

Федеральное государственное бюджетное учреждение науки
Санкт-Петербургский институт информатики и автоматизации
Российской академии наук
(СПИИРАН)

На правах рукописи



Охтилев Павел Алексеевич

**АЛГОРИТМЫ И ОНТОЛОГИЧЕСКИЕ МОДЕЛИ
ИНФОРМАЦИОННО-АНАЛИТИЧЕСКОЙ ПОДДЕРЖКИ
ПРОЦЕССОВ СОЗДАНИЯ И ПРИМЕНЕНИЯ
КОСМИЧЕСКИХ СРЕДСТВ**

Специальность 05.13.01 – Системный анализ, управление и обработка
информации (технические системы)

Диссертация на соискание ученой степени
кандидата технических наук

Научный руководитель:
Заслуженный деятель науки РФ,
Лауреат премии Правительства РФ
в области науки и техники,
доктор технических наук, профессор
Соколов Борис Владимирович

Санкт-Петербург – 2019

СОДЕРЖАНИЕ

Введение	5
1. Системный анализ состояния исследований задачи информационно-аналитической поддержки жизненного цикла космических средств как сложных объектов.....	16
1.1. Характеристика задачи автоматизации информационно-аналитической поддержки жизненного цикла космических комплексов и входящих в их состав изделий.....	16
1.2. Технологии информационной поддержки жизненного цикла сложных организационно-технических объектов.....	29
1.3. Перспективные технологии программной инженерии информационных систем	45
1.4. Постановка задачи автоматизации информационно-аналитической поддержки жизненного цикла космических средств как сложных объектов и обоснование выбора пути её решения	59
1.5. Выводы по разделу	69
2. Синтез моделей и алгоритмов автоматизированной информационно-аналитической поддержки жизненного цикла космических средств	70
2.1. Онтолого-управляемое моделирование предметной области.....	70
2.2. Онтологическая модель представления знаний о бизнес-процессах информационно-аналитической поддержки жизненного цикла космических средств как сложных объектов.....	84
2.3. Онтологическая модель представления знаний о согласовании вычислительных задач информационно-аналитической поддержки жизненного цикла космических средств как сложных объектов	96

2.4.	Алгоритмы формирования концептуальных моделей и схем программ решения аналитических вычислительных задач и организация вычислений по ним	109
2.5.	Выводы по разделу	140
3.	Квалиметрия полимодельного комплекса автоматизированной информационно-аналитической поддержки жизненного цикла космических средств	141
3.1.	Системный подход к оцениванию показателей качества моделей автоматизированной информационно-аналитической поддержки жизненного цикла космических средств как сложных объектов	141
3.2.	Алгоритмы проверки согласованности моделей бизнес-процессов и моделей согласованных вычислительных задач.....	143
3.3.	Верификация схем программ информационно-аналитической поддержки жизненного цикла космических средств	162
3.4.	Реализуемость единого информационного пространства в рамках функционирования системы информационно-аналитической поддержки жизненного цикла космических средств	169
3.5.	Методика оценивания функциональной эффективности информационно-аналитической поддержки жизненного цикла космических средств.....	176
3.6.	Выводы по разделу	181
4.	Практическое применение результатов теоретических исследований в задаче информационно-аналитической поддержки жизненного цикла ракеты-носителя «Союз-2».....	182
4.1.	Описание задачи и элементов системы автоматизированного формирования электронного паспорта ракеты-носителя «Союз-2» .	182

4.2.	Программный комплекс и порядок автоматизированного онтолого-управляемого проектирования системы информационно-аналитической поддержки жизненного цикла космических средств	194
4.3.	Некоторые примеры реализации элементов системы информационно-аналитической поддержки жизненного цикла ракеты-носителя «Союз-2»	199
4.4.	Экспериментальные расчеты эффективности применения системы информационно-аналитической поддержки жизненного цикла ракеты-носителя «Союз-2»	209
4.5.	Выводы по разделу	214
Заключение		215
Список сокращений и условных обозначений		217
Список литературы		222
Список иллюстраций и таблиц		253
Приложения		260
1.	Технологические аспекты информационной поддержки жизненного цикла сложных объектов	260
2.	Модели, методы и технологии программной инженерии информационных систем	317
3.	Модели и алгоритмы доступа к гетерогенным информационным ресурсам на основе онтологий	378
4.	Некоторые элементы системы информационно-аналитической поддержки жизненного цикла ракеты-носителя «Союз-2»	404
Предметный указатель		407

ВВЕДЕНИЕ

Актуальность темы. В настоящее время можно наблюдать повсеместное внедрение информационных технологий (ИТ) в различных предметных областях (ПрО) как в Российской Федерации (РФ), так и за рубежом. Так, например, в 2011 г. в Давосе президентом Всемирного экономического форума Клаусом Швабом была сформулирована и стала определяющей для многих государств концепция четвертой промышленной революции «Индустрия 4.0» [246]. Суть этой концепции состоит в ускоренной интеграции так называемых социо-киберфизических систем в заводские и эксплуатационные процессы, во всесторонней автоматизации этих процессов на основе ИТ с целью увеличения производительности труда, экономического роста и качества жизни в целом. «Четвертая промышленная революция приведет к перераспределению мест стран в глобальной конкуренции — это представляет собой шанс для России» [171]. В рамках внедрения «Индустрии 4.0» развивается также концепция «цифровой экономики», согласно которой данные в цифровой форме являются ключевым фактором повышения эффективности во всех сферах социально-экономической деятельности [90, 166].

Развитие этих концепций определено тем фактом, что современные высокотехнологичные организации и предприятия относятся к классу сложных организационно-технических объектов (СОТО), представленных организационно-техническими полиструктурами, цели деятельности которых связаны с их изделиями (или проектами) [137, 138, 178, 181]. В диссертационном исследовании в качестве примера прикладной области, представленной такими объектами, рассматривается создание и применение космических средств (КСр) организациями и предприятиями ракетно-космической отрасли. В таких объектах циркулируют сверхбольшие объемы информации различной природы, используемые информационные ресурсы размещены в территориально распределенных узлах и представляют собой совокупность квазиструктурированных

данных, бизнес-процессы слабо формализованы и трудно поддаются оптимизации, а информационное обеспечение технологических процессов не образует единую сквозную и замкнутую технологическую цепочку поддержки и управления жизненным циклом (ЖЦ) изделий (проектов), поскольку решаемые задачи автоматизированы фрагментарно, а их результаты зачастую не согласованы. Совокупность указанных факторов ведет к слабо прогнозируемому уровню качества получаемых организациями результатов деятельности и его потенциальному снижению.

В связи с этим, возникает необходимость развития более совершенных технологий, обеспечивающих комплексный учет информационных аспектов функционирования СОТО на всех этапах их ЖЦ с целью повышения эффективности и обоснованности принимаемых решений по управлению и повышению качества результатов в условиях ограниченных ресурсов. Среди основных современных методологий и технологий, посредством которых предлагается достигать поставленных целей, называются большие данные (Big Data), искусственный интеллект (ИИ), системы распределенного реестра, моделирование и прогнозирование, «сквозные» технологии, обеспечивающие создание единого информационного пространства (ЕИП) и др [166]. Таким образом, в масштабе государств ставятся задачи комплексной автоматизации деятельности предприятий на основе развития соответствующих методов, технологий, их научного обоснования, а также их внедрения в различных ПрО.

Одной из таких задач в соответствии с ГОСТ 1410-002-2010 и Стратегией цифровой трансформации ракетно-космической отрасли до 2025 г. и перспективу до 2030 г. (одобрена на научно-техническом семинаре №КШ-672-пр госкорпорации «Роскосмос») является создание так называемой системы информации о техническом состоянии (ТС) и надежности космических комплексов (КК) и входящих в их состав изделий, вопросы совершенствования которой и рассматриваются в настоящей работе в качестве примера наукоемкой отрасли, связанной с СОТО [58]. Здесь под ТС понимается совокупность из-

меняющихся в процессе проектирования, производства, испытаний, эксплуатации значений параметров (атрибутов) изделия, характеризующих его функциональную пригодность в заданных условиях применения [28, 66, 138].

Современный этап развития ракетно-космической отрасли характеризуется возрастанием сложности КК и процессов управления ими, что приводит к ужесточению требований к оперативности получения, качеству и достоверности результатов обработки и анализа информации различной природы, используемой аналитиками при оценивании ТС и надежности изделий, а также поддержки принятия решений (ППР) при управлении ЖЦ КК и входящих в их состав изделий.

Таким образом, в рамках диссертационной работы рассматривается актуальная задача разработки моделей и алгоритмов автоматизированной информационно-аналитической поддержки ЖЦ СОТО. При этом под СОТО в рассматриваемой ПрО в общем случае понимается КК и его изделия, к числу которых относятся КСр, над которыми проводятся работы представителями организаций разработчиков и эксплуатантов КК. Решение указанной задачи ввиду специфики организационных аспектов в ПрО предполагает создание модельно-алгоритмического комплекса, основанного на описании бизнес-процессов, связанных с оцениванием ТС СОТО и проводимых при работе над изделием на всех этапах его ЖЦ. При этом такое описание может быть задано непосредственно специалистами ПрО как экспертами, обладающими наиболее полным знанием о предметной специфике, что предполагает необходимость обеспечить в автоматизированном режиме инвариатное ему функционирование информационно-аналитической системы (ИАС), ориентированной на формирование электронного паспорта изделия, характеризующего ТС СОТО.

Степень разработанности темы исследования. Сформулированная в диссертации задача была исследована и решена на основе корректного использования и глубинного сочетания фундаментальных концепций, принципов и подходов, используемых в системном анализе, общей теории систем, про-

граммной инженерии, инженерии требований, теориях ИИ, теории концептуального программирования, теории процессов, теориях формальных логик (в особенности, дескрипционной логики), теории принятия решений, теории вычислительных процессов, теории мультиагентных систем, теории мониторинга состояния и управления структурной динамикой сложных объектов. Среди наиболее выдающихся отечественных исследователей указанных направлений можно назвать Соколова Б.В., Охтилева М.Ю., Юсупова Р.М., Гаврилову Т.А., Городецкого В.И., Ильина В.Д., Когаловского М.Р., Нариньяни А.С., Погорелова В.И., Поспелова Д.А., Смирнова А.В., Тыугу Э.Х., Федорова И.Г., Цуканову О.А. Среди зарубежных исследователей можно отметить Буча Г., Ванда Я., Вебера Р., Гелерта А., Кальванезе Д., Кодда Э., Люггера Джорджа Ф., МакГинесс Дебору Л., Норвига П., Пан Джефа З., Рассела С., Розмана М., Рудольфа С., Хоррокса И.

Необходимо отметить, что рассматриваемая задача информационно-аналитической поддержки в силу целого ряда особенностей, рассмотренных в работе, требует привлечения системно-кибернетического подхода к созданию соответствующего модельно-алгоритмического комплекса. В связи с этим, в рамках исследования необходимо на глубинном межмодельном уровне сочетать предлагаемые указанными авторами формальные подходы, позволяющие успешно решать задачи, составляющие основу информационно-аналитической поддержки. Так, например, вопросам комплексного моделирования и многовариантного планирования технологических процессов, а также квалитметрии моделей и полимодельных комплексов уделено внимание в работах Соколова Б.В., вопросам мониторинга ТС СОТО – в работах Охтилева М.Ю., мультиагентным системам в среде распределенных вычислительных устройств – в работах Городецкого В.И., Рассела С., Норвига П., онтологическому моделированию – в работах Цукановой О.А., Кальванезе Д., Пан Джефа З., экспертным системам – в работах Поспелова Д.А., Гавриловой Т.А., Люггера Джорджа Ф.

Цель исследования заключается в повышении функциональной эффективности обеспечения заинтересованных организаций актуальной и достоверной информацией о ТС КСр на основе разработки модельно-алгоритмического комплекса автоматизированной информационно-аналитической поддержки их ЖЦ.

Задачи диссертационного исследования. В рамках исследования для достижения поставленной цели в рассматриваемой ПрО необходимо обеспечить возможность:

- идентификации и совместного учета информационных, поведенческих и функциональных требований к системе информационно-аналитической поддержки (СИАП) ЖЦ КСр на основе комплексного моделирования разноаспектных экспертных знаний о рассматриваемой ПрО;
- формализации и совместного представления взаимосвязанных бизнес-процессов информационно-аналитической поддержки ЖЦ КСр на основе извлечения неполных и противоречивых экспертных знаний о них;
- уточнения в бизнес-процессах функциональных задач интеграции данных и приложений информационно-аналитической поддержки и порядка обработки и анализа этих данных с целью оценивания ТС КСр;
- осуществления инвариантного перехода от описания ПрО к вычислительным моделям (ВМ), приблизив их структурные свойства к исходным требованиям, и, в последующем, к схемам программ с возможностью организации потоковых асинхронных и параллельных вычислений по ним в соответствии с аналогичной спецификой и требованиями к ходу бизнес-процессов информационно-аналитической поддержки;
- автоматизированной верификации формируемого полимодельного комплекса и оценивания функциональной эффективности СИАП на основе разработки методики и алгоритмов расчета её частных показателей.

Объект исследования – процесс информационно-аналитической поддержки ЖЦ КСр.

Предмет исследования – модели и алгоритмы информационно-аналитической поддержки ЖЦ КСр.

Научная новизна заключается в следующем:

- впервые разработана формальная онтологическая система, позволяющая на конструктивной основе специфицировать и совместно учитывать информационные, поведенческие и функциональные требования, предъявляемые к облику СИАП ЖЦ СОТО (в т.ч. КСр) и осуществлять инвариантный переход от описания ПрО к моделям программного обеспечения (ПО) СИАП;
- разработана оригинальная онтологическая модель представления знаний (МПЗ) о бизнес-процессах информационно-аналитической поддержки ЖЦ КСр как СОТО, отличающаяся от существующих моделей ориентацией на извлечение и совместное представление разноаспектных знаний экспертов ПрО об организационных и технологических процессах этапов ЖЦ СОТО в виде поведенческих требований, предъявляемых к комплексу программ СИАП и их интероперабельности;
- впервые разработана онтологическая МПЗ о согласовании вычислительных задач (СВЗ), позволяющая уточнять бизнес-процессы информационно-аналитической поддержки ЖЦ КСр как СОТО в виде функциональных требований к порядку интеграции данных, их обработки и анализа;
- разработан новый алгоритм структурно-параметрического синтеза ВМ в виде G-моделей, обеспечивающий в автоматизированном режиме инвариантный переход от специфицирования требований к конструктивизму задания отношений вычислимости, и, как следствие, получению схем программ, по которым могут быть организованы вычисления;
- разработан новый алгоритмический комплекс верификации моделей информационно-аналитической поддержки ЖЦ КСр как СОТО, позволяющий в рамках созданной методики обоснованно оценивать её функциональную эффективность на основе сформированной структуры показателей.

Теоретическая и практическая значимость. Теоретическая значимость диссертационного исследования состоит в разработке совокупности взаимосвязанных МПЗ с ориентацией на моделирование последовательности уточняющих спецификаций задач автоматизированной информационно-аналитической поддержки ЖЦ СОТО (в т.ч. КСр), а также алгоритмов синтеза и верификации указанных моделей, обеспечивающих на формальной основе возможность осуществить инвариантный переход от описания требований в ПрО к моделям ПО СИАП. Практическая значимость выполненной работы состоит в повышении функциональной эффективности информационно-аналитической деятельности представителей организаций разработчиков и эксплуатантов КК по оперативному обеспечению заинтересованных лиц актуальной и достоверной информацией о ТС КСр на всех этапах их ЖЦ на основе комплексного моделирования экспертных знаний, их верификации и последующей автоматизации их применения при оценивании ТС КСр, что положительно сказывается на оперативности и обоснованности вырабатываемых лицами, принимающими решения (ЛПР), управленческих воздействий на ЖЦ КСр по обеспечению повышения качества и надежности КК и их изделий.

Методология и методы исследования. Методология исследования основана на использовании междисциплинарного подхода, базирующегося на фундаментальных положениях таких научных отраслей как системный анализ, программная инженерия, инженерия требований, математическое моделирование, теория искусственного интеллекта. При решении поставленных задач в числе прочих применялись методы комплексного имитационно-аналитического, онтологического, процессного и объектного моделирования, концептуального программирования и распознавания образов.

Положения, выносимые на защиту.

1. Онтологическая модель представления знаний о бизнес-процессах информационно-аналитической поддержки создания и применения космических средств.

2. Онтологическая модель представления знаний о согласовании вычислительных задач информационно-аналитической поддержки создания и применения космических средств.
3. Алгоритм структурно-параметрического синтеза вычислительных моделей в виде G-моделей по онтологическим спецификациям требований к системе информационно-аналитической поддержки создания и применения космических средств.
4. Алгоритм проверки согласованности (выполнимости и онтологической выразительности) спецификаций требований к системе информационно-аналитической поддержки создания и применения космических средств.

Соответствие паспорту специальности. Исследование выполнено по специальности 05.13.01 – Системный анализ, управление и обработка информации (технические системы) и соответствует следующим пунктам её паспорта: «7. Методы и алгоритмы структурно-параметрического синтеза и идентификации сложных систем», «8. Теоретико-множественный и теоретико-информационный анализ сложных систем», «10. Методы и алгоритмы интеллектуальной поддержки при принятии управленческих решений в технических системах», «13. Методы получения, анализа и обработки экспертной информации».

Обоснованность и достоверность научных положений подтверждаются всесторонним анализом современного состояния исследований задачи информационно-аналитической поддержки ЖЦ СОТО, корректностью предложенных моделей и алгоритмов, согласованностью результатов экспериментов, проведенных с использованием программной реализации, апробацией основных теоретических положений диссертации в печатных трудах, докладах на научно-технических конференциях и семинарах, положительными результатами внедрения основных научных положений диссертации в рассматриваемой ПрО.

Личный вклад автора. Содержание диссертационной работы и научная новизна отражают персональный вклад автора. Научное исследование было

проведено при общем руководстве д.т.н., профессора Соколова Б.В. Представленные научные результаты получены автором лично. Программная реализация модельно-алгоритмического комплекса получена при участии автора.

Реализация и внедрение результатов исследований. Теоретические результаты диссертационной работы были использованы в СПИИРАН в рамках следующих проектов и НИР: грант РФФИ № 16-07-00779-а «Разработка методологии и модельно-алгоритмического обеспечения проактивного управления структурной динамикой социо-кибер-физических систем», грант РФФИ № 17-29-07073-офи_м «Теоретические и технологические основы формирования и децентрализованного планирования поведения коалиций интеллектуальных роботов на основе механизмов социо-инспирированной самоорганизации и умных контрактов», грант РФФИ № 18-08-01505-а «Разработка и исследование методов и алгоритмов проактивного управления восстановлением работоспособности бортовых систем сложных динамических объектов при возникновении нештатных ситуаций», грант РФФИ № 0073-2018-0003 «Методология и технологии интеграции существующих и перспективных государственных и коммерческих информационно-управляющих и телекоммуникационных систем и сетей на различных этапах их жизненного цикла» – при разработке онтологической системы информационно-аналитической поддержки ЖЦ СОТО. Теоретические исследования и практические результаты также были использованы в рамках выполняемой АО «РКЦ «Прогресс» опытно-конструкторской работы по созданию комплекса ракеты-носителя (РН) «Союз-2» (шифр «Русь») – при решении задач совершенствования системы информации о ТС и надежности КК и входящих в их состав изделий на основе разработки полимодельного комплекса информационно-аналитической поддержки жизненного цикла РН «Союз-2», в рамках выполняемой АО «НИО ЦИТ Петрокомета» (Госкорпорация «Ростех») составной части опытно-конструкторской работы на тему «Разработка единого виртуального электронного паспорта космической ракеты-носителя «Союз-2» – при разработке алгоритмического комплекса синтеза и верификации моделей и разработке СИАП ЖЦ КСр, а также

в рамках учебно-педагогической деятельности при ФГАОУ ВО «СПбГУАП» и сопряженных с ней дисциплин «Методология программной инженерии (спецификация требований)», «Поддержка жизненного цикла программного обеспечения (спецификация требований)» – при разработке методики оценивания функциональной эффективности информационно-аналитической поддержки ЖЦ СОТО.

Апробация результатов работы. Материалы диссертационной работы были представлены и обсуждены на следующих международных, всероссийских и ведомственных конференциях и семинарах: Научная сессия ГУАП, посвященная дню Космонавтики (2016, 2017, 2018 гг.); XIX и XX международные конференции «Проблемы управления и моделирования в сложных системах» (2017, 2018 гг.); 10-я Всероссийская Мультиконференция по проблемам управления (МКПУ-2017); Всероссийская научно-техническая конференция «Актуальные проблемы ракетно-космической техники» (V Козловские чтения) (2017 г.); Восьмая всероссийская научно-практическая конференция по имитационному моделированию и его применению в науке и промышленности «Имитационное моделирование. Теория и практика» (ИММОД-2017); Международная научная конференция по проблемам управления в технических системах (CTS'2017); III Всероссийская научно-техническая конференция «Теоретические и прикладные проблемы развития и совершенствования автоматизированных систем управления военного назначения» (2017 г.); Computer Science On-line Conference 2018 (CSOC 2018); XVI Санкт-Петербургская международная конференция «Региональная информатика (РИ-2018)»; 3-я Международная научная конференция «Интеллектуальные информационные технологии в технике и на производстве» (2018 г.).

Публикации по теме диссертации. Результаты диссертационной работы опубликованы в 29 печатных работах, среди них 5 работ в журналах, индексируемых в Scopus и Web of Science, 4 работы в рецензируемых журналах из перечня ВАК. Также было зарегистрировано 4 программы для ЭВМ.

Структура и объем работы. Диссертационная работа объемом в 408 страниц состоит из введения, 4 разделов, списка литературы из 274 наименований, списка сокращений и условных обозначений, списка из 85 иллюстраций и 32 таблиц, 4 приложений и предметного указателя. В разделе 1 диссертационного исследования проводится системный анализ состояния исследований задачи автоматизации информационно-аналитической поддержки ЖЦ СОТО, рассматриваются перспективные технологии информационной поддержки ЖЦ изделий, программной инженерии информационных систем (ИС). В разделе 2 рассматривается онтолого-управляемое моделирование бизнес-процессов и СВЗ, модели доступа к разнородным источникам данных, синтез ВМ и схем программ, особенности организации вычислений по ним в рамках мультиагентной архитектуры. В разделе 3 представлена методика оценивания функциональной эффективности СИАП ЖЦ КСр и алгоритмы верификации полученных моделей. В разделе 4 рассматриваются аспекты практического применения результатов теоретических исследований на примере СИАП ЖЦ РН «Союз-2». Проводится анализ эффективности её применения при решении задач информационно-аналитической поддержки ЖЦ РН «Союз-2» на основе предложенной в диссертационной работе структуры показателей.

1. СИСТЕМНЫЙ АНАЛИЗ СОСТОЯНИЯ ИССЛЕДОВАНИЙ ЗАДАЧИ ИНФОРМАЦИОННО-АНАЛИТИЧЕСКОЙ ПОДДЕРЖКИ ЖИЗНЕННОГО ЦИКЛА КОСМИЧЕСКИХ СРЕДСТВ КАК СЛОЖНЫХ ОБЪЕКТОВ

1.1. Характеристика задачи автоматизации информационно-аналитической поддержки жизненного цикла космических комплексов и входящих в их состав изделий

1.1.1. Описание предметной области и решаемой задачи

В рассматриваемой ПрО существует КК, связанный с множеством организаций, ответственных за проектирование, производство, эксплуатацию и утилизацию изделий КК (в т.ч. КСр). В рамках каждой из организаций проводятся работы над изделием, влекущие за собой изменение его состояния. Иначе говоря, процессы выполнения работ тесно связаны с процессом изменения состояния изделия. При этом каждый из этапов ЖЦ изделия связан с одной из организационных единиц КК (организацией, предприятием, отделом, департаментом и т.д.) в том смысле, что выполняемые представителями этой организации работы влекут за собой изменение состояния изделия так, что в начале выполнения этих работ изделие находится в некотором начальном состоянии одного этапа ЖЦ, а в результате выполнения работ последовательно переходит в другие промежуточные или целевые состояния других этапов ЖЦ. Процессы выполнения работ могут проходить асинхронно и параллельно. С информационной точки зрения, промежуточные результаты хода выполнения работ фиксируются в виде технологической, технической и организационной информации.

В рамках рассматриваемой ПрО в организациях функционируют отделы контроля качества, выполняющие работы по оцениванию ТС изделия на всех этапах его ЖЦ и осуществляющие ППР по управлению ЖЦ изделий для ЛПР, направленную на обеспечение повышения качества и надежности производи-

мых и эксплуатируемых изделий. Кроме того, таким отделам необходимо проводить оценивание ТС, агрегацию, обработку и анализ данных о состоянии изделия в результате выполнения работ на каждом из этапов ЖЦ с целью обеспечения актуальной и достоверной информацией ЛПР, связанных со всеми этапами ЖЦ.

Работа этих отделов по оцениванию состояния изделия связана со сбором больших объемов неполной, недоопределенной и разнородной информации из различных источников, представленной как в бумажной, так и в электронной форме. При выполнении работ необходимо осуществлять приведение собранной информации к единому виду, её обработку и содержательный анализ с целью обобщения результатов работы организаций с изделием в виде, так называемого, дела изделия. Такую деятельность будем называть *информационно-аналитической поддержкой ЖЦ КК* и его изделий.

Поскольку информационно-аналитическая поддержка ЖЦ на сегодняшний день в КК осуществляется в большей мере вручную, она сопряжена с ошибками в процессе оценивания, невыполнением работ в надлежащий срок, неполным соответствием выполняемых работ технологическим и бизнес-процессам, неполным сбором информации, неполным приведением её к единому виду, что снижает возможности по оптимизации работ с изделиями и ведет к снижению их потенциального качества и надежности.

В связи с вышесказанным приобретает актуальность задача автоматизации информационно-аналитической поддержки ЖЦ организаций и их изделий как в КК, так и в подобных отраслях деятельности различных организаций и предприятий. При этом автоматизация информационно-аналитической поддержки ЖЦ также сопряжена с рядом трудностей. К ним можно отнести:

- территориальную распределенность организаций, что ведет к потребности в распределенных вычислениях;
- учет как организационных, так и технических аспектов функционирования организаций, и, следовательно, необходимость в разнообразии используемых технологий обработки данных;

- наличие уже функционирующих аппаратно-программных комплексов (АПК) организаций и предприятий, результаты вычислений которых необходимо обобщить в рамках ИАС для формирования электронного дела изделия;
- разнородность и слабую (в некоторых случаях сложную) структурированность исходной информации, размещенной в рамках хранилищ данных (ХД), созданных по различным технологиям;
- сложность обобщения информации о ТС изделия на всех этапах ЖЦ в силу разнообразия происходящих в организациях процессов и отсутствие обобщенных знаний об этих процессах.
- необходимость ведения и хранения хронологии изменения состояния изделия на всех этапах его ЖЦ с целью выявления особенностей работы с изделием и ошибок в процессах, повлекших снижение качества и надежности изделия, для последующей оптимизации указанных процессов.

При этом входными данными для соответствующей системы являются:

- организационная информация, представленная в форме документов об изделии и выполнении работ над ним со всех этапов его ЖЦ;
- технологическая информация об изделии, представленная в форме электронных данных о структуре изделия (3D-модель изделия или, так называемая, электронная структура изделия), документов о технологических аспектах производства, эксплуатации и утилизации изделия;
- техническая информация, формируемая в результате функционирования АПК организаций и предприятий КК.

В качестве выходных данных выступает электронное дело изделия, являющееся результатом оценивания ТС изделия на всех этапах его ЖЦ.

В рамках исследования считается, что:

- этапы ЖЦ связаны с множеством параллельных и асинхронных организационных и технологических процессов;
- за каждым процессом закреплена отдельная организационная единица;

- состояние изделия и состояние организационных и технологических процессов работ с изделием тесно взаимосвязаны, т.е. состояние процесса характеризует текущее состояние изделия, и, наоборот, состояние изделия косвенно характеризует этап процесса;
- входные данные, описанные выше, достоверны, поскольку они являются результатом функционирования организационных и технических систем предприятий КК, в рамках которых доказывается их достоверность;
- в случае, если входные данные являются результатом функционирования АПК, эти данные доступны для чтения в любой момент времени и представляют собой:
 - электронные файлы документов (например, доступные посредством Web-DAV (Web Distributed Authoring and Versioning)- и FTP (File Transfer Protocol)-серверов). При этом содержательная информация этих документов может дополнительно вноситься пользователем системы;
 - электронные файлы с содержимым, описанным посредством языков разметки таких, как, например, eXtensible Markup Language (XML), JavaScript Object Notation (JSON) и др. Иначе говоря, содержимое таких файлов должно быть доступно для синтаксического разбора на электронно-вычислительной машине (ЭВМ) для того, чтобы существовала возможность обработки структурированных данных, которые могут быть, в частности, представлены в такой форме. В рассматриваемой Про таким способом хранится, например, информация об электронной структуре изделия, формируемой на этапе его проектирования;
 - данные из БД, выполненных по различным технологиям. БД могут быть отнесены к реляционному или сетевому типу;
- организации КК территориально распределены и находятся в одной вычислительной сети;
- необходимо учитывать конфиденциальность данных организаций в том смысле, что информация о состоянии изделия на одном этапе ЖЦ доступна

другим организациям в объеме, необходимом для решения стоящих перед ними задач.

1.1.2. Система информации о техническом состоянии и надежности космических комплексов и их изделий

Система информации предназначена для своевременного обеспечения федеральных органов исполнительной власти, заинтересованных предприятий и организаций, участвующих в создании, производстве и эксплуатации КК, достоверными сведениями, необходимыми для выполнения работ по обеспечению и повышению уровня ТС, качества и *надежности* (см. определение 1.1) комплексов и входящих в их состав изделий, совершенствованию систем менеджмента качества предприятий и организаций [58]. Работы по системе информации проводят на всех этапах ЖЦ изделий (при эскизном проектировании, разработке рабочей документации, наземной отработке, летных испытаниях, опытном и серийном (единичном) производстве и эксплуатации КК и входящих в их состав изделий), причем под изделиями, в частности, можно понимать КСр (см. определение 1.2) [51].

Определение 1.1. Под *надежностью* понимается собирательный термин, применяемый для описания характеристики эксплуатационной готовности и влияющих на нее характеристик безотказности, ремонтпригодности, технического обслуживания и ремонта [239, серия ISO 9000].

Определение 1.2. *Космическое средство* – техническое средство, включающее в себя орбитальные средства и средства выведения, предназначенное для функционирования в космическом пространстве и решения или обеспечения решения задач освоения и использования космического пространства [52].

Можно отметить, что создание системы информации направлено, в первую очередь, на управление ЖЦ производимых и эксплуатируемых изделий. В связи с этим, в литературе появилось специальное понятие для такой задачи – «электронный паспорт» (см. определение 1.3).

Определение 1.3. Под *электронным паспортом* понимают комплексное описание объекта или юридического лица (субъекта), включающее сведения о текущих показателях (индикаторах) его состояния или деятельности [28].

Следует уточнить, что информация о состоянии изделия формируется на всех этапах его ЖЦ и необходима в виде совокупного своего представления для ЛПР. Иначе говоря, можно конкретизировать решаемую задачу, как задачу создания электронного паспорта изделия.

Существующая система информации о ТС и надежности КК характеризуется следующими недостатками:

- большие объемы разнородной, зачастую неполной и недоопределенной информации о ТС и надежности КК не интегрированы и не систематизированы, соответственно, не созданы единые БД обо всем ЖЦ КК и не сформировано ЕИП КК;
- наличие многочисленных несовместимых между собой и выполненных на различных платформах информационных систем (ИС) предприятий и организаций, участвующих в задачах системы информации;
- слабый уровень автоматизации аналитической обработки информации о ТС и надежности КК;
- ограничение оперативного доступа к информации о ТС и надежности КК, что приводит к снижению возможностей всех участников системы, а особенно заказчиков КК, объективно оценивать ТС и достигнутый уровень надежности КК в режиме реального времени (РВ), а также принимать решения по управлению ЖЦ КК и входящих в их состав изделий.

Результаты проведенного анализа показали, что решение специальных задач системы информации на всех этапах ЖЦ КК и входящих в их состав изделий невозможно без рассмотрения их как СОТО (см. приложение 1.1), использования эффективных инструментов автоматизации процессов информационно-аналитической поддержки при управлении ЖЦ КК и соответствующих изделий на основе формирования ЕИП.

1.1.3. Единое информационное пространство

Неотъемлемым элементом системы информации в соответствии с принципами «Индустрии 4.0» является ЕИП (см. определение 1.4).

Определение 1.4. ЕИП представляет собой совокупность баз и банков данных, технологий их ведения и использования, информационно-телекоммуникационных систем и сетей, функционирующих на основе единых принципов и по общим правилам, обеспечивающим информационное взаимодействие организаций и граждан, а также удовлетворение их информационных потребностей [90, 109].

Иными словами, ЕИП складывается из следующих существенных компонент [90]:

- информационные ресурсы, содержащие данные, сведения и знания, зафиксированные на соответствующих носителях информации (существенную роль в формировании ЕИП в рассматриваемой задаче играют *метаданные* и технологии их обработки (см. приложение 1.2));
- организационные структуры, обеспечивающие функционирование и развитие ЕИП, в частности, сбор, обработку, хранение, распространение, поиск и передачу информации;
- средства информационного взаимодействия граждан и организаций, обеспечивающие им доступ к информационным ресурсам на основе соответствующих ИТ, включающие программно-технические средства и организационно-нормативные документы.

Удачный пример иллюстрации повышения сложности информационного взаимодействия с ростом объемов информации и сложности организационных структур современных предприятий представлен в работе [93] (см. рис. 1.1).



Рис. 1.1. Уровни организации информационного взаимодействия

Таким образом, эффективность современных сложно структурированных предприятий существенным образом зависит от формирования ЕИП, в рамках которого их представители должны иметь возможность оперативного информационного обмена, составляющего основу успешного совместного функционирования организаций. Однако вопрос теоретических основ формирования такого ЕИП остается открытым.

1.1.4. Информационные и информационно-аналитические системы

В настоящее время отсутствует устоявшееся определение понятия ИС. В широком понимании под ИС подпадает любая система обработки информации. В различных источниках литературы можно найти следующие трактовки.

Определение 1.5. Информационная система – это система, которая организует хранение и манипулирование информацией о ПрО [49].

Определение 1.6. Информационная система – это совокупность содержащейся в БД информации и обеспечивающих ее обработку ИТ и технических средств [50].

Определение 1.7. Информационная система – это компьютерная система, включающая вычислительное и коммуникационное оборудование, ПО, данные и метаданные, лингвистические средства, а также системный персонал, и обеспечивающая поддержку информационной модели некоторой части

реального мира для удовлетворения информационных потребностей пользователей [84].

Поскольку в рассматриваемой задаче речь идет о КК как о совокупности организаций имеет смысл учесть определение корпоративной ИС (или ИС предприятий).

Определение 1.8. Корпоративная информационная система – это информационная система, автоматизирующая значительную часть бизнес-процессов, составляющих деятельность предприятия (организации) [84].

Исходя из того, что в рамках *системы информации* необходимо осуществлять также ППР при управлении ЖЦ СОТО, существует необходимость проведения анализа хранимой в рамках ИС информации. Проблема анализа исходной информации для принятия решений оказалась настолько серьезной, что появилось отдельное направление или вид ИС – *ИАС*, под которой в [20, 65] авторы понимают комплекс аппаратных, программных средств, информационных ресурсов, методик, которые используются для обеспечения автоматизации аналитических работ в целях обоснования принятия управленческих решений.

Проблема аналитической подготовки принятия решений имеет следующие аспекты:

- извлечение из многих источников разнородных данных, представленных в различных форматах и приведение их к единому формату и единой структуре;
- организация хранения и предоставления пользователям необходимой для принятия решений информации;
- анализ информации, в том числе оперативный (в режиме РВ (методы «*On-Line Analytical Processing*» (OLAP)) и интеллектуальный (с применением методов теории ИИ), и подготовка плановой или регулярной оценки состояния управляемого объекта в виде бумажных документов или экранных форм;

– подготовка результатов оперативного и интеллектуального анализа для эффективного их восприятия потребителями и принятия на их основе адекватных решений.

ИАС необходимо относить к классу так называемых открытых систем, определение которых принято Комитетом IEEE в стандарте POSIX 1003.0 [177]. К свойствам открытых систем относят: расширяемость и масштабируемость путем настройки параметров, а не путем перепроектирования и программирования заново, переносимость данных и приложений, *интероперабельность*, т.е. взаимодействие с другими системами при обращении к информационным или вычислительным ресурсам, способность к интеграции.

Таким образом, задача совершенствования системы информации о ТС и надежности СОТО – есть, в первую очередь, задача создания соответствующей автоматизированной ИАС, охватывающей все этапы ЖЦ СОТО. Будем такую систему называть системой *информационно-аналитической поддержки* ЖЦ СОТО. При этом под информационно-аналитической поддержкой будем понимать решение круга задач аналитической подготовки информации для принятия решений, указанных выше.

Замечание 1.1. Поскольку в рамках работы рассматривается задача создания электронного паспорта изделия, необходимым элементом ИАС является банк данных, считающийся разновидностью ИС, в которой реализованы функции централизованного хранения и накопления обрабатываемой информации, организованной в одну или несколько БД (см. определение 1.9).

Определение 1.9. База данных представляет собой совокупность специальным образом организованных данных, хранимых в памяти вычислительной системы и отображающих состояние объектов и их взаимосвязей в рассматриваемой ПрО [204].

1.1.5. Обоснование необходимости автоматизации информационно-аналитической поддержки жизненного цикла сложных организационно-технических объектов

Необходимость автоматизации *информационно-аналитической поддержки* ЖЦ изделий для решения задач ППР и управления ЖЦ изделий продиктована неудовлетворительным состоянием внедрения передовых ИТ в областях государственного и промышленного управления по целому ряду причин [1, 123, 152].

На современном уровне развития промышленной и государственной кооперации отсутствие единого комплекса стандартов «электронного описания» различных этапов ЖЦ, обеспечивающих информационное взаимодействие электронных технологий (в рамках одной организации или «виртуального» объединения организаций), приводит к значительным издержкам в процессах проектирования, подготовки производства, изготовления и эксплуатации продукции. Эти издержки западными аналитиками оцениваются, например, в масштабах промышленности США, в десятки миллиардов долларов в год [163].

Наличие подобных издержек связано с тем, что поведение систем организационно-технического управления зачастую не согласовано с точки зрения ЛПР на уровне автоматизированного информационного обмена, результаты обработки информации на различных этапах ЖЦ несопоставимы, а также отсутствуют общие принципы и правила, затрудняющие возможность тиражировать обобщенный опыт экспертов ПрО, что, в частности, приводит к необходимости подготовки и содержания узкоспециализированных специалистов, ориентированных на решение задач в рамках частных ИС. Таким образом, можно говорить об отсутствии согласованного функционирования различных автоматизированных систем (АС), существующих в рамках организаций и предприятий.

В приложении 1.3 представлен анализ существующих технологий, применяемых при интеграции и анализе информационных ресурсов предприятий,

который показал, что на сегодняшний день лишь обозначена проблема необходимости автоматизации информационно-аналитической поддержки ЖЦ СОТО, а существующие технологии и решения имеют частный характер и не позволяют в существующем виде обеспечить автоматизацию интеграции и анализа информации о процессах в СОТО на всех этапах его ЖЦ. Тогда по совокупности изложенных фактов может быть сформирована следующая структура задачи исследования, проиллюстрированная на рис. 1.2.

Таким образом, подводя итоги анализа характеристики задачи исследования по автоматизации информационно-аналитической поддержки ЖЦ СОТО, можно отметить, что на сегодняшний день в явном виде представлена сложная научно-техническая задача, характеризующаяся, с одной стороны, высокими требованиями, предъявляемыми к облику ИАС в части решения аналитических задач по оцениванию ТС СОТО, а с другой – необходимостью интеграции информации о состоянии подсистем и элементов СОТО при создании ЕИП организаций в рамках соответствующих АС.

Решение данной задачи предполагает разработку моделей и алгоритмов создания и использования информационно-вычислительных систем и соответствующих ИТ, обеспечивающих сбор, интеграцию, обработку и анализ сверхбольших потоков разнородной информации, поступающих в реальном масштабе времени от различных элементов и подсистем, входящих в состав организаций и предприятий.

При этом опосредованно в результате внедрения СИАП ЖЦ СОТО должны существенно повышаться производительность труда, экономические показатели, качество изделий, а также конкурентоспособность и эффективность функционирования организаций и предприятий в различных ПрО таких, как проектирование, производство, испытания и эксплуатация ракетно-космической техники (РКТ) и других изделий, организационно-техническое управление объектами государственного и промышленного управления.

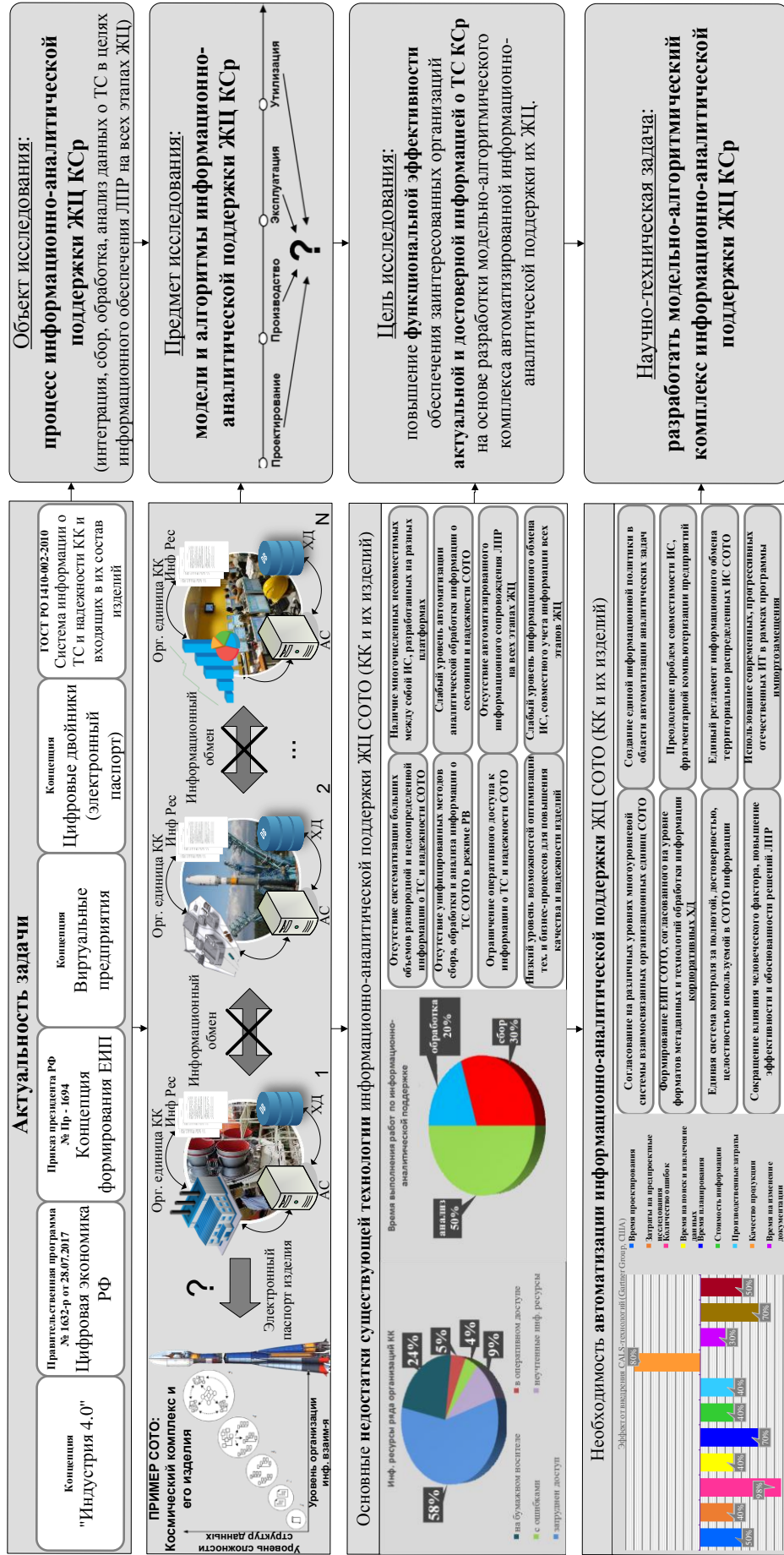


Рис. 1.2. Структура задачи диссертационного исследования

1.2. Технологии информационной поддержки жизненного цикла сложных организационно-технических объектов

1.2.1. Информационная поддержка жизненного цикла сложных организационно-технических объектов

1.2.1.1. Проблема фрагментарной компьютеризации предприятий

Для многих современных предприятий характерна частичная, фрагментарная компьютеризация отдельных видов производственной деятельности [188]. Это связано с внедрением качественно новых средств автоматизации в устоявшиеся технологические процессы. Зачастую эффект от такого подхода невелик. Примерами таких попыток могут служить:

- многочисленные АСУ, роль которых сводилась к автоматизации учетных и отчетных функций;
- конструкторские системы автоматизированного проектирования (САПР) (CAD – Computer Aided Design), заменявшие ручной чертежный труд;
- технологические САПР (САМ - Computer Aided Manufacturing), облегчавшие подготовку технологической документации и процесс управления станками с числовым программным управлением;
- АС инженерных расчетов (САЕ -Computer Aided Engineering)

и т.д.

Такие системы создавались на различных вычислительных платформах, с помощью различных языков программирования и, как правило, были несовместимы между собой, что предопределяло их автономное использование. Кроме того, требовалась многократная перекодировка информации для ввода ее в ту или иную систему, что влекло за собой возрастание объемов рутинного труда, приводило к многочисленным ошибкам и, как следствие, к снижению эффективности систем.

Эта ситуация привела к осознанию необходимости интеграции информационно-вычислительных систем в единый комплекс. Стала развиваться идея создания ЕИП предприятий или интегрированной информационной

среды (ИИС), охватывающей все этапы ЖЦ изделий, которыми занимаются предприятия (см. определения ниже).

Определение 1.10. ИИС – это совокупность распределенных БД, содержащих сведения об изделиях, производственной среде, ресурсах и процессах предприятия, обеспечивающая корректность, актуальность, сохранность и доступность данных [188].

Определение 1.11. ЖЦ изделия военной техники – это совокупность взаимосвязанных процессов последовательного изменения состояния изделий военной техники от формирования исходных требований к ним до снятия их с эксплуатации и списания [57].

ЖЦ изделия – совокупность этапов, через которые проходит изделие за время своего существования: маркетинговые исследования, составление технического задания (ТЗ), проектирование, технологическая подготовка производства, изготовление, поставка, эксплуатация, утилизация [68, 188].

Именно идея ИИС и информационной интеграции стадий ЖЦ стала базовой при выработке подхода, получившего в США название информационной поддержки изделий (ИПИ).

1.2.1.2. Концепция информационной поддержки изделий

В настоящее время в космической отрасли предпринимаются попытки внедрить технологии электронного сопровождения наукоемкой продукции на всех этапах ЖЦ (ИПИ) или CALS (Continuous Acquisition and Life cycle Support)-технологии). ИПИ-технологии – это подход к разработке, производству, эксплуатации и обслуживанию изделий путем информационной поддержки их ЖЦ на основе электронного обмена данными [16, 127, 161]. Концепция ИПИ определяет правила, регламенты, стандарты, по которым строится информационное взаимодействие участников проектирования, производства, испытаний и других этапов ЖЦ. Стоит отметить 2 определения CALS-технологий из научной и промышленной отраслей (см. определения 1.12, 1.13).

Определение 1.12. CALS (Непрерывная информационная поддержка жизненного цикла продукции) – это концепция и идеология информационной поддержки ЖЦ продукции на всех его стадиях, основанная на использовании ИИС, обеспечивающая единообразные способы информационного взаимодействия всех участников этого цикла, реализуемого в форме международных стандартов, регламентирующих правила указанного взаимодействия (преимущественно посредством электронного обмена данными) [188].

Определение 1.13. Технологии комплексной компьютеризации сфер промышленного производства, цель которых – унификация и стандартизация спецификаций промышленной продукции на всех этапах ее ЖЦ, называют *CALS-технологиями* [161].

Основные спецификации, рассматриваемые в CALS, представлены проектной, технологической, производственной, маркетинговой, эксплуатационной документацией. В CALS-системах должны быть предусмотрены хранение, обработка и передача информации о соответствующих спецификациях в компьютерных средах, оперативный доступ к данным ЛПП.

При применении ИПИ значительно повышается эффективность деятельности предприятий за счет ускорения процессов исследования, разработки, сокращения издержек при производстве и эксплуатации изделий, существенного повышения качества выпускаемой продукции.

Результаты анализа, проведенные в приложении 1.4, показали, что повышение эффективности предприятий при применении CALS-технологий достигается посредством формирования ИИС АС предприятий, консолидации результатов их функционирования в рамках специализированной OLAP-системы, реализации принципов ПИ, оперативного информационного обмена, что в совокупности позволяет обеспечивать ЛПП актуальной и достоверной информацией о ТС и надежности изделия и ходе работ, проводимыми над ним.

1.2.1.3. Задачи информационной поддержки изделий

В соответствии с [188] основные задачи, решаемые в рамках CALS-технологий посредством введения ИИС, представлены на рис. 1.3.

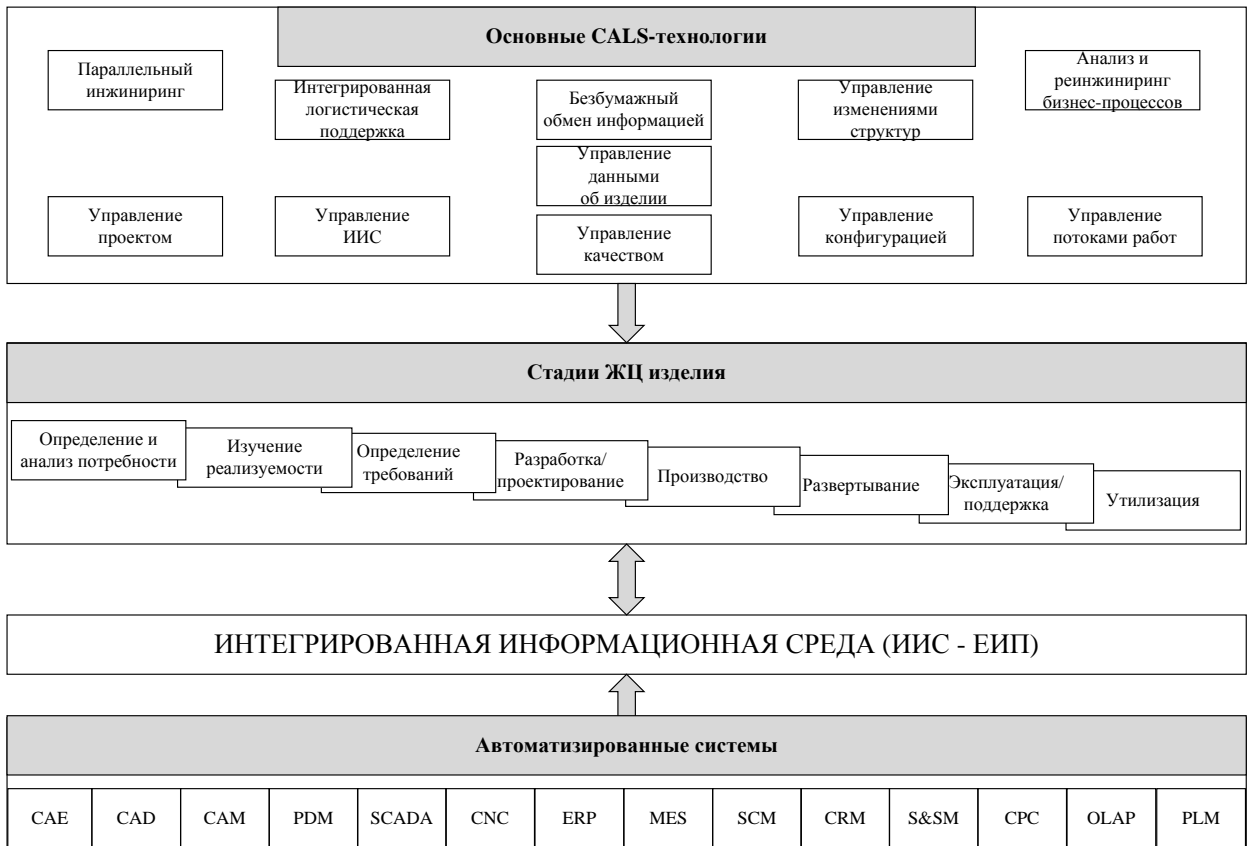


Рис. 1.3. Основные CALS-технологии, реализуемые при формировании интегрированной информационной среды

Анализ этих задач на основе литературы [68, 127, 161, 188, 195, 239, 241] в контексте их решения в рамках СИАП ЖЦ СОТО представлен в приложении 1.5. В то же время решение этих задач на сегодняшний день предполагает опору на ряд существующих стандартов ИПИ, среди которых наибольшую популярность приобрел стандарт STEP. Анализ этих стандартов приведен в приложении 1.7. Результаты анализа показали неудовлетворительное состояние внедрения этих стандартов по отмеченным в приложении причинам, к числу которых, прежде всего, необходимо отнести потребность в изменении существующей на предприятиях инфраструктуры АС, что требует существенных ресурсов и длительного срока внедрения, а также отсутствие ориентации на решение задач оценивания ТС и надежности СОТО в соответствии с рассматриваемой задачей.

1.2.2. Задача мониторинга состояния в информационно-аналитической поддержке жизненного цикла сложных объектов

Как было отмечено в разделе 1.1.4, функционирование СИАП ЖЦ СОТО связано, в том числе, с необходимостью оценивания состояния СОТО. Также в разделе 1.2.1 в рамках CALS-технологий были отмечены задачи информационной поддержки в режиме РВ управления качеством, обеспечения надежности, управления конфигурацией, т.е. обработки и анализа информации, характеризующей качественные и надежность характеристики изделий, и формирования соответствующей модели состояния, опирающейся на конфигурацию изделия. Также можно отметить задачу информационной поддержки ПИ, предполагающего параллельную и асинхронную обработку информации. Эти аспекты определяют необходимость оценивания состояния СОТО на всех этапах его ЖЦ. В приложении 1.8.1 рассмотрены основные понятия, связанные с оцениванием состояния СОТО, такие как *ТС, мониторинг, система мониторинга* и др., а также непосредственно задача мониторинга ТС СОТО.

В результате рассмотрения общей задачи мониторинга состояния было выявлено, что, поскольку СИАП ЖЦ СОТО одной из своих целей имеет решение задачи наблюдения состояния СОТО, она должна включать в себя задачи систем мониторинга состояния, что позволяет говорить о необходимости применения соответствующих моделей, методов и алгоритмов. При этом СОТО должен рассматриваться как динамическая система, результатом функционирования которой является формирование электронного паспорта СОТО, отражающего те или иные аспекты его функционирования, его качественные характеристики, степень работоспособности и степень соответствия заявленным требованиям качества и надежности.

Анализ современных подходов к решению задачи мониторинга показал, что существующие методы предполагают создание систем параметрической идентификации, т.е. систем, которые оперируют не состояниями объекта, а лишь его наблюдаемыми параметрами (см. приложение 1.8.2). Это не позво-

ляет полно и адекватно оценивать состояние СОТО, поскольку при использовании рассмотренных методов не наблюдаются скрытые качественные свойства СОТО, проявление которых существенно влияет на анализ совокупности его характеристик и выводы о его ТС на этапе функционирования.

С положительной стороны на практике себя показала методология автоматизированного анализа состояния объектов на основе G-моделей (см. приложения 1.8.3, 1.8.4). Заложенные в её основу концепция инвариантности состояния реального объекта и состояния вычислительного процесса, модель представления, извлечения и формализации знаний, модель схем программ потоковых вычислений, модель топологии параметров состояния, методы верификации схем программ и принцип «программирования без программирования» позволяют на формальной модельно-алгоритмической основе быстро, качественно и корректно проектировать полимодельные ПК анализа состояния сложных объектов, обеспечивая автоматизацию решения аналитических задач, связанных с необходимостью выявления качественных характеристик СОТО во время их функционирования. Предпосылки, исходя из которых разрабатывался указанный методологический аппарат, актуальны и для рассматриваемой темы. Тем не менее, ряд особенностей требует расширения возможностей теоретического аппарата для решения поставленной задачи в части моделирования процессов решения аналитических задач, обработки и хранения сложно структурированных данных, многоуровневой декомпозиции процесса проектирования СИАП ЖЦ СОТО.

1.2.3. Системы оперативных аналитических вычислений в режиме реального времени

1.2.3.1. Необходимость систем аналитических вычислений в реальном времени в задаче оценивания технического состояния и надежности сложных объектов

Рассмотренные в разделах 1.2.1, 1.2.2 аспекты, связанные с решением аналитических задач по оцениванию ТС и надежности СОТО определяют ряд требований к СИАП ЖЦ СОТО, к которым можно отнести следующие:

- необходимость сбора информации из различных источников данных о состоянии СОТО на всех этапах его ЖЦ;
- необходимость ведения хронологии изменения состояния СОТО – информация, однажды, попавшая в систему, должна в ней оставаться и быть доступной для последующего использования;
- необходимость формирования ИИС (ЕИП) для доступа к источникам данных и их совместной обработки;
- выполнение задачи мониторинга состояния в режиме РВ, которая предполагает сбор, обработку и анализ информации о состоянии СОТО.

Среди систем, связанных с обработкой структурированных данных из различных БД и отвечающих представленным требованиям, особое место занимают два класса систем – OLTP (Online Transaction Processing) и OLAP [11]. OLAP и OLTP – это системы, предназначенные для ввода, структурированного хранения и обработки информации из БД в режиме РВ.

Следует отметить, что, как видно из приложения 1.4, особый интерес для рассматриваемой задачи представляют именно *OLAP-системы*, предназначенные для аналитической обработки больших объемов информации, представленной в разнородных источниках и в различных форматах. Приложения OLTP, как правило, автоматизируют структурированные, повторяющиеся задачи обработки данных, такие как ввод заказов и банковские транзакции, для которых критична скорость выполнения и безотказность. В то же время OLAP-системы (также называемые ИАС) не имеют таких строгих требований и предполагают агрегацию информации по многомерному принципу, формирование выводов и их последующую запись, что характерно для задачи оценивания ТС и надежности СОТО. Отличия требований, предъявляемых к OLTP и OLAP-системам, представлены в таблице 1.1 [11].

Таблица 1.1. Требования, предъявляемые к OLTP и OLAP-системам

Характеристика	Требования к OLTP-системе	Требования к системе анализа
Степень детализации хранимых данных	Хранение только детализированных данных	Хранение как детализированных, так и обобщенных данных
Качество данных	Допускаются неверные данные из-за ошибок ввода	Не допускаются ошибки в данных

Формат хранения данных	Может содержать данные в разных форматах в зависимости от приложений	Единый согласованный формат хранения данных
Допущение избыточности данных	Должна обеспечивать максимальная нормализация	Допускается контролируемая денормализация (избыточность) для эффективного извлечения данных
Управление данными	Должна быть возможность в любое время добавлять, удалять и изменять данные	Должна быть возможность периодически добавлять данные
Количество хранимых данных	Должны быть доступны все оперативные данные, требующиеся в данный момент	Должны быть доступны все данные, накопленные в течении продолжительного интервала времени
Характер запросов к данным	Доступ к данным пользователей осуществляется по заранее составленным запросам	Запросы к данным могут быть произвольные и заранее не оформлены
Время обработки данных	Время отклика системы измеряется в секундах	Время отклика системы может составлять несколько минут
Характер вычислительной нагрузки на систему	Постоянно средняя загрузка процессора	Загрузка процессора формируется только при выполнении запроса, но на 100%
Приоритетность характеристик системы	Основными приоритетами являются высокая производительность и доступность	Приоритетными являются обеспечение гибкости системы и независимости работы пользователей

1.2.3.2. Назначение OLAP-систем

Определение 1.14. OLAP – технология оперативной аналитической обработки данных, использующая методы и средства для сбора, хранения и анализа многомерных данных в целях поддержки процессов принятия решений [11].

Основное назначение OLAP-систем в сфере BI – поддержка аналитической деятельности, запросов (часто называемых «ad-hoc») пользователей-аналитиков [4, 11, 20, 172, 230]. Основоположник реляционной алгебры Э. Кодд в 1993 г. опубликовал статью, в которой дал критическую оценку реляционного подхода в связи с его малой пригодностью в задачах многомерного анализа данных в смысле множественности представлений, удобных для бизнес-аналитиков, и определил 12 требований к OLAP-системам (см. таблицу 1.2) [230].

Таблица 1.2. 12 требований Э. Кодда к OLAP-системам

№	Требование к многомерному представлению данных	Содержательное пояснение
1.	Многомерное представление данных	Средства должны поддерживать многомерный на концептуальном уровне взгляд на данные.
2.	Прозрачность	Пользователь не должен знать о том, какие конкретные средства используются для хранения и обработки данных, как они организованы и откуда они берутся.

3.	Доступность	Средства должны сами выбирать источник данных и связываться с ним для формирования ответа на данный запрос.
4.	Согласованная производительность	Производительность не должна зависеть от количества измерений в запросе.
5.	Поддержка архитектуры «клиент-сервер»	Средства должны работать в архитектуре «клиент-сервер».
6.	Равноправность всех измерений	Ни одно из измерений не должно быть базовым, все они должны быть равноправными.
7.	Динамическая обработка разреженных матриц	Неопределенные значения должны храниться и обрабатываться наиболее эффективными способами.
8.	Поддержка многопользовательского режима работы с данными	Все многомерные операции должны поддерживаться многими пользователями.
9.	Поддержка операций на основе различных измерений	Все многомерные операции должны единообразно и согласованно применяться к любому числу любых измерений.
10.	Простота манипулирования данными	Средства должны иметь максимально удобный и естественный пользовательский интерфейс.
11.	Развитые средства представления данных	Средства должны поддерживать различные способы представления данных.
12.	Неограниченное число измерений и уровней агрегации данных	Не должно быть ограничений на число поддерживаемых измерений.

К числу дополнительных требований Э. Кодд отметил хранение результатов отдельно от исходных данных. OLAP-система должна реализовывать так называемый *ETL-принцип* (E – extraction, T – transformation, L – loading, англ. – извлечение – преобразование – загрузка), работая в режиме чтения-записи, при котором после модификации исходных данных результаты должны быть сохранены в другое место [11]. Таким образом обеспечивается безопасность и целостность исходных данных. Кроме того, OLAP-система не вмешивается в работу внешних АС, работающих с источниками данных.

Также получил распространение свод требований к ИАС «FASMI». Название теста является аббревиатурой английских слов [11, 20]:

- «*Fast*», Быстрый. Ограничения по времени к ответам системы на запросы пользователей.
- «*Analysis*», Анализ. Требование по возможности решения аналитических задач различного характера в ПрО в соответствии с принципом «программирования без программирования».
- «*Shared*», Разделяемый. Требование по ограничению доступа пользователей к исходным для анализа данным в соответствии с выполняемыми ими задачами.

- «*Multidimensional*», Многомерный. Требование по возможности многомерного концептуального представления данных различных источников.
- «*Information*», Информация. Требование по обеспечению доступа к разнородным источникам данных. Иначе говоря, реализация интеграции данных.

В разработке OLAP-систем существенную роль играют методы согласования данных при их интеграции на основе концепции *гиперкуба* (см. приложение 1.9.1). Зачастую при этом в качестве методической основы используется стандарт CWM (Common Warehouse Metamodel), который имеет ряд существенных недостатков при решении задач интеграции данных и приложений (см. приложение 1.9.2).

1.2.3.3. Анализ возможности применения OLAP в задаче информационно-аналитической поддержки ЖЦ СОТО [64]

Проведенный анализ существующих принципов и технологий OLAP показывает, что СИАП ЖЦ СОТО в определенном контексте относится к классу OLAP-систем. Предпосылки для этого утверждения рассмотрены в разделе 1.2.3.1. Кроме того, можно выделить следующие требования к СИАП ЖЦ СОТО:

- возможность делать запросы к источникам данных в терминах ПрО;
- возможность автоматического выбора источника данных при осуществлении запроса;
- соответствие СИАП ЖЦ СОТО требованиям теста FASMI (см. раздел 1.2.3.2);
- функционирование на основе ETL-принципа: извлечение – преобразование – загрузка;
- результаты обработки данных о состоянии СОТО должны сохраняться в отдельном от источников данных ресурсе с целью поддержания целостности и безопасности исходных данных для нормального функционирования уже существующих в организациях АС;

– использование концепции витрин данных как метамodelей источников данных при условии возможности построения сложных зависимостей между параметрами различных систем в соответствии с принципами многомерного гиперкуба и операциями на нем.

Стоит также отметить, что OLAP-системы пока не получили широкого распространения на отечественных промышленных предприятиях, хотя примеры их использования существуют [64]. В связи с этим, следует сделать вывод, что в этой области требуется разработка новых ПК, решающих соответствующие задачи.

1.2.4. Методы интеграции информационных ресурсов

Интеграция информационных ресурсов является актуальной задачей при формировании ЕИП организаций, связанных единой Про и общими задачами. Такая информация требует обработки, обобщения и анализа для формирования выводов о функционировании комплекса организаций в целом. «Наиболее часто эта задача встречается при агрегации данных – учетные системы генерируют стандартные периодические отчеты, которые являются входными данными для аналитических систем» [21]. Задача технологий интеграции состоит в преодолении неоднородности, присущей ИС, которые создавались и создаются на разных аппаратных платформах, используя разную функциональность, различные типы и модели данных, их компоненты различаются по автономности, имеют различную производительность и др. [207].

Как уже ранее было отмечено (см. разделы 1.2.1, 1.2.3) в задачи СИАП ЖЦ СОТО входит интеграция информационных ресурсов (см. определение 1.15). В соответствии с определенным выше интеграция основана на концепции ИИС (ЕИП). С этой позиции необходимо рассмотреть существующие методы интеграции информационных ресурсов организаций, представленных в виде совокупности данных из существующих в организациях ИС, а также методов интеграции приложений, задачи которых будут изложены далее.

Определение 1.15. Интеграция данных в ИС понимается как обеспечение единого унифицированного интерфейса для доступа к некоторой совокупности неоднородных независимых источников данных [86].

Характерные черты модели ИИС представлены в таблице 1.3 [11, 20].

Таблица 1.3. Характерные черты модели ИИС

№	Черта ИИС	Содержательное пояснение
1.	Предметная ориентированность	Означает компоновку пулов информации по определенным ПрО или целям в виде соответствующих согласованных моделей. Модели должны предполагать возможность анализа данных.
2.	Интегрированность	Предусматривает сбор и обработку информации по определенной ПрО из различных источников и превращение ее в организованный по заданным правилам, подчиненным определенной цели, массив в виде гиперкуба.
3.	Неизменчивость	Результаты обработки данных в ИИС подлежат только накоплению в заданном порядке и при необходимости предполагают только чтение.
4.	Поддержка хронологии	Заключается в привязке данных ко времени. Информация, содержащаяся в хранилище, рассматривается в историческом аспекте. Используют такие типы привязки ко времени, как: к моменту совершения события, к моменту фиксации изменения в ИС, комбинированные методы, к моментам прохождения данных по логическим цепочкам.

Интеграцию с точки зрения программной поддержки можно разделить на три вида: процедурный, декларативный, смешанный (или процедурно-декларативный). Сравнительный анализ видов интеграции представлен в таблице 1.4. [23].

Таблица 1.4. Виды программной поддержки интеграции информационных ресурсов

Вид интеграции	Содержательное пояснение	Достоинства	Недостатки
Процедурный	Предполагает жесткое кодирование процедур интеграции, основанное на конкретных моделях данных.	Отсутствие ограничений по применению к БД различных структур и моделей.	– необходимость реализации модулей под каждый источник данных; – сложность сопровождения; – недостаточная гибкость; – неустойчивость к изменению модели данных.
Декларативный	Соотносит модели данных между собой на уровне описания, обрабатываемого стандартным модулем объединения, не зависящим от моделей данных	– максимальная универсальность; – устойчивость к изменению модели данных; – минимальные средние накладные расходы на каждый источник данных.	Ограниченная область применимости, так как требуется использование соответствующей универсальной метамодели во всех источниках данных.

Смешанный	Использует процедурный метод для интерпретации данных каждого из источников в целях перевода данных в универсальную мета-модель более высокого уровня, в которой модели данных описываются однообразно.	Обладает преимуществами и декларативного и, процедурного подходов	Лишен недостатков декларативного и процедурного подходов
-----------	---	---	--

В концепции ЕИП предусматривается, что в роли информационных ресурсов ИС выступают не только данные, но и различные приложения [23]. В таком случае создаются предметно-ориентированные модули обработки данных, общающиеся между собой. Такой подход соответствует распределенной, одноранговой архитектуре взаимодействия ИС. Это позволяет минимизировать дублирование приложений.

Таким образом, говорят о двух взаимодополняющих технологиях: используя Enterprise Information Integration (ЕИИ), можно объединить источники данных, а с помощью интеграции приложений предприятия (Enterprise Application Integration (ЕАI)) объединяются процессы [23, 109, 207]. Технологии ЕИИ ориентированы на отчетные и аналитические функции, а ЕАI – на автоматизацию бизнес-процессов. Основными подходами интеграции являются: передача сообщений, физическая интеграция данных (ХД, репликация данных), логическая интеграция данных, мониторы транзакций и серверы приложений, интеграция автоматизированных бизнес-процессов. В приложении 1.10 был проведен анализ существующих методов интеграции информационных ресурсов. Результат анализа показал, что современное состояние интеграционных технологий, моделей и методов не позволяет эффективно решать множество практических задач из-за функциональной ограниченности существующих подходов. При этом существующие программные продукты не ориентированы на совместную семантическую интеграцию данных и приложений и большей частью представлены только за рубежом.

1.2.5. Результаты анализа технологий информационной поддержки жизненного цикла сложных объектов

В результате исследования технологий информационной поддержки ЖЦ СОТО можно заключить следующее.

Для многих современных предприятий характерна частичная, фрагментарная компьютеризация отдельных видов деятельности при высоком уровне человеческого фактора в организационных и технологических процессах в СОТО. Кроме того, характерна неоднородность АС, которые создавались и создаются на разных аппаратных платформах, используя разную функциональность, различные типы и модели данных, их компоненты различаются по автономности, имеют различную производительность и др. В результате, по совокупности факторов, это приводит к снижению эффективности функционирования организаций, качества и надежности производимых и эксплуатируемых изделий.

Для устранения этих недостатков в настоящее время предлагается комплекс CALS-технологий, ориентированных на интеграцию информационных ресурсов и процессов (формирование ЕИП), связанных с изделием на всех этапах его ЖЦ, однако не учитывающий задачу автоматизированного оценивания ТС, качества и надежности изделий в ориентации на снижение количества ошибок при проведении анализа, автоматизацию рутинных задач обработки информации о состоянии, оперативную актуализацию сведений для ЛПР.

Интеграция информационных ресурсов является актуальной задачей при формировании ЕИП организаций. Однако проведенный анализ показал, что современное состояние интеграционных технологий, моделей и методов в настоящее время не позволяет эффективно решать множество практических задач из-за функциональной ограниченности существующих подходов и предполагают лишь консолидацию данных (без учета их семантики) на том или ином уровне абстракции информационного взаимодействия.

Проведенный анализ также показал, что большей частью реализация CALS на основе стандарта STEP используется в крупных иностранных компаниях при низком уровне внедрения в РФ. Эта ситуация может быть объяснена тем, что предлагаемый в STEP подход требует изменения существующей на предприятиях инфраструктуры АС, что требует существенных ресурсов и длительного срока внедрения (в среднем, 3-5 лет). Кроме того, в целом, систем, реализующих CALS, практически нет (см. приложение 1.7). Распространены ИС, выполняющие частные задачи. Стоит отметить, что стандарт STEP не поднимает вопрос об оценивании ТС и надежности изделий, рассматриваемый в рамках CALS-технологий. Предлагаемый в рамках STEP язык EXPRESS также имеет множество недостатков (см. приложение 1.7.2).

В связи с этим, в настоящее время требуется разработка модельно-алгоритмического комплекса и соответствующей реализующей СИАП ЖЦ СОТО, характеризующейся следующими аспектами.

Информационно-аналитическую поддержку ЖЦ СОТО следует рассматривать, с одной стороны, как мониторинг состояния СОТО специального вида, а с другой, как комплекс оперативных аналитических вычислений (OLAP) с использованием методологического аппарата на основе G-моделей для решения частных аналитических задач различных этапов ЖЦ (см. разделы 1.2.2, 1.2.3.1).

В соответствии с принципами CALS в системе должны выполняться в режиме РВ задачи информационной поддержки управления качеством, обеспечения надежности, управления данными об изделии, управления конфигурацией, ПИ, контроля исполнения бизнес-процессов. В соответствии с задачей управления изменениями должна быть возможность оперативного внесения модификаций в модели функционирования СИАП ЖЦ СОТО.

Необходимо формирование ИИС (ЕИП), характеризующегося чертами предметной ориентированности, интегрированности, неизменчивости и под-

держки хронологии, для доступа к источникам данных, их совместной обработки, анализа совокупности характеристик СОТО и записи результата с использованием концепции многомерного гиперкуба и операций на нем.

Функционирование должно опираться на ETL-принцип: извлечение – преобразование – загрузка. При этом результаты обработки и анализа должны в явном виде определять текущее состояние СОТО и предполагать сохранение в отдельном от источников данных ресурсе в соответствии с такой моделью состояния, которая описывает конфигурацию изделия и связанные с ней атрибуты. Отдельный ресурс необходим для поддержания целостности, сохранности и конфиденциальности исходных данных и независимого функционирования уже существующих в организациях АС.

Необходимо обеспечить совместную реализацию принципов ЕП и ЕАИ на основе создания одноранговой сети параллельных и асинхронных автоматизированных интероперабельных процессов решения аналитических задач в соответствии с требованиями FASMI (см. раздел 1.2.3.2), поддерживающих обмен сообщениями и доступ к источникам данных с помощью витрин данных, основанных на спецификациях в соответствии с их предметной ориентацией.

Необходимо использование комбинированных декларативно-процедурных спецификаций интеграционных событийно-ориентированных процессов как семантических посредников (медиаторов), где возможно проектирование запросов выборки и функций обработки интегрируемых ресурсов с возможностью оперирования терминами ПрО, и обеспечивается возможность автоматического выбора источника данных.

1.3. Перспективные технологии программной инженерии информационных систем

1.3.1. Инженерия требований при разработке информационных систем

1.3.1.1. Необходимость определения требований к информационным системам

Поскольку в рамках исследования была сформулирована задача разработки специализированной СИАП ЖЦ СОТО необходимо определить, на каких модельно-алгоритмических и методических основах должна функционировать такая система. На сегодняшний день сформировалось несколько направлений формализации процессов разработки ПО. Можно выделить такие актуальные аспекты разработки, как задание *спецификаций* требований к ПО, выбор модели ЖЦ разработки ИС, *концептуальное моделирование* ПрО с использованием различных САПР, *верификация* и валидация моделей, автоматическая генерация кода, выбор оптимальной методологии разработки ПО, интеллектуальные информационные технологии (ИИТ) и др. [70, 87, 128, 194, 202, 238].

Разработка надежного и качественного ПО является на настоящий момент весьма актуальной и сложной задачей. Современные разрабатываемые комплексы ПО достигают существенных размеров. В связи с этим, становится критически важным нахождение способов такой организации самой программной системы и процессов ее ЖЦ (проектирование, разработка, тестирование, эксплуатация и др.), чтобы свести к минимуму вероятность появления серьезных ошибок [194]. Принято начинать проектирование ПО с задания программных требований (Software Requirements) – характеристик ПО, которым оно должно соответствовать. В этой связи в рамках научного направления, называемого инженерией требований, рассматриваются вопросы извлечения (сбора), анализа, специфицирования и утверждения требований [128, 167, 238]. При этом существенную роль играет формализация *спецификаций*, анализ проектирования которых рассмотрен в приложении 2.1.

1.3.1.2. Выводы об использовании спецификаций при проектировании систем информационно-аналитической поддержки

В результате проведенного анализа можно отметить, что наблюдается тенденция, в рамках которой ведутся попытки научно обоснованного определения требований к разрабатываемым ИС. При этом особым образом определяют процесс предъявления требований и выделяют необходимость:

- формализации спецификаций требований для их однозначной интерпретации и возможности компьютерной обработки;
- выделения функциональных и нефункциональных требований;
- соответствия требований критериям, описанным в приложении 2.1.1;
- комбинирования разнотипных требований;
- определения спецификации в виде некоторой модели;
- декомпозиции требований на иерархию спецификаций по разным уровням представления от общего представления о ПРО и вплоть до моделей вычислительных процессов;
- согласования совокупности спецификаций на основе процесса поэтапного уточнения от постановки задачи и вплоть до исходного кода программы с использованием средств верификации.

1.3.2. Методы разработки и жизненный цикл информационных систем

1.3.2.1. Связь автоматизации проектирования информационной системы и её жизненного цикла

Так как в конечном счете СИАП ЖЦ СОТО представляет собой ПО и с учетом рассмотренного в разделе 1.3.1, важным вопросом является формирование упорядоченной последовательности этапов ЖЦ разработки (проектирования) ПО для повышения эффективности этого процесса и качества его результата [129, 146]. При этом, повышение требований к функционированию организаций и предприятий, многоаспектное усложнение СОТО требует в свою очередь наличия более сложных ПК СИАП ЖЦ СОТО, которые необходимо поддерживать, масштабировать, модифицировать. В этих условиях

наибольшего эффекта можно добиться, если каждый этап ЖЦ проектирования ПО и переходы между ними будут автоматизированы таким образом, чтобы на каждом из них можно было выделить формализованные в явном виде знания на машинно-интерпретируемых языках различной степени абстракции, начиная от формализации целей, задач, требований и описания ПрО и вплоть до схем организации вычислительного процесса. При этом должна быть реализована обратная связь от нижестоящих этапов к вышестоящим для облегчения процессов модификации и масштабирования ПО. В настоящее время при разработке ПО зачастую используется международный стандарт программной инженерии ISO/IEC 12207, основанный на книге знаний SWEBOOK комитета Software Engineering Coordinating Committee [55, 240].

К настоящему времени в рамках ИТ сложились определенные устоявшиеся подходы разработки ПО, которые показали на практике свою применимость. Например, в литературе [129] предлагаются следующие этапы ЖЦ ПО: подготовка, планирование, моделирование, конструирование, развертывание. Такое описание имеет, скорее, только содержательное значение, характеризует ручной переход между этапами разработки ПО и привычно в традиционном программировании. Более формальный и значимый в контексте ИИТ подход был предложен Нариньяни А.С., предполагающий следующую упорядоченную информационную совокупность, указывающую на ход разработки ИС: исследуемый объект, его модель, задача, функция, алгоритм, вычисление [120]. Такой подход указывает на то, что объект непосредственно связан с своей моделью, которая его описывает через его характеристики (атрибуты) и связи между ними, описываемые посредством вычислительных задач, для которых может быть найден алгоритм их решения, по которому могут быть проведены вычисления. Ильин В.Д. в своей работе [73] указывал на то, что процесс разработки ПО должен предполагать декомпозицию в виде иерархии *спецификаций* проектных решений с повышением степени детализации на каждом уровне, где каждая такая спецификация ориентирована на описание определенных аспектов ПрО и будущего ПО [73]. На каждом этапе осуществляется

верификация результатов проектирования, затем осуществляется переход на следующий нижележащий уровень. Процесс заканчивается тогда, когда детализация достаточна для решения исходно поставленных задач.

Указанные подходы намечают тенденцию, в которой возможно использование ЭВМ уже на ранних этапах разработки ПО. При этом наблюдается взаимосвязь между степенью формализованности используемых методов и моделей на различных этапах разработки ПО и, соответственно, началом использования ЭВМ. В связи с этим, в приложении 2.2 был проведен анализ того, насколько удовлетворяют существующие на сегодня модели ЖЦ ПО и методы его разработки предъявленным выше требованиям. Результат анализа показал, что на сегодняшний день видится перспективным такой автоматизированный процесс разработки ПО на основе *CASE-систем* (Computer Aided Software Engineering), при котором достигается на основе *системного моделирования* разделение труда по проектированию так, что каждый специалист на разных этапах проектирования от постановки задачи и вплоть до формирования исходного кода программы, оперируя своими терминами и знаниями, мог создавать собственную спецификацию, уточняя то, что известно ему по специализации, с помощью некоторой формальной системы и соответствующей ей реализующей системы на ЭВМ, языковые средства которой по своим синтаксическим и семантическим свойствам ориентированы в точности на описываемые им аспекты ПрО. При этом создаваемые спецификации верифицируются в автоматическом режиме как в отдельности, так и по их совокупности, а код генерируется автоматически. Этот подход позволил бы существенно сократить количество ошибок и сохранить высокую степень точности описания решаемой задачи, а значит и качества ПО. При этом в рассматриваемой задаче информационно-аналитической поддержки ЖЦ СОТО явно выделяются следующие этапы специфицирования: описание словаря ПрО, как совокупности спецификаций источников, данные которых подлежат интеграции, описание совокупности моделей бизнес-процессов и уточняющих спецификаций процессов решения

взаимосвязанных аналитических задач, а также описание функциональных моделей, конкретизирующих вышеуказанные процессы с точек зрения порядка сбора данных (интеграционных процессов) и их обработки и анализа (алгоритмов вычисления). Однако анализ также показал, что существующие методы проектирования, основанные на *концептуальном моделировании*, недостаточно формализованы и используются в большей степени аналитиками для визуального описания без прямой привязки к созданию ПО, а для предлагаемых диаграмм, характеризующих точки зрения специалистов на функционирование системы, не вполне ясно, как их соотносить между собой, чтобы получить согласованную систему спецификаций.

1.3.2.2. Концептуальное моделирование предметной области

Существенную роль в современных CASE-системах играет *концептуальное моделирование*. Этап концептуального моделирования определен в рамках ИС в областях инженерии знаний (или ИИТ), технологий программирования и БД [89, 145]. В соответствии со сложившимися методологиями нисходящего проектирования ИС, начальным этапом этого процесса является формирование спецификации абстрактного представления ПрО, независимого от ИТ, которые предполагается использовать для реализации разрабатываемой системы. Такая спецификация представляется с помощью подходящих выразительных средств – языков концептуального моделирования, а этот этап называется концептуальным моделированием ПрО. Концептуальные модели описывают задачи независимо от технологии и стратегии, используемых для их решения. При этом моделируются сущности ПрО, их концептуальные структуры, характерные отношения между ними и их поведение.

Определение 1.16. *Концептуальное моделирование* – это процесс выявления, анализа и описания релевантных его целям сущностей ПрО, взаимосвязей между ними, ограничений, которым они должны удовлетворять, а также их поведения [89]. Результатом концептуального моделирования является аб-

структное описание ПрО, которое называют *концептуальной моделью* (схемой) ПрО. Степень формальности концептуальной модели зависит от характера языка, используемого для её представления.

Концептуальная схема представляется в терминах типов сущностей ПрО и типов связей между ними, а также налагаемых на них ограничений целостности. Выразительными средствами для представления результатов концептуального моделирования ПрО являются разнообразные языки концептуального моделирования. Существует большое разнообразие языков концептуального моделирования, различающихся степенью их формализации, выразительности и базовыми парадигмами моделирования. По одной из классификаций языков этого назначения, предусматривается три их категории: языки, базирующиеся на сущностях (*entity-oriented*), коммуникациях (*communication-oriented*) и поведении (*behavior-oriented*) [89]. Рассмотрим некоторые аспекты концептуального моделирования более подробно.

Концептуальное моделирование в теории баз данных

Как уже было отмечено, неотъемлемым элементом СИАП ЖЦ СОТО является банк данных, и соответственно БД в его составе. В традиционных технологиях разработки систем БД концептуальная схема ПрО используется только на стадии проектирования БД [89]. Синтезированная концептуальная схема отображается в среду СУБД, выбранной для реализации системы. Результат отображения представляется средствами языков определения данных конкретной выбранной СУБД. Концептуальная схема БД является результатом отображения концептуальной схемы ПрО в среду данной СУБД и описывает представление БД в целом в терминах даталогической концептуальной модели данных, поддерживаемой этой СУБД. Для концептуального моделирования схем БД зачастую используются модели IDEF1X, Entity-Relation Diagrams (ERD, англ., диаграммы «сущность-связь»), диаграммы UML [30, 127].

Как было отмечено в разделе 1.1, ПрО характеризуется наличием разнообразных структурированных и слабо структурированных данных, разнотипных БД, для которых требуется совместная обработка. В результате рассмотрения различных технологий интеграции данных в разделе 1.2 было выявлено, что в качестве метода семантической интеграции могут быть использованы медиаторы, как семантические посредники. В связи с этим, практического смысла для решаемой задачи в использовании указанных выше моделей нет. В то же время в целом ряде работ [17, 18, 19, 25, 23, 86, 89, 207, 211, 223, 256] отмечается, что наряду с концептуальной схемой ПрО в последние годы в разработках систем БД используют *онтологии*. В то время как концептуальная схема описывает структуру и, возможно, поведенческие аспекты ПрО, онтология определяет понятийную систему ПрО, т.е. совокупность ее понятий (концептов) и отношений между ними. Главное различие между ними состоит в том, что концептуальная схема ПрО отображает релевантные целям системы сущности и связи между ними, а онтология – понятия (концепты) ПрО и семантические отношения между ними. Онтология, таким образом, терминологически дополняет описание ПрО, представленное ее концептуальной схемой. Когаловский М.Р. отмечает, что в технологиях БД формируется направление, в котором предусматривается использовать онтологии в качестве высокоуровневых концептуальных схем ПрО и основы пользовательских интерфейсов SQL-систем БД, позволяющих формулировать запросы данных в терминах онтологий и вместе с тем осуществлять рассуждения (*reasoning*) на онтологиях. Механизмы основанного на онтологии интерфейса становятся при этом посредниками между пользователем с его информационными потребностями, выраженными в терминах онтологии, и традиционной системой БД. Такие системы называют системами доступа к данным, основанными на онтологиях (*Ontology-Based Data Access Systems, OBDA- systems, OBDA-системами*).

Концептуальное моделирование динамики процессов и функциональных отношений

В рамках рассматриваемой в работе задачи существенную роль играет моделирование процессов ПрО, которые описывают порядок выполнения аналитических задач по оцениванию ТС, качества и надежности СОТО. В связи с этим, необходимо рассмотреть существующие концептуальные поведенческие и функциональные модели, ориентированные на описание процессов как:

- взаимосвязанных потоков работ, причинно-следственные связи в которых описывают порядок их выполнения, с использованием категорий событий и действий;
- потоков данных, определяющих порядок их обработки (с реализацией принципа управления по данным);
- структурно-функциональных моделей, определяющих отношения вычислимости между параметрами, характеризующими атрибуты СОТО;
- процессов интеграции данных из разнородных источников (использование запросов к источникам данных в качестве элементов модели) и записи результатов (с реализацией *ETL-принципа* (см. раздел 1.2.3)).

В целом, существует множество формальных и неформальных подходов, связанных с описанием указанных аспектов. В основе формального описания процессов лежат такие системы как алгебра процессов Милнера и взаимодействие параллельных процессов Т. Хоара, π -исчисления процессов, в теории логик примерами описания процессов могут служить исчисление событий, язык STRIPS, основанный на исчислении предикатов [71, 113, 169, 191, 194]. В отечественном научном пространстве в разрезе теории вероятностей над вопросами случайных процессов работали такие выдающиеся математики как А.Н. Колмогоров, А.Я. Хинчин, Е.Е. Слуцкий, А.А. Марков старший и др. К существенным недостаткам таких подходов с позиции решаемой в работе задачи является их ориентация на описание функционирования динамических систем как последовательностей смен состояний (для рассматриваемой ПрО

более характерно описание в виде последовательности взаимосвязанных событий), трудность их применения с точки зрения выразительных возможностей, ряд сложно решаемых вопросов (такие как проблемы представительного и выводимого окружения в логических системах), отсутствие графических нотаций.

В то же время существует направление, связанное с описанием бизнес-процессов как потоков работ, которое уже упоминалось в рамках CALS-технологий в разделе 1.2. Методика Workflow может быть использована как технология интеграции, которая связывает бизнес-процессы с информацией, необходимой для их реализации. Практическая реализация Workflow-систем ставит ряд вопросов интеграции, связанных с необходимостью свести все элементы многокомпонентной системы воедино, преодолеть функциональную несовместимость различных систем [68, 205]. Кроме Workflow, большое распространение получили системы управления документооборотом, или Docflow. Парадигмой Docflow-системы является «поток документов». Для Docflow-систем, так же как и для Workflow-систем, существуют схемы в виде графов, которые состоят из узлов, соединенных различными переходами. Однако по этим графам перемещаются не точки управления, а наборы документов. С позиции решаемой в работе задачи такое представление существенно, так как позволяет описывать последовательность событий и действий и причинно-следственные связи между ними, характерные для процессов анализа информации в ПрО, где события характеризуются появлением некоторой информации о СОТО, а задачи определяют ход обработки такой информации, как это в действительности происходит в рассматриваемой ПрО. Таким образом, подобного вида модели могут быть использованы для описания ПрО. К их числу можно отнести Event Driven Process Chain (англ., цепь процессов, управляемых событиями), Workflow Net (Сеть потоков работ), Workflow Process Definition Language (WPDL, англ., язык определения потоков работ), Yet Another Workflow Language (YAWL, англ., еще один язык потоков работ), Graph-based Workflow Language (GBWL, англ., язык потоков работ на основе

графов), Business Process Modeling Language / Notation (BPML, BPMN англ., язык / нотация моделирования бизнес-процессов), диаграмма активностей и диаграмма переходов состояний UML, IDEF0, IDEF2, IDEF3, Scenario Knowledge Base Language (SKBL, англ., язык баз знаний сценариев), в ряде работ для описания динамики также используются сети Петри, Specification and Description Language (SDL, англ., язык спецификаций и описаний), поведенческие модели Object-Oriented Software Engineering (OOSE, англ., объектно-ориентированная разработка ПО), вычислительные модели и их разновидности – G-модели, Дракон-схемы алгоритмов, диаграммы потоков данных Йордана, функциональный модели SADT [10, 30, 54, 68, 74, 79, 108, 127, 132, 159, 161, 169, 170, 191, 193, 205, 225, 243, 244, 253].

Сравнительный анализ указанных моделей с точки зрения возможности их применения в решаемой задаче и примеры программных продуктов, их реализующих, представлены в разделе 1.4.

1.3.2.3. Анализ результатов исследования методов разработки информационных систем

В результатов проведенного анализа методов разработки и моделей ЖЦ ИС в программной инженерии были выявлены следующие существенные аспекты.

Использование спецификаций, задаваемых на машинно-интерпретируемых языках и спроектированных на основе принципов системного моделирования, позволяет за счет их формального согласования в некоторой степени автоматизировать переходы между этапами ЖЦ разработки ПО за счет использования процессов синтеза, анализа и верификации проектных решений.

Разработка ПО представляет собой последовательное уточнение целей, задач, структуры и компонентов ПО в виде иерархии формально согласованных спецификаций, где последний этап предполагает автоматическую кодогенерацию.

Разработка ПО формально есть задача синтеза проектного решения.

Спецификации разных уровней должны создаваться с помощью соответствующих проблемно-ориентированных языков на каждом уровне на основе экспертного моделирования специалистами с соответствующими специализациями. Например, цепочка может быть следующей: представитель заказчика, бизнес-аналитик, инженер по знаниям, порождающая система вывода.

Выполнение таких работ может осуществляться в автоматизированном режиме с помощью CASE-инструментов на основе нисходящей спиральной или каскадной модели разработки с использованием концептуального моделирования.

В рамках решаемой задачи можно выделить следующие типы моделей: информационные онтологические модели, как описание словарей источников данных, поведенческие модели, характеризующие событийно-ориентированные процессы ПрО и связанные с ними модели интеграционных процессов и согласования аналитических вычислительных задач, функциональные модели для описания решения аналитических вычислительных задач. Их сравнительный анализ представлен в разделе 1.4.

1.3.3. Интеллектуальные информационные технологии в программной инженерии

1.3.3.1. Необходимость использования интеллектуальных информационных технологий для задач информационно-аналитической поддержки

Необходимость соответствия предъявляемым требованиям при проектировании ИАС является аргументом использования формальных методов, обеспечивающих прогнозируемые или заданные показатели качества и достоверность результатов обработки данных в АС. Наилучших результатов можно добиться, внедрив в систему фундаментальные математические теоремы и методы, использование которых гарантирует наличие необходимых показателей точности и достоверности вычисленных результатов. Однако, зачастую разработка ПО на основе строгих аналитических методов либо невозможна, либо не оправдывает своих затрат в силу сложности проектирования. Это связано с тем, что современные организации, предприятия и их изделия относятся к

классу СОТО, характеризуемых территориальной распределенностью, большим количеством циркулирующих параллельных и асинхронных информационных потоков, неполнотой, недоопределенностью и неоднородностью информации, высокими рисками и большими масштабами последствий некорректных решений по управлению, наличием директивного времени на принятие решения, многоструктурностью компонентов и др. [134, 185]. Учет совокупности этих свойств СОТО требует построения сложных математических теорий и больших временных и финансовых затрат на программную реализацию, что непозволительно в современных условиях. В то же время известно, что эмпирического неформализованного опыта специалистов ПрО зачастую достаточно, чтобы обеспечить требуемый уровень качества функционирования организаций. Эта особенность, с точки зрения научного метода, определяет возможность применения эвристических моделей и методов, ограниченных сферой своего применения и в то же время являющихся сравнительно простыми в проектировании по отношению к своим аналитическим "аналогам". В связи с этим, следует предположить, что в ИАС должны быть положены принципы и методы, позволяющие за счет формализации моделей осуществлять извлечение и представление знаний специалистов (экспертов ПрО), а также организовывать вычислительный процесс в ЭВМ на основе этих знаний. Иначе говоря, методы разработки современных ИАС лежат в поле теории ИИ в виде соответствующих экспертных систем (ЭС), реализующих нисходящую модель проектирования, и касаются вопросов формирования баз знаний (БЗ) и БД, решателя и систем общения для специалиста-проектировщика (инженера по знаниям) и конечного пользователя (КП). Системный анализ способов проектирования ЭС с использованием МПЗ, представленный в приложениях 2.3.2, 2.3.3, 2.3.4 показал необходимость использования модельно-ориентированной парадигмы проектирования (подтверждая утверждения о формализации спецификаций в виде соответствующих моделей) комбинированных декларативно-процедуральных *G-моделей* как МПЗ при решении аналитических вычислительных задач, связанных с оцениванием ТС СОТО, для формирования

полимодельных имитационно-аналитических ПК. В то же время был сделан вывод о недостаточности выразительных возможностей G-моделей для моделирования бизнес-процессов и специфирования порядка интеграции данных и приложений в рамках задач информационно-аналитической поддержки ЖЦ СОТО. При этом рассмотрение современных ИИТ в приложениях 2.3.5, 2.3.6 показало, что перспективным направлением является использование совокупности взаимосвязанных онтологий в виде *онтологической системы* как способа интеграции и согласования разнородных знаний с использованием мультиагентного подхода при решении задачи обеспечения семантической *интероперабельности* элементов распределенного ПК информационно-аналитической поддержки ЖЦ СОТО.

1.3.3.2. Анализ результатов исследования существующих интеллектуальных технологий

В результате проведенного анализа существующих ИИТ применительно к решению поставленной в работе задачи были сформулированы следующие выводы.

- Разработка СИАП ЖЦ СОТО предполагает использование методов, развиваемых в рамках ИИТ. Сама система относится к категории ЭС с ориентацией на извлечений и представлений экспертных знаний.
- Необходимо задействование нескольких взаимосвязанных декларативно-процедуральных МПЗ, выразительные возможности которых ориентированы на описание поведенческих, информационных и функциональных аспектов рассматриваемой ПрО.
- На основе проектируемых экспертами моделей должен осуществляться синтез программ СИАП ЖЦ СОТО.
- Процесс проектирования должен осуществляться на основе модельно-ориентированного принципа без обращения к алгоритмическому уровню, реализуя возможность «программирования без программирования».

- Процесс проектирования должен предполагать «многоязычное моделирование» в ориентации на пользователей разных этапов ЖЦ ПО (описание процессов, описание порядка выполнения задач, описание самих задач).
- Для аннотирования и семантической поддержки моделей совместно с МПЗ могут использоваться онтологии, позволяющие явно определять БЗ, ограничения и допущения ПрО, а также верифицировать и согласовывать модели посредством онтологической системы.
- Онтологии также могут быть использованы в составе медиаторов (см. приложение 1.10) для семантической интеграции информационных ресурсов и бизнес-процессов.
- СИАП ЖЦ СОТО должна поддерживать мультиагентность для согласованного функционирования процессов информационно-аналитической поддержки ЖЦ СОТО. Агенты при этом основаны на событийно-ориентированной модели функционирования с использованием экспертных знаний.

1.3.4. Результаты анализа моделей, методов и технологий программной инженерии информационных систем

Анализ моделей, методов и технологий программной инженерии показал следующее.

Соответствие исходной постановке задачи, повышение качества, надежности создаваемого ПО должно основываться на формальной системе последовательно уточняющих согласованных спецификаций требований, моделей ПрО, спецификаций программ с реализацией их верификации по ряду критериев и автоматизированного перехода от одного этапа разработки ПО к другому на основе спиральной или каскадной модели проектирования.

Для реализации указанного принципа в основу системы должны быть положены ИИТ в виде соответствующей ЭС и процедур извлечения и представления экспертных знаний с помощью концептуального моделирования и использования графических нотаций без обращения к алгоритмическому уровню (без программирования в кодах).

Спецификации разных уровней должны создаваться на основе системного моделирования с помощью соответствующих проблемно-ориентированных декларативно-процедуральных МПЗ на каждом этапе разработки.

Система должна покрывать задачи информационного моделирования в части задания онтологических отображений моделей источников данных, поведенческого моделирования в части описания совокупности взаимодействующих событийно-ориентированных бизнес-процессов сбора и обработки данных и функционального моделирования в части описания решений аналитических задач, возникающих в ходе исполнения бизнес-процессов.

Для аннотирования и семантической поддержки моделей совместно с МПЗ должны использоваться онтологии, позволяющие явно определять БЗ, ограничения и допущения ПрО, а также верифицировать и согласовывать модели разных уровней представления с помощью онтологической системы.

Конечной целью системы является синтез программ СИАП ЖЦ СОТО на основе проектируемых экспертами моделей.

Создание СИАП ЖЦ СОТО предполагает её функционирование на основе мультиагентной архитектуры как способа организации взаимодействия программ автоматизированных бизнес-процессов и порядка их информационного взаимодействия.

1.4. Постановка задачи автоматизации информационно-аналитической поддержки жизненного цикла космических средств как сложных объектов и обоснование выбора пути её решения

1.4.1. Требования, предъявляемые к полимодельному комплексу интеллектуальной информационно-аналитической поддержки жизненного цикла сложных организационно-технических объектов

В результате анализа текущего состояния и перспектив развития ИТ в части решения задачи автоматизации информационно-аналитической поддержки ЖЦ СОТО, а также исходя из функционального назначения СИАП ЖЦ СОТО можно сформулировать следующие требования к целевой системе (см. рис. 1.4).

1.	По степени автоматизации информационно-аналитической поддержки ЖЦ СОТО и реализации концепции SALS
1.1	Рассмотрение СИАП как OLAP-системы специального вида, включающей множество частных задач мониторинга состояния СОТО по содержательно различным видам информации
1.2	Ориентация на решение задач информационной поддержки управления качеством, обеспечения надежности, управления данными об изделии, управления конфигурацией, параллельного инжиниринга, контроля исполнения технологических и бизнес-процессов
1.3	Формирование ИИС с чертами предметной ориентированности, интегрированности, неизменчивости и поддержки хронологии
1.4	Модель состояния (электронного паспорта) должна предполагать возможность представления конфигурации изделий
2.	По способу проектирования
2.1	Использование формальной системы последовательно уточняющих спецификаций поведенческих, функциональных и информационных требований
2.2	Спецификации разных этапов разработки должны создаваться на основе совокупности декларативно-процедурных МПЗ с графическими нотациями с целью описания Про без обращения к алгоритмическому уровню (без программирования в кодах)
2.3	Использование «многоязычного моделирования» и проблемной ориентированности МПЗ для экспертов разных специализаций: бизнес-процессов, информационных структур, аналитических задач, алгоритмов их решения
2.4	Использование онтологической системы как формализма для согласования проектируемых моделей и их верификации (семантическое аннотирование или «онтолого-управляемость» модели)
2.5	Автоматический синтез схем программ по описанию каждой модели процесса анализа состояния СОТО
3.	По содержательной интерпретации и выразительным возможностям используемых концептуальных моделей
3.1	Реализация принципа ЕП за счет задания онтологических отображений на источники данных для обеспечения доступа к ним в терминах Про
3.2	Возможность описания совокупности бизнес-процессов анализа состояния СОТО в терминах последовательности событий и действий, порядка их взаимодействия через сообщения
3.3	Возможность описания порядка выполнения аналитических задач, подготовки данных для них и записи их результатов в терминах Про
3.4	Возможность описания решения аналитических вычислительных задач
4.	По способу организации системы
4.1	Результаты обработки должны сохраняться на отдельном от источников данных ресурсе
4.2	Программы могут получать промежуточные данные из ресурса результатов обработки с целью сохранения конфиденциальности источников данных
4.3	Реализация EAI-принципа на основе мультиагентной архитектуры с целью обеспечения interoperабельности процессов информационно-аналитической поддержки в виде комплекса взаимосвязанных программ
4.4	Программы СИАП должны использоваться как семантические медиаторы, осуществляющие интеграцию данных, их совместную обработку и анализ
5.	По принципу функционирования и организации вычислительного процесса
5.1	Возможность осуществлять запросы к разнообразным, структурированным и квазиструктурированным источникам данных с реализацией ETL-принципа
5.2	Функционирование в режиме реального времени
5.3	Параллелизм, асинхронность и потоковость вычислительных процессов; функционирование на основе принципа управления по данным
5.4	Верифицируемость схем программ на предмет их корректности
5.5	Направленность процессов вычислений в соответствии с описанием бизнес-процессов

Рис. 1.4. Требования, предъявляемые к автоматизированной информационно-аналитической поддержке ЖЦ СОТО

1.4.2. Сравнительный анализ моделей решения задачи информационно-аналитической поддержки

В соответствии с материалами разделов 1.1, 1.2, 1.3 был проведен сравнительный анализ существующих *концептуальных моделей* и CASE-систем, их реализующих, применительно к рассматриваемой в работе задаче. Программные продукты выбраны из литературы [74, 127, 34, 113, 133, 212]. В качестве критериев для оценивания были выбраны следующие, предполагающие возможность (см. таблицу 1.5):

- в части информационного моделирования:
 1. описания концептуальных схем разнотипных источников данных в терминах ПрО;
 2. описания взаимосвязей терминов ПрО различных источников данных;
- в части поведенческого моделирования:
 3. описания бизнес-процессов информационно-аналитической поддержки в терминах действий, событий и условий их применения;
 4. описания взаимодействия бизнес-процессов (ЕАИ);
- в части функционального моделирования:
 5. описания структурно-функциональных моделей решения аналитических вычислительных задач, ориентированных на распознавание образов (классов состояния исследуемых объектов);
 6. описания моделей запросов к источникам данных в терминах ПрО;
 7. соответствия ETL-принципу;
- в части проектирования:
 8. верификации спецификаций, схем программ, текста запросов на предмет их корректности;
 9. формальной проверки согласованности спецификаций;
 10. наличия графической нотации;
 11. автоматического синтеза совокупности схем программ информационно-аналитической поддержки с запросами к источникам данных;
- в части функционирования:

12. параллелизма, асинхронности и потоковости вычислительного процесса с реализацией принципа управления по данным;
13. функционирования в режиме РВ;
14. синхронизации программ информационно-аналитической поддержки по событиям.
15. направленности процесса вычислений в соответствие с целью функционирования.

Таблица 1.5. Сравнительный анализ моделей решения задачи информационно-аналитической поддержки ЖЦ СОТО

Модели		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Программные продукты
Инф. модели	IDEF1X, ERD, Диаграмма классов UML	+/-	-	-	-	-	-	-	+/-	-	+	-	-	-	-	-	Erwin, Design/IDEF, S-Designer, PowerDesigner, Oowin, Designer/2000 Rational Rose
	Онтологии для OBDA-систем, Express-модели	+	+	-	-	-	+/-	-	+/-	-	+/-	-	-	-	-	+	Apache Jena, Protege, Unigraphics NX, T-Flex CAD
Концепт. модели процессов	Процессы Милнера, STRIPS, PSL	-	-	-	+	-	-	-	+/-	-	-	-	+	-	-	+	UPPAAL, KRONOS, STRIPS
	EDPC, Workflow Net, WPD, YAWL, GBWL, BPMN, UML, SKBL, SDL, OOSE, IDEF0,2,3, SADT	-	-	+	+	-	-	-	+/-	-	+	-	+/-	-	+	+	SmartDraw, Workflow engine, Div. WF and Modeling Tools, YAWL system, ELMA BPM, Microsoft Visio, IBM Rational SDL, BPwin, PowerDesigner, Dia, Design/IDEF
Функц. модели, потоки данных	DFD, Д-ммы переходов состояний, Сети Петри	-	-	-	-	-	-	-	+/-	-	+	-	+	+/-	-	+	Silverrun, PowerDesigner, CASE – Аналитик, Workflow Petri Net Designer
	Дракон-схемы, Н-модели, Выч. модели, G-модели	-	-	-	-	+	-	-	+/-	-	+	+/-	+	+	+	+	Дракон, ПРИЗ, АПИД/Анализ, ТАО
Онтолог. система СИАП ЖЦ СОТО		+	+/-	+	+	+	+	+	+/-	+	+	+	+	+	+	+	СИАП ЖЦ СОТО

Результаты анализа показали, что создаваемый в рамках исследования полимодельный комплекс СИАП ЖЦ СОТО опережает программные продукты – аналоги, а также в большей мере соответствует решаемой задаче с точки зрения возможностей моделирования.

1.4.3. Формальная постановка задачи автоматизированной информационно-аналитической поддержки жизненного цикла космических средств как сложных объектов

В соответствии с описанием ПрО (см. раздел 1.1), результатами анализа, проведенного в разделах 1.2, 1.3, 1.4.2, и требованиями раздела 1.4.1 может быть сформулирована научно-техническая задача, которую можно выразить в виде следующей коммутативной диаграммы (см. рис. 1.5). Пояснение см. ниже.

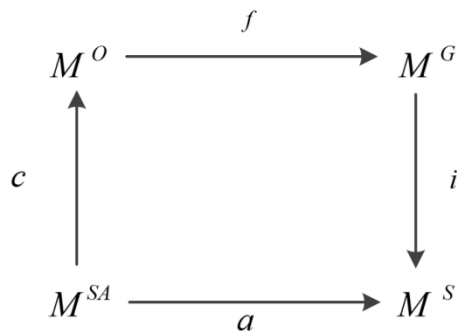


Рис. 1.5. Коммутативная диаграмма задачи автоматизированной информационно-аналитической поддержки ЖЦ КСр как СОТО

Автоматизация информационно-аналитической поддержки ЖЦ СОТО в общем случае предполагает разработку соответствующего ПО. В связи с этим, задачу исследования можно рассматривать как задачу синтеза моделей (схем) программ по совокупности последовательно уточняющих формальных моделей (спецификаций), осуществляя в автоматизированном режиме переход от описания ПрО к описанию программ (см. раздел 1.3). Тогда условия решаемой задачи можно выразить следующим образом.

Дано:

- 1) $M^{SA} = \{m_i^{SA} | i = 1, \dots, n\}$ – конечное множество моделей ПрО, каждая из которых характеризует некоторое типовое (одиночное или серийное) изделие и описывается собственной структурой в соответствии с принципами системного моделирования в виде некоторой семантической сети [164], причем:

$$m_i^{SA} = \langle Ob, Ph, Pr, R \rangle, \quad (1.1)$$

где

– $Ob = \{ob_j | j = 1, \dots, m\}$; $Ph = \{ph_k | k = 1, \dots, l\}$ – конечные множества соответственно объектов и явлений ПрО, соотносимых с некоторыми информационными единицами, характеризующими состояние СОТО: документами, данными БД, файлами.

– $Pr = \{pr_v | v = 1, \dots, p\}$ – конечное множество бизнес-процессов информационно-аналитической поддержки ЖЦ СОТО, соотносимых с некоторой ПрО m_i^{SA} и характеризующих порядок выполнения работ по сбору и анализу информации о состоянии СОТО, причем каждый процесс представляется как:

$$pr_v = \langle Event, Action, Condition, Follow \rangle, \quad (1.2)$$

где *Event* – множество событий, характеризующих появление содержательно новой информации о СОТО (исходной или определяемой в процессе анализа), *Action* – множество действий по обработке и анализу информации, *Condition* – множество условий, определяющих начало выполнения тех или иных действий, *Follow* – множество отношений следования, задающих порядок выполнения действий. При этом ЖЦ СОТО в соответствии с определением 1.11:

$$LC = \langle Pr, In \rangle, \quad (1.3)$$

где Pr – множество процессов из (1.1), а $In \subset R$ – конечное множество отношений интероперабельности процессов, существующих в ЖЦ;

– R – конечное множество отношений на множествах Ob, Ph, Pr .

2) $M^O = \{Sk, Sf\}$ – множество, совокупность элементов которого называют операциональной моделью [69], где:

– $Sk = \{sk_h | h = 1, \dots, t\}$ – множество спецификаций (концептуальных схем), задающих семантические отображения на источники данных для доступа к ним в единой терминологии ПрО и конкретизирующие описания элементов Ob, Ph и отношений между ними, причем среди них есть $sk_h^{ep} \in Sk$ – схема (модель состояния) электронного паспорта изделия;

- $Sf = \{sf_v \mid v = 1, \dots, p\}$ – множество спецификаций функциональных требований (моделей согласованных вычислительных задач), уточняющих процессы pr_v .
- 3) $M^G = \{m_v^G \mid v = 1, \dots, p\}$ – множество ВМ, определяющих отношения вычислимости на параметрах в соответствии со спецификациями sf_v [193] (см. раздел 1.2.2).
- 4) $M^S = \{s_v^S \mid v = 1, \dots, p\}$ – конечное множество схем программ, по каждой из которых могут быть организованы вычисления в соответствии с m_v^G [94, 107, 132].

Для решения задачи синтеза множества схем программ по описанию модели ПрО, характеризуемой множеством взаимосвязанных процессов информационно-аналитической поддержки ЖЦ СОТО необходимо сформировать следующее отображение:

$$a: M^{SA} \rightarrow M^S. \quad (1.4)$$

К настоящему времени конструктивного описания решения такой задачи в общем случае не было найдено. Поэтому предлагается указанное отображение сформировать через следующую, представленную на рис. 1.5 композицию частных отображений:

$$1) \quad a = i \circ f \circ c, \quad (1.5)$$

где c – отображение, ставящее в соответствие элементам m_i^{SA} элементы sk_h и sf_v , конкретизирующие информационные и функциональные требования к ПО, f – отображение, ставящее в соответствие спецификациям sf_v ВМ m_v^G , причем операторы в составе m_v^G могут использовать спецификации sk_h , i – известное отображение, ставящее в соответствие m_v^G схему программы m_v^S [132]. При этом для реализации такой композиции отображений необходимо определить формальный аппарат задания моделей M^{SA} , Sk и Sf , считая известным из работы [132] порядок проектирования M^G и M^S .

Для композиции отображений (1.5) необходимо найти оптимальную СИАП ЖЦ СОТО, при которой показатель её функциональной эффективности принимает экстремальное значение:

$$2) \quad E^{ias} = f(\delta) \rightarrow \max_{\delta \in \Delta}, \quad (1.6)$$

где E^{ias} – показатель функциональной эффективности СИАП, рассчитываемый с учетом влияющих на него следующих показателей: F^{ds} – показатель реализуемости ЕИП в соответствии с требованиями и разделом 1.2 (см. раздел 3.4), Q^{ias} – показатель уровня информационно-аналитической поддержки, характеризующий оперативностью получения и достоверностью результатов анализа (см. раздел 1.1.2), причем качество получаемых решений определено, в первую очередь, достоверностью и полнотой исходных данных, согласованностью моделей M^{SA} , M^O , M^G , M^S (см. раздел 3.2.2 и [198, 199, 200, 234]), корректностью программ (см. раздел 3.3 и [132]). В этом случае допустимые альтернативы реализации СИАП задаются как кортежи из заданного множества Δ , которые формируются как декартово произведение $M^{SA} \times M^O \times M^G \times M^S$ рассматриваемых множеств моделей.

1.4.4. Структура и содержание исследований, проведенных в работе

На основе сформулированных выше положений предлагается модельно-алгоритмический комплекс СИАП ЖЦ СОТО, в основу которого положена совокупность взаимосвязанных посредством онтологической системы МПЗ. Такая система позволяет извлечь и формализовать разноаспектные знания экспертов ПрО на всех этапах ЖЦ ПО. Она ориентирована на автоматизацию этапов разработки ПО СИАП ЖЦ СОТО и призвана облегчить процесс его разработки, масштабирования и модификации, повысить эффективность этих процессов, обеспечить прогнозируемый уровень качества ПО и достоверность результатов его функционирования, обеспечить возможность проектирования и тиражирования комплексных БЗ без специальных навыков в программировании на языках высокого уровня, приблизить ЯПЗ к естественным языкам для проектирования в терминах ПрО, обеспечить формализацию спецификаций

требований, автоматический синтез моделей ПО на основе сформированных для ПрО БЗ. Указанные свойства позволяют приблизить качественные характеристики создаваемого ПО СИАП к заданным в требованиях и постановке задачи, за счет чего может быть достигнуто повышение эффективности информационно-аналитической поддержки ЖЦ СОТО. Основные компоненты СИАП ЖЦ СОТО и их связь с ЖЦ ПО в соответствии со стандартом SWEBOOK, а также с требованиями к СИАП и процессу её проектирования представлены в таблице 1.6.

В основе этапа моделирования ПрО лежит положение о том, что ЖЦ ПО начинается с формирования словаря ПрО [240]. В связи с этим, в состав СИАП ЖЦ СОТО включены онтологии, как средство концептуализации ПрО. Онтологии позволяют аннотировать концептуальные модели ПрО, т.е. задавать ограничения на создаваемые сущности и их отношения. Это свойство позволяет верифицировать модели в автоматическом режиме с точки зрения их онтологической выразительности [88, 198]. В СИАП ЖЦ СОТО используется онтологическая система, позволяющая создавать взаимосвязанную расширяемую систему моделей, описывающих различные аспекты ПрО [235]. В качестве онтологии верхнего уровня была разработана модификация онтологии Бунге-Ванда-Вебера (БВВ-онтология, [41]), используемая для формирования концептуальных моделей совокупности бизнес-процессов на основе нотации BPMN, использованы принципы построения мультиагентной архитектуры, введены СВЗ-модели, обеспечивающие концептуализацию процессов интеграции данных и решения аналитических вычислительных задач на основе нотации IDEF5, а также используются G-модели и G-сети [59, 140, 144, 173, 193, 226].

Таблица 1.6. Соотношение этапов ЖЦ ПО, элементов модельно-алгоритмического комплекса СИАП ЖЦ СОТО и требований, предъявляемых к ним

Этапы ЖЦ ПО и задачи в их рамках	Компоненты СИАП ЖЦ СОТО	Требования, предъявляемые к СИАП ЖЦ СОТО
Идентификация требований; формулирование цели; оценка требований; согласование требований; регистрация требований.	Онтологическая система: метаонтология (онтология проектирования ИАС), предметные онтологии (онтологии источников данных), онтология СВЗ, прикладная онтология ПО.	Обеспечение наличия формализованного общего специального словаря (тезауруса) ПрО, ориентированного на однозначно интерпретируемую систему взаимосвязанных понятий в рассматриваемой ПрО.

<p>Спецификация требований:</p> <ul style="list-style-type: none"> - информационных; - поведенческих; - функциональных; - к интерфейсам; - по ограничениям; - к моделям данных. 	<p>Онтологическая МПЗ о бизнес-процессах;</p> <p>Онтологическая МПЗ о согласовании вычислительных задач;</p> <p>ОВМ в виде G-моделей;</p> <p>модель классной доски взаимодействия интеллектуальных агентов;</p> <p>алгоритм структурно-параметрического синтеза G-модели;</p> <p>алгоритм синтеза онтологических запросов.</p>	<p>Рассмотрение СОТО как динамической системы в силу необходимости оценивания состояния СОТО по протекающим во времени процессам на протяжении всего его ЖЦ;</p> <p>использование модельно-ориентированной парадигмы проектирования ПО;</p> <p>полимодельное представление СОТО в силу его децентрализованности, неоднородности процессов и информации;</p> <p>использование методов имитационно-аналитического, структурно-функционального моделирования вычислительных задач в силу невозможности их решения строго аналитически;</p> <p>использование известных нотаций при проектировании моделей ПО в целях облегчения процесса извлечения и представления знаний экспертов ПрО;</p> <p>обеспечение СИАП ЖЦ СОТО языковыми средствами с визуальной нотацией.</p>
<p>Создание архитектуры системы:</p> <p>распределение (декомпозиция) компонент и распределение требований по этим компонентам;</p> <p>верификация архитектуры.</p>		<p>Возможность создания БЗ для последующего тиражирования знаний в задачи других ПрО;</p> <p>возможность проектирования распределенной системы программных компонент на основе единого модельного представления для децентрализованных СОТО;</p> <p>возможность взаимодействия ПО с распределенной системой БД;</p> <p>обеспечение явно определенного порядка взаимодействия программ, решающих аналитические вычислительные задачи.</p>
<p>Реализация:</p> <ul style="list-style-type: none"> - определение: - функций; - интерфейсов; - ограничений; - моделей данных. 	<p>Схемы программ в виде G-сетей;</p> <p>Решатель, основанный на G-автомате и управляющей структуре G-сети;</p> <p>Алгоритм автоматического выбора источника данных;</p> <p>Алгоритм исполнения запросов манипулирования данными, учитывающий потоковую модель вычислений;</p> <p>Использование семантических медиаторов, действующих онтологические схемы в качестве отображений на источники данных;</p> <p>Интегрированная БЗ.</p>	<p>Возможность синтеза онтологического отображения источников данных, схем программ решения вычислительных задач, запросов манипулирования данными;</p> <p>использование принципа "программирование без программирования" и автоматического синтеза схем программ ПО;</p> <p>обеспечение получения результата функционирования в условиях недоопределенной и неполной информации;</p> <p>нахождение результата анализа состояния СОТО за прогнозируемое время (функционирование СИАП в режиме РВ);</p> <p>использование принципа управления по данным, потоковых, асинхронных и параллельных вычислений для обеспечения соответствия реальным процессам в СОТО, характеризуемым естественным параллелизмом.</p>
<p>Интеграция (комплексирование):</p> <p>объединение реализованных компонент в единую систему;</p> <p>верификация интегрированной системы.</p>	<p>Онтолого-управляемое проектирование моделей ПрО на основе табло-алгоритма проверки выполнимости утверждений в онтологиях;</p> <p>Единый аппарат представления терминов источников данных, подлежащих интеграции;</p> <p>Алгоритм проверки онтологической выразительности;</p> <p>Алгоритмы проверки реализуемости ЕИП;</p> <p>Алгоритмы проверки корректности G-сетей;</p> <p>Проверка разрешимости вычислительных задач на основе р-грамматик.</p>	<p>Возможность формирования ЕИП на основе взаимосвязанного комплекса структурно-функциональных моделей с обеспечением интероперабельности, взаимосвязанных понятий словаря ПрО и единой среды проектирования;</p> <p>возможность моделирования децентрализованных СОТО в едином представлении;</p> <p>возможность распределенных вычислительных процессов по техническим компонентам АПК в соответствии с семантикой задач в ПрО.</p> <p>обеспечение единого представления гетерогенных информационных ресурсов в ПрО, в частности данных из выполненных по различным технологиям БД.</p>
<p>Тестирование:</p> <ul style="list-style-type: none"> - верификация; - валидация. 	<p>Аналогично предыдущему пункту.</p>	<p>Ограничение ЯПЗ проектируемых концептуальных моделей понятиями, введенными в словаре ПрО;</p> <p>обеспечение корректности и однозначности понятий словаря ПрО;</p> <p>обеспечение автоматизации верификации моделей в целях упрощения этапа тестирования ПО.</p>
<p>Документирование:</p> <p>описание результатов процесса разработки на каждом из этапов ЖЦ.</p>	<p>Частично обеспечивается сформированными на предыдущих этапах концептуальными моделями с помощью ЯПЗ с визуальной нотацией.</p>	<p>Обеспечение возможности автоматического формирования части документации (ТЗ, программ и методик испытаний), связанной с демонстрацией схем архитектуры ПО, моделей решения вычислительных задач, порядка сбора данных.</p>
<p>Инсталляция:</p> <p>установка ПО на автоматизированных рабочих местах.</p>	<p>Исполнительная система СИАП ЖЦ СОТО на автоматизированном рабочем месте.</p>	<p>Установка программного продукта, удовлетворяющего заданным требованиям, в целевую среду применения.</p>
<p>Сопровождение ПО</p>	<p>Обеспечивается сформированной на предыдущих этапах системой взаимосвязанных концептуальных моделей и наличием БЗ.</p>	<p>При необходимости масштабирования или модификации ПО в соответствии с задачей управления изменениями система должна позволять осуществлять оперативное перепроектирование моделей без программирования в кодах.</p>
<p>Прекращение применения ПО</p>	<p>-</p>	<p>-</p>

1.5. Выводы по разделу

В результате системного анализа состояния исследований задачи автоматизации информационно-аналитической поддержки ЖЦ КСр могут быть сформулированы следующие выводы.

1. Задача автоматизации информационно-аналитической поддержки ЖЦ КК и их изделий продиктована необходимостью семантической интеграции и совместного анализа информационных ресурсов организаций и их АС с целью оперативного обеспечения ЛПР актуальной и достоверной информацией о ТС изделий для повышения эффективности функционирования этих организаций, качества и надежности их изделий.
2. Для решения этой задачи может быть использована концепция CALS, предполагающая решение комплекса задач информационной поддержки ЖЦ СОТО, однако на текущий момент не получившая полноценной реализации ни в отечественных, ни в зарубежных ПК.
3. В результате сравнительного анализа существующих моделей, методов и технологий, потенциально применимых для указанной задачи, оказалось, что ни одна из них в существующем виде не удовлетворяет комплексу требований, предъявляемых к СИАП ЖЦ КСр как сложных объектов.
4. Таким образом, была сформулирована научно-техническая задача, предполагающая разработку модельно-алгоритмического комплекса автоматизированной информационно-аналитической поддержки ЖЦ КСр, предполагающего формирование совокупности взаимосвязанных программ СИАП ЖЦ КСр на основе онтологической системы и концептуализации спецификаций бизнес-процессов, сценариев интеграции данных и приложений, вычислительных задач как моделей извлечения и формализации разноаспектных знаний об организационных и технологических процессах оценивания ТС КСр.

2. СИНТЕЗ МОДЕЛЕЙ И АЛГОРИТМОВ АВТОМАТИЗИРОВАННОЙ ИНФОРМАЦИОННО-АНАЛИТИЧЕСКОЙ ПОДДЕРЖКИ ЖИЗНЕННОГО ЦИКЛА КОСМИЧЕСКИХ СРЕДСТВ

2.1. Онтолого-управляемое моделирование предметной области

2.1.1. Исходные положения, используемые при формализации разрабатываемых моделей

Предлагаемый в работе подход основан на том, что модели *бизнес-процессов* могут быть использованы как модели потоков информационных единиц (например, документов), где события модели – есть эти информационные единицы, а действия – проводимые над ними работы. В конкретном случае, этой работой являются задачи анализа документов (их метаданных, содержащих информацию о СОТО) с целью оценивания ТС, качества и надежности СОТО на основе информации, поставляемой из таких электронных документов. В то же время можно логично предположить, что в более общем случае событием может являться не только документ, но и любая другая информация, имеющаяся в источниках данных, которую необходимо обработать для обеспечения полноты интегрируемой информации в процессе анализа. При этом сами такие процессы характеризуют в таком случае некоторый организационный план, или, возможно, технологический процесс выполнения работ по сбору, обработке и анализу информации из разнообразных источников данных, имеющихся на предприятиях СОТО. В рамках автоматизации таких процессов можно выдвинуть гипотезу, что с событиями необходимо соотносить появление информации о СОТО в источниках данных, т.е. выполнение некоторых запросов к соответствующей СУБД, выполняемых с заданной частотой опроса в потоковом режиме, чтобы постоянно актуализировать информацию, а с работами необходимо соотносить аналитические вычислительные задачи, автоматизирующие ход распознавания состояния СОТО по той информации, которая получена. При этом результат распознавания должен быть записан

(также запросом) в отдельный ресурс, информацию в котором для рассматриваемой задачи и следует называть электронным паспортом изделия. При этом, поскольку сам план выполнения работ предполагает наличие причинно-следственных связей между информационными единицами, выводы, сформированные в результате решения некоторой аналитической задачи, могут быть использованы для дальнейших этапов процесса анализа. Этим самым реализуется принцип «сжимающих отображений» программирования в ограничениях [120], что позволяет шаг за шагом приближаться к конечной точке процесса – факта об окончании этапа работ над изделием. Также необходимо обратить внимание на тот факт, что для унифицированного доступа к разнообразным источникам данных должен быть реализован некоторый механизм, позволяющий формулировать запросы в терминах ПрО, а функционирование системы предполагает сбор результатов исполнения запросов от разных источников и их «переплетение» в ходе решения задач, что позволит семантически связывать эти данные, обнаруживая взаимосвязи в ПрО. При этом сами бизнес-процессы также взаимосвязаны между собой, что характеризует *интероперабельность* процессов этапов ЖЦ. В то же время, по своей сути, бизнес-процессы являются *спецификацией* требований к поведению ПК, и они должны быть уточнены в виде соответствующих структурно-функциональных моделей, где структура наследуется от бизнес-процесса, а функции уточняют, как решать аналитические вычислительные задачи. Известно, что существуют подходы, позволяющие синтезировать схемы программ по таким моделям [120, 133, 193].

2.1.2. Взаимосвязь концептуального и онтологического моделирования

Традиционно, концептуальное моделирование предполагает выявление сущностей ПрО и отношений между ними. Однако в то же время, как было указано в разделе 1.3, прежде всего необходимо определить словарь ПрО. Эта особенность характеризуется известным треугольником Г. Фреге (см. рис. 2.1), указывающим на прямую взаимосвязь реальных объектов (денотатов), их

концептов (понятий, т.е. десигнатов) и знаков (синтаксических имен или слов) [41, 164, 198, 212].

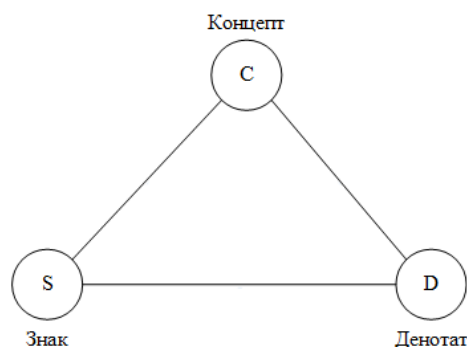


Рис. 2.1. Треугольник Г. Фреге

Если концептуальные схемы инкапсулируют информацию о типах сущностей в ПрО, то *онтологии* связаны с определениями понятий, заданием ограничений на них. Их совместное использование называют онтолого-управляемым моделированием (или онтологическим аннотированием моделей), указывая на то, что сущности ПрО являются индивидами определенных категорий (понятий), а отношения между ними характеризуются отношениями между концептами [89]. В связи с этим, онтологическую модель можно называть семантической, а её конкретную интерпретацию *концептуальной моделью* ПрО. В [118, 119] отмечается, что необходимо ввести построение онтологий как предварительный этап проектирования всех программных систем и особенно традиционных систем обработки данных, так как технологии проектирования современных систем обработки данных предполагают разработку их спецификаций.

Такой подход развивается в рамках OBDA-систем для онтологического аннотирования концептуальных схем БД с целью абстрагирования от технических аспектов СУБД и обращения к источникам данных в терминах ПрО [27, 37, 69, 88, 89, 187, 193]. Поскольку концептуальное моделирование активно используется и для описания поведенческих и функциональных аспектов, аналогичный подход может быть использован и в этой части ИИТ, однако на текущий момент еще не получил широкого распространения [41, 88,

162, 264]. Например, в работе [235] определяется понятие «онтологической чистоты» – показателя корректности и полноты концептуальной модели относительно её онтологии, и отмечается, что онтолого-управляемое проектирование становится легче для понимания при использовании онтологий в силу оперирования естественными для пользователя понятиями. Таким образом, при разработке ЭС онтология может использоваться для [41]:

- формирования и фиксации общего разделяемого всеми экспертами знания о ПрО;
- явной концептуализации ПрО, позволяющей описывать семантику данных;
- обеспечения возможности повторного использования знаний;
- описания функционала ЭС (типов решаемых задач).

В процессе функционирования ЭС онтология может использоваться для обеспечения:

- совместного использования разнородных данных и знаний в рамках одной системы;
- процесса решения задач, составляющих функционал системы;
- лучшего понимания ПрО пользователями системы.

Реализация такого подхода для решения рассматриваемой задачи будет рассмотрена далее.

2.1.3. Онтологии как формализм представления знаний

В качестве теоретического аппарата для формализации онтологий зачастую используется дескрипционная логика [17, 72, 88, 124, 206, 249, 256]. Дескрипционные логики – это семейство формализмов для структурированного представления знаний, созданное для привнесения формальной семантики в структурированные, не основанные на логике, ЯПЗ – семантические сети и языки фреймов. Они реализуют компромисс между выразительными возможностями и вычислительной сложностью рассуждений, сочетая черты разрешимых фрагментов логики первого порядка и модальных логик с некоторыми

расширениями, поскольку некоторые аспекты реального мира трудно представить в этих логиках [33].

Основная сложность состоит в том, что почти все обобщения имеют исключения или соблюдаются только до определенной степени (принцип естественных разновидностей) [169]. В связи с этим, необходимо отделять типовые сущности от исключительных (индивидов). Вторым важным аспектом – использование категорий. Суждение о конкретных объектах проводится на уровне категорий с заданием на них отношений (таксономии, партономии, тезаурусы и пр.), что позволяет задавать описания и ограничения этих объектов. Дескрипционная логика позволяет явно разрешать эти проблемы и в ряде случаев имеет более низкую вычислительную сложность, чем, например, выражения в логике с двуместными предикатами.

Формально онтология в дескрипционной логике определяется следующим образом [72].

Пусть $CN = \{A_1, \dots, A_m\}$ и $RN = \{R_1, \dots, R_n\}$ – конечные непустые множества атомарных *концептов* и атомарных *ролей* (называемых также именами концептов и именами ролей). Аксиоматика дескрипционной логики для CN и RN определяется в зависимости от используемого диалекта (см. приложение 2.3.8.1). В работе использован диалект $SRQIQ^{(D)}$, выразительных возможностей которого оказалось достаточно для моделирования. Традиционно принято задавать описание синтаксиса и семантики на примере тривиального и разрешимого диалекта ALC . Далее будет введен ряд определений из [72] для понимания дальнейшего изложения.

Определение 2.1. Множество концептов логики ALC задается следующим индуктивным определением:

- символы \top и \perp – концепты (истина и ложь);
- всякий атомарный концепт A является концептом;
- если C – концепт, то $\neg C$ – концепт (дополнение концепта C);
- если C и D – концепты, то $C \sqcap D$ и $C \sqcup D$ – концепты (пересечение и объединение);

- если C – концепт, а R – атомарная роль, то $\exists R.C$ и $\forall R.C$ – концепты, причем C называют -последователем;
- никакие другие выражения не являются концептами.

В нотации Бэкуса-Науэра можно сформулировать аналогичную более краткую запись:

$$C ::= \top \mid \perp \mid A \mid \neg C \mid C \sqcap D \mid C \sqcup D \mid \exists R.C \mid \forall R.C, \quad (2.1)$$

где A – атомарный концепт, R – атомарная роль C, D – произвольные концепты.

Определение 2.2. *Интерпретацией* называется пара $\mathcal{I} = (\Delta, \cdot^{\mathcal{I}})$, состоящая из непустого множества Δ , называемого областью данной интерпретации, и интерпретирующей функции $\cdot^{\mathcal{I}}$, которая сопоставляет:

- каждому атомарному концепту $A \in CN$ – произвольное подмножество $A^{\mathcal{I}} \subseteq \Delta$;
- каждой атомарной роли $R \in RN$ – произвольное подмножество $R^{\mathcal{I}} \subseteq \Delta \times \Delta$.

Интерпретирующая функция распространяется на множество всех концептов логики \mathcal{ALC} однозначным образом – индукцией по построению концепта:

- $\top^{\mathcal{I}} = \Delta, \perp^{\mathcal{I}} = \emptyset, (\neg C)^{\mathcal{I}} = \Delta \setminus C^{\mathcal{I}}$;
- $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}, (C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$;
- $(\exists R.C)^{\mathcal{I}} = \{e \in \Delta \mid \text{существует } d \in \Delta \text{ такой, что } \langle e, d \rangle \in R^{\mathcal{I}} \text{ и } d \in C^{\mathcal{I}}\}$;
- $(\forall R.C)^{\mathcal{I}} = \{e \in \Delta \mid \text{для всех } d \in \Delta \text{ таких, что } \langle e, d \rangle \in R^{\mathcal{I}}, \text{ выполнено } d \in C^{\mathcal{I}}\}$.

Определение 2.3. Концепт C *выполним*, если существует такая интерпретация \mathcal{I} , что $C^{\mathcal{I}} \neq \emptyset$. При этом \mathcal{I} называется моделью концепта C .

Определение 2.4. Концепты C и D называются *эквивалентными* (обозначение: $C \equiv D$), если в любой интерпретации \mathcal{I} имеется $C^{\mathcal{I}} = D^{\mathcal{I}}$.

Определение 2.5. Концепт C *вложен* в концепт D (обозначается $C \sqsubseteq D$), если в любой интерпретации \mathcal{I} имеется $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$.

Определение 2.6. Концепты C и D называются *непересекающимися*, если в любой интерпретации \mathcal{I} имеется $C^{\mathcal{I}} \cap D^{\mathcal{I}} = \emptyset$.

Формально концепт соответствует одноместному предикату в исчислении предикатов, а роль – двуместному.

Дескрипционные логики позволяют описывать стабильные интенциональные знания в виде терминологических аксиом (понятий и их взаимосвязей) и изменяющиеся экстенциональные знания в виде утверждений об индивидах. В соответствии с этим делением дескрипционная логика явно определяет БЗ:

$$\mathcal{K} = \mathcal{T} \cup \mathcal{A}, \quad (2.2)$$

где $TBox \mathcal{T}$ (англ., Terminological Box) – конечное множество терминологических аксиом, $ABox \mathcal{A}$ (англ., Assertional Box) – конечное множество утверждений (фактов) об индивидах.

Определение 2.7. Терминологической аксиомой называется выражение вида $C \sqsubseteq D$ или $C \equiv D$, где C и D – произвольные концепты.

Аксиома $C \sqsubseteq D$ истинна в интерпретации \mathcal{I} , если $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$; при этом \mathcal{I} называют моделью данной аксиомы и пишут $\mathcal{I} \models C \sqsubseteq D$. Аналогично, $\mathcal{I} \models C \equiv D$ означает, что $C^{\mathcal{I}} \equiv D^{\mathcal{I}}$.

Определение 2.8. Интерпретацию \mathcal{I} называют моделью терминологии \mathcal{T} и пишут $\mathcal{I} \models \mathcal{T}$, если \mathcal{I} является моделью всех аксиом из \mathcal{T} . Терминология \mathcal{T} называется выполнимой, если она имеет модель.

Для определения $ABox$ вводится также конечное множество IN имён индивидов. Утверждение о принадлежности индивида a концепту C записывается как $a:C$. Утверждение о связи двух индивидов a и b ролью R записывается как aRb .

Определение 2.9. Системой фактов или $ABox$ называется конечное множество \mathcal{A} утверждений вида:

$$a:C, \quad (2.3)$$

где $a, b \in IN$ есть индивиды, C – произвольный концепт, и

$$a R b, \quad (2.4)$$

где $a, b \in IN$ есть индивиды, R – роль.

Факт $a: C$ или aRb верен в интерпретации \mathcal{I} , если $a^{\mathcal{I}} \in C^{\mathcal{I}}$ или $\langle a^{\mathcal{I}}, b^{\mathcal{I}} \rangle \in R^{\mathcal{I}}$, соответственно. При этом \mathcal{I} называется моделью этого факта, что записывается как $\mathcal{I} \models a: C$ и $\mathcal{I} \models aRb$, соответственно. При этом каждому индивиду $a \in IN$ сопоставляется единственный элемент области интерпретации $a^{\mathcal{I}} \in \Delta$.

Определение 2.10. Интерпретация \mathcal{I} называется *моделью системы фактов* \mathcal{A} , если \mathcal{I} является моделью всех фактов из \mathcal{A} . *АВох* \mathcal{A} называется *выполнимым* (в терминологии \mathcal{T}), если \mathcal{A} имеет модель (являющуюся одновременно моделью терминологии \mathcal{T}).

Введенные определения позволяют говорить о том, что проектируемые онтологии можно проверять на корректность (выполнимость аксиом) и осуществлять вывод новых знаний.

Для диалекта $\mathcal{SROIQ}^{(D)}$ также справедливо следующее [125]:

- если R и S – атомарные роли, то $R \sqsubseteq S$ – вложенность ролей;
- если R есть роль, C концепт, а $n > 0$ натуральное число, то $\geq n R.C$ и $\leq n R.C$ есть концепты (кардинальность ролей). Выразимы также ограничения вида $(< n R.C)$ и $(> n R.C)$. Выражение вида $\leq 1R$ называют функциональным отношением, где в данном случае R -последователем является \top ;
- если R – атомарная роль, то R^- является ролью, при этом интерпретация задается как $(R^-)^{\mathcal{I}} = \{\langle e, d \rangle \in \Delta \times \Delta \mid \langle d, e \rangle \in R^{\mathcal{I}}\}$;
- если a есть имя индивида (т.е. $a \in IN$), то $\{a\}$ есть концепт, причем $\{a\}^{\mathcal{I}} = \{a^{\mathcal{I}}\}$.
- если $D \rightarrow d \mid \{c_1, \dots, c_n\}$ – множество примитивных типов данных, то $\exists R_{d_1} \dots R_{d_n}.D$ и $\forall R_{d_1} \dots R_{d_n}.D$ являются концептами, причем $\{c_1, \dots, c_n\} \subseteq \mathbb{C}_d$ – множество констант (значений) типов данных, $n \geq 1$.

Из [69] известно, что у моделей выделяют носитель модели и ее сигнатуру в соответствии с теорией алгебраических систем. Носитель модели представляет собой совокупность сущностей ПрО, сигнатура определяет операции и отношения, возможные на данном носителе:

$$M = \langle A; R_1^{s_1}, \dots, R_n^{s_n} \rangle, \quad (2.5)$$

где M – модель, A – носитель модели, R_i^{si} – i -ое отношение ($i = 1, \dots, n$), совокупность отношений определяет сигнатуру модели. С учетом этого и традиционных определений онтологий в [42, 176], можно утверждать следующее.

Утверждение 2.1. Онтологической МПЗ O называется такая тройка $\langle CN, RN, \mathcal{T} \rangle$, где CN определяет носитель модели, а RN – множество бинарных отношений на CN , определяющее её сигнатуру, \mathcal{T} – интерпретация (см. определение 2.2). Тогда (концептуальной) моделью, спроектированной по МПЗ O , называется такая тройка $\langle IN, RN, \mathcal{T} \rangle$ системы фактов \mathcal{A} , которая выполнима относительно терминологии \mathcal{T} , заданной для O , причем на аксиомах в \mathcal{T} определено множество операций в соответствии с выбранным диалектом дескрипционной логики.

Таким образом, аппарат дескрипционных логик определяет целый класс МПЗ с заданными на них алгебрами в соответствии с диалектом дескрипционной логики. Этот факт может быть использован для того, чтобы определять с помощью онтологий необходимого вида концептуальные модели, где в \mathcal{T} можно задавать их аксиоматику, что и называется онтолого-управляемым проектированием.

2.1.4. Онтологической система для согласования спецификаций предметной области

Для многих исследований характерно использование отдельно взятых онтологий или их линейной совокупности (например, в виде множества предметных онтологий) [6, 21, 25, 27, 43, 48, 82, 116]. В некоторых случаях создается иерархия онтологий для интеграции гетерогенных источников данных. В то же время онтологии как формализм представления знаний могут быть использованы не только для описания статичных представлений о ПрО, но также описывать и процессуальные аспекты, иерархии целей и задач и др. [162, 169, 198, 199, 200]. Кроме того, возможно определение некоторой совокупности взаимосвязанных онтологий, так что при переходе от одной онтологии к другой могут задаваться ограничения, либо расширения для задаваемой системы

фактов, что позволяет более гибко проектировать модели. Такую совокупность онтологий называют *онтологической системой*. На сегодняшний день встречается ряд определений этого понятия, однако на практике соответствующей формальной и реализующей системы в источниках литературы найдено не было [42, 69, 104].

Практическое назначение онтологической системы в рассматриваемом контексте состоит в том, чтобы создавать иерархию согласованных спецификаций, т.е. моделей бизнес-процессов информационно-аналитической поддержки ЖЦ СОТО, концептуальных схем источников данных, моделей согласованных вычислительных задач и в качестве структурно-функциональных моделей, характеризующих функциональные отношения – моделей аналитических вычислительных задач. В связи с этим, онтологическую систему информационно-аналитической поддержки ЖЦ СОТО (далее, для краткости, «онтологическую систему») можно определить следующим образом (см. рис. 2.2).

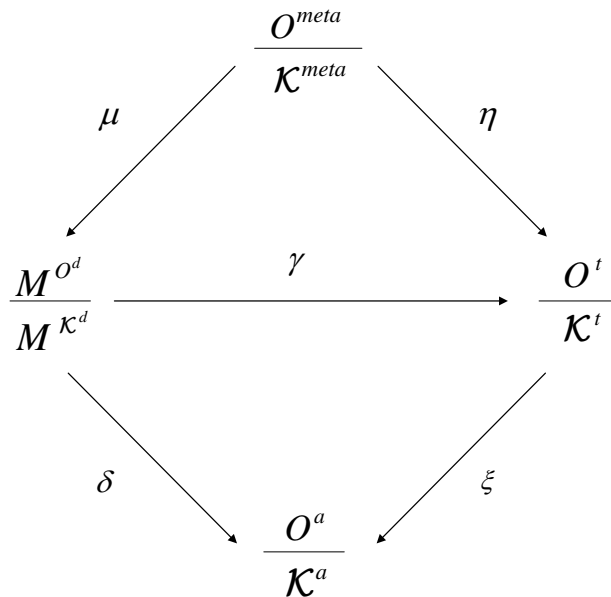


Рис. 2.2. Диаграмма онтологической системы информационно-аналитической поддержки ЖЦ СОТО

Определение 2.11. Формально *онтологической системой* называется кортеж вида:

$$\Sigma^{ias} = \langle O^{meta}, M^{O^d}, O^t, O^a, \Xi^i \rangle, \quad (2.6)$$

где

– $O^{meta} = \langle CN^{meta}, RN^{meta}, \mathcal{I}^{meta} \rangle$ – онтология верхнего уровня ПрО, позволяющая описывать ПрО и бизнес-процессы в ней в соответствии с (1.1) и (1.2), причем для неё определена БЗ $\mathcal{K}^{meta} = \mathcal{T}^{meta} \cup \mathcal{A}^{meta}$, где \mathcal{A}^{meta} – конечное непустое множество утверждений, в рамках которых определено множество экземпляров бизнес-процессов Pr и их взаимосвязи In , что обеспечивает представление ЖЦ СОТО, а в \mathcal{T}^{meta} фиксированное множество правил (определений), по которым они могут быть сформированы (см. раздел 2.2);

$M^{O^d} = \{O_h^d | h = 1, \dots, t\}$, где $O_h^d = \langle CN_h^d, RN_h^d, AN_h^d, \mathcal{I}_h^d \rangle$ – предметная онтология ПрО, позволяющая описывать совокупность объектов и явлений ПрО через множество атрибутов и их значений, причем определено множество БЗ $M^{\mathcal{K}^d} = \{\mathcal{K}_h^d | h = 1, \dots, t\}$ таких, что каждое конечное непустое множество \mathcal{A}_h^d характеризует h -й источник данных в виде соответствующей модели sk_h . В приложении 3 приведены разработанные и используемые в исследовании модели и алгоритмы доступа к информационным ресурсам СИАП ЖЦ СОТО на основе предметных онтологий;

– $O^t = \langle CN^t, RN^t, \mathcal{I}^t \rangle$ – онтология задач ПрО, уточняющая бизнес-процессы в виде совокупности онтологических запросов к источникам данных посредством схем Sk , аналитических вычислительных задач и порядка их применения. Для неё определена БЗ $\mathcal{K}^t = \mathcal{T}^t \cup \mathcal{A}^t$ такая, что \mathcal{T}^t – фиксированное множество правил, по которым могут быть сформированы модели согласованных вычислительных задач, а конечное непустое множество фактов

$$\mathcal{A}^t = \bigcup_{v=1}^p (\mathcal{A}_v^{t,sf} \cup \mathcal{A}_v^{t,def}),$$

где каждый $\mathcal{A}_v^{t,sf}$ характеризует некоторый бизнес-про-

цесс pr_v в виде соответствующей СВЗ-модели sf_v , а $\mathcal{A}_v^{t,def}$ – определяет, как соотносятся элементы pr_v и sf_v (см. раздел 2.3);

– $O^a = \langle CN^a, RN^a, \mathcal{I}^a \rangle$ – *прикладная онтология* PrO , уточняющая СВЗ-модели в виде ВМ как совокупности операторов и параметров, причем среди операторов имеются операторы исполнения онтологических запросов к источникам данных с использованием схем из Sk . Для неё определена БЗ $\mathcal{K}^a = \mathcal{T}^a \cup \mathcal{A}^a$ такая, что \mathcal{T}^a – фиксированное множество правил, по которым могут быть сформированы ВМ (определенное в алгебре ВМ), а $\mathcal{A}^a = \bigcup_{v=1}^p \mathcal{A}_v^a$, где каждый

\mathcal{A}_v^a характеризует некоторую модель sf_v в виде соответствующей ВМ m_v^G (заметим, что отображение из sf_v в m_v^G задается не в рамках аппарата дескрипционных логик, см. раздел 2.4);

Ξ^i – *модель машины вывода*, ассоциированной с онтологической системой Σ^{ias} и реализующей отображения $\mu, \eta, \gamma, \xi, \delta$ (задается совокупностью алгоритмов с заданной последовательностью их применения). Машина вывода предполагает выполнение задач параметрического и структурного синтеза. Параметрический синтез – создание экземпляров понятий. Структурный синтез – активизация отношений между индивидами понятий [69]. Далее, будет показано, что на данном этапе развития ИТ в автоматическом режиме возможна реализация только отображений ξ и δ . Остальные же осуществляются в автоматизированном режиме.

Совокупность связей на диаграмме (см. рис. 2.2) указывает на взаимосвязь онтологий:

- μ характеризует отображение из CN_h^d в $Ob \sqcup Ph$, где именованя ($Ob \cup Ph$) $\subset CN^{meta}$, т.е. все концепты онтологий источников данных принадлежат объектам или явлениям;
- η характеризует отображение из $(CN^{meta} \setminus (Ob \cup Ph))$ в CN^t , реализуемое так, что $\mathcal{A}^{meta,t} = \mathcal{A}^{meta} \cup \mathcal{A}^t$;
- γ характеризует отображение из CN_h^d, RN_h^d (задающих аксиомы для \mathcal{A}_h^d) в элементы онтологических запросов из CN_v^t , формируемых в \mathcal{A}_v^t ;

- ξ характеризует отображение из CN^t в CN^a и из RN^t в RN^a ;
- δ характеризует отображение из CN_h^d, RN_h^d (задающих аксиомы для \mathcal{A}_h^d) в параметры, значения которых представляют непосредственно текст онтологического запроса, являющегося входным значением – константой для оператора исполнения онтологического запроса.

Метаонтология содержит концепты и отношения, необходимые для предметной онтологии и онтологии задач, которые в совокупности могут обеспечивать построение операциональной модели M^O ПрО. Эта модель указывает, как осуществлять преобразование входных данных (информации о СОТО) в выходные данные, характеризующие состояние СОТО (см. раздел 1.4.3). Операциональная модель M – это совокупность концептуальной sk_h (отражает понятийную структуру ПрО) и функциональной sf_v (моделирует функции преобразования входных данных в выходные) структур [69]. Концептуальная структура представляет собой синтаксический аспект предметной онтологии, включающий в себя описание семантики понятий, а функциональная – синтаксический аспект онтологии задач, содержащий их смысловое описание, где sk_h выступает как пассивный компонент, состоящий из данных, а sf_v – как спецификация активного компонента, преобразующего данные. Поскольку каждый pr_v может быть связан с несколькими источниками данных, и, в том числе, с БД электронного паспорта СОТО, то с каждым sf_v может быть связано некоторое подмножество схем источников данных Sk .

Таким образом, онтологическая система со специальной машиной вывода позволяет описывать совокупность бизнес-процессов Pr СИАП ЖЦ СОТО, аннотирующих схем источников данных Sk , уточняющих бизнес-процессы СВЗ-моделей Sf , характеризующих функциональные требования к ПО, по каждой sf_v создавать соответствующую вычислительную модель m_v^G . Взаимосвязи между онтологиями позволяют представить совокупность введенных в онтологическую систему концептов и ролей в единой схеме (см. рис. 2.3). В следующих разделах будет дано более подробное пояснение.

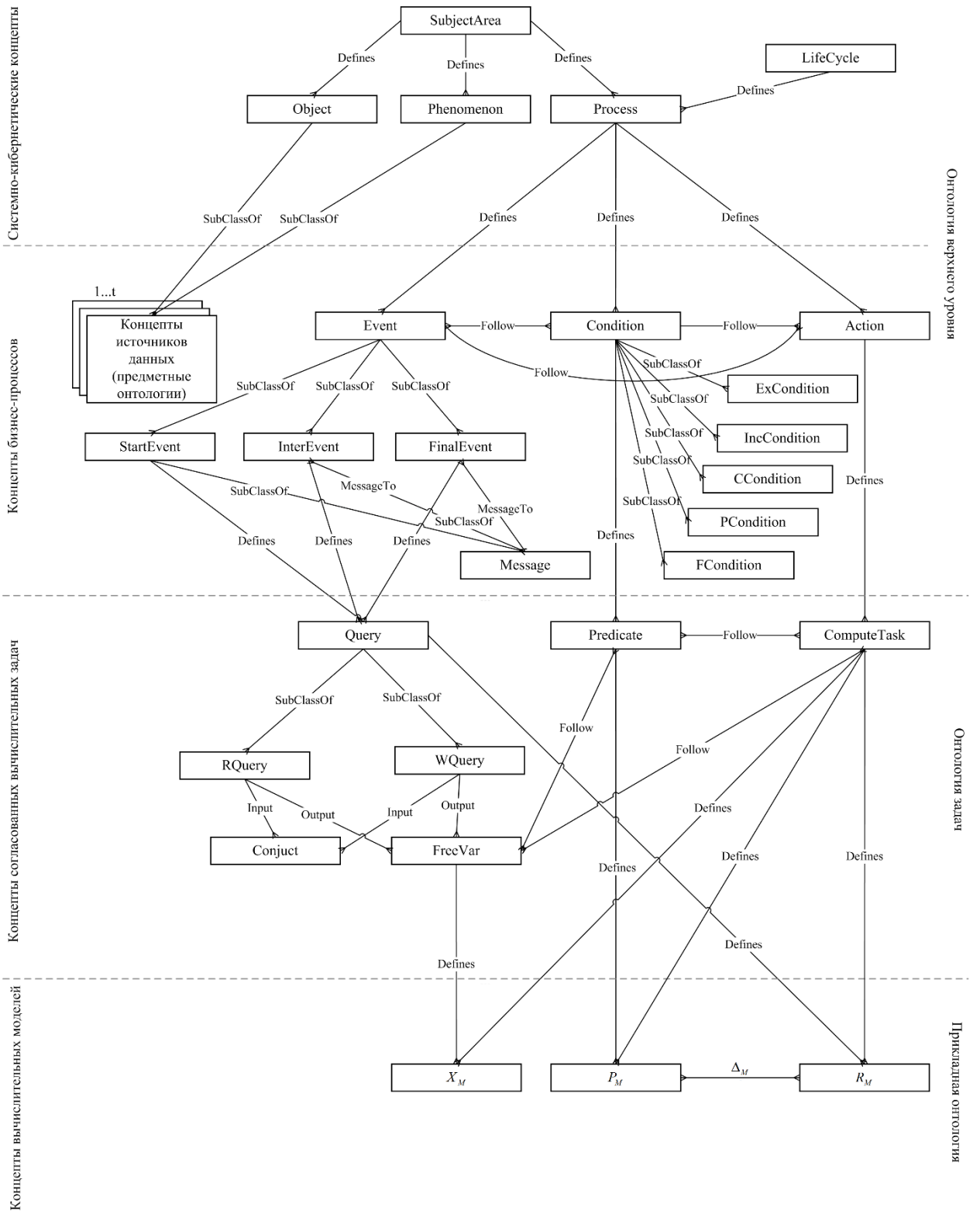


Рис. 2.3. Схема взаимосвязей концептов и ролей онтологической системы информационно-аналитической поддержки ЖЦ СОТО

2.1.5. Результаты исследования задачи онтолого-управляемого моделирования предметной области

В рамках раздела было выявлено, что онтологии могут быть использованы для задания концептуальных моделей необходимого вида, при этом конкретная онтология позволяет определить онтологическую МПЗ, в которой определены правила для построения выполнимых (корректных и полных относительно терминологии) моделей. В качестве аппарата для моделирования при этом необходимо использовать дескрипционную логику, позволяющую явно определять категории (концепты, понятия), в которых осуществляется моделирование, БЗ в виде совокупности утверждений, являющихся интерпретацией, задаваемой для онтологической МПЗ.

В качестве базиса для формирования согласованной системы уточняющих спецификаций бизнес-процессов ЖЦ СОТО, функциональных требований и спецификаций источников данных была разработана онтологическая система информационно-аналитической поддержки ЖЦ СОТО, позволяющая связать воедино разноаспектные модели ПрО, формируемые в составе наборов фактов онтологий. Далее будет дано более подробное пояснение каждой из онтологий и порядка их согласования.

2.2. Онтологическая модель представления знаний о бизнес-процессах информационно-аналитической поддержки жизненного цикла космических средств как сложных объектов

2.2.1. Онтология верхнего уровня для моделирования бизнес-процессов

Метаонтологии (онтологии верхнего уровня) нужны для формального обозначения таких категорий, как сущность, класс, событие, свойство, атрибут, которые позволяют определять в рамках этой онтологии экземпляры этих категорий [176]. Это свойство позволяет говорить о возможности проектирования различного вида концептуальных моделей ПрО на основе этих категорий, формально определяя правила по которым экземпляры этих категорий могут быть связаны. В контексте решаемой задачи в качестве категорий

должны использоваться системно-кибернетические понятия о ПрО и, в частности, категории, связанные с описанием процессов информационно-аналитической поддержки ЖЦ СОТО.

На сегодняшний день разработано множество онтологий верхнего уровня, к числу которых можно отнести Cys, Basic Formal Ontology (BFO), General formal ontology (GFO), Suggested Upper Merged Ontology (SUMO), Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE), онтология Джона Сова (Sowa's ontology) [116, 118, 119]. Однако все они не вполне отвечают задаче, решаемой в работе, в силу их сложности (> 50 концептов) для эксперта ПрО и отсутствия ориентированности на проектирование концептуальных моделей ПрО в виде соответствующих спецификаций процессов анализа.

В ряде работ [6, 43, 162, 169, 193] отмечается необходимость формального онтологического описания процессов (в том числе на предприятиях) для таких задач, как реализация процессного подхода к управлению и мониторингу состояния предприятий, конструктивизация слабо формализованных организационных процессов в виде соответствующих технологических процессов с четко заданной последовательностью действий. В работе [43] отмечается, что процессы необходимо описывать как системы переходов, которые работают с понятиями состояния и перехода. В частности, для онтологий планирования конкретное время и место событий и действий нужно уметь выражать в терминах состояний. Однако также отмечается, что на сегодняшний день отсутствуют онтологии, имеющие формальную семантику для логики поведения процессов, что усложняет задачу. В [193] отмечается, что ЭВМ может быть использована для формирования плана и отслеживания его выполнения. В [6] отмечается, что современное информационное пространство начинает стремительно преумножаться так называемыми процессуальными знаниями. В то же время деятельность большинства современных специалистов в области онтологического инжиниринга знаний нацелено на иерархизацию объектов, отражающих исключительно сущностный аспект материального мира. В

[169] логические системы ситуационного исчисления используются для описания процессов непосредственно этапа функционирования с квантификацией ситуаций по времени, задаются аксиомы предусловий и постусловий наступления событий, но при этом возникают сложности, связанные с так называемой «проблемой окружения» и продолжительностью действий, когда наступившее действие влечет за собой ряд последствий для окружения, не обозримых в текущий момент для системы вывода. В [169] отмечается, что задачи бизнес-процессов в определенном контексте относятся и к планированию, т.е. процессу выработки последовательности действий, позволяющих достичь цели. Например, язык STRIPS использовался как язык планирования, описывая последовательность смен состояний по определенным предусловиям, ассоциированным с действиями. При этом процесс прохождения плана направлен от начального состояния к конечной цели (конечному состоянию). Выполнение плана характеризуется поиском в пространстве состояний на каждом шаге планирования. В работе [176] рассматривается применение онтологий в теории интересубъективного ситуационного управления предприятиями в РВ, предполагающей наличие множества акторов как программных агентов, описываемых онтологиями и взаимодействующих посредством них. «В этих целях становится необходимой онтологическая модель деятельности актора, которая должна ... реагировать на события, находить решения путем перепланирования». Ситуация (сцена) описывает экземпляры понятий и отношений в заданный момент времени (как набор фактов), как мгновенная «фотография» состояния на каждом конкретном предприятии в заданный момент времени. Модель предприятия характеризует его устойчивые аспекты, выражаемые в соответствующих источниках данных. При этом под ситуацией можно понимать информацию, относящуюся к некоторому конечному (реальному или воображаемому) фрагменту реальной или воображаемой действительности, связанной с некоторой конечной частью пространства и конечным промежутком времени, причем этот фрагмент содержит конечное множество объектов, между которыми существует конечная совокупность отношений [82].

Все указанные аспекты характеризуют необходимость формализации процессов в виде некоторых моделей, инвариантных реальным процессам, происходящим в СОТО, причем автоматизация этих процессов предполагает, что ПО должно функционировать в соответствии с этими моделями, что указывает на потоковый характер, параллелизм и асинхронность процессов обработки данных с прохождением этапов решения аналитических задач (см. раздел 1.1, 1.4.1) [132]. При этом необходимо определять, как процессы ЖЦ взаимосвязаны между собой, а модели должны быть достаточно просты, чтобы аналитик мог специфицировать их в предпочтительных для него терминах. Существующие нотации и модели либо не имеют соответствующих выразительных возможностей, либо недостаточно формализованы (см. раздел 1.4.2). В связи с этим, для описания моделей процессов в работе выбран аппарат дескрипционных логик, как наиболее удачный с точки зрения формализации концептуализации процессов (см. раздел 2.1).

Среди существующих онтологий верхнего уровня, ориентированных на описание процессов широкое распространение получила БВВ-онтология, определяющая процессы через концепты события, свойства, трансформации, состояния и др. [198, 199, 200, 234, 235, 264, 265]. Эта онтология наиболее приближена к представлению потоков работ. Однако среди двух известных подходов описания процессов – через смену состояний и через последовательности событий, для процессов информационно-аналитической поддержки ЖЦ СОТО более характерны причинно-следственные связи между событиями, как более высокоуровневое представление о ПрО, как контрольные точки выполнения этапов работ и как выводы о состоянии СОТО (процесс оценивания состояния) [158, 209]. В результате была разработана специальная онтология описания процессов на основе неформализованной нотации BPMN, как одной из известных среди бизнес-аналитиков и явно задающей потоки работ и связи между ними, поскольку ориентация БВВ-онтологии на смену состояний приводит к отсутствию её онтологической чистоты относительно нотации BPMN [234].

2.2.2. Концептуализация бизнес-процессов информационно-аналитической поддержки жизненного цикла СТО

В рамках онтологии $O^{meta} = \langle CN^{meta}, RN^{meta}, \mathcal{I}^{meta} \rangle$ определён следующий набор концептов, наследуемых от \top , т.е. базовых атомарных концептов в соответствии с (1.1), (1.2), (1.3) и рис. 2.3:

$$\{SubjectArea, Object, Phenomenon, Process, LifeCycle\} \subset CN^{meta}, \quad (2.7)$$

где

- $SubjectArea|_{\Delta}^{SubjectArea}$ – конечное непустое множество сущностей типов изделий (или группы изделий) в ПрО, информационную поддержку которых необходимо осуществлять;
- $Object|_{\Delta}^{Object}$ – конечное непустое множество сущностей, характеризующих объекты в ПрО;
- $Phenomenon|_{\Delta}^{Phenomenon}$ – конечное непустое множество сущностей, характеризующих явления в ПрО;
- Остальные элементы см. в таблице 2.1.


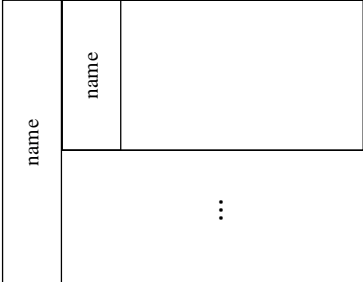

Множество ролей RN^{meta} , задающих отношения на концептах CN^{meta} рассмотрены в таблице 2.1:

$$RN^{meta} = \{Follow, Follow^-, MessageTo\} \quad (2.8)$$

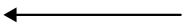
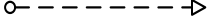

Индивиды концептов $Object$ и $Phenomenon$ задают элементы словаря ПрО и в последующем будут необходимы для предметных онтологий источников данных (см. приложение 3).

В таблице 2.1 представлено множество всех концептов CN^{meta} и ролей RN^{meta} в соотнесении с элементами графической нотации BPMN из [226], кратким пояснением и областями определения концептов. При описании использовались определения из [5].

Таблица 2.1. Связь элементов нотации BPMN и концептов и ролей онтологии бизнес-процессов информационно-аналитической поддержки ЖЦ СОТО

Элементы нотации бизнес-процессов BPMN	Концепты и роли онтологии	Область определения, содержательное пояснение
1. Пул 	<i>LifeCycle</i>	$\Delta^{LifeCycle}$ Конечное непустое множество сущностей, характеризующих ЖЦ изделий. В рассматриваемом контексте состоит из единственного элемента.
2. Дорожка 	<i>Process</i>	$\Delta^{Process}$ Конечное непустое множество сущностей, характеризующих процессы информационно-аналитической поддержки ЖЦ СОТО.
3. Событие	<i>Event</i>	Δ^{Event} Конечное непустое множество сущностей, определяющих всевозможные типы событий, характерные для ПрО и содержательно выражаемые в виде их информационного описания (тип документа, тип аналитического вывода о состоянии, тип характеристики или их совокупности и т.д.).
3.1. Начальное событие 	<i>StartEvent</i>	$\Delta^{StartEvent}$ Конечное непустое множество сущностей, определяющих начальные (исходные) события, характеризующие появление новых фактов из ПрО.
3.2. Промежуточное событие 	<i>InterEvent</i>	$\Delta^{InterEvent}$ Конечное множество сущностей, задающих промежуточные (определяемые) события, содержательно характеризующие происходящие в ПрО явления как результат оценивания фактов предыдущих этапов процесса.
3.3. Конечное событие 	<i>FinalEvent</i>	$\Delta^{FinalEvent}$ Конечное непустое множество сущностей, определяющих конечные (целевые) события, являющиеся целями процесса информационно-аналитической поддержки и явно указывающие на целевые характеристики СОТО, по которым оценивается ТС СОТО по окончании процесса.
4. Действие (Задача)	<i>Action</i>	Δ^{Action}

		<p>Конечное непустое множество сущностей, определяющих задачи (действия), необходимые для оценивания ТС по фактам и явлениям ПрО, соотносимым с этой задачей. Задаются из предположения, что эти действия можно автоматизировать.</p>
<p>5. Шлюз</p>	<p><i>Condition</i></p>	<p>$\Delta Condition$</p> <p>Конечное множество сущностей, определяющих маршрутизацию процессов в виде условий, задающих альтернативы путей выполнения задач или их параллельное выполнение.</p>
<p>5.1. Эксклюзивный шлюз</p> 	<p><i>ExCondition</i></p>	<p>$\Delta ExCondition$</p> <p>Конечное множество сущностей, определяющих прохождение по одному из двух альтернативных маршрутов процесса («исключающее ИЛИ»).</p>
<p>5.2. Инклюзивный шлюз</p> 	<p><i>IncCondition</i></p>	<p>$\Delta IncCondition$</p> <p>Конечное множество сущностей, определяющих прохождение по одному из двух или обоим маршрутам процесса («ИЛИ»).</p>
<p>5.3. Комплексный шлюз</p> 	<p><i>CCCondition</i></p>	<p>$\Delta CCCondition$</p> <p>Конечное множество сущностей, определяющих прохождение по одному или нескольким маршрутам процесса («сложные условия»).</p>
<p>5.4. Параллельный шлюз</p> 	<p><i>PCondition</i></p>	<p>$\Delta PCondition$</p> <p>Конечное множество сущностей, определяющих условия прохождения по нескольким параллельным маршрутам (параллельное исполнение).</p>
<p>5.5. Условный поток операций</p> 	<p><i>FCondition</i></p>	<p>$\Delta FCondition$</p> <p>Конечное множество сущностей, определяющих условия прохождения по маршруту от одного события к одной задаче.</p>
<p>6. Отношение следования</p> 	<p><i>Follow</i></p>	<p>Роль, являющаяся бинарным отношением связывающим элементы множеств индивидов, и содержательно характеризующая следование элементов друг за другом, что и определяет порядок элементов в модели процесса. Отношение является антирефлексивным, антитранзитивным и антисимметричным. В BPMN называется потоком операций.</p>

7.Отношение предшествования 	$Follow^-$	Роль, являющаяся обратным отношением $Follow$, и содержательно характеризующая предшествование элементов.
8. Интероперабельность 	$MessageTo$	Роль, являющаяся бинарным отношением связывающим индивиды событий и содержательно характеризующая связь двух процессов (интероперабельность) по двум событиям. Отношение является антирефлексивным, антитранзитивным и антисимметричным. В BPMN называется пересылкой сообщения.
9. Сообщение 	$Message$	$\Delta^{Message} = \Delta^{InterEvent} \cup \Delta^{FinalEvent}$ Непосредственно передаваемое сообщение, указывающее, что информация о данном событии должна быть передана для другой задачи.

Для проектирования моделей процессов информационно-аналитической поддержки ЖЦ СОТО (далее для краткости «процесс») для онтологии O^{meta} должно быть задано множество терминологических аксиом \mathcal{T}^{meta} , как интенциональных знаний, аппроксимирующих концептуализацию ПрО. Использование этих аксиом позволит в \mathcal{A}^{meta} эксперту-аналитику задавать выполнимые относительно них утверждения, совокупность которых и будет определять модель бизнес-процессов ЖЦ LC для каждого процесса $pr_v | v = 1, \dots, p$ в нем. Тогда в \mathcal{T}^{meta} могут быть определены следующие терминологические аксиомы.

Прежде всего определим специальную роль $Defines \in RN^{meta}$, содержательно интерпретируемую как «определяет что-либо». Тогда выражение (1.1) может быть переопределено в формальных конструкциях дескрипционной логики (см. раздел 2.1.3) следующими аксиомами:

$$\begin{aligned}
 Object &\sqsubseteq \exists Defines. SubjectArea, \\
 Phenomenon &\sqsubseteq \exists Defines. SubjectArea, \\
 Process &\sqsubseteq \exists Defines. SubjectArea,
 \end{aligned}
 \tag{2.9}$$

утверждающими, что Про определяется совокупностью объектов, явлений и процессов информационно-аналитической поддержки ЖЦ СОТО.

В соответствии с определением ЖЦ (см. определение 1.11) может быть введена аксиома:

$$Process \sqsubseteq \leq 1 \text{ Defines. } LifeCycle, \quad (2.10)$$

указывающая, что процесс определяет не более одного (качественная кардинальность роли) индивида *LifeCycle* ЖЦ.

Согласно (1.2) могут быть определены следующие аксиомы:

$$\begin{aligned} Event &\sqsubseteq \leq 1 \text{ Defines. } Process, \\ Condition &\sqsubseteq \leq 1 \text{ Defines. } Process, \\ Action &\sqsubseteq \leq 1 \text{ Defines. } Process, \end{aligned} \quad (2.11)$$

указывающие, что процесс определяется посредством событий, условий и действий, где $\Delta^{Event} \cap \Delta^{Condition} \cap \Delta^{Action} = \emptyset$, причем:

$$\begin{aligned} StartEvent &\sqsubseteq Event, \\ InterEvent &\sqsubseteq Event, \\ FinalEvent &\sqsubseteq Event, \end{aligned} \quad (2.12)$$

что говорит о том, что $(\Delta^{StartEvent} \cup \Delta^{InterEvent} \cup \Delta^{FinalEvent}) \subseteq \Delta^{Event}$.

Аналогичным образом определены все виды условий, заданные в таблице 2.1.

Непосредственно сами концепты событий действий и условий будем определять следующим образом.

$$\begin{aligned} Event &\equiv ((StartEvent \sqcap \neg InterEvent \sqcap \neg FinalEvent) \sqcup \\ &(\neg StartEvent \sqcap InterEvent \sqcap \neg FinalEvent) \sqcup \\ &(\neg StartEvent \sqcap \neg InterEvent \sqcap FinalEvent)) \sqcap \\ &\leq 1 \text{ Follow. } Action \sqcap \neg(\exists \text{ Follow. } Condition) \sqcap \\ &\neg(\exists \text{ Follow. } Event), \end{aligned} \quad (2.13)$$

где с помощью «исключающего ИЛИ» обозначено, что событие может быть либо исходным, либо определяемым, либо целевым, что оно может следовать только за одним действием и не может следовать за условием и за самим собой. Дополнительно введем ограничения:

$$\begin{aligned} \forall Follow^-. StartEvent \sqsubseteq \perp, \\ \forall Follow. FinalEvent \sqsubseteq \perp, \end{aligned} \quad (2.14)$$

указывающие, что ничто не может предшествовать исходным событиям и ничто не может следовать за целевым событием.

Действия могут быть определены следующим образом:

$$\begin{aligned} Action \equiv (\exists Follow. Event \sqcup \exists Follow. Condition) \sqcap \\ \neg(\exists Follow. Action) \sqcap \exists Follow^-. Event, \end{aligned} \quad (2.15)$$

что указывает на то, что действие следует за событием или условием и предшествует некоторому событию при условии, что:

$$\forall Follow. Event \sqsubseteq \neg(\exists Follow^-. Event), \quad (2.16)$$

поскольку указанная аксиома позволяет избежать циклических следований.

Различные типы условий, совокупность которых обеспечивает их полную группу в соответствии с нотацией BPMN, могут быть определены следующим образом:

– условный поток операций:

$$\begin{aligned} FCondition \equiv \leq 1 Follow. Event \sqcap \leq \\ 1 Follow^-. Action \sqcap \neg(\exists Follow. Action) \sqcap \\ \neg(\exists Follow. Condition); \end{aligned} \quad (2.17)$$

– параллельный шлюз:

$$\begin{aligned} PCondition \equiv \geq 2 Follow. Event \sqcap \leq \\ 1 Follow^-. Action \sqcap \neg(\exists Follow. Action) \sqcap \\ \neg(\exists Follow. Condition); \end{aligned} \quad (2.18)$$

– эксклюзивный шлюз:

$$\begin{aligned} ExCondition \equiv \geq 2 Follow. Event \sqcap \\ \leq 2 Follow^-. Action \sqcap \geq 2 Follow^-. Action \sqcap \\ \neg(\exists Follow. Action) \sqcap \neg(\exists Follow. Condition); \end{aligned} \quad (2.19)$$

– инклюзивный шлюз:

$$IncCondition \equiv \geq 2 Follow. Event \sqcap \quad (2.20)$$

$$\leq 2 \text{Follow}^- . \text{Action} \sqcap \geq 2 \text{Follow}^- . \text{Action} \sqcap \\ \neg(\exists \text{Follow} . \text{Action}) \sqcap \neg(\exists \text{Follow} . \text{Condition});$$

– комплексный шлюз:

$$\text{CCondition} \equiv \exists \text{Follow} . \text{Event} \sqcap \exists \text{Follow}^- . \text{Action} \sqcap \\ \neg(\exists \text{Follow} . \text{Action}) \sqcap \neg(\exists \text{Follow} . \text{Condition}). \quad (2.21)$$

Утверждение 2.2. Рассмотренные терминологические аксиомы \mathcal{T}^{meta} , совокупность концептов CN^{meta} и ролей RN^{meta} позволяют моделировать в рамках \mathcal{A}^{meta} сеть событий, действий и условий их применения, определяемую как спецификация бизнес-процесса информационно-аналитической поддержки ЖЦ СОТО и представляющую собой трёхдольный ориентированный граф по определению в [106], вершинами которого являются события, действия и условия, а дугам (ребрам) соответствуют отношения следования *Follow*.

Утверждение 2.3. Терминологические аксиомы \mathcal{T}^{meta} , связанные с понятиями непосредственно ЖЦ и процесса позволяют связывать графы спецификаций процессов между собой с помощью задания дополнительных вершин типа *Process*, *LifeCycle* и дополнительной дуги *Defines*, в результате чего с учетом выражений (2.10) и (2.11) граф обращается в дерево, корнем которого является индивид концепта ЖЦ *LifeCycle*.

Замечание 2.1. В рамках исследования приведены наиболее существенные для моделирования терминологические аксиомы \mathcal{T}^{meta} . Для полноты описания необходимы также аксиомы о непересекаемости концептов, принадлежности концептов, аксиом с обратными ролями, которые очевидно вытекают из указанных выше. Кроме того, при задании спецификации считается, что роль Follow^- , концепты *Event* и *Condition* не могут быть использованы при наполнении системы фактов \mathcal{A}^{meta} .

Указанная в таблице 2.1 роль *MessageTo* и концепт *Message* необходимы для моделирования взаимосвязей процессов и будут рассмотрены далее в разделе 2.4.6.

Рассмотрим пример моделирования *бизнес-процесса*, т.е. формирования набора фактов \mathcal{A}^{meta} , связанного с \mathcal{T}^{meta} в БЗ \mathcal{K}^{meta} . Отметим, что в соответствии с (2.3) факт наличия интерпретации для некоторого концепта в виде соответствующего индивида (его интерпретации) будет соответствовать наличию элемента нотации и его наименованию в графическом представлении модели. Факт, что некоторый элемент (находится в отношении) следует за другим в соответствии с (2.4) и нотацией будем изображать, как показано на рис. 2.4. Факт наличия отношения *Defines* для индивидов событий, действий и условий с процессом и ЖЦ будем изображать, как показано на рис. 2.5.

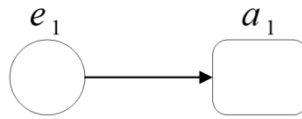


Рис. 2.4. Пример отношения следования между событием и действием

Пример 2.1. Пусть области определения концептов онтологии O^{meta} заданы следующими множествами: $\Delta^{SubjectArea} = \{sa_1\}$, $\Delta^{LyfeCycle} = \{lc_1\}$, $\Delta^{Process} = \{process_1\}$, $\Delta^{StartEvent} = \{e_1, e_2\}$, $\Delta^{InterEvent} = \{e_3, e_4\}$, $\Delta^{FinalEvent} = \{e_5\}$, $\Delta^{PCondition} = \{c_1\}$, $\Delta^{FCondition} = \{c_2\}$, $\Delta^{Action} = \{a_1, a_2, a_3\}$.

А в \mathcal{A}^{meta} задан следующий набор утверждений:

sa_1 : <i>SubjectArea</i>	$process_1$ <i>Defines</i> sa_1	a_3 <i>Defines</i> $process_1$
lc_1 : <i>LyfeCycle</i>	$process_1$ <i>Defines</i> lc_1	c_1 <i>Follow</i> e_1
$process_1$: <i>Process</i>	e_1 <i>Defines</i> $process_1$	c_1 <i>Follow</i> e_2
e_1 : <i>StartEvent</i>	e_2 <i>Defines</i> $process_1$	a_1 <i>Follow</i> e_1
e_2 : <i>StartEvent</i>	e_3 <i>Defines</i> $process_1$	a_2 <i>Follow</i> c_1
e_3 : <i>InterEvent</i>	e_4 <i>Defines</i> $process_1$	e_3 <i>Follow</i> a_1
e_4 : <i>InterEvent</i>	e_5 <i>Defines</i> $process_1$	e_4 <i>Follow</i> a_2
e_5 : <i>FinalEvent</i>	c_1 <i>Defines</i> $process_1$	c_2 <i>Follow</i> e_4
c_1 : <i>PCondition</i>	c_2 <i>Defines</i> $process_1$	a_3 <i>Follow</i> e_3
c_2 : <i>FCondition</i>	a_1 <i>Defines</i> $process_1$	a_3 <i>Follow</i> c_2
a_1 : <i>Action</i>	a_2 <i>Defines</i> $process_1$	e_5 <i>Follow</i> a_3
a_2 : <i>Action</i>		
a_3 : <i>Action</i>		

Тогда указанный набор утверждений \mathcal{A}^{meta} позволяет задать спецификацию бизнес-процесса информационно-аналитической поддержки ЖЦ СОТО следующего вида (см. рис. 2.5):

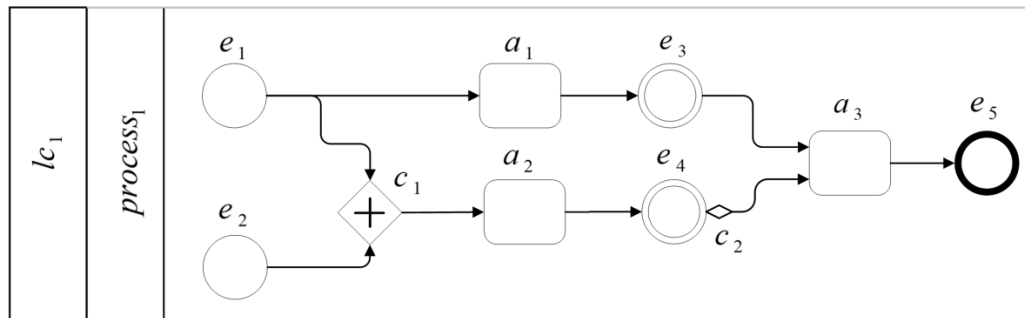


Рис. 2.5. Пример бизнес-процесса информационно-аналитической поддержки ЖЦ СОТО

2.2.3. Результаты исследования задачи концептуализации бизнес-процессов информационно-аналитической поддержки

Разработана онтологическая МПЗ бизнес-процессов информационно-аналитической поддержки ЖЦ СОТО, позволяющая задавать формальные спецификации с графической нотацией BPMN совокупности событийно-ориентированных процессов, связанных ЖЦ СОТО, в терминах событий, действий и условий их применения в ориентации на извлечение знаний о ПрО как требований, предъявляемых к СИАП ЖЦ СОТО с одной стороны, и как структурное описание реальных организационных и технологических процессов – с другой.

2.3. Онтологическая модель представления знаний о согласовании вычислительных задач информационно-аналитической поддержки жизненного цикла космических средств как сложных объектов

2.3.1. Онтология задач для специфицирования структурно-функциональных моделей согласованных вычислительных задач

В работе [73] делается замечание, что «представить деятельность, подлежащую автоматизации в виде совокупности задач – так же естественно, как вслед за этим выбрать те из них, которые изучены настолько, что имеет смысл

передать их решение автомату». При этом программирование может уступить место процессу явного специфицирования порядка решений этих задач, уточнения их входных и выходных параметров, иначе говоря, постановке задачи, которую система должна преобразовать в программу. При этом можно абстрагироваться непосредственно от моделирования решения вычислительных задач, при этом их модели заменяются обобщенным представлением или так называемым частичным отношением, при условии что вычислимые отношения, характеризующие подзадачи являются разрешимыми [193]. Таким образом, достигается возможность разделить процесс специфицирования функциональных требований и непосредственно моделирования самих функций. Такой подход позволит специалистам ПрО охарактеризовать постановку задач без уточнения их решения в терминах параметров и операторов и без необходимости в алгоритмическом мышлении, несвойственным экспертам-непрограммистам. В то же время специалистам, моделирующим решения задач «невозможно будет увернуться от спецификаций» [73].

Для этой задачи могут быть использованы онтологии, называемые в рассматриваемом контексте онтологиями задач. Так, например, в [118, 119] отмечается, что «в перспективе предполагается строить онтологии задач, содержащие методы и их решения, представленные в виде базы продукционных правил». Для рассматриваемой ПрО такая онтология должна учитывать следующие аспекты:

- она должна уточнять события, действия и шлюзы бизнес-процессов в виде соответствующих онтологических запросов, аналитических вычислительных задач и условий их применимости;
- модели бизнес-процессов информационно-аналитической поддержки ЖЦ СОТО задают поведенческие аспекты создаваемой СИАП ЖЦ СОТО. В связи с этим, модель согласованных вычислительных задач (далее, *СВЗ-модель*) должна сохранять структуру, определенную для моделей бизнес-процессов, при этом реализуя переход от описания процессов в ПрО к описанию вычислительных аспектов создаваемой программы;

- онтология должна позволять извлекать экспертные знания о порядке интеграции данных из информационных ресурсов засчет моделирования онтологических запросов в терминах ПрО, обработки их результатов в виде связей с вычислительными задачами, записи результатов расчетов вычислительных задач с использованием графической нотации;
- онтология должна позволять абстрагироваться от непосредственно моделирования решений аналитических вычислительных задач, обобщая их в виде частичных отношений вычислимости;
- концептуализация СВЗ-моделей должна выступать как описание информационных потоков специального вида, позволяя таким образом явно наблюдать входы и выходы моделей аналитических вычислительных задач с одной стороны, и задавать ограничения на ход вычислений создаваемых в последующем по ним программ – с другой, т.е. элементы СВЗ-модели должны задавать спецификацию отношений вычислимости между параметрами и операторами вычислительных моделей [132, 193];
- должна соблюдаться направленность СВЗ-модели, чтобы по спецификации был организован вычислительный процесс, предполагающий исполнение запросов на выборку данных, их обработку и анализ, запись результатов в указанном порядке в соответствии с архитектурой, описанной в приложении 3.3.3;
- каждая СВЗ-модель должна уточнять единственный бизнес-процесс из совокупности бизнес-процессов информационно-аналитической поддержки ЖЦ СОТО.




С учетом описанных требований, онтология задач информационно-аналитической поддержки ЖЦ СОТО может быть задана так, как это указано ниже.



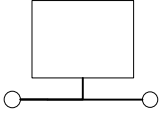

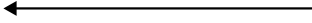

2.3.2. Концептуализация согласования вычислительных задач информационно-аналитической поддержки жизненного цикла сложных объектов

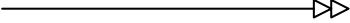
Для моделирования СВЗ-моделей в онтологии $O^t = \langle CN^t, RN^t, \mathcal{I}^t \rangle$ может быть использован тот же диалект дескрипционной логики, который был определен в разделе 2.1.3.

Пусть в O^t заданы множества CN^t и RN^t так, как это показано в таблице 2.2. В этой же таблице изображены элементы нотации концептуальных моделей, являющейся модификацией нотации IDEF5, используемой для схематичного изображения онтологий [236].

Таблица 2.2. Описание элементов нотации моделей согласованных вычислительных задач

Элементы нотации СВЗ	Концепты и роли онтологии	Область определения, содержательное пояснение
1. Запрос	<i>Query</i>	Δ^{Query} Конечное непустое множество экземпляров онтологических запросов, уникальные именованные которых позволяют задать множество операторов исполнения запросов.
1.1. Запрос выборки 	<i>RQuery</i>	Δ^{RQuery} Конечное непустое множество экземпляров онтологических запросов выборки, принадлежащее Δ^{Query} .
1.2. Запрос на запись 	<i>WQuery</i>	Δ^{WQuery} Конечное непустое множество экземпляров онтологических запросов на запись, принадлежащее Δ^{Query} .
2. Конъюнкт запроса 	<i>Conjunct</i>	$\Delta^{Conjunct}$ Конечное непустое множество конъюнктов запросов из определения П. 3.1, представляющиеся триплетами связанных Y или свободных переменных X (со знаком «?» в соответствии с синтаксисом SPARQL запросов), значений $c = val(D)$ или индивидов из онтологий $O_h^d h = 1, \dots, t$, определённых в рамках диалекта DL-Lite _A , и считающиеся тройками литералов для $\Delta^{Conjunct}$. Для каждого элемента

		тройки определен его уникальный идентификатор $URI_h i = 1, \dots, t$. В более общем случае конъюнкт может быть подзапросом запроса <i>Query</i> .
3. Свободные переменные запроса 	<i>FreeVar</i>	$\Delta^{FreeVar}$ Конечное непустое множество переменных X запросов из определения П. 3.1. Причем $X = \{X^+, X^-\}$, где X^+ – свободные переменные запросов выборки, X^- – входные для запросов записи, необходимые для подстановки их значений в конъюнкты запросов.
4. Предикат (условие применимости) 	<i>Predicate</i>	$\Delta^{Predicate}$ Конечное множество описаний предикатных функций (операторов) P , состоящих из констант \mathbb{C} и переменных X , таких что $P_i(\mathbb{C}, X) \rightarrow \{True, False\}$, задающих условия применимости для начала решения вычислительных задач.
5. Вычислительная задача 	<i>ComputeTask</i>	$\Delta^{ComputeTask}$ Конечное непустое множество идентификаторов (именований) вычислительных задач $Ct^T = \{ct_k^T k = 1, \dots, m\}$, где T – вычислительная задача.
6. Отношение следования 	<i>Follow</i>	Роль, являющаяся бинарным отношением, связывающим элементы множеств индивидов, и содержательно характеризующая следование элементов друг за другом. Отношение является антирефлексивным, антитранзитивным и антисимметричным.
7. Отношение предшествования 	<i>Follow⁻</i>	Роль, являющаяся обратным отношением <i>Follow</i> , и содержательно характеризующая предшествование элементов.
8. Отношение по входу 	<i>Input</i>	Роль, являющаяся бинарным отношением, характеризующим связь конъюнктов запросов, переменных, входных для запросов записи, и самих экземпляров запросов. Отношение является антирефлексивным, антитранзитивным и антисимметричным.

<p>9. Отношение по выходу</p> 	<p><i>Output</i></p>	<p>Роль, являющаяся бинарным отношением, характеризующим связь свободных переменных запросов и самих экземпляров запросов. Отношение является антирефлексивным, антитранзитивным и антисимметричным.</p>
---	----------------------	--

По аналогии с моделями бизнес-процессов (см. раздел 2.2.2) для проектирования СВЗ-моделей $sf_v | v = 1, \dots, p$ для онтологии O^t должно быть задано множество терминологических аксиом \mathcal{T}^t , как интенциональных знаний, аппроксимирующих концептуализацию согласования вычислительных задач. Использование этих аксиом позволит в \mathcal{A}^t эксперту-аналитику задавать выполнимые относительно них утверждения, совокупность которых и будет определять СВЗ-модель для каждого бизнес-процесса $pr_v | v = 1, \dots, p$ модели бизнес-процессов информационно-аналитической поддержки ЖЦ СОТО.

Утверждение 2.4. Для согласования моделей sf_v с моделями бизнес-процессов pr_v фактически необходимо, чтобы в рамках O^t набор терминологических аксиом состоял из двух подмножеств $\mathcal{T}^t = \mathcal{T}^{t,sf} \cup \mathcal{T}^{t,def}$, где $\mathcal{T}^{t,sf}$ – набор аксиом, определяющий правила проектирования СВЗ-моделей, а $\mathcal{T}^{t,def}$ – набор аксиом, определяющий отношения концептов O^t и O^{meta} , задающий отображения между ними. При этом, тогда набор фактов $\mathcal{A}^t = \bigcup_{v=1}^p (\mathcal{A}_v^{t,sf} \cup \mathcal{A}_v^{t,def})$ для каждой СВЗ-модели, а также $CN^{meta} \subset CN^t$, $RN^{meta} \subset RN^t$.

Утверждение 2.5. Совокупность терминологических аксиом и наборов фактов онтологий O^t и O^{meta} определяет некоторую расширенную БЗ $\mathcal{K}^{meta,t} = (\mathcal{T}^{meta} \cup \mathcal{T}^t) \cup (\mathcal{A}^{meta} \cup \mathcal{A}^t)$, позволяющую по их совокупности задавать такие модели sf_v , которые могут быть согласованы с pr_v .

Замечание 2.2. Задание в рамках СВЗ-моделей конъюнктов запросов (области определения концепта *Conjunct*), элементы которых определены в рамках O_h^d источников данных, позволяет говорить об отображении из каждой O_h^d

в O^t , в результате чего формируется операциональная модель $M^O = \{Sk, Sf\}$, определяемая посредством \mathcal{A}^t и \mathcal{A}^d .

Таким образом, онтология O^t является расширением онтологии O^{meta} . Тогда в \mathcal{T}^t могут быть определены терминологические аксиомы, указанные ниже. Пусть в дополнение к роли *Defines* определена обратная ей роль *IsDefinedBy* $\in RN^t$, содержательно интерпретируемая как «определен чем-либо». Тогда концепты событий модели бизнес-процессов могут быть определены в СВЗ-модели следующим образом:

$$StartEvent \sqsubseteq \leq 1 IsDefinedBy.RQuery, \quad (2.22)$$

$$FinalEvent \sqsubseteq \leq 1 IsDefinedBy.WQuery, \quad (2.23)$$

$$\begin{aligned} InterEvent \sqsubseteq & (\leq 1 IsDefinedBy.WQuery \sqcup \\ & \leq 1 IsDefinedBy.(\exists Follow.(\exists Output.WQuery) \sqcap \\ & \neg(\exists Defines.Action) \sqcap \neg(\exists Defines.Condition))) \sqcap \\ & \leq 2 IsDefinedBy.Query, \end{aligned} \quad (2.24)$$

что означает, что любое исходное событие определено единственным запросом на чтение, любое целевое событие определено единственным запросом на запись. Необходимо более подробное пояснение для промежуточного события: его необходимо определять либо единственным запросом на запись (в таком случае необходимо проверять корректность связанных фактов \mathcal{A}^t), либо последовательностью двух запросов на запись и чтение по следующим причинам:

- промежуточное событие в модели бизнес-процесса всегда следует за действием и характеризует явления в ПрО как результаты оценивания предыдущих этапов процесса. Эти результаты должны быть записаны в хранилище электронного паспорта в качестве результатов оценивания ТС СОТО. В связи с этим, промежуточное событие определяется запросом на запись. С таким запросом всегда связан единственный параметр, указывающий на

результат исполнения запроса $\{True, False\}$, который можно интерпретировать как сигнал об окончании выполнения задачи;

- для обеспечения вычислительного процесса для последующих задач в соответствии со схемой бизнес-процесса далее может следовать запрос на чтение, причем для обоих запросов может быть определен общий набор конъюнктов, что, очевидно, обеспечит в процессе функционирования в результатах выборки запроса на чтения в числе прочих данные, записанные запросом на запись.

Для онтологических запросов могут быть определены следующие выражения:

$$RQuery \sqsubseteq Query, \quad (2.25)$$

$$WQuery \sqsubseteq Query, \quad (2.26)$$

$$Query \equiv (RQuery \sqcap \neg WQuery) \sqcup (\neg RQuery \sqcap WQuery), \quad (2.27)$$

обозначающие, что экземпляр запроса может быть одного из двух типов – на чтение и на запись.

Дополнительно должно быть введено ограничение следующего вида:

$$Query \equiv \leq 1 \text{ Defines. Event}, \quad (2.28)$$

позволяющее задать точную кардинальность ролей совместно с выражениями (2.22), (2.23), (2.24), однозначно связывая запросы и события один к одному, кроме случая промежуточных событий (2.24) (по два запроса на одно событие).

Концепт свободной переменной *FreeVar* должен определяться таким образом, чтобы аксиомы указывали на все возможные варианты связей переменной с запросами:

$$\begin{aligned} FreeVar \equiv & (\leq 1 \text{ Output. RQuery} \sqcap \neg(\exists \text{ Input. WQuery}) \sqcap \\ & \neg(\exists \text{ Output. WQuery}) \sqcap \exists \text{ Follow}^-. \text{ ComputeTask}) \sqcup \\ & (\neg(\exists \text{ Output. RQuery}) \sqcap \exists \text{ Input. WQuery} \sqcap \\ & \neg(\exists \text{ Output. WQuery}) \sqcap \leq 1 \text{ Follow. ComputeTask}) \sqcup \\ & (\neg(\exists \text{ Output. RQuery}) \sqcap \neg(\exists \text{ Input. WQuery}) \sqcap \end{aligned} \quad (2.29)$$

$$\leq 1 \text{ Output.WQuery} \sqcap (\exists \text{Follow}^-. \text{ComputeTask} \sqcup \exists \text{Follow}^-. \text{RQuery})).$$

Выражение задано в дизъюнктивной нормальной форме, указывая на варианты связей, т.е. трёх случаев, когда переменная является:

- выходной для запроса выборки (связана с записью результата исполнения запроса) и предшествующей вычислительной задаче;
- выходной для запроса на запись (не более одной), как указывалось выше, и предшествующей вычислительной задаче или запросу выборки;
- входной для запроса на запись и следующей за вычислительной задачей. Связь задана с целью дальнейшей возможности подстановки значений переменной в конъюнкты запроса для получения триплетов литералов (конкретных значений).

При этом должно быть задано ограничение:

$$\forall \text{Follow}^-. \text{ComputeTask} \sqsubseteq \neg(\exists \text{Follow}^-. \text{ComputeTask}) \sqcap \neg(\exists \text{Output.WQuery}), \quad (2.30)$$

позволяющее исключить циклические связи для вычислительных задач и запросов записи.

В свою очередь конъюнкты запросов могут быть определены следующим образом:

$$\text{Conjunct} \sqsubseteq \exists \text{Input}^-. (\exists \text{Defines}^-. \text{Event}), \quad (2.31)$$

указывая, что конъюнкт является входным для одного из запросов, связанных с событиями, причем один и тот же конъюнкт может быть общим для нескольких запросов.

Описания предикатов (условий применимости) задаются следующим образом:

$$\text{Predicate} \sqsubseteq \exists \text{Defines}^-. \text{Condition} \sqcap \exists \text{Follow}^-. (\exists \text{Output}^-. (\exists \text{Defines}^-. (\exists \text{Follow}^-. \text{Condition}))), \quad (2.32)$$

указывая, что шлюз бизнес-процесса определен предикатом, который следует за переменной, являющейся выходной для запроса, который определяет предшествующее условию событие. Заметим, что условие может быть определено несколькими предикатами в зависимости от семантики шлюзов из выражений (2.17) – (2.21).

В свою очередь, действие определяется вычислительной задачей:

$$Action \sqsubseteq \leq 1 IsDefinedBy. ComputeTask, \quad (2.33)$$

при этом сама вычислительная задача определяется выражением:

$$\begin{aligned} ComputeTask \equiv \leq 1 Defines. Action \sqcap \\ (\exists Follow. (\exists Output. (\exists Defines. (\exists Follow^-. Action))) \sqcap \\ \neg(\exists Follow^-. Predicate)) \sqcup \\ \exists Follow. ((\exists Output. (\exists Defines. (\exists Follow^-. (\exists Follow^-. Action)))) \quad (2.34) \\ \sqcap \exists Follow^-. (\exists Defines. (\exists Follow^-. Action))) \sqcup \\ \exists Defines. (\exists Follow^-. Action))) \sqcap \\ \exists Follow^-. (\exists Input. (\exists Defines. (\exists Follow. Action))), \end{aligned}$$

означающим, что вычислительная задача:

- определяет действие;
- предшествует переменным, входным для запросов на запись, определяющим события, следующими за действием;
- следует за свободной переменной запроса:
 - не предшествующей никаким предикатам;
 - предшествующей предикату.

Утверждение 2.6. Рассмотренные терминологические аксиомы \mathcal{T}^t , совокупность концептов CN^t и ролей RN^t позволяют моделировать в рамках \mathcal{A}^t сеть запросов на чтение и запись, предикатов и вычислительных задач, определяемую как СВЗ-модель информационно-аналитической поддержки ЖЦ СОТО и представляющую собой пятидольный ориентированный граф по определению в [106], вершинами которого являются запросы на чтение, за-

просы на запись, конъюнкты запросов, свободные переменные запросов и вычислительные задачи, а дугам (ребрам) соответствуют отношения следования *Follow*, отношения, описывающие связи по входу и выходу *Input*, *Output* соответственно.

Необходимо заметить, что в терминологических аксиомах, определяемых для O^t используются такие концепты, где R-последователями являются концепты с использованием ролей. Такая «вложенность» концептов в виде, так называемых, составных концептов (где глубина вложенности называется кванторной сложностью) позволяет определять одни термины через множество связанных других. Такой способ задания семантики используется в \mathcal{T}^t для связывания концептов бизнес-процессов и СВЗ-моделей.

Утверждение 2.7. Терминологические аксиомы \mathcal{T}^t позволяют уточнять спецификации множества бизнес-процессов Pr в виде СВЗ-моделей в \mathcal{A}^t за счет определения бинарных отношений между концептами онтологии бизнес-процессов и концептов онтологии задач. При этом увеличивается глубина дерева, определенного в утверждении 2.3 заданием дополнительных вершин, связанных дугами *Defines*.

Замечание 2.3. В рамках исследования приведены наиболее существенные для моделирования терминологические аксиомы \mathcal{T}^t . Для полноты описания необходимы также аксиомы о непересекаемости концептов, принадлежности концептов, аксиом с обратными ролями, которые очевидно вытекают из указанных выше. Кроме того, при задании спецификации считается, что роль *IsDefinedBy* и концепт *Query* не могут быть использованы при наполнении системы фактов \mathcal{A}^t .

Замечание 2.4. Важным ограничением формируемой СВЗ-модели является факт, что в рамках описания одного запроса не может более одного *URI*, т.е. один запрос может быть обращен только к одному источнику данных в соответствии с определённым в приложении 3.3.

Замечание 2.5. Корректность формируемых в \mathcal{A}^t запросов не рассматривается в рамках \mathcal{T}^t и остается в ответственности специалиста, проектирующего СВЗ-модель, поскольку конъюнкты запросов являются синтаксически неразличимыми триплетами литералов по определению концепта *Conjunct*. Такое допущение делается для того, чтобы визуально упростить формируемые модели и сохранить семантическую «целостность» фактов, определяемых в рамках запросов.

Рассмотрим пример моделирования *sf* согласования задач, т.е. формирования набора фактов \mathcal{A}^t , связанного с \mathcal{T}^t в БЗ $\mathcal{K}^{meta.t}$. В соответствии с (2.3) факт наличия интерпретации для некоторого концепта в виде соответствующего индивида (его интерпретации) будет соответствовать наличию элемента нотации и его наименованию в графическом представлении модели. Факт, что некоторый элемент (находится в отношении) следует за другим в соответствии с (2.4) и нотацией будем изображать, как показано на рис. 2.4. Факт наличия отношения *Defines* и обратного ему не будем изображать графически. Для запросов элементы, связанные по входу, будем соединять стрелкой слева, а по выходу – справа, как это показано на рисунке 2.6 в соответствии с нотацией, заданной в таблице 2.2.

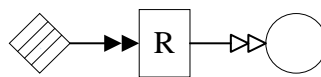


Рис. 2.6. Пример связи по входу и выходу некоторого запроса

Пример 2.2. Пусть задана модель бизнес-процесса pr_1 , определенная в рамках примера 2.1. Идентификатор источника данных будем условно обозначать URI_i . Пусть также области определения концептов онтологии O^t заданы следующими множествами: $\Delta^{RQuery} = \{q_1^R, q_2^R, q_3^R\}$, $\Delta^{WQuery} = \{q_1^W, q_2^W, q_3^W\}$, $\Delta^{FreeVar} = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}\}$, $\Delta^{ComputeTask} = \{ct_1^T, ct_2^T, ct_3^T\}$, $\Delta^{Predicate} = \{p_1, p_2\}$, $\Delta^{Conjunct} = \{? x_1 URI_1 \# R_1 URI_1 / b, URI_1 / b URI_1 \# R_2 ? x_2, ? x_3 ? x_4 URI_2 / d, URI_3 /$

$f \text{ URI}_3\#R_3 ? x_7, ? x_8 \text{ URI}_3\#R_4 \text{ URI}_3/h, ? x_{10} ? x_{11} \text{ URI}_3/l\}$, где b, d, f, h, l – индивиды или значения атрибутов, заданных в онтологиях M^{O^d} , а R_1, R_2, R_3, R_4 – роли или атрибуты, заданные в онтологиях M^{O^d} , $\text{URI}_1, \text{URI}_2, \text{URI}_3$ – идентификаторы трёх источников данных. Заметим, что в примере для всех констант b, d, f, h, l заданы их URI , однако в случае если константа является значением атрибута, уникального идентификатора у неё не будет, поскольку она является литералом с позиции O^d . Пусть в $\mathcal{A}^{t, sf}$ задан следующий набор утверждений:

$q_1^R: RQuery$	$? x_1 \text{ URI}_1\#R_1 \text{ URI}_1/b: Conject$	$x_9 \text{ Output } q_2^W$
$q_2^R: RQuery$	$\text{URI}_1/b \text{ URI}_1\#R_2 ? x_2: Conject$	$x_7 \text{ Output } q_3^R$
$q_3^R: RQuery$	$? x_3 ? x_4 \text{ URI}_2/d: Conject$	$x_{12} \text{ Output } q_3^W$
$q_1^W: WQuery$	$\text{URI}_3/f \text{ URI}_3\#R_3 ? x_7: Conject$	$ct_1^T \text{ Follow } x_1$
$q_2^W: WQuery$	$? x_8 \text{ URI}_3\#R_4 \text{ URI}_3/h: Conject$	$ct_1^T \text{ Follow } x_2$
$q_3^W: WQuery$	$? x_{10} ? x_{11} \text{ URI}_3/l: Conject$	$ct_2^T \text{ Follow } x_3$
$x_1: FreeVar$	$? x_1 \text{ URI}_1\#R_1 \text{ URI}_1/b \text{ Input } q_1^R$	$ct_2^T \text{ Follow } x_4$
$x_2: FreeVar$	$\text{URI}_1/b \text{ URI}_1\#R_2 ? x_2 \text{ Input } q_1^R$	$p_1 \text{ Follow } x_2$
$x_3: FreeVar$	$? x_3 ? x_4 \text{ URI}_2/d \text{ Input } q_2^R$	$p_1 \text{ Follow } x_4$
$x_4: FreeVar$	$\text{URI}_3/f \text{ URI}_3\#R_3 ? x_7 \text{ Input } q_1^W$	$ct_2^T \text{ Follow } p_1$
$x_5: FreeVar$	$\text{URI}_3/f \text{ URI}_3\#R_3 ? x_7 \text{ Input } q_3^R$	$x_5 \text{ Follow } ct_1^T$
$x_6: FreeVar$	$? x_8 \text{ URI}_3\#R_4 \text{ URI}_3/h \text{ Input } q_2^W$	$x_8 \text{ Follow } ct_2^T$
$x_7: FreeVar$	$? x_{10} ? x_{11} \text{ URI}_3/l \text{ Input } q_3^W$	$q_3^R \text{ Follow } x_6$
$x_8: FreeVar$	$x_5 \text{ Input } q_1^W$	$p_2 \text{ Follow } x_9$
$x_9: FreeVar$	$x_8 \text{ Input } q_2^W$	$ct_3^T \text{ Follow } p_2$
$x_{10}: FreeVar$	$x_{10} \text{ Input } q_3^W$	$x_{10} \text{ Follow } ct_3^T$
$x_{11}: FreeVar$	$x_{11} \text{ Input } q_3^W$	$x_{11} \text{ Follow } ct_3^T$
$x_{12}: FreeVar$	$x_1 \text{ Output } q_1^R$	
$p_1: Predicate$	$x_2 \text{ Output } q_1^R$	
$p_2: Predicate$	$x_3 \text{ Output } q_2^R$	
$ct_1^T: ComputeTask$	$x_4 \text{ Output } q_2^R$	
$ct_2^T: ComputeTask$	$x_6 \text{ Output } q_1^W$	
$ct_3^T: ComputeTask$		

А в $\mathcal{A}^{t, def}$ задан набор утверждений в соответствии с \mathcal{A}^{meta} из примера

2.1:

$q_1^R \text{ Defines } e_1$	$q_2^W \text{ Defines } e_4$	$ct_1^T \text{ Defines } a_1$
$q_2^R \text{ Defines } e_2$	$q_3^W \text{ Defines } e_5$	$ct_2^T \text{ Defines } a_2$
$q_3^R \text{ Defines } e_3$	$p_1 \text{ Defines } c_1$	$ct_3^T \text{ Defines } a_3$
$q_1^W \text{ Defines } e_3$	$p_2 \text{ Defines } c_2$	

Тогда указанный набор утверждений \mathcal{A}^t позволяет задать СВЗ-модель sf информационно-аналитической поддержки ЖЦ СОТО следующего вида (см. рис. 2.7):

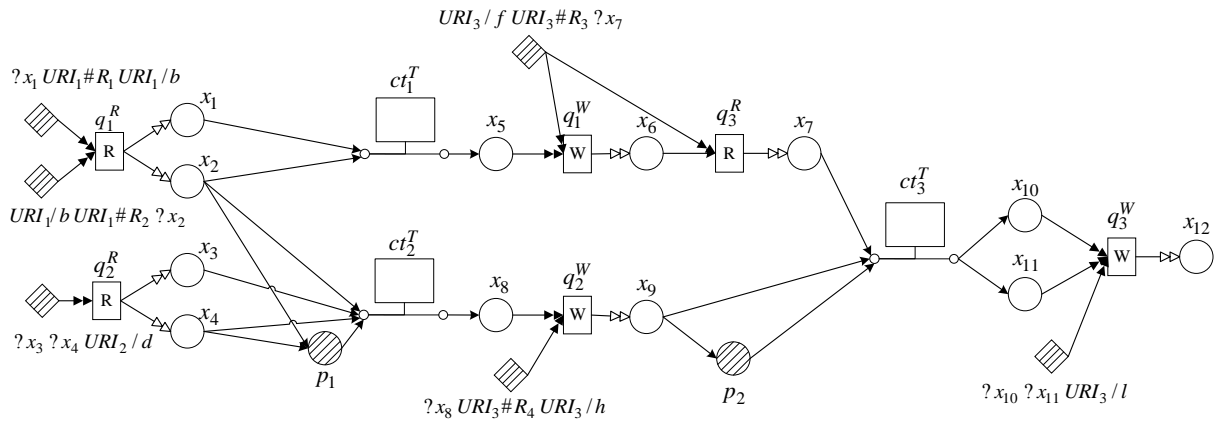


Рис. 2.7. Пример модели согласованных вычислительных задач

2.3.3. Результаты исследования задачи концептуализации согласования вычислительных задач информационно-аналитической поддержки

Разработана онтологическая МПЗ согласования вычислительных задач, позволяющая на формальной основе уточнять спецификации бизнес-процессов информационно-аналитической поддержки ЖЦ СОТО в виде СВЗ-моделей, осуществляя переход от описания ПрО к спецификации ПО СИАП ЖЦ СОТО, сохраняя структуру, введенную на моделях бизнес-процессов, обеспечивая реализацию ETL-принципа и абстрагируясь от непосредственно моделирования решений вычислительных задач, тем самым обеспечивая декомпозицию этапов разработки ПО, разделяя этап специфицирования потоков данных и этап моделирования вычислимых функций.

2.4. Алгоритмы формирования концептуальных моделей и схем программ решения аналитических вычислительных задач и организация вычислений по ним

2.4.1. Моделирование решения аналитических вычислительных задач

С учетом описанной в разделе 2.3 онтологии задач, позволяющей проектировать СВЗ-модели необходимо теперь перейти непосредственно к модели-

рованию функциональных отношений на вводимых спецификациях с последующим синтезом схем их программ. Эта задача предполагает решение двух подзадач:

- моделирование решения аналитических вычислительных задач, заданных в онтологической МПЗ о согласовании задач в ориентации на возможность мониторинга (распознавания) состояния СОТО по совокупности тех явлений в ПрО, типы которых определены в модели бизнес-процессов, и информация о которых поступает в качестве входной в СИАП ЖЦ СОТО;
- задание функциональных отношений на параметрах, описание которых задано в спецификации (СВЗ-модели), в виде соответствующей структурно-функциональной модели в соответствии с модельно-ориентированной парадигмой проектирования ИС (см. приложение 2.3.4).

2.4.2. G-модель как модель представления знаний для решения аналитических вычислительных задач

В соответствии с уже рассмотренными аспектами методологии автоматизированного анализа ТС СОТО (см. раздел 1.2.2) в качестве формального аппарата для моделирования решения аналитических вычислительных задач были выбраны *G-модели* [133, 151, 148, 150]. Среди существенных преимуществ этого аппарата можно назвать следующие.

G-модели ориентированы на процесс извлечения и представления знаний о процессах анализа входной информации в виде гибридной продукционно-фреймовой модели, позволяющей создавать сети параметров, операторов и условий их применения без навыков в программировании.

G-модель как разновидность обобщенной вычислительной модели (см. [193]) объединяет в себе преимущества традиционного программирования (задания абстрактных типов, их интерпретации, вычислительных процедур) и структурно-функционального представления (семантической сети) с хорошо реализуемыми отношениями, позволяя, с одной стороны описывать процесс анализа в ПрО, а с другой – задавать спецификацию ПО;

Организация вычислений в соответствии с G-моделями позволяет в потоковом режиме рекуррентно доопределять значения параметров в ходе СПМ-распознавания, как метода теории распознавания образов (последовательно-параллельных процедур локального распознавания на основе совокупности вводимых вычислимых отношений как вычисляющих алгоритмов, т.е. процессов вычисления оценок), при этом вычисляемые параметры и их структура несут в себе семантику, связанную с явным определением (оцениванием) ТС СОТО (сравнительный анализ методов распознавания образов см. в приложении 2.3.2, графическая интерпретация СПМ-распознавания представлена на рис. 2.8).

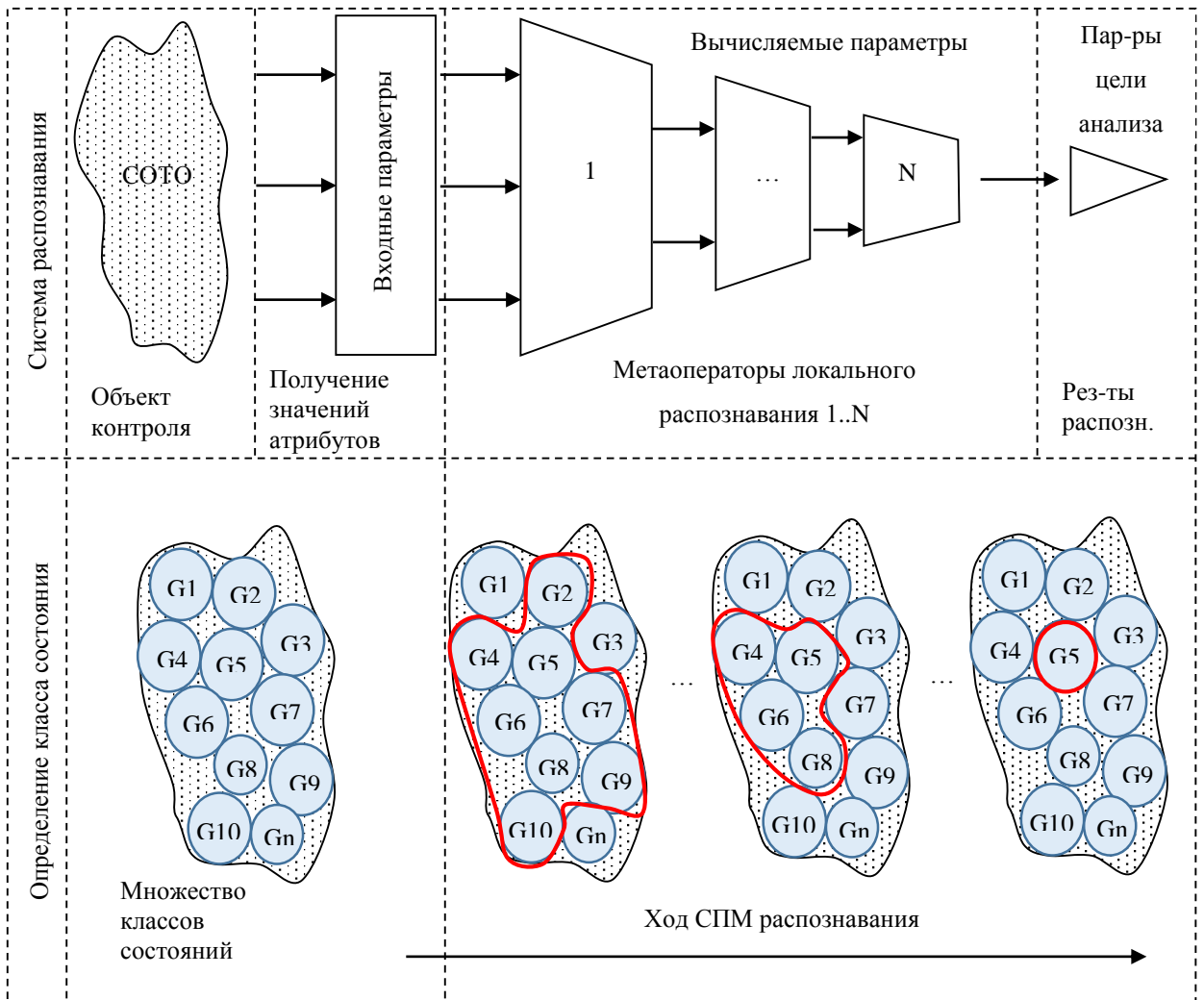


Рис. 2.8. Графическая интерпретация структурно-потоково-многоуровневого распознавания образов

Задание вычислимых отношений в терминах параметров (в сопоставлении с денотатами ПрО) и операторов (как отношений между ними) позволяет явно выразить, с одной стороны, ход решения вычислительной задачи, тем самым отделив моделирование решения задач от моделирования требований (в моделях, представленных выше), а с другой, задать явное представление отношений в ПрО в виде семантической сети. В результате о G-модели можно говорить как о прикладной онтологии O^a в соответствии с определением онтологической системы (см. определение 2.11, раздел 2.1.4). При этом на ней использован другой формальный аппарат, отличный от дескрипционной логики, однако это не уменьшает значимости сделанного замечания, поскольку содержательно G-модели по своим выразительным свойствам отражают именно онтологический аспект моделирования (концептуализации) ПрО.

Суть G-модели в концептуальном программировании заключается в следующем:

- вводится понятие вычислительной модели как фрейма, описывающего вычислительную обстановку;
- знания о сложной ситуации представляются набором признаков, действий, а также условий их применения;
- слоты фрейма заполняются при конкретизации знаний, когда фрейм применяется в связи с некоторой конкретной ситуацией;
- МПЗ представляется в виде семантической сети, дополненной описанием действий и совокупностью правил для выполнения этих действий;
- в узлах сети находятся денотаты предметной области, представленные с помощью измеряемых и вычисляемых параметров;
- отношения между денотатами представляются в форме отображений, ограниченных набором вычислимых функций. В терминах программирования такие функциональные отношения можно назвать операторами;
- сети, объектами которых являются переменные, связанные между собой отношениями, называются вычислительными моделями. В расширенных вычислительных моделях также определены операторы условий и циклов;

- синтез программы вычислений происходит после аксиоматизации МПЗ, заключающейся в формировании для каждого оператора сети некоторого вычислимого отношения.

G-модель является разновидностью ОВМ, определенной в рамках программирования в ограничениях. Остановимся на содержании технологии программирования в ограничениях. Она предполагает:

- описание модели задачи, а не алгоритмов её решения;
- модель описывается в виде неупорядоченной совокупности отношений, которые соответствуют связям между денотатами ПрО;
- отношения называют ограничениями. Могут быть представлены уравнениями, неравенствами, логическими выражениями, символьными операторами и пр.
- постановка задачи конкретизируется введенными в модель ограничениями на допустимые значения параметров.
- параметры задачи не разделены на входные и выходные.
- пользователь самостоятельно решает, какие параметры заданы точно, какие неизвестны совсем, а какие – приблизительно.
- методы программирования в ограничениях обеспечивают автоматическое решение задачи.

Вычислительный процесс в ОВМ имеет потоковый характер: вершины-объекты активизируют вершины-функции. Вычислительный процесс закончится тогда, когда не остается активных функциональных вершин, или когда функция проверки корректности вырабатывает ложь.

Определение 2.12. Вычислительной G-моделью (или просто G-моделью) называется кортеж [132]:

$$M^G = \langle X_M, R_M, P_M, \Delta_M \rangle, \quad (2.35)$$

где

$X = \{x_i | i = 1, \dots, n\}$ - конечное множество переменных (параметров) ТС, характеризующих явления (процессы) из ПрО;

R_M (или в терминах операторов – Φ_M) – конечное множество (вычисли-
мых) отношений на множестве параметров из X ;

P_M (или в терминах операторов – Φ_M^P) – конечное множество предикатов,
предметными переменными в которых являются элементы из X ;

Δ_M – отображение $R_M \rightarrow P_M$ (в терминах операторов $\Phi_M \rightarrow \Phi_M^P$), ставя-
щее в соответствие каждому отношению из R_M элемент из P_M .

G-модель можно представить в виде операторной схемы и просто схемы
(см. рис. 2.9).

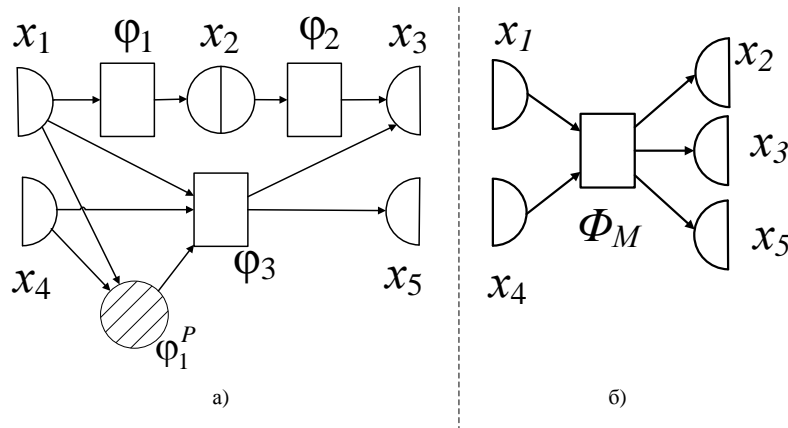


Рис. 2.9. Операторная схема (а) и схема (б) G-модели

Операторная схема (а) несет в себе информацию о структуре модели, по-
казывая структурные взаимосвязи между элементами. Простая схема G-
модели (б) отображает единственный общий оператор, отображая частичное
вычисляемое отношение (при условии, что моделируемая задача разрешима) и
выявляя параметры, которые в конечном счете могут быть вычислены.

2.4.3. Переход от моделей согласованных вычислительных задач к вычислительным моделям

При моделировании реальной системы с помощью G-моделей, образу-
ется полимодельный комплекс, где каждая модель представлена своей опера-
торной схемой. При этом на G-моделях определена операция конкатенации,
обозначаемая « \oplus » и позволяющая «склеивать» элементарные G-модели
между собой, в которых $|R_M| = |\Phi_M| = 1$.

Для любой конкретной G-модели m^G множество X_M разбивается на:

- множество входных параметров X^+ ;
- множество выходных параметров X^- ;
- множество промежуточных параметров X^\sim .

Вычисления на G-моделях предполагают по своему замыслу нахождение значений некоторых целевых параметров (параметров цели анализа) X^g из X^- .

В соответствии с формальным представлением о вычислительной задаче с учетом акцента на нахождение интересующих значений параметров *вычислительная задача* может быть определена следующим образом [132, 193]:

$$T(m^G; X^+ \mapsto X^g). \quad (2.36)$$

В то же время спецификация некоторой вычислительной задачи в СВЗ-модели предполагает наличие триплетов в \mathcal{A}^t , связывающих переменные запросов и вычислительную задачу, что может быть проиллюстрировано на части примера 2.2 и выражениями: $ct_1^T \text{ Follow } x_1$, $ct_1^T \text{ Follow } x_2$, $x_5 \text{ Follow } ct_1^T$ (см. рис. 2.10).

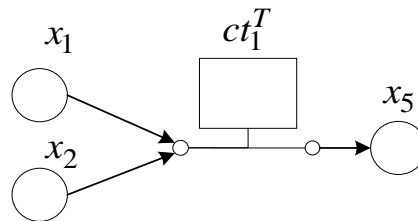


Рис. 2.10. Пример отношения следования индивидов переменных и вычислительной задачи

Такое представление было определено с целью абстрагирования в СВЗ-модели от непосредственно проектирования ВМ, предполагающих описание процедуры анализа информации, поступившей на её вход в результате исполнения запроса.

Замечание 2.6. Совокупность индивидов свободных переменных, определяемых в системе фактов БЗ $\mathcal{K}^{meta,t}$ в соответствии с терминологическими

аксиомами (2.29) и (2.34) задает ограничения на используемые входные и выходные параметры ВМ на основе задания отношения следования *Follow*.

Утверждение 2.8. Пусть имеется некоторое конечное множество индивидов концепта вычислительных задач *ComputeTask*: $Ct^T = \{ct_k^T | k = 1, \dots, m\}$, тогда каждому ct_k^T может быть сопоставлена некоторая G-модель $m_k^{G,T}$ из множества моделей $M^{G,T} = \{m_k^{G,T} | k = 1, \dots, m\}$:

$$ct_k^T \rightarrow m_k^{G,T}, \quad (2.37)$$

причем в соответствии с определением концепта *FreeVar* в таблице 2.2 для конкретной СВЗ-модели если некоторое множество свободных переменных X^+ предшествует (*Follow*⁻) вычислительной задаче ct_k^T , то его элементам могут быть однозначно сопоставлены параметры X^+ , являющиеся входными в $m_k^{G,T}$, и если некоторое множество свободных переменных X^- следует за (*Follow*) вычислительной задачей ct_k^T , то его элементам могут быть однозначно сопоставлены параметры X^g , являющиеся параметрами цели анализа и выходными в $m_k^{G,T}$, так что может быть получено выражение (2.36).

Таким образом, в рамках предлагаемого подхода предполагается разработка G-моделей для каждого элемента – индивида концепта вычислительной задачи в СВЗ-модели. В то же время некоторая СВЗ-модель sf_v может быть полностью отображена в соответствующую G-модель m_v^G , так что для множества процессов $Pr = \{pr_v | v = 1, \dots, p\}$ модели бизнес-процессов информационно-аналитической поддержки ЖЦ СОТО, при условии что каждому pr_v сопоставлена уточняющая спецификация sf_v , будет задано отображение:

$$pr_v \rightarrow m_v^G, \quad (2.38)$$

причем модели $M^{G,T}$ вложены в соответствующие модели M^G в соответствии с описанием в СВЗ-моделях.

Для реализации такого отображения необходима процедура структурно-параметрического синтеза G-модели m_v^G по соответствующей СВЗ-модели sf_v , связанная с синтезом соответствующих параметров по описаниям переменных

запросов, операторов по описаниям запросов и предикатов, отношений на них, а также предполагающая «склеивку» полученных моделей с моделями $M^{G,T}$.

Рассмотрим алгоритм синтеза G-модели по СВЗ-модели в соответствии с описанными выше требованиями. Для этого определим ряд важных фактов:

- будем считать, что для задаваемых в G-модели параметров X определены области их значений $\mu(x_i)$ в соответствии с типами данных, определенными для онтологий O^d , поскольку в соответствии с рассмотренным в приложении 3.3.2 области значений переменных запросов (а в O^t определены только они) должны соответствовать типам данных, определенным в непосредственно хранилищах данных, к СУБД которых осуществляются запросы;
- в множестве операторов Φ_M будем выделять три их типа, так что $\Phi_M = \{\Phi_M^T, \varphi_M^{q^R}, \varphi_M^{q^W}\}$, где Φ_M^T – множество операторов, связанных с моделированием аналитических вычислительных задач, $\varphi_M^{q^R}$ – оператор исполнения запроса выборки, алгоритм которого определен в приложении 3.3.4, $\varphi_M^{q^W}$ – оператор исполнения запроса на запись известного вида [229], предполагающий запись данных в хранилище электронного паспорта;
- множество операторов Φ_M^P будем соотносить с множеством индивидов *Predicate*;
- в множестве параметров X^+ будем выделять три их типа, так что $X^+ = \{X^{q^R}, X^{q^W}, X^{timer}\}$, где X^{q^R} и X^{q^W} – переменные, значения которых являются синтаксическим представлением онтологических запросов, исполняемых $\varphi_M^{q^R}$ и $\varphi_M^{q^W}$ соответственно, причем каждому $\varphi_M^{q^R}$ и $\varphi_M^{q^W}$ соответствуют единственные x^{q^R} и x^{q^W} , а X^{timer} – переменные, значения которые являются входными для СИАП ЖЦ СОТО и задаются некоторым генератором с необходимой частотой (эта частота будет определять время t между двумя последовательными запусками оператора исполнения запроса);
- элементы множества параметров X^{\sim} будем соотносить с переменными СВЗ-модели (индивидами концепта *FreeVar*), за исключением случаев когда за

переменной ничего не следует и она не является входной ни для какого запроса. В этом случае эту переменную будем соотносить с X^g ;

- вычислительные модели $m_k^{G,T}$, связанные с индивидами вычислительных задач в sf_v , будем заменять для упрощения и наглядности единственным оператором φ_k^T из предположения, что $m_k^{G,T}$ разрешима;
- отношению следования *Follow* между индивидами предикатов *Predicate* и вычислительных задач *ComputeTask* может быть сопоставлено отношение δ^G между условиями применимости и операторами.

Тогда с учетом указанных фактов схема алгоритма формирования G-модели m_v^G по некоторой СВЗ-модели sf_v в составе \mathcal{A}^t представлен на рисунках 2.11 и 2.12.

Утверждение 2.9. С учетом утверждения 2.3, очевидно, что каждой модели процесса pr_v из модели бизнес-процессов может быть сопоставлена своя модель sf_v , представленная в виде $\mathcal{A}_v^{t,sf}$, поскольку имеет место $\mathcal{K}^{meta,t} = (\mathcal{T}^{meta} \cup \mathcal{T}^t) \cup (\mathcal{A}^{meta} \cup \mathcal{A}^t)$ и определены связи между концептами моделей pr_v и sf_v в виде множества терминологических аксиом. В связи с этим, по модели бизнес-процессов может сформировано множество G-моделей $M^G | v = 1, \dots, p$ по количеству бизнес-процессов в *Pr*.

С учетом этого, для наглядности будем считать, что на входе алгоритма есть некоторая СВЗ-модель в виде $\mathcal{A}_v^{t,sf}$.

Суть работы алгоритма заключается в циклическом переборе фактов из $\mathcal{A}_v^{t,sf}$ с целью формирования утверждений вида $\varphi_i: x_j \rightarrow x_k$, задающих вычислимые отношения (операторы), преобразующие вход $\mu(x_j)$ в выход $\mu(x_k)$, утверждений о связях операторов-предикатов и других операторов $\delta_i: \varphi_i \rightarrow \varphi_j^P$, о связях параметров и операторов-предикатов $\varphi_i^P: x_j \rightarrow \{True, False\}$. Такие утверждения формируются исходя из терминологических аксиом для O^t , задающих правила описания операторов исполнения запросов, их входов и выходов, вычислительных задач, с ними связанных, условий применимости.

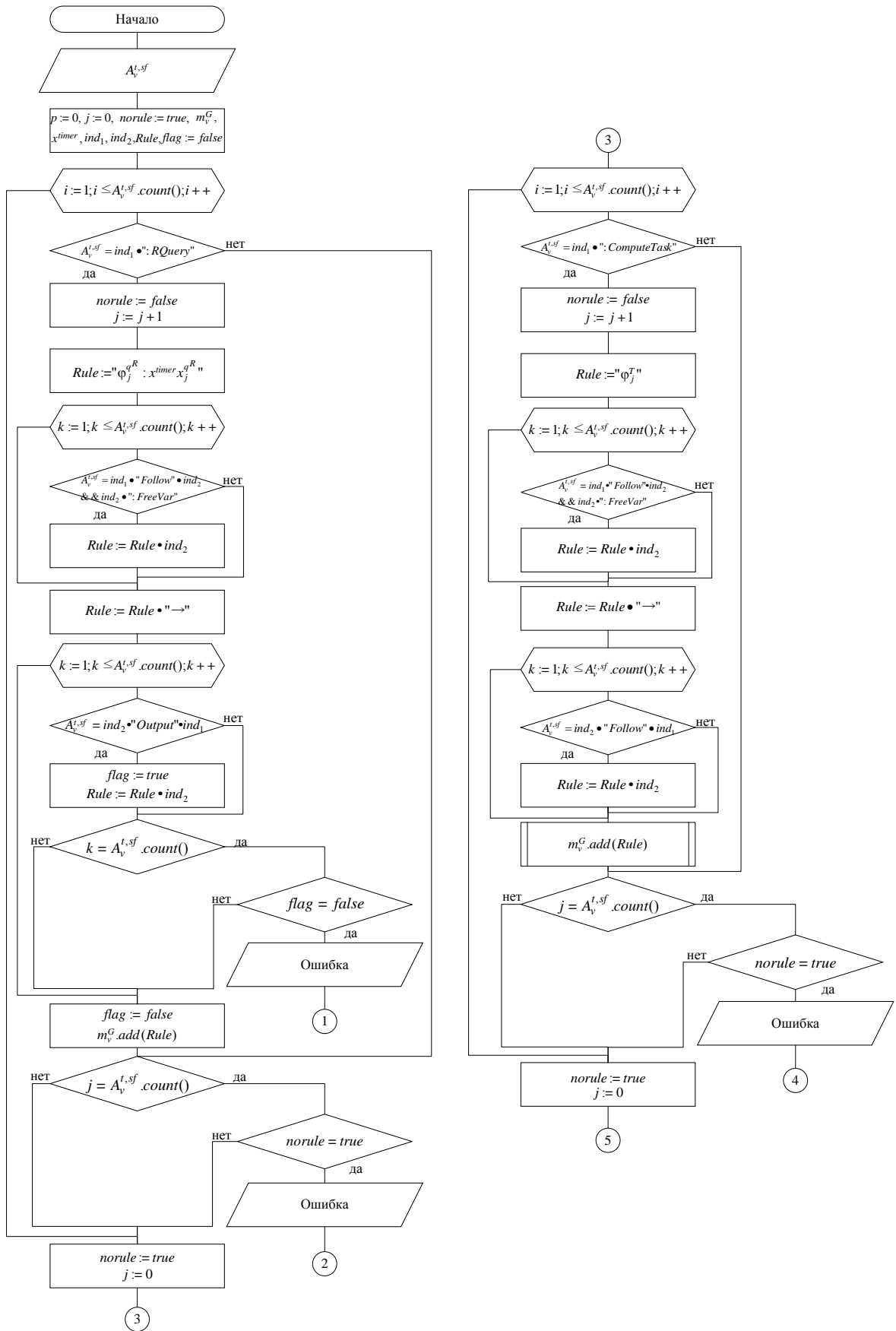


Рис. 2.11. Блок-схема алгоритма формирования G-модели по модели согласованных вычислительных задач (часть 1)

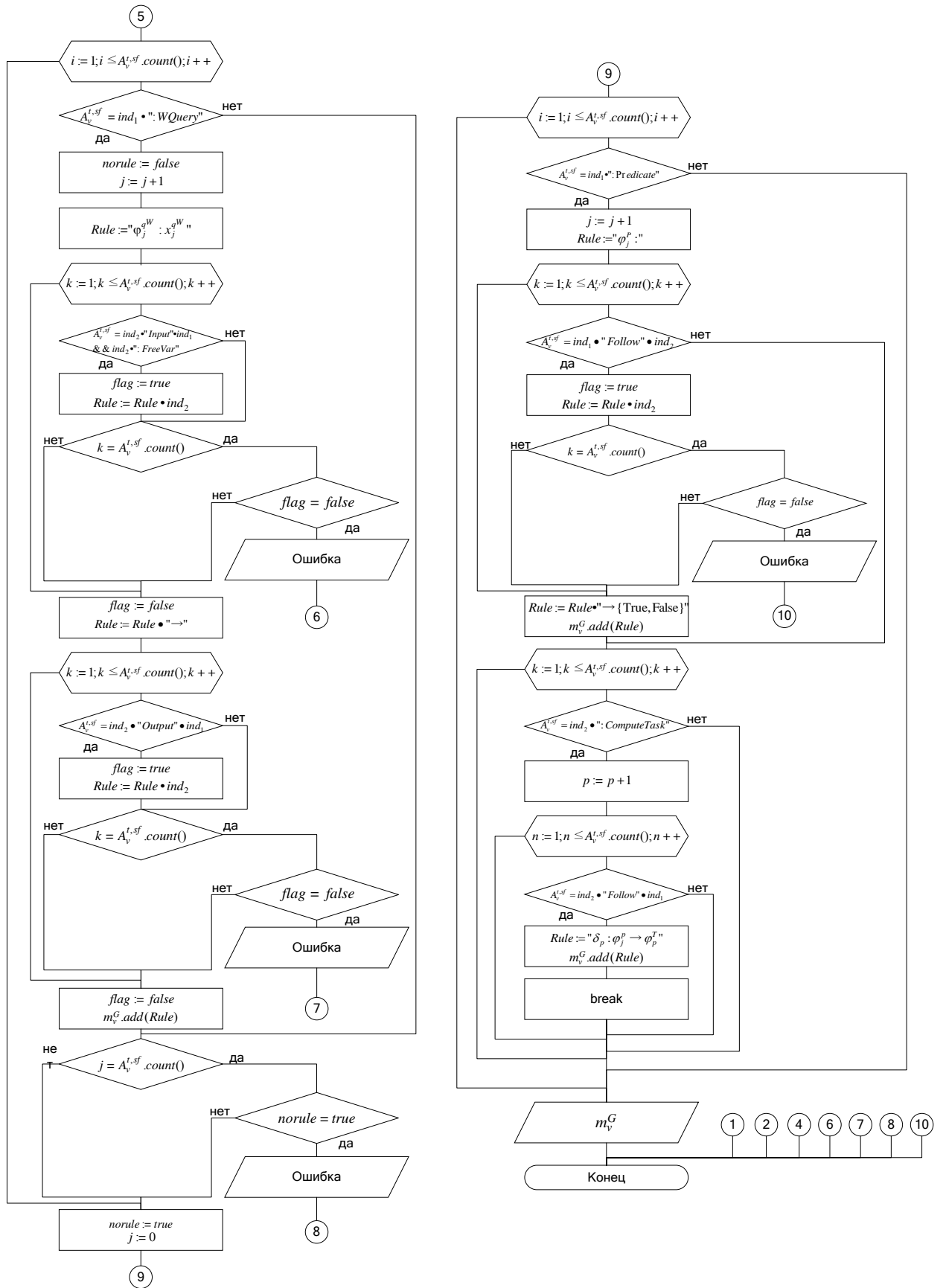


Рис. 2.12. Блок-схема алгоритма формирования G-модели по модели согласованных вычислительных задач (часть 2)

В рамках блок-схемы алгоритма из рисунка 2.11 считается, что « \cdot » – операция конкатенации, $count()$ – процедура, возвращающая в качестве результата количество триплетов в $\mathcal{A}_v^{t,sf}$. При этом некоторые из записей в условных операторах сокращены с целью сохранения наглядности блок-схемы. Переменные $ind1$ и $ind2$ используются для временного хранения индивидов в триплетах, $Rule$ – переменная, в значении которой формируются выражения вычислимых отношений, причем add – процедура конкатенации, склеивающая элементарные G-модели, переменные-счетчики j и p используются для сохранения упорядоченности элементов множеств операторов, логические переменные $flag$ и $norule$ используются для проверки отсутствия необходимых связей в триплетах. Содержательно они обозначают следующее: если по окончании работы циклов они принимают значение $false$, то это означает, что не найдены входные или выходные параметры соответствующего вычислимого отношения. Последовательность внешних циклов алгоритма предполагает следующий порядок создания элементов G-модели: набор вычислимых отношений запросов выборки, набор вычислимых отношений вычислительных задач, набор вычислимых отношений запросов на запись, набор отношений параметров и предикатов.

Пример 2.3. Пусть имеется СВЗ-модель sf из примера 2.2. Воспользуемся алгоритмом, описанным выше, для формирования G-модели m_v^G по sf . Тогда в результате работы алгоритма может быть сформирована G-модель, которая описана следующей совокупностью выражений вычислимых отношений при условии, что задано множество параметров $X = \{x^{timer}, x_1^{qR}, x_2^{qR}, x_3^{qR}, x_1^{qW}, x_2^{qW}, x_3^{qW}, x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}\}$ (см. рис. 2.13):

$$\begin{array}{ll}
 \varphi_1^{qR} : x^{timer} x_1^{qR} \rightarrow x_1 x_2 & \varphi_2^{qW} : x_2^{qW} x_8 \rightarrow x_9 \\
 \varphi_2^{qR} : x^{timer} x_2^{qR} \rightarrow x_3 x_4 & \varphi_3^{qW} : x_3^{qW} x_{10} x_{11} \rightarrow x_{12} \\
 \varphi_3^{qR} : x^{timer} x_3^{qR} x_6 \rightarrow x_7 & \varphi_1^P : x_2 x_4 \rightarrow \{True, False\} \\
 \varphi_1^T : x_1 x_2 \rightarrow x_5 & \varphi_2^P : x_9 \rightarrow \{True, False\} \\
 \varphi_2^T : x_3 x_4 \rightarrow x_8 & \delta_1 : \varphi_1^P \rightarrow \varphi_2^T
 \end{array}$$

$$\varphi_3^T : x_7 x_9 \rightarrow x_{10} x_{11}$$

$$\varphi_1^{qW} : x_1^{qW} x_5 \rightarrow x_6$$

$$\delta_3 : \varphi_2^P \rightarrow \varphi_3^T$$

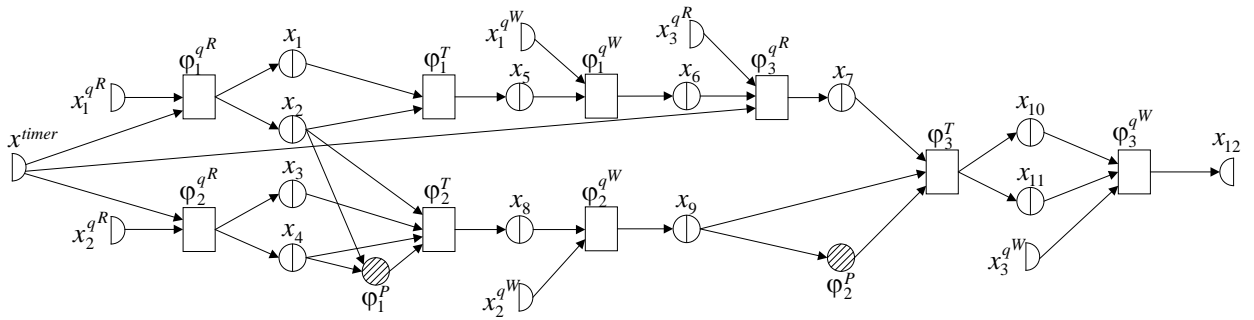


Рис. 2.13. Пример G-модели, сформированной по СВЗ-модели

Таким образом, по СВЗ-модели может быть сформирована G-модель. Этот важный факт позволяет говорить о двух важных аспектах предлагаемого подхода [132].

Во-первых, определенная для G-моделей концепция инвариантности состояния реального объекта и состояния вычислительного процесса с учетом возможности описания потоковых, параллельных и асинхронных процессов позволяет в рамках автоматизированной информационно-аналитической поддержки ЖЦ СОТО обеспечивать пользователей СИАП ЖЦ СОТО в потоковом режиме вычислений на основе управления по данным актуальной и достоверной информацией о ТС СОТО. При этом хотя информация из источников данных не является носителем 100% достоверных сведений о рассматриваемом объекте, тем не менее, такая информация – это, во-первых, единственный объективно существующий источник сведений о поведении СОТО, и, во-вторых, этот набор сведений может быть подвергнут контролю, обработке, анализу (см. рис. 2.14). В результате использования такого подхода наблюдается соответствие полимодельного комплекса составному элементу архитектуры, предложенной в приложении 3.3.3 и обеспечивается наблюдаемость ТС СОТО.

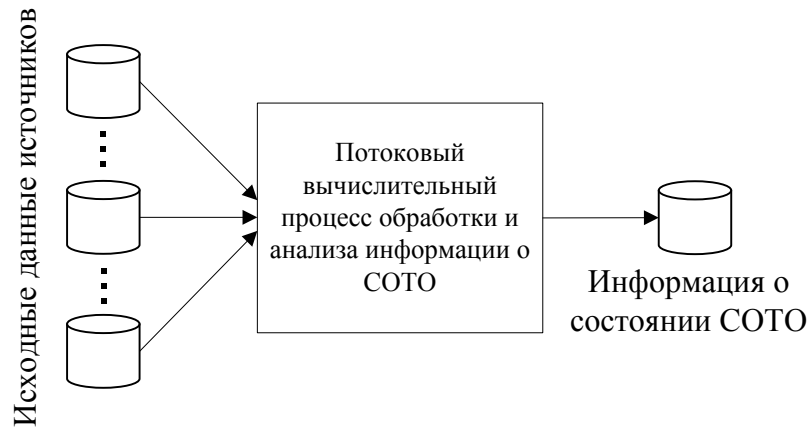


Рис. 2.14. Схема оценивания технического состояния СОТО

Во-вторых, аппарат G-моделей позволяет осуществлять формальную проверку корректности и адекватности описываемых явлений, и на них можно исследовать вопрос о разрешимости задачи анализа информации для конкретной цели анализа, а для разрешимой задачи синтезировать программу ее выполнения, не прибегая к вычислению значений параметров.

2.4.4. Автоматическая генерация схем программ и организация вычислений по ним

Под программой в широком смысле можно понимать символьное описание способа взаимодействия с символьным окружением, однозначно определяющее изменение его состояния [73]. В теоретическом программировании часто рассматриваются два направления – верификация программ и их синтез, которые предполагают, что сущность программы можно выразить отношением, связывающим исходные данные с результатами вычислений. При успешном решении проблемы синтеза отпадает необходимость не только писать программу, но и отлаживать ее. Одним из приоритетных направлений исследований в формировании программ является модельно-ориентированный подход. Так, например, к числу различных подходов формализации моделей программ можно отнести [132, 158, 159, 209]: классические языки логического управления, булевы функции, таблицы истинности и таблицы решений, функциональные схемы, граф-схемы алгоритмов, логические схемы алгоритмов,

язык SDL, P-схемы, язык «Графсет», биологические графы, схемы Карпа-Миллера, потоковые схемы, сети Петри и графы операций.

Синтез программы может быть осуществлен на основе метода доказательства теорем из предположения, что всякая теория строится на базе некоторого языка [193]. Синтаксически распознаваемые выражения в языке называют формулами и задают правила вывода, позволяющие из одних формул выводить другие. При этом в языке выделяют несколько формул, считающихся априори истинными (аксиомами). Формулы, выводимые на основе аксиом, называются теоремами. С учетом этого получение программы путем может быть произведено за три этапа:

- Описание условий задачи в виде отношения $R(X^+, X^-)$, связывающего исходные данные X^+ с результатами решения X^- , в рассматриваемом случае задаваемое некоторой G-моделью.
- Построение конструктивного доказательства теоремы существования решения задачи, заданной в виде:

$$\forall(X^+ \subseteq X)\exists(X^- \in X_g)R[X^+, X^-]. \quad (2.39)$$

- Перевод полученного доказательства в форму программы или схемы программы, с которой работает операционная среда (решатель искусственно интеллектуальной системы).

Одним из теоретических направлений исследований, связанных с построением моделей программ является схематология [107, 132]. В её рамках программа представляется как $P = (S, I)$, где P – программа вычислений, S – схема программы, I – интерпретация, связывающая символы переменных с областями их значений и их начальными значениями, а функциональные и предикатные символы – с конкретными функциями и предикатами. Сама схема программы при этом задается выражением $S = (M, A, C)$, где M – конечное множество элементов памяти (абстракции ячеек, очередей, стеков и т.д.), A – конечное множество операторов (условные оп-ры, и оп-ры обработки), C – элементы управления вычислительным процессом. Такое представление позволяет изучать свойства достаточно широких классов программ, игнорировать

несущественные синтаксические детали программы, допускать различные способы эквивалентных преобразований схем, определять разрешимые и неразрешимые свойства программ.

В основе моделирования программ на основе G-моделей лежат схемы программ, называемые G-сетями, представляющие собой граф, на котором определены вершины-позиции, вершины-операторы и дуги, наделенные весами – временем выполнения оператора. G-сеть определяется как на основе понятий G-модели так и потоковых схем программ, адекватно объединяя в себе управляющую и информационную структуру, свойственные программам вычислений. По существу G-сеть является модификацией сети Петри (предназначенной для моделирования асинхронных параллельных процессов динамических дискретных систем) и представляется кортежем [132]:

$$S^G = \langle A_S, X_S, V_S, R_S, F_S, \mu_S \rangle, \quad (2.40)$$

где

$A_S = \{a_i | i \in I_A\}$ – конечное непустое множество позиций G-сети. Множество позиций A_S разбивается на два непересекающихся подмножества: $A_S = A_S^0 \cup A_S^P$, среди которых A_S^0 – множество основных позиций и A_S^P – множество р-позиций;

$X_S = \{x_i | i \in I_X\}$ – конечное непустое множество атрибутов G-сети, причем такое, что отображение $A_S \rightarrow X_S$ является биекцией;

$V_S = \{v_i | i \in I_V\}$ – конечное непустое множество переходов G-сети. Множество переходов V_S разбивается на два непересекающихся подмножества: $V_S = V_S^0 \cup V_S^P$, среди которых V_S^0 – множество основных переходов и V_S^P – множество р-переходов;

$R_S = R_S^+ \cup R_S^-$ – множество инцидентности, причем $R_S^+ \subseteq X_S \times V_S$, $R_S^- \subseteq V_S \times X_S$;

$F_S = \{\varphi_i | i \in I_F\}$ – конечное множество операторов таких, что отображение $V_S \rightarrow F_S$ является биекцией;

$\mu_S: A_S \rightarrow \mathbf{N}^+$ - функция разметки G-сети, сопоставляющая каждому $a_i \in A_S$ целое неотрицательное число.

G-сети позволяют моделировать различные по мощности вычисления и представляют собой универсальную моделирующую систему, порождающую рекурсивно-перечислимые классы языков и эквивалентную по своей мощности машине Тьюринга. При этом на такой модели выполняются свойства сетей Петри, такие как ограниченность, безопасность, сохраняемость, достижимость, живость [12, 13, 127, 133].

Процесс синтеза программ предполагает дедуктивное вычисление утверждений программы, основанное на утверждениях о её функциях на основе некоторой формально-дедуктивной системы [73]. В качестве такой системы может выступать алгоритм унификации, метод резолюций и др., которые однако не получили широкого распространения в силу своей сложности [73, 106]. В связи с этим в [132] была введена формальная атрибутивная р-грамматика, которая, с одной стороны, позволяет осуществлять формальный вывод G-сетей по G-моделям с построением графа вывода, а с другой – осуществлять проверку разрешимости вычислительной задачи в формируемой схеме программы.

Заметим, что процесс формирования G-сети по сформированной ранее G-модели, в целом, не отличается от предложенного формального вывода на основе р-грамматик в [132] за исключением одной особенности. В связи с тем, что для G-моделей было введено несколько типов операторов (см. раздел 2.4.3) и параметров, обеспечение сквозной индексации терминальных и нетерминальных символов для корректной работы алгоритма вывода требует, чтобы:

- индексация начиналась с $x_i^{timer} \mid i = 1$;
- для параметров запросов выборки $x_i^{q^R} \mid i \in I^{q^R}, i = i + 1$ (следование за x^{timer});
- для параметров запросов на запись $x_i^{q^W} \mid i \in I^{q^W}, i = i + |I^{q^R}|$, где $|I^{q^R}|$ – мощность множества I^{q^R} ;

– для промежуточных и выходных параметров $x_i \mid i \in I^x, i = i + |I^{q^R}| + |I^{q^W}|$, где $|I^{q^W}|$ – мощность множества I^{q^W} .

Пример 2.4. Пусть имеется G-модель m^G из примера 2.3. Тогда с использованием r-грамматики может быть получена следующая G-сеть s^G (см. рис. 2.15) с заданием матриц инцидентности R_S^+ (см. таблицу 2.3) и R_S^- (см. таблицу 2.4).

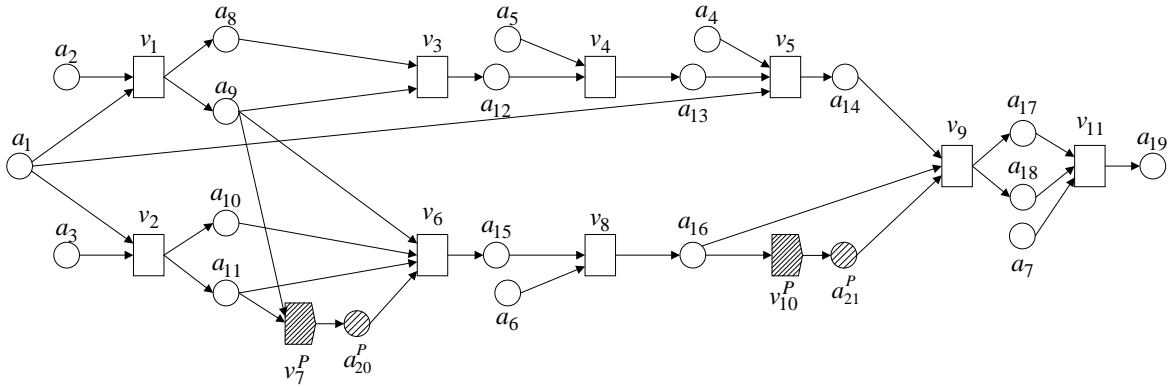


Рис. 2.15. Пример G-сети

Таблица 2.3. Матрица инцидентности R_S^+

	v_1	v_2	v_3	v_4	v_5	v_6	v_7^P	v_8	v_9	v_{10}^P	v_{11}
a_1	1	1			1						
a_2	1										
a_3		1									
a_4					1						
a_5				1							
a_6								1			
a_7											1
a_8			1								
a_9			1			1	1				
a_{10}						1					
a_{11}						1	1				
a_{12}				1							
a_{13}					1						
a_{14}									1		
a_{15}								1			
a_{16}										1	
a_{17}											1
a_{18}											1
a_{19}											
a_{20}^P						1					
a_{21}^P									1		

Таблица 2.4. Матрица инцидентности R_S^-

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}	a_{16}	a_{17}	a_{18}	a_{19}	a_{20}^P	a_{21}^P	
v_1								1	1													
v_2										1	1											
v_3												1										
v_4													1									
v_5														1								
v_6															1							
v_7^P																					1	
v_8																1						
v_9																	1	1				
v_{10}^P																						1
v_{11}																				1		

Семантика конструкции G-сети, связанная непосредственно с организацией потокового режима вычислений с управлением по данным, предполагает наличие монитора как некоторой системы массового обслуживания, позволяющей «оживлять» событийно-ориентированные имитационные модели [127]. Такая система обслуживает поток заявок, интерпретируемый в данном случае как процесс поступления новых фишек в позиции. Можно заметить, что любая позиция $a_i \in A_S$ G-сети представляет собой элемент памяти, предназначенный для хранения «фишек». Причем тип этого элемента памяти – очередь с дисциплиной обслуживания хранимых в ней фишек FIFO (First In – First Out) – «первый вошел – первый вышел» [127]. В качестве системы массового обслуживания для G-сетей используется специального вида Q-схемы [132]. Непосредственно же вычисления определяются G-автоматом – конечным автоматом, осуществляющим смену состояний G-сети в виде смены её маркировки, причем функция переходов – есть отношение вычислимости на параметрах (позициях) G-сети. При этом, поскольку данные на вход системы (фишек в позиции) могут поступать непрерывно, а процесс анализа может в целом вообще не предполагать остановки программы, реакция системы должна укладываться в прогнозируемое время (РВ жестого или мягкого типа) [194]. В связи с этим вычислительная система на основе G-сетей организует процесс вычислений на основе формирования системы приоритетов (динамических, с вытеснением), которой традиционно обладают системы РВ.

2.4.5. Некоторые аспекты организации вычислений для систем информационно-аналитической поддержки

Необходимо обратить внимание, что процесс вычислений на G-сети для рассматриваемого в работе случае всегда начинается с появления фишки в позиции a_1^{timer} , полученной на основе параметра x^{timer} , которая затем инициализирует соответствующие операторы запросов выборки в соответствии с заданной G-сетью.

Следующим немаловажным аспектом является непосредственно исполнение запросов, определенных в рамках СВЗ-моделей. При сравнении примеров 2.2 и 2.3 можно заметить, что при переходе на G-модель не были учтены индивидуальные конъюнктов запросов. Это связано с тем, что эти элементы введены не как переменные значения, а как константы, указывающие, с одной стороны, на структурный аспект G-модели (по свободным переменным запросов были получены параметры G-модели, в значениях которых и должны оказаться результаты исполнения запроса), а с другой – указывающие на факты ПрО, которые необходимо извлечь из источников данных (или поместить их в хранилище электронного паспорта). Поскольку входной информацией для оператора исполнения запроса выборки является непосредственно текст запроса (в рассматриваемом случае в нотации SPARQL) и множество спецификаций источников данных (что упоминалось в приложении 3.3.4), то необходимо также определить алгоритм трансляции описания запроса в СВЗ-модели в нотацию ЯЗ SPARQL, а результат его исполнения поместить в качестве фишки в позицию, связанную со специальным параметром x_i^{qR} (x_j^{qW}), входным для оператора исполнения запроса φ_i^{qR} (φ_j^{qW}) (см. раздел 2.4.3). Схема такого алгоритма представлена ниже на рисунке 2.16.

В данном алгоритме « \cdot » – операция конкатенации строк, *count* – процедура, возвращающая количество элементов в $\mathcal{A}_v^{t, sf}$, q – индивид некоторого запроса выборки СВЗ-модели, x – переменная для сохранения промежуточных

результатов расчетов. Суть работы алгоритма можно пояснить на примере трансляции спецификации запроса из примера 2.2.

Пример 2.5. Пусть имеется СВЗ-модель из примера 2.2, представленная системой фактов в $\mathcal{A}_v^{t, sf}$, в которой описание запроса q_1^R задано отношениями по входу и выходу выражениями:

$$\begin{array}{ll} ? x_1 \text{ URI}_1 \# R_1 \text{ URI}_1 / b \text{ Input } q_1^R & x_1 \text{ Output } q_1^R \\ \text{URI}_1 / b \text{ URI}_1 \# R_2 ? x_2 \text{ Input } q_1^R & x_2 \text{ Output } q_1^R \end{array}$$

Тогда в результате вычислений по алгоритму, представленному выше, будет получен онтологический запрос в нотации SPARQL вида:

```
SELECT ? x1 ? x2
WHERE {
  ? x1 URI1 # R1 URI1 / b
  URI1 / b URI1 # R2 ? x2
}
```

Очевидно, аналогичным образом может быть задан алгоритм трансляции для запросов на запись, чтобы в результате получалось выражение вида П.(3.4) (см. приложение 3). Необходимо заметить, что в запросах на запись также фигурируют связанные переменные в составе конъюнктов. Эта особенность связана с тем, что на этапе функционирования СИАП ЖЦ СОТО оператор исполнения запроса на запись $\varphi_j^{q^W}$ должен осуществлять операцию подстановки конкретного значения c из области значений $\mu(x_i)$ параметра x_i , входного для оператора $\varphi_j^{q^W}$, на место литерала связанной переменной в описании конъюнкта: $c \rightarrow ? x_k$, причем для этого в каждом операторе $\varphi_j^{q^W}$ должны быть заданы правила подстановки $x_i \rightarrow ? x_k$.

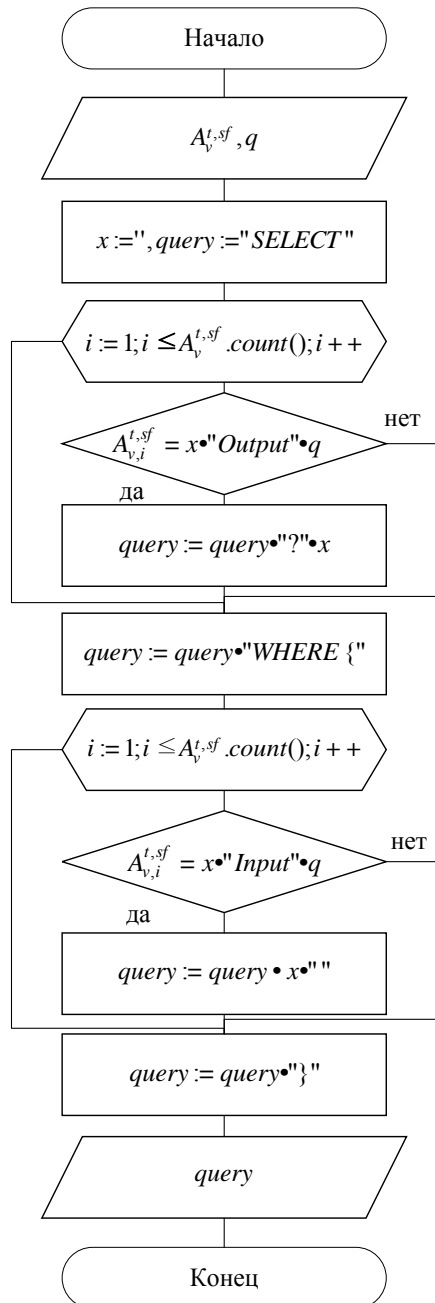


Рис. 2.16. Блок-схема алгоритма трансляции описания запроса в СВЗ-модели на язык SPARQL

Необходимо обратить внимание на следующий факт. Последовательное исполнение очереди запросов выборки в рамках одного оператора с заданной частотой (как это было определено в разделе 2.4.3) при условии, что не предполагается останова программы, и возвращаемый результат представляет собой непустое множество кортежей отношения $R1$ (таблицы результатов исполнения запроса в терминах реляционной алгебры [204]) вида $\langle c_1, c_2, \dots, c_n \rangle$,

где c_1, c_2, \dots, c_n – константы и задан вектор свободных переменных \vec{x} размерности n , приведет к бесконечному пополнению очередей фишек с одинаковыми данными в позициях G-сети a_1, a_2, \dots, a_n , что, безусловно, не является ожидаемым поведением вычислительной системы. Предполагается, что процесс информационно-аналитической поддержки ЖЦ СОТО должен обрабатывать только новые порции данных Δ_d на момент времени t_i этапа функционирования, такие что $\Delta_d = d_i \setminus d_{i-1}$, где d_i – данные, поступившие на момент времени t_i , а d_{i-1} – данные поступившие ранее на предыдущей инициализации оператора. В связи с этим необходимо, чтобы в рамках оператора был определен вектор \vec{y} размерности n , определяющий некоторое отношение $R2$, а алгоритм исполнения запроса после получения результата выполнял следующую последовательность шагов:

- выполнить операцию вычисления разности отношений $R1 \setminus R2$, определяемую как разность множеств, и получить дополнение R' ;
- запомнить R' : $R2 = R2 \cup R'$;
- вернуть результат R' .

Еще одним аспектом проектирования СИАП ЖЦ СОТО является тот факт, что поскольку в рамках модели бизнес-процессов может быть определено несколько связанных ЖЦ моделей процессов $Pr = \{pr_v | v = 1, \dots, p\}$, для каждого из которых в конечном счете может быть синтезирована своя G-модель m_v^G из $M^G = \{m_v^G | v = 1, \dots, p\}$, то и для каждой модели m_v^G должна быть синтезирована своя G-сеть m_v^S из $M^S = \{m_v^S | i = 1, \dots, n\}$. В результате по описанию модели бизнес-процессов может быть получена совокупность программ, что полностью отражает архитектуру, рассмотренную в приложении 3.3.3. При этом порядок формирования соответствующих моделей совместно с заданием онтологических отображений источников данных представлен на рисунке 2.17. В таком случае, остается открытым лишь вопрос порядка взаимодействия программ (в соответствии с требованием интероперабельности процессов) информационно-аналитической поддержки ЖЦ СОТО, который будет рассмотрен далее.

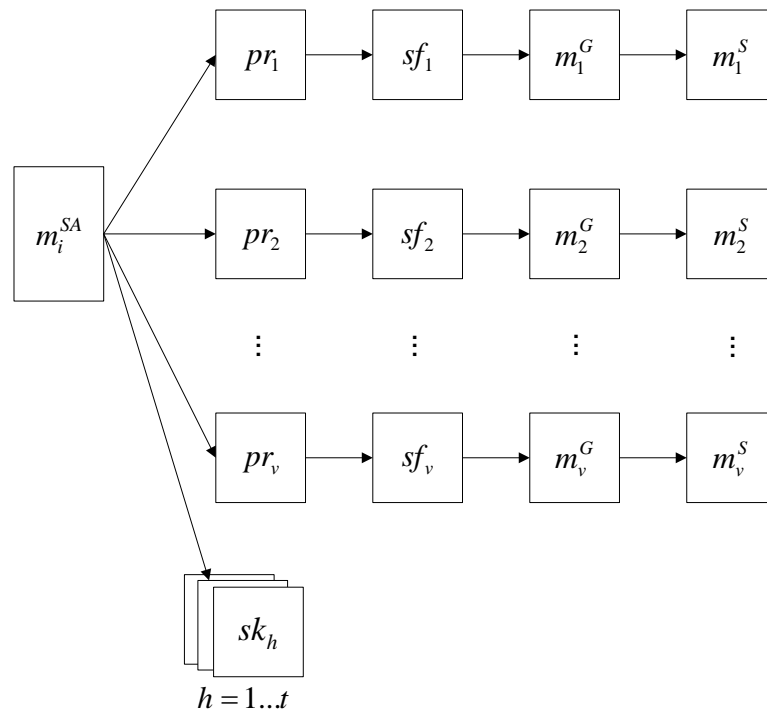


Рис. 2.17. Схема порядка формирования моделей информационно-аналитической поддержки ЖЦ СО

2.4.6. Мультиагентная интерпретация полимодельного комплекса информационно-аналитической поддержки жизненного цикла сложных объектов

До сих пор определенные ранее онтологии верхнего уровня (раздел 2.2), предметные онтологии (приложение 3), онтологии задач (раздел 2.3) и прикладные онтологии (раздел 2.4) использовались для спецификации концептуализации ПрО и последовательного уточнения этих спецификаций. В то же время существенное преимущество онтологического подхода связано с возможностью решения задачи интеграции знаний (или, в рассматриваемом контексте, ЕАИ – интеграции приложений), позволяющей, в отличие от известных МПЗ типа семантических сетей, фреймов и продукций, координировать работу междисциплинарных команд по разработке прикладного ПО в рамках методологии Workflow без непосредственно программирования в кодах [176]. Так, в отличие от классического таксонометрического подхода использования онтологий предложенные онтологические МПЗ позволяют на конструктивной

основе формировать модели процессов, последовательно уточняя их до моделей программ. При этом модели процессов можно наблюдать в рамках единого представления, явно определяя связи между ними, таким образом определяя их семантическую интероперабельность. При этом существенно, что в таком случае обеспечивается единая терминологическая семантика моделей процессов [54].

В то же время такая семантическая «совместимость» в том контексте, в каком эти онтологии используются, может быть использована для последующей организации взаимодействия элементов синтезируемого ПК, представляющего собой совокупность взаимодействующих программ. Два программных приложения могут быть полностью интегрированы только тогда, когда они используют одну и ту же семантику терминологии [54]. Одним из способов идентификации семантики является рассмотрение поведения приложения. Поведение приложения полностью зависит от его логических выводов, что, в свою очередь, полностью зависит от семантики его аксиом. В действительности поведение приложения ограничивается семантикой его онтологии, потому что любые выводы, сделанные приложением, должны соответствовать семантике его внутренних знаний в сочетании с любой информацией, полученной из программного окружения или других приложений.

При подобном рассмотрении можно говорить о формировании кооперативной *MAC* (что упоминалось в приложении 2.3.6), представляющей собой совокупность рациональных агентов (как программ), решающих вычислительные задачи и идущих к общей цели распознавания ТС СОГО на основе поиска в пространстве состояний [46, 169, 176]. В любой конкретный момент времени оценка рациональности действий агента зависит от факторов производительности (функционирования в потоковом режиме в РВ), знаний агента о среде (выражаемых схемами источников данных Sk), плана агента (выражаемого в виде альтернативного И-ИЛИ графа модели процесса). При этом в предлагаемом подходе для каждого агента явно определена его БЗ $\mathcal{K}^{meta,t}$. В результате

совокупность получаемых программ информационно-аналитической поддержки ЖЦ СОТО можно рассматривать как распределенный искусственный интеллект.

При этом в рамках исследований агентных технологий предполагается, что должна быть задана некоторая модель переговоров агентов, необходимая для достижения их общих целей и связанная с определением операций «Tell» (сообщить) и «Ask» (запросить) [169]. Для специфирования такой модели переговоров при взаимодействии процессов дополним \mathcal{T}^{meta} БЗ онтологии \mathcal{O}^{meta} выражениями, определяющими механизм передачи сообщений между процессами:

$$Message \equiv \exists MessageTo. (StartEvent \sqcup InterEvent), \quad (2.41)$$

$$Message \sqsubseteq InterEvent \sqcup FinalEvent, \quad (2.42)$$

означающими, что сообщением является определяемое или целевое событие, которое может быть передано исходному или определяемому событию.

Говорят, что два объекта взаимодействуют, если история состояний одного из них зависит от присутствия другого объекта [198, 199, 200, 234]. Состояние объекта 1 можно рассматривать как внешнее событие (сообщение) для объекта 2, которое инициирует трансформацию этого объекта в терминах БВВ-онтологии или вызывает смену состояния (маркировки G-сети) в соответствии с ходом функционирования G-автомата.

В рассматриваемом случае введенные выражения (2.41), (2.42) с учетом того, что указанные в них события определяются соответствующими запросами выборки и запросами на запись (см. (2.22) – (2.24)), позволяют специфицировать взаимодействие двух процессов в виде пары структурно подобных запросов на запись и на чтение. Поскольку в соответствии с архитектурой, рассмотренной в приложении 3.3.3, программы записывают результаты в хранилище электронного паспорта и могут зачитывать данные оттуда же, используя единый тезаурус, заданный в концептуальной схеме хранилища, то два про-

цесса могут взаимодействовать следующим образом: если некоторый оператор $\varphi_i^{q^W}$ исполнил запрос на запись данных в хранилище, то другой оператор $\varphi_j^{q^R}$ исполнения запроса выборки в соответствии с описанным в разделе 2.4.5, получит эти же данные при его инициализации. Для реализации такого механизма очевидно необходимо осуществлять проверку того, что запросы:

- обращены к одному информационному ресурсу по *URI*;
- имеют одинаковое количество конъюнктов;
- имеют конъюнкты одинаковой структуры, т.е. в соответствующих позициях триплетов из П.(3.1) (см. приложение 3) находятся соответственно переменные или литералы.

Суть такого подхода можно проиллюстрировать на примере.

Пример 2.6. Пусть имеется модель бизнес-процессов из примера 2.1. Дополним эту модель описанием еще одного процесса и зададим сообщение для передачи в другой процесс (иллюстрация примера представлена на рис. 2.18):

<i>process₂:Process</i>	<i>e₆ Defines process₂</i>	<i>e₇:Message</i>
<i>e₆:StartEvent</i>	<i>e₇ Defines process₂</i>	<i>e₇ MessageTo e₂</i>
<i>e₇:FinalEvent</i>	<i>a₄ Defines process₂</i>	
<i>a₄:Action</i>	<i>a₄ Follow e₆</i>	
<i>process₂ Defines sa₁</i>	<i>e₇ Follow a₄</i>	
<i>process₂ Defines lc₁</i>		

В соответствии с примером 2.2 задано описание запроса для e_2 . Дополним также рассматриваемый пример описанием запроса e_7 :

<i>? x₃ ? x₄ URI₂/d Input q₂^R</i>	<i>q₄^W Defines e₇</i>
<i>x₃ Output q₂^R</i>	<i>? x₁₃ ? x₁₄ URI₂/d Input q₄^W</i>
<i>x₄ Output q₂^R</i>	<i>? x₁₅ Output q₄^W</i>

В результате трансляции онтологического описания запросов с использованием алгоритма, блок-схема которого представлена на рисунке 2.16, будут получены SPARQL-запросы следующего вида:

<i>SELECT ? x₃ ? x₄</i>	<i>INSERT DATA {</i>
<i>WHERE {</i>	<i>? x₁₃ ? x₁₄ URI₂/d</i>
<i>? x₃ ? x₄ URI₂/d</i>	<i>}</i>
<i>}</i>	

Очевидно, что на этапе функционирования в результате подстановки значений в x_{13} x_{14} данные, записанные в хранилище по адресу URI_2 , затем будут получены среди прочих новых значений в результате исполнения запроса выборки в силу идентичности расстановки по позициям переменных в триплетах.

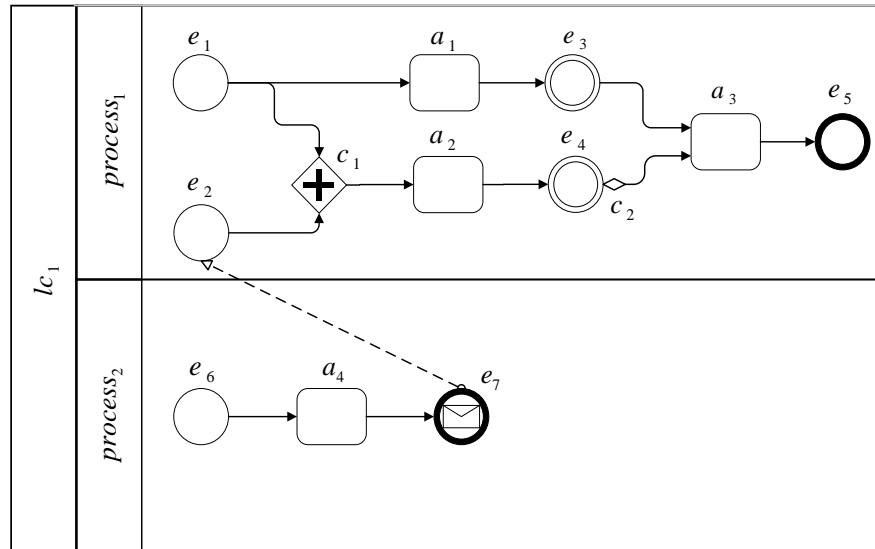


Рис. 2.18. Пример модели бизнес-процессов информационно-аналитической поддержки ЖЦ СОТО

Предложенный подход реализует опосредованную коммуникацию агентов посредством модели «классной доски», описываемую выражениями (2.41), (2.42) [101]. Канонически, это предполагает, что агенты анализируют информацию на доске, пытаясь найти применение своим возможностям, связанным с решением задач, а по окончании помещают результат обратно на доску. В рассматриваемом случае в качестве такой доски выступает непосредственно хранилище электронного паспорта СОТО, а потоковый характер функционирования в соответствии с закладываемым планом в виде И-ИЛИ графа определяет тот факт, что сообщение, отправленное одним агентом другому, инициирует выполнение связанных с запросом задач, следующих (*Follow*) за ним.

Необходимо также отметить, что указанный подход находит интерпретацию в формальной модели взаимодействия интеллектуальных агентов, предложенной Рыбиной Г.В. и ставшей достаточно популярной и цитируемой во

многих работах [173]. Таким образом, в соответствии с ней модель интеллектуального агента может быть представлена следующим кортежем:

$$M^{IA} = \langle M, Q, P, R, E, L \rangle, \quad (2.43)$$

где

M – множество моделей окружения, представимое совокупностью концептуальных схем источников данных Sk ;

Q – множество целей, задаваемое в модели бизнес-процессов в виде совокупности целевых событий *FinalEvent*;

P – множество планов, задаваемое альтернативными путями (трассами) бизнес-процессов на основе условий применимости (начала решения задач);

R – множество действий, явно присутствующее в модели бизнес-процессов, и, в конечном счете, отображаемое в множество вычислимых отношений;

E – множество событий в окружении, явно заданное в модели бизнес-процессов и уточняемое в виде совокупности онтологических запросов;

L – язык взаимодействия агентов, представляющий в рассматриваемом случае совокупность терминов (слов языка) концептуальных схем источников данных, используемых в конъюнктах онтологических запросов.

Соответствие предлагаемого подхода формализации агента в [173], обеспечение единого языка взаимодействия агентов, возможность спецификации бизнес-процессов в едином представлении с явным заданием порядка их взаимодействия и последующим уточнением их вплоть до формирования схем программ обеспечивает семантическую *интероперабельность* между получаемыми программами, в результате чего можно говорить о реализуемости МАС как полимодельного комплекса автоматизированной информационно-аналитической поддержки ЖЦ СОТО.

2.4.7. Результаты исследования задачи моделирования решения аналитических вычислительных задач и синтеза их схем программ

Разработан алгоритм формирования моделей решения аналитических вычислительных задач по СВЗ-моделям, позволяющий осуществить автоматический переход от спецификации согласования задач к структурно-функциональным G-моделям, что в результате обеспечивает возможность на конструктивной основе осуществлять проверку разрешимости соответствующих задач, синтезировать модели программ, а также декомпозировать процесс разработки прикладного ПО СИАП ЖЦ СОТО, отделив моделирование решения задач от специфицирования поведенческих и функциональных требований.

Разработан алгоритм трансляции спецификаций онтологических запросов, задаваемых в СВЗ-моделях, на синтаксис SPARQL-запросов, что позволяет в рамках процедуры специфицирования ПрО в терминах, ориентированных на описание фактов и явлений ПрО, формировать программно-ориентированное синтаксическое представление команд ЯЗ.

Дополнены и обоснованы аспекты организации вычислений по G-моделям в ориентации на процессы автоматизированной информационно-аналитической поддержки ЖЦ СОТО, связанные с исполнением онтологических запросов в рамках парадигмы потоковых вычислений.

Предложена и формально обоснована мультиагентная интерпретация полимодельного комплекса информационно-аналитической поддержки ЖЦ СОТО, позволяющая обеспечивать решение задачи семантической интероперабельности процессов и интеграции знаний.

2.5. Выводы по разделу

1. В качестве формального базиса для формирования согласованной системы уточняющих спецификаций бизнес-процессов ЖЦ СОТО, функциональных требований, порядка интеграции информационных ресурсов и решения аналитических вычислительных задач была разработана онтологическая система информационно-аналитической поддержки ЖЦ СОТО, позволяющая в рамках интегрированной БЗ связать воедино и согласовать разноаспектные модели ПрО, формируемые в составе систем фактов онтологий.
2. Разработана онтологическая МПЗ о бизнес-процессах информационно-аналитической поддержки ЖЦ СОТО, позволяющая задавать формальные спецификации совокупности событийно-ориентированных процессов, связанных ЖЦ СОТО, в ориентации на извлечение и интеграцию знаний о ПрО как требований, предъявляемых к СИАП ЖЦ СОТО с одной стороны, и как структурно-логическое описание реальных организационных и технологических процессов – с другой, с формализацией мультиагентной архитектуры ПК и обеспечением семантической интероперабельности его элементов.
3. Разработана онтологическая МПЗ о СВЗ, позволяющая на формальной основе уточнять спецификации бизнес-процессов информационно-аналитической поддержки ЖЦ СОТО в виде СВЗ-моделей, осуществляя инвариантный переход от описания ПрО в виде совокупности бизнес-процессов к спецификациям ПО СИАП ЖЦ СОТО с реализацией ETL-принципа при описании порядка интеграции информационных ресурсов и решения связанных с ними аналитических задач.
4. Разработан алгоритм синтеза моделей решения аналитических вычислительных задач по СВЗ-моделям, позволяющий в автоматическом режиме осуществить переход от спецификаций требований к структурно-функциональным ВМ в виде G-моделей и синтезируемым на их основе схемам программ. Дополнены и обоснованы аспекты организации вычислений по схемам программ СИАП ЖЦ СОТО, связанные с исполнением онтологических запросов в рамках парадигмы потоковых вычислений.

3. КВАЛИМЕТРИЯ ПОЛИМОДЕЛЬНОГО КОМПЛЕКСА АВТОМАТИЗИРОВАННОЙ ИНФОРМАЦИОННО- АНАЛИТИЧЕСКОЙ ПОДДЕРЖКИ ЖИЗНЕННОГО ЦИКЛА КОСМИЧЕСКИХ СРЕДСТВ

3.1. Системный подход к оцениванию показателей качества моделей автоматизированной информационно-аналитической поддержки жизненного цикла космических средств как сложных объектов

Повышение эффективности информационно-аналитической поддержки ЖЦ СОТО очевидно связано с необходимостью повышения уровня автоматизации непосредственно процессов информационно-аналитической поддержки, поскольку в их рамках приходится обрабатывать существенные объемы зачастую неполной и неоднородной информации о ТС и надежности СОТО [216]. Такая информация может быть слабо структурирована, а отсутствие ИИС значительно снижает скорость обмена необходимой информацией, формируемой в удаленных источниках данных. При этом могут быть заданы довольно жесткие временные рамки на получение результатов обработки, а человеческий фактор может существенно снижать достоверность результатов обработки и без того не абсолютно достоверной информации. К методам, позволяющим повысить эффективность в рассматриваемой задаче на основе автоматизации можно отнести:

- использование ИИТ в виде *онтологической системы*, как способа декомпозиции процесса проектирования сложных ИАС, требующих привлечения множества специалистов на различных этапах проектирования, и обеспечения семантической машинной интерпертации связанного с ПрО тезауруса специалистов в виде совокупности уточняющих спецификаций с использованием декларативно-процедуральных моделей представления и извлечения знаний о структурных взаимосвязях терминов ПрО. При этом должна обеспечиваться «сквозная» согласованность таких моделей;

- использование методов системного моделирования и проектирования систем распределенных интеллектуальных агентов в ориентации на формирование архитектуры ПК, обеспечивающего реализуемость ЕИП;
- автоматический синтез схем программ анализа ТС СОТО, обеспечивающих корректные вычисления по сформированным специалистами ПрО моделям;
- использование парадигмы потоковых, асинхронных и параллельных вычислений при формализации моделей с целью обеспечения инвариантности состояния вычислительных процессов явлениям, реально происходящим в СОТО. При этом достигается повышение уровня информационно-аналитической поддержки ЖЦ СОТО.

Таким образом, необходимо говорить о системном подходе к оцениванию качества моделей и полимодельных комплексов, развиваемого в рамках теоретического направления, которое называют *квалиметрией* моделей [112, 184]. При этом применительно к моделям зачастую пользуются термином «*верификация*», понимая под системой верификации поддержку доказательства соответствия программной реализации и спецификации задачи, также состоящую в том, чтобы поставить в соответствие спецификации более высокого уровня спецификациям более низкого [73, 169]. В конечном счете целью верификации ставится обеспечение непротиворечивости и достоверности результатов обработки информации, которая в рассматриваемом случае представляется существенными сведениями о ТС СОТО в рамках хранилища электронного паспорта. В то же время при формировании МАС производительность агента оценивают по показателям полноты (гарантии, что алгоритм обнаружит решение, если оно имеется), оптимальности (наименьшей стоимости пути решения), временной сложности (времени нахождения решения) и пространственной сложности (объемов необходимой памяти) [169].

С учетом указанных аспектов рассмотрим способы верификации предлагаемого модельно-алгоритмического комплекса и порядок решения оптимизационной задачи повышения эффективности информационно-аналитической поддержки ЖЦ СОТО, представленной выражением (1.6) в разделе 1.4.3.

3.2. Алгоритмы проверки согласованности моделей бизнес-процессов и моделей согласованных вычислительных задач

Одним из аспектов повышения эффективности информационно-аналитической поддержки является проверка согласованности создаваемых спецификаций, как указывалось ранее. Например, в работе [179] отмечается необходимость согласования моделей полимодельного комплекса на глубинном модельно-алгоритмическом, информационном и программном уровнях. При этом отмечается, что для достижения такой согласованности необходимо положить в основу системы свойство функциональной преемственности (функториальности) моделей, или, по-другому, обеспечить *гомоморфизм* отношений, заданных в межмодельных связях. Для рассматриваемого модельно-алгоритмического комплекса формально под бинарным показателем *согласованности* будем понимать:

$$C^{cs} = C^f \wedge C^{exp}, \quad (3.1)$$

где

C^f – показатель *выполнимости* фактов спецификаций бизнес-процессов и СВЗ в БЗ $\mathcal{K}^{meta,t}$ при задании интерпретации относительно совокупности терминологических аксиом, заданных в разделах 2.2, 2.3, и тесно связанный с понятием разрешающего алгоритма в дескрипционных логиках;

C^{exp} – показатель *онтологической выразительности* спецификаций относительно друг друга (см. раздел 3.2.2).

Необходимо отметить, что отсутствует необходимость оценивания согласованности спецификаций источников данных, поскольку эти спецификации представляют собой совокупность несвязанных между собой схем Sk в соответствии с описанным в приложении 3. Аналогично отсутствует необходимость согласования СВЗ-моделей и получаемых по ним G-моделей и соответственно G-сетей, поскольку однозначный переход от одних к другим осуществляется на основе предложенного алгоритма (см. раздел 2.4.3) и грамматики в отличие от бизнес-процессов и СВЗ, переход между которыми осуществляется с участием специалистов, проектирующих эти модели.

3.2.1. Использование табло-алгоритма для проверки логической разрешимости онтологических высказываний

Фундаментальное значение для дескрипционных логик имеет так называемый tableau algorithm (фр., табло-алгоритм), который проверяет выполнимость БЗ в этой логике [72, 205, 249, 266]. Если в рамках некоторого формализма задан язык и его семантика, пытаются построить исчисление, выводящее высказывания в этом языке, которые являются «верными» в этой семантике:

- исчисление называется корректным, если для любой формулы φ из выводимости $\vdash \varphi$ следует верность $\models \varphi$;
- исчисление называется полным, если для любой формулы φ из $\models \varphi$ следует $\vdash \varphi$.

В дескрипционной логике вместо непосредственно исчисления задают алгоритм \mathfrak{A} , проверяющий, истинность аксиом или выполнимость концептов.

Определение 3.1. Табло-алгоритм \mathfrak{A} является разрешающим для проблемы выполнимости концептов относительно терминологий, если выполнены следующие три условия [169]:

- завершаемость: для любых (C, \mathcal{T}) алгоритм \mathfrak{A} выдает ответ $\mathfrak{A}(C, \mathcal{T})$ за конечное время;
- корректность: для любых (C, \mathcal{T}) если концепт C выполним относительно \mathcal{T} , то $\mathfrak{A}(C, \mathcal{T}) = 1$;
- полнота: для любых (C, \mathcal{T}) если $\mathfrak{A}(C, \mathcal{T}) = 1$, то концепт C выполним относительно \mathcal{T} .

Назначение алгоритма \mathfrak{A} состоит в том, чтобы показать, что \mathcal{A} имеет модель. Фактически сам \mathcal{A} и представляет модель, если он выполнен: индивиды концептов нужно считать элементами области интерпретации, а присутствующие в \mathcal{A} элементарные факты (вида xRy или $y: A$) считать заданием интерпретации атомарных концептов и ролей. При этом «табло» можно понимать как синтаксическую структуру, имитирующую некоторую модель (концепт, терминологию, БЗ, которые нужно проверить на выполнимость).

Продemonстрируем суть алгоритма на примере диалекта дескрипционной логики \mathcal{ALC} [72]. Пусть дан концепт C_0 , выполнимость которого требуется выяснить, и терминология \mathcal{T} , состоящая из единственной аксиомы $\top \sqsubseteq E$; требуется проверить, выполним ли C_0 относительно \mathcal{T} , т.е. существует ли интерпретация \mathcal{I} , в которой $E^{\mathcal{I}} = \Delta$ и $C_0^{\mathcal{I}} \neq \emptyset$. Пусть концепты C_0 и E нормализованы (то есть в них все отрицания поставлены лишь перед атомарными концептами, что может быть сделано с использованием законов де-Моргана [5]). Задается начальный $\mathcal{A}_0 = \{x_0: C_0\}$. Далее на каждом шаге к текущему \mathcal{A} применяется правило – и получается один или два новых \mathcal{A} . Список правил приведен в таблице 3.1 [72]. Порядок применения правил произволен. Правила для \sqcap, \exists, \forall из текущего \mathcal{A} создают один новый \mathcal{A}' , тогда как \sqcup -правило из \mathcal{A} создает два новых \mathcal{A}' и \mathcal{A}'' (и далее правила применяются к каждому из них «независимо»).

Таблица 3.1. Правила табло-алгоритма

Правило	Условия применения	Действие
\sqcap -правило	если 0. точка x – активная 1. $x: (C \sqcap D) \in \mathcal{A}$ 2. $x: C \notin \mathcal{A}$ или $x: D \notin \mathcal{A}$	то $\mathcal{A}' := \mathcal{A} \cup \{x: C, x: D\}$
\sqcup -правило	если 0. точка x – активная 1. $x: (C \sqcup D) \in \mathcal{A}$ 2. $x: C \notin \mathcal{A}$ и $x: D \notin \mathcal{A}$	то $\mathcal{A}' := \mathcal{A} \cup \{x: C\}$ $\mathcal{A}'' := \mathcal{A} \cup \{x: D\}$
\exists -правило	если 0. точка x – активная 1. $x: \exists R. C \in \mathcal{A}$ 2. нет такого y , что $xRy \in \mathcal{A}$ и $y: C \in \mathcal{A}$	то создать новую точку y и $\mathcal{A}' := \mathcal{A} \cup \{xRy, y: C\}$
\forall -правило	если 0. точка x – активная 1. $x: \forall R. C \in \mathcal{A}$ 2. есть такой y , что $xRy \in \mathcal{A}$ и $y: C \notin \mathcal{A}$	то $\mathcal{A}' := \mathcal{A} \cup \{y: C\}$
\mathcal{T} -правило	если 0. точка x – активная 1. $x: E \notin \mathcal{A}$	то $\mathcal{A}' := \mathcal{A} \cup \{x: E\}$

В результате по исходному \mathcal{A}_0 путем применения описанных монотонных правил будет построено дерево поиска, у которого в корне находится \mathcal{A}_0 и у каждого \mathcal{A} есть 0, 1 или 2 последователя. Применение правил прекращается, если к очередному \mathcal{A} не применимо ни одно из правил, либо если в \mathcal{A}

содержится явное противоречие, так что применять дальнейшие правила не имеет смысла. Эти два вида \mathcal{A} будут листьями дерева поиска, причем:

- \mathcal{A} называется *противоречивым*, если он содержит факты $x:A$ и $x:\neg A$ для некоторого индивида x и атомарного концепта A , либо он содержит факт $x:\perp$ для некоторого индивида x ;
- \mathcal{A} называется *полным непротиворечивым*, если он не является противоречивым, но ни одно из правил табло-алгоритма к нему не применимо.

Если алгоритм встречается полный непротиворечивый \mathcal{A} , то он выдает ответ «1» и оставшиеся ветви дерева поиска уже не обходит. Если алгоритм встречается противоречивый \mathcal{A} , то алгоритм продолжает строить остальные ветви дерева поиска. Наконец, когда алгоритм обошел всё дерево поиска и оказалось, что все листовые \mathcal{A} противоречивы, алгоритм выдает ответ «0» (в этом случае моделей нет). Табло-алгоритм можно описать в виде рекурсивной процедуры $SAT(\mathcal{A})$, описанной в таблице 3.2 [72]. Она принимает на вход произвольный \mathcal{A} (с условием, что все концепты в нем нормализованы) и выдает ответ «0» или «1».

Таблица 3.2. Табло-алгоритм, описанный на мнемоязыке

Функция $SAT(\mathcal{A})$ { если в \mathcal{A} есть $x:\perp$ для некоторого x , то return 0; если в \mathcal{A} есть $x:A$ и $x:\neg A$ для некоторых x и A , то return 0; если к \mathcal{A} не применимо ни одно из правил, то return 1; если применимо правило Π , \exists или \forall , то применить любое, получив \mathcal{A}' , и return $SAT(\mathcal{A}')$; если применимо \sqcup -правило, то применить его, получив \mathcal{A}' и \mathcal{A}'' , и далее: { если $SAT(\mathcal{A}') = 1$, то return 1; если $SAT(\mathcal{A}'') = 1$, то return 1; } return 0; }
--

Необходимо заметить, что \mathcal{A} можно рассматривать как размеченный граф (G, L) . Вершинами графа G являются индивиды x , встречающиеся в \mathcal{A} ; ребра соответствуют имеющимся в \mathcal{A} фактам вида xRy ; меткой $L(x)$ вершины x является множество таких концептов C , что в \mathcal{A} имеется факт $x:C$, то есть $L(x) = \{C \mid x:C \in \mathcal{A}\}$. Этот граф фактически представляет собой искомую модель, для поиска которой и предназначен табло-алгоритм. В ориентированном графе вершина x называется предком вершины y , а вершина y – потомком

вершины x , если из x в y ведет ориентированный путь. Точка x блокирует точку y , если x есть предок y и $L(x) \supseteq L(y)$. Это правило необходимо для обеспечения завершаемости табло-алгоритма с терминологией. Блокированные точки называют неактивными, остальные – активными. Для алгоритма, учитывающего терминологию также необходимо к каждому правилу добавить дополнительное предусловие «точка x – активная», а также ввести новое правило, добавляющее концепт E в каждую точку. Необходимо добавить упоминание \mathcal{T} -правила, которое аналогично создает один новый \mathcal{A} при проверке применимости терминологических аксиом к текущей точке дерева поиска.

Граф G является деревом с корнем x_0 , а новые вершины порождаются лишь \exists -правилом, которое соединяет вновь созданную вершину ровно с одной из уже существующих вершин. Как следствие, из корня x_0 в любую вершину $x \neq x_0$ ведет единственная цепочка ребер, называемая уровнем вершины x . Ширина дерева G ограничена количеством кванторов \exists в исходном концепте. Глубина дерева G ограничена кванторной сложностью концепта C_0 . Это следует из того, что кванторная сложность концептов в метках вершин уменьшается с увеличением уровня вершины: если на каком-то уровне были концепты вида $\exists R. C$ и $\forall R. C$, то эти правила на следующий уровень переносят лишь концепт C (сложности на 1 меньшей).

В результате может быть построена модель \mathcal{I} для \mathcal{A} , откуда следует, что \mathcal{A} выполним. Если \mathcal{A} выполним и относительно \mathcal{T} , то БЗ $\mathcal{K} = (\mathcal{T} \cup \mathcal{A})$ выполняема.

Дерево поиска можно схематично изобразить элементами, где символом « \circ » обозначен произвольный \mathcal{A} , « \boxtimes » – противоречивый \mathcal{A} , « \square » – полный непротиворечивый \mathcal{A} (см. рис. 3.1).

Замечание 3.1. Представленный табло-алгоритм может в некоторой степени отличаться набором предусловий для правил в зависимости от выбранного диалекта дескрипционной логики. Так, например, с учетом рассмотрения ограничений кардинальности ролей вида $\leq n$ ($\geq n$) в терминологических аксиомах, правила могут быть дополнены проверкой условия, что в \mathcal{A} не более

(не менее) n R-последователей, т.е. при создании новой точки u в дереве поиска необходимо убедиться, что таких точек не более (не менее) n .

Замечание 3.2. Табло-алгоритм может быть использован для проверки того, является ли модель бизнес-процессов (также как и СВЗ-модель) выполнимой относительно терминологий, представленных в разделах соответственно 2.2.2, 2.3.2.

Это существенное свойство позволяет осуществлять проверку корректности формируемых систем фактов $\mathcal{A}^{meta,t}$ (утверждениях о моделях pr и sf) для того, чтобы убедиться в том, что эти факты удовлетворяют ограничениям, накладываемым на структуру таких моделей, т.е. что в $\mathcal{A}^{meta,t}$ имеется интерпретация для $\mathcal{T}^{meta,t}$. Соответствие заданным ограничениям в последующем позволит осуществлять проверку согласованности моделей pr и sf , как это будет показано в разделе 3.2.2. Продемонстрируем на примере проверку выполнимости \mathcal{A}^{meta} для модели бизнес-процессов.

Пример 3.1. Пусть имеется модель бизнес-процесса из примера 2.6. Рассмотрим для наглядности случай, когда эта модель представлена единственным процессом $process_2$. Таким образом, модель бизнес-процесса представлена следующей системой фактов:

$lc_1: LifeCycle$	$a_4: Action$	$a_4 Follow e_6$
$process_2: Process$	$e_6 Defines process_2$	$e_7 Follow a_4$
$process_2 Defines lc_1$	$e_7 Defines process_2$	
$e_6: StartEvent$	$a_4 Defines process_2$	
$e_7: FinalEvent$		

При проверке выполнимости фактов будут необходимы терминологические аксиомы \mathcal{T}^{meta} из выражений (2.10) – (2.15). Проверка выполнимости состоит в последовательной проверке фактов из \mathcal{A}^{meta} и построении дерева поиска для каждого из них. Шаги работы алгоритма представлены в таблице 3.3. Столбцы таблицы слева направо по порядку обозначают следующее: шаг работы алгоритма (увеличивается на 1 при переходе на новый факт \mathcal{A}^{meta} , уве-

личивается на 0.1 при обнаружении применимости правила из 2.2.2, 2.3.2), создаваемая система фактов \mathcal{A}_i на каждом шаге алгоритма, новые факты, подлежащие проверке, найденное правило, которое может быть применено к факту.

Таблица 3.3. Пример работы табло-алгоритма при проверке выполнимости фактов, описывающих модель бизнес-процесса

Шаг №	Сис-ма фактов	Факты	Применяемое правило
1	\mathcal{A}_0	$lc_1: LifeCycle$	– \mathcal{A}_0 полный непротиворечивый
2	\mathcal{A}_0	$process_2: Process$	\mathcal{T} -правило: (2.1) $Process \sqsubseteq \leq 1 Defines. LifeCycle$
2.1	\mathcal{A}_1	$process_2: \leq 1 Defines. LifeCycle$	\exists -правило: (2.2)
2.2	\mathcal{A}_1	–	– Имеется $process_2 Defines lc_1$ \mathcal{A}_1 полный непротиворечивый
3	\mathcal{A}_0	$e_6: StartEvent$	\mathcal{T} -правило: (3.1) $StartEvent \sqsubseteq Event$
3.1	\mathcal{A}_1	$e_6: Event$	\mathcal{T} -правило: (3.2) $Event \sqsubseteq \leq 1 Defines. Process$ $Event \equiv (StartEvent \sqcap \neg InterEvent \sqcap \neg FinalEvent) \sqcup (\neg StartEvent \sqcap InterEvent \sqcap \neg FinalEvent) \sqcup (\neg StartEvent \sqcap \neg InterEvent \sqcap FinalEvent) \sqcap \leq 1 Follow. Action \sqcap \neg (\exists Follow. Condition) \sqcap \neg (\exists Follow. Event)$
3.2	\mathcal{A}_2	$e_6: (StartEvent \sqcap \neg InterEvent \sqcap \neg FinalEvent) \sqcup (\neg StartEvent \sqcap InterEvent \sqcap \neg FinalEvent) \sqcup (\neg StartEvent \sqcap \neg InterEvent \sqcap FinalEvent) \sqcap \leq 1 Follow. Action \sqcap \neg (\exists Follow. Condition) \sqcap \neg (\exists Follow. Event)$	Имеется $e_6 Defines process_2$ \sqcup -правило: (3.3, 3.4, 3.5)
3.3	\mathcal{A}_3	$e_6: (StartEvent \sqcap \neg InterEvent \sqcap \neg FinalEvent) \sqcup (\neg StartEvent \sqcap InterEvent \sqcap \neg FinalEvent) \sqcup (\neg StartEvent \sqcap \neg InterEvent \sqcap FinalEvent) \sqcap \leq 1 Follow. Action \sqcap \exists Follow. \neg Condition$	\sqcap -правило: (3.6)
3.4	\mathcal{A}_4	$e_6: (\neg StartEvent \sqcap InterEvent \sqcap \neg FinalEvent) \sqcup (\neg StartEvent \sqcap \neg InterEvent \sqcap FinalEvent) \sqcap \leq 1 Follow. Action \sqcap \exists Follow. \neg Condition$	\sqcap -правило: (3.7)
3.5	\mathcal{A}_5	$e_6: (\neg StartEvent \sqcap \neg InterEvent \sqcap FinalEvent) \sqcap \leq 1 Follow. Action \sqcap \exists Follow. \neg Condition$	\sqcap -правило: (3.8)
3.6	\mathcal{A}_6	$e_6: StartEvent$ $e_6: \neg InterEvent$ $e_6: \neg FinalEvent$ $e_6: \leq 1 Follow. Action$	\exists -правило: (3.9) $e_6: \leq 1 Follow. Action$ Создать новую точку x_1 \exists -правило: (3.9)

		$e_6: \exists Follow. \neg Condition$	$e_6: \exists Follow. \neg Condition$ Та же точка x_1 (связь по П)
3.7	\mathcal{A}_7	$e_6: \neg StartEvent$ $e_6: InterEvent$ $e_6: \neg FinalEvent$ $e_6: \leq 1 Follow. Action$ $e_6: \exists Follow. \neg Condition$	\mathcal{A}_7 противоречивый: $e_6: \neg StartEvent$ $e_6: StartEvent$
3.8	\mathcal{A}_8	$e_6: \neg StartEvent$ $e_6: \neg InterEvent$ $e_6: FinalEvent$ $e_6: \leq 1 Follow. Action$ $e_6: \exists Follow. \neg Condition$	\mathcal{A}_8 противоречивый: $e_6: \neg StartEvent$ $e_6: StartEvent$
3.9	\mathcal{A}_9	$x_1: Action$ $x_1: \neg Condition$ $e_6 Follow x_1$	\mathcal{T} -правило: (3.10) $Action \sqsubseteq \leq 1 Defines. Process$ \mathcal{T} -правило: (3.10) $Action \equiv (\exists Follow. Event \sqcup \exists Follow. Condition) \sqcap$ $\neg(\exists Follow. Action) \sqcap \exists Follow^-. Event$
3.10	\mathcal{A}_{10}	$x_1: \leq 1 Defines. Process$ $x_1: (\exists Follow. Event$ $\sqcup \exists Follow. Condition)$ $\sqcap \neg(\exists Follow. Action)$ $\sqcap \exists Follow^-. Event$	\exists -правило: (3.11) $x_1: \leq 1 Defines. Process$ Создать новую точку x_2
3.11	\mathcal{A}_{11}	$x_2: Process$ $x_1 Defines x_2$	\mathcal{T} -правило (3.12): $Process \sqsubseteq \leq 1 Defines. LifeCycle$ Создать новую точку x_3
3.12	\mathcal{A}_{12}	$x_3: LifeCycle$ $x_2 Defines x_3$	– \mathcal{A}_{12} полный непротиворечивый
4	\mathcal{A}_0	$e_7: FinalEvent$	По аналогии с (3) до шага (3.5). Переход на шаги (4.6, 4.7, 4.8)
...
4.6	\mathcal{A}_6	$e_7: StartEvent$ $e_7: \neg InterEvent$ $e_7: \neg FinalEvent$ $e_7: \leq 1 Follow. Action$ $e_7: \exists Follow. \neg Condition$	\mathcal{A}_7 противоречивый: $e_7: InterEvent$ $e_7: \neg InterEvent$
4.7	\mathcal{A}_7	$e_7: \neg StartEvent$ $e_7: InterEvent$ $e_7: \neg FinalEvent$ $e_7: \leq 1 Follow. Action$ $e_7: \exists Follow. \neg Condition$	\exists -правило: (4.9) $e_7: \leq 1 Follow. Action$ \exists -правило: (4.9) $e_7: \exists Follow. \neg Condition$ Та же точка (связь по П)
4.8	\mathcal{A}_8	$e_7: \neg StartEvent$ $e_7: \neg InterEvent$ $e_7: FinalEvent$ $e_7: \leq 1 Follow. Action$ $e_7: \exists Follow. \neg Condition$	\mathcal{A}_7 противоречивый: $e_7: InterEvent$ $e_7: \neg InterEvent$
4.9	\mathcal{A}_7	$a_4: \neg Condition$	– Имеется: $a_4: Action$ $e_7 Follow a_4$ \mathcal{A}_7 полный непротиворечивый
5	\mathcal{A}_0	$a_4: Action$	\mathcal{T} -правило: (5.1) $Action \sqsubseteq \leq 1 Defines. Process$ \mathcal{T} -правило: (-) $Action \equiv (\exists Follow. Event \sqcup \exists Follow. Condition) \sqcap$ $\neg(\exists Follow. Action) \sqcap \exists Follow^-. Event$
5.1	\mathcal{A}_1	$a_4: (\exists Follow. Event$ $\sqcup \exists Follow. Condition)$ $\sqcap \neg(\exists Follow. Action)$ $\sqcap \exists Follow^-. Event$	Имеется $a_4: Defines process_2$ \sqcup -правило: (5.2, 5.3)
5.2	\mathcal{A}_2	$a_4: \exists Follow. Event$	\sqcup -правило: (5.3)

		$\Pi \exists Follow. \neg Action$ $\Pi \exists Follow^-. Event$	
5.3	\mathcal{A}_3	$a_4: \exists Follow. Event$ $a_4: \exists Follow. \neg Action$ $a_4: \exists Follow^-. Event$	\exists -правило: (5.4) $a_4: \leq 1 Follow. Event$ \exists -правило: (5.4) $a_4: \exists Follow. \neg Action$ Та же точка (связь по Π) \exists -правило: (5.5) $a_4: \exists Follow^-. Event$ (обратная роль)
5.4	\mathcal{A}_3	$e_6: \neg Action$	– Имеется $a_4 Follow e_6$
5.5	\mathcal{A}_4	–	– Имеется $e_7 Follow a_4$ \mathcal{A}_4 полный непротиворечивый

Из примера 3.1 видно, что система фактов \mathcal{A}^{meta} имеет интерпретацию в \mathcal{T}^{meta} , т.е. имеет модель, которая и является моделью бизнес-процесса $process_2$. Шаги алгоритма в пунктах 1 и 2 очевидны. На этапе 3 алгоритм осуществляет проверку выполнимости выражений \mathcal{T}^{meta} при наличии индивида $e_6: StartEvent$ в \mathcal{A}^{meta} . Можно заметить, что в соответствии с терминологическими аксиомами событию должно предшествовать действие или условие. В связи с этим на шагах 3.6, 3.10, 3.11 создаются дополнительные условные индивиды действия, процесса и ЖЦ соответственно. В результате выражения оказываются выполнимыми при $e_6: StartEvent$. Аналогичным образом на этапах 4 и 5 проверяются выражения \mathcal{T}^{meta} при $e_7: FinalEvent$, $a_4: Action$. На рисунке 3.1 представлено формируемое табло-алгоритмом дерево поиска при проверке $e_6: StartEvent$.

Необходимо также заметить, что аналогичным образом могут быть проверены СВЗ-модели.

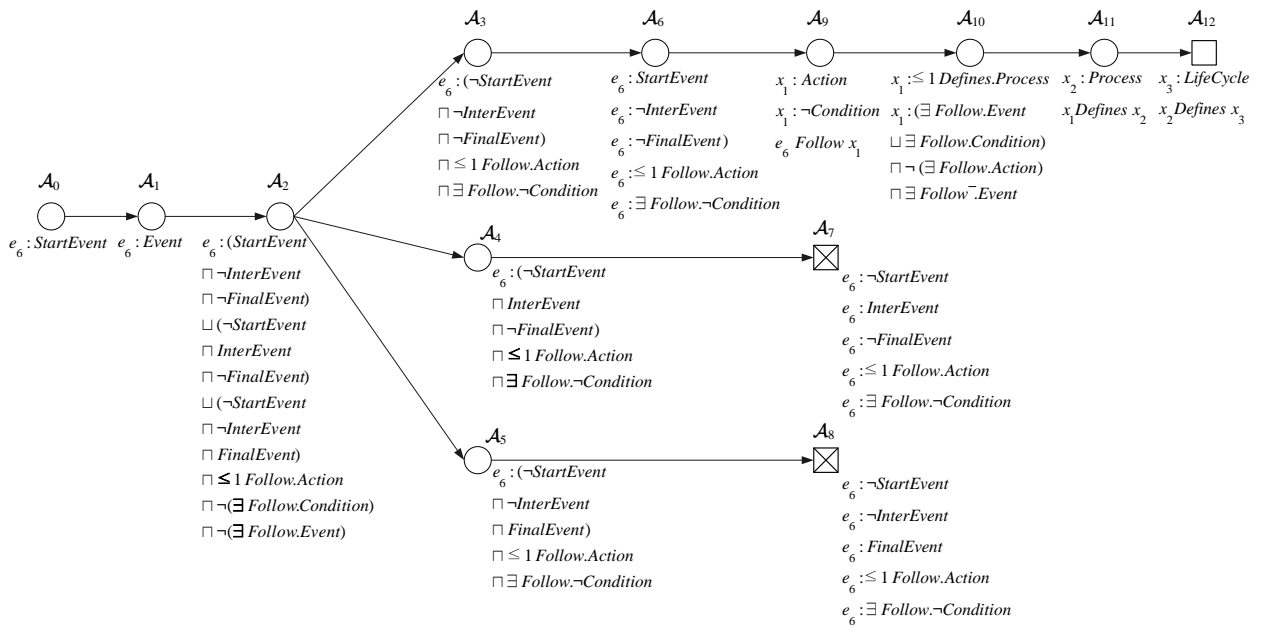


Рис. 3.1. Дерево поиска табло-алгоритма для примера модели бизнес-процесса

3.2.2. Использование алгоритма проверки онтологической выразительности для оценивания согласованности спецификаций

В работах [198, 199, 200, 234] рассматриваются понятия *онтологической чистоты* (англ., clarity) и *выразительности* (англ., expressiveness) языка моделирования. Модель, в которой отсутствует неразличимость, избыточность или неоднозначность, называют онтологически чистой. Если в дополнение к ясности отсутствует дефицит, то модель считается выразительной (см. рис. 3.2). Я. Ванд и Р. Вебер предположили, что онтологическую выразительность языка или нотации моделирования можно оценить путем сравнения с конструкциями предлагаемой ими онтологии. Одним из главных факторов успеха использования языка является его способность предоставить пользователям набор знаков (примитивов моделирования), которые могут непосредственно выражать соответствующие концепты (абстракции) Про.

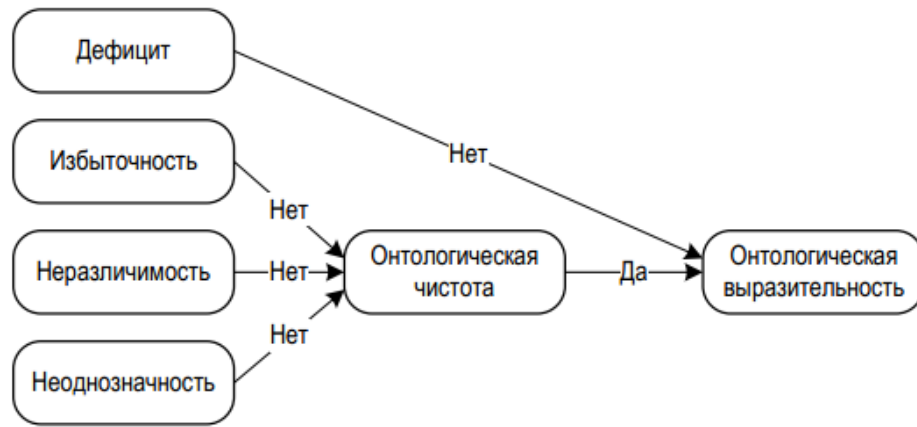


Рис. 3.2. Взаимосвязь показателей онтологической выразительности

Взаимосвязь сущностей реального мира, концептов онтологии и примитивов моделирования по аналогии с треугольником Г. Фреге (см. рис. 2.1) можно описать следующими выражениями:

- сущность реального мира абстрагирует концепт онтологии;
- концепт онтологии отражает примитив языка моделирования;
- примитив языка моделирования соответствует сущности реального мира.

В таком случае можно говорить о следующих качественных показателях, отражающих связи между указанными элементами (см. рис. 3.3) [198, 199, 200]:

- каждому знаку языка можно сопоставить ровно один концепт онтологии — можно говорить о взаимнооднозначном соответствии двух множеств или об их *эквивалентности*;
- один концепт может быть отображен сразу на несколько знаков нотации — имеет место *неразличимость* знаков языка моделирования;
- отдельные концепты онтологии не имеют соответствующего им знака — существует *дефицит* выразительности языка;
- ни один концепт не может быть отображен в примитив нотации — имеется *избыточность* языка;
- одному знаку соответствует несколько концептов — в наличии *неоднозначность* нотации.

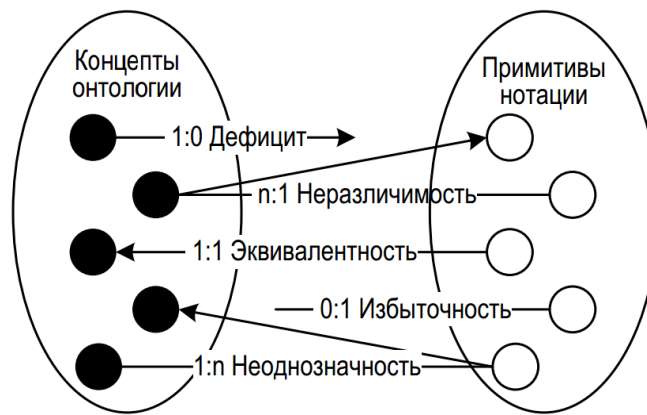


Рис. 3.3. Показатели онтологической выразительности, отражающие связь онтологии и примитивы нотации моделирования

Замечание 3.3. Показатели онтологической выразительности могут быть использованы при оценивании согласованности спецификаций бизнес-процессов и СВЗ информационно-аналитической поддержки ЖЦ СОТО, поскольку в предлагаемом подходе примитивы языков моделирования определены непосредственно в рамках онтологий O^{meta} и O^t .

Существенно, что оценивание онтологической выразительности для традиционных языков моделирования осуществляется методом экспертной оценки, в то время как в рассматриваемом случае этот процесс может быть автоматизирован, и соответственно, может быть разработан алгоритм проверки онтологической выразительности.

В рассматриваемом случае понятия, связанные с онтологической выразительностью, могут быть расширены и приобретают особую интерпретацию. Их можно понимать и использовать двояко (см. также разделы 2.2.2, 2.3.2):

- с одной стороны, при задании спецификации бизнес-процессов даже при условии выполнимости концептов (наличия модели в \mathcal{A}^{meta} и успешного завершения работы табло-алгоритма) такая модель может оставаться некорректной в смысле правил нотации BPMN, поскольку в этой спецификации могут отсутствовать необходимые элементы (например, исходные или целевые события);

– с другой стороны, несмотря на явное задание в рамках терминологических аксиом $\mathcal{T}^{t,def}$ однозначных связей между концептами онтологий O^{meta} и O^t , формируемая в рамках \mathcal{A}^t СВЗ-модель sf_v может не соответствовать по своему составу и по структуре модели связанного с ней бизнес-процесса pr_v . Т.е. могут отсутствовать необходимые факты в $\mathcal{A}^{t,sf}$ или $\mathcal{A}^{t,def}$ даже при условии выполнимости концептов, так что это приведет к неполному соответствию одной модели другой. Необходимо также отметить, что этот случай может привести к отсутствию разрешимости соответствующей G-модели m_v^G .

Таким образом, под *онтологической выразительностью* можно понимать выразительность модели бизнес-процесса относительно нотации BPMN (правила которой и определены в рамках терминологических аксиом \mathcal{T}^{meta}) и соответствие СВЗ-модели модели бизнес-процесса, так что каждому индивиду из O^{meta} может быть сопоставлен индивид в O^t , и при этом сохраняется и структура модели.

На рисунке 3.4 приведены типовые примеры наличия дефицита, избыточности и неразличимости между спецификациями бизнес-процессов и СВЗ.

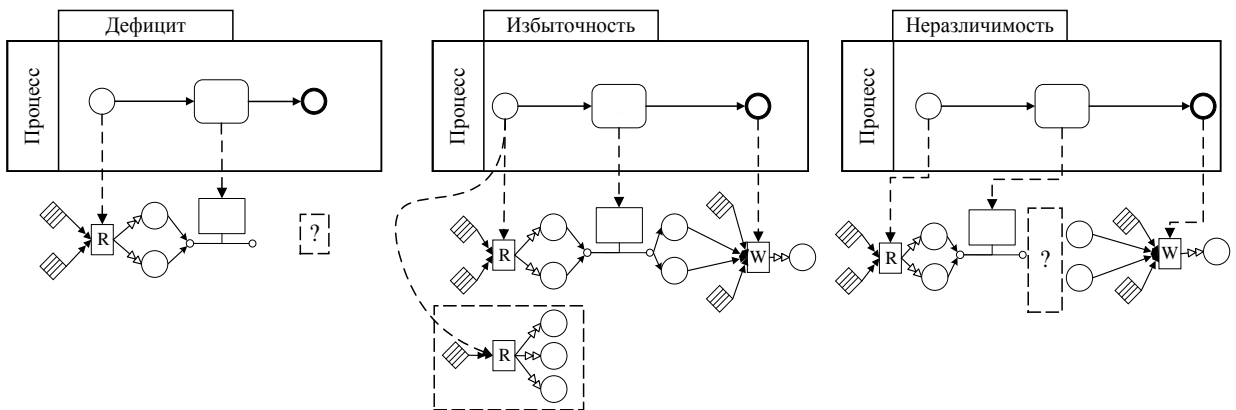


Рис. 3.4. Примеры наличия дефицита, избыточности и неразличимости несогласованных спецификаций

С учетом рассмотренного выше, при оценивании показателя онтологической выразительности будем считать, что он рассчитывается следующим образом:

$$C^{exp} = \overline{C^{деф}} \wedge \overline{C^{неразл}} \wedge C^{экв} \wedge \overline{C^{изб}} \wedge \overline{C^{неодн}}. \quad (3.2)$$

Утверждение 3.1. Оценивание показателей эквивалентности $C^{экв}$, неоднозначности $C^{неодн}$ и избыточности $C^{изб}$ не требует дополнительных проверок кроме тех, которые обеспечивает непосредственно табло-алгоритм, определённый в разделе 3.2.1. Этот факт вытекает из того, как определены терминологические аксиомы в разделах 2.2, 2.3, связанные с заданием попарных связей концептов онтологий O^{meta} и O^t и использованием ограничений кардинальности ролей (« ≤ 1 »), что обеспечивает однозначный переход от одной спецификации к другой, а в случае некорректного задания фактов в $\mathcal{A}^{t,def}$ немедленно приведет к противоречию в дереве поиска табло-алгоритма.

Однако необходимо также обратить внимание на тот факт, что на практике специалист, проектирующий модель бизнес-процессов и СВЗ-модель по модели бизнес-процессов, может фактически не задать необходимые связи между элементами, т.е. не внести ряд необходимых фактов в $\mathcal{A}^{meta,t}$. В связи с этим необходимо осуществлять дополнительную проверку после этапа проектирования на отсутствие дефицита (отсутствия элементов) и неразличимости (отсутствия связей). Для этой цели был разработан алгоритм, представленный на рисунках 3.5, 3.6. Вычисление по этому алгоритму предполагает отсутствие неоднозначности и избыточности согласуемых спецификаций и эквивалентность их элементов исходя из утверждения 3.1.

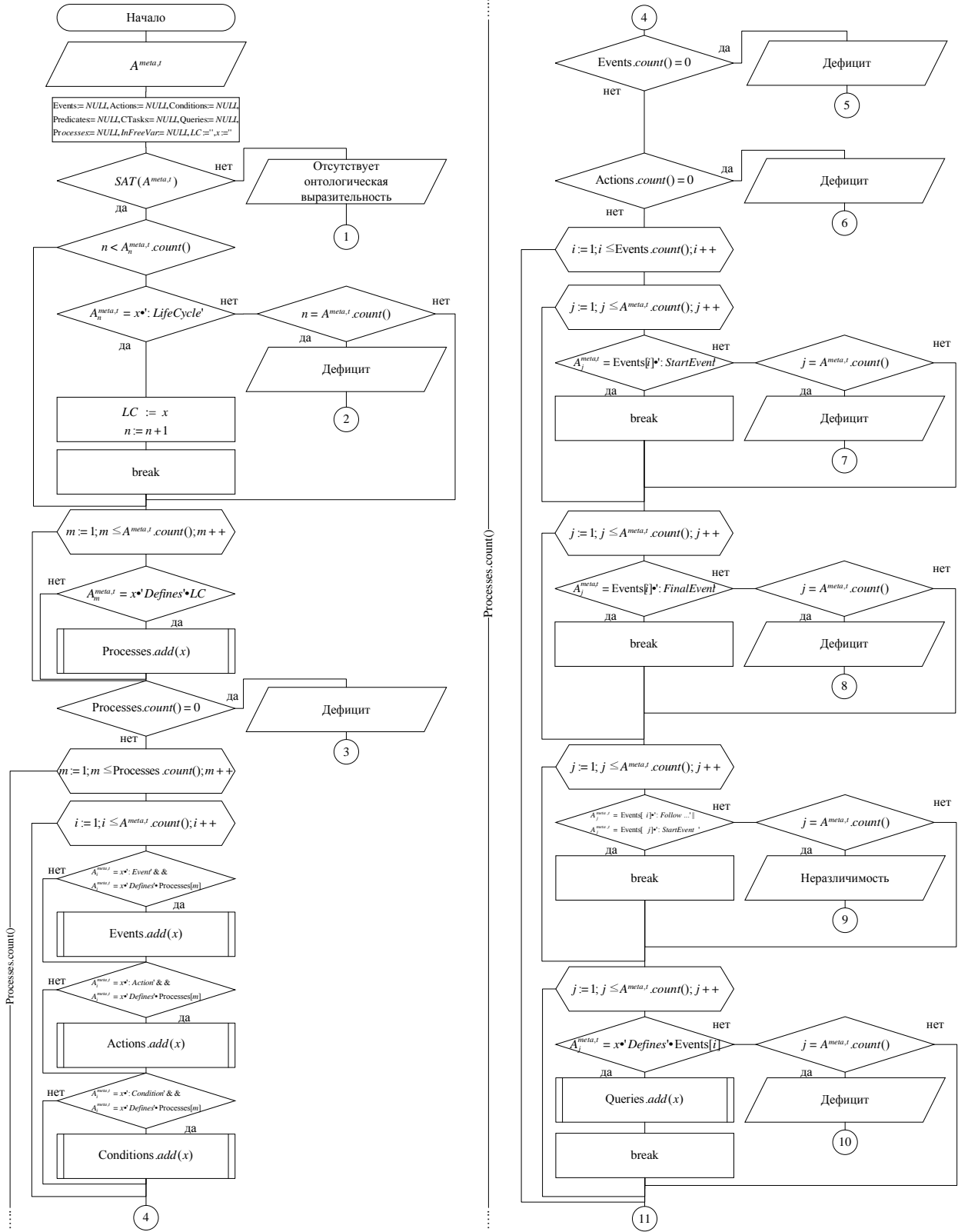


Рис. 3.5. Блок-схема алгоритма проверки онтологической выразительности (часть 1)

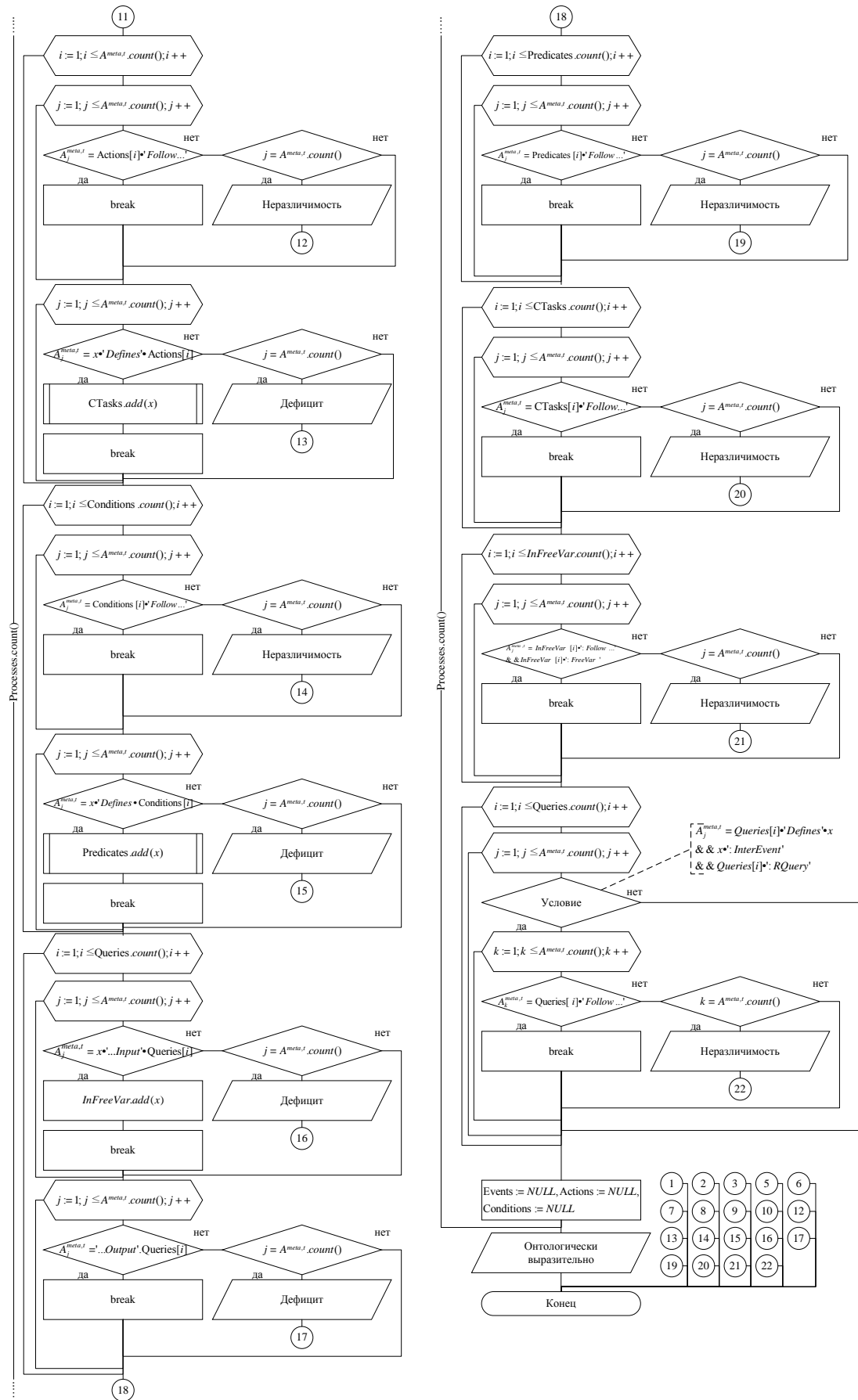


Рис. 3.6. Блок-схема алгоритма проверки онтологической выразительности (часть 2)

Поясним суть работы алгоритма. На входе алгоритма имеется система фактов $\mathcal{A}^{meta,t}$, в которой определены спецификации бизнес-процессов и СВЗ. Прежде всего, как было отмечено ранее, $\mathcal{A}^{meta,t}$ проверяется табло-алгоритмом с целью оценивания выполнимости концептов терминологии $\mathcal{T}^{meta,t}$. При завершении работы табло-алгоритма с результатом «1» можно говорить об отсутствии неоднозначности и избыточности согласуемых спецификаций и эквивалентности их элементов. В рассматриваемом алгоритме объявляются векторы *Events*, *Actions*, *Conditions*, *Predicates*, *Ctasks*, *Queries*, *Processes*, *InFreeVar* для того, чтобы запомнить соответственно индивиды событий, действий, условий, предикатов, вычислительных задач, запросов, процессов и входных элементов запросов, найденных в $\mathcal{A}^{meta,t}$. В переменной *LC* запоминается найденный в $\mathcal{A}^{meta,t}$ индивид ЖЦ. Переменная *x* используется для нахождения и временного хранения необходимых индивидов при прохождении по набору фактов $\mathcal{A}^{meta,t}$. Символом « \cdot » обозначена операция конкатенации. Процедура *add* обозначает добавление элемента в вектор. Процедура *count* возвращает количество фактов в $\mathcal{A}^{meta,t}$. Оператор *break* прерывает выполнение соответствующих циклов прохