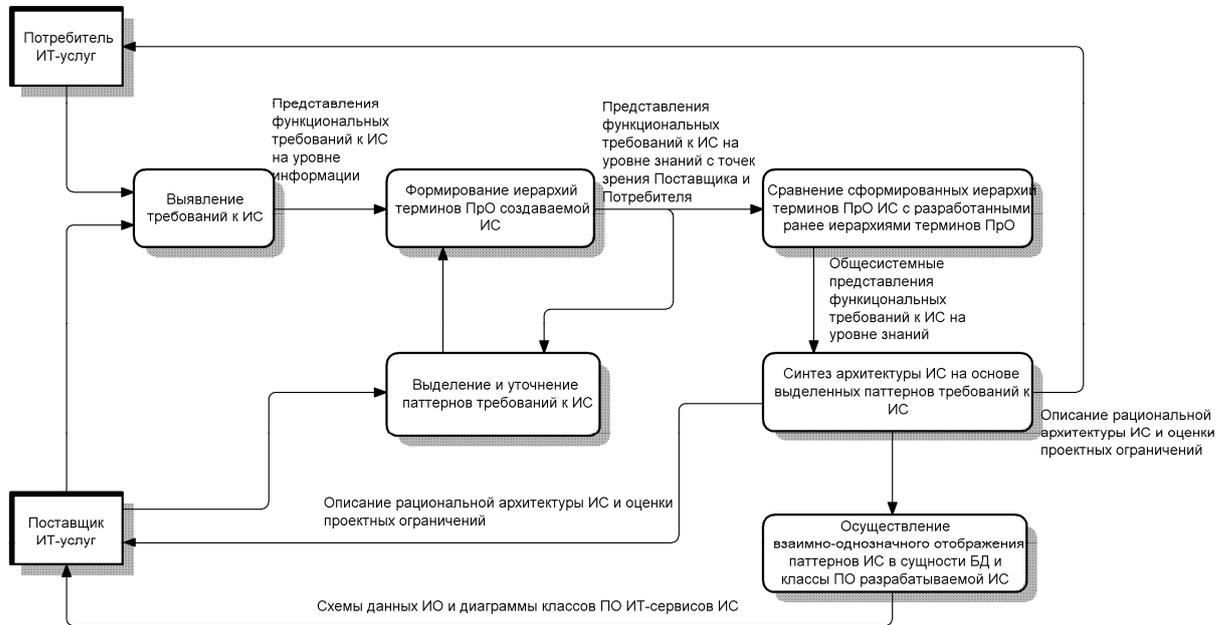


Рисунок 5.5 – Схема потоков данных, описывающая основные взаимосвязи ИТ-услуг информационной технологии формирования и анализа требований к информационной системе

ИТ-услуга «Выявление требований к ИС» представляет собой совокупность методов, приемов и способов сбора требований к создаваемой ИС вне зависимости от способа представления этих требований, а также их последующей обработки и формализованного описания. Основным результатом данной ИТ-услуги следует считать сформированные представления требований к создаваемой ИС на уровне информации и на уровне данных, необходимые для последующего отображения этих представлений в представления требований на уровне знаний.

ИТ-услуга «Формирование иерархий терминов ПрО создаваемой ИС» представляет собой совокупность методов, приемов и способов создания представлений ФТ к ИС на уровне знаний с точек зрения отдельного УАП, Потребителя и Поставщика. Основным результатом данной ИТ-услуги следу-

ет считать множество сформированных представлений требований к создаваемой ИС на уровне знаний, выделение из этих представлений универсума требований к проектируемой ИС и универсума библиотеки ранее реализованных требований, описанных с применением фреймового представления знаний. Выполнение данной ИТ-услуги завершает подготовку исходных данных для сопоставления различных ПрО.

ИТ-услуга «Сравнение сформированных иерархий терминов ПрО ИС с разработанными ранее иерархиями терминов ПрО» представляет собой совокупность методов, приемов и способов анализа и сопоставления сформированных иерархий терминов ПрО проектируемой ИС с разработанными ранее иерархиями терминов ПрО. Основными результатами данной ИТ-услуги следует считать общесистемные представления ФТ к ИС на уровне знаний, формируемые в результате принятия Поставщиком и Потребителем следующих решений:

а) решение о наследовании терминов (добавление терминов к существующим деревьям онтологий в роли ветвей или листьев) и, соответственно, о ре-использовании существующих элементов ИС;

б) решение о формировании новых узлов иерархий онтологий и, соответственно, о разработке новых элементов.

ИТ-услуга «Выделение и уточнение паттернов требований к ИС» представляет собой совокупность методов, приемов и способов выполнения предварительных работ, необходимых для синтеза варианта конфигурации создаваемой ИС. Полностью формализовать процесс принятия решений о наследовании терминов невозможно. Оценка подобия структур данных позволяет только сформировать положительную или отрицательную рекомендацию, а собственно принятие решения о применении термина, как и при анализе синонимов терминов, остается за человеком-экспертом. Таким образом, основным результатом данной ИТ-услуги следует считать доработку существующих или создание новых аналитических описаний паттернов требований в виде онтологий, представленных иерархиями фреймов.

IT-услуга «Синтез архитектуры ИС на основе выделенных паттернов требований к ИС» представляет собой совокупность методов, приемов и способов синтеза ОРА ИС с применением существующих, доработанных или новых описаний паттернов требований к ИС, а также реализаций этих паттернов в виде отдельных IT-услуг и соответствующих IT-сервисов. Основным результатом данной IT-услуги следует считать синтезированные варианты AD создаваемой ИС, которые позволяют описывать систему одновременно и как единый целостный объект исследования, и как совокупность IT-услуг, предлагаемых в рамках создаваемой ИС.

IT-услуга «Осуществление взаимно-однозначного отображения паттернов ИС в сущности БД и классы ПО разрабатываемой ИС» представляет собой совокупность методов, приемов и способов выполнения работ по физическому проектированию ИС в соответствии с сформированным ранее планом проекта создания данной ИС. В рамках данной IT-услуги осуществляется взаимно-однозначное отображение паттернов требований к ИС в сущности БД и классы ПО создаваемой ИС для уточняемых или новых паттернов требований, а также организация их взаимодействия в составе конфигурации создаваемой ИС.

Основными результатами данной IT-услуги следует считать описания схемы БД и классов прикладного ПО ИС, сформированные на основе выделенных ранее паттернов требований к ИС, в том числе – по результатам реиспользования элементов библиотеки ранее реализованных требований к ИС.

На основе результатов рассмотрения IT-услуг и функций ИТ формирования и анализа требований к ИС можно сделать вывод, что отрицательными сторонами применения данной ИТ являются [134, 207]:

- а) повышение сложности выполнения процесса анализа требований к ИС;
- б) повышение необходимого уровня качества ведения проектной документации на создаваемую ИС (в частности, возникает необходимость формирования и поддержания в актуальном состоянии описаний библиотеки ранее реализованных требований).

Положительными сторонами разрабатываемой ИТ следует считать [134, 207]:

- а) формализацию определения возможности ре-использования элементов библиотеки ранее реализованных требований в ходе создания новой ИС;
- б) автоматизацию работ по расширению библиотеки ранее реализованных требований;
- в) снижение сложности решения задач обновления версий ИС и сопровождения вариантов тиражируемых ИС, внедренных в БП различных предприятий.

5.2 Хранение и обработка представлений требований на уровне информации

Представление ФТ к ИС на уровне информации является основным представлением требований, которым непосредственно оперирует как Поставщик, так и Потребитель. Как следует из рис. 5.1, данное представление может включать как неформализованные текстовые описания требований, так и их формализованные описания, созданные с применением различных языков визуального моделирования (таких как UML, нотации IDEF, ARIS и т.п.) или в виде программного прототипа. Поэтому возникает задача организации хранения и обработки слабоструктурированной информации с целью извлечения данных, необходимых для формирования представлений ФТ к ИС на уровне знаний.

Анализ и извлечение данных и знаний из текстовых описаний требований является слабоформализуемой задачей, для которой в настоящее время уже существует целый ряд решений [208], и в данной работе рассматриваться не будет. Основой для формирования представлений требований на уровне информации следует считать предположение, согласно которому любое текстовое описание требований может быть достоверно и полно представлено в виде набора основных ВМ, понятных одновременно и Поставщику, и Потребителю (все виды UML-диаграмм, а также основные виды структурных диаграмм: IDEF0, DFD,

IDEFX3, ERD и т.п.). Аналогичным способом могут быть описаны требования, уже реализованные в виде элементов ИС и ее видов обеспечений.

Данное предположение позволяет рассматривать задачу реализации отображения представлений ФТ к ИС на уровне информации в представлении ФТ к ИС на уровне знаний как задачу обработки диаграмм ВМ с целью извлечения из них знаний и формирования на их основе МТ (4.14).

Необходимость обработки различных видов ВМ (структурных, объектных, отражающих статические понятия и динамические процессы), множество различных видов диаграмм, выполненных с использованием различных алфавитов построения ВМ, требует приведения всего многообразия используемых диаграмм к единому алфавиту описания их элементов. С учетом разработанного в подразд. 2.5 формализованного описания ОКФ, устанавливающего связь между представлениями ФТ к ИС на уровнях информации и знаний, данную задачу формализованно можно описать в виде коммутативных диаграмм [209]

$$\begin{array}{ccc}
 MTez(m_a^A) \rightarrow MTez(m_a^B) & MTez(r_a^A) \rightarrow MTez(r_a^B) & \\
 \uparrow & \downarrow & , \\
 Tez(m_v^A) \rightarrow Tez(m_v^B) & Tez(r_v^A) \rightarrow Tez(r_v^B) &
 \end{array} \quad (5.1)$$

где m_a^A - элемент класса объектов категории A , которая используется для формирования описания метамодели ВМ, используемой для описания ФТ к ИС;

m_a^B - элемент класса объектов категории B , которая используется для формирования представления ФТ к ИС на уровне информации;

m_v^A - элемент ВМ, используемой для описания ФТ к ИС;

m_v^B - элемент представления ФТ к ИС на уровне информации;

r_a^A - элемент класса морфизмов категории A , которая используется для формирования описания метамодели ВМ, используемой для описания ФТ к ИС;

r_a^B - элемент класса морфизмов категории B , которая используется для формирования представления ФТ к ИС на уровне информации;

r_v^A - элемент множества отношений, которые определены на элементах m_v^A ВМ, используемой для описания ФТ к ИС;

r_v^B - элемент множества отношений, которые определены на элементах m_v^B представления ФТ к ИС на уровне информации;

Tez - тезаурус, который используется для описания элементов m_v^A ВМ, используемой для описания ФТ к ИС, или элементов m_v^B представления ФТ к ИС на уровне информации;

$MTez$ - тезаурус метамодели ВМ, используемой для описания ФТ к ИС, или метамодели представления ФТ к ИС на уровне информации, для которых тезаурус Tez является частным случаем, $Tez \subseteq MTez$.

Тогда единый комплекс знаний и правил, который обеспечивает реализацию схем переходов (5.1), можно определить как совокупность следующих баз правил [209]:

- переход $MTez(m_a^A) \rightarrow MTez(m_a^B)$ как совокупность баз правил, определяемых на основе функтора $\Phi_{m_a^A}^{m_a^B}$, который принадлежит к типу баз правил

$\Phi_{Ob^A}^{Ob^B}$, определяемых на основе ОКФ (2.35);

- переход $MTez(r_a^A) \rightarrow MTez(r_a^B)$ как совокупность баз правил, определяемых на основе $\Phi_{r_a^A}^{r_a^B}$, $\Phi_{r_v^A}^{r_v^B}$ и $\Phi_{r_t^A}^{r_t^B}$, который принадлежит к типу баз правил

$\Phi_{Mor^A}^{Mor^B}$, определяемых на основе ОКФ (2.35);

- переход $Tez(m_v^A) \rightarrow MTez(m_a^A)$ как совокупность баз правил, определя-

емых на основе функтора $\Phi_{m_v^A}^{m_a^A}$, который принадлежит к типу баз правил

$\Phi_{Ob^A}^{Ob^B}$, определяемых на основе ОКФ (2.35);

- переход $Tez(r_v^A) \rightarrow MTez(r_a^A)$ как совокупность баз правил, определяемых на основе функтора $\Phi_{r_v^A}^{r_a^A}$, который принадлежит к типу баз правил

$\Phi_{Mor^A}^{Mor^B}$, определяемых на основе ОКФ (2.35).

Исходя из сказанного выше, совокупность знаний и правил, определяющих преобразование элементов ВМ, описывающих ФТ к ИС, в представления ФТ на уровне информации, устанавливается метамоделями конкретных ВМ. Так, например, в случае применения для описания ФТ такой ВМ, как ДК языка UML, а для описания представления ФТ к ИС на уровне информации – ER-диаграммы, данная задача требует разработки их формализованных описаний и выделения общих элементов данных описаний как базиса для последующего определения формализованного описания отображения (5.1). Рассмотрим в качестве подобных общих элементов двух алфавитов классы ДК с одной стороны и сущности ER-диаграммы – с другой. Описания указанных элементов имеют одинаковую структуру. Однако, детализируя описание классов и сущностей, можно увидеть, что некоторое подмножество A'_C множества A_C атрибутов ДК не может быть адекватно отображено в множество $Attr_E$ атрибутов ER-диаграммы. Это связано с тем, что атрибут в ER-диаграмме обязан быть атомарной единицей, а в ДК атрибут может не являться атомарным. Таким образом, ДК напрямую можно отобразить в ER-диаграмму, описывающую представление ФТ к ИС на уровне информации, только в том случае, когда типы атрибутов T_j , определенные для всей ДК, можно отобразить в типы атрибутов D^j , определенные для всей ERD [166].

Для преодоления этой и других подобных проблем конвертации метамodelей языков и нотаций ВМ требований к ИС в разработанное описание представления

ФТ к ИС на уровне информации был проведен анализ нотаций наиболее распространенных видов ВМ. В результате данного анализа был выделен следующий набор общих понятий, составляющий единый алфавит их описания: «экземпляр», «сущность», «сообщение» и «атрибут». Остальные элементы ВМ являются частными случаями, расширяющими приведенные базовые [134].

Использование предложенных концептов описания единого алфавита множества ВМ позволяет реализовать представление ФТ к ИС на уровне информации в виде схемы данных, фрагмент которой приведен на рис. 5.6 [210]. На данной схеме опущен ряд незначительных элементов, серьезно не изменяющих суть предлагаемого решения.

Первая группа таблиц, выделенная в виде фрагмента APP_PROJECT, содержит общие сведения о проекте разрабатываемой ИС. Базовым понятием ИТ-проекта (таблица Project) является ИТ-сервис (таблица Service), который рассматривается как составная часть ИТ-услуги и в свою очередь может быть декомпозирован на отдельные функциональные операции введения, сохранения, обработки и формирования выходных данных (таблицы Functional_operations, Functional_operations_types). Таблицы данной схемы могут быть автоматически заполнены при обработке IDEF0, EDEF3, Use case и Activity диаграмм (таблица Diagramm) соответствующего уровня абстракции (таблица File_in_project). Следует отметить отсутствие прямой взаимосвязи между таблицами Project и Service – важным аспектом предлагаемой ИТ является использование концепта Про «сервис» в качестве первичного понятия.

Поскольку каждая диаграмма публикуется как отдельный файл, описание диаграммы как файла приведено в таблице Files фрагмента APP_PUBLICATION. При этом более расширенное описание диаграммы как источника знаний приведено в таблицах фрагмента APP_DIAGRAMM.

Таблицы Diagramm и DiagrammTypes описывают виды диаграмм и CASE-средства в которых они были сформированы. Связь диаграммы с ИТ-сервисами и отдельными функциональными операциями отображают таблицы Described_tasks и Described_operations соответственно.

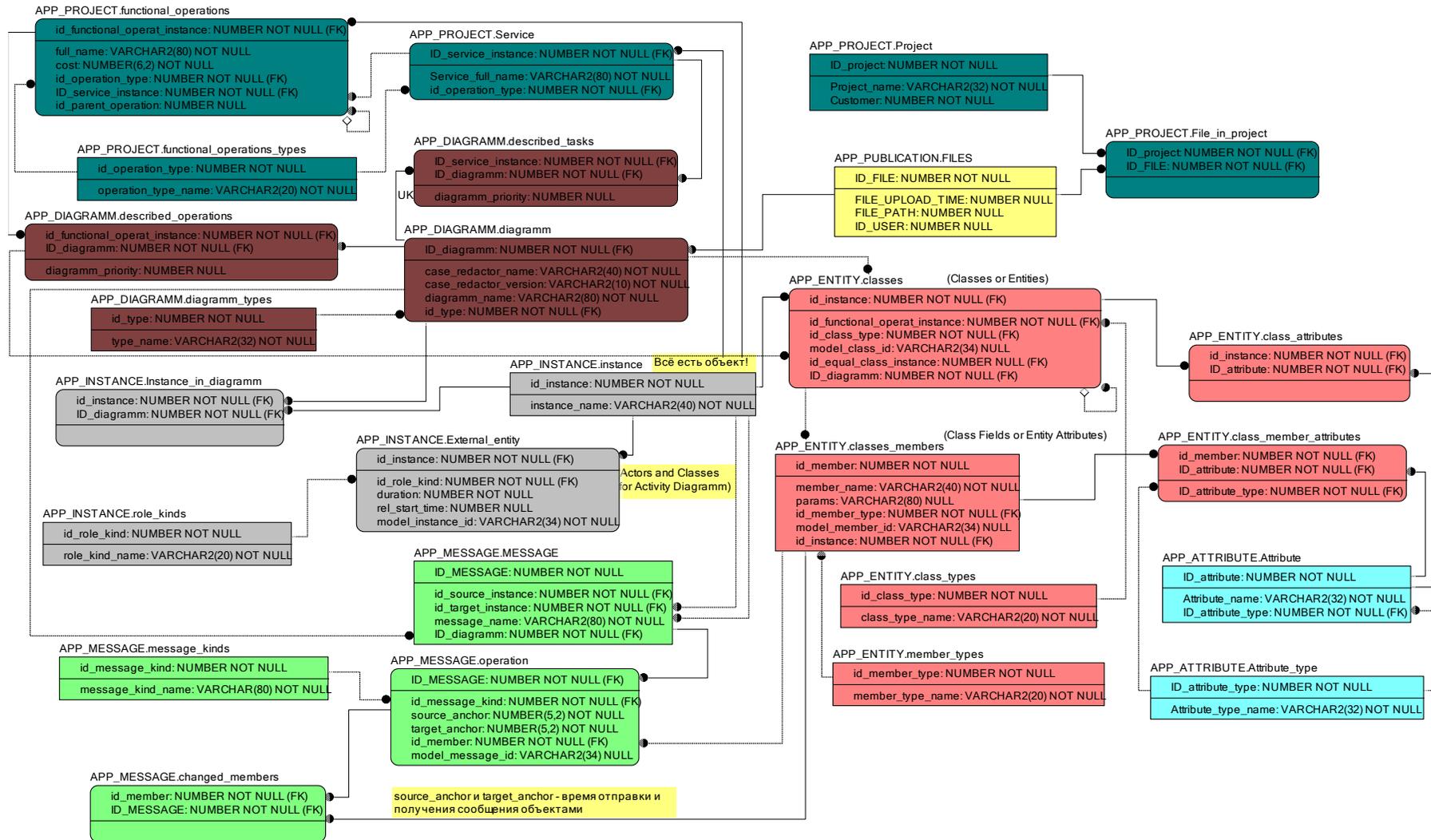


Рисунок 5.6 – Диаграмма «сущность – связь» фрагмента схемы данных для хранения описаний представлений требований к информационной системе на уровне информации

Содержимое ВМ с применением единого набора базовых элементов (общего алфавита) описывается таблицами фрагментов APP_INSTANCE, APP_ENTITY, APP_ATTRIBUTE и APP_MESSAGE. Наборы заполняемых таблиц зависят от типа диаграммы.

Наиболее абстрактным базовым понятием является «экземпляр» (таблица Instance). Данное понятие описывает объекты, используемые как в статических, так и поведенческих диаграммах. Наличие «экземпляра» на диаграмме и его роль в ней отображают таблицы Instance_in_diagramm, External_entity (является слабой сущностью, производной от таблицы Instance при эмуляции наследования в реляционной БД) и Role_kinds.

Понятие «сущность» наследуется от понятия «экземпляр» и расширяет его. В фрагменте APP_ENTITY под сущностями понимаются записи в таблицах БД и экземпляры классов ПО (таблица Classes, которая является слабой сущностью, производной от таблицы Instance), которые включают наборы операций и атрибутов (таблицы Classes_members и Classes_attributes). В свою очередь, операции сущностей делятся на процедуры и функции (таблица Member_types) и также могут иметь наборы атрибутов (таблица Classes_member_attributes), которые описываются и классифицируются на входные, выходные и хранимые в фрагменте APP_ATTRIBUTE (таблицы Attribute и Attribute_type).

Таблица Class_types используется для классификации классов и сущностей (сильные и слабые сущности, абстрактные классы, интерфейсы и т.д.).

На диаграммах, описывающих БП или поведение системы, взаимодействие «экземпляров» между собой осуществляется с помощью «сообщений» (фрагмент APP_MESSAGE). Подвидом сообщений (таблица Message) является сообщение между классами, предполагающее синхронный или асинхронный (таблица Message_kind) вызов определенного метода класса (таблица Operation, которая является слабой сущностью, производной от таблицы Message). Такое сообщение может привести к изменению состояния объекта класса и значений его свойств (таблица Changed_members).

Рассмотренная схема данных является частью схемы данных ИТ формирования и анализа требований к ИС и обеспечивает унификацию набора структурных элементов основных видов визуальных диаграмм, автоматизацию их обработки и наполнения схемы данными, полученными в результате анализа диаграмм. Извлечение данных из диаграмм обеспечивает возможность автоматического формирования SQL-запросов на сохранение метаданных о требованиях в процессе анализа представлений требований на уровне информации, но не исключает и участие аналитика. Соответственно, автоматизация обработки диаграмм значительно повышает требования к качеству документирования ФТ к ИС, актуальности и достоверности формируемой документации.

5.3 Хранение и обработка представлений требований на уровне знаний

В рамках ИТ формирования и анализа требований к ИС представления требований на уровне знаний можно разделить на два подвида. Первый подвид в ходе реализации ИТ получил название «аналитические требования», поскольку представления данного подвида формируются в результате аналитической обработки представлений требований к создаваемой ИС на уровне информации. Вторым подвидом – это представления ранее реализованных требований, доведенных до практической реализации в виде ИТ-сервисов ИС. Фрагмент схемы данных, реализующей описание выделенных подвидов представлений требований, приведен на рис. 5.7.

Для удобства восприятия на рис. 5.7 был использован следующий подход к расположению таблиц схемы данных:

- а) таблицы, описывающие аналитические требования, расположены над разделяющей чертой;
- б) таблицы, описывающие ранее реализованные требования, расположены под разделяющей чертой;

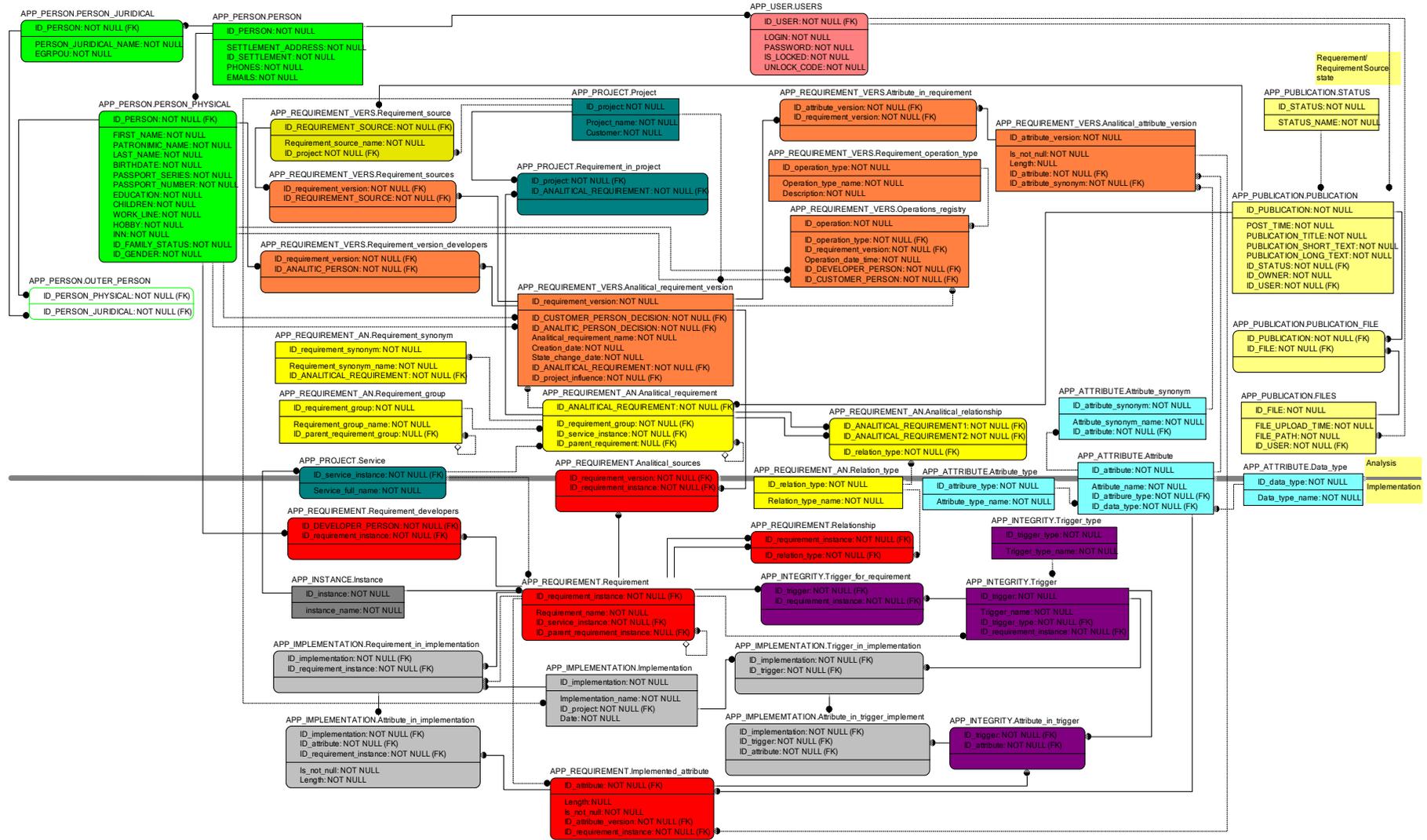


Рисунок 5.7 – Диаграмма «сущность – связь» фрагмента схемы данных для хранения описаний представлений требований к информационной системе на уровнях знаний и данных

в) черта пересекает таблицы, используемые при описании двух подвидов требований одновременно.

Следует отметить, что предлагаемый на рис. 5.7 фрагмент является одним из вариантов реализации рассмотренных в подразд. 3.4 структурных ППТ к ИС на уровне знаний. На данной схеме опущен ряд незначительных элементов, серьезно не изменяющих суть предлагаемого решения.

Согласно приведенной на рис. 5.7 схеме данных, формирование сети фреймов, описывающих представления ФТ в виде знаний, осуществляется для аналитических требований на основе таблиц фрагментов APP_REQUIREMENT_AN, APP_REQUIREMENT_VERS и APP_ATTRIBUTE. Для ранее реализованных требований формирование сети фреймов, описывающих представления ФТ в виде знаний, осуществляется на основе таблиц фрагментов APP_REQUIREMENT и APP_ATTRIBUTE. В общем случае реализация рассмотренных в п. 4.2.2-4.2.5 методов формирования представлений ФТ к ИС на уровне знаний осуществляется следующим образом:

а) для аналитических требований:

- содержимое таблиц Analitical_requirements и Analitical_requirement_version отображается в описания названий фреймов и иерархических связей между ними;

- содержимое таблицы Analitical_relationship отображается в описания горизонтальных связей между фреймами;

- содержимое таблиц Attribute, Analitical_attribute_version и Attribute_in_requirement отображается в описания атрибутов фреймов;

б) для ранее реализованных требований:

- содержимое таблицы Requirement отображается в описания названий фреймов и иерархических связей между ними;

- содержимое таблиц Relationship отображается в описания горизонтальных связей между фреймами;

- содержимое таблиц Attribute и Implemented_attribute отображается в описания атрибутов фреймов.

При отображении представлений ФТ к ИС на уровне информации в представления этих же требований на уровне знаний осуществляется формирование иерархий описаний аналитических требований. При этом введенное ранее понятие «экземпляр» отображается в понятие «аналитическое требование», которое в свою очередь является производным от понятия «публикация». Другими словами, при создании представления требования на уровне знаний оно публикуется в системе УТ (таблица `Analitical_requirement`, которая является слабой сущностью, производной от таблицы `Publication`). При этом таблицы `Publication` и `Analitical_requirement` (рис. 5.7) заполняются на основе таблицы `Instance` и её производных. Как и любая публикация, аналитическое требование имеет состояние, определяющее набор допустимых операций, которые с ним могут осуществляться (таблица `Publication_status`).

Аналитические требования, в свою очередь, подразделяются на функциональные и нефункциональные (таблица `Requirement_group`). Из представления требования на уровне информации также ре-используются сведения о входных и выходных атрибутах требований (таблицы `Attribute` и `Attribute_type`), которые также характеризуются типами данных (таблица `Data_type`).

Любое требование к ИС определяет особенности механизмов и алгоритмов обработки входных и формирования выходных данных. Сведения об этих особенностях, как правило, описываются визуальными диаграммами только на уровне представления требований в виде информации. Алгоритмы, описанные соответствующими визуальными диаграммами, реализуются непосредственно в коде ПО и ИО ИС. Поэтому представление требований на уровне знаний расширяется набором файлов с упомянутыми диаграммами, прикрепленными к публикации требования (таблицы `Files` и `Publication_files`).

Файлы с ВМ также являются «публикациями» и одновременно выступают в роли источников аналитических требований (таблица `Requirement_source`, которая является слабой сущностью, производной от таблицы `Publication`, и `Requirement_sources`, обеспечивающая связь с множеством версий требований

множества источников, на основе которых они были созданы). Источники требований могут быть опубликованы любым пользователем ИТ (таблица Users, является слабой сущностью, производной от таблицы Person) независимо от того, является ли он сотрудником Поставщика или Потребителя.

Кроме того, аналитические требования могут дополняться и уточняться, образуя иерархические и горизонтальные связи между требованиями (таблицы *Analitical_relationship* и *Relation_type*).

Поскольку одно и то же аналитическое требование может использоваться и уточняться в нескольких ИТ-проектах, на рис. 5.7 отражена связь аналитического требования с проектами (таблицы *Project* и *Requirement_in_project*). Эти сведения также могут быть получены из представления требований на уровне информации. Таблицы фрагмента схемы данных APP_PROJECT применяются для описания ИТ-проекта и его структурной декомпозиции при проектировании системы.

Наиболее поздняя и актуальная версия аналитического требования является его текущим описанием. Поскольку каждое требование характеризуется набором входных, хранимых и выходных атрибутов, набор и характеристики которых могут изменяться от версии к версии, в ИТ формировании и анализа требований к ИС атрибуты (таблица *Attribute*) также имеют версии (*Analitical_attribute_version*) и характеризуют не требование в целом, а версию аналитического требования (*Attribute_in_requirement*).

Сведения об особенностях процессов обработки данных могут быть извлечены из структурных и объектных ВМ, описывающих требования на уровне информации (рис. 5.6). Их описание в виде знаний может быть реализовано с применением каузальных сценариев. В таком случае информационные единицы сети фреймов, описывающей требование, могут быть дополнены сигнатурами присоединенных процедур фреймов, описанными в виде сценариев.

Для описания библиотеки ранее реализованных требований в ИТ формировании и анализа требований к ИС используется такое базовое понятие,

как IT-сервис (таблица Service, которая является слабой сущностью, производной от таблицы Instance). Как отдельные требования, так и IT-сервисы или даже IT-услуги в целом могут быть ре-использованы в различных проектах. При этом принимается утверждение о том, что одно и то же требование может быть реализовано одним и только одним IT-сервисом системы. Это утверждение отражено в таблице Requirement (рис. 5.7), которая описывает реализованное требование. Эта таблица также является слабой сущностью, производной от таблицы Instance. Перечень атрибутов, созданных при разработке требований, отражен в таблице Implemented_attribute, которая является слабой сущностью по отношению к таблице Analitical_attribute_version. Связь реализованных атрибутов и требований с аналитическими (таблица Analitical_sources) отражает механизм физической реализации требований на базе одной или нескольких аналитических версий, состояния которых по окончании разработки фиксируются.

Аналогично аналитическим требованиям, в ИТ учитываются связи между реализованными требованиями (таблица Relationship) и разработчиками физически реализованных требований (таблица Requirement_developers). Описание реализованных требований отражается в таблицах фрагмента схемы APP_REQUIREMENT.

Реализованные требования тоже не являются статичными. При разработке новых проектов реализованные требования также могут модифицироваться и совершенствоваться. Таблицы фрагмента схемы APP_REQUIREMENT содержат актуальное состояние библиотеки ранее реализованных требований, то есть описание универсума требований Поставщика. При внедрении ИС (таблица Implementation), реализующей заданный Потребителем набор требований (таблица Requirement_in_implementation) с заданным набором атрибутов (Attribute_in_implementation) фиксируется внедряемое подмножество универсума требований Поставщика, актуальное на момент внедрения. Это позволяет автоматизировать расширение и обновление внедренных версий ИС при разви-

тии типовой системы Поставщика, то есть обеспечить ре-использование требований при развитии и сопровождении внедренной системы. Для описания требований, реализованных в версиях ИС, внедрённых на конкретных ОА, используются таблицы фрагмента схемы APP_IMPLEMENTATION.

Объекты фрагмента схемы APP_INTEGRITY и их аналоги в фрагменте схемы APP_IMPLEMENTATION применяются для описания ограничений целостности данных, реализуемых с помощью триггеров, и используются при объектно-реляционном отображении представлений требований на уровне знаний в элементы ПО и ИО ИС.

Реализация предложенных в подразд. 4.5 ОТ обусловлена рекурсивными связями: для аналитических требований – таблицы Analytical_requirements; для ранее реализованных требований – таблицы Requirement. Наличие иерархических связей между объектами ПрО обуславливает образование фреймами таксономий, которые дополнительно могут быть соединены между собой горизонтальными связями. Полученная таким образом сеть фреймов, описывающая требования, будет иметь вид «низкорослого леса» или «кустарника», в соответствии со связями объектов ПрО, для которой проектируется ИС. Как отмечено в подразд. 4.5, такая ОТ в ходе создания ПО IT-сервисов ИС отражается в иерархию классов-наследников одного базового (часто абстрактного) класса, а в ходе создания ИО – в схемы данных типа «звезда» или «снежинка», в которых консольные таблицы и таблицы измерений сгруппированы вокруг базовой таблицы фактов, которую они расширяют и уточняют. Для удобства работы такие наборы таблиц могут объединяться в один фрагмент схемы данных (принадлежащий одному пользователю). Примерами таких фрагментов на рис. 5.6 и рис. 5.7 являются фрагменты APP_PERSON, APP_PROJECT и т.п., названия которых идентичны наименованиям соответствующих таблиц фактов.

Характерным свойством подобной организации схемы данных является возможность денормализации и объединения всех таблиц схемы в одно универсальное отношение, описывающее базовое понятие ПрО, соответствующее таб-

лице фактов витрины данных. Такое универсальное отношение может быть создано, например, в виде представления (виртуальной таблицы). Применение универсальных отношений позволяет изменять масштаб моделирования требований к ИС, оперируя реализованными требованиями как на уровне подтребований, отдельных таблиц БД и классов ПО, так и на уровне отдельных требований к ИС, физически представленных ИТ-сервисами или их компонентами.

Вследствие этого одной из наиболее сложных задач при формировании онтологии ПрО в виде сети фреймов является выбор фреймов, являющихся корневыми элементами ОТ, и принятие решения о наследовании фреймов от существующих или о формировании нового корня ветви таксономии фреймов. Для решения данной задачи введем следующее правило: если между понятиями А и В существует связь типа «1..∞», «0..∞», «1..m», «0..m», «m..m» или «∞..∞», то такая связь является признаком выделения понятия В как самостоятельного, а не наследуемого от А. Выполнение данного правила обеспечит декомпозицию иерархии требований на отдельные узлы онтологий. Реализация таких онтологий целесообразна в виде отдельных сервисов и обеспечит возможность их двустороннего отображения в ПО и реляционные схемы данных, в которых в явном виде иерархические связи между таблицами отсутствуют.

Как отмечено в подразд. 5.1, предлагаемая ИТ формирования и анализа требований к ИС должна являться в определённой степени самоописательной. Следовательно, решения по ИО и ПО данной ИТ должны определяться моделями и методами, рассмотренными в разд. 2, разд. 3 и разд. 4 данной работы.

Результатом такого влияния на ИО ИТ формирования и анализа требований к ИС является разделение схемы данных ИТ на отдельные фрагменты (множества объектов БД, принадлежащие одному пользователю), соответствующие реализованным требованиям к ИТ, а именно: APP_PROJECT, APP_DIAGRAMM, APP_PUBLICATION, APP_INSTANCE, APP_ENTITY, APP_ATTRIBUTE, APP_MESSAGE, APP_REQUIREMENT_AN, APP_REQUIREMENT_VERS, APP_REQUIREMENT, APP_IMPLEMENTATION, APP_INTEGRITY,

APP_PERSON, APP_USER. Каждый из этих фрагментов схем организован по типу витрины данных, в которой обязательно присутствует таблица фактов и могут присутствовать уточняющие или расширяющие её таблицы. Такая группировка классов и таблиц БД осуществлена по принципу выделения наиболее абстрактного и независимого базового объекта, максимально независимого от ПрО проектируемой ИС, который потом детализируется наследниками в соответствии со спецификой конкретной ПрО и ОА.

Данный подход обеспечивает возможность ре-использования требований, сформулированных в ходе создания ИТ формирования и анализа требований к ИС, в процессе формирования и анализа ФТ других ИТ-проектов. Такое ре-использование будет происходить, когда базовый концепт ИТ детализируется концептом-наследником из ПрО создаваемой ИС, соответствующим или детализирующим исходный концепт в другом ИТ-проекте создания ИС. Использование данного подхода позволяет Поставщику использовать выгоды от библиотеки ранее реализованных требований даже в случае отказа от специального заполнения ее описаниями требований ранее выполненных ИТ-проектов создания ИС.

5.4 Хранение и обработка представлений требований на уровне данных

Как показано в подразд. 3.2, ППТ на уровне данных позволяют формализовать процессы идентификации, учета и управления ФТ и их представлениями на уровнях информации и знаний. При этом следует также учесть, что в рамках ИТ формирования и анализа требований к ИС представление требований на уровне знаний разделено на два подвида – аналитические требования, формируемые в результате обработки представлений ФТ на уровне информации, и представления ранее реализованных требований, доведенных до практической реализации в виде элементов ИС. Каждый из этих подвидов на

уровне данных описывается набором записей в таблицах БД, отражающих определённые характеристики требований. Фрагмент схемы данных, реализующей подобное описание, приведен на рис. 5.7.

Следует отметить, что предлагаемый на рис. 5.7 фрагмент является одним из вариантов реализации рассмотренных в подразд. 3.2 структурных ППТ, определяющих подход к описанию представлений требований к ИС на уровне данных. На данной схеме опущен ряд незначительных элементов, серьезно не изменяющих суть предлагаемого решения.

Для описания особенностей реализации управления требованиями к ИС с помощью представлений требований на уровне данных необходимо определить понятие ЖЦ требования, этапы которого будут отражены в ИТ формировании и анализа требований к ИС.

Создание ИС, независимо от выбранной модели ЖЦ, методологии проектирования или АФ, включает процессы формирования и анализа требований к системе. Инициация разработки новой ИС означает старт нового проекта (таблица Project). В результате анализа ОА и формирования описаний ФТ к создаваемой ИС на уровне информации формируются аналитические требования (Analytical_requirement). Следующей стадией ЖЦ аналитического требования является уточнение требования и формирование различных версий аналитических требований (Analytical_requirement_version). При этом работа с этими версиями требований требует реализации в ИТ некоторых функций, характерных для систем контроля версий. Кроме того, при ре-использовании одних и тех же требований в различных проектах также могут создаваться различные версии требования, в результате чего аналитическое требование приобретает черты, характерные для нескольких проектов. Поэтому для реализации паттерна идентификации требования к ИС предлагается применять таблицы фрагмента схемы APP_REQUIREMENT_AN, а для реализации паттерна идентификации версий требований – таблицы фрагмента схемы APP_REQUIREMENT_VERS.

Текущим описанием аналитического требования является его наиболее

поздняя и актуальная версия. Как и в системе контроля версий, у каждой версии требования есть разработчики (таблица `Requirement_version_developers`), которые могут являться как сотрудниками Поставщика (таблица `Person_physical`, которая является слабой сущностью, производной от таблицы `Person`), так и сотрудниками Потребителя (таблицы `Outer_person`, которая является слабой сущностью, производной от таблицы `Person_physical`, и `Person_juridical`, которая является слабой сущностью, производной от таблицы `Person`). Для описания сведений о физических и юридических лицах, участвующих в процессе формирования и разработки требования, используются таблицы фрагмента схемы `APP_PERSON`.

Поскольку каждое требование характеризуется набором входных, хранимых и выходных атрибутов, набор и характеристики которых могут изменяться от версии к версии, в ИТ формирования и анализа требований к ИС описания атрибутов (таблица `Attribute`) также имеют версии (`Analytical_attribute_version`), характеризующие не требование в целом, а версию аналитического требования (`Attribute_in_requirement`).

Одно и то же требование в терминах Про различных ИС может иметь различные названия. Поэтому в состав ИТ также входит функция управления синонимами названий требований и их атрибутов (таблицы `Requirement_synonym` и `Attribute_synonym`). Для описания атрибутов требований применяются объекты фрагмента схемы `APP_ATTRIBUTE`.

Как и в системе управления версиями, в рассматриваемой ИТ необходима реализация паттерна учета результатов выполнения операций над требованиями к ИС в виде фрагмента схемы данных, обеспечивающего аудит действий пользователей по формированию и уточнению аналитических требований (таблицы `Operation_registry` и `Requirement_operation_type`).

Следующей стадией ЖЦ требования является его реализация в качестве элементов обеспечений ИТ-сервиса ИС. На этой стадии аналитическому требованию ставится в соответствие его практическая реализация в виде прототипа – ИТ-сервиса ИС или его элемента. При этом базовым концептом ИТ формирования и

анализа требований к ИС является понятие «IT-сервис». Данное понятие реализовано в виде таблицы Service, которая является слабой сущностью, производной от таблицы Instance. Как отдельные требования, так и IT-сервисы или даже IT-услуги в целом могут быть ре-использованы в различных проектах. При этом принимается утверждение о том, что одно и то же требование может быть реализовано одним и только одним IT-сервисом системы. Это утверждение воплощено в виде таблицы Requirement (рис. 5.6), которая описывает реализованное требование. Эта таблица также является слабой сущностью, производной от таблицы Instance. Перечень атрибутов, созданных при разработке требований, отражен в таблице Implemented_attribute, которая является слабой сущностью по отношению к таблице Analitical_attribute_version. Связь реализованных атрибутов и требований с аналитическими (таблица Analitical_sources) отражает механизм физической реализации требований на базе одной или нескольких аналитических версий, состояния которых по окончанию разработки фиксируются.

Аналогично аналитическим требованиям в ИТ учитываются связи между реализованными требованиями (таблица Relationship) и разработчиками физически реализованных требований (таблица Requirement_developers). Описание реализованных требований отражается в таблицах фрагмента схемы APP_REQUIREMENT.

Реализованные требования тоже не являются статичными. При разработке новых проектов реализованные требования также могут модифицироваться и совершенствоваться. Таблицы фрагмента схемы APP_REQUIREMENT содержат актуальное состояние библиотеки ранее реализованных требований, то есть описание универсума требований Поставщика. При внедрении ИС (таблица Implementation), реализующей заданный Потребителем набор требований (таблица Requirement_in_implementation) с заданным набором атрибутов (Attribute_in_implementation) фиксируется внедряемое подмножество универсума требований Поставщика, актуальное на момент внедрения. Это позволяет автоматизировать расширение и обновление внедренных версий ИС при развитии типовой ИС Поставщиком, то есть обеспечить ре-использование требова-

ний при развитии и сопровождении внедренной системы. Для описания требований, реализованных в версиях ИС, внедрённых на конкретных ОА, используются таблицы фрагмента схемы APP_IMPLEMENTATION.

Объекты фрагмента схемы APP_INTEGRITY и их аналоги в фрагменте схемы APP_IMPLEMENTATION применяются для описания ограничений целостности данных, реализуемых с помощью триггеров, и используются при объектно-реляционном отображении представлений требований на уровне знаний в элементы ПО и ИО ИС.

Последующие стадии ЖЦ реализованного требования предполагают его применение и развитие в ходе создания новых ИС в качестве элемента библиотеки готовых компонентов Поставщика.

Таким образом, рассмотренные в п. 1.2.5 состояния требований к ИС, определяемые ЖЦ требования, должны быть расширены. Для предлагаемой ИТ формирования и анализа требований к ИС ЖЦ требования к ИС, как показано выше, включает стадии, приведенные в табл. 5.1 [211].

Данные стадии определяют необходимость включения в ИТ формирования и анализа требований к ИС некоторых функций систем управления проектами, систем управления содержимым и систем контроля версий (или предполагают интеграцию с системами данных классов), а именно:

- функции управления сведениями о проектах разработки ИС и их структурной декомпозиции (функциональность систем управления проектами);
- функции управления текстовыми и графическими описаниями на уровне информации в виде публикуемых в системе представителями Поставщика и Потребителя файлов (функциональность CMS-систем);
- функции учета и контроля версий аналитических и внедрённых требований и их атрибутов, а также осуществляемых над ними операций (функциональность систем контроля версий).

Компоненты ИТ, реализующие эти функции, также основаны на фрагментах схемы данных, показанных на рис. 5.6 и 5.7.

Таблица 5.1 –Перечень стадий жизненного цикла функционального требования к информационной системе

№ стадии	Название стадии
1	Инициация проекта создания ИС Поставщиком
2	Описание требований на уровне информации в виде текстовых документов, ВМ и т.п.
3	Отображение требований в представления на уровне данных и знаний (формирование первых версий аналитических требований)
4	Анализ и уточнение аналитических требований, их атрибутов и связей между ними (в том числе, формирование новых версий аналитических требований)
5	Разработка реализуемых требований на базе фиксированных версий аналитических требований
6	Внедрение реализованных требований и фиксация описаний внедрённых требований
7	Формирование библиотеки ранее реализованных требований для их ре-использования
8	Ре-использование и развитие аналитических и реализованных требований в ходе создания новых ИС, развития и сопровождения внедренных ИС
9	Удаление нереализованных или неиспользуемых аналитических и реализованных требований или их версий

Предлагаемая реализация фрагмента ИТ, обеспечивающего обработку и хранение представлений ФТ к ИС на уровне данных, позволяет автоматизировать процессы идентификации требований, учета операций, выполняемых над требованиями в рамках их ЖЦ, а также процессы управления ФТ в ходе выполнения ИТ-проектов создания ИС различного назначения до окончания ЖЦ соответствующих ФТ.

5.5 Выводы к пятому разделу

1. Уточнено определение ИТ формирования и анализа требований к ИС как совокупности унифицированных в рамках АФ макропроектирования ИС и основанных на знаниях методов, приемов и способов применения КСА или его отдельных элементов, используемой для выполнения одной или нескольких работ в рамках процессов создания ИС, непосредственно работающих с требованиями к данной системе. Исходя из данного определения, сформулирована концепция (рис. 5.1) и выделены основные концепты Про ИТ (рис. 5.2, рис. 5.3). На их основе были установлены основные ИТ-услуги предлагаемой ИТ (рис. 5.5).

2. Сформулирована задача унификации описаний ФТ на уровне информации в виде ВМ и предложены варианты ее решения. Исходя из этих вариантов, разработаны решения по ИО ИТ формирования и анализа требований к ИС, реализующие разработанные в подразд. 3.3 ППТ и обеспечивающие хранение и обработку представлений ФТ к ИС на уровне информации (рис. 5.6).

3. Для описания ФТ в рамках ЖЦ ИС выделены основные подвиды представлений ФТ к ИС на уровне знаний – аналитические требования и ранее реализованные требования. На основе выделенных подвидов разработаны решения по ИО ИТ формирования и анализа требований к ИС, обеспечивающие хранение и обработку представлений ФТ к ИС на уровне знаний (рис. 5.7).

4. Для организации управления ФТ к ИС в рамках ИТ формирования и анализа требований к ИС предложено описание ЖЦ ФТ. Исходя из предложенного описания, разработаны решения по ИО ИТ формирования и анализа требований к ИС, реализующие разработанные в подразд. 3.2 ППТ и обеспечивающие хранение и обработку представлений ФТ к ИС на уровне данных (рис. 5.6, 5.7).

Основные результаты данного раздела изложены в работах [134, 203-207, 209-211].

РАЗДЕЛ 6

РЕАЛИЗАЦИЯ И АПРОБАЦИЯ ИНФОРМАЦИОННОЙ ТЕХНОЛОГИИ ФОРМИРОВАНИЯ И АНАЛИЗА ТРЕБОВАНИЙ К ИНФОРМАЦИОННОЙ СИСТЕМЕ

6.1 Реализация информационной технологии формирования и анализа требований к информационной системе

Предложенное в подразд. 2.6 подмножество ИТ формирования и анализа требований к ИС (2.45), частным случаем которого является описанная в разд. 5 ИТ, допускает множество вариантов реализации каждого из своих элементов. Однако рассмотренные в разд. 5 особенности ИТ формирования и анализа требований к ИС, определяемые разработанными в разд. 3 и разд. 4 моделями и методами, позволяют уточнить основные требования к реализации данной ИТ. Среди этих требований необходимо в первую очередь выделить:

а) возможность использования для публикаций требований как инструментов самой ИТ, так и наиболее распространенных CASE-средств визуального моделирования ИС;

б) необходимость обеспечения самоописания ИТ формирования и анализа требований к ИС;

в) необходимость хранить и обрабатывать одними и теми же инструментами аналитические и ранее реализованные требования, описывающие представления требований на уровне знаний Потребителя и Поставщика соответственно;

г) необходимость выполнения предложенного в подразд. 5.4 ЖЦ УТ к ИС.

Эти и другие требования к реализации ИТ формирования и анализа требований к ИС определяют необходимость отказа от реализации рассмотренных в разд. 4 методов формирования и анализа представлений требований к ИС исключительно в виде программных модулей. Такой отказ требует модификации шаблонов проектирования, которые могут быть использованы в ходе разработки обеспечивающей части ИТ формирования и анализа требований к ИС. Так,

например, для применения шаблона проектирования «Model – View – Controller» (MVC) [150] его следует изменить в соответствии с предложенной в [212, 213] концепцией. Согласно данной концепции, сервис, реализующий ИТ-услугу, должен включать в себя на уровне компонента «Model» не только модуль, обеспечивающий взаимодействие с БД, но и модуль, обеспечивающий формирование и управление метаданными [213]. Поэтому основные особенности реализации ИТ формирования и анализа требований к ИС будут определяться особенностями реализации компонента «Model» шаблона проектирования MVC.

Особенности реализации компонента «Model» будут определяться, главным образом, теми представлениями требований к ИС, для работы с которыми разрабатываются элементы обеспечивающей части ИТ, а также особенностями формирования данных представлений. Так, для реализации разработанного в п. 4.2.1 метода формирования представления СТ на уровне информации следует принять во внимание, что большинство современных CASE-средств, используемых для создания ВМ в ходе создания ИС, позволяют экспортировать данные о ВМ в документы формата XML и предоставляют формальные описания DOM таких документов в виде DTD или XSD [149, 210]. Данные способы формального описания ВМ позволяют автоматизировать их обработку с целью извлечения знаний.

Упомянутая в подразд. 5.2 необходимость применения единого алфавита обуславливает формирование описания множества диаграмм ВМ ФТ к ИС с использованием единого набора базовых элементов, то есть наполнение схемы данных, описывающей набор визуальных диаграмм, применяемых в проекте и описывающих отдельные элементы ПрО или ИС. Поскольку исходным набором данных для этого описания являются XML-документы с открытыми XSD или DTD, извлечение из них данных и наполнение схемы данных, приведенной на рис. 5.6, являются инженерными задачами, предполагающими обработку DOM XML-документа и формирование на основе данных, содержащихся в XML-документе, запросов на заполнение БД. В рамках ИТ формирования и анализа требований к ИС данные задачи реализованы средствами XSL (рис. 6.1-6.3).

```

<?xml version="1.0" encoding="windows-1251" standalone="yes"?>
<?xml-stylesheet type="text/xsl" href="ClassDiagram.xsl"?>
<UMLdiagram DiagramName="Dean Kontingent" DiagramType="ClassDiagram" UMLredactor="ModelMaker 7" UMLredactorVersio
  <classsymbol ShapeID="1" InstanceName="DomainObject">
    <references>
      <codemodelref EntityType="tyClass" ID="54" NamePath="TDomainObject"/>
    </references>
    <list Name="members">
      <member Name="Caption" ID="1129"/>
      <member Name="Create" ID="1120"/>
      <member Name="GetValue" Params="AKeyID: Integer" ID="1123"/>
      <member Name="Id" ID="1127"/>
      <member Name="QueryInterface" Params="const IID: TGUID; out Obj" ID="1124"/>
      <member Name="SetValue" Params="AKeyID: Integer; Value: Variant" ID="1126"/>
      <member Name="_AddRef" ID="1118"/>
      <member Name="_Release" ID="1119"/>
    </list>
  </classsymbol>
  <classsymbol ShapeID="3" InstanceName="IaList">
    <references>
      <codemodelref EntityType="tyClass" ID="48" NamePath="TiaList"/>
    </references>
    <list Name="members">
      <member Name="FindItemById" Params="AID: integer" ID="1137"/>
      <member Name="GetItemObject" ID="1138"/>
      <member Name="GetItems" Params="Index: Integer" ID="1139"/>
      <member Name="Items" ID="1142"/>
      <member Name="Notify" Params="Ptr: Pointer; Action: TListNotification" ID="1140"/>
      <member Name="SetItems" Params="Index: Integer; const Value: TDomainObject" ID="1141"/>
    </list>
  </classsymbol>
  <propertyassociation ShapeID="5" SourceSymbol="3" TargetSymbol="1">
    <references>
      <codemodelref EntityType="tyMember" ID="1142" NamePath="TiaList.Items"/>
    </references>
  </propertyassociation>
  <classsymbol ShapeID="8" InstanceName="CustomMan">
    <references>
      <codemodelref EntityType="tyClass" ID="12" NamePath="TCustomMan"/>
    </references>
    <list Name="members">
      <member Name="BirthDate" ID="75"/>
      <member Name="FullFio" ID="77"/>
      <member Name="GetValue" Params="AKeyID: Integer" ID="63"/>
      <member Name="Name" ID="67"/>
      <member Name="Patronym" ID="69"/>
      <member Name="SetValue" Params="AKeyID: Integer; Value: Variant" ID="64"/>
      <member Name="Sex" ID="71"/>
      <member Name="SexStr" ID="73"/>
      <member Name="Surname" ID="65"/>
    </list>
  </classsymbol>
  <generalization ShapeID="10" SourceSymbol="8" TargetSymbol="1"/>

```

Рисунок 6.1 – Фрагмент исходного кода XML-документа, описывающего UML-диаграмму классов, являющуюся публикацией функционального требования к информационной системе

Для реализации элементов ИТ формирования и анализа требований к ИС, работающих с представлениями СТ к ИС на уровне знаний, предлагается использовать подход, основанный на представлении разработанных в диссертации

```

<?xml version="1.0" ?>
- <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
- <xsl:template match="/">
  - <html>
    - <head>
      <title>SQL-script for ClassDiagramm generated by XSLT XML-document
        processing</title>
    </head>
  - <body>
    - <center>
      - <b>
        <h2>SQL-script for ClassDiagramm generated by XSLT XML-document
          processing</h2>
      </b>
    </center>
    <br>SET TERMOUT ON</br>
    <br>SET ECHO ON</br>
    <br>CONNECT MODELER/METAMODEL</br>
  - <p>
    <b>----- Saving Diagramm Info -----</b>
    <br>
  - <br>
    insert into DIAGRAMM (id_diagramm, diagramm_name, case_redactor_name,
      case_redactor_version, id_type)
  - <br>
    values ('
      <xsl:value-of select="UMLdiagram/@ModelID" />
      ',
      <xsl:value-of select="UMLdiagram/@DiagramName" />
      ',
      <xsl:value-of select="UMLdiagram/@UMLredactor" />
      ',
      <xsl:value-of select="UMLdiagram/@UMLredactorVersion" />
      ', (Select id_type from DIAGRAMM_TYPES where type_name='
      <xsl:value-of select="UMLdiagram/@DiagramType" />
      ');
  - <br>
  - <br>
    <xsl:variable name="ModelID" select="UMLdiagram/@ModelID" />
  - </p>
+ <p>
- <xsl:for-each select="UMLdiagram/classsymbol">
  <xsl:variable name="classID" select="@ShapeID" />
- <p>
  - <b>
    ----- Beginning of class section for
    <xsl:value-of select="references/codemodelref/@NamePath" />
    -----
  </b>
- <br>
  insert into CLASSES (id_class, class_name, id_diagramm, model_class_id,
    id_functional_operation, id_class_type, id_equal_class)
  .

```

Рисунок 6.2 – Фрагмент исходного кода XSLT-документа, описывающего трансформацию XML-документа в набор INSERT-команд заполнения базы данными об элементах диаграммы классов, являющейся публикацией требования к информационной системе

SQL-script for ClassDiagramm generated by XSLT XML-document processing

```
SET TERMOUT ON
SET ECHO ON
CONNECT MODELER/METAMODEL
```

```
----- Saving Diagramm Info -----
insert into DIAGRAMM (id_diagramm, diagramm_name, case_redactor_name, case_redactor_version, id_type)
values ('0CC4344B-22E3-43C2-BD0D-60BB24C26A16', 'Dean Kontingent', 'ModelMaker 7', '7.04', (Select id_type
from DIAGRAMM_TYPES where type_name='ClassDiagram'));
```

```
----- Beginning of class section for TDomainObject -----
insert into CLASSES (id_class, class_name, id_diagramm, model_class_id, id_class_type, id_equal_class)
values (classes_seq.NextVal,
'TDomainObject',
'0CC4344B-22E3-43C2-BD0D-60BB24C26A16',
'1',
(select id_class_type from class_types where lower(class_type_name)='class'),
(select id_class from classes where class_name=substr('TDomainObject',2,length('TDomainObject')) and class_type =
(select id_class_type from class_types where lower(class_type_name)='entity')));
```

```
-- params:
insert into CLASSES_MEMBERS (id_member, id_class, member_name, params, id_member_type, model_member_id)
values (members_seq.NextVal, (Select id_class from classes where model_class_id='1' and id_diagramm='0CC4344B-
22E3-43C2-BD0D-60BB24C26A16'), 'Caption', '', (select id_member_type from MEMBER_TYPES where
member_type_name='attribute'), 1129);
```

```
insert into CLASSES_MEMBERS (id_member, id_class, member_name, params, id_member_type, model_member_id)
values (members_seq.NextVal, (Select id_class from classes where model_class_id='1' and id_diagramm='0CC4344B-
22E3-43C2-BD0D-60BB24C26A16'), 'Create', '', (select id_member_type from MEMBER_TYPES where
```

Рисунок 6.3 – Результат трансформации XML-документа в набор INSERT-команд заполнения базы данными об элементах диаграммы классов, являющейся публикацией функционального требования к информационной системе

ционной работе моделей, описывающих исходную информацию и результаты применения данных методов, в виде одного или нескольких материализованных представлений. Однако воплощения этого подхода в виде конкретных проектных решений ИТ формирования и анализа требований к ИС в сильной степени зависят от следующих различий прав на использование данной ИТ:

а) если данная ИТ передается в пользование без прав на самостоятельное расширение, модификацию или адаптацию моделей, описывающих представление требований на уровнях данных, информации и знаний, а также указанных выше методов, либо в ходе эксплуатации данной ИТ предполагается отказаться от изменения МТ к ИС, то наиболее целесообразным является применение методов управления сводными данными, позволяющих представить результаты применения упомянутых выше методов в виде материализованных представлений (МП);

б) если данная ИТ передается в пользование с предоставлением прав на самостоятельное расширение, модификацию или адаптацию моделей, описывающих представления требований на уровнях данных, информации и знаний, а также упомянутых выше методов, либо в ходе эксплуатации данной ИТ предполагается возможность изменения МТ к ИС, то наиболее целесообразным является применение методов динамического SQL, позволяющих реализовать этапы упомянутых выше методов без жесткой привязки к изменяемой схеме данных ИТ.

Вариант б) более сложен в реализации, чем вариант а), однако его использование позволяет значительно продлить время эксплуатации ИТ формирования и анализа требований к ИС.

Рассмотрим особенности применения варианта а) для реализации полученных в диссертационной работе моделей и методов на примере реализации методов формирования представлений СТ на уровне знаний (см. пп. 4.2.3-4.2.6). Так, алгоритм реализации метода формирования представления СТ на уровне знаний отдельного УАП (см. п. 4.2.3) может быть представлен следующим образом.

Шаг 1. Для выбранного ранее УАП и диаграммы-публикации выдвинутого им требования к создаваемой ИС сформировать МП «`requirement_version_person_frame`», в котором будут находиться знания об индексах и наименованиях фреймов, описывающих структуры данных, которые присутствуют в диаграмме-публикации версии требования к ИС, выдвинутого конкретным УАП.

Шаг 2. Для фреймов, выделенных в ходе выполнения Шага 1, сформировать МП «`requirement_version_person_attributes`», в котором будут находиться знания об атрибутах данных фреймов.

Шаг 3. Для фреймов, выделенных в ходе выполнения Шага 1, сформировать МП «`requirement_version_person_relation`», в котором будут находиться знания о наименованиях и типах связей между данными фреймами, а также об атрибутах, участвующих в образовании этих связей.

Шаг 4. Сохранить результаты применения метода в БД ИТ формирования и анализа требований к ИС. Завершить выполнение алгоритма.

Фрагменты команд, создающих МП «`requirement_version_person_frame`», «`requirement_version_person_attributes`» и «`requirement_version_person_relation`» в ходе выполнения Шагов 1-4 рассмотренного выше алгоритма, приведены на рис. Д.1-Д.3 Приложения Д. Для выполнения Шага 4 рассмотренного выше алгоритма результаты запросов, выполняемых в ходе создания рассмотренных выше МП, следует сохранить в БД ИТ формирования и анализа требований к ИС (например, с применением стандартной конструкции «`INSERT INTO... SELECT...`»).

Для реализации метода формирования представления СТ на уровне Потребителя (см. п. 4.2.4) используются МП, аналогичные рассмотренным на рис. Д.1-Д.3 Приложения Д, а также дополнительное МП «`requirement_versions_person`». Это МП содержит данные об УАП, выдвигающих требования к создаваемой ИС со стороны Потребителя.

Применение метода формирования представления СТ на уровне знаний Поставщика (см. п. 4.2.5) предполагает сравнение знаний о требованиях, выдвигаемых к создаваемой ИС, со знаниями о требованиях, выдвинутых и успешно реализованных в предыдущих ИТ-проектах. Поэтому алгоритм реализации данного метода может быть представлен следующим образом.

Шаг 1. Для установленного ранее Потребителя создаваемой ИС, сформировать МП «`requirement_user_frame`», в котором будут находиться знания об индексах и наименованиях фреймов, описывающих структуры данных, которые присутствуют в диаграммах-публикациях требований к ИС, выдвинутых Потребителем.

Шаг 2. Сформировать МП «`requirement_provider_frame`», в котором будут находиться знания о ранее реализованных фреймах и интерфейсах, чьи наименования полностью совпадают с наименованиями соответствующих фреймов и ат-

рибутов, выделенных на Шаге 1.

Шаг 3. Сформировать МП «`requirement_provider_frame_extended`», в котором будут находиться знания о ранее реализованных фреймах и интерфейсах, чье описание является расширенным описанием соответствующих фреймов и атрибутов, выделенных на Шаге 1.

Шаг 4. Сформировать МП «`requirement_provider_attributes`», в котором будут находиться знания об атрибутах фреймов и интерфейсов, выделенных на Шаге 2 и Шаге 3.

Шаг 5. Сформировать МП «`requirement_provider_relations`», в котором будут находиться знания о связях между фреймами или между фреймами и интерфейсами, выделенными на Шаге 2 и Шаге 3.

Шаг 6. Сохранить результаты применения метода в БД ИТ формирования и анализа требований к ИС. Завершить выполнение алгоритма.

Фрагменты команд, создающих МП «`requirement_user_frame`», «`requirement_provider_frame`», «`requirement_provider_frame_extended`», «`requirement_provider_attributes`» и «`requirement_provider_relations`» в ходе выполнения Шагов 1-5 рассмотренного выше алгоритма, приведены на рис. Д.4-Д.8 Приложения Д. Для выполнения Шага 6 рассмотренного выше алгоритма результаты запросов, выполняемых в ходе создания рассмотренных выше МП, следует сохранить в БД ИТ формирования и анализа требований к ИС (например, с применением стандартной конструкции «`INSERT INTO... SELECT...`»).

Применение метода формирования общесистемного представления СТ на уровне знаний (см. п. 4.2.6) предполагает сравнение знаний о представлениях требования как с точек зрения Потребителя, так и с точек зрения Поставщика. Поэтому алгоритм реализации данного метода может быть представлен следующим образом.

Шаг 1. Создать МП «`requirement_idea`», которое будет содержать данные о СФТ к создаваемой ИС.

Шаг 2. Создать МП «`requirement_frame_user`», которое будет содержать описание фреймов и интерфейсов СФТ с точки зрения Потребителя.

Шаг 3. Создать МП «`requirement_attributes_user`», которое будет содер-

жать описание атрибутов выделенных на Шаге 2 фреймов и интерфейсов.

Шаг 4. Создать МП «requirement_relations_user», которое будет содержать описание связей между выделенными на Шаге 2 фреймами, а также между выделенными на Шаге 2 фреймами и интерфейсами.

Шаг 5. Создать МП «requirement_frame_provider», которое будет содержать описание фреймов и интерфейсов СФТ с точки зрения Поставщика.

Шаг 6. Создать МП «requirement_attributes_provider», которое будет содержать описание атрибутов выделенных на Шаге 5 фреймов и интерфейсов.

Шаг 7. Создать МП «requirement_relations_provider», которое будет содержать описание связей между выделенными на Шаге 5 фреймами, а также между выделенными на Шаге 5 фреймами и интерфейсами.

Шаг 8. Сформировать МП «requirement_frame_is», содержащее описания фреймов и интерфейсов, которые сформулированы Потребителем и могут быть ре-реализованы Поставщиком.

Шаг 9. Сформировать МП «requirement_attributes_is», содержащее описания атрибутов фреймов и интерфейсов, которые сформулированы Потребителем и могут быть ре-реализованы Поставщиком.

Шаг 10. Сформировать МП «requirement_relations_is», содержащее описания связей, которые сформулированы Потребителем и могут быть ре-реализованы Поставщиком, между фреймами, а также между фреймами и интерфейсами.

Шаг 11. Дополнить данные в МП «requirement_frame_is» данными из МП «requirement_frame_provider», которые отсутствуют в МП «requirement_frame_user».

Шаг 12. Дополнить данные в МП «requirement_attributes_is» данными из МП «requirement_attributes_provider», которые отсутствуют в МП «requirement_attributes_user».

Шаг 13. Дополнить данные в МП «requirement_relations_is» данными из МП «requirement_relations_provider», которые отсутствуют в МП «requirement_relations_user».

Шаг 14. Дополнить данные в МП «requirement_frame_is» данными из

МП «requirement_frame_user», которые отсутствуют в МП «requirement_frame_provider».

Шаг 15. Дополнить данные в МП «requirement_attributes_is» данными из МП «requirement_attributes_user», которые отсутствуют в МП «requirement_attributes_provider».

Шаг 16. Дополнить данные в МП «requirement_relations_is» данными из МП «requirement_relations_user», которые отсутствуют в МП «requirement_relations_provider».

Шаг 17. Сохранить результаты выполнения Шага 16 в в БД ИТ формирования и анализа требований к ИС. Завершить выполнение алгоритма.

Фрагменты команд, создающих МП «requirement_idea», «requirement_frame_user», «requirement_attributes_user» и «requirement_relations_user» в ходе выполнения Шагов 1-4 рассмотренного выше алгоритма, приведен на рис. Д.9-Д.12 Приложения Д. Команды, выполняющие Шаги 5-7 рассмотренного выше алгоритма, аналогичны рассмотренным выше командам, выполняющим Шаги 2-4 данного алгоритма. Фрагменты команд, создающих МП «requirement_frame_is», «requirement_attributes_is» и «requirement_relations_is» в ходе выполнения Шагов 8-10 рассмотренного выше алгоритма, приведены на рис. Д.13-Д.15 Приложения Д. Фрагменты команд, модифицирующих содержимое МП «requirement_frame_is», «requirement_attributes_is» и «requirement_relations_is» в ходе выполнения Шагов 11-13 рассмотренного выше алгоритма, приведены на рис. Д.16-Д.18 Приложения Д. Команды, выполняющие Шаги 14-16 рассмотренного выше алгоритма, во многом аналогичны рассмотренным выше командам, выполняющим Шаги 11-13 данного алгоритма. Для выполнения Шага 17 рассмотренного выше алгоритма результаты запросов, выполняемых в ходе создания МП «requirement_frame_is», «requirement_attributes_is» и «requirement_relations_is», следует сохранить в БД ИТ формирования и анализа требований к ИС (например, с применением стандартной конструкции «INSERT INTO... SELECT...»).

Рассмотренные особенности реализации ИТ формирования и анализа

требований к ИС позволяют значительно повысить обособленность компонента «Model» от других компонентов шаблона проектирования MVC. Это, в свою очередь, позволяет использовать рассмотренные варианты реализации предложенных в работе моделей и методов без значительных изменений как для клиент-серверной, так и для веб-базированной реализации данной ИТ. В частности, становятся равновозможными такие варианты:

а) продажа клиент-серверной версии предлагаемой ИТ конкретной ИТ-компания и эксплуатация в рамках корпоративной сети данной компании;

б) продажа web-базированной версии предлагаемой ИТ конкретной ИТ-компания и эксплуатация в рамках Internet/Intranet сетей компании и ее возможных клиентов;

в) предоставление ряду ИТ-компаний предлагаемой ИТ как самостоятельной ИТ-услуги, эксплуатируемой в рамках Internet/Intranet сетей компаний и ее возможных клиентов, или же в рамках облачных технологий.

6.2 Описание требований и решений, ре-используемых в ходе апробации информационной технологии

Для отражения особенностей практического применения предлагаемой ИТ формирования и анализа требований к ИС целесообразно рассмотреть два следующих основных момента [134]:

а) использование ранее выполненного проекта для выделения представлений требований к ИС и соответствующих этим требованиям проектных решений по ИО и ПО ИС;

б) использование библиотеки ранее реализованных требований к ИС в ходе создания новой ИС.

В качестве ранее выполненного проекта предлагается использовать выполненный в рамках инициативной некоммерческой разработки ИТ-проект со-

здания информационно-аналитической системы (ИАС) «Реестр Лиги украинских клубов интеллектуальных игр (ЛУК)». Данная система создавалась по соглашению с Всеукраинской общественной организацией (ВОО) «ЛУК». В качестве основной методологии разработки ИС в ходе создания ИАС «Реестр ЛУК» была использована методология, изложенная в ГОСТах группы 34 «Информационные технологии». Данная методология основана на каскадной модели ЖЦ создания ИС. Описание особенностей Про ВОО «ЛУК», основных потребностей Потребителя, определяющих требования к ИС, а также описание ФС ИАС и основных проектных решений приведены в [134]. Исходная схема данных ИАС «Реестр ЛУК» приведена на рис. Е.1 Приложения Е.

С целью преобразования проектных решений ИАС «Реестр ЛУК» к виду, пригодному для выделения ре-используемых представлений требований к ИС и соответствующих этим требованиям проектных решений по ИО и ПО ИС, был проведен рефакторинг данной системы. В ходе рефакторинга была использована взаимосвязь языка программирования Java (jre 1.8.25) и СУБД Oracle XE 11g Release 11.2.0.2.0, а также web-сервер Apache Tomcat 8.0.15. В качестве сред разработки использованы IDE Eclipse Luna (Release 4.4) и Quest Software SQL Navigator for Oracle v.6.7.

Основными целями данного рефакторинга являются:

- модификация ФС системы, вызванная уточнением требований в процессе опытной эксплуатации базовой версии ИАС;
- усовершенствование решений по ПО и ИО ИАС, внедрение новых архитектурных стилей, определяемых возможностями отображения описания архитектуры ИАС в описания решений по ИО и ПО ИАС;
- дополнительное включение в состав ИАС «Реестр ЛУК» стандартных функций CMS-системы, что позволяет отказаться от использования и интеграции с эксплуатируемой ранее в ВОО ЛУК CMS-системы Word Press.

Следует отметить, что расширение существующей функциональности системы за счет включения ранее не реализованных функций в ходе данного рефакторинга не может и не должно рассматриваться в качестве одной из целей рефакторинга.

В качестве результата рефакторинга предлагается рассматривать:

- а) адаптацию существующей ИАС к уточнённым требованиям и возникшим изменениям в ПрО;
- б) сформированную библиотеку ранее реализованных требований и компонентов, подготовленную для использования в других IT-проектах.

В ходе рефакторинга ИАС «Реестр ЛУК» на основе существующих проектных решений было выделено общесистемное AD ИАС, приведенное на рис. Е.2 Приложения Е. Для этого применялся следующий метод [214].

Шаг 1. С применением процессного подхода осуществляется анализ основных БП ПрО. Результатом данного анализа является набор IDEF0- или IDEF3-диаграмм, отражающих автоматизируемые БП с различных точек зрения их участников.

Шаг 2. Выделение БП, подлежащих автоматизации, на основе анализа описаний БП, созданных с различных точек зрения. Критерием выделения БП является получение в результате выполнения процесса данных, которые востребованы теми или иными их УАП и должны быть сохранены в БД или получены из неё.

Шаг 3. В ходе объектно-ориентированного анализа ПрО осуществляется выделение терминов, которые максимально полно и комплексно характеризуют данную ПрО и отражают её существенные особенности. Результатом данного анализа является перечень терминов, их определений и синонимов, приведенный, например, в табличном виде.

Шаг 4. В результате комбинированного применения процессного и объектно-ориентированного подходов из терминов, описывающих рассматриваемые процессы ПрО, формируются соответствующие автоматизируемым операциям БП деревья онтологий. Такие деревья онтологий состоят из групп терминов, которые непосредственно задействованы в конкретной операции.

Результатом применения данного метода являются деревья онтологий, которые строятся для каждой операции, выделенной на IDEF0- или IDEF3-диаграммах. Деревья онтологий представляются в виде UML-диаграмм, где все термины являются объектами, имеющими как иерархические, так и горизонтальные связи. Поэтому каждая такая UML-диаграмма операции БП име-

ет вид так называемого леса онтологий [215]. Однако, как было отмечено в подразд. 5.1, представление AD ИАС в виде низкорослого леса концептов ПрО, визуализируемых с помощью диаграмм классов UML, является очень громоздким. Поэтому в дальнейшем для описания архитектуры целесообразным следует считать совместное использование ДК и ER-диаграмм или же модифицированных ER-диаграмм. Кроме того, при формировании AD ИАС «Реестр ЛУК» следует учитывать тот факт, что все таблицы БД ИАС, показанные на рис. Е.1 Приложения Е, относятся к одной схеме и не сгруппированы в витрины данных по признаку совместного использования.

В ходе уточнения требований к функциям ИАС «Реестр ЛУК» были выделены описания требований с точек зрения участников и организаторов мероприятий. Описание словаря терминов ПрО, используемых в данных описаниях приведен в табл. Е.1 Приложения Е. Примеры представлений этих требований на уровне знаний приведены на рис. Е.3-Е.12 Приложения Е.

По результатам уточнения ФТ к ИАС «Реестр ЛУК» был синтезирован базовый вариант AD ИАС, представленный в виде UML-диаграммы на рисунке 6.4. С целью корректного отображения связей между узлами данный рисунок разбит на несколько частей с дублированием граничных на рисунке узлов концептов.

На этапе формирования общесистемного AD ИАС важной процедурой является выделение из существующих элементов обобщённых узлов онтологии (концептов), которые могут быть ре-использованы при проектировании других ИС. Далее выделяются специализированные концепты (новые или наследуемые от обобщённых концептов), уточняющие обобщённые концепты для ПрО создаваемой ИС. Также следует отметить приведенные на рис. 6.4 результаты выделения третьего уровня элементов онтологии ПрО ИС, обеспечивающих реализацию нефункциональных требований к ИАС (администрирование учетных записей и ролей пользователей системы, разграничение прав доступа пользователей к функциям ИС, аудит действий и т.д.) [216]. Этот уровень не зависит от ПрО и вводится в AD для унификации нотаций визуального моделирования при разработке ИАС.

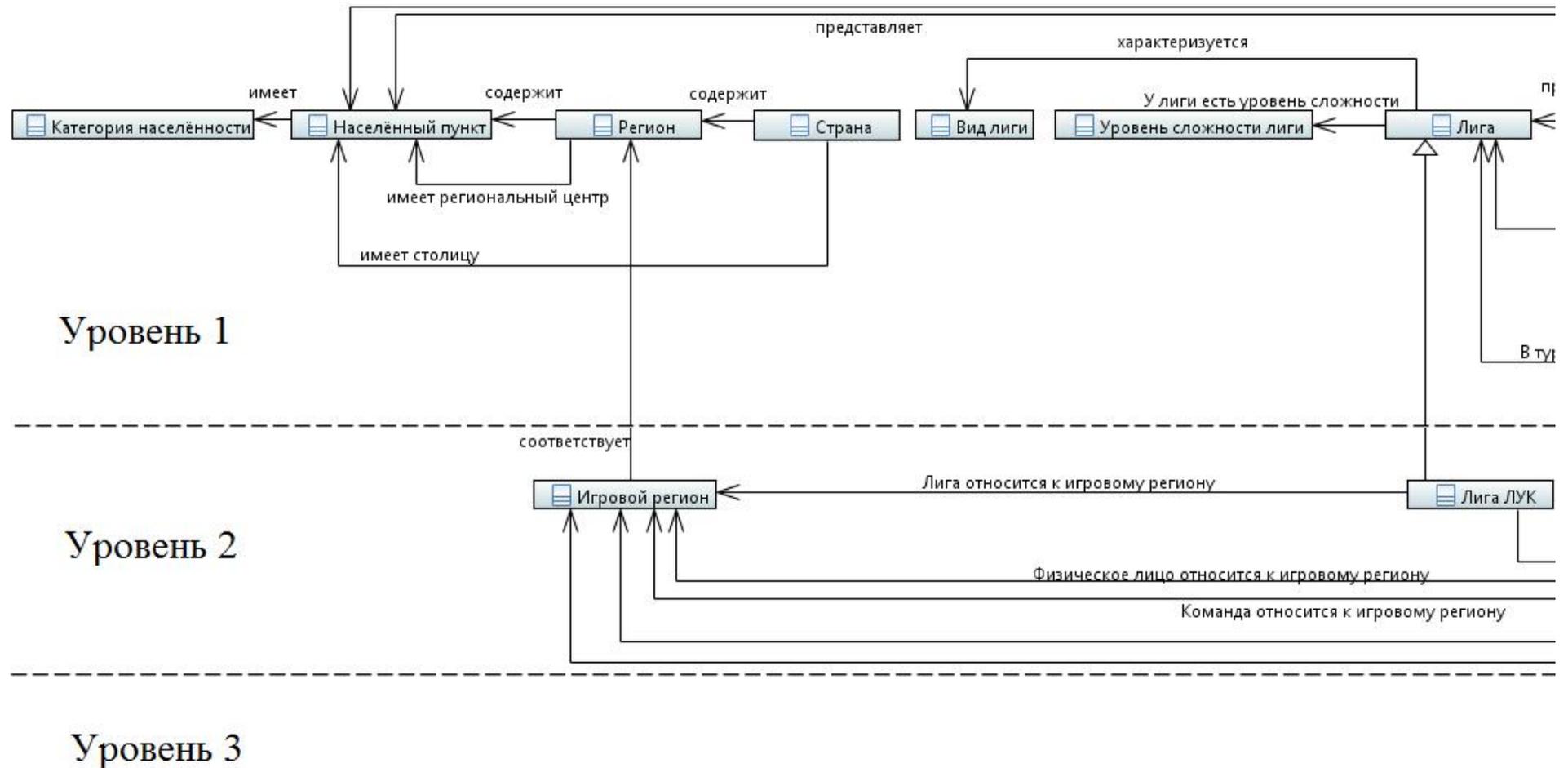


Рисунок 6.4 – Скорректированное общесистемное описание архитектуры информационно-аналитической системы «Реестр Лиги украинских клубов интеллектуальных игр», декомпозированное по уровням абстракции, лист 1

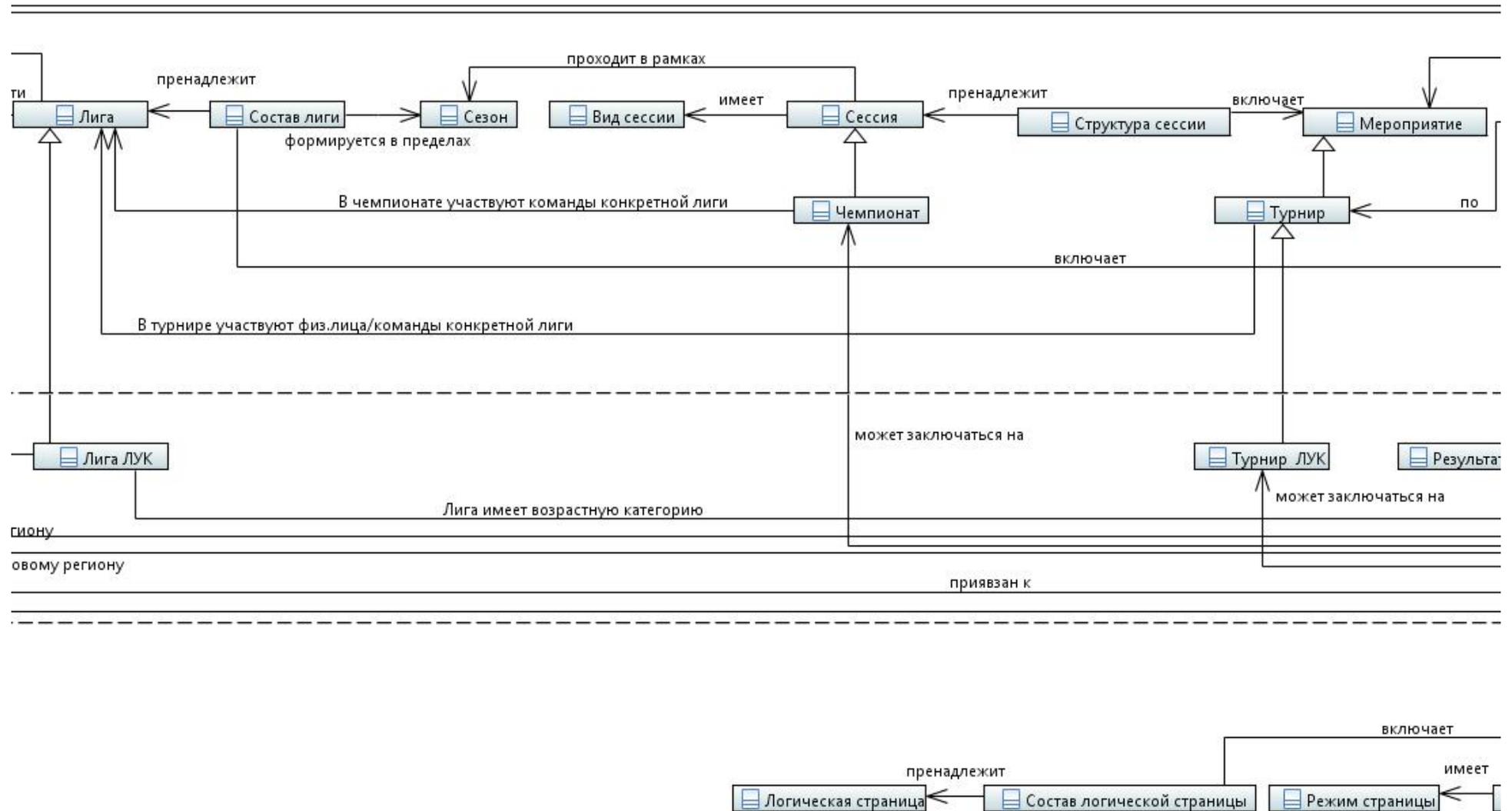


Рисунок 6.4 – Скорректированное общесистемное описание архитектуры информационно-аналитической системы «Реестр Лиги украинских клубов интеллектуальных игр», декомпозированное по уровням абстракции, лист 2

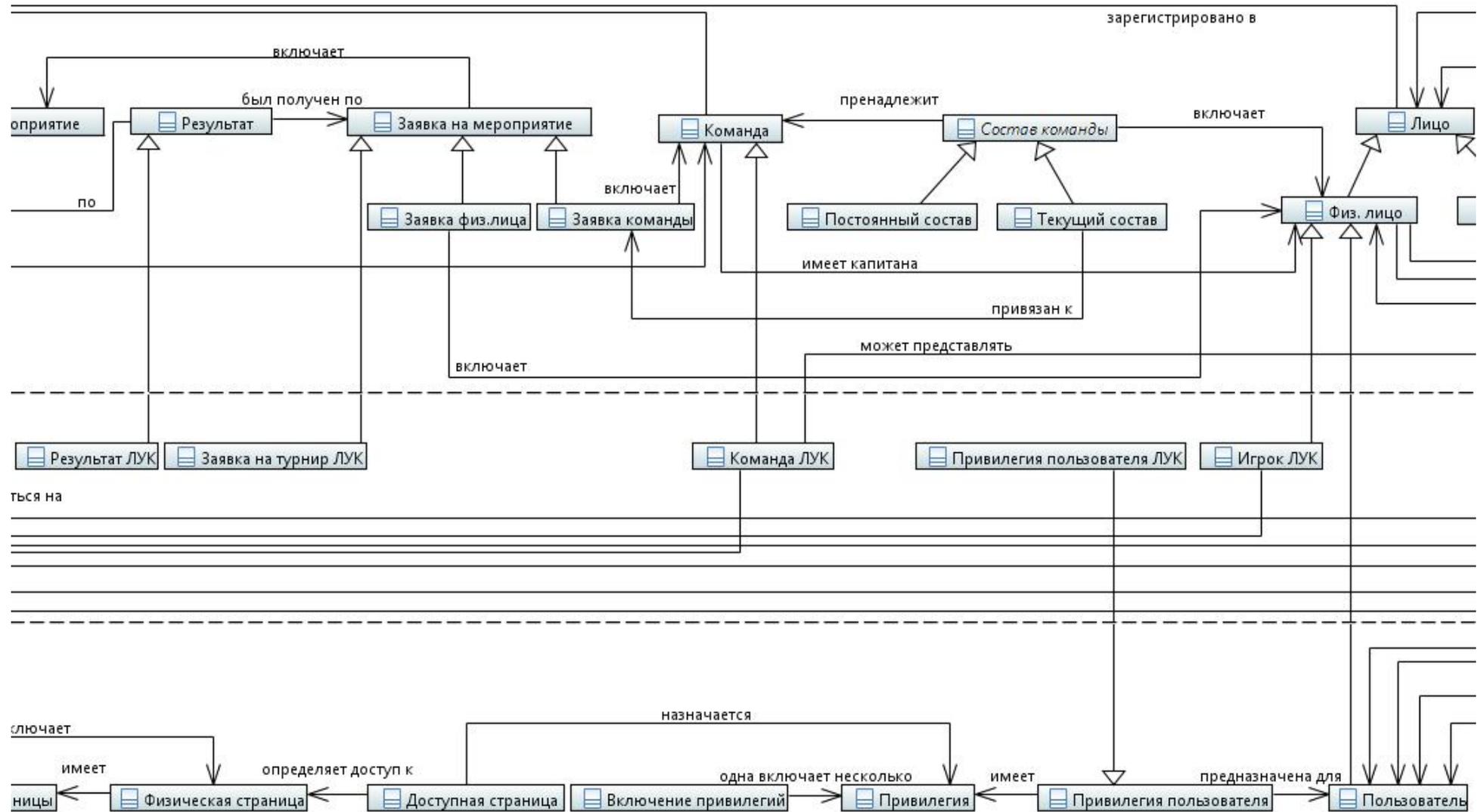


Рисунок 6.4 – Скорректированное общесистемное описание архитектуры информационно-аналитической системы «Реестр Лиги украинских клубов интеллектуальных игр», декомпозированное по уровням абстракции, лист 3

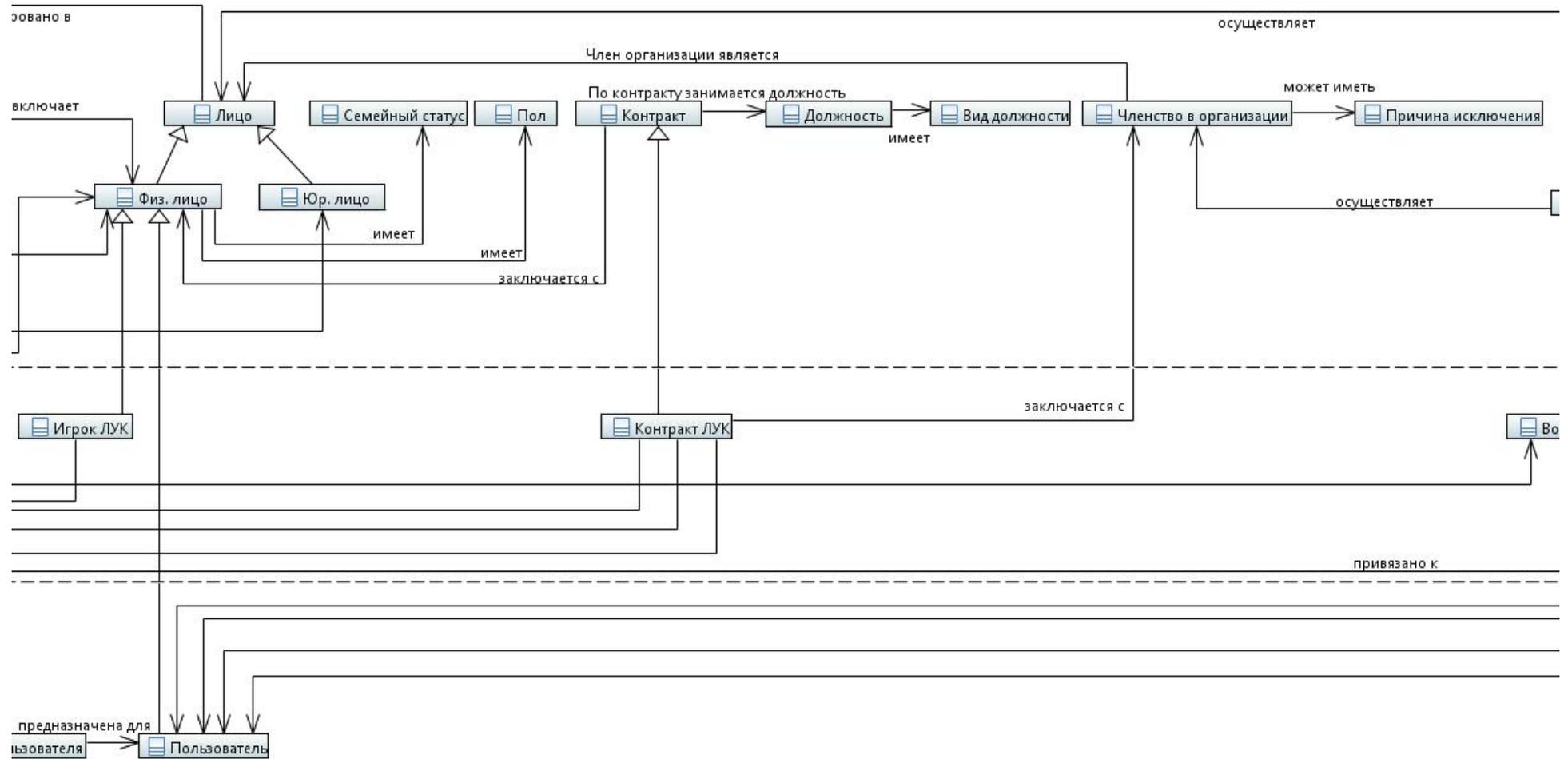


Рисунок 6.4 – Скорректированное общесистемное описание архитектуры информационно-аналитической системы «Реестр Лиги украинских клубов интеллектуальных игр», декомпозированное по уровням абстракции, лист 4

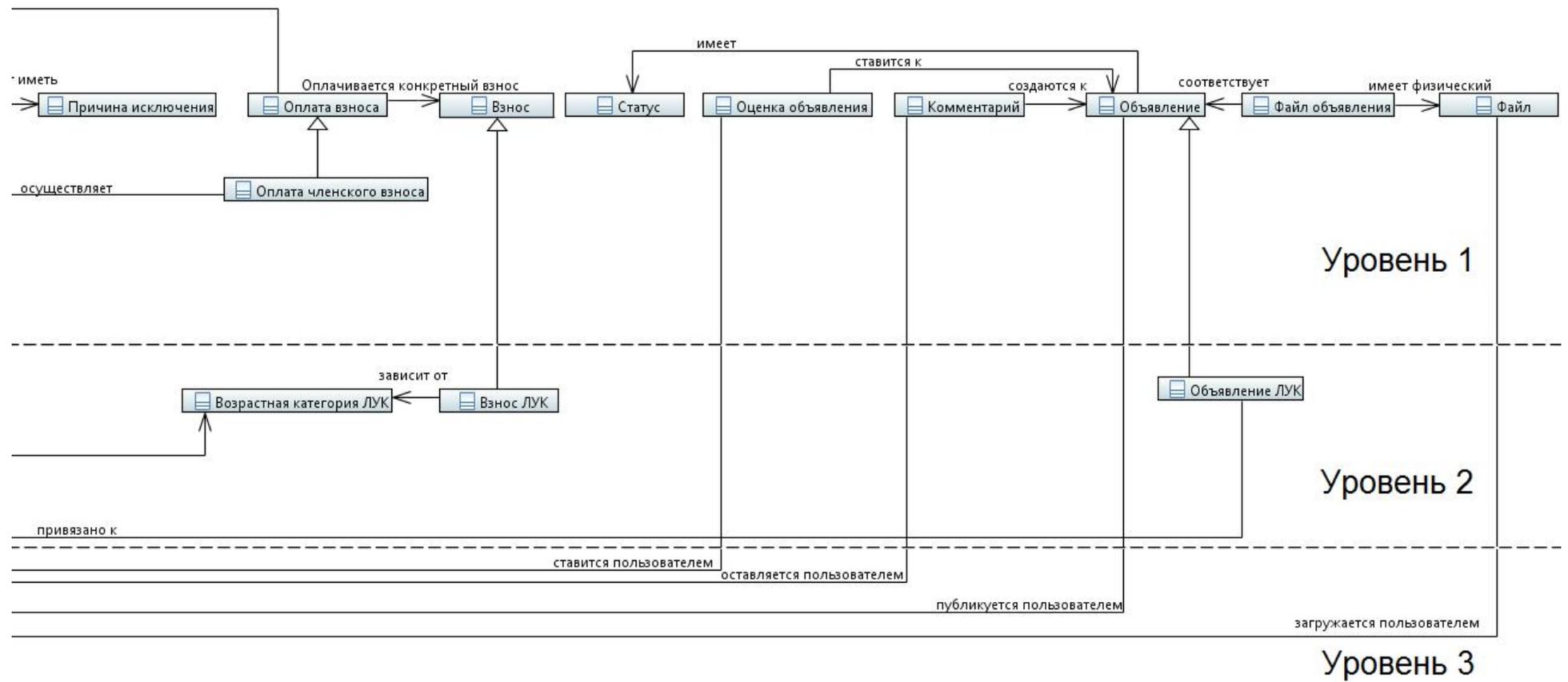


Рисунок 6.4 – Скорректированное общесистемное описание архитектуры информационно-аналитической системы «Реестр Лиги украинских клубов интеллектуальных игр», декомпозированное по уровням абстракции, лист 5

Скорректированное общесистемное АД ИАС «Реестр ЛУК» было использовано для синтеза схемы данных БД ИАС как основного элемента ИО создаваемой системы. При этом основные концепты АД стали родительскими таблицами схемы данных с соответствующими атрибутами. Выделенные на втором уровне (см. рис. 6.4) концепты, характерные для ПрО ИАС «Реестр ЛУК», стали в синтезированной схеме данных наследуемыми таблицами, зависимыми от обобщающих сущностей. В целом необходимо отметить, что таблицы синтезированной схемы данных имеют однозначное соответствие (по названиям, структуре и наборам связей) элементам АД ИАС, в котором предварительно были выделены уровни служебных, абстрактных и специфичных для ПрО элементов.

Результат синтеза схемы данных приведен на рис. Е.13 Приложения Е. При этом, как упоминалось в подразд. 4.5, синтезированная схема данных представляет собой набор витрин данных типа «звезда» или «снежинка», объединенных в отдельные схемы данных с названиями, соответствующими их уровням в описании архитектуры ИАС («АРС_» – для служебного уровня, «АРМ_» – для абстрактного уровня и «АРР_» – для уровня конкретной ПрО).

При разработке ПО ИАС «Реестр ЛУК» были использованы паттерны проектирования Data Access Objects (DAO) и Model-View-Controller (MVC).

В результате применения DAO были созданы следующие пакеты классов:

- классы «Mapper», которые выполняют функции сохранения, загрузки и изменения объектов в БД;
- классы «Service», которые предоставляют доступ к классам «Mapper»;
- классы «ServiceFactory», которые создают сервисы на этапе загрузки web-сервера, с применением паттерна программирования «factory» (фабрика).

С применением MVC в качестве модели (model) выступают бизнес-классы (классы «PageLogical», «PagePhysical», «PersonPhysical», «Team», «Player», «TeamLuk» и т.д.). В качестве контроллера (controller) используются одноименные классы Controller, которые занимаются реализацией бизнес-логики и перенаправлением пользователя между web-страницами системы в зависимости от его действий. Представление (view) – это jsp-страницы, кото-

рые представляют собой интерфейс пользователя [217].

Кроме того, разработанное ПО структурировано с учётом применяемой ИТ комплексного рефакторинга обеспечение ИС. Исходя из этого выделяются следующие наборы классов в соответствии с областями их применения:

- классы, реализующие нефункциональные требования (пакет «common»);
- максимально абстрагированные от конкретных ПрО наборы классов, реализующие ФТ (пакет «module»);
- классы, специфичные для ПрО ЛУК (пакет «sluk»).

Примеры описания архитектуры ПО ИАС «Реестр ЛУК» приведены в виде диаграмм классов на рис. Е.14–Е.18 Приложения Е.

Полученные проектные решения позволяют утверждать, что скорректированное АД ИАС «Реестр ЛУК» полностью воплощено в ИТ-сервисах, реализующих функции данной системы. Поэтому все представления ФТ к ИАС «Реестр ЛУК» на уровне знаний включаются в библиотеку ранее реализованных требований автоматически. Таким образом, будет реализована возможность использования уже готовых решений из данной библиотеки при проектировании новых ИС в будущем. При этом в эту библиотеку включаются только концепты, которые классифицируются как абстрагированные от специфики конкретной ПрО (уровень 1) и служебные (уровень 3).

Описание ре-используемых элементов ИС, сформированное в результате рефакторинга ИАС «Реестр ЛУК», представлено в виде набора витрин данных (схемы «АРС_» и «АРМ_») и показано на рис. 6.5.

Полученные описания реализованных ФТ к ИАС «Реестр ЛУК» и соответствующие проектные решения могут быть использованы в ходе формирования и анализа требований к другим ИС, в рамках которых возможна реализация аналогичных ИТ-услуг.

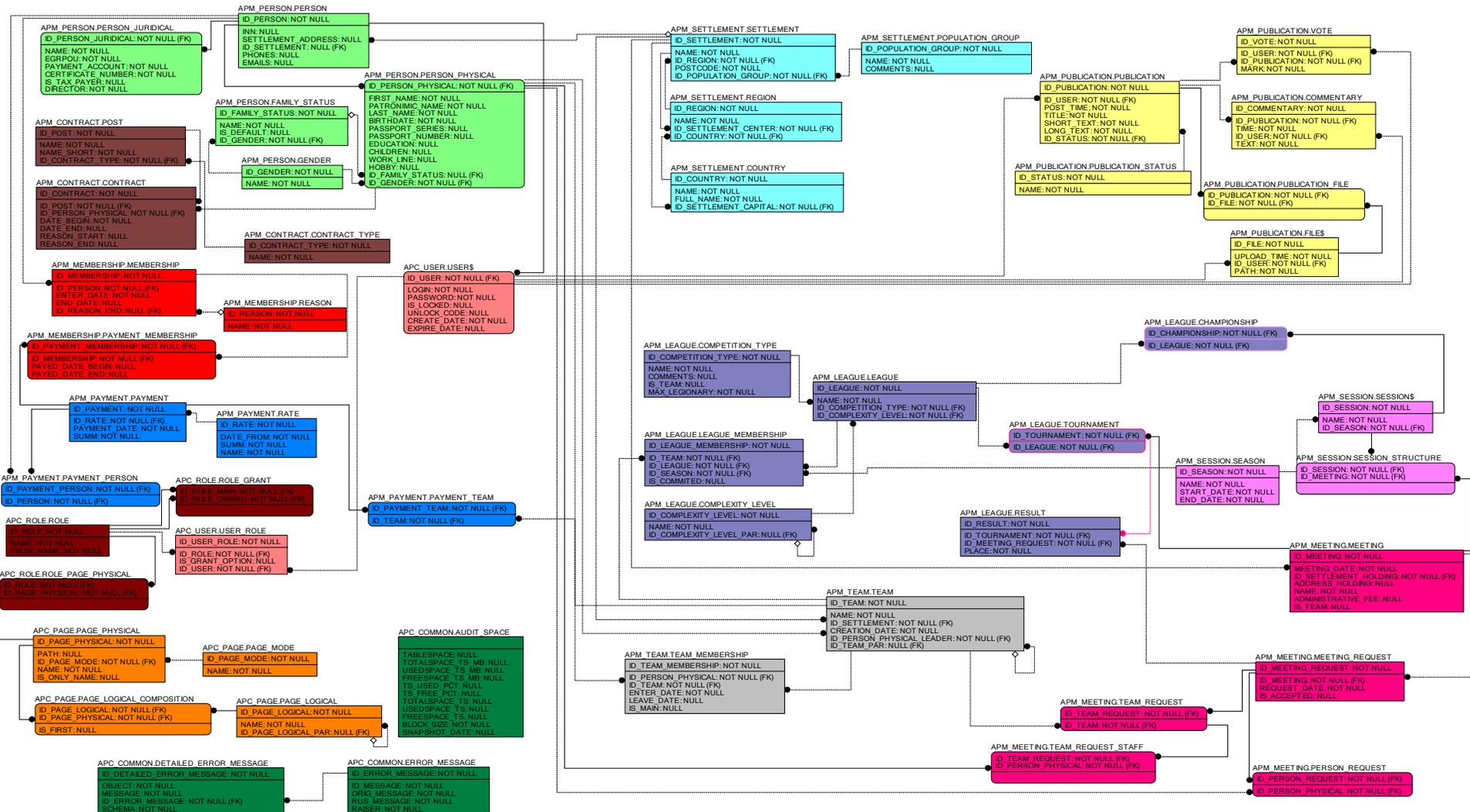


Рисунок 6.5 – Описание ре-используемых элементов архитектуры информационной системы, сформированное в результате рефакторинга информационно-аналитической системы «Реестр Лиги украинских клубов интеллектуальных игр»

6.3 Апробация информационной технологии на примере функционального модуля безопасности труда комплексной системы управления ПАО «Харьковгоргаз»

Для выявления особенностей применения ИТ формирования и анализа требований к ИС в ходе создания новой ИС рассмотрим процессы формирования ОРА функционального модуля (ФМ) безопасности труда (БТ). ОА для ФМ БТ является ПАО «Харьковгоргаз».

ПАО «Харьковгаз» - одно из крупнейших газоснабжающих предприятий Украины. Штат работников предприятия составляет более 3000 человек. Основным направлением деятельности ПАО «Харьковгаз» является поставка природного газа населению, учреждениям, организациям бюджетной сферы и промышленным потребителям по регулируемому тарифу, определенному Национальной комиссией, которая осуществляет государственное регулирование в сфере энергетики. Ежегодно предприятие транспортирует более 1,0 млрд. m^3 природного газа. В своей деятельности ПАО «Харьковгаз» целенаправленно применяет новейшие компьютерные технологии, позволившие автоматизировать все основные производственные и учетные процессы на предприятии.

Одними из приоритетных программ и проектов деятельности ПАО «Харьковгаз» в настоящее время являются программа «Здоровье и безопасность на работе и дома», и проект «Комплексные мероприятия по охране труда». Их общей целью является повышение уровня производительности труда путем выполнения целенаправленных действий по обеспечению БТ на предприятии и создания условий для оздоровления и отдыха работников [218].

Для выполнения этих программы и проекта в ПАО «Харьковгаз» предусматривается, в частности, создание специализированного ФМ БТ. Среди целей данного ФМ можно выделить следующие:

а) обеспечение возможности учета вредных производственных факторов (ВПФ), которые действуют на сотрудников ПАО «Харьковгаз» в ходе выполнения ими своей профессиональной деятельности;

б) прогноз воздействия комплекса ВПФ на организмы сотрудников предприятия, по результатам которого может быть принято решение о временном прекращении производственной деятельности сотрудника с целью его оздоровления.

Однако в настоящее время ни одна из существующих ИС управления БТ не позволяет выполнять функции подобного прогноза согласно стандарту OHSAS 18001:2007, Требованиям ILO/OSH 2001 и ДСТУ OHSAS 18001:2010 «Система управління гігієною та безпекою праці». Большинство из них представляют собой разновидности специализированных систем электронного документооборота, при этом стандартные функции управления таким документооборотом в рассмотренных системах представлены минимально [219-221]. Главной причиной данного недостатка следует признать изначальную неполноту моделей ФЗ анализа и прогноза, на основе которых могут быть сформированы управляющие воздействия по обеспечению БТ на предприятии.

Поэтому в ходе создания ФМ БТ представителями ПАО «Харьковгаз», выступающего в роли Потребителя, были заявлены следующие потребности [219]:

а) «реализовать функцию учета сведений о предприятии и процессах (работах), которые выполняются на данном предприятии и в ходе выполнения которых возможно воздействие на сотрудников предприятия комплекса ВПФ» (первое функциональное требование);

б) «реализовать функцию учета кадровых данных (данных о сотрудниках предприятия), минимально необходимых для принятия управленческих решений по обеспечению БТ на предприятии» (второе функциональное требование);

в) «реализовать функцию формирования и ведения справочника ВПФ, которые действуют или могут действовать в ходе выполнения отдельных процессов или работ предприятия» (третье функциональное требование);

г) «реализовать функцию учета результатов наблюдений действия каждого из ВПФ в ходе выполнения процессов или отдельных работ предприятия» (четвертое функциональное требование);

д) «реализовать функцию прогноза воздействия комплекса ВПФ на организм сотрудника предприятия, выполняющего отдельный процесс или ра-

боту на предприятии» (пятое функциональное требование).

Из этих потребностей следует, что большинство функций ИС управления БТ на предприятии должны выполнять стандартные операции учета данных. Однако для реализации функции прогноза Разработчику необходимо знать математическую модель, которую предполагается использовать для прогноза, и методы решения данной задачи. В ходе разработки ФМ БТ с этой целью были использованы соответствующие модели, а также метод определения состояния организма сотрудника предприятия, предложенные в [222].

Для преобразования приведенных выше потребностей в СФТ к ФМ БТ была выполнена публикация соответствующих требований в виде DFD. Визуальное представление этих DFD приведено на рис. 6.4-6.8.

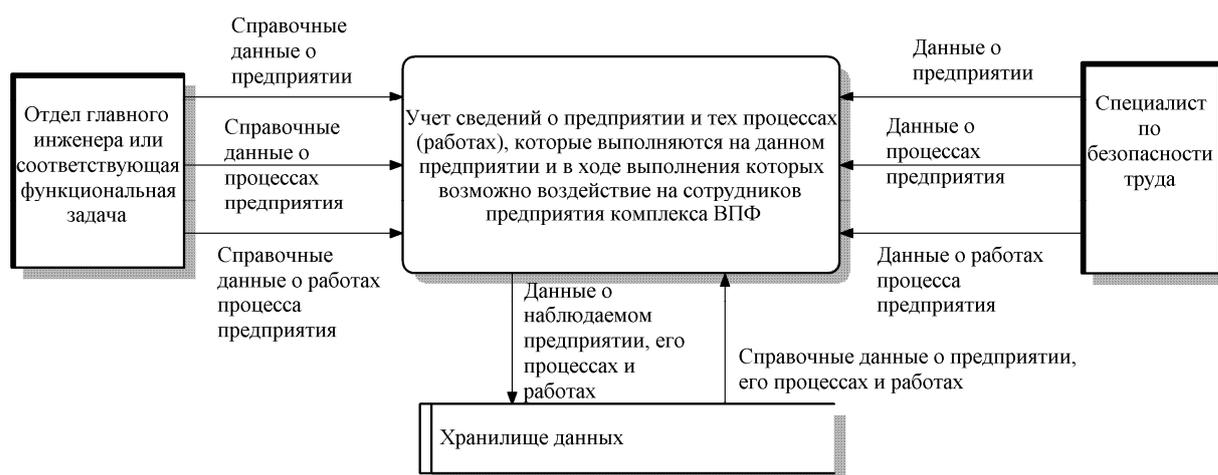


Рисунок 6.4 – Диаграмма потоков данных публикации первого функционального требования к функциональному модулю безопасности труда на предприятии ПАО «Харьковгоргаз»

Публикации этих требований послужили исходными данными для применения метода формирования представлений ФТ на уровне информации. Результаты применения данного метода – представления СФТ к ФМ БТ на уровне информации $(I_1^f - I_5^f)$ – являются исходной информацией для реализации методов формирования представления ФТ к ФМ на уровне знаний.

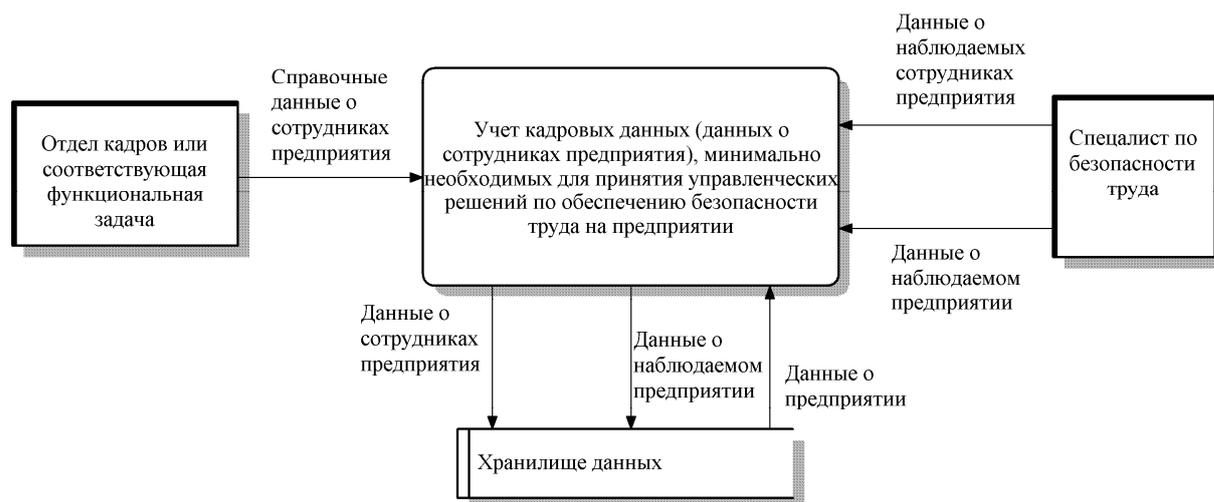


Рисунок 6.5 – Диаграмма потоков данных публикации второго функционального требования к функциональному модулю безопасности труда на предприятии ПАО «Харьковгоргаз»



Рисунок 6.6 – Диаграмма потоков данных публикации третьего функционального требования к функциональному модулю безопасности труда на предприятии ПАО «Харьковгоргаз»

При этом, поскольку УАП является только специалист по охране труда ПАО «Харьковгаз», результаты применения метода формирования представления СТ на уровне знаний отдельного УАП и результаты применения метода формирования представления СТ на уровне знаний Потребителя будут совпадать. Исключением будет являться только пятое СТ к ФМ БТ, в котором вво-

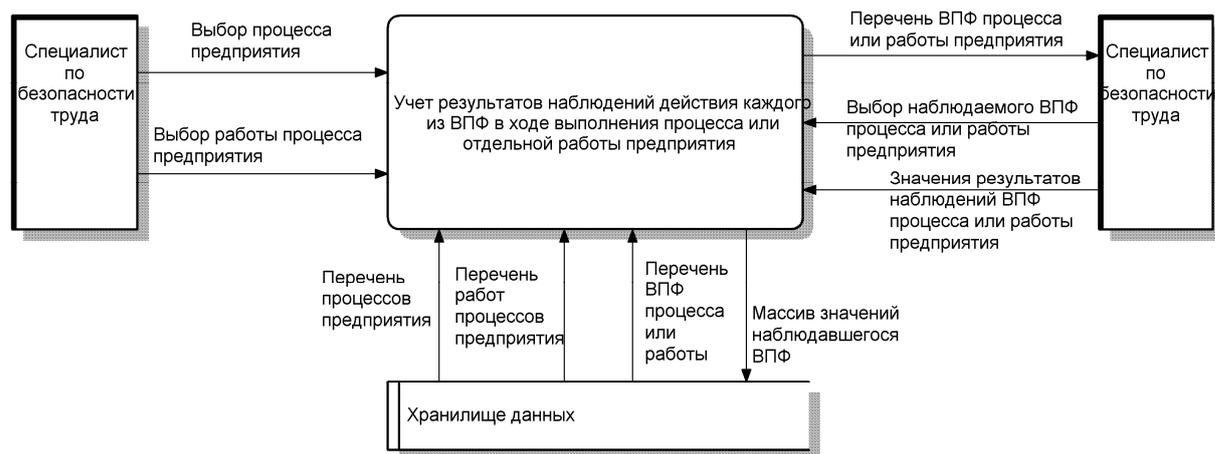


Рисунок 6.7 – Диаграмма потоков данных публикации четвертого функционального требования к функциональному модулю безопасности труда на предприятии ПАО «Харьковгоргаз»



Рисунок 6.8 – Диаграмма потоков данных публикации пятого функционального требования к функциональному модулю безопасности труда на предприятии ПАО «Харьковгоргаз»

дится дополнительный пользователь ФМ БТ. На ОА таким пользователем может выступать представитель руководства ПАО «Харьковгоргаз», ответственный за принятие управленческого решения по обеспечению БТ сотрудников предприятия.

Результатом применения метода формирования представления ФТ на уровне знаний отдельного УАП являются представления СТ к ФМ БТ $K_1^f(U_{st_1}) - K_5^f(U_{st_1})$ на уровне знаний специалиста по БТ (U_{st_1}), а также представление СТ к ФМ БТ $K_5^f(U_{st_2})$ на уровне знаний пользователя ФМ (U_{st_2}). Визуальное описание данных представлений показано на рис. 6.9-6.14.

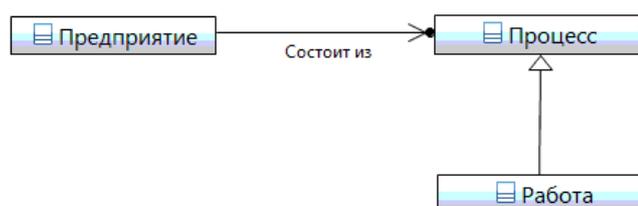


Рисунок 6.9 – Диаграмма классов, описывающая представление первого функционального требования на уровне знаний специалиста по безопасности труда

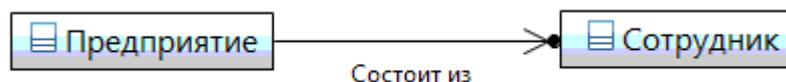


Рисунок 6.10 – Диаграмма классов, описывающая представление второго функционального требования на уровне знаний специалиста по безопасности труда

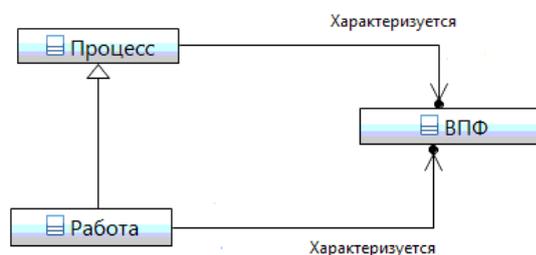


Рисунок 6.11 – Диаграмма классов, описывающая представление третьего функционального требования на уровне знаний специалиста по безопасности труда

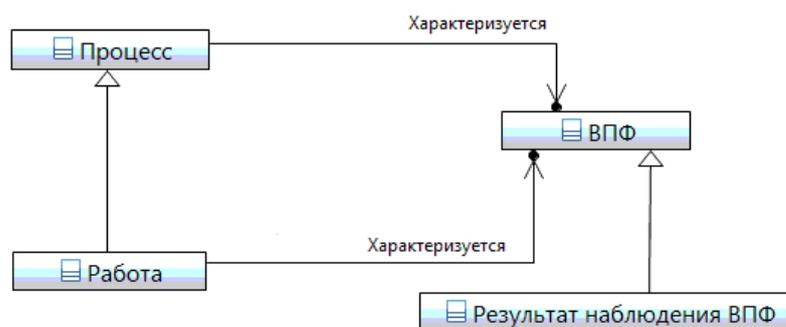


Рисунок 6.12 – Диаграмма классов, описывающая представление четвертого функционального требования на уровне знаний специалиста по безопасности труда

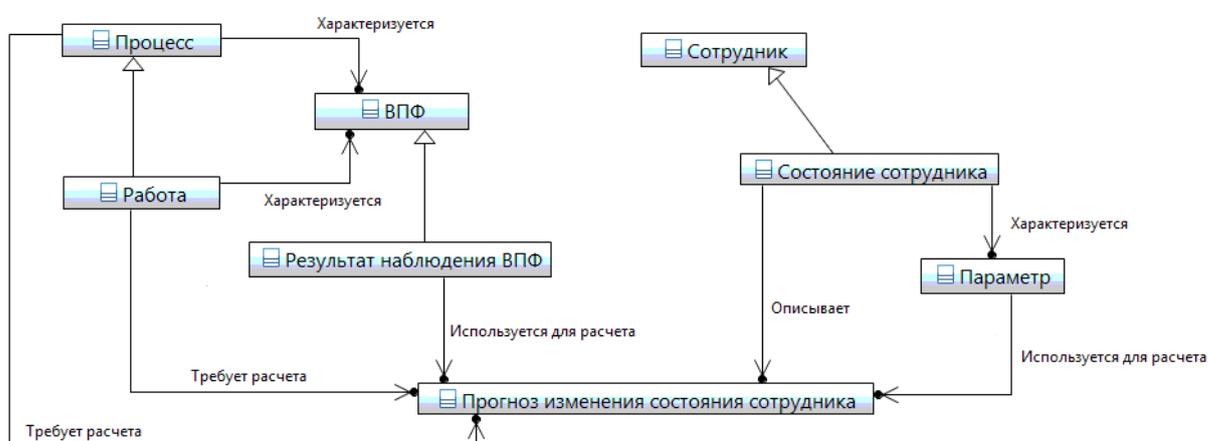


Рисунок 6.13 – Диаграмма классов, описывающая представление пятого функционального требования на уровне знаний специалиста по безопасности труда

В ходе формирования представлений СТ к ФМ БТ на уровне знаний Потребителя $K_1^{fU} - K_5^{fU}$ представления $K_1^{fU} - K_4^{fU}$ будут иметь вид, совпадающий с видом представлений $K_1^f(U_{st_1}) - K_4^f(U_{st_1})$ (см. рис. 6.9-6.12). Это связано с тем, что требования с первого по четвертое были выдвинуты одним УАП – специалистом по БТ. Представление пятого требования на уровне знаний Потребителя K_5^{fU} после слияния представлений $K_5^f(U_{st_1})$ и $K_5^f(U_{st_2})$ будет иметь вид, показанный на рис. 6.15.

Применение метода анализа отдельных фреймов для представлений $K_1^{fU} - K_5^{fU}$ на непротиворечивость показывает отсутствие противоречий

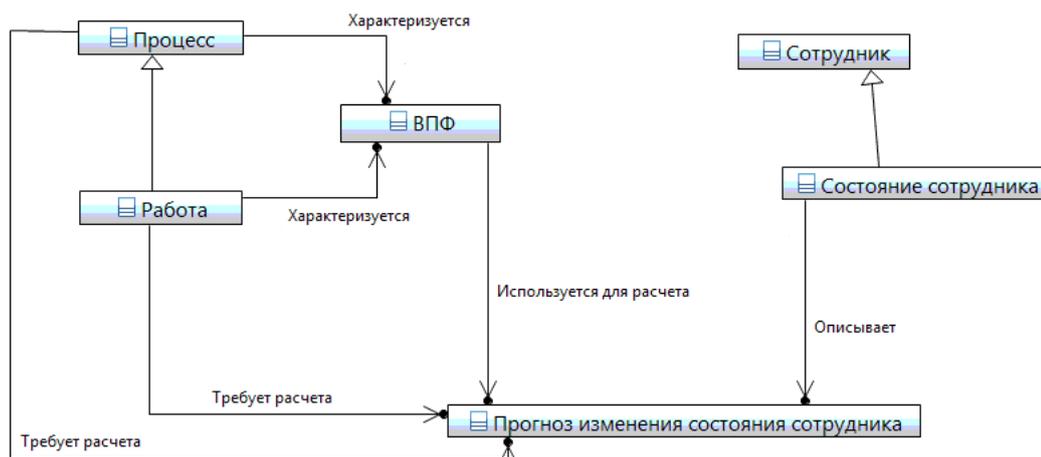


Рисунок 6.14 – Диаграмма классов, описывающая представление пятого функционального требования на уровне знаний пользователя информационно-аналитической системы (представителя руководства)

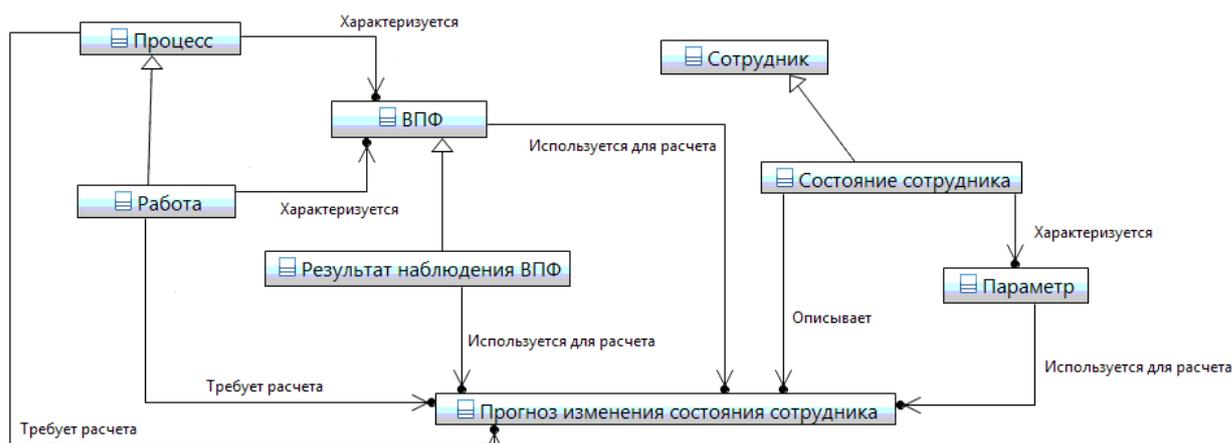


Рисунок 6.15 – Диаграмма классов, описывающая представление пятого функционального требования на уровне знаний Потребителя

между фреймами данных представлений, поскольку:

а) фреймы с одинаковыми наименованиями используют одни и те же идентификаторы для обозначения своего присутствия в представлениях различных требований на уровне знаний, что приводит к невыполнению условия (4.66);

б) для сочетаний фреймов «Сотрудник» и «Состояние сотрудника», «Сотрудник» и «Прогноз изменения состояния сотрудника», «Состояние сотрудника» и «Прогноз изменения состояния сотрудника», «ВПФ» и «Результат наблюдения ВПФ» выполняется первая, но не выполняется вторая часть условия (4.67).

Применение метода анализа отдельных связей представлений K_1^{fU} - K_5^{fU} на непротиворечивость также показывает отсутствие противоречий, поскольку все описания связей с одинаковыми наименованиями (например, связей «Характеризуется» между парой фреймов «Процесс» и «ВПФ», а также между парой фреймов «Работа» и «ВПФ») не совпадают друг с другом.

Результаты применения методов анализа представлений ФТ K_1^{fU} - K_5^{fU} на непротиворечивость позволяют сделать вывод о том, что представления требований на уровне знаний с точки зрения Потребителя непротиворечивы и, следовательно, позволяют сформировать единое информационное представление управляемого объекта.

Ход применения метода выявления нелогичных ФТ к ФМ БТ для представлений K_1^{fU} - K_5^{fU} показан в табл. 6.1.

По результатам применения метода выявления нелогичных ФТ можно сделать следующие выводы:

а) третье ФТ к ФМ БТ нелогично на 50%, поскольку для описания термина ПрО «Вредный производственный фактор» использует упрощенную ОТ «ВПФ» вместо детализированной ОТ «ВПФ + Результат наблюдения ВПФ», используемой в четвертом и пятом ФТ;

б) пятое ФТ к ФМ БТ нелогично на 60%, поскольку использует характерные только для данного требования ОТ «Сотрудник + Состояние сотрудника», «Параметр» и «Прогноз изменения состояния сотрудника».

Выявленные нелогичности признаны такими, которые невозможно устранить без серьезного изменения требований к ФМ БТ. Поэтому принято решение продолжить разработку первой очереди ФМ БТ.

В ходе применения метода формирования представления СТ на уровне знаний Поставщика с использованием элементов библиотеки ранее реализованных требований, показанных на рис. 6.5, представителем Поставщика были приняты и согласованы с представителем Потребителя следующие решения:

Таблица 6.1 – Результаты применения метода выявления нелогичных функциональных требований

для представлений K_1^{fU} - K_5^{fU}

Номер анализируемого требования	Номер итерации метода	Количество онтологических точек требования	Количество нелогичных онтологических точек требования	Номер сравниваемого требования	Количество совпадающих онтологических точек	Значение I_{gr} анализируемого требования, %
1	1	2	2	2	1	50%
1	2	2	1	3	1	– (требование логично)
2	1	2	2	1	1	50%
2	2	2	1	3	0	50%
2	3	2	1	4	0	50%
2	4	2	1	5	1	– (требование логично)
3	1	2	2	1	1	50%
3	2	2	1	2	0	50%
3	3	2	1	4	0	50%
3	4	2	1	5	0	50%
4	1	2	2	1	1	50%
4	2	2	1	2	0	50%
4	3	2	1	3	0	50%
4	4	2	1	5	1	– (требование логично)
5	1	5	5	1	1	80%
5	2	5	4	2	0	80%
5	3	5	4	3	0	80%
5	4	5	4	4	1	60%

а) термин «предприятие» следует рассматривать как термин «юридическое лицо» (для варианта внедрения и эксплуатации ФМ БТ, при котором данная система может использоваться для управления БТ нескольких предприятий одновременно [6.8]);

б) термин «сотрудник» определяет ситуацию, в которой физическое лицо определенного пола занимает некую должность из списка должностей.

Эти решения позволяют описать представления СФТ к ФМ БТ на уровне знаний Поставщика K_1^{fPr} - K_5^{fPr} следующим образом:

а) представления первого, второго и пятого ФТ K_1^{fPr} , K_2^{fPr} и K_5^{fPr} примут вид, показанный на рис. 6.16-6.17 соответственно;

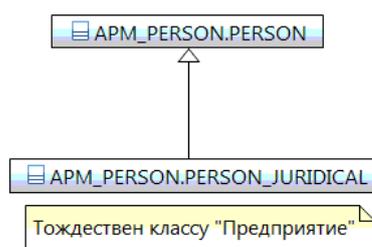


Рисунок 6.16 – Диаграмма классов, описывающая представление первого функционального требования на уровне знаний Поставщика

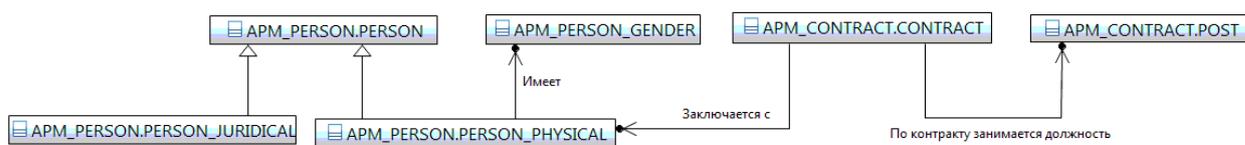


Рисунок 6.17 – Диаграмма классов, описывающая представления второго и пятого функциональных требований на уровне знаний Поставщика

б) представления третьего и четвертого ФТ не могут быть сформированы, поскольку данные требования не могут быть описаны элементами библиотеки ранее реализованных требований.

Сформированные представления K_1^{fPr} , K_2^{fPr} , K_5^{fPr} и K_1^{fU} - K_5^{fU} явля-

ются исходной информацией для применения метода формирования общесистемного представления СТ на уровне знаний. Результатом применения данного метода являются общесистемные представления ФТ к ФМ БТ K_1^{fIS} - K_5^{fIS} , имеющие вид, показанный на рис. 6.18-6.22.

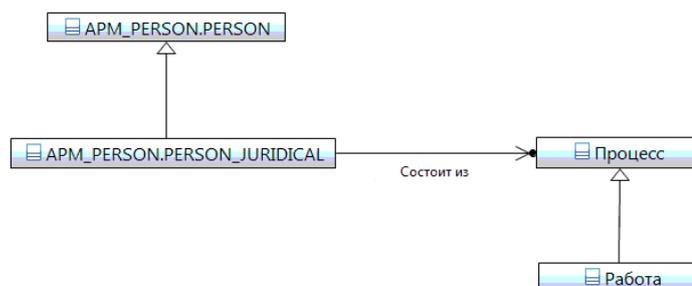


Рисунок 6.18 – Диаграмма классов, описывающая общесистемное представление первого функционального требования на уровне знаний

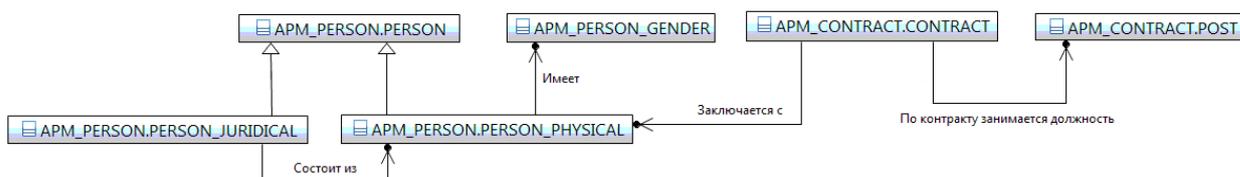


Рисунок 6.19 – Диаграмма классов, описывающая общесистемное представление второго функционального требования на уровне знаний



Рисунок 6.20 – Диаграмма классов, описывающая общесистемное представление третьего функционального требования на уровне знаний

Применение методов анализа отдельных фреймов и связей для представлений K_1^{fIS} - K_5^{fIS} на непротиворечивость показывает отсутствие проти-

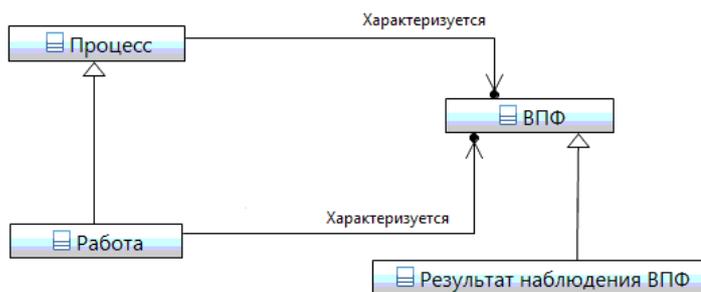


Рисунок 6.21 – Диаграмма классов, описывающая общесистемное представление четвертого функционального требования на уровне знаний

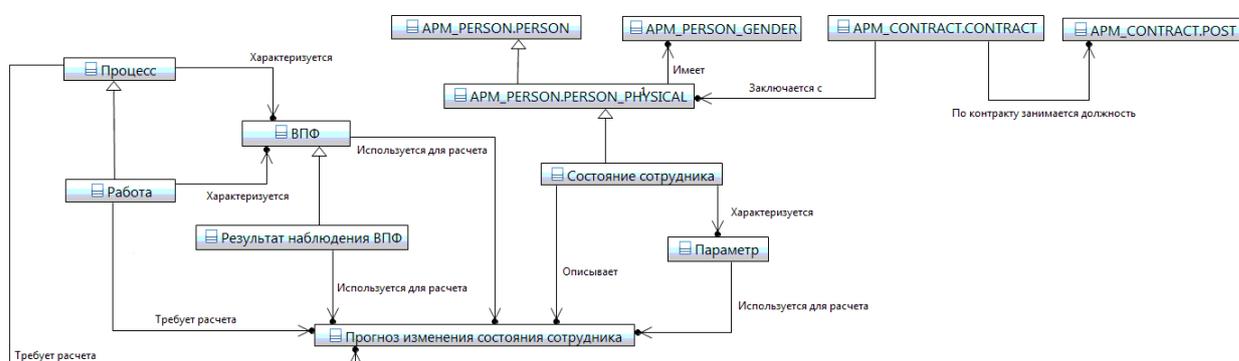


Рисунок 6.22 – Диаграмма классов, описывающая общесистемное представление пятого функционального требования на уровне знаний

воречий по причинам, совпадающим с рассмотренными выше для представлений K_1^{fU} - K_5^{fU} .

Ход применения метода выявления нелогичных ФТ к ФМ БТ для представлений K_1^{fIS} - K_5^{fIS} показан в табл. 6.2.

По результатам применения метода выявления нелогичных ФТ можно сделать следующие выводы:

а) третье ФТ к ФМ БТ остается нелогичным на 50%, поскольку для описания термина ПрО «Вредный производственный фактор» продолжает использовать упрощенную ОТ «ВПФ» вместо детализированной ОТ «ВПФ + Результат наблюдения ВПФ», используемой в четвертом и пятом ФТ;

Таблица 6.2 – Результаты применения метода выявления нелогичных функциональных требований

для представлений K_1^{fIS} - K_5^{fIS}

Номер анализируемого требования	Номер итерации метода	Количество онтологических точек требования	Количество нелогичных онтологических точек требования	Номер сравниваемого требования	Количество совпадающих онтологических точек	Значение I_{gr} анализируемого требования, %
1	1	2	2	2	1	50%
1	2	2	1	3	1	– (требование логично)
2	1	5	5	1	1	80%
2	2	2	4	3	0	80%
2	3	2	4	4	0	80%
2	4	2	4	5	4	– (требование логично)
3	1	2	2	1	1	50%
3	2	2	1	2	0	50%
3	3	2	1	4	0	50%
3	4	2	1	5	0	50%
4	1	2	2	1	1	50%
4	2	2	1	2	0	50%
4	3	2	1	3	0	50%
4	4	2	1	5	1	– (требование логично)
5	1	8	8	1	1	87,5%
5	2	8	7	2	3	50%
5	3	8	4	3	0	50%
5	4	8	4	4	1	37,5%

б) нелогичность пятого ФТ к ФМ БТ снизилась с 60% до 37,5% за счет унификации представления элементов, описывающих термин Про «Сотрудник».

В силу рассмотренных выше причин принято решение продолжить разработку первой очереди ФМ БТ. Следует отметить, что уже на стадии анализа ФТ к ФМ БТ за счет ре-использования элементов библиотеки ранее реализованных требований нелогичность СТ уменьшилась. Это, в свою очередь, позволяет повысить эффективность разработки обеспечивающей части ФМ БТ за счет использования одних и тех же сервисов, реализующих соответствующие ОТ, в ходе реализации различных ИТ-услуг (функций ФМ БТ).

Сформированные общесистемные представления ФТ к ФМ БТ на уровне знаний K_1^{fIS} - K_5^{fIS} являются исходными данными для применения метода синтеза вариантов АД создаваемой системы. Для применения метода было также выбрано значение коэффициента отталкивания $r = 2$, поскольку ФМ БТ рассматривается как самостоятельный ФМ. Значение величины ε принимается по умолчанию $\varepsilon = 0,1 \times Profit_{max}$ (см. подразд. 4.3).

В ходе выполнения Этапа 1 метода синтеза вариантов АД формируется базовый вариант АД, описывающий ФМ БТ как совокупность из пяти ИТ-услуг IT_{act_1} - IT_{act_5} , каждая из которых описана соответствующим общесистемным представлением СФТ на уровне знаний K_1^{fIS} - K_5^{fIS} .

Результат выполнения Этапа 2 метода синтеза вариантов АД приведен в табл. Ж.1 Приложения Ж.

Результаты выполнения итераций Этапа 3 метода синтеза вариантов АД приведены в табл. Ж.2 Приложения Ж.

Результаты выполнения Этапа 4 метода синтеза вариантов АД приведен в табл. 6.3.

В результате применения метода синтеза вариантов АД ФМ БТ были выделены следующие варианты описания:

а) вариант, предусматривающий объединение третьего и четвертого ФТ и общесистемных представлений этих требований K_3^{fIS} и K_4^{fIS} на уровне

знаний (оптимальный вариант);

Таблица 6.3 – Результат выполнения Этапа 4 метода синтеза вариантов описаний архитектуры функционального модуля безопасности труда

Номер ИТ-услуги	Общесистемные представления, описывающие ИТ-услугу	Значение $S(IT_{асм_j})$ для ИТ-услуги	Значение $W(IT_{асм_j})$ для ИТ-услуги
1	K_1^{fIS}	7	7
2	K_2^{fIS}	12	12
3	–	–	–
4	$\{K_3^{fIS}, K_4^{fIS}\}$	14	8
5	K_5^{fIS}	28	28
Результат расчета функции $Pr ofit(IT_{асм}, r)$ на итерации 11			
$ IT_{асм_j} = 5$	$r = 2, \varepsilon = 0,0110714$	$Pr ofit(IT_{асм}, r) = 0,139881$ (зафиксирован в результате выполнения Шага 3.9, $m = 1$)	
1	K_1^{fIS}	7	7
2	–	–	–
–	–	–	–
4	$\{K_3^{fIS}, K_4^{fIS}\}$	14	8
5	$\{K_2^{fIS}, K_5^{fIS}\}$	40	31
Результат расчета функции $Pr ofit(IT_{асм}, r)$ на итерации 17			
$ IT_{асм_j} = 4$	$r = 2, \varepsilon = 0,0139881$	$Pr ofit(IT_{асм}, r) = 0,132721$ (зафиксирован в результате выполнения Шага 3.10)	
1	K_1^{fIS}	7	7
2	$\{K_2^{fIS}, K_5^{fIS}\}$	40	31
–	–	–	–
4	$\{K_3^{fIS}, K_4^{fIS}\}$	14	8
5	–	–	–
Результат расчета функции $Pr ofit(IT_{асм}, r)$ на итерации 22			
$ IT_{асм_j} = 4$	$r = 2, \varepsilon = 0,0139881$	$Pr ofit(IT_{асм}, r) = 0,132721$ (зафиксирован в результате выполнения Шага 3.10)	

б) вариант, предусматривающий объединение третьего и четвертого ФТ и общесистемных представлений этих требований K_3^{fIS} и K_4^{fIS} на уровне знаний, а также пятого и второго ФТ и общесистемных представлений этих требований K_5^{fIS} и K_2^{fIS} на уровне знаний (рациональный вариант);

в) вариант, предусматривающий объединение второго и пятого ФТ и общесистемных представлений этих требований K_2^{fIS} и K_5^{fIS} на уровне знаний, а также третьего и четвертого ФТ и общесистемных представлений этих требований K_3^{fIS} и K_4^{fIS} на уровне знаний (рациональный вариант).

Необходимость отдельного рассмотрения вариантов б) и в) обусловлена различием пользователей функции, которая рассматривается в качестве основной. Для второго варианта основными пользователями базовой функции являются специалист по БТ и пользователь ФМ (представитель руководства предприятия). Для третьего варианта основным пользователем базовой функции является специалист по БТ.

Полученные варианты АД ФМ БТ являются исходной информацией для применения метода поиска равновесий Нэша в чистых стратегиях биматричной игры (4.43). Результаты выполнения Этапов 1-3 данного метода приведены в табл. 6.4-6.5.

В результате выполнения Этапа 4 метода поиска равновесий Нэша в чистых стратегиях биматричной игры было обнаружено, что равновесие по Нэшу достигается в случае выбора Поставщиком и Потребителем второго или третьего вариантов АД ФМ БТ.

Выбранные варианты ОРА ФМ БТ имеют вид, показанный на рис. 6.22.

Для выбранных вариантов была проведена оценка объема работ по созданию ФМ БТ методом ОТ. Исходные данные, на основе которых проводилась оценка, приведены в табл. Ж.3 Приложения Ж. Результаты расчета трудозатрат с использованием ОТ приведены в табл. 6.6. При расчете была учтена низкая квалификация исполнителей работ по созданию обеспечивающей части ФМ БТ.

Таблица 6.4 – Матрица выигрыша Поставщика для выбранных вариантов описания архитектуры функционального модуля безопасности труда

Номер вы- бранного варианта	Значение функции вы- игрыша По- ставщика для первого тре- бования	Значение функции вы- игрыша По- ставщика для второго тре- бования	Значение функции вы- игрыша По- ставщика для третьего и четвертого требований	Значение функции вы- игрыша По- ставщика для пятого требо- вания
1	0,429	0,917	0	0,321
2	0,429	0	0	0,355
3	0,429	0,355	0	0

Таблица 6.5 – Матрица выигрыша Потребителя для выбранных вариантов описания архитектуры функционального модуля безопасности труда

Номер вы- бранного варианта	Значение функции вы- игрыша По- требителя для первого тре- бования	Значение функции вы- игрыша По- требителя для второго тре- бования	Значение функции вы- игрыша По- требителя для третьего и чет- вертого требо- ваний	Значение функции вы- игрыша По- требителя для пятого требования
1	0,571	0,25	1	0,714
2	0,571	0	1	0,677
3	0,571	0,677	1	0

Оценка объема работ методом ОТ по созданию ФМ БТ на основе базового варианта АД данной системы, в котором каждая ИТ-услуга должна была реализовать отдельное ФТ к ФМ, приведена в табл. 6.7.

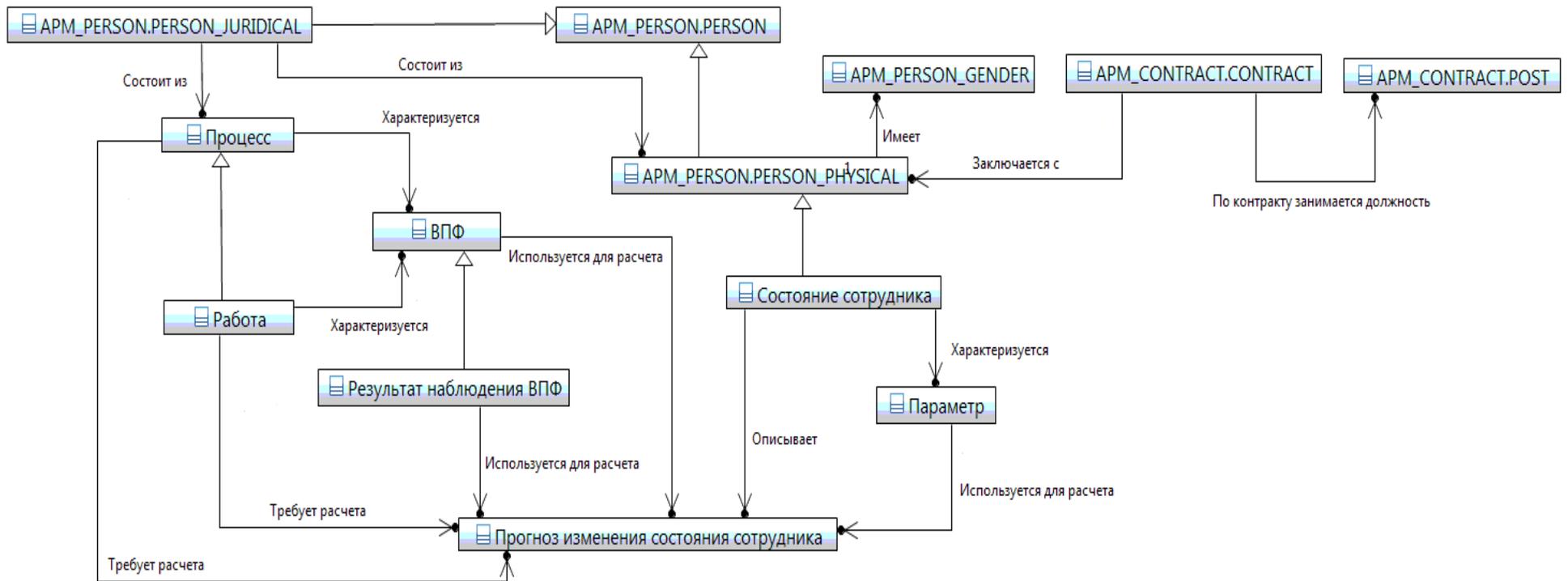


Рисунок 6.22 – Диаграмма классов, описывающая выбранные варианты описания рациональной архитектуры функционального модуля безопасности труда

Таблица 6.6 – Результаты расчета величины объема работ по реализации ИТ-услуг варианта описания рациональной архитектуры функционального модуля безопасности труда

Требования, реализуемые ИТ-услугой	<i>OntP</i>	<i>%reuse</i>	<i>NOntP</i>	<i>PROD</i>	<i>PM_i</i>
Первое	23	42,9	13,133	7	1,876
Второе и пятое	44	35,5	28,38	7	4,057
Третье и четвертое	20	0	20	7	2,857
$PM = \sum PM_i = 8,787$ чел/мес					

Таблица 6.7 – Результаты расчета величины объема работ по реализации ИТ-услуг варианта описания рациональной архитектуры функционального модуля безопасности труда

Требования, реализуемые ИТ-услугой	<i>OntP</i>	<i>%reuse</i>	<i>NOntP</i>	<i>PROD</i>	<i>PM_i</i>
Первое	23	42,9	13,133	7	1,876
Второе	19	91,67	1,5827	7	0,2261
Третье	14	0	14	7	2
Четвертое	16	0	16	7	2,286
Пятое	37	32,14	25,1022	7	3,587
$PM = \sum PM_i = 9,9751$ чел/мес					

Значения основных характеристик ИТ-проектов по созданию ФМ БТ на основе базового варианта АД и варианта ОРА приведены в табл. 6.8. При расчете оценок времени выполнения проекта и потребности в персонале использова-

лись выражения (4.63)-(4.65), являющиеся элементами стандартной модели COCOMO II Early Design [200].

Таблица 6.8 – Результаты сравнения характеристик IT-проектов по созданию функционального модуля безопасности труда на основе базового варианта описания архитектуры и варианта описания рациональной архитектуры

Вариант описания архитектуры	Количество реализуемых ИТ-услуг	Оценка трудозатрат, чел/мес	Оценка времени выполнения IT-проекта, мес.	Оценка потребности в персонале, чел.
Базовый	5	9,9751	5,266	1,894
Вариант описания рациональной архитектуры	3	8,787	5,142	1,709

Результаты сравнения приведенных характеристик позволяют сделать следующие выводы:

а) принятие любого из предлагаемых вариантов ОРА позволяет сократить количество реализуемых ИТ-услуг на две (с пяти до трех), что позволит снизить сложность создаваемого ФМ БТ;

б) принятие любого из предлагаемых вариантов ОРА позволяет сократить оцениваемые трудозатраты на выполнение IT-проекта на 1,1881 человеко-месяцев;

в) принятие любого из предлагаемых вариантов ОРА позволяет сократить оцениваемые затраты времени выполнения IT-проекта на 0,124 месяца (около 3 рабочих дней сорокачасовой рабочей недели);

г) принятие любого из предлагаемых вариантов ОРА позволяет сократить оцениваемую потребность в персонале на 0,185 единиц (почти на 0,2 базовой ставки оплаты труда основных исполнителей ИТ-проекта).

Дополнительным преимуществом предлагаемой ИТ следует признать то, что ее применение, как показано выше, позволяет повысить эффективность макропроектирования даже для ИТ-проектов создания ИС и ИТ управления предприятиями и организациями с малым количеством реализуемых ИТ-услуг (функций).

Рассмотренный способ применения ИТ формирования и анализа требований к ИС является универсальным и может использоваться как для реализации остальных ФМ комплексной системы управления ПАО «Харьковгоргаз», так и для реализации ИС или их отдельных ФМ в любой другой ПрО. Подтверждением этому являются акты о внедрении результатов диссертационной работы, приведенные в Приложении А. При этом эффект от внедрения увеличивается с увеличением масштаба ИС вне зависимости от того, создается ли данная ИС «с нуля» или же проходит реинжиниринг. Так, для указанной в акте внедрения комплексной системы автоматизации страховой компании (компьютерные программы «ProfITsoft FRONT-OFFICE» и «ProfITsoft BACK-OFFICE») сравнительные результаты применения ИТ формирования и анализа требований к ИС имеют вид, приведенный в табл. 6.9.

Результаты сравнения приведенных характеристик позволяют сделать следующие выводы:

а) принятие предлагаемого варианта ОРА позволяет сократить количество реализуемых ИТ-услуг на четыре (с двадцати семи до двадцати трех), что позволит снизить сложность модернизируемой ИС;

б) принятие предлагаемого варианта ОРА позволяет сократить оцениваемые трудозатраты на выполнение ИТ-проекта на 13,50146 человеко-месяцев;

в) принятие предлагаемого варианта ОРА позволяет сократить оцениваемые затраты времени выполнения ИТ-проекта на 0,308 месяца (немногим более, чем на 7 рабочих дней сорокачасовой рабочей недели);

Таблица 6.9 – Результаты сравнения характеристик IT-проектов по внедрению комплексной системы автоматизации страховой компании (компьютерные программы «ProfITsoft FRONT-OFFICE» и «ProfITsoft BACK-OFFICE») на основе базового варианта описания архитектуры и варианта описания рациональной архитектуры

Вариант описания архитектуры	Количество реализуемых ИТ-услуг	Оценка затрат, чел/мес	Оценка времени выполнения ИТ-проекта, мес.	Оценка потребности в персонале, чел.
Базовый	27	67,307	7,534	8,934
Вариант описания рациональной архитектуры	23	53,86554	7,226	7,454

г) принятие предлагаемого варианта ОРА позволяет сократить оцениваемую потребность в персонале на 1,48 единиц (почти на 1,5 базовой ставки оплаты труда основных исполнителей ИТ-проекта).

6.4 Выводы к шестому разделу

1. Рассмотрены основные элементы реализации предложенной ИТ формирования и анализа требований к ИС. Сформулированы основные требования к реализации данной ИТ. Предложена концепция модернизации основного шаблона проектирования MVC обеспечивающей части данной ИТ. Рассмотрены особенности основных методов, изложенных в разд. 4, на примере

метода формирования представления СТ на уровне информации и группы методов формирования представлений СТ на уровне знаний.

2. Проведена апробация ИТ формирования и анализа требований к ИС в ходе заполнения библиотеки ранее реализованных требований путем рефакторинга ранее разработанной ИС. В качестве примера такой ИС рассмотрена ИАС «Реестр ЛУК». В результате рефакторинга данной ИС сформированы описания реализованных ФТ к ИТ-услугам, а также описания проектных решений по ИО и ПО (ИТ-сервисов), реализующих эти ИТ-услуги. Приведены доступные для ре-использования описания элементов архитектуры ИС (см. рис. 6.5).

3. Проведена апробация ИТ формирования и анализа требований к ИС, демонстрирующая эффект от ре-использования элементов библиотеки ранее реализованных требований в ходе макропроектирования первой очереди ФМ БТ. Был проведен синтез вариантов АД ФМ БТ и выбраны варианты ОРА (см. рис. 6.22). Показан эффект от использования выбранных вариантов, составляющий сокращение трудозатрат на 1,18 чел./мес и сокращение затрат времени на 0,124 месяца. Данные эффект позволяет утверждать, что применение предложенной ИТ повышает эффективность макропроектирования даже для ИС и ИТ с малым количеством ИТ-услуг (функций). Также показан эффект от применения данной ИТ в ходе инициации ИТ-проектов создания и модернизации ИС со средним и большим количеством ИТ-услуг (функций). Для комплексной системы автоматизации страховой компании (компьютерные программы «ProfITsoft FRONT-OFFICE» и «ProfITsoft BACK-OFFICE») данный эффект заключается в сокращении трудозатрат на 13,5 чел./мес и сокращении затрат времени на 0,3 месяца.

Основные результаты данного раздела изложены в работах [134, 212-215, 219, 222].

ВЫВОДЫ

В данной диссертационной работе решена актуальная научно-прикладная проблема разработки методологических основ, моделей, методов и информационной технологии, позволяющих сформировать рациональный вариант описания архитектуры создаваемой информационной системы с максимально возможным удовлетворением функциональных требований Поставщика и Потребителя и минимальными затратами на реализацию.

1. Проведен анализ существующих подходов к формированию требований и созданию АД ИС управления организационно-техническими объектами и процессами. Определены основные особенности этих подходов, а также главные цели Поставщика и Потребителя как основных участников ИТ-проекта создания или реинжиниринга ИС. Исходя из результатов анализа сформулирована научно-прикладная проблема, которая определяет цель и задачи исследования.

2. Получил дальнейшее развитие сервисный подход к созданию ИС, который, в отличие от существующих, рассматривает ИС как совокупность ИТ-услуг и реализующих эти услуги ИТ-сервисов, формируемых на основе имеющихся проектных решений, удовлетворяющих множеству требований, выдвинутых к системе, что позволяет сократить затраты на создание ИС за счет ре-использования системных компонентов.

3. Разработана концепция, согласно которой требование к ИС рассматривается как совокупность взаимосвязанных формальных представлений этого требования на уровнях данных, информации и знаний. В соответствии с данной концепцией получили дальнейшее развитие математические модели, описывающие множество требований к ИС на разных стадиях их формирования, которые, в отличие от существующих, описывают связи между представлениями требований к ИС в ходе их формирования. Это позволяет формально описать множество возможных АФ создания ИС при помощи сервисного подхода.

4. Впервые разработана математическая модель ППТ к ИС, унифициру-

ющая описание отдельного требования к ИС, что позволяет формализовать процесс ре-использования требований к ИС в ходе создания новых и реинжиниринга существующих систем.

5. Впервые разработана математические модели АФ макропроектирования ИС и СФТ на уровне знаний как основного элемента данного фреймворка, основанные на формальном представлении фреймворка в виде множеств структурных и поведенческих ППТ к ИС, что позволяет установить общие особенности и ограничения и решить задачу синтеза АД создаваемой ИС.

6. Усовершенствованы теоретико-множественные модели, описывающих синтаксис и семантику описаний требований к ИС на уровнях информации и знаний, который, в отличие от существующих, основаны на модели ППТ к ИС. Результаты усовершенствования позволяют унифицировать работы по формированию, хранению и обработке представлений требований к ИС на уровнях информации и знаний вне зависимости от конкретных проектов создания ИС, для которых требования были сформулированы.

7. Впервые разработаны методы формирования представлений ФТ к ИС на уровнях информации и знаний, позволяющие формализовать процессы формирования описаний и анализа требований к ИС, что обеспечивает сокращение затрат времени на выполнение данных процессов и возможность автоматизированного проектирования архитектуры ИС.

8. Получил дальнейшее развитие метод синтеза вариантов АД создаваемой ИС, который, в отличие от существующего, выделяет для дальнейшего исследования варианты описаний, незначительно хуже оптимального. Это позволит в максимально возможной мере удовлетворить требования Поставщика и Потребителя в процессе формирования компромиссного ОРА создаваемой ИС за счет увеличения множества исходных вариантов АД.

9. Усовершенствована теоретико-игровая модель синтеза ОРА создаваемой ИС, которая, в отличие от существующих, описывает процесс синтеза ОРА как выбор варианта описания, в наибольшей степени соответствующего представлениям создаваемой системы Поставщика и Потребителя. Использование данной

модели позволяет автоматизировать процесс проектирования архитектуры системы путем поиска точек равновесия по Нэшу в чистых и смешанных стратегиях.

10. Впервые разработана модель ОТ, которая описывает создаваемую ИС как множество отдельных ветвей таксономий фреймов, и метод формирования описаний ОТ на основе данной модели, что позволяет автоматизировать процесс количественного оценивания объема работ по созданию ИС на основе разработанных моделей ФТ на уровне знаний.

11. Усовершенствованы методы анализа СФТ к создаваемой ИС путем выявления противоречивых и нелогичных требований в ходе решения задачи синтеза ОРА ИС. Применение этих методов позволяет унифицировать и автоматизировать выполнение работ по анализу ФТ в рамках макропроектирования.

12. Уточнено определение ИТ формирования и анализа требований к ИС как совокупности унифицированных в рамках АФ макропроектирования ИС и основанных на знаниях методов, приемов и способов применения КСА или его отдельных элементов, используемой для выполнения одной или нескольких работ в рамках процессов создания ИС, непосредственно работающих с требованиями к данной системе. Исходя из данного определения сформулирована концепция ИТ формирования и анализа требований к ИС. На основе этой концепции были разработаны решения по ИО, реализующие разработанные модели ППТ к ИС на уровнях данных, информации и знаний.

13. Предложена концепция модификации шаблона проектирования ПО MVC, положенная в основу реализации основных элементов ИТ формирования и анализа требований к ИС. Рассмотрены основные решения по реализации разработанных методов в рамках ПО и ИО предлагаемой ИТ.

14. Разработанные методы, модели и элементы ИТ формирования и анализа требований к ИС прошли апробацию, в ходе которой были рассмотрены два основных способа применения данной ИТ. Первый способ позволяет наполнить библиотеку ранее реализованных требований путем рефакторинга ранее выполненных ИТ-проектов создания ИС. Второй способ позволяет ре-использовать ранее реализованные требования к ИС и соответствующие этим требованиям ИТ-

сервисы в ходе выполнения процессов макропроектирования новых ИС. Апробация проводилась во время выполнения работ по разработке ФМ БТ комплексной системы управления ПАО «Харьковгоргаз». Анализ результатов разработки показал сокращение трудозатрат и затрат времени на создание ФМ БТ на основе варианта ОРА по сравнению с аналогичными затратами на создание и внедрение базового варианта описания архитектуры данного модуля на 1,18 чел./мес. Для ИС автоматизации страховой компании ООО «ProfITsoft» результаты применения предложенной ИТ привели к сокращению трудозатрат на 13,5 чел./мес. Результаты диссертационной работы были также внедрены во время выполнения таких работ: модернизация ИС ИТ СКБ «ПОЛИСВИТ» ГНПП «Объединение Коммунар»; разработке ИС дистанционного мониторинга транспорта ООО ТД «Технодар». Кроме того, результаты работы нашли применение в учебном процессе Харьковского национального университета радиоэлектроники при подготовке магистров специальности 122 - «Компьютерные науки и информационные технологии» (специализация «Информационные управляющие системы и технологии»). Общий экономический эффект от внедрения результатов диссертационной работы составляет более 860 тыс. грн. Все результаты внедрения подтверждены соответствующими актами и свидетельствуют об эффективности применения результатов диссертационной работы.

Использование результатов диссертационной работы позволяет разработать принципиально новые стандарты, методы и ИТ разработки, сопровождения и модернизации ИС управления организационными и организационно-техническими ОА различного уровня сложности.

ПЕРЕЧЕНЬ ССЫЛОК

1. ГОСТ 34.003–90 Автоматизированные системы. Требования и определения [Электронный ресурс]. – Введ. 01.01.1992. – Режим доступа: <http://www.vashdom.ru/gost/34.003-90/>.
2. ДСТУ 2226-93. Автоматизовані системи. Терміни і визначення. – Введ. 01.07.1994 р. – К.: Держстандарт України, 1994. – 92 с.
3. ДСТУ 2941-94. Системи оброблення інформації. Розроблення систем. Терміни та визначення: чинний від 1996-01-01. – К.: Держстандарт України, 1995. – 20 с.
4. Кузнецов, С.Д. Базы данных: языки и модели [Текст]: / С.Д. Кузнецов. – М.: Бином-Пресс, 2008. – 720 с.
5. Чернавский, Д.С. Синергетика и информация (динамическая теория информации) [Текст] / Д.С. Чернавский. – М.: Едиториал УРСС, 2004. – 288 с.
6. Когаловский, М.Р. Перспективные технологии информационных систем [Текст] / М.Р. Когаловский. – М.: ДМК Пресс, 2003. – 288 с.
7. An Introduction to Structured Systems Analysis & Design Methodology (SSADM) [Электронный ресурс] / Сайт «Office of Government Chief Information Officer». – Режим доступа: http://www.ogcio.gov.hk/en/infrastructure/methodology/ssadm/doc/s3a_pub.pdf. - Заголовок с экрана.
8. ГОСТ ИСО/МЭК 15288–2005. Системная инженерия. Процессы жизненного цикла систем [Текст]. – Введ. 01–01–2007. – М.: Стандартиформ, 2006. – 57 с.
9. ГОСТ ИСО/МЭК 12207–2010. Системная и программная инженерия. Процессы жизненного цикла программных средств [Текст]. – Введ. 01–03–2012. – М.: Стандартиформ, 2011. – 106 с.

10. Руководство к своду знаний по управлению проектами (Руководство PMBOK) – Пятое издание [Текст]. – Newton Square: Project Management Institute, Inc., 2013. – 586 с.
11. Фатрелл, Р.Т. Управление программными проектами: достижение оптимального качества при минимуме затрат [Текст] / Р.Т. Фатрелл, Д.Ф. Шафер, Л.И. Шафер. – М.: Вильямс, 2003. – 1136 с.
12. Young, R.R. Effective Requirements Practices [Text] / R.R. Young. – Reading: Addison-Wesley Professional, 2001. – 400 p.
13. Леффингуэлл, Д. Принципы работы с требованиями к программному обеспечению. Унифицированный подход [Текст] / Д. Леффингуэлл, Д. Уидриг. – М.: Вильямс, 2002. – 448 с.
14. Requirements [Электронный ресурс] // Сайт «IBM Knowledge Center». – Режим доступа: http://www-01.ibm.com/support/knowledgecenter/SSSHCT_7.1.0/com.ibm.reqpro.reqweb_help/c_req.html. – Заголовок с экрана.
15. Lawrence, B. Unresolved ambiguity: The silent source of risk in your project [Text] / B. Lawrence // American Programmer. – 1996. – Vol. 9, № 4. – P. 18–22.
16. Sommerville, I. Requirements Engineering: A Good Practice Guide [Text] / I. Sommerville, P. Sawyer. – Chichester : John Wiley & Sons, 1997. – 391 p.
17. РД 50–34.698–90. Автоматизированные системы. Требования к содержанию документов [Электронный ресурс]. – Введ. 01.01.1992. – Режим доступа: <http://document.ua/kompleks-standartov-i-rukovodjashih-dokumentov-na-avtomatizirovanih-sistemah-3019.html>. – Заголовок с экрана.
18. ГОСТ 34.602–89. Техническое задание на создание автоматизированной системы [Электронный ресурс]. – Введ. 01.01.1990. – Режим доступа: http://vt.ulstu.ru/sites/default/files/%D0%93%D0%9E%D0%A1%D0%A2%2034.602-89_0.pdf. – Заголовок с экрана.
19. SSADM Version 4 Reference Manual [Text]. – Oxford: NCC Blackwell, 1990. – 1400 p.

20. Кириллов, В.П. Технология SSADM: методика определения требований к автоматизированной системе [Текст] / В.П. Кириллов // Компьютеры + Программы, 1994. – № 3 (11). – С. 30–36.
21. Виггерс, К.И. Разработка требований к программному обеспечению [Текст] / К.И. Виггерс. – М.: Русская редакция, 2004. – 576 с.
22. Ralph, P. The illusion of requirements in software development [Text] / P. Ralph // Requirements Engineering. –2013. – Vol. 18, Iss. 3. – pp 293-296.
23. Халл, Э. Разработка и управление требованиями: практическое руководство пользователя [Электронный ресурс] / Э. Халл, К. Джексон, Дж. Дик. – 2-е изд. – 2005. – Режим доступа: http://80.250.162.180/2002/Yury.Kupriyanov/eBook_RU_Requirements_Engineering.pdf. – Заголовок с экрана.
24. Watson, M. Managing Smaller Projects: A Practical Approach [Text] / M. Watson. – Multi-Media Publications Inc., 2006. – 240 p.
25. Capturing Architectural Requirements [Электронный ресурс] // Сайт «IBM DeveloperWorks». – Режим доступа: <http://www.ibm.com/developerworks/rational/library/4706.html#N100A7>. – Заголовок с экрана.
26. Business Analysis Body of Knowledge (BABOK Guide). Version 2.0 [Text]. – International Institute of Business Analysis, 2009. – 263 p.
27. Шамаев, И. Типы требований и методики сбора требований в разрезе различных стандартов [Электронный ресурс] / И. Шамаев // Сайт «Business Analysis in Russia. База знаний по бизнес-анализу». – Режим доступа: http://iiba.ru/wp-content/uploads/2013/04/typy_trebovanij_i_metodiki_sbor_trebovanij_v_razr_eze_razlichnyx_standartov.pdf. – Заголовок с экрана.
28. Руководство к своду знаний по управлению проектами (руководство PMBOK). Четвертое издание [Текст]. – Project Management Institute, Inc., 14 Campus Boulevard Newtown Square, Pennsylvania 19073-3299 USA, 2008. – 242 p.

29. Подходы к управлению требованиями в SWEBOK [Электронный ресурс] // Сайт «Business Analysis in Russia. База знаний по бизнес-анализу». – Режим доступа: <http://iiba.ru/requirements-in-swebok/>. – Заголовок с экрана.
30. Requirements Engineering Body of Knowledge. Introduction to Requirement Engineering. – Global Association for Software Quality, 2013. – 59 p.
31. ISO/IEC/IEEE 29148:2011. Systems and software engineering — Life cycle processes — Requirements engineering [Электронный ресурс]. – Режим доступа: https://webstore.iec.ch/preview/info_isoiecieee29148%7Bed1.0%7Den.pdf. – Заголовок с экрана.
32. CMMI® for Development, Version 1.3 [Электронный ресурс] // Сайт «Software Engineering Institute. Carnegie Mellon University». – Режим доступа: <http://www.sei.cmu.edu/reports/10tr033.pdf>. – Заголовок с экрана.
33. Брукс, П. Метрики для управления ИТ-услугами [Текст] / П. Брукс. – М.: Альпина Бизнес-Букс, 2008. – 283 с.
34. Кобёрн, А. Современные методы описания функциональных требований к системам [Текст] / А. Кобёрн. – М.: «Лори», 2002. – 288 с.
35. Maciaszek, L.A. Requirements Analysis and System Design [Text] / L.A. Maciaszek: 2d ed. – Reading: Addison Wesley, Harlow England, 2005. – 504 p.
36. Davis, Alan M. Principles of Software Development / Alan M. Davis. – New York: McGraw-Hill, 1995. – 240 p.
37. Моделирование требований пользователей [Электронный ресурс] // Сайт «Microsoft Developer Network». – Режим доступа: <https://msdn.microsoft.com/ru-ru/library/dd409376.aspx>. – Заголовок с экрана.
38. Петров, Э.Г. Методология структурного системного анализа и проектирования крупномасштабных ИУС. Концепции и методы [Текст] / Э.Г. Петров. С.И. Чайников, А.О. Овезгельдыев. – Харьков: Рубикон, 1997. – Часть 1. – 140 с.

39. Формализованные методики определения требований [Электронный ресурс] // Сайт «Plastyle.com.ua Разработка компьютерных систем проектирования». – Режим доступа: <http://www.plastyle.com.ua/shema/268-2010-09-16-06-27-29>. – Заголовок с экрана.
40. Dietz, J.L.G. From Information Systems Development to Enterprise Engineering [Text] / J.L.G. Dietz // Information System Development. Business Systems and Services: Modeling and Development / J. Pokorny, V. Repa, K. Richta, W. Wojtkowski, H. Linger, C. Barry, M. Lang (Eds.). – Springer, 2011. – pp. 1-24.
41. Paulk, M. The Capability Maturity Model: Guidelines for Improving the Software Process [Text] / M. Paulk et al. – Reading, MA: Addison-Wesley, 1995.
42. Davis, A.M. Just Enough Requirements Management: Where Software Development Meets Marketing [Text] / A.M. Davis. – Dorset House, 2005. – 240 p.
43. Perry, D.E. Software architecture [Электронный ресурс] / D.E. Perry, A.L. Wolf // Режим доступа: <http://users.ece.utexas.edu/~perry/work/papers/swa89.pdf>. – Заголовок с экрана.
44. Kruchten, Ph. The Rational Unified Process: An Introduction [Text] / Ph. Kruchten. – Reading: Addison-Wesley Professional, 2004. – 310 p.
45. Сайт ISO/IEC/IEEE 42010 Website [Электронный ресурс]. – Режим доступа: <http://www.iso-architecture.org/ieee-1471/index.html>. – Заголовок с экрана.
46. Architecture [Электронный ресурс] // Сайт «Oxford Dictionaries». – Режим доступа: <http://oxforddictionaries.com/definition/english/architecture?q=architecture>. – Заголовок с экрана.
47. Helland, P. Metropolis [Электронный ресурс] / P. Helland // Сайт MSDN. – Режим доступа: <http://msdn.microsoft.com/en-us/library/aa480026.aspx>. – Заголовок с экрана.
48. Veryard, R. Metropolis and SOA Governance [Электронный ресурс] / R. Veryard, Ph. Boxer // Сайт MSDN. – Режим доступа:

- <http://msdn.microsoft.com/en-us/library/aa480051.aspx>. – Заголовок с экрана.
49. IEEE Std 1471-2000. IEEE Recommended Practice for Architectural Description of Software-Intensive Systems [Text]. – Approved 21 September 2000. – New York: The Institute of Electrical and Electronics Engineers, Inc., 2000. – 34 p.
50. Захман, Дж. Бизнес и информационные технологии [Электронный ресурс] / Дж. Захман. – Режим доступа: <http://www.intuit.ru/department/itmngt/entarc/1/>. – Заголовок с экрана.
51. Holcman, S.B. Driving Efficiency and Innovation by Consistently Managing Complexity and Change [Электронный ресурс] // Сайт MSDN Architecture Center. – Режим доступа: <http://msdn.microsoft.com/en-us/architecture/ff476941>. – Заголовок с экрана.
52. Enterprise Architecture Definition [Электронный ресурс]. – Режим доступа: <http://samvak.tripod.com/earf.pdf>. – Заголовок с экрана.
53. Информационная система [Электронный ресурс] // Сайт «Глоссарий.ру». – Режим доступа: [http://www.glossary.ru/cgi-bin/gl_sch2.cgi?RI\(uwsg.outt:!!xoxyls:](http://www.glossary.ru/cgi-bin/gl_sch2.cgi?RI(uwsg.outt:!!xoxyls:). – Заголовок с экрана.
54. Соммервил, И. Инженерия программного обеспечения [Текст] / И. Соммервил. – М.: Издательский дом «Вильямс», 2002. – 624 с.
55. Aksirt, M. Software Architectures and Component Technology [Text] / M. Aksirt. – Boston, Kluwer Academic Publisher, 2002. – 385 p.
56. Палагин, А.В. Проблемная ориентация в развитии компьютерных архитектур [Текст] / А.В. Палагин, А.Ф. Кургаев // Кибернетика и системный анализ. – 2003. – № 4. – С. 167-180.
57. Глушков, В.М. Проблемная ориентация и другие пути повышения эффективности ЭВМ [Текст] / В.М. Глушков // Кибернетика. Вопросы теории и практики. – М.: Наука, 1986. – С. 162-170.

58. Смит, К. Эффективные решения: практическое руководство по созданию гибкого и масштабируемого программного обеспечения: пер. с англ. [Текст] / К. Смит, Л. Уильямс. – М.: Изд. «Вильямс», 2003. – 448 с.
59. Starke, G. Effektive Software-Architekturen: Ein praktischer Leitfaden [Text] / G. Starke. – Muenchen: Hanser Verlag, 2002. – 432 p.
60. Foegen, M. Die Rolle der Architektur in der Anwendungsentwicklung [Text] / M. Foegen, J. Batterfeld // Informatik-Spektrum. Springer. – 2001. - № 5(24). – p. 290-301.
61. Архитектуры, модели и технологии программного обеспечения информационно-управляющих систем: монография / Ткачук Н.В., Шеховцов В.А., Кукленко Д.В., Сокол В.Е. Под ред. М.Д. Годлевского. – Харьков: НТУ «ХПИ», 2005. – 546 с.
62. MOD Architecture Framework [Электронный ресурс] // Сайт «GOV.UK». – Режим доступа: <https://www.gov.uk/guidance/mod-architecture-framework>. – Заголовок с экрана.
63. The Open Group Architecture Framework (TOGAF) – Core Concepts [Электронный ресурс] // Сайт TOGAF®. Version 9 Enterprise Edition. – Режим доступа: http://www.togaf.org/togaf9/chap02.html#tag_03_01. – Заголовок с экрана.
64. Kruchten, Ph. Architectural Blueprints – The “4+1” View Model of Software Architecture [Text] / Ph. Kruchten // IEEE Software. – 1995. – Vol. 12 (6). – p. 42-50.
65. Farooqui, K. The ISO Reference Model for Open Distributed Processing - An Introduction [Электронный ресурс] / К. Farooqui, L. Logrippo, J. de Meer // Сайт «Institute For Enterprise Architecture Developments». – Режим доступа: <http://www.enterprise-architecture.info/Images/Documents/RM-ODP2.pdf>. - Заголовок с экрана.
66. OMG Systems Modeling Language [Электронный ресурс] // Сайт «OMG Systems Modeling Language. The Official OMG SysML site». – Режим доступа: <http://www.omgsysml.org/>. – Заголовок с экрана.

67. ArchiMate® [Электронный ресурс] // Сайт «The Open Group». – Режим доступа: <http://www.opengroup.org/subjectareas/enterprise/archimate>. – Заголовок с экрана.
68. The Acme Project [Электронный ресурс] // Сайт «Carnegie Mellon University. School of Computer Science». – Режим доступа: <http://www.cs.cmu.edu/~acme/>. – Заголовок с экрана.
69. xADL 3.0. Highly-extensible Architecture Description Language for Software and Systems [Электронный ресурс] // Сайт «Institute for Software Research. University of California. Irvine». – Режим доступа: <http://isr.uci.edu/projects/xarchuci/>. – Заголовок с экрана.
70. Моделирование архитектуры приложения [Электронный ресурс] // Сайт «Microsoft Developer Network». – Режим доступа: <https://msdn.microsoft.com/ru-ru/library/dd490886.aspx>. – Заголовок с экрана.
71. Бек, К. Шаблоны реализации корпоративных приложений [Текст] / К. Бек. – М.: Диалектика-Вильямс, 2008. – 176 с.
72. Иванов, Д. Документирование архитектуры программных систем средствами UML [Электронный ресурс] / Д. Иванов, Ф. Новиков // Режим доступа: http://uml3.ru/library/architecture_via_uml.pdf. – Заголовок с экрана.
73. SSADM V4.2 Structural Standards [Электронный ресурс] / Office of Government Chief Information Officer. – Режим доступа: http://www.ogcio.gov.hk/en/infrastructure/methodology/ssadm/ssadm_42_structural_standards.htm. – Заголовок с экрана.
74. Щелканов, В. Введение в стандарт TOGAF – архитектура предприятия [Электронный ресурс] / В. Щелканов // Сайт «Свободное общество Датавед». – Режим доступа: <http://www.dataved.ru/2014/04/togaf.html>. – Заголовок с экрана.
75. Щелканов, В. TOGAF: основные структурные элементы [Электронный ресурс] / В. Щелканов // Сайт «Свободное общество Датавед». – Режим доступа: http://www.dataved.ru/2014/04/togaf_26.html. – Заголовок с экрана.

76. ГОСТ ИСО/МЭК 10746-1–2004. Информационная технология. Открытая распределенная обработка. Базовая модель. Часть 1. Основные положения [Текст]. – Введ. 04–02–2004. – М.: ИПК Издательство стандартов, 2004. – 81 с.
77. ГОСТ ИСО/МЭК 10746-2–2000. Информационная технология. Взаимосвязь открытых систем. Управление данными и открытая распределенная обработка. Часть 2. Базовая модель [Текст]. – Введ. 26–12–2000. – М.: Стандартинформ, 2006. – 27 с.
78. ГОСТ ИСО/МЭК 10746-3–2001. Информационная технология. Взаимосвязь открытых систем. Управление данными и открытая распределенная обработка. Часть 3. Архитектура [Текст]. – Введ. 20–11–2001. – М.: ИПК Издательство стандартов, 2002. – 57 с.
79. ГОСТ ИСО/МЭК 10746-4–2004. Информационная технология. Открытая распределенная обработка. Базовая модель. Часть 4. Архитектурная семантика [Текст]. – Введ. 04–02–2004. – М.: ИПК Издательство стандартов, 2004. – 34 с.
80. COCOMO II Model Definition Manual [Электронный ресурс] // Сайт «Center for Systems and Software Engineering». – Режим доступа: ftp://ftp.usc.edu/pub/soft_engineering/COCOMOII/cocomo99.0/modelman.pdf. – Заголовок с экрана.
81. На старт! Внимание! И? [Электронный ресурс] // Сайт «ITCua». – Режим доступа: http://itc.ua/articles/na_start_vnimanie_i_21814/. – Заголовок с экрана.
82. Соловьева, Е.И. Подход к повышению точности оценивания трудозатрат на создание IT-проекта [Текст] / Е.И. Соловьева // Радіоелектронні і комп'ютерні системи. – 2014. - № 1 (65). – С. 144-149.
83. Functional Point Counting Practices Manual. Release 4.1.1 [Text]. – Troy: IFPLUG, 2001. – 370 p.
84. Software Engineering. Barry W. Boehm's Lifetime Contributions to Software Development, Management, and Research [Text] / edited by R.W. Selby. – Hoboken: John Wiley & Sons, Inc., 2007. – 832 p.

85. Кульдин, С.П. Генетический подход к проблеме оценки сроков и трудоемкости разработки программного обеспечения с заданными требованиями к качеству [Текст] / С.П. Кульдин // Прикладная информатика. – 2010. – № 5. – С. 30-42.
86. Соловьева, Е. И. Усовершенствование методов оценивания трудозатрат на создание IT-проекта / Е. И. Соловьева // Вестник Национального технического университета «ХПИ»: тематический выпуск: «Новые решения в современных технологиях». – Харьков: НТУ «ХПИ». – 2014. – № 26 (1069). – С. 88-96.
87. Issa, A.A. Reverse engineering Function Points to Object Points [Text] / A.A. Issa, M. Al-Diabat // Applications of Digital Information and Web Technologies, 2009 (ICADIWT'09). Second International Conference of the 4-6 Aug. 2009. – London, 2009. – P. 151-156.
88. Rational Requisite Pro [Электронный ресурс] / Сайт «IBM developerWorks». – Режим доступа: https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Wbcd69e09400c_4f72_9665_66f116225986/page/Rational%20RequisitePro. – Заголовок с экрана.
89. Новичков А. Роль процесса Управления Требованиями при разработке сложных программных систем. Практика применения методологии IBM RUP и инструмента IBM Rational RequisitePro [Электронный ресурс] // Сайт компании СМ Консалт. – Режим доступа: http://cmcons.com/articles/upravlenie_trebovanijami_instrument_ibm_rational_r/rol_protsesta_upravlenija_trebovanijami_pri_razrabotke_slozhnykh_programmnykh_sistem_praktika_primenenija_metodologii_ibm_rup_i_instrumenta_ibm_rational_requisitepro/. – Заголовок с экрана.
90. IBM Rational DOORS Next Generation. An efficient requirements management tool for complex systems [Электронный ресурс] // Режим доступа: http://www-01.ibm.com/common/ssi/cgi-bin/ssialias?subtype=SP&infotype=PM&appname=SWGE_RA_IR_USEN&h

[tmlfid=RAD14128USEN&attachment=RAD14128USEN.PDF](#). – Заголовок с экрана.

91. Cradle Overview [Электронный ресурс] // Сайт компании «3SL». – Режим доступа: <https://www.threesl.com/en/cradle/index.php>. – Заголовок с экрана.
92. Система управления требованиями Devprom Requirements [Электронный ресурс] // Сайт компании DEVPROM. – Режим доступа: <http://devprom.ru/features/Система-управления-требованиями-Devprom-Requirements>. – Заголовок с экрана.
93. Система управления требованиями. Краткое описание [Электронный ресурс] // Сайт «Система управления требованиями». – Режим доступа: <http://www.am-programs.ru/ReqProgrammDsc.shtml>. – Заголовок с экрана.
94. Мадорская, Ю.М. Системы управления требованиями: что и зачем? [Электронный ресурс] / Ю.М. Мадорская // Сайт «ReqCenter.pro». – Режим доступа: <http://edu.reqcenter.pro/?p=2433>. – Заголовок с экрана.
95. Гриценко, В.И. Модель распределенной информационной системы широкого применения / В.И. Гриценко, Е.А. Котиков, А.А. Урсатьев, В.Н. Никулин // Управляющие системы и машины. – 1999. – № 5. – С. 32-42.
96. Гриценко, В.І. Інтелектуальні інформаційні технології: перспективи розвитку та застосування / В.І. Гриценко // НТІ. – 2000. – № 3. – С. 9-10 .
97. Павлов, А.А. Информационные технологии и алгоритмизация в управлении / А.А. Павлов, С.Ф. Теленик. – К.: Техника, 2002. – 344 с.
98. Бакаєв О.О. Мікроекономічне моделювання і інформаційні технології / О.О. Бакаєв, В.І. Гриценко, Л.І. Бажан, Л.О. Бакаєв; Ред.: Т.С. Мельник. – К.: Наукова думка, 2003. – 182 с.
99. Гриценко, В.И. Распределенные информационные системы широкого применения. Концепция. Опыт разработки и внедрения / В.И. Гриценко, А.А. Урсатьев: моногр. – К.: Наук. думка, 2005. – 318 с.

100. Бакаев, А.А. Автоматное моделирование в задачах исследования сложных систем / А.А. Бакаев, В.И. Гриценко, И.С. Сакунова: моногр. – К.: Логос, 2007. — 208 с.
101. Теленик, С.Ф. Концепція, моделі, алгоритми та засоби адаптивної технології створення інформаційно-керуючих систем: автореф. дис. на здобуття наукового ступеня д-ра техн. наук: спец. 05.13.06 „АСУ та прогресивні інформаційні технології” / С.Ф. Теленик. – К., 2000. – 33 с.
102. Левикін, В.М. Модели и информационные технологии мониторинга и оптимизации процесса выращивания монокристаллов полупроводников: монографія [Текст] / Левикін В.М., Шевченко І.В., Оксанич І.Г. – Кременчук: Видавець ПП Щербатих О.В. 2014. – 216 с.
103. Греков, Л.Д. Геораспределенные производственные системы. Часть 2. Размещение на земной поверхности, оптимизация магистральных систем, космический мониторинг: монографія [Текст] / Л.Д. Греков, В.М. Илюшко, О.Е. Федорович. – К.: Издательство Сергея Пантюка, 2014. – 206 с.
104. Казимир В.В. Модельно-орієнтоване управління інтелектуальними виробничими системами: автореф. дис. на здобуття наукового ступеня д-ра техн. наук: спец. 05.13.06 „АСУ та прогресивні інформаційні технології” / В.В. Казимир. – К., 2006. – 35 с.
105. Ткачук М.В. Моделі, методи та інформаційні технології адаптивної розробки та реінжинірингу інформаційно-управляючих систем: автореф. дис. на здобуття наукового ступеня д-ра техн. наук: спец. 05.13.06 „АСУ та прогресивні інформаційні технології” [Текст] / М.В. Ткачук. – Харків, 2006. – 36 с.
106. Гамаюн, І.П. Моделі та методи інформаційної технології адаптивного синтезу оптимальної технологічної схеми складання: автореф. дис. на здобуття наукового ступеня д-ра техн. наук: спец. 05.13.06 „АСУ та прогресивні інформаційні технології” [Текст] / І.П. Гамаюн. - Харків, 2006. – 40 с.

107. Горбенко, А.В. Методологічні основи та інформаційні технології створення гарантоздатних сервіс-орієнтованих Web-систем: автореф. дис. на здобуття наукового ступеня д-ра техн. наук: спец. 05.13.06 „АСУ та прогресивні інформаційні технології” [Текст] / А.В. Горбенко. - Харків, 2012. – 40 с.
108. Vilpola, I. H. A method for improving ERP implementation success by the principles and process of user-centred design [Text] / I. H. Vilpola // Enterprise Information Systems. – 2008. – Vol. 2, № 1. – P. 47–76.
109. Sutcliffe, A. Scenario-based requirements analysis [Text] / A. Sutcliffe // Requirements Engineering. – 1998. – Vol. 3, № 1. – P. 48–65.
110. Mansilla, D. A Proposal of a Process Model for Requirements Elicitation in Information Mining Projects [Text] / D. Mansilla, M. Pollo-Cattaneo, P. Britos, R García-Martínez // Lecture Notes in Business Information Processing. – 2013. – P. 165–173.
111. Липко, Ю. Алгоритм формализации требований при разработке информационных систем [Текст] / Ю. Липко // Известия Южного федерального университета. Технические науки. – 2014. – № 6 (155). – С. 153–158.
112. Yue, T. A systematic review of transformation approaches between user requirements and analysis models [Text] / T. Yue, L. C. Briand, Y. Labiche // Requirements Engineering. – 2010. – Vol. 16, № 2. – P. 75–99.
113. Тюрганов, А. Г. Особенности формализации предметной области корпоративных информационных систем [Текст] / А. Г. Тюрганов // Вестник Уфимского государственного авиационного технического университета. – 2007. – Т. 9, № 5. – С. 72–76.
114. Cleland-Huang, J. Mining Domain Knowledge [Requirements] [Text] / J. Cleland-Huang // IEEE Softw. – 2015. – Vol. 32, № 3. – P. 16–19.
115. Палагин А.В. Архитектура онтологоуправляемых компьютерных систем / А.В. Палагин // Кибернетика и системный анализ. – 2006. – 42, № 2. – С. 111-124.

116. Палагін О.В. Архітектурно-онтологічні принципи розбудови інтелектуальних інформаційних систем / О.В. Палагін, М.Г. Петренко // Математичні машини і системи. – 2006. – № 4. – С. 15-20.
117. Палагін О.В. Розбудова абстрактної моделі мовно-онтологічної інформаційної системи / О.В. Палагін, М.Г. Петренко // Математичні машини і системи. – 2007. – № 1. – С. 42-50.
118. Евланов, М.В. Глобальные цели поставщика и потребителя ИТ-услуг / М.В. Евланов, О.Е. Неумывакина, А.Ю. Карамышева // Восточно-европейский журнал передовых технологий. – 2012. – № 5/2 (59). – С. 12-17.
119. Евланов М.В. Онтологическая модель архитектуры информационной системы на основе сервисного подхода [Текст] / М.В. Евланов // Радиоелектроніка, інформатика, управління. – 2013. - № 2. – С. 130-135.
120. Деревянко, А.С. Технологии и средства консолидации информации / А.С. Деревянко, М.Н. Солощук. – Харьков: НТУ «ХПИ», 2008. – 432 с.
121. Проектирование и внедрение АСУП / под общ. ред. В.М. Глушкова. – К.: Техніка, 1974. – 191 с.
122. Справочник разработчика АСУ / под ред. Н.П. Федоренко, В.В. Карибского. – М.: Экономика, 1978. – 583 с.
123. Мамиконов, А.Г. Проектирование АСУ / А.Г. Мамиконов. – М.: Высшая школа, 1987. – 303 с.
124. Основы системного анализа и проектирования АСУ / Под общ. ред. А.А. Павлова. – К.: Выща школа, 1991. – 367 с.
125. ГОСТ 34.601–90. Автоматизированные системы. Стадии создания. – Введ. 01.01.1992. – М. : Изд-во стандартов, 1997. – 10 с.
126. Горобец, Н. ISO 20000: зрелое управление ИТ-услугами / Н. Горобец // Директор информационной службы. – 2006. – № 9. – С. 62–67.
127. ГОСТ Р ИСО/МЭК 20000–200X (проект, окончательная редакция). Управление услугами. Ч. 1. Спецификация [Электронный ресурс]. – М., 2009. – Режим доступа:

http://itsmforum.ru/reference/ISO20000/GOST_R_ISO_MEK_20000-1.pdf.

– Заголовок с экрана.

128. Евланов, М.В. Определение понятия «требование к информационной системе» / М.В. Евланов // Вісник Академії митної служби України. Серія «Технічні науки». – 2012. - № 2. – С. 71-77.
129. Евланов, М.В. Концепция представления требований к информационной системе [Текст] / М.В. Евланов // Вісник національного технічного університету «ХПІ». Збірник наукових праць. Серія «Нові рішення в сучасних технологіях». – 2012. - № 68 (974). – С. 32-40.
130. Философский энциклопедический словарь / редкол.: Л.Ф. Ильичёв, П.Н. Федосеев, С.М. Ковалёв, В.Г. Панов. – М.: Советская энциклопедия, 1983. – 840 с.
131. Фаулер, М. UML в кратком изложении. Применение стандартного языка объектного моделирования / М. Фаулер, К. Скотт. – М.: Мир, 1999. – 191 с.
132. Bellinger, G. Data, Information, Knowledge and Wisdom [Электронный ресурс] / Gene Bellinger, Durval Castro, Anthony Mills. – Режим доступа: <http://www.systems-thinking.org/dikw/dikw.htm>. – Заголовок с экрана.
133. Davis, Alan M. Principles of Software Development / Alan M. Davis. – New York: McGraw-Hill, 1995. – 240 p.
134. Левыкин В.М. Паттерны проектирования требований к информационной системе: моделирование и применение [Текст] / В.М. Левыкин, М.В. Евланов, М.А. Керносов: монография. – Харьков: ООО «Компанія СМІТ», 2014. – 320 с.
135. Васильцова, Н.В. Разработка метамодели требований к информационной системе / Н.В. Васильцова, М.В. Евланов, И.Ю. Панферова // АСУ и приборы автоматики. – 2004. – Вып. 129. – С. 19–27.
136. Калянов, Г.Н. CASE: структурный системный анализ / Г.Н. Калянов. – М.: Лори, 1996. – 242 с.

137. Вендров, А.М. CASE-технологии. Современные методы и средства проектирования информационных систем / А.М. Вендров. – М.: Финансы и статистика, 1998. – 177 с.
138. Репин, В.В. Процессный подход к управлению. Моделирование бизнес-процессов / В.В. Репин, В.Г. Елиферов. – М.: Стандарты и качество, 2004. – 408 с.
139. Рубцов, С.В. Уточнение понятия «бизнес-процесс» / С.В. Рубцов // Менеджмент в России и за рубежом. – 2001. – № 6. – С. 26–33.
140. Евланов, М.В. Концепция представления требований к информационной системе / М.В. Евланов // Информационные системы и технологии: материалы Междунар. науч.-техн. конф., Морское-Харьков, 22-29 сентября 2012 г.: тезисы докладов / редкол.: А.Д. Тевяшев (отв. ред.) и др. – Харьков: НТМТ, 2012. – С. 34.
141. Цаленко, М.Ш. Основы теории категорий / М.Ш. Цаленко, Е.Г. Шульгейфер. – М.: Наука, 1974. – 256 с.
142. Фейс, К. Алгебра: кольца, модули и категории. В 2-х т. – Т. 1. / К. Фейс. – М.: Мир, 1977. – 688 с.
143. Букур, И. Введение в теорию категорий и функторов / И. Букур, А. Деляну. – М.: Мир, 1978. – 259 с.
144. Комраков Б.П. Структуры на многообразиях и однородные пространства / Б.П. Комраков. – Минск: Наука и техника, 1978. – 352 с.
145. Евланов, М.В. Модель методологии предпроектного обследования объекта автоматизации / М.В. Евланов // Збірник тез доповідей науково-практичної конференції «Застосування інформаційних технологій у підготовці та діяльності сил охорони правопорядку». – Харків: Академія Внутрішніх військ МВС України, 2013. – С. 76-78.
146. Левыкин В.М. Задача определения функторов между категорными моделями информационной системы / В.М. Левыкин. М.В. Евланов // Проблемы бионики. – 2003. – Вып. 58. – С. 62-67.

147. Левыкин В.М. Концепция построения CASE-системы разработки информационных управляющих систем / В.М. Левыкин. М.В. Евланов, Мухайрат Мохаммад // АСУ и приборы автоматики. – 2001. – Вып. 114. – С. 55-59.
148. Евланов М.В. Формализация взаимных отображений моделей информационных систем / М.В. Евланов // Materialy IV Miedzynarodowej naukowii-praktycznej konferencji „Nowoczesnych naukowych osiagniec - 2008”. – Тум 13. Matematyka. Fizyka. Nowoczesne informacyjne technologie. – Przemysl: Nauka i studia. – Str. 82-85.
149. Маклаков, С.В. Создание информационных систем с AllFusion Modeling Suite / С.В. Маклаков. – М.: ДИАЛОГ-МИФИ, 2003. – 422 с.
150. Гамма, Э. Приемы объектно-ориентированного проектирования. Паттерны проектирования / Э. Гамма, Р. Хелм, Р. Джонсон, Д. Влоссидес. – СПб.: Питер, 2010. – 366 с.
151. Фримен, Э. Паттерны проектирования / Э. Фримен, Э. Фримен, К. Сьерра, Б. Бейтс. – СПб.: Питер, 2011. – 656 с.
152. Кериевски, Дж. Рефакторинг с использованием шаблонов / Дж. Кериевски. – М.: Издательский дом «Вильямс», 2006. – 400 с.
153. Эмблер, С.В. Рефакторинг баз данных. Эволюционное проектирование / С.В. Эмблер, П.Дж. Садаладж. – М.: Издательский дом «Вильямс», 2007. – 368 с.
154. Фаулер, М. Рефакторинг. Улучшение существующего кода / М. Фаулер. – М.: Символ-Плюс, 2008. – 432 с.
155. Refactoring [Электронный вариант] / Сайт «SourceMaking». – Режим доступа: <http://sourcemaking.com/refactoring>. – Заголовок с экрана.
156. Larman, C. Applying UML and Patterns. An Introduction to Object-Oriented Analysis and Design and Iterative Development. Third Edition / C. Larman. – 2004, Prentice Hall PTR. – 736 p.
157. Левыкин, В.М. Подход к использованию паттернов проектирования при работе с требованиями к информационной системе / В.М. Левыкин,

- М.В. Евланов, М.А. Керносков // Системний аналіз. Інформатика. Управління (САГУ-2013): матеріали IV Міжнародної науково-практичної конференції (м. Запоріжжя, 13-16 березня 2013 р.). – Запоріжжя: КПУ, 2013. –С. 150-152.
158. Евланов М.В. Паттерны проектирования требований к информационной системе [Текст] / М.В. Евланов, О.Е. Неумывакина // Информационные системы и технологии: материалы 2-й Международ. науч.-техн. конф., Евпатория-Харьков, 16-22 сентября 2013: тезисы докладов. – Харьков: НТМТ, 2013. – С. 32-33.
159. Євланов М.В. Паттерни проектування вимог до інформаційної системи [Текст] / М.В. Євланов // Вісник національного університету «Львівська політехніка». -2014. – № 783. – С. 429-434.
160. Левыкин В.М. Модель архитектурного фреймворка ускоренной разработки информационной системы [Текст] / В.М. Левыкин. М.В. Евланов // Науковий вісник Кременчуцького університету економіки, інформаційних технологій і управління «Нові технології». – 2013. - № 1-2 (39-40). – С. 51-57.
161. Левыкин В.М. Архитектурный фреймворк ускоренной разработки информационной системы [Текст] / В.М. Левыкин, М.В. Евланов // Научна конференция с международно участие «Хранителна наука, техника и технологии 2013». Пловдив 18-19 октомври 2013. – Университет по хранителни технологии – Пловдив. Научни трудове. – 2013. – Т. LX, св.2. – С. 230-234.
162. Месарович, М. Общая теория систем: математические основы [Текст] / М. Месарович, Я.Такахара.: пер. с англ. – М.: Мир, 1978. – 312 с.
163. Лачинов, В.М. Информодинамика или Путь к Миру открытых систем [Текст] / В.М. Лачинов, А.О. Поляков. – СПб.: Издательство СПбГТУ, 1999. – 122 с.
164. Поляков, А.О. Информодинамическая общность систем [Электронный ресурс] / А.О. Поляков // Сайт «Теория информации». – Режим

доступа: <http://inftech.webservis.ru/it/information/index.html>. – Заголовок с экрана.

165. Евланов М.В. Модели паттернов проектирования требований к информационной системе на уровне данных [Текст] / М.В. Евланов // Радиоэлектронні і комп'ютерні системи. – 2014. - № 1 (65). – С. 128-138.
166. Левыкин, В.М. Параллельное проектирование информационного и программного комплексов информационной системы / В.М. Левыкин. М.В. Евланов, В.С. Сугробов [Текст] // Радиотехника. – 2006. – Вып. 146. – С. 89-98.
167. Евланов, М.В. Представление требований к информационной системе на уровне информации [Текст] / М.В. Евланов, О.Е. Неумывакина // Информационные системы и технологии: материалы 3-й Международ. науч.-техн. конф., Харьков, 15-21 сентября 2014 г.: тезисы докладов. - Харьков: ХНУРЭ, 2014. – С. 37-38.
168. Ievlanov, M.V. Concept of a information system element requirement representation [Text] / M.V. Ievlanov, N.V. Vasiltsova, I.Ju. Panforova // First International forum “IT-Trends: big data, artificial intelligence, social media”: Book of abstracts. – Kremenchuk: Kremenchuk Mykhailo Ostrohradskiy National University, 2014. – pp. 14-16.
169. Евланов, М.В. Концепция представления требования к элементу информационной системы [Текст] / М.В. Евланов, Н.В. Васильцова, И.Ю. Панферова // Вісник Кременчуцького національного університету ім. Михайла Остроградського. – 2014. – Випуск 5/2014 (88). – С. 34-41.
170. Левыкин, В.М. Подход к формализации требований к информационной системе [Текст] / В.М. Левыкин. М.В. Евланов, М.Ю. Хрипкова // Тез. докл. Междунар. конф. «Теория и техника передачи, приема и обработки информации». – Харьков: ХНУРЭ, 2003. – С. 242-243.
171. Lassila O. Frames or Objects, or Both? [Текст] / O. Lassila // Workshop Notes from the Eight National Conference on Artificial Intelligence (AAAI-90): Object-Oriented Programming in AI, Boston (Massachusetts, U.S.A.), 1990. – 8 p.

172. Wu X. A Comparison of Objects with Frames and OODBs [Текст] / Wu X. // Object Currents, 1996. – Vol 1.– No 1. – 9 p.
173. Минский, М. Фреймы для представления знаний [Текст] / М. Минский. – М.: Энергия, 1979. –152 с.
174. Искусственный интеллект: в 3-х кн. Кн. 2. Модели и методы: Справочник / Под ред. Д.А. Поспелова – М.: Радио и связь, 1990. – 304 с.
175. Гаврилов, А.В. Системы искусственного интеллекта / А.В. Гаврилов. – Новосибирск: НГТУ, 2004. – 59 с.
176. Savitch, W. Java: An Introduction to Computer Science and Programming / W. Savitch: 2d ed. – Pearson: Prentice Hall, Inc, 2001 – 1039 p.
177. Deitel, H.M. C++ How to Program / H.M. Deitel, P.J. Deitel: 5th ed. – Pearson: Prentice Hall, Inc, 2005 – 1536 p.
178. Левыкин, В. М. Исследование и разработка фреймовой модели структуры документа / В. М. Левыкин, М. А. Керносов // Нові технології. – 2008. – № 1 (19). – С. 149–154.
179. Фленов, М.Е. Библия Delphi / М.Е. Фленов. – СПб.: БХВ-Петербург, 2004. – 880 с.
180. Евланов, М.В. Разработка методов формирования представления сформулированного требования к информационной системе на уровне знаний [Текст] / М.В. Евланов // Восточно-европейский журнал передовых технологий. – 2015. - № 4. – С. 4-11.
181. Євланов М.В. Задача синтезу опису архітектури інформаційної системи [Текст] / М.В. Євланов // Вісник національного університету «Львівська політехніка». – 2014. – № 805. – С. 114-123.
182. Евланов М.В. Задачи проектирования архитектуры информационной системы учета и обработки результатов физических исследований [Текст] / М.В. Евланов // Тезисы докладов XII конференции по физике высоких энергий, ядерной физике и ускорителям 17-21 марта 2014 г., Харьков. – Харьков: ННЦ ХФТИ, 2014. –С. 48.

183. Евланов, М.В. Метод формирования представления требований к информационной системе на уровне информации [Текст] / М.В. Евланов, Н.В. Васильцова, О.Е. Неумывакина, И.Ю. Панферова // Інформаційні управляючі системи та технології (ІУСТ-Одеса-2015): Матеріали IV Міжнародної науково-практичної конференції. – Одеса: «ВидавІнформ» ОНМА, 2015. – С. 231-234.
184. Євланов, М.В. Моделі і методи синтезу опису раціональної архітектури інформаційної системи [Текст] / М.В. Євланов, Н.В. Васильцова, І.Ю. Панфьорова // Вісник наукового університету «Львівська політехніка». Серія «Інформаційні системи та мережі». – 2015. - № 829. – С. 135-152.
185. Евланов, М.В. Разработка модели и метода выбора описания рациональной архитектуры информационной системы [Текст] / М.В. Евланов // Восточно-европейский журнал передовых технологий. – 2016. - № 1/2(79) – С. 4-12.
186. Yang , Y. CLOPE: A fast and Effective Clustering Algorithm for Transaction Data [Текст] / Y. Yang, H. Guan, J. You // Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. – New York: ACM, 2002. – P. 682-687.
187. Паклин, Н. Кластеризация категориальных данных: масштабируемый алгоритм CLOPE [Электронный ресурс] // Сайт BaseGroupe Labs. – Режим доступа: <http://basegroup.ru/community/articles/clope>. – Заголовок с экрана.
188. Левыкин, В.М. Метамодел ь функциональной структуры информационной системы [Текст] / В.М. Левыкин, М.В. Евланов // Нові технології. Науковий вісник Кременчуцького університету економіки, інформаційних технологій і управління. – 2006. – № 1(11). – С. 67-72.
189. Левыкин, В.М. Модель синтеза описания рациональной архитектуры информационной системы [Текст] / В.М. Левыкин, М.В. Евланов // Інформаційні управляючі системи та технології (ІУСТ-Одеса-2015): Матеріали IV Міжнародної науково-практичної конференції. – Одеса: «ВидавІнформ» ОНМА, 2015. – С. 209-212.

190. Євланов, М.В. Модель синтезу опису раціональної архітектури інформаційної системи [Текст] / М.В. Євланов, Н.В. Васильцова, І.Ю. Панфьорова // Информационные системы и технологии: материалы 4-й Международной науч.-техн. конф., Харьков, 21-27 сентября 2015 г.: тезисы докладов / [редкол.: А.Д. Тевяшев (отв. ред.) и др.]. – Харьков: НТМТ, 2015 – С. 50-51.
191. Левикін, В.М. Підхід до моделювання прийняття рішень при управлінні передпроектними стадіями створення інформаційної системи [Текст] / В.М. Левикін, М.В. Євланов, В.О. Антонов // Вісник Харківського національного технічного університету сільського господарства імені Петра Василенка. – 2006. – Вип. 43 «Проблеми енергозабезпечення та енергозбереження в АПК України». – Том 2. – С. 177-181.
192. Евланов М.В. Теоретико-игровая модель взаимоотношений «поставщик-потребитель информационной системы» / М.В. Евланов, Е.Ю. Альбова // Materialy IX mezinarodni vedecko – prakticka konference «Aplikovane vedecke novinky - 2013». Dil 13. Moderni informacni technologie. Vystavba a architektura. – Praha: Publishing House «Education and Science». – S. 20-22.
193. Ларіонов, Ю.І. Дослідження операцій в інформаційних системах [Текст] / Ю.І. Ларіонов, В.М. Левикін, М.А. Хажмурадов. – Харків: Компанія СМІТ, 2005. – 364 с.
194. Губко, М.В. Теория игр в управлении организационными системами [Текст] / М.В. Губко, Д.А. Новиков. – М.: 2005. – 138 с.
195. Матвеев, В.А. Конечные бескоалиционные игры и равновесия [Текст] / В.А. Матвеев. – Псков: ПГПИ им. С.М.Кирова, 2005. – 176 с.
196. Барсегян А.А. Технологии анализа данных: Data Mining, Visual Mining, Text Mining, OLAP [Текст] / А.А. Барсегян, М.С. Куприянов, В.В. Степаненко, И.И. Холод. – СПб.: БХВ-Санкт-Петербург, 2008. – 384 с.
197. Евланов, М.В. Унификация методов оценивания затрат на создание современных информационных систем [Текст] / М.В. Евланов, Е.И.

- Соловьева // Вісник Кременчуцького національного університету ім. Михайла Остроградського. – 2014. – Випуск 5/2014 (88). – С. 62-67.
198. Ievlanov M. Unification of methods for estimating the cost of creating modern information systems [Text] / M. Ievlanov, K. Solovyova // First International forum “IT-Trends: big data, artificial intelligence, social media”: Book of abstracts. – Kremenchuk: Kremenchuk Mykhailo Ostrohradskiy National University, 2014. – pp. 16-17.
 199. Евланов, М.В. Оценка объема работ по реализации описания архитектуры информационной системы [Текст] / М.В. Евланов // Materialy XI mazarodni vedecko – prakticka konference «Predni vedecke novinky – 2015». – Dil 5. Technicke vedy. Fyzika. Moderni informacni technologie.: Praha. Publishing House «Education and Science». – p. 87-94.
 200. COCOMO II Model Definition Manual [Электронный ресурс] // Сайт «Center for Systems and Software Engineering». – Режим доступа: ftp://ftp.usc.edu/pub/soft_engineering/COCOMOII/cocomo99.0/modelman.pdf – Заголовок с экрана.
 201. Левыкин, В.М. Модификация метода объектных точек для оценивания объема трудозатрат на создание информационной системы [Текст] / В.М. Левыкин, М.В. Евланов, М.А. Керносов // Информационные системы и технологии: материалы 3-й Международ. науч.-техн. конф., Харьков, 15-21 сентября 2014 г.: тезисы докладов. - Харьков: ХНУРЭ, 2014. – С. 112-113.
 202. Евланов, М.В. Методы анализа сформулированных функциональных требований к информационной системе [Текст] / М.В. Евланов, О.Е. Неумывакина // Информационные системы и технологии: материалы 4-й Международной науч.-техн. конф., Харьков, 21-27 сентября 2015 г.: тезисы докладов / [редкол.: А.Д. Тевяшев (отв. ред.) и др.]. – Харьков: НТМТ, 2015 – С. 52-53.
 203. Левыкин, В.М. Формирование структуры работ IT-проекта создания информационной системы [Текст] / В.М. Левыкин, М.В. Евланов // Вісник Національного технічного університету «ХПІ». Збірник наукових праць.

- Серія: Стратегічне управління, управління портфелями, програмами та проектами. – Харків: НТУ «ХП», 2014. - № 2 (1045). – С. 81-86.
204. Евланов, М.В. Информационная технология ускоренной разработки информационных систем [Текст] / М.В. Евланов, М.А. Керносов, М.Э. Керносова // Управляющие системы и машины. – 2014. - № 1 (249). – С. 62-69.
205. Левыкин В.М. ИТ-услуги информационной технологии управления проектами ускоренной разработки информационных систем [Текст] / В.М. Левыкин, М.В. Евланов // Информационные системы и технологии: материалы 2-й Международ. науч.-техн. конф., Евпатория-Харьков, 16-22 сентября 2013: тезисы докладов. – Харьков: НТМТ, 2013. – С. 53-55.
206. Евланов М.В. Информационная технология ускоренной разработки информационных систем [Текст] / М.В. Евланов, М.А. Керносов // Конкурс инновационных проектов в сфере информационно-коммуникационных технологий «IT-Kharkiv». Избранные проекты. – Харьков: Научный парк «Радиоэлектроника и информатика», 2013. – С. 45-46.
207. Евланов, М.В. Технология быстрого проектирования информационных систем [Текст] / М.В. Евланов, М.А. Керносов, М.Э. Лотфулина // Информационные системы и технологии: материалы Междунар. науч.-техн. конф., Морское-Харьков, 22-29 сентября 2012 г.: тезисы докладов / [редкол.: А.Д. Тевяшев (отв. ред.) и др.]. - Харьков: НТМТ, 2012. - С. 35.
208. Абдикеев, Н.М. Управление знаниями корпорации и реинжиниринг бизнеса [Текст] / Н.М. Абдикеев, А.Д. Киселев. – М.: ИНФРА-М, 2010. – 384 с.
209. Євланов, М.В. Визначення лексикона візуального моделювання інформаційних систем / М.В. Євланов // Науковий вісник Інституту економіки та нових технологій „Нові технології”. –2004. – № 1-2 (4-5). – С. 204-208.
210. Евланов, М.В. Разработка технологии синтеза информационной системы управления предприятием [Текст] / М.В. Евланов, М.А. Керносов // Materialy

- VI Międzynarodowej naukowo-praktycznej konferencji “Dynamika naukowych badań - 2010”. Volume 10. Techniczne nauki. Nowoczesne informacyjne technologie. Matematyka. Fizyczna kultura i sport. – Przemysł, Nauka i studia, 2010. – С. 76-79.
211. Левыкин, В.М. Технология управления функциональными требованиями в рамках жизненного цикла информационной системы [Текст] / В.М. Левыкин, М.В. Евланов // Информационные системы и технологии: материалы 4-й Международной науч.-техн. конф., Харьков, 21-27 сентября 2015 г.: тезисы докладов / [редкол.: А.Д. Тевяшев (отв. ред.) и др.]. – Харьков: НТМТ, 2015 – С. 78-79.
212. Евланов М.В. Теоретико-категорная модель IT-сервиса информационной системы управления предприятием [Текст] / М.В. Евланов, В.А. Никитюк, О.И. Калмыкова // VI-я Международная научно-практическая конференция «Наука и социальные проблемы общества: информатизация и информационные технологии». Сборник научных трудов. – Харьков: ХНУРЭ, 2011. – С. 321-322.
213. Евланов М.В. Обобщенная модель фреймворка функционального сервиса информационной системы [Текст] / М.В. Евланов, В.А. Никитюк // Системи обробки інформації. – 2013. – Вип. 1(108). – С. 194-199.
214. Евланов М.В. Синтез схемы данных информационной системы на основе онтологического описания предметной области [Текст] / М.В. Евланов, М. Э. Керносова // Проблемы информационных технологий – 2014. – № 1 (5). – С. 124–131.
215. Levykin V.M. Features of domain ontology into the elements of information system mapping [Text] / V.M. Levykin, M.V. Ievlanov, M.A. Kernosov, M.E. Kernosova // First International forum “IT-Trends: big data, artificial intelligence, social media”: Book of abstracts. – Kremenchuk: Kremenchuk Mykhailo Ostrohradskiy National University, 2014. – pp. 30-32.
216. Керносов М.А. Подход к администрированию привилегий пользователей информационной системы организационного управления территориально-распределённой проектной организацией [Текст] / М.А. Керносов,

М.Э. Лотфулина // Застосування інформаційних технологій у підготовці та діяльності сил охорони правопорядку : науково-практична конференція, 20-21 березня 2013 р. : зб. тез. доп. – Харків. – 2013. – С.86-88.

217. Керносов М.А. Реализация проекта информационной системы с применением онтологической модели представления требований к системе [Текст] / М.А. Керносов, М.Э. Керносова // Информационные системы и технологии: материалы 2-й Международ. науч.-техн. конф., Евпатория-Харьков, 16-22 сентября 2013 г.: тезисы докладов / [редкол.: А.Д. Тевяшев (отв. ред.) и др.]. – Х.: НТМТ, 2013. – С. 44-45.
218. Социальные проекты [Электронный ресурс] // Сайт региональной газовой компании «Хакривгаз». – Режим доступа: <http://kh.104.ua/ru/informacija-pro-kompaniju/sustainability/id/socialni-proekti-870#sub880>. – Заголовок с экрана.
219. Евланов. М.В. Формирование и анализ требований к информационно-аналитической системе управления безопасностью труда на предприятии [Текст] / М.В. Евланов, Н.Н. Сердюк // Технологический аудит и резервы производства. – 2015. – № 4/3(24). – С. 41-45.
220. Автоматизированная система «Труд-Эксперт» v.4.0 for Windows [Электронный ресурс] / Клинский институт условий и охраны труда.— Режим доступа: <http://www.kiout.ru> / — Заголовок с экрана.
221. ДСТУ ОHSAS 18001:2010 Система управління гігієною та безпекою праці. [Электронный ресурс]. – Режим доступа: http://www.dnaop.com/html/34112/doc.%D0%94%D0%A1%D0%A2%D0%A3_OHSAS_18001_2010.— Заголовок с экрана.
222. Евланов, М. В. Модели и метод определения состояния организма сотрудника предприятия [Текст] / М. В. Евланов, Н. Н. Сердюк // Вісник Національного технічного університету «ХПІ». – 2015. - № 21(1130). – С. 163-169.

ПРИЛОЖЕНИЕ А
АКТЫ О ВНЕДРЕНИИ РЕЗУЛЬТАТОВ ДИССЕРТАЦИИ

АКТ

впровадження результатів дисертаційної роботи
Євланова Максима Вікторовича на здобуття вченого ступеня доктора
технічних наук за фахом 05.13.06 - інформаційні технології

1. Даним актом підтверджуємо, що наукові і практичні результати дисертаційної роботи Євланова Максима Вікторовича, а саме:

- вдосконалений сервісний підхід до створення інформаційних систем,
- теоретико-категорну модель інтелектуальної інформаційної технології прискореної розробки інформаційних систем,
- комплекс моделей, що описують функціональні вимоги до інформаційної системи на рівнях даних, інформації і знань,
- методи формування представлення функціональних вимог до інформаційної системи на рівнях інформації і знань,
- модель процесу синтезу раціонального опису архітектури інформаційної системи,
- метод автоматизованого формування раціонального опису ІТ-послуг інформаційної системи,
- знання-орієнтовану модель інформаційної системи як сукупності ІТ-Послуг,
- інтелектуальна інформаційна технологія прискореної розробки інформаційних систем

були використані в процесі створення функціонального модуля безпеки праці комплексної системи управління ПАТ «Харківміськгаз».

2. Використання зазначених вище наукових та практичних результатів дисертаційної роботи Євланова Максима Вікторовича дозволило підвищити якість результатів формування та аналізу вимог до комплексної системи управління ПАТ «Харківміськгаз», виділити процеси, які оброблюють дубльовану інформацію, та запропонувати шляхи їхнього усунення.

3. Впровадження запропонованих Євлановим Максимом Вікторовичем наукових та практичних результатів дозволило отримати економічний ефект у розмірі 56 тис. грн.

Цей акт виданий без фінансових зобов'язань ПАТ «Харківміськгаз» перед автором дисертаційної роботи.

Перший заступник Голови Правління –
Заступник з постачання та обліку газу

16.06.2015 р.



А.О. Яворовський

УТВЕРЖДАЮ

Первый зам. главного конструктора
ГНПП «Объединение Коммунар»
Главный инженер НТ СКБ «ПОЛИСВИТ»
Кандидат технических наук, доцент
Заслуженный изобретатель Украины



 Н.Ф. Сидоренко

«27» новоїя 2015 г.

АКТ ВНЕДРЕНИЯ

результатов диссертационной работы Евланова Максима Викторовича, представленной на соискание ученой степени доктора технических наук по специальности 05.13.06 – информационные технологии

Комиссия в составе начальника отдела Ковинько Н.М. и заместителя начальника отдела Мариненко С.А. провела экспертизу и составила настоящий акт о том, что результаты диссертации Евланова М.В. внедрены при модернизации информационной системы НТ СКБ «ПОЛИСВИТ».

В ходе проведения проектных работ по модернизации информационной системы НТ СКБ «ПОЛИСВИТ» использовались разработанные в диссертации:

- а) математические модели сформулированных требований к информационной системе;
- б) комплекс моделей паттернов проектирования требований к информационной системе на уровнях информации, данных и знаний;
- в) методы формирования представлений функциональных требований к информационной системе на уровне информации и знаний;
- г) методы синтеза описания архитектуры информационной системы и выбора из множества вариантов рационального описания архитектуры модернизируемой информационной системы;
- д) методы анализа сформулированных функциональных требований на уровнях информации и знаний в ходе модернизации информационной системы.

Существенными положительными особенностями предложенных в диссертации подхода, моделей и методов является возможность формального выделения и обработки знаний об автоматизируемых процессах ИТ СКБ «ПОЛИСВИТ» как объекте автоматизации с целью проектирования описания архитектуры информационной системы, максимально удовлетворяющей сформулированным функциональным требованиям, а также возможность повторного использования накопленного ранее опыта разработки и модернизации информационных систем, реализованные в виде совокупности функциональных решений по адаптации информационной системы к требованиям ИТ СКБ «ПОЛИСВИТ» и расширению функциональных возможностей информационной системы.

Внедрение указанных выше подхода, моделей и методов позволило сократить время и стоимость выполнения работ по формированию и анализу требований к модернизируемой информационной системе и синтезу описания архитектуры информационной системы ИТ СКБ «ПОЛИСВИТ». Эффект от внедрения результатов диссертации Евланова М.В. составляет 68 тыс. грн.

Этот акт выдан без финансовых обязательств ГНПП «Объединение Коммунар» и его подразделения ИТ СКБ «ПОЛИСВИТ» перед автором диссертации.

Начальник отдела ИТ СКБ «ПОЛИСВИТ»  Н.М. Ковинько

Заместитель начальника отдела
ИТ СКБ «ПОЛИСВИТ»  С.А. Мариненко

«ЗАТВЕРДЖУЮ»

Генеральний

директор

ТОВ «ПрофІТсофт», канд. техн. наук



О.В. Петриченко

АКТ ВПРОВАДЖЕННЯ

результатів дисертаційної роботи Євланова Максима Вікторовича,
яку подано на здобуття наукового ступеня доктора технічних наук
за спеціальністю 05.13.06 – інформаційні технології

м. Харків

«15» грудня 2015 р.

Комісія у складі:

Петриченко Олександр Вячеславович - голова комісії;

Мізерник Андрій Миколайович та

Петренко Андрій Олександрович - члени комісії

провела експертизу і склала цей акт про те, що результати дисертації Євланова М.В. впроваджено при розробці комплексної системи автоматизації страхової компанії (комп'ютерні програми «ProfITsoft FRONT-OFFICE» та «ProfITsoft BACK-OFFICE»).

Під час проведення робіт зі створення комплексної системи автоматизації страхової компанії використовувались розроблені в дисертації:

а) вдосконалений сервісний підхід до створення інформаційних систем;
б) математичні моделі сформульованих вимог до інформаційної системи;

в) комплекс моделей паттернів проектування вимог до інформаційної системи на рівнях інформації та знань;

г) методи формування представлень функціональних вимог до інформаційної системи на рівні інформації та знань;

д) методи синтезу опису архітектури інформаційної системи та вибору раціонального опису архітектури створюваної інформаційної системи з множини синтезованих варіантів;

е) методи аналізу сформульованих функціональних вимог на рівнях інформації та знань під час створення інформаційної системи.

Суттєвими перевагами запропонованих в дисертації підходу, моделей і методів є:

- формалізація обробки знань про предметну галузь створюваної системи;

- підвищення точності оцінювання обсягу робіт зі створення функцій інформаційної системи;

- спрощення прийняття рішень про повторне використання наявних результатів розробки попередніх ІТ-проектів під час створення інформаційної системи.

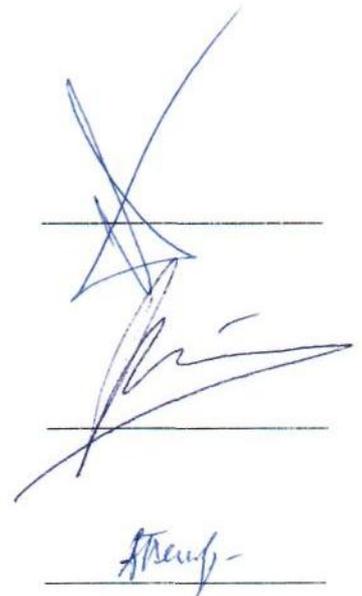
Впровадження зазначених вище підходу, моделей і методів дозволило скоротити вартість виконання робіт з формування та аналізу вимог та синтезу архітектури створюваної комплексної системи автоматизації страхової компанії.

Цей акт видано без фінансових зобов'язань ТОВ «ПрофІТсофт» перед автором дисертації.

Голова комісії,
Петриченко Олександр Вячеславович

Члени комісії:
Мізерник Андрій Миколайович

Петренко Андрій Олександрович



The image shows three handwritten signatures in blue ink, each placed above a horizontal line. The top signature is a large, stylized 'X' shape. The middle signature is a cursive signature that appears to be 'Mizer'. The bottom signature is a cursive signature that appears to be 'Andriy'.

«УТВЕРЖДАЮ»

Генеральный директор

ООО ТД «Технодар»



А.В. Якушев

АКТ ВНЕДРЕНИЯ

результатов диссертационной работы Евланова Максима Викторовича,
представленной на соискание ученой степени доктора технических наук
по специальности 05.13.06 – информационные технологии

г. Петрозаводск

«24» 01 2016 г.

Комиссия в составе технического директора Спиридонова А.Ю. и руководителя отдела информационных технологий Керносова М.А. провела экспертизу и составила настоящий акт о том, что результаты диссертации Евланова М.В. внедрены при разработке информационной системы дистанционного мониторинга транспорта.

В ходе проведения проектных работ по разработке информационной системы использовались разработанные в диссертации:

а) методы формирования представлений функциональных требований к информационной системе на уровне информации и знаний;

б) методы синтеза описания архитектуры информационной системы и выбора из множества вариантов рационального описания архитектуры разрабатываемой информационной системы;

в) методы анализа сформулированных функциональных требований на уровнях информации и знаний.

Основными преимуществами предложенных в диссертации моделей и методов является:

а) возможность формального выделения и обработки знаний об объекте автоматизации и создаваемой системе с целью проектирования описания архитектуры данной системы, максимально удовлетворяющей сформулированным функциональным требованиям;

б) возможность повторного использования накопленного ранее опыта разработки информационной системы дистанционного мониторинга транспорта, реализованного в виде совокупности функциональных требований к системе и соответствующих данным требованиям технических решений по разработке обеспечивающей части информационной системы.

Внедрение указанных выше моделей и методов позволило сократить время и стоимость выполнения работ по формированию и анализу требований к разрабатываемой информационной системе и синтезу описания архитектуры информационной системы дистанционного мониторинга транспорта. Эффект от внедрения результатов диссертации Евланова М.В. составляет 25 тыс. евро.

Этот акт выдан без финансовых обязательств ООО ТД «Технодар» перед автором диссертации. ООО ТД «Технодар» является владельцем эксклюзивных прав на разработанную систему мониторинга.

Технический директор



А. Ю. Спиридонов

Руководитель отдела ИТ,
к.т.н., доц.



М.А. Керносов

«ЗАТВЕРДЖУЮ»

Перший проректор Харківського
національного університету



радіоелектроніки,

к.т.н., проф. І.І. Ключник

_____ 2016 р.

АКТ

про впровадження в навчальний процес результатів дисертаційної роботи
професора кафедри інформаційних управляючих систем
Євланова Максима Вікторовича

Комісія у складі:

Голова: докт. техн. наук, професор, декан факультету комп'ютерних наук Єрохін А.Л.

члени комісії:

докт. техн. наук, професор, професор кафедри
інформаційних управляючих систем Чалий С.Ф.;

канд. техн. наук, доцент, доцент кафедри інформаційних
управляючих систем Панфьорова І.Ю.;

канд. техн. наук, доцент, доцент кафедри інформаційних
управляючих систем Міхнова А.В.

склала цей акт про впровадження в навчальний процес лекційних матеріалів і методичних вказівок до практичних занять, розроблених Євлановим М.В. на основі результатів його дисертаційної роботи.

Місце впровадження: факультет комп'ютерних наук Харківського національного університету радіоелектроніки, кафедра інформаційних управляючих систем.

Склад впровадження:

– лекції «Виявлення вимог до інформаційної системи, що дублюються», «Синтез архітектури створюваної інформаційної системи», «Оцінка обсягу робіт з реалізації інформаційної системи» з курсу «Формування та аналіз вимог до інформаційних управляючих систем»;

– методичні вказівки до практичних занять з курсу «Формування та аналіз вимог до інформаційних управляючих систем» (практичні заняття «Формування функціональних вимог до створюваної інформаційної управляючої системи чи технології», «Аналіз формальних описів функціональних вимог створюваної інформаційної управляючої системи чи технології», «Вибір раціонального опису архітектури створюваної інформаційної управляючої системи чи технології»).

Голова комісії,
д-р техн. наук, професор



А.Л. Єрохін

Члени комісії,
д-р техн. наук, професор



С.Ф. Чалий

канд. техн. наук, доцент



І.Ю. Панфьорова

канд. техн. наук, доцент



А.В. Міхнова

ПРИЛОЖЕНИЕ Б

ИССЛЕДОВАНИЕ СПОСОБОВ ПРЕДСТАВЛЕНИЙ ТРЕБОВАНИЙ К
ИНФОРМАЦИОННОЙ СИСТЕМЕ

Б.1 Описание существующих классификаций требований к информационной системе

В ГОСТах группы 34 «Информационные технологии» на первой стадии разработки АС требования к АС разделяются на две основные группы [17]:

а) требования к характеристикам реализации функций и задач в соответствии с действующими нормативно-техническими документами, определяющими общие технические требования к АС конкретного вида;

б) дополнительные требования к АС в целом и ее частям, учитывающие специфику создаваемой АС.

В дополнение к этим группам требований в документе, описывающем результаты выполнения работ на стадии «Формирование требований к автоматизированной системе», выделяют ряд общих рекомендаций следующего характера [17]:

а) по виду создаваемой АС, ее совместимости с другими АС и неавтоматизируемой частью соответствующей системы;

б) по организационной и функциональной структуре создаваемой АС;

в) по составу и характеристикам подсистем и видов обеспечений АС;

г) по организации использования имеющихся и приобретению дополнительных средств вычислительной техники;

д) по рациональной организации разработки и внедрения АС;

е) по определению основных и дополнительных, внешних и внутренних источников и видов объемов финансирования и материального обеспечения разработок АС;

ж) по обеспечению производственных условий создания АС;

и) другие рекомендации по созданию АС.

Данные рекомендации являются, по сути, слабо формализованными описа-

ниями требований к АС, которые на данной стадии сформулированы недостаточно четко.

В документе, содержащем описание результатов выполнения работ на второй стадии разработки АС, дополнительно к перечисленным выше приводятся описания требований, гарантирующих качество АС [17].

В ГОСТ 34.602-89 «Техническое задание на создание автоматизированной системы» требования к АС делятся на три основные группы [18]:

- а) требования к системе в целом;
- б) требования к функциям (задачам), выполняемым системой;
- в) требования к видам обеспечений.

Группа требований к системе в целом, в свою очередь, делится на следующие виды требований: требования к структуре и функционированию системы; требования к численности и квалификации персонала системы и режиму его работы; показатели назначения; требования к надежности; требования безопасности; требования к эргономике и технической эстетике; требования к транспортабельности для подвижных АС; требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов системы; требования к защите информации от несанкционированного доступа; требования по сохранности информации при авариях; требования к защите от влияния внешних воздействий; требования по стандартизации и унификации; дополнительные требования.

Группа требований к видам обеспечений, в свою очередь, делится на следующие требования (в зависимости от вида системы): к математическому, информационному, лингвистическому, программному, техническому, метрологическому, организационному, методическому и другим видам обеспечений.

В методологии SSADM существует деление требований к ИС на функциональные и нефункциональные. Функциональные требования определяют, что должна делать создаваемая система и зачастую формируются на основе выявленных ранее бизнес-правил. Нефункциональные требования в общем случае могут быть самыми разнообразными. Обычно выделяют следующие

типы нефункциональных требований [19, 20]:

- требования к качеству выполнения функций;
- требования к ограничению доступа;
- требование к безопасности;
- требования к обеспечению мониторинга и контролируемости;
- требования к ограничениям;
- архитектурные требования и т.д.

В [13] Д. Леффингуэллом и Д. Уидригом предлагается разделять требования на следующие группы:

а) область проблемы (технические или бизнес-задачи предметной области (ПрО));

б) потребности заинтересованных лиц, в том числе пользователей системы (отражение некоей личной, рабочей или бизнес-проблемы (или возможности), решение которой оправдывает замысел, покупку или использование новой системы);

в) функции системы (предоставляемое системой обслуживание для удовлетворения одной или нескольких потребностей);

г) требования к ПО, а именно: функциональные требования к ПО; нефункциональные требования к ПО (практичность, надежность, производительность, возможность обслуживания); ограничения проектирования.

К.И. Виггерс предлагает разделять требования на следующие группы, которые он называет «уровнями» [21]:

а) бизнес-требования, содержащие высокоуровневые цели организации или заказчиков системы;

б) требования пользователей, описывающие цели и задачи, которые пользователям позволит решить система;

в) функциональные требования, определяющие функциональность ПО, которое разработчики должны построить, чтобы пользователи смогли выполнить свои задачи в рамках бизнес-требований;

г) системные требования – высокоуровневые требования к системе, спо-

собные определять функциональные требования.

К.И. Виггерс также предлагает при документировании функциональных требований выделить нефункциональные требования, описывающие цели и атрибуты качества функций системы.

Сотрудники компании Telelogic, разрабатывающей пакеты управления требованиями Telelogic DOORS, предлагают разделение требований к системе на следующие группы [23]:

- а) формулировки потребностей;
- б) пользовательские требования;
- в) системные требования;
- г) требования для компонентов системы;
- д) требования для компонентов подсистем.

В модели классификации требований FURPS, предложенной Р. Грэди (компания «Hewlett-Packard»), классы требований выделяются по следующим признакам [24]:

- а) функциональность (Functionality);
- б) удобство использования (Usability);
- в) надежность (Reliability);
- г) производительность (Performance)
- д) поддерживаемость (Supportability).

В современном варианте этой модели – FURPS+ – дополнительно выделяются такие классы требований [25]:

- а) проектные требования;
- б) требования к реализации;
- в) требования к интерфейсу;
- г) физические требования.

С точки зрения анализа БП современных предприятий, в том числе – анализа с целью определения целесообразности автоматизации данных БП – требования следует разделять на такие группы [26]:

- а) архитектурные требования (определяют структуру и поведение си-

стемы);

б) бизнес-требования (определяют цели и задачи или потребности организации);

в) требования пользователей/заинтересованных лиц (определяют потребности отдельных лиц или групп заинтересованных лиц в системе);

г) функциональные требования или требования к решению (описывают возможности, поведение и информацию, которые должны быть реализованы в рамках системного решения);

д) требования к качеству сервиса или нефункциональные требования (описывают условия, в которых решение должно работать эффективно и качественно, или ограничения, накладываемые на решение);

е) требования к реализации или переходные требования (описывают возможности и поведение системы при переходе от текущего состояния предприятия к желаемому).

Соответствие рассмотренных вариантов классификации требований техническим процессам ЖЦ системы показано в табл. Б.1.

Б.2 Описание результатов анализа частоты рекомендаций отдельных методов сбора требований к информационной системе

Результаты анализа частоты рекомендаций отдельных методов сбора требований к ИС приведены в табл. Б.2.

Таблица Б.1 – Соответствие вариантов классификации требований техническим процессам жизненного цикла системы

Процесс ЖЦ системы	Варианты классификации требований						
	ГОСТы группы 34	SSADM	Д. Леффингуэлл и Д. Уидриг	К.И. Виггерс	Telelogic	FURPS+	BAVOK
Формирование требований правообладателей	Требования к характеристикам реализации функций и задач в соответствии с действующими нормативно-техническими документами, определяющими общие технические требования к АС конкретного вида	- функциональные требования; - требования к ограничениям.	Область проблемы	Бизнес-требования		Функциональность	Бизнес-требования
	Дополнительные требования к АС в целом и ее частям, учитывающие специфику создаваемой АС		Потребности заинтересованных лиц, в том числе пользователей системы	Требования пользователей	- формулировки потребностей; - пользовательские требования.		- архитектурные требования; - требования пользователей/заинтересованных лиц.
Анализ требований	Требования, гарантирующие качество АС	- требования к качеству выполнения функций; - требования к обеспечению мониторинга и контролируемости.				- удобство использования; - надежность; - производительность; - поддерживаемость.	

Продолжение табл. Б.1

Процесс ЖЦ системы	Варианты классификации требований						
	ГОСТы группы 34	SSADM	Д. Леффингуэлл и Д. Уидриг	К.И. Виггерс	Telelogic	FURPS+	BAVOK
Проектирование архитектуры системы	Требования к системе в целом	- архитектурные требования; - требования к ограничению доступа; - требования к безопасности.		Системные требования	Системные требования		
	Требования к функциям (задачам), выполняемым системой	Функциональные требования	Функции системы	Функциональные требования	- требования для компонентов системы; - требования для компонентов подсистем.	Функциональность	Функциональные требования
	Требования к видам обеспечений	Нефункциональные требования	- функциональные требования к ПО; - нефункциональные требования к ПО.			- проектные требования; - требования к реализации; - требования к интерфейсу; - физические требования.	- требования к реализации; - требования к качеству сервиса.

Таблица Б.2 – Результаты анализа частоты рекомендаций методов сбора требований к информационной системе

Метод сбора требований	Идентификатор источника, упоминающего метод												Количество упоминаний	Частота рекомендации
	1	2	3	4	5	6	7	8	9	10	11	12		
Интервью	1	1	1	1	1	1	1	1	1			1	10	0,833
Наблюдение	1	1	1	1	1	1		1	1		1	1	10	0,833
Прототипирование	1	1	1		1	1	1	1	1			1	9	0,75
Анализ документов (обзор технической документации, операционные пошаговые руководства, описательная документация)	1	1		1	1	1			1			1	7	0,583
Семинары по сбору требований (совместный семинар)	1	1		1	1		1	1			1		7	0,583
Анкетирование	1	1		1	1	1						1	6	0,5
Мозговой штурм	1	1			1	1	1						5	0,417
Сценарии (сценарный анализ)			1			1		1	1	1			5	0,417
Изучение аналогов (опыт работы с аналогичными системами)						1	1		1			1	4	0,333
Фокус-группы	1	1									1		3	0,25
Разговор с командой поддержки (проблемы, обнаруженные в существующих системах, и идеи как их исправить, изучение отчетов о проблемах работающих систем с целью поиска новых идей)							1		1		1		3	0,25
Профилирование пользователя (анализ задач конечного пользователя, погружение в среду пользователя)				1		1	1						3	0,25
Демонстрация технологий (возможности новых технологий)						1			1				2	0,167
Действующие системы, которые необходимо модернизировать									1			1	2	0,167
Опросы		1							1				2	0,167
Бенчмаркинг		1			1								2	0,167
Просмотр журнала действий пользователя (анализ протоколов)				1				1					2	0,167
Анализ рынка или оценка конкурентности системы					1				1				2	0,167

Продолжение табл. Б.2

Метод сбора требований	Идентификатор источника, упоминающего метод												Количество упоминаний	Частота рекомендации
	1	2	3	4	5	6	7	8	9	10	11	12		
Симуляция, моделирование					1	1							2	0,167
Варианты использования						1				1			2	0,167
Анализ бизнес-кейсов						1				1			2	0,167
Исследования удовлетворенности клиентов (изучение «книги жалоб и предложений»)						1	1						2	0,167
Методы группового принятия решений		1											1	0,083
Разъясняющие встречи			1										1	0,083
Анализ интерфейсов	1												1	0,083
Контекстные диаграммы		1											1	0,083
Методы организационного анализа (SWOT-анализ, анализ портфеля продуктов)					1								1	0,083
Разработка функции качества						1							1	0,083
Бета-тестирование						1							1	0,083
Реверсивное проектирование						1							1	0,083
«Слежка» за опытным сотрудником								1					1	0,083
Результаты исследований									1				1	0,083
Повторное использование требований в разных проектах											1		1	0,083
Совместная разработка приложений (JAD-метод)												1	1	0,083
Быстрая разработка приложений (RAD- метод)												1	1	0,083

1 – свод знаний BABOK v2.0; 2 – свод знаний PMBOK v.4; 3 – свод знаний SWEBOOK; 4 – свод знаний REBOOK; 5 – стандарт ISO/IEC 29148; 6 – комплекс моделей CMMI DEV v1.3; 7 – методологии IBM OpenUP и FURPS+; 8 – библиотека показателей ITIL v2; 9 – Э. Халл, К Джексон и Дж. Дик; 10 – А. Коберн; 11 - К.И. Виггерс; 12 – Л.А. Мацяшек.

Б.3 Описание моделей требований методологии SSADM и компании «Microsoft»

Описание модели требований методологии SSADM приведено в табл. Б.3. Описание модели требования компании «Microsoft» приведено в табл. Б.4.

Таблица Б.3 – Описания требований к системе, образующие модель требований методологии SSADM

Способ представления (модель или документ)	Предмет описания в модели требований	Назначение описания
Документ «Каталог требований»	Полное описание требования на языке участников IT-проекта создания системы	Централизованное представление основных аспектов требования к системе
Реляционная модель данных	Данные, входящие, выходящие и обрабатываемые в ИС	Описание данных, характеризующих информационную среду, окружающую ИС, и саму ИС
Модель потоков данных	Кто использует систему, и какие действия они в ней совершают	Описание использования системы
Объектно-событийные модели	Рабочий процесс и обмен сведениями в процессе взаимодействия процессов обработки данных и хранилищ данных	Описание путей доступа процессов обработки данных к данным

Продолжение табл. Б.3

Способ представления (модель или документ)	Предмет описания в модели требований	Назначение описания
Логические модели данных	Обмен сведениями в процессе взаимодействия процессов обработки данных и хранилищ данных	Описание путей доступа запросов к данным
Прототипы	Упрощенное описание требования на языке программирования	Дальнейшее исследование и проверка правильности требования
Документ «Каталог пользователя»	Кто использует систему	Описание пользователей системы
Матрица «Роль пользователя/требование»	Кто какие требования выдвигает к системе	Описание пользователей как источников требований

Таблица Б.4 – Описания требований к системе, образующие модель требований компании «Microsoft»

Способ представления (модель или документ)	Предмет описания в модели требований	Назначение описания
Модель вариантов использования	Кто использует систему, и какие действия они в ней совершают	Описание использования системы
Модель деятельности	Рабочий процесс и обмен сведениями в процессе взаимодействия пользователей и системы или ее частей	Отображение рабочего процесса между пользователями и системой

Продолжение табл. Б.4

Способ представления (модель или документ)	Предмет описания в модели требований	Назначение описания
Модель последовательностей	Последовательность взаимодействий между пользователями и системой или ее частями (альтернативное представление модели деятельности)	Отображение взаимодействий между пользователями и системой
Дополнительные документы или рабочие элементы	Критерии производительности, безопасности, полезности и надежности	Описание требований к качеству обслуживания
Дополнительные документы или рабочие элементы	Ограничения и правила, не относящиеся к конкретному варианту использования	Отображение бизнес-правил

Результаты сравнительного анализа моделей требований к ИС методологии как наиболее типичных представителей подобных моделей приведены в табл. Б.5.

Таблица Б. 5 – Сравнительный анализ моделей требований методологии SSADM и компании «Microsoft»

Описываемый объект	Предмет описания в модели требований	Способ представления в модели требований SSADM	Способ представления в модели требований «Microsoft»
1	2	3	4
Бизнес-процессы ОА			Дополнительные документы или рабочие элементы, отображающие бизнес-правила
Пользователь системы	Описание пользователей системы	Документ «Каталог пользователя», модель потоков данных	Модель вариантов использования
Требование к системе	Описание требования, выдвигаемого к ИС	Документ «Каталог требований», прототипы	Концептуальная модель классов
	Описание требований к качеству обслуживания		Дополнительные документы или рабочие элементы, описывающие требования к качеству обслуживания

Продолжение табл. Б.5

1	2	3	4
Взаимосвязь пользователя с требованиями	Перечень требований, выдвигаемых пользователем к системе	Матрица «Роль пользователя / требование»	
Процессы системы	Описание процессов, происходящих в системе	Объектно-событийные модели, логические модели данных	Модель деятельности, модель последовательностей (альтернатива модели деятельности)
Взаимосвязь пользователя с процессами системы	Описание действий, совершаемых пользователем в системе	Модель потоков данных	Модель вариантов использования
Данные системы	Описание данных, входящих, выходящих и обрабатываемых в ИС	Реляционная модель данных, логические модели данных	Концептуальная модель классов

Б.4 Описание шаблона определения статуса требования по К.И. Виггерсу

Описание типового шаблона определения статуса требования по К.И. Виггерсу приведено в табл. Б.6.

Таблица Б.6 – Шаблон определения статуса требования

Состояние	Определение
Proposed (Предложено)	Требование запрошено авторизованным источником
Approved (Одобрено)	Требование проанализировано, его влияние на проект просчитано, требование было размещено в базовой версии проекта. Ключевые заинтересованные в проекте лица согласились с этим требованием, а разработчики ПО обязались реализовать его
Implemented (Реализовано)	Код, реализующий требование, разработан, написан и протестирован. Требование отслежено до соответствующих элементов дизайна и кода
Verified (Про- верено)	Корректное функционирование реализованного требования подтверждено в соответствующем продукте. Требование отслежено до соответствующих вариантов тестирования. Теперь требование считается завершенным
Deleted (Уда- лено)	Утвержденное требование удалено из базовой версии. Необходимо описать причины удаления и назвать того, кто принял решение об удалении требования
Rejected (От- клонено)	Требование предложено, но не запланировано для реализации ни в одной из будущих версий. Необходимо описать причины отклонения и назвать того, кто принял это решение

ПРИЛОЖЕНИЕ В
ТЕОРЕТИКО-КАТЕГОРНЫЕ МОДЕЛИ ЭЛЕМЕНТОВ УНИВЕРСУМА
ТРЕБОВАНИЙ К ИНФОРМАЦИОННОЙ СИСТЕМЕ

Модель множества данных, информации и знаний, неизвестных Поставщику, $\langle UnD^{Sp}, UnI^{Sp}, UnK^{Sp} \rangle$ будет иметь следующий вид:

$$L_{Sp} = [UnD^{Sp}, UnI^{Sp}, UnK^{Sp}, H(UnD^{Sp}), H(UnI^{Sp}), H(UnK^{Sp}), \\ H(UnD^{Sp}, UnI^{Sp}), H(UnI^{Sp}, UnD^{Sp}), H(UnD^{Sp}, UnK^{Sp}), \\ H(UnK^{Sp}, UnD^{Sp}), H(UnI^{Sp}, UnK^{Sp}), H(UnK^{Sp}, UnI^{Sp})]. \quad (B.1)$$

Модель множества данных, информации и знаний, неизвестных Потребителю, $\langle UnD^{Cs}, UnI^{Cs}, UnK^{Cs} \rangle$ будет иметь следующий вид:

$$L_{Cs} = [UnD^{Cs}, UnI^{Cs}, UnK^{Cs}, H(UnD^{Cs}), H(UnI^{Cs}), H(UnK^{Cs}), \\ H(UnD^{Cs}, UnI^{Cs}), H(UnI^{Cs}, UnD^{Cs}), H(UnD^{Cs}, UnK^{Cs}), \\ H(UnK^{Cs}, UnD^{Cs}), H(UnI^{Cs}, UnK^{Cs}), H(UnK^{Cs}, UnI^{Cs})]. \quad (B.2)$$

Модель группы СТ к ИС как к аспекту бизнеса L_{IS}^{IB} будет иметь следующий вид:

$$L_{IS}^{IB} = [D_{IS}^{IB}, I_{IS}^{IB}, K_{IS}^{IB}, H(D_{IS}^{IB}), H(I_{IS}^{IB}), H(K_{IS}^{IB}), H(D_{IS}^{IB}, I_{IS}^{IB}), \\ H(I_{IS}^{IB}, D_{IS}^{IB}), H(D_{IS}^{IB}, K_{IS}^{IB}), H(K_{IS}^{IB}, D_{IS}^{IB}), H(I_{IS}^{IB}, K_{IS}^{IB}), H(K_{IS}^{IB}, I_{IS}^{IB})]. \quad (B.3)$$

Модель группы СТ к ИС в целом L_{IS}^S будет иметь следующий вид:

$$L_{IS}^s = [D_{IS}^s, I_{IS}^s, K_{IS}^s, H(D_{IS}^s), H(I_{IS}^s), H(K_{IS}^s), H(D_{IS}^s, I_{IS}^s), H(I_{IS}^s, D_{IS}^s), H(D_{IS}^s, K_{IS}^s), H(K_{IS}^s, D_{IS}^s), H(I_{IS}^s, K_{IS}^s), H(K_{IS}^s, I_{IS}^s)]. \quad (B.4)$$

Модель группы СФТ к ИТ-услугам L_{IS}^f будет иметь следующий вид:

$$L_{IS}^f = [D_{IS}^f, I_{IS}^f, K_{IS}^f, H(D_{IS}^f), H(I_{IS}^f), H(K_{IS}^f), H(D_{IS}^f, I_{IS}^f), H(I_{IS}^f, D_{IS}^f), H(D_{IS}^f, K_{IS}^f), H(K_{IS}^f, D_{IS}^f), H(I_{IS}^f, K_{IS}^f), H(K_{IS}^f, I_{IS}^f)]. \quad (B.5)$$

Модель группы сформулированных нефункциональных требований к ИТ-услугам L_{IS}^{nf} будет иметь следующий вид:

$$L_{IS}^{nf} = [D_{IS}^{nf}, I_{IS}^{nf}, K_{IS}^{nf}, H(D_{IS}^{nf}), H(I_{IS}^{nf}), H(K_{IS}^{nf}), H(D_{IS}^{nf}, I_{IS}^{nf}), H(I_{IS}^{nf}, D_{IS}^{nf}), H(D_{IS}^{nf}, K_{IS}^{nf}), H(K_{IS}^{nf}, D_{IS}^{nf}), H(I_{IS}^{nf}, K_{IS}^{nf}), H(K_{IS}^{nf}, I_{IS}^{nf})]. \quad (B.6)$$

Модель группы СФТ к ИТ-сервисам L_{IS}^{fw} будет иметь следующий вид:

$$L_{IS}^{fw} = [D_{IS}^{fw}, I_{IS}^{fw}, K_{IS}^{fw}, H(D_{IS}^{fw}), H(I_{IS}^{fw}), H(K_{IS}^{fw}), H(D_{IS}^{fw}, I_{IS}^{fw}), H(I_{IS}^{fw}, D_{IS}^{fw}), H(D_{IS}^{fw}, K_{IS}^{fw}), H(K_{IS}^{fw}, D_{IS}^{fw}), H(I_{IS}^{fw}, K_{IS}^{fw}), H(K_{IS}^{fw}, I_{IS}^{fw})]. \quad (B.7)$$

Модель группы сформулированных нефункциональных требований к ИТ-сервисам L_{IS}^{nfw} будет иметь следующий вид:

$$L_{IS}^{nfw} = [D_{IS}^{nfw}, I_{IS}^{nfw}, K_{IS}^{nfw}, H(D_{IS}^{nfw}), H(I_{IS}^{nfw}), H(K_{IS}^{nfw}), H(D_{IS}^{nfw}, I_{IS}^{nfw}), H(I_{IS}^{nfw}, D_{IS}^{nfw}), H(D_{IS}^{nfw}, K_{IS}^{nfw}), H(K_{IS}^{nfw}, D_{IS}^{nfw}), H(I_{IS}^{nfw}, K_{IS}^{nfw}), H(K_{IS}^{nfw}, I_{IS}^{nfw})]. \quad (B.8)$$

ПРИЛОЖЕНИЕ Г

МОДЕЛИ ПАТТЕРНОВ ПРОЕКТИРОВАНИЯ ВЕРСИЙ ТРЕБОВАНИЙ

Г.1 Модели структурных паттернов проектирования версий требований к информационной системе на уровне данных

В случае использования модели (3.8) для идентификации версии требования к ИС данная модель принимает следующий вид:

$$Pt_{Id} = \langle At_{rg}, At_r, At_{rv}, At_{asp}, At_{acs}, At_{proj}, \langle at_{rg}, at_r, at_{rv}, at_{asp}, at_{acs}, at_{proj} \rangle \rangle, \quad (Г.1)$$

где At_{rv} – кортеж атрибутов, описывающих версии требования к ИС;

at_{rv} – атрибут, идентифицирующий версию требования к ИС.

Интерпретация модели (Г.1) в ходе разработки ИТ формирования и анализа требований к ИС аналогична интерпретации модели (3.8).

В случае необходимости осуществления управления отдельными версиями требования учету подлежат значения следующих показателей:

- а) уникальное описание управляемого требования;
- б) уникальное описание управляемой версии требования;
- в) описание текущего статуса управляемой версии требования;
- г) уникальное описание лиц, принимающих решение об управлении версией требования со стороны Поставщика;
- д) уникальное описание лиц, принимающих решение об управлении версией требования со стороны Потребителя;
- е) дата и время принятия управленческого решения об изменении текущего статуса версии требования;
- ж) описание причины принятия управленческого решения об изменении статуса версии требования;
- и) дата и время изменения текущего статуса версии требования.

В случае использования модели (3.9) для описания процесса управления версией требования к ИС данная модель уточняется путем добавления кортежа атрибутов At_{rv} , идентифицирующих управляемую версию требования к ИС. В результате этого уточнения модель (3.9) будет иметь следующий вид:

$$Pt_{Ctrl} = \langle At_r, At_{rv}, At_{cst}, At_{dsp}, At_{dcs}, \langle at_r, at_{rv}, at_{cst}, at_{dsp}, at_{dcs}, at_{ddt}, at_{cause}, at_{cdt} \rangle \rangle. \quad (Г.2)$$

Интерпретация модели (Г.2) в ходе разработки ИТ формирования и анализа требований к ИС аналогична интерпретации модели (3.9).

В случае использования модели (3.10) для учета результатов выполнения операций над версиями требований к ИС данная модель будет иметь следующий вид:

$$Pt_{Op} = \langle At_r, At_{rv}, At_{opr}, At_{aopsp}, At_{aopcs}, \langle at_r, at_{rv}, at_{opr}, at_{aopsp}, at_{aopcs}, at_{dopr} \rangle \rangle, \quad (Г.3)$$

где at_{dopr} – атрибут, описывающий дату и время выполнения операции над версией требования к ИС.

Интерпретация модели (4.13) в ходе разработки ИТ формирования и анализа требований к ИС аналогична интерпретации модели (3.10).

Для случая использования отдельных версий требований к ИС в качестве элементарных описаний создаваемой системы подкласс объектов D_{IS}^{Pt} модели (2.40) следует рассматривать как множество структурных паттернов проектирования версий требований к ИС, устанавливающих конкретный вид элемента $\langle M_D^{AtPt} \rangle$ модели (3.7). Это множество паттернов будет в общем случае иметь следующий вид:

$$\begin{aligned}
D_{IS}^{Pt} = \{ Pt_{Id}, Pt_{Ctrl}, Pt_{Op} \} = \{ At_{rg}, At_r, At_{rv}, At_{asp}, At_{acs}, At_{proj}, \\
\langle at_{rg}, at_r, at_{rv}, (at_{apr}), (at_{au}), at_{proj} \rangle, \langle At_r, At_{rv}, At_{cst}, At_{dsp}, At_{dcs}, \\
\langle at_r, at_{rv}, at_{cst}, at_{dsp}, at_{dcs}, at_{ddt}, at_{cause}, at_{cdt} \rangle, \langle At_r, At_{rv}, At_{opr}, \\
At_{aopsp}, At_{aopcs}, \langle at_r, at_{rv}, at_{opr}, at_{aopsp}, at_{aopcs}, at_{dopr} \rangle \}. \quad (Г.4)
\end{aligned}$$

Г.2 Модели структурных паттернов проектирования версий требований к информационной системе на уровне информации

В случае использования модели (3.31) для идентификации версии требования к ИС, данная модель принимает следующий вид:

$$\begin{aligned}
Pt_{r_publ} = \langle At_r, At_{rv}, At_{pub_t}, At_{pub}, At_{files}, At_{proj}, \\
\langle at_r, at_{rv}, at_{pub_t}, at_{pub}, at_{files}, at_{proj} \rangle \}. \quad (Г.5)
\end{aligned}$$

Интерпретация модели (Г.5) в ходе разработки ИТ формирования и анализа требований к ИС аналогична интерпретации модели (3.31).

Для случая использования в качестве элементарных описаний создаваемой системы отдельных версий требований к ИС подкласс объектов I_{IS}^{Pt} модели (2.40) следует рассматривать как множество структурных паттернов проектирования версий требований к ИС на уровне информации, устанавливающих конкретный вид элемента $\langle M_I^{AtPt} \rangle$ модели (3.30). Это множество паттернов в общем случае будет иметь следующий вид:

$$\begin{aligned}
I_{IS}^{Pt} &= \{ Pt_r_publ, Pt_r_kn_dst, Pt_r_kn_eobj, Pt_r_kn_proc \} = \\
&= \{ \langle At_r, At_{rv}, At_{pub_t}, At_{pub}, At_{files}, At_{proj}, \langle at_r, at_{rv}, at_{pub_t}, at_{pub}, \\
&\quad at_{files}, at_{proj} \rangle \rangle, \langle At_{pub_t}, At_{pub}, At_{str_t}, At_{str}, At_{str_m_t}, At_{str_m}, \\
&\quad At_{attr_t}, At_{attr}, \langle at_{pub_t}, at_{pub}, at_{str_t}, at_{ztr}, at_{str_m_t}, at_{str_m}, \\
&\quad at_{attr_t}, at_{attr} \rangle \rangle, \langle At_{pub_t}, At_{pub}, At_{eobj_t}, At_{eobj}, \langle at_{pub_t}, at_{pub}, \\
&\quad at_{eobj_t}, at_{eobj} \rangle \rangle, \langle At_{pub_t}, At_{pub}, At_{proc}, At_{inst_t}, At_{inst}, \\
&\quad \langle at_{pub_t}, at_{pub}, at_{proc}, at_{inst_t}, at_{inst} \rangle \rangle \}. \tag{\Gamma.6}
\end{aligned}$$

ПРИЛОЖЕНИЕ Д

ПРИМЕРЫ РЕАЛИЗАЦИИ РЕЗУЛЬТАТОВ ДИССЕРТАЦИОННОЙ РАБОТЫ

На рис. Д.1 приведен фрагмент команды, создающей МП «requirement_version_person_frame» в ходе выполнения Шага 1 алгоритма реализации метода формирования представления СТ на уровне знаний отдельного УАП.

```
CREATE MATERIALIZED VIEW requirement_version_person_frame
BUILD IMMEDIATE
REFRESH FORCE
ON DEMAND
ENABLE QUERY REWRITE
AS
SELECT a.id_instance, a.instance_name
FROM APP_INSTANCE.instance a, APP_DIAGRAMM.diagramm i, APP_INSTANCE.Instance_in_diagramm j
WHERE i.diagramm_name= :diagramm_name AND j.ID_diagramms=i.ID_diagramms AND a.id_instance=j.id_instance
```

Рисунок Д.1 – Фрагмент команды на создание материализованного представления «requirement_version_person_frame»

На рис. Д.2 приведен фрагмент команды, создающей МП «requirement_version_person_attributes» в ходе выполнения Шага 2 алгоритма реализации метода формирования представления СТ на уровне знаний отдельного УАП.

На рис. Д.3 приведен фрагмент команды, создающей МП «requirement_version_person_relation» в ходе выполнения Шага 3 алгоритма реализации метода формирования представления СТ на уровне знаний отдельного УАП.

На рис. Д.4 приведен фрагмент команды, создающей МП «requirement_user_frame» в ходе выполнения Шага 1 алгоритма реализации метода формирования представления СТ на уровне знаний Поставщика.

```

CREATE MATERIALIZED VIEW requirement_version_person_attributes
BUILD IMMEDIATE
REFRESH FORCE
ON DEMAND
ENABLE QUERY REWRITE
AS
SELECT k.id_instance, b.model_class_id, d.ID_attribute, d.Attribute_name, e.ID_attribute_type,
       e.Attribute_type_name
FROM requirement_version_person_frame k, APP_ENTITY.classes b, APP_ENTITY.classes_members c,
     APP_ATTRIBUTE.Attribute d, APP_ATTRIBUTE.Attribute_type e, APP_ENTITY.class_attributes f,
     APP_ENTITY.class_members_attributes g
WHERE b.id_instance=k.id_instance AND c.id_instance=k.id_instance AND f.id_instance=k.id_instance AND
      c.id_member=b.model_class_id AND d.ID_attribute=f.ID_attribute AND
      e.ID_attribute_type=d.ID_attribute_type

```

Рисунок Д.2 – Фрагмент команды на создание материализованного представления «requirement_version_person_attributes»

```

CREATE MATERIALIZED VIEW requirement_version_person_relation
BUILD IMMEDIATE
REFRESH FORCE
ON DEMAND
ENABLE QUERY REWRITE
AS
SELECT h.ID_MESSAGE, h.message_name, n.message_kind_name, h.id_source_instance, h.id_target_instance,
       l.ID_attribute, l.Attribute_name, l.ID_attribute_type, l.Attribute_type_name
FROM APP_MESSAGE.MESSAGE h, requirement_version_person_frame k, requirement_version_person_attributes l,
     APP_MESSAGE.operation m, APP_MESSAGE.message_kinds n
WHERE (k.id_instance=h.id_source_instance OR k.id_instance=h.id_target_instance) AND
      (l.id_instance=h.id_source_instance OR l.id_instance=h.id_target_instance) AND
      (m.ID_MESSAGE=h.ID_MESSAGE AND n.id_message_kind=m.id_message_kind)
GROUP BY l.id_instance, l.id_attribute, l.ID_attribute_type

```

Рисунок Д.3 – Фрагмент команды на создание материализованного представления «requirement_version_person_relation»

```

CREATE MATERIALIZED VIEW requirement_user_frame
BUILD IMMEDIATE
REFRESH FORCE
ON DEMAND
ENABLE QUERY REWRITE
AS
SELECT b.id_instance, b.instance_name
FROM APP_REQUIREMENT_AN.Analytical_requirement a, APP_REQUIREMENT_AN.Instance_in_requirement b
WHERE b.id_requirement_version=a.id_requirement_version AND a.ID_CUSTOMER_PERSON_DESIGN=&id_CUSTOMER

```

Рисунок Д.4 – Фрагмент команды на создание материализованного представления «requirement_user_frame»

На рис. Д.5 приведен фрагмент команды, создающей МП

«requirement_provider_frame» в ходе выполнения Шага 2 алгоритма реализации метода формирования представления СТ на уровне знаний Поставщика.

```
CREATE MATERIALIZED VIEW requirement_provider_frame
BUILD IMMEDIATE
REFRESH FORCE
ON DEMAND
ENABLE QUERY REWRITE
AS
SELECT c.id_instance, c.instance_name
FROM requirement_user_frame k, APP_INSTANCE.Instance c
WHERE c.instance_name=k.instance_name
```

Рисунок Д.5 – Фрагмент команды на создание материализованного представления «requirement_provider_frame»

На рис. Д.6 приведен фрагмент команды, создающей МП «requirement_provider_frame_extended» в ходе выполнения Шага 3 алгоритма реализации метода формирования представления СТ на уровне знаний Поставщика.

```
CREATE MATERIALIZED VIEW requirement_provider_frame_extended
BUILD IMMEDIATE
REFRESH FORCE
ON DEMAND
ENABLE QUERY REWRITE
AS
SELECT c.id_instance, c.instance_name
FROM requirement_user_frame k, APP_INSTANCE.Instance c
WHERE NOT EXIST
(SELECT APP_ATTRIBUTE.Attribute.Attribute_name
FROM APP_ATTRIBUTE.Attribute, APP_REQUIREMENT_AN.Instance_in_requirement
WHERE APP_REQUIREMENT_AN.Instance_in_requirement.id_instance=requirement_user_frame.id_instance
MINUS
SELECT APP_ATTRIBUTE.Attribute.Attribute_name
FROM APP_ATTRIBUTE.Attribute w, APP_REQUIREMENT.Implemented_attribute x, APP_INSTANCE.Instance y,
APP_IMPLEMENTATION.Attribute_in_implementation z
WHERE x.ID_attribute=w.ID_attribute AND z.ID_attribute=w.ID_attribute AND
z.ID_requirement_instance=y.id_instance)
```

Рисунок Д.6 – Фрагмент команды на создание материализованного представления «requirement_provider_frame_extended»

На рис. Д.7 приведен фрагмент команды, создающей МП «requirement_provider_attributes» в ходе выполнения Шага 4 алгоритма реализации метода формирования представления СТ на уровне знаний Поставщика.

```

CREATE MATERIALIZED VIEW requirement_provider_attributes
BUILD IMMEDIATE
REFRESH FORCE
ON DEMAND
ENABLE QUERY REWRITE
AS
SELECT l.id_instance, d.ID_attribute, d.Attribute_name, e.ID_attribute_type, e.Attribute_type_name
FROM requirement_provider_frame l, APP_ATTRIBUTE.Attribute d, APP_ATTRIBUTE.Attribute_type e,
     APP_IMPLEMENTATION.Attribute_in_implementation f, APP_REQUIREMENT.Implemented_Attribute g
WHERE f.requirement_instance=l.id_instance AND g.ID_attribute=f.ID_attribute AND
     d.ID_attribute=g.ID_attribute AND e.ID_attribute_type=d.ID_attribute_type
UNION
SELECT m.id_instance, d.ID_attribute, d.Attribute_name, e.ID_attribute_type, e.Attribute_type_name
FROM requirement_provider_frame_extended m, APP_ATTRIBUTE.Attribute d, APP_ATTRIBUTE.Attribute_type e,
     APP_IMPLEMENTATION.Attribute_in_implementation f, APP_REQUIREMENT.Implemented_Attribute g
WHERE f.requirement_instance=m.id_instance AND g.ID_attribute=f.ID_attribute AND
     d.ID_attribute=g.ID_attribute AND e.ID_attribute_type=d.ID_attribute_type

```

Рисунок Д.7 – Фрагмент команды на создание материализованного представления «requirement_provider_attributes»

На рис. Д.8 приведен фрагмент команды, создающей МП «requirement_provider_relations» в ходе выполнения Шага 5 алгоритма реализации метода формирования представления СТ на уровне знаний Поставщика.

```

CREATE MATERIALIZED VIEW requirement_provider_relations
BUILD IMMEDIATE
REFRESH FORCE
ON DEMAND
ENABLE QUERY REWRITE
AS
SELECT l.id_instance, h.ID_relation_type
FROM requirement_provider_frame l, APP_IMPLEMENTATION.Attribute_in_implementation f,
     APP_REQUIREMENT.Relationship h
WHERE f.requirement_instance=l.id_instance AND h.ID_requirement_instance=f.id_instance
UNION
SELECT m.id_instance, h.ID_relation_type
FROM requirement_provider_frame m, APP_IMPLEMENTATION.Attribute_in_implementation f,
     APP_REQUIREMENT.Relationship h
WHERE f.requirement_instance=m.id_instance AND h.ID_requirement_instance=f.id_instance

```

Рисунок Д.8 – Фрагмент команды на создание материализованного представления «requirement_provider_relations»

На рис. Д.9 приведен фрагмент команды, создающей МП «requirement_idea» в ходе выполнения Шага 1 алгоритма реализации

формирования общесистемного представления СТ на уровне знаний.

```
CREATE MATERIALIZED VIEW requirement_idea
BUILD IMMEDIATE
REFRESH FORCE
ON DEMAND
ENABLE QUERY REWRITE
AS
SELECT a.ID_requirement_version, b.ID_ANALITICAL_REQUIREMENT, a.ID_CUSTOMER_PERSON_DECISION,
       a.ID_ANALITIC_PERSON_DECISION, a.Analitical_requirement_name
FROM APP_REQUIREMENT_VERS.Analitical_requirement_version a, APP_REQUIREMENT_AN.Analitical_requirement b,
     APP_PROJECT. Requirement_in_project c
WHERE c.ID_project=:sid_project AND b.ID_ANALITICAL_REQUIREMENT=c.ID_ANALITICAL_REQUIREMENT AND
     a.ID_ANALITICAL_REQUIREMENT=b.ID_ANALITICAL_REQUIREMENT
```

Рисунок Д.9 – Фрагмент команды на создание материализованного представления «requirement_idea»

На рис. Д.10 приведен фрагмент команды, создающей МП «requirement_frame_user» в ходе выполнения Шага 2 алгоритма реализации формирования общесистемного представления СТ на уровне знаний.

```
CREATE MATERIALIZED VIEW requirement_frame_user
BUILD IMMEDIATE
REFRESH FORCE
ON DEMAND
ENABLE QUERY REWRITE
AS
SELECT a.ID_requirement_version, b.ID_ANALITICAL_REQUIREMENT, a.ID_CUSTOMER_PERSON_DECISION,
       c.id_instance, c.instance_name
FROM APP_REQUIREMENT_VERS.Analitical_requirement_version a, APP_REQUIREMENT_AN.Analitical_requirement b,
     APP_REQUIREMENT_AN.Instance_in_requirement c, requirement_idea d
WHERE a.ID_requirement_version=d.ID_requirement_version AND
     b.ID_ANALITICAL_REQUIREMENT=d.ID_ANALITICAL_REQUIREMENT AND a.ID_CUSTOMER_PERSON_DECISION=:customer AND
     c.id_requirement_version=a.id_requirement_version
```

Рисунок Д.10 – Фрагмент команды на создание материализованного представления «requirement_frame_user»

На рис. Д.11 приведен фрагмент команды, создающей МП «requirement_attributes_user» в ходе выполнения Шага 3 алгоритма реализации формирования общесистемного представления СТ на уровне знаний.

```

CREATE MATERIALIZED VIEW requirement_attributes_user
BUILD IMMEDIATE
REFRESH FORCE
ON DEMAND
ENABLE QUERY REWRITE
AS
SELECT a.ID_requirement_version, b.ID_ANALITICAL_REQUIREMENT, a.ID_CUSTOMER_PERSON_DECISION,
       c.id_instance, d.ID_attribute_version, e.ID_attribute, e.Attribute_name, d.Is_not_null,
       d.Length, f.ID_attribute_type, f.Attribute_type_name
FROM APP_REQUIREMENT_VERS.Analitical_requirement_version a, APP_REQUIREMENT_AN.Analitical_requirement b,
     APP_REQUIREMENT_AN.Instance_in_requirement c, APP_REQUIREMENT_VERS.Analitical_attribute_version d,
     APP_ATTRIBUTE.Attribute e, APP_ATTRIBUTE.Attribute_type f,
     APP_REQUIREMENT_VERS.Attribute_in_requirement g, requirement_idea h, requirement_frame_user i
WHERE a.ID_requirement_version=g.ID_requirement_version AND
      b.ID_ANALITICAL_REQUIREMENT=g.ID_ANALITICAL_REQUIREMENT AND a.ID_CUSTOMER_PERSON_DECISION=&customer AND
      c.id_requirement_version=a.id_requirement_version AND g.ID_requirement_version=a.ID_requirement_version
      AND d.ID_attribute_version=g.ID_attribute_version AND e.ID_attribute=d.ID_attribute AND
      f.ID_attribute_type=e.ID_attribute_type

```

Рисунок Д.11 – Фрагмент команды на создание материализованного представления «requirement_attributes_user»

На рис. Д.12 приведен фрагмент команды, создающей МП «requirement_relations_user» в ходе выполнения Шага 4 алгоритма реализации формирования общесистемного представления СТ на уровне знаний.

```

CREATE MATERIALIZED VIEW requirement_relations_user
BUILD IMMEDIATE
REFRESH FORCE
ON DEMAND
ENABLE QUERY REWRITE
AS
SELECT a.ID_requirement_version, a.ID_CUSTOMER_PERSON_DECISION, d.id_relation, d.relation_name,
       e.ID_relation_type, e.Relation_type_name
FROM APP_REQUIREMENT_VERS.Analitical_requirement_version a, APP_REQUIREMENT_VERS.Relation d,
     APP_REQUIREMENT_AN.Relation_type e, APP_REQUIREMENT_VERS.Relationship f,
     requirement_idea h, requirement_frame_user i, requirement_attributes_user j
WHERE a.ID_requirement_version=h.ID_requirement_version AND a.ID_CUSTOMER_PERSON_DECISION=&customer AND
      f.ID_requirement_version=a.ID_requirement_version AND d.ID_relation=f.ID_relation AND
      e.ID_relation_type=d.ID_relation_type AND (d.id_source_instance=i.id_instance OR
      d.id_target_instance=i.id_instance)

```

Рисунок Д.12 – Фрагмент команды на создание материализованного представления «requirement_relations_user»

На рис. Д.13 приведен фрагмент команды, создающей МП «requirement_frame_is» в ходе выполнения Шага 8 алгоритма реализации формирования общесистемного представления СТ на уровне знаний.

```

CREATE MATERIALIZED VIEW requirement_frame_is
BUILD IMMEDIATE
REFRESH FORCE
ON DEMAND
ENABLE QUERY REWRITE
AS
SELECT b.ID_requirement_version, b.ID_ANALITICAL_REQUIREMENT, b.ID_CUSTOMER_PERSON_DECISION,
       c.ID_ANALITIC_PERSON_DESICION, b.id_instance, b.instance_name
FROM requirement_idea a, requirement_frame_user b, requirement_frame_provider c
WHERE b.ID_requirement_version=a.ID_requirement_version AND c.ID_requirement_version=a.ID_requirement_version
      AND c.id_instance=b.id_instance

```

Рисунок Д.13 – Фрагмент команды на создание материализованного представления «requirement_frame_is»

На рис. Д.14 приведен фрагмент команды, создающей МП «requirement_attributes_is» в ходе выполнения Шага 9 алгоритма реализации формирования общесистемного представления СТ на уровне знаний.

```

CREATE MATERIALIZED VIEW requirement_attributes_is
BUILD IMMEDIATE
REFRESH FORCE
ON DEMAND
ENABLE QUERY REWRITE
AS
SELECT b.ID_requirement_version, b.ID_ANALITICAL_REQUIREMENT, b.ID_CUSTOMER_PERSON_DECISION,
       c.ID_ANALITIC_PERSON_DESICION, b.id_instance, b.ID_attribute_version, b.ID_attribute,
       b.Attribute_name, b.Is_not_null, b.Length, b.ID_attribute_type, b.Attribute_type_name
FROM requirement_idea a, requirement_attribute_user b, requirement_attribute_provider c
WHERE b.ID_requirement_version=a.ID_requirement_version AND c.ID_requirement_version=a.ID_requirement_version
      AND c.Attribute_name=b.Attribute_name

```

Рисунок Д.14 – Фрагмент команды на создание материализованного представления «requirement_attributes_is»

На рис. Д.15 приведен фрагмент команды, создающей МП «requirement_relations_is» в ходе выполнения Шага 10 алгоритма реализации формирования общесистемного представления СТ на уровне знаний.

На рис. Д.16 приведен фрагмент команды, модифицирующей содержимое МП «requirement_frame_is» в ходе выполнения Шага 11 алгоритма реализации формирования общесистемного представления СТ на уровне знаний.

```

CREATE MATERIALIZED VIEW requirement_relations_is
BUILD IMMEDIATE
REFRESH FORCE
ON DEMAND
ENABLE QUERY REWRITE
AS
SELECT b.ID_requirement_version, b.ID_ANALITICAL_REQUIREMENT, b.ID_CUSTOMER_PERSON_DECISION,
       c.ID_ANALITIC_PERSON_DESICION, b.id_relation, b.relation_name, b.ID_relation_type,
       b.Relation_type_name
FROM requirement_idea a, requirement_attribute_user b, requirement_attribute_provider c
WHERE b.ID_requirement_version=a.ID_requirement_version AND c.ID_requirement_version=a.ID_requirement_version
      AND c.relation_name=b.relation_name

```

Рисунок Д.15 – Фрагмент команды на создание материализованного представления «requirement_relations_is»

```

INSERT INTO requirement_frame_is
AS SELECT b.ID_requirement_version, b.ID_ANALITICAL_REQUIREMENT, b.ID_CUSTOMER_PERSON_DECISION,
       b.ID_ANALITIC_PERSON_DESICION, b.id_instance, b.instance_name
FROM requirement_frame_provider b, requirement_frame_user c
WHERE b.id_instance<>c.id_instance AND b.instance_name<>c.instance_name

```

Рисунок Д.16 – Фрагмент команды на дополнение содержимого материализованного представления «requirement_frame_is» описаниями фреймов Поставщика, отсутствующими среди описаний фреймов Потребителя

На рис. Д.17 приведен фрагмент команды, модифицирующей содержимое МП «requirement_attributes_is» в ходе выполнения Шага 12 алгоритма реализации формирования общесистемного представления СТ на уровне знаний.

```

INSERT INTO requirement_attributes_is
AS SELECT b.ID_requirement_version, b.ID_ANALITICAL_REQUIREMENT, b.ID_CUSTOMER_PERSON_DECISION,
       b.ID_ANALITIC_PERSON_DESICION, b.id_instance, b.attribute_version, b.ID_attribute,
       b.Attribute_name, b.Is_not_null, b.Length, b.ID_attribute_type, b. Attribute_type_name
FROM requirement_attributes_provider b, requirement_attributes_user c
WHERE b.ID_attribute<>c.ID_attribute AND b.Attribute_name<>c.Attribute_name

```

Рисунок Д.17 – Фрагмент команды на дополнение содержимого материализованного представления «requirement_attributes_is» описаниями атрибутов фреймов Поставщика, отсутствующими среди описаний атрибутов фреймов Потребителя

На рис. Д.18 приведен фрагмент команды, модифицирующей содержимое МП «requirement_relations_is» в ходе выполнения Шага 13 алгоритма реализации формирования общесистемного представления СТ на уровне знаний.

```
INSERT INTO requirement_relations_is
AS SELECT b.ID_requirement_version, b.ID_ANALITICAL_REQUIREMENT, b.ID_CUSTOMER_PERSON_DECISION,
        b.ID_ANALITIC_PERSON_DESICION, b.id_relation, b.relation_name, b.ID_relation_type,
        b.Relation_type_name
FROM requirement_relations_provider b, requirement_relations_user c
WHERE b.id_relation<>c.id_relation AND b.relation_name<>c.relation_name
```

Рисунок Д.18 – Фрагмент команды на дополнение содержимого материализованного представления «requirement_attributes_is» описаниями связей Поставщика, отсутствующими среди описаний связей Потребителя

ПРИЛОЖЕНИЕ Е
ВИЗУАЛЬНЫЕ МОДЕЛИ ИНФОРМАЦИОННО-АНАЛИТИЧЕСКОЙ
СИСТЕМЫ «РЕЕСТР ЛИГИ УКРАИНСКИХ КЛУБОВ
ИНТЕЛЛЕКТУАЛЬНЫХ ИГР»

На рис. Е.1 приведена исходная схема данных ИАС «Реестр ЛУК» – реляционная схема данных, которая включает как таблицы, описывающие непосредственно Про ВОО «ЛУК», так и таблицы, необходимые для реализации нефункциональных требований к системе (разграничение прав доступа пользователей к системе, корректная обработка исключительных ситуаций при возникновении исключительных ситуаций в процессе работы с БД системы и т.п.).

Для реализации нефункциональных требований применяются следующие таблицы:

- ERROR_MESSAGES (классы ошибок, возникающих при работе с БД);
- DETAILED_ERROR_MESSAGES (детализированные описания ошибок в корреляции с объектами БД, при работе с которыми они могут возникнуть);
- MEMBER_WEB_USER (аккаунты пользователей ИАС, которые одновременно должны являться членами ВОО «ЛУК»);
- USER_PRIVILEGE (назначение пользователям системы привилегий доступа к web-страницам ИАС, реализующим те или иные её функции);
- WEB_PRIVILEGE (виды привилегий или ролей пользователей системы);
- WEB_PRIVILEGE_HIERARCHY (описание вложенности привилегий друг в друга);
- MENU_AVAILABLE_PAGE (описание прав доступа к web-страницам ИАС для различных привилегий пользователей);
- MENU_PAGE_JSP (описание физических web-страниц системы, права доступа к которым назначаются ролям пользователей);
- MENU_PAGE_MODE (описание функционального назначения физических web-страниц: просмотр, редактирование и т.п.);

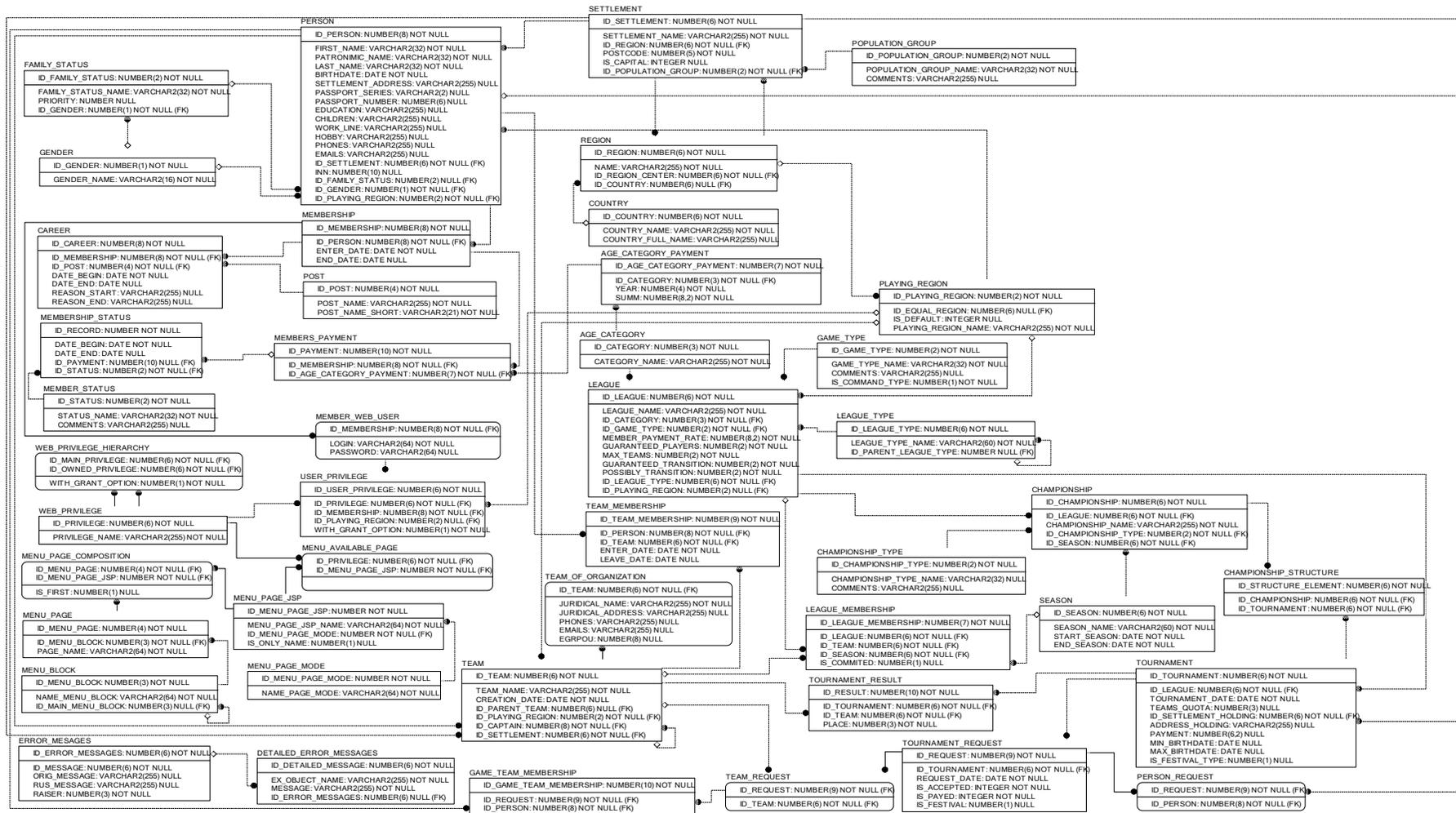


Рисунок Е.1 – Схема данных информационно-аналитической системы «Реестр Лиги украинских клубов интеллектуальных игр», спроектированная с применением методологии ГОСТов 34 группы

- MENU_PAGE_COMPOSITION (включение физических web-страниц в состав логических страниц меню навигации web-ориентированной ИАС);
- MENU_PAGE (описание логических пунктов меню навигации системы);
- MENU_BLOCK (группировка пунктов меню).

Остальные таблицы реализуют ФТ, определяющие приведенную в [134] схему ФС ИАС «Реестр ЛУК».

На рис. Е.2 приведено общесистемное АД ИАС «Реестр ЛУК», сформированное по результатам анализа существующих проектных решений и принятой схемы ФС системы, рассмотренной в [134].

В ходе уточнения ФТ к отдельным функциям ИАС «Реестр ЛУК» был уточнен и дополнен словарь терминов ПрО, приведенный в табл. Е.1. По результатам этого уточнения были скорректированы публикации и, соответственно, изменены представления этих ФТ на уровне знаний. Результаты коррекции представлений отдельных ФТ к ИАС «Реестр ЛУК» приведены на рис. Е.3-Е.12.

На рис. Е.13 приведен результат синтеза схемы данных по скорректированному АД ИАС «Реестр ЛУК». Здесь заливкой не выделены специализированные слабые типы сущностей; исходная схема данных была разбита на несколько фрагментов, реализованных в виде следующих витрин данных:

- APP_PERSON: таблицей фактов является PERSON (персона, лицо), производными от неё слабыми сущностями являются таблицы PERSON_JURIDICAL (юридическое лицо, организация) и PERSON_PHYSICAL (физическое лицо), таблицами измерений являются FAMILY_STATUS (семейный статус) и GENDER (пол);
- APP_CONTRACT: таблицей фактов является CONTRACT (контракт, факт занятия должности), таблицами измерений – POST (должность) и TYPE (тип должности);
- APP_PUBLICATION: таблицей фактов являются PUBLICATION (публикация), VOTE (голос, оценка, рейтинг), COMMENTARY (комментарий), FILES (файл), PUBLICATION_FILE (прикреплённый к публикации файл);

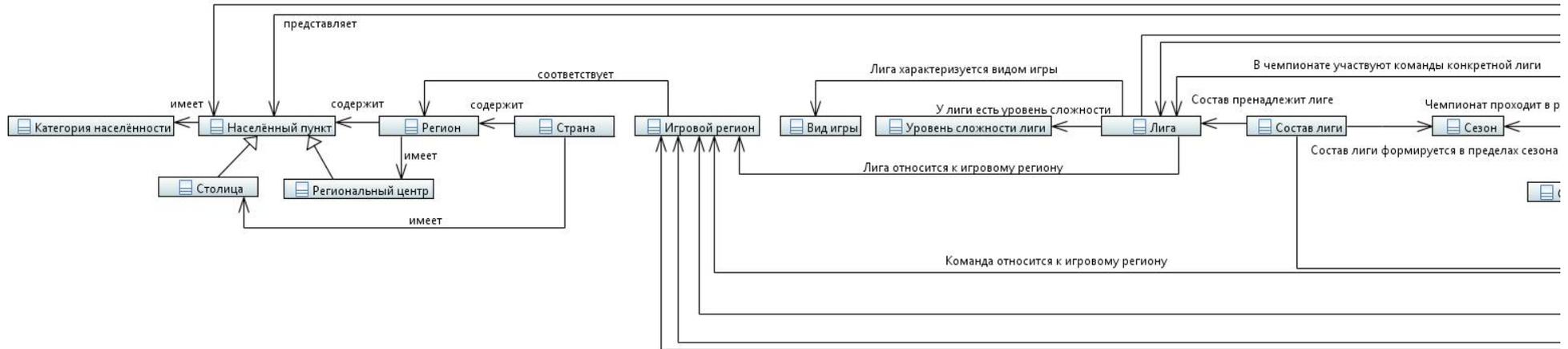


Рисунок Е.2 – Описание архитектуры информационно-аналитической системы «Реестр Лиги украинских клубов интеллектуальных игр» на основе существующих решений, часть 1

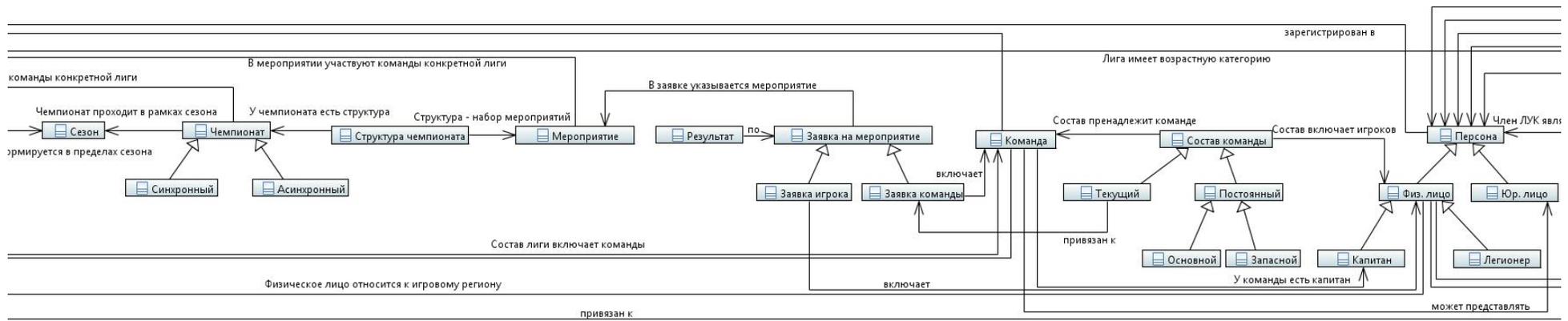


Рисунок Е.2 – Описание архитектуры информационно-аналитической системы «Реестр Лиги украинских клубов интеллектуальных игр» на основе существующих решений, часть 2



Рисунок Е.2 – Описание архитектуры информационно-аналитической системы «Реестр Лиги украинских клубов интеллектуальных игр» на основе существующих решений, лист 2, часть 3

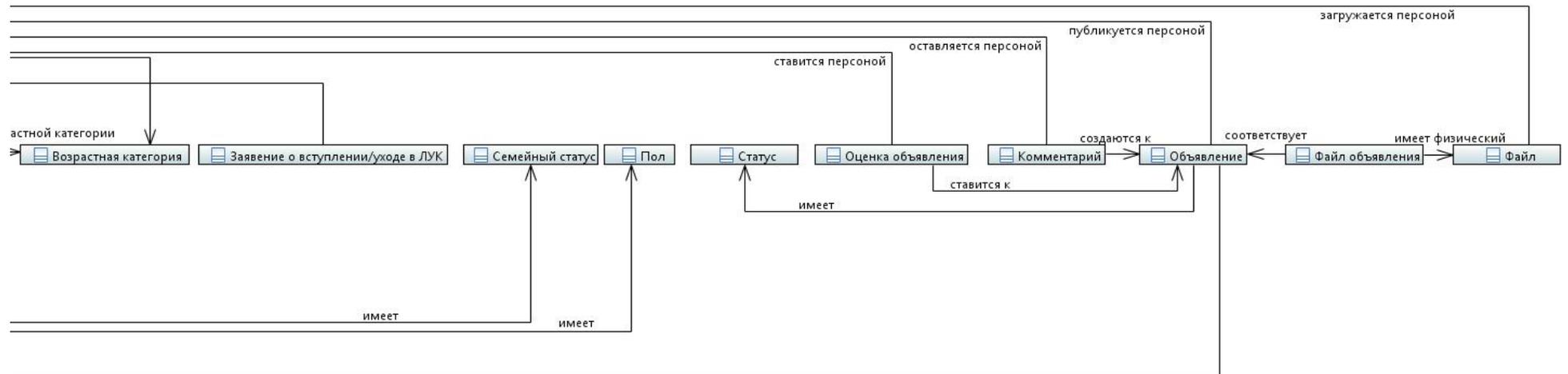


Рисунок Е.2 – Описание архитектуры информационно-аналитической системы «Реестр Лиги украинских клубов интеллектуальных игр» на основе существующих решений, лист 2, часть 4

Таблица Е.1 – Словарь терминов предметной области

№	Термин	Определение
1	2	3
1	Персона (лицо)	Физическое или юридическое лицо, принимающее участие в мероприятиях ВОО «ЛУК».
2	Физическое лицо (игрок)	Человек (носитель прав и обязанностей), который участвует в мероприятиях ЛУК.
3	Семейный статус	Статус физического лица, который определяет его семейное положение.
4	Пол	Половая принадлежность физического лица.
5	Юридическое лицо	Организация, предприятие и т.д., набирающее группу физических лиц для участия и представления его на мероприятиях ЛУК.
6	Команда	Объединение группы физических лиц для участия в мероприятиях ЛУК и достижения определённых целей.
7	Состав команды	Группа физических лиц, объединившихся в команду для участия в мероприятиях ЛУК и достижения определённых целей, руководство которой осуществляется капитаном.
8	Постоянный состав команды	Заявленный командой состав, игроки которого относительно постоянно участвуют в мероприятиях ЛУК. Делится на основной и запасной составы.
9	Основной состав команды	Заявленный командой состав, включающий игроков, которые регулярно участвуют в мероприятиях ЛУК.
10	Запасной состав команды	Заявленный командой состав, игроки которого участвуют в мероприятиях ЛУК при условии отсутствия кого-либо из игроков основного состава.
11	Текущий состав команды	Фактический состав команды на текущий момент времени (на текущем мероприятии). Может включать как членов постоянного состава команды, так и легионеров.
12	Капитан	Физическое лицо (игрок), который представляет интересы команды в лице подчинённой ему группы игроков.
13	Игровой регион	Регион страны, который территориально объединяет игроков и команды ЛУК для проведения мероприятий.
14	Мероприятие (этап, турнир)	Совокупность действий, объединённых одной общественно значимой задачей – проведением соревнования по интеллектуальным играм между игроками/командами.
15	Заявка на мероприятие (турнир)	Документ, выражающий желание игрока/команды принять участие в конкретном мероприятии ЛУК.

Продолжение таблицы Е.1

1	2	3
16	Заявка игрока	Документ, выражающий желание игрока принять участие в конкретном мероприятии ЛУК.
17	Заявка команды	Документ, выражающий желание команды принять участие в конкретном мероприятии ЛУК.
18	Результат	Результат игрока/команды, полученный после проведения мероприятия (количество набранных очков, занятое место и т.д.).
19	Чемпионат (сессия)	Объединение набора мероприятий ЛУК в рамках конкретной игры, лиги и сезона.
20	Синхронный чемпионат	Чемпионат, который допускает синхронное во времени использование игровых вопросов в мероприятиях других чемпионатов.
21	Асинхронный чемпионат	Чемпионат, который не допускает использование игровых вопросов его мероприятий в мероприятиях других чемпионатов.
22	Структура чемпионата	Набор мероприятий ЛУК в рамках чемпионата, упорядоченный во времени.
23	Сезон	Временные рамки проведения чемпионатов ЛУК.
24	Лига	Общественное объединение группы игроков/команд, обладающих определённым уровнем опыта и участвующих в мероприятиях одного уровня сложности, имеющее ограничение по количеству участников.
25	Состав лиги	Группа команд игроков/команд, которые обладают определённым уровнем опыта и в настоящий момент завоевали право на участие в мероприятиях определённого уровня сложности.
26	Уровень сложности лиги	Уровень, который характеризуется сложностью проводимых мероприятий (в частности, задаваемых вопросов) и опытом игроков/команд.
27	Вид игры	Вид интеллектуальной игры, по которой проводятся чемпионаты и мероприятия.
28	Членство в ЛУК	Принадлежность игрока к ассоциированным членам ВОО «ЛУК» на определённый срок.
29	Причина исключения	Причина, по которой физическое или юридическое лицо перестаёт быть членом ВОО «ЛУК».
30	Легионер	Игрок ЛУК, который участвует в мероприятии с определённой командой, но при этом не является членом её постоянного состава.
31	Заявление о вступлении в члены ЛУК	Документ, выражающий желание физического/юридического лица вступить в члены ВОО «ЛУК».

Продолжение таблицы Е.1

1	2	3
32	Должность	Первичная неделимая структурная единица в ВОО «ЛУК», замещаемая членом ЛУК (физическим лицом), отвечающим установленным требованиям и наделённым должностными полномочиями.
33	Руководящая должность	Должность, которая имеет управленческие полномочия.
34	Вакансия	Рабочее место или должность, на которую может быть принят новый работник.
35	Свободная вакансия	Незамещённое рабочее место или должность.
36	Занятая вакансия	Замещённое рабочее место или должность.
37	Контракт (трудовой)	Один из документов, отражающих условия трудовых взаимоотношений сотрудника с администрацией.
38	Возрастная категория	Категория со строгими возрастными ограничениями, которая используется для классификации игроков по возрасту.
39	Взнос	Плата за вступление в организацию или предоставление услуг по проведению мероприятий.
40	Уплата взноса	Возмещение определенной суммы денежных средств в виде платы за вступление в организацию или за предоставление услуг по проведению мероприятий.
41	Вступительный взнос	Взнос, который уплачивается один раз при вступлении в члены ЛУК.
42	Членский взнос	Взнос, который периодически уплачивается на развитие организации.
43	Страна	Территория, имеющая определённые национальные, климатические, культурные, исторические или политические границы.
44	Регион	Определённая территория, обладающая целостностью и взаимосвязью её составных элементов, которая используется в значении территориальной единицы государства.
45	Населённый пункт	Населённое людьми место (поселение) в пределах одного застроенного земельного участка (город, посёлок городского типа, село и пр.), обязательным признаком которого является постоянство использования его как места обитания.
46	Столица	Официальный главный город государства, в котором, как правило, располагаются высшие органы власти и управления.

Продолжение таблицы Е.1

1	2	3
47	Категория населённости	Количественные рамки численности населения в населённом пункте.
48	Объявление	Официальное извещение, которое несёт определённую информацию о проведении мероприятий и публикуется в сети интернет для обеспечения возможности её просмотра игроками ЛУК.
50	Комментарий	Рассуждения, пояснительные и критические замечания об информации в объявлении.
51	Статус	Состояние объявления.
52	Оценка объявления	Отношение пользователя к опубликованному объявлению.
53	Файл	Поименованная область на носителе информации.
54	Файл объявления	Файл, который прикреплен к объявлению, опубликованному в сети интернет.

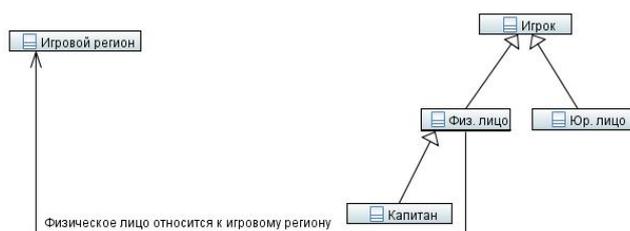


Рисунок Е.3 – Представление на уровне знаний требования к функции «Учёт игроков»



Рисунок Е.4 – Представление на уровне знаний требования к функции «Учёт команд»

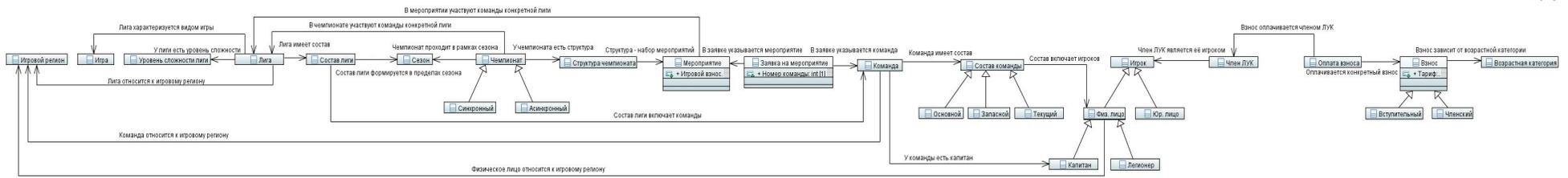


Рисунок Е.5 – Представление на уровне знаний требования к функции «Учёт поданных заявок на мероприятие»

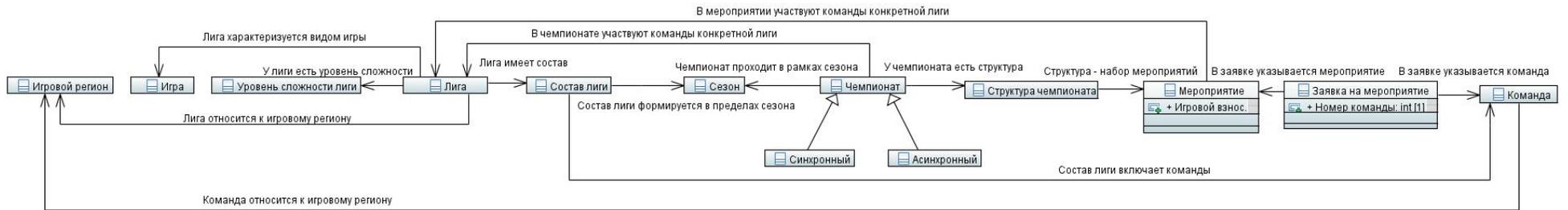


Рисунок Е.6 – Представление на уровне знаний требования к функции «Учёт уплаты игровых взносов»

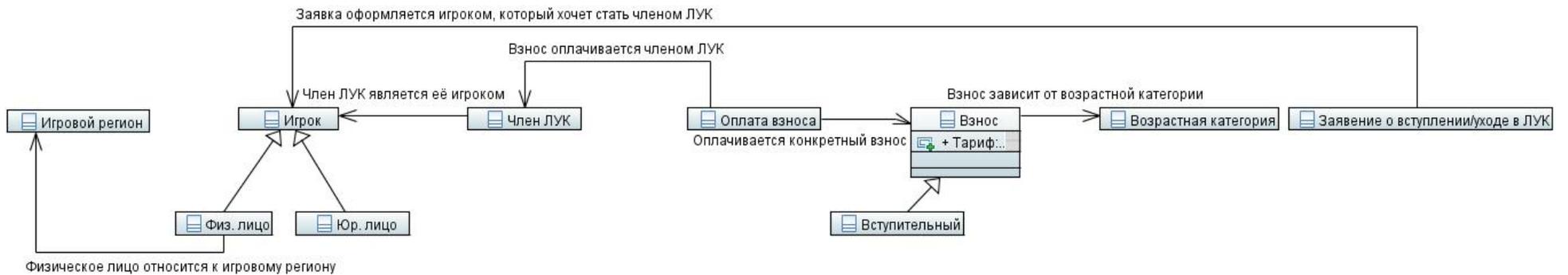


Рисунок Е.7 – Представление на уровне знаний требования к функции «Учёт членства ЛУК»



Рисунок Е.8 – Представление на уровне знаний требования к функции «Учёт уплаты вступительных и членских взносов»



Рисунок Е.9 – Представление на уровне знаний требования к функции «Учёт должностей, занимаемых членами ЛУК»

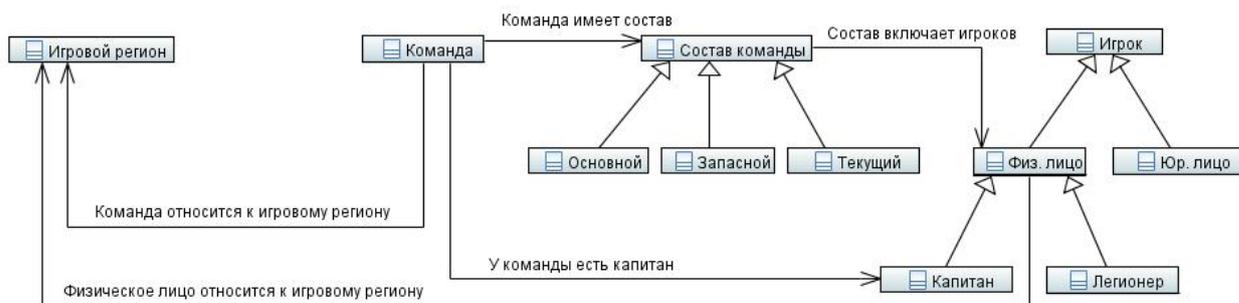


Рисунок Е.10 – Представление на уровне знаний требования к функции
«Учёт составов команд»



Рисунок Е.11 – Представление на уровне знаний требования к функции
«Учёт должностей ЛУК»



Рисунок Е.12 – Представление на уровне знаний требования к функции
«Учёт тарифов вступительных и членских взносов»

- APP_GEOGRAPHY: таблицей фактов в зависимости от ситуации могут выступать SETTLEMENT (населённый пункт), REGION (регион, область), COUNTRY (страна, государство); таблица POPULATION_GROUP_DICT (категория населённости) всегда является таблицей измерений;

- APP_PAYMENT: таблицей фактов является PAYMENT (платёж), таблицей измерений – RATE (тариф);

- APP_MEMBERSHIP: таблицами фактов являются MEMBERSHIP

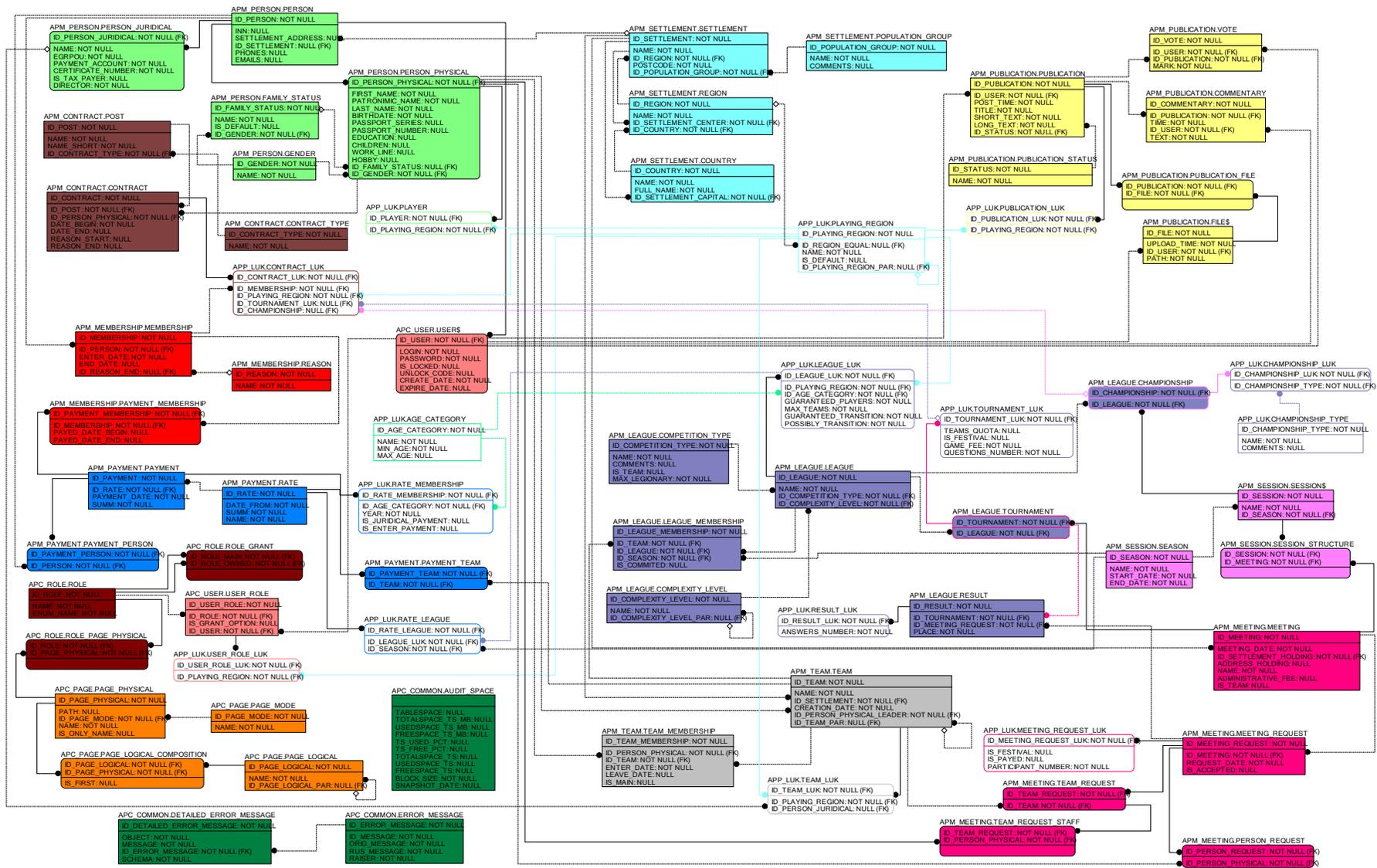


Рисунок Е.13 – Схема данных информационно-аналитической системы «Реестр Лиги украинских клубов интеллектуальных игры, синтезированная на основе скорректированного описания архитектуры

(членство) и MEMBER_PAYMENT (членский взнос, слабая сущность, производная от PAYMENT), таблицей измерений является REASON. При рассмотрении членских взносов таблица MEMBERSHIP выступает в роли таблицы измерения по отношению к таблице фактов MEMBER_PAYMENT;

- APP_TEAM: таблицей фактов является TEAM (команда, которая содержит иерархическую связь, отражающую преемственность команд), таблицей измерений – TEAM_MEMBERSHIP (состав команды как объединения игроков);

- APP_SESSION: таблицами фактов являются SEASON (сезон, период проведения турниров) и SESSION (сессия, чемпионат). При этом таблица SEASON выступает таблицей измерений по отношению к таблице SESSION, совместно с таблицами TYPE (тип чемпионата) и SESSION_STRUCTURE (состав чемпионата в виде календаря турниров);

- APP_MEETING: таблицами фактов являются MEETING (мероприятие, турнир) и MEETING_REQUEST (запрос на участие в турнире). При этом таблица MEETING выступает таблицей измерений по отношению к таблице MEETING_REQUEST. Таблицы TEAM_REQUEST (командная заявка) и PERSON_REQUEST (персональная заявка) являются слабыми сущностями, производными от MEETING_REQUEST, а таблица MEETING_TEAM_MEMBERSHIP (заявленный состав команды) является таблицей измерений для таблицы TEAM_REQUEST;

- APP_LEAGUE: таблицами фактов являются LEAGUE (лига) и LEAGUE_MEMBERSHIP (членство команды в лиге), MEETING_LEAGUE (мероприятие определённой лиги, слабая сущность, производная от MEETING). При этом таблица LEAGUE выступает таблицей измерений по отношению к таблицам LEAGUE_MEMBERSHIP и MEETING_LEAGUE. Таблицы TYPE_DICT (тип лиги) и COMPLEXITY_LEVEL (уровень сложности) являются таблицами измерений;

- APP_USER: таблицей фактов является USERS (пользователь системы), таблицей измерений – USER_PRIVILEGE (привилегии пользователя в web-базированной ИАС, которым назначаются права доступа к отдельным физическим web-страницам, реализующим определённые функциональные операции ИТ-сервисов системы);

- APP_PRIVILEGE: таблицей фактов является PRIVILEGE (привилегия, роль), таблицами измерений – PRIVILEGE_GRANT (назначение вложенных ролей) и PRIVILEGE_AVAILABLE_PAGE (назначение прав доступа ролей к страницам web-приложения);

- APP_PAGE: таблицами фактов являются PAGE_LOGICAL (логическая страница), PAGE_PHYSICAL (физическая страница). Данные таблицы не являются производными общего понятия «Страница», поскольку логическая страница рассматривается в качестве пункта меню, а физическая в качестве адреса в структуре web-приложения, по которому осуществляется физический доступ к одной или несколькими функциональным операциям ИТ-сервиса. Таблицами измерения являются PAGE_MODE_DICT (тип функциональных операций, реализуемых страницей, например, редактирование или просмотр данных) и PAGE_COMPOSITION (объединение нескольких физических страниц в логическую страницу);

- APP_CORE: схема данных, применяемая для реализации нефункциональных требований к ИАС, описанных выше, для корректного отображения пользователю ошибок, возникающих при работе с БД (таблица фактов ERROR_MESSAGE и таблица измерений DETAILED_ERROR_MESSAGE).

Для мониторинга использования дискового пространства сервера БД ИАС применяется таблица фактов AUDIT_SPACE.

Исключением являются таблицы схемы APP_LUK. Данная схема реализована не как витрина данных и содержит таблицы, которые либо являются слабыми сущностями, дополняющими абстрактные таблицы фактов описанных выше витрин данных свойствами, характерными для Про ВОО «ЛУК», либо дополнительными таблицами измерений:

- PERSON_PHYSICAL_LUK (игрок, слабая сущность, производная от PERSON_PHYSICAL, расширена свойством «игровой регион»);

- PUBLICATION_LUK (публикация, слабая сущность, производная от PUBLICATION, расширена свойством «игровой регион»);

- USER_PRIVILEGE_LUK (привилегия/роль, слабая сущность, произ-

водная от PRIVILEG, расширена свойством «игровой регион», применительно к которому назначается привилегия);

- PLAYING_REGION (игровой регион) – таблица, дополняющая схему данных APP_GEOGRAPHY административно-территориальным делением ВОО «ЛУК»;

- AGE_CATEGORY (возрастная категория) таблица, дополняющая схему данных APP_PERSON возрастными категориями физических лиц;

- CONTRACT_LUK (контракт, слабая сущность, производная от CONTRACT, расширена свойствами, отражающими привязку должностей ВОО «ЛУК» к игровым регионам и наличие временных контрактов, действующих на время проведения одного чемпионата или отдельного турнира);

- RATE_LUK (тариф, слабая сущность, производная от RATE, расширена свойствами, отражающими зависимость тарифов членских и вступительных взносов ВОО «ЛУК» для физических и юридических лиц от возрастных категорий игроков и их принадлежности к лигам, а также неизменность размеров членских взносов в течение календарного года);

- TEAM_LUK (команда, слабая сущность, производная от TEAM, отражает привязку команд к населённым пунктам и игровым регионам, а также наличие команд, спонсируемых организациями);

- MEETING_LUK (турнир/игра, слабая сущность, производная от MEETING_LEAGUE, отражает лимит на количество участников мероприятия, а также размеры игровых взносов и признак зачета результатов мероприятия в итог чемпионата);

- MEETING_REQUEST_LUK (заявка, слабая сущность, производная от MEETING_REQUEST, отражает результаты участия в турнире и признак их влияния на общий зачет чемпионата);

- LEAGUE_LUK (лига, слабая сущность, производная от LEAGUE, отражает классификацию лиг ВОО «ЛУК» по игровым регионам, возрастным категориям, а также особенности переходов команд между лигами по результатам чемпионатов).

На рис. Е.14–Е.18 приведены примеры АД ПО ИАС «Реестр ЛУК» в виде диаграмм классов. Эти примеры отражают архитектурные стили разработки ПО, реализованные с применением паттернов DAO и MVC.

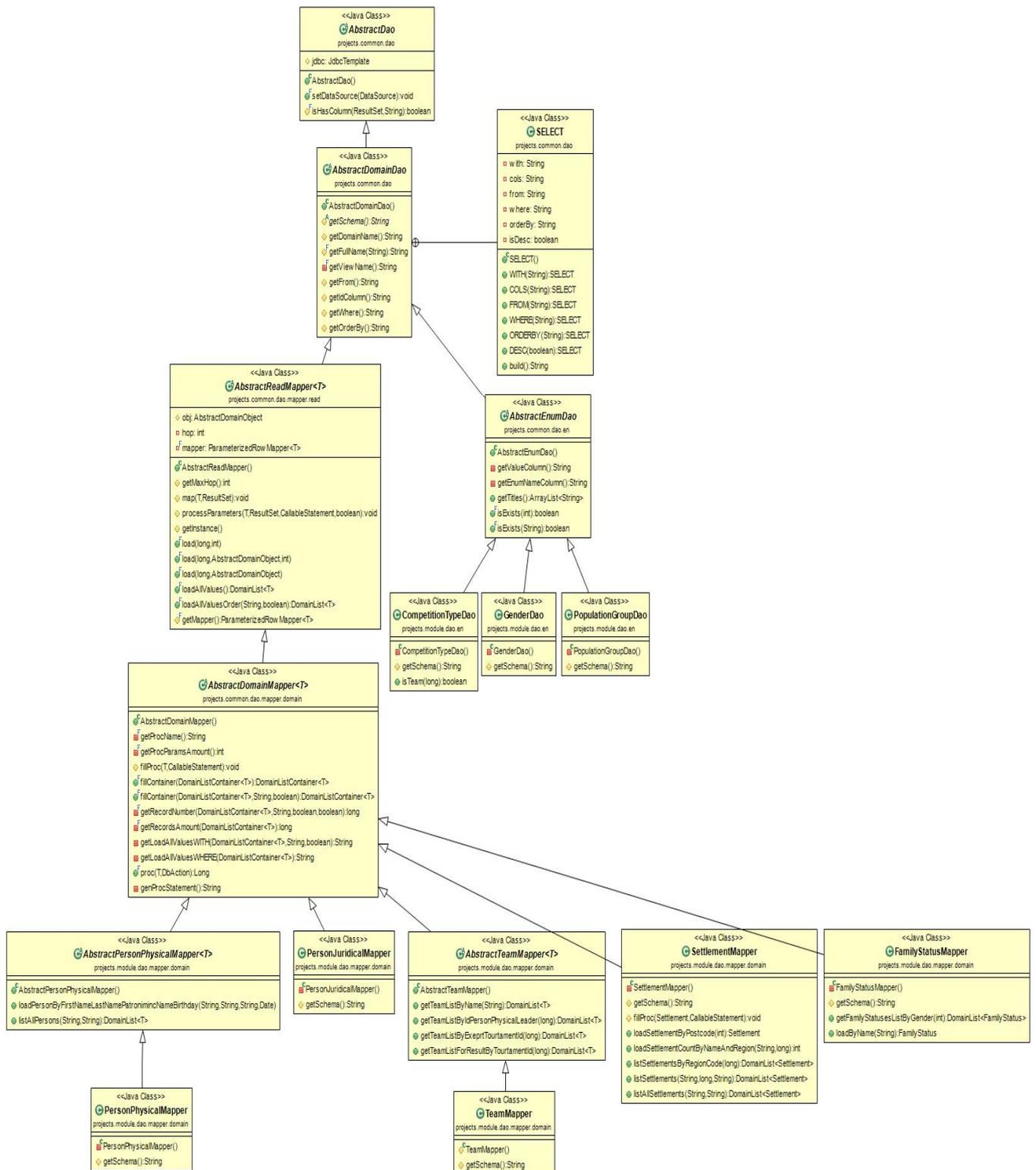


Рисунок Е.14 – Диаграмма классов, отражающая особенности применения паттерна проектирования DAO для разработки программного обеспечения, реализующего функциональные требования к информационно-аналитической системе «Реестр Лиги украинских клубов» с абстрагированными от конкретных предметных областей наборами классов

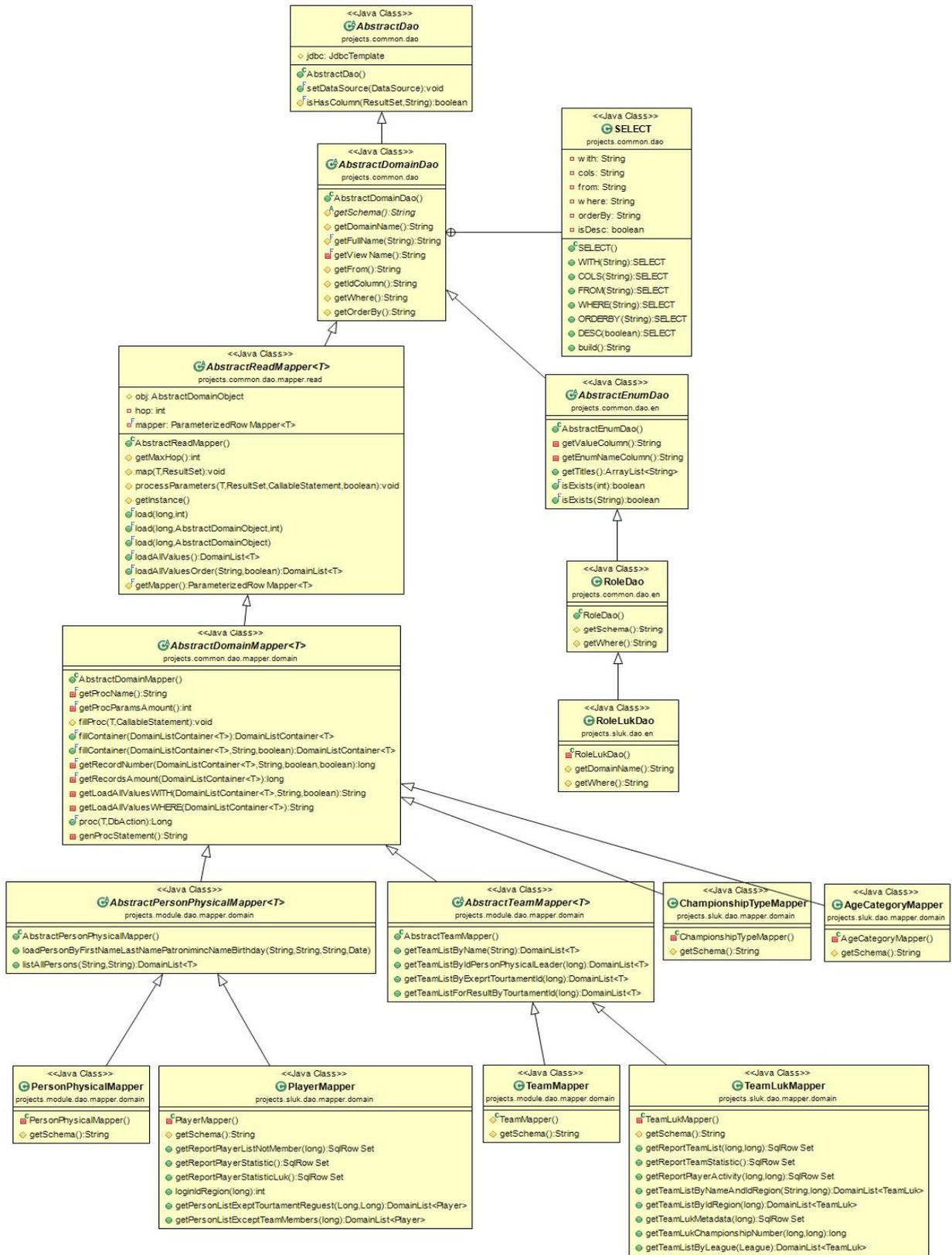


Рисунок Е.15 – Диаграмма классов, отражающая особенности применения паттерна проектирования DAO для разработки программного обеспечения, реализующего специфичные для предметной области функциональные требования

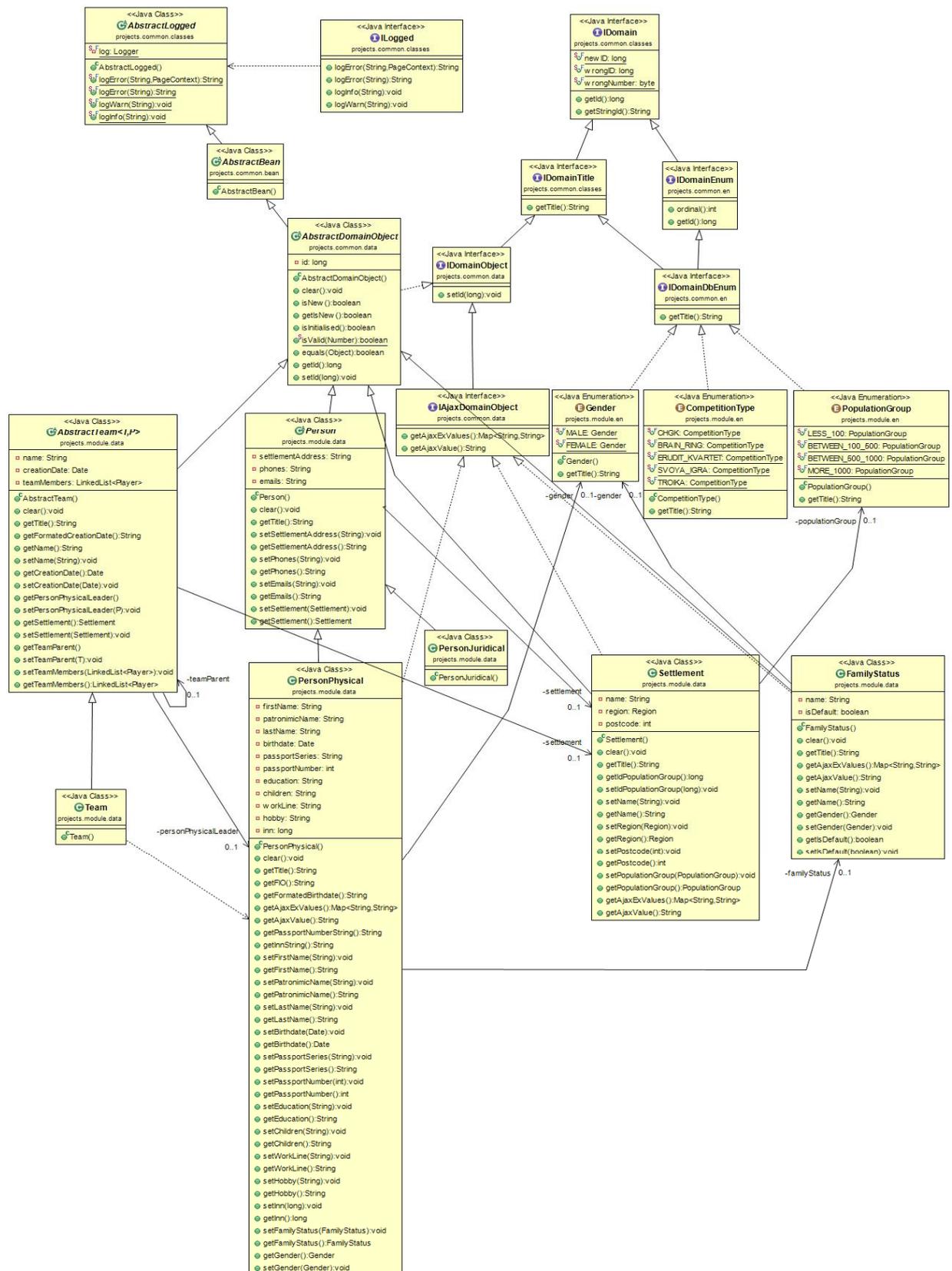


Рисунок Е.17 – Диаграмма, отражающая бизнес-классы программного обеспечения, реализующего функциональные требования к информационно-аналитической системе «Реєстр Лиги українських клубів» с абстрагированными от конкретных предметных областей наборами классов

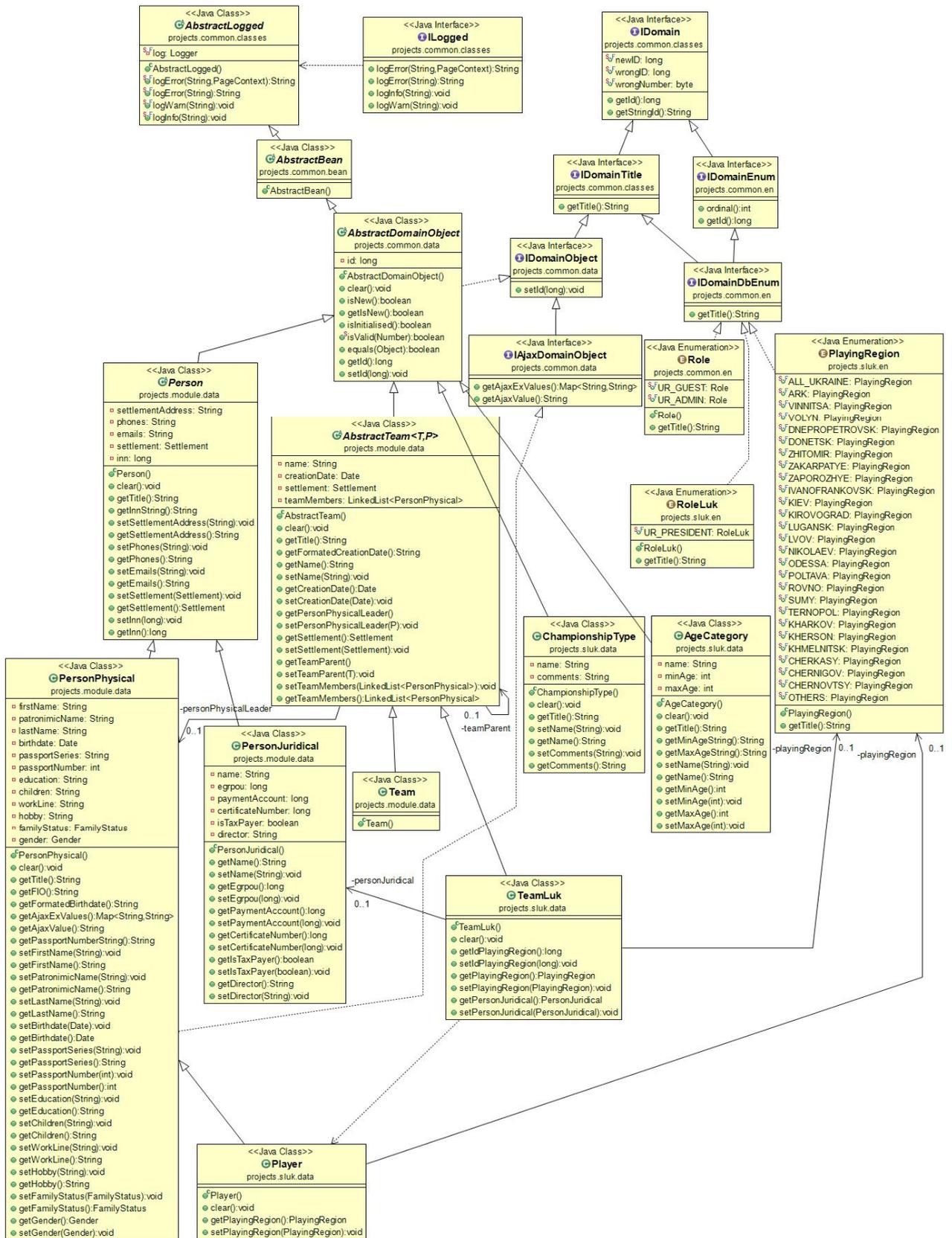


Рисунок Е.18 – Диаграмма, отражающая бизнес-классы программного обеспечения, реализующего специфичные для предметной области функциональные требования

жее с приведенными на рис. Е.4 классами расположение в структуре и описываются аналогичным образом.

При реализации ПО используются технологии JSP (Java Server Pages) и Enterprise JavaBean, поэтому классы, реализующие модель, являются одновременно и реализацией технологии JavaBean. Данные классы имеют пустой конструктор и обязательные методы `get()` и `set()`. Они не реализуют практически никакой бизнес-логики и являются только структурами данных для создания объектов, имеющих ряд свойств, которые инициализируются на определенных jsp-страницах. Созданные объекты данных классов хранят своё состояние и используются для передачи информации между jsp-страницами в процессе работы пользователя в системе с использованием http-запросов или http-сессии, элементами которых экземпляры классов JavaBean и являются.

Кроме того, при разработке ПО используется фреймворк программирования «Spring», с помощью которого используется ряд реализованных функций для работы с драйвером JDBC.

Фреймворк Spring позволяет с помощью XML-файлов автоматически конфигурировать соединение с БД, создать ряд хранящихся в памяти фабрик (factory) и методов, которые являются статическими объектами и реализованы с использованием паттерна программирования «Singleton». Классы, реализованные с использованием данного паттерна, могут иметь только один объект, который создается на этапе запуска web-сервера и хранится в контексте web-приложения под управлением фреймворка «Spring». Обращаясь к этому контексту, всегда можно получить доступ к singleton-классу, который является фабрикой, возвращающей сервисы и классы «Mapper».

Инициализация web-приложения в целом и его конфигурирование также осуществляются с применением конфигурационных XML-файлов и фреймворка «Spring».

Внутри класса «Mapper» реализован внутренний класс (inner Class) «SELECT», который разработан с использованием паттерна программирования «Builder», что позволяет ему посредством последовательного вызова методов выстраивать объект, последовательно инициализируя его свойства.

Используются также библиотеки java-тегов и jsp-тегов.

ПРИЛОЖЕНИЕ Ж
ПРОМЕЖУТОЧНЫЕ РЕЗУЛЬТАТЫ ВЫПОЛНЕНИЯ
ИНФОРМАЦИОННОЙ ТЕХНОЛОГИИ

Результат Этапа 2 метода синтеза вариантов АД ФМ БТ приведен в табл. Ж.1.

Таблица Ж.1 – Результат выполнения Этапа 2 метода синтеза вариантов описаний архитектуры функционального модуля безопасности труда

Номер ИТ-услуги	Общесистемные представления, описывающие ИТ-услугу	Значение $S(IT_{acm_j})$ для ИТ-услуги	Значение $W(IT_{acm_j})$ для ИТ-услуги
1	K_1^{fIS}	7	7
2	K_2^{fIS}	12	12
3	K_3^{fIS}	6	6
4	K_4^{fIS}	8	8
5	K_5^{fIS}	28	28
Результат расчета функции $Profit(IT_{acm}, r)$			
$ IT_{acm_j} = 5$	$r = 2$	$Profit(IT_{acm}, r) = 0,110714$	

Результаты выполнения итераций Этапа 3 метода синтеза вариантов АД приведены в табл. Ж.2.

Исходные данные, на основе которых проводилась оценка объема работ по созданию вариантов ОРА ФМ БТ методом ОТ, приведены в табл. Ж.3.

Таблица Ж.2 – Результат выполнения итераций Этапа 3 метода синтеза вариантов описаний архитектуры функционального модуля безопасности труда

Номер ИТ-услуги	Общесистемные представления, описывающие ИТ-услугу	Значение $S(IT_{act_j})$ для ИТ-услуги	Значение $W(IT_{act_j})$ для ИТ-услуги
1	–	–	–
2	$\{K_1^{fIS}, K_2^{fIS}\}$	19	16
3	K_3^{fIS}	6	6
4	K_4^{fIS}	8	8
Номер ИТ-услуги	Общесистемные представления, описывающие ИТ-услугу	Значение $S(IT_{act_j})$ для ИТ-услуги	Значение $W(IT_{act_j})$ для ИТ-услуги
5	K_5^{fIS}	28	28
Результат расчета функции $Pr\ ofit(IT_{act}, r)$ на итерации 1			
$ IT_{act_j} = 5$	$r = 2, \varepsilon = 0,0110714$	$Pr\ ofit(IT_{act}, r) = 0,095164$ (отброшен)	
1	–	–	–
2	K_2^{fIS}	12	12
3	$\{K_1^{fIS}, K_3^{fIS}\}$	13	10
4	K_4^{fIS}	8	8
5	K_5^{fIS}	28	28
Результат расчета функции $Pr\ ofit(IT_{act}, r)$ на итерации 2			
$ IT_{act_j} = 5$	$r = 2, \varepsilon = 0,0110714$	$Pr\ ofit(IT_{act}, r) = 0,10081$ (зафиксирован в результате выполнения Шага 3.10)	
1	–	–	–
2	K_2^{fIS}	12	12
3	K_3^{fIS}	6	6
4	$\{K_1^{fIS}, K_4^{fIS}\}$	15	12
5	K_5^{fIS}	28	28

Продолжение табл. Ж.2

Номер ИТ-услуги	Общесистемные представления, описывающие ИТ-услугу	Значение $S(IT_{acm_j})$ для ИТ-услуги	Значение $W(IT_{acm_j})$ для ИТ-услуги
Результат расчета функции $Profit(IT_{acm}, r)$ на итерации 3			
$ IT_{acm_j} = 5$	$r = 2, \varepsilon = 0,0110714$	$Profit(IT_{acm}, r) = 0,09881$ (отброшен)	
1	–	–	–
2	K_2^{fIS}	12	12
3	K_3^{fIS}	6	6
4	K_4^{fIS}	8	8
5	$\{K_1^{fIS}, K_5^{fIS}\}$	35	32
Результат расчета функции $Profit(IT_{acm}, r)$ на итерации 4			
$ IT_{acm_j} = 5$	$r = 2, \varepsilon = 0,0110714$	$Profit(IT_{acm}, r) = 0,088672$ (отброшен)	
1	$\{K_1^{fIS}, K_2^{fIS}\}$	19	16
2	–	–	–
3	K_3^{fIS}	6	6
4	K_4^{fIS}	8	8
5	K_5^{fIS}	28	28
Результат расчета функции $Profit(IT_{acm}, r)$ на итерации 5			
$ IT_{acm_j} = 5$	$r = 2, \varepsilon = 0,0110714$	$Profit(IT_{acm}, r) = 0,095164$ (отброшен)	
1	K_1^{fIS}	7	7
2	–	–	–
3	$\{K_2^{fIS}, K_3^{fIS}\}$	18	18
4	K_4^{fIS}	8	8
5	K_5^{fIS}	28	28
Результат расчета функции $Profit(IT_{acm}, r)$ на итерации 6			
$ IT_{acm_j} = 5$	$r = 2, \varepsilon = 0,0110714$	$Profit(IT_{acm}, r) = 0,082937$ (отброшен)	
1	K_1^{fIS}	7	7
2	–	–	–

Продолжение табл. Ж.2

Номер ИТ-услуги	Общесистемные представления, описывающие ИТ-услугу	Значение $S(IT_{act_j})$ для ИТ-услуги	Значение $W(IT_{act_j})$ для ИТ-услуги
3	K_3^{fIS}	6	6
4	$\{K_2^{fIS}, K_4^{fIS}\}$	20	20
5	K_5^{fIS}	28	28
Результат расчета функции $Profit(IT_{act}, r)$ на итерации 7			
$ IT_{act_j} = 5$	$r = 2, \varepsilon = 0,0110714$	$Profit(IT_{act}, r) = 0,089048$ (отброшен)	
1	K_1^{fIS}	7	7
2	–	–	–
3	K_3^{fIS}	6	6
4	K_4^{fIS}	8	8
5	$\{K_2^{fIS}, K_5^{fIS}\}$	40	31
Результат расчета функции $Profit(IT_{act}, r)$ на итерации 8			
$ IT_{act_j} = 5$	$r = 2, \varepsilon = 0,0110714$	$Profit(IT_{act}, r) = 0,103554$ (зафиксирован в результате выполнения Шага 3.10)	
1	$\{K_1^{fIS}, K_3^{fIS}\}$	13	10
2	K_2^{fIS}	12	12
3	–	–	–
4	K_4^{fIS}	8	8
5	K_5^{fIS}	28	28
Результат расчета функции $Profit(IT_{act}, r)$ на итерации 9			
$ IT_{act_j} = 5$	$r = 2, \varepsilon = 0,0110714$	$Profit(IT_{act}, r) = 0,10081$ (зафиксирован в результате выполнения Шага 3.10)	
1	K_1^{fIS}	7	7
2	$\{K_2^{fIS}, K_3^{fIS}\}$	18	18
3	–	–	–
4	K_4^{fIS}	8	8

Продолжение табл. Ж.2

Номер ИТ-услуги	Общесистемные представления, описывающие ИТ-услугу	Значение $S(IT_{асм_j})$ для ИТ-услуги	Значение $W(IT_{асм_j})$ для ИТ-услуги
5	K_5^{fIS}	28	28
Результат расчета функции $Profit(IT_{асм}, r)$ на итерации 10			
$ IT_{асм_j} = 5$	$r = 2, \varepsilon = 0,0110714$	$Profit(IT_{асм}, r) = 0,082937$ (отброшен)	
1	K_1^{fIS}	7	7
2	K_2^{fIS}	12	12
3	–	–	–
4	$\{K_3^{fIS}, K_4^{fIS}\}$	14	8
5	K_5^{fIS}	28	28
Результат расчета функции $Profit(IT_{асм}, r)$ на итерации 11			
$ IT_{асм_j} = 5$	$r = 2, \varepsilon = 0,0110714$	$Profit(IT_{асм}, r) = 0,139881$ (зафиксирован в результате выполнения Шага 3.9, $m = 1$)	
1	–	–	–
2	$\{K_1^{fIS}, K_2^{fIS}\}$	19	16
–	–	–	–
4	$\{K_3^{fIS}, K_4^{fIS}\}$	14	8
5	K_5^{fIS}	28	28
Результат расчета функции $Profit(IT_{асм}, r)$ на итерации 12			
$ IT_{асм_j} = 4$	$r = 2, \varepsilon = 0,0139881$	$Profit(IT_{асм}, r) = 0,12433$ (отброшен)	
1	–	–	–
2	K_2^{fIS}	12	12
–	–	–	–
4	$\{K_1^{fIS}, \{K_3^{fIS}, K_4^{fIS}\}\}$	21	12
5	K_5^{fIS}	28	28
Результат расчета функции $Profit(IT_{асм}, r)$ на итерации 13			
$ IT_{асм_j} = 4$	$r = 2, \varepsilon = 0,0139881$	$Profit(IT_{асм}, r) = 0,11131$ (отброшен)	
1	–	–	–

Продолжение табл. Ж.2

Номер ИТ-услуги	Общесистемные представления, описывающие ИТ-услугу	Значение $S(IT_{асм_j})$ для ИТ-услуги	Значение $W(IT_{асм_j})$ для ИТ-услуги
2	K_2^{fIS}	12	12
–	–	–	–
4	$\{K_3^{fIS}, K_4^{fIS}\}$	14	8
5	$\{K_1^{fIS}, K_5^{fIS}\}$	35	32
Результат расчета функции $Profit(IT_{асм}, r)$ на итерации 14			
$ IT_{асм_j} = 4$	$r = 2, \varepsilon = 0,0139881$	$Profit(IT_{асм}, r) = 0,117839$ (отброшен)	
1	$\{K_1^{fIS}, K_2^{fIS}\}$	19	16
2	–	–	–
–	–	–	–
4	$\{K_3^{fIS}, K_4^{fIS}\}$	14	8
5	K_5^{fIS}	35	32
Результат расчета функции $Profit(IT_{асм}, r)$ на итерации 15			
$ IT_{асм_j} = 4$	$r = 2, \varepsilon = 0,0139881$	$Profit(IT_{асм}, r) = 0,12433$ (отброшен)	
1	K_1^{fIS}	7	7
2	–	–	–
–	–	–	–
4	$\{K_2^{fIS}, \{K_3^{fIS}, K_4^{fIS}\}\}$	26	20
5	K_5^{fIS}	35	32
Результат расчета функции $Profit(IT_{асм}, r)$ на итерации 16			
$ IT_{асм_j} = 4$	$r = 2, \varepsilon = 0,0139881$	$Profit(IT_{асм}, r) = 0,074714$ (отброшен)	
1	K_1^{fIS}	7	7
2	–	–	–
–	–	–	–
4	$\{K_3^{fIS}, K_4^{fIS}\}$	14	8
5	$\{K_2^{fIS}, K_5^{fIS}\}$	40	31

Продолжение табл. Ж.2

Номер ИТ-услуги	Общесистемные представления, описывающие ИТ-услугу	Значение $S(IT_{асм_j})$ для ИТ-услуги	Значение $W(IT_{асм_j})$ для ИТ-услуги
Результат расчета функции $Pr ofit(IT_{асм}, r)$ на итерации 17			
$ IT_{асм_j} = 4$	$r = 2, \varepsilon = 0,0139881$	$Pr ofit(IT_{асм}, r) = 0,132721$ (зафиксирован в результате выполнения Шага 3.10)	
1	$\{K_1^{fIS}, \{K_3^{fIS}, K_4^{fIS}\}\}$	21	12
2	K_2^{fIS}	12	12
–	–	–	–
4	–	–	–
5	K_5^{fIS}	28	28
Результат расчета функции $Pr ofit(IT_{асм}, r)$ на итерации 18			
$ IT_{асм_j} = 4$	$r = 2, \varepsilon = 0,0139881$	$Pr ofit(IT_{асм}, r) = 0,11131$ (отброшен)	
1	K_1^{fIS}	7	7
2	$\{K_2^{fIS}, \{K_3^{fIS}, K_4^{fIS}\}\}$	26	20
–	–	–	–
4	–	–	–
5	K_5^{fIS}	28	28
Результат расчета функции $Pr ofit(IT_{асм}, r)$ на итерации 19			
$ IT_{асм_j} = 4$	$r = 2, \varepsilon = 0,0139881$	$Pr ofit(IT_{асм}, r) = 0,074714$ (отброшен)	
1	K_1^{fIS}	7	7
2	K_2^{fIS}	12	12
–	–	–	–
4	–	–	–
5	$\{K_3^{fIS}, \{K_4^{fIS}, K_5^{fIS}\}\}$	42	28
Результат расчета функции $Pr ofit(IT_{асм}, r)$ на итерации 20			
$ IT_{асм_j} = 4$	$r = 2, \varepsilon = 0,0139881$	$Pr ofit(IT_{асм}, r) = 0,077381$ (отброшен)	
1	$\{K_1^{fIS}, K_5^{fIS}\}$	35	32
2	K_2^{fIS}	12	12

Продолжение табл. Ж.2

Номер ИТ-услуги	Общесистемные представления, описывающие ИТ-услугу	Значение $S(IT_{act_j})$ для ИТ-услуги	Значение $W(IT_{act_j})$ для ИТ-услуги
4	$\{K_3^{fIS}, K_4^{fIS}\}$	14	8
5	–	–	–
Результат расчета функции $Profit(IT_{act}, r)$ на итерации 21			
$ IT_{act_j} = 4$	$r = 2, \varepsilon = 0,0139881$	$Profit(IT_{act}, r) = 0,117839$ (отброшен)	
1	K_1^{fIS}	7	7
2	$\{K_2^{fIS}, K_5^{fIS}\}$	40	31
–	–	–	–
4	$\{K_3^{fIS}, K_4^{fIS}\}$	14	8
5	–	–	–
Результат расчета функции $Profit(IT_{act}, r)$ на итерации 22			
$ IT_{act_j} = 4$	$r = 2, \varepsilon = 0,0139881$	$Profit(IT_{act}, r) = 0,132721$ (зафиксирован в результате выполнения Шага 3.10)	
1	K_1^{fIS}	7	7
2	K_2^{fIS}	12	12
–	–	–	–
4	$\{K_3^{fIS}, \{K_4^{fIS}, K_5^{fIS}\}\}$	42	28
5	–	–	–
Результат расчета функции $Profit(IT_{act}, r)$ на итерации 23			
$ IT_{act_j} = 4$	$r = 2, \varepsilon = 0,0139881$	$Profit(IT_{act}, r) = 0,077381$ (отброшен)	

Таблица Ж.3 – Исходные данные для оценки объема работ по реализации ИТ-услуг варианта описания рациональной архитектуры функционального модуля безопасности труда

Оцениваемый объект	Значение $ \{OntPD_{ijm}\} $	Значение $ \{FR_{OntPD_{ijm}}\} $ (для экранов и отчетов) или $ \{ATTR_FR_{OntPD_{im}}\} $ (для модулей)	Значение f
1	2	3	4
ИТ-услуга, реализующая первое функциональное требование			
Экран «Справочные данные о предприятии»	1	2	1
Экран «Справочные данные о процессах предприятия»	2	3	1
Экран «Справочные данные о работах процесса предприятия»	2	4	1
Экран «Данные о предприятии»	1	2	1
Экран «Данные о процессах предприятия»	2	3	1
Экран «Данные о работах процесса предприятия»	2	4	1
Экран «Данные о наблюдаемом предприятии, его процессах и работах»	2	4	1
Экран «Справочные данные о предприятии, его процессах и работах»	2	4	1
Модуль	2	67	15
ИТ-услуга, реализующая второе и пятое функциональные требования			
Экран «Справочные данные о сотрудниках предприятия»	5	6	2
Экран «Данные о наблюдаемых сотрудниках предприятия»	5	6	2
Экран «Данные о наблюдаемом предприятии»	1	2	1
Экран «Данные о сотрудниках предприятия»	5	6	2

Продолжение табл. Ж.3

1	2	3	4
Экран «Данные о наблюдаемом предприятии»	1	2	1
Экран «Данные о предприятии»	1	2	1
Экран «Выбор процесса предприятия»	2	3	1
Экран «Выбор работы процесса предприятия»	2	4	1
Экран «Выбор сотрудников, выполняющих процесс или работу»	6	8	3
Экран «Перечень процессов предприятия»	2	3	1
Экран «Перечень работ процессов предприятия»	2	4	1
Экран «Значения даты и времени начала и конца смены»	1	1	1
Экран «Значения параметров состояния сотрудников, выполняющих процесс или работу»	5	7	2
Экран «Значения диапазона изменения параметров состояния сотрудника»	5	7	2
Экран «Перечень состояний сотрудника»	1	3	1
Экран «Перечень ВПФ процесса или работы»	2	3	1
Экран «Массив значений ВПФ процесса или работы»	2	4	1
Экран «Значения даты и времени начала и окончания смены»	1	1	1
Экран «Значения параметра состояния сотрудника, выполняющего процесс или работу»	6	9	3
Отчет «Прогнозируемое состояние сотрудника на момент времени смены»	2	4	1
Модуль	9	263	15
ИТ-услуга, выполняющая третье и четвертое функциональные требования			

Продолжение табл. Ж.3

1	2	3	4
Экран «Выбор процесса предприятия»	1	1	1
Экран «Выбор работы процесса предприятия»	1	2	1
Экран «Описание ВПФ процесса или работы предприятия»	2	3	1
Экран «Перечень ВПФ процесса или работы предприятия»	2	3	1
Экран «Перечень процессов предприятия»	1	1	1
Экран «Перечень работ процессов предприятия»	1	2	1
Экран «Перечень ВПФ предприятия»	2	3	1
Экран «Выбор наблюдаемого ВПФ процесса или работы предприятия»	2	3	1
Экран «Значения результатов наблюдений ВПФ процесса или работы предприятия»	2	4	1
Экран «Массив значений наблюдавшегося ВПФ»	2	4	1
Модуль	2	27	10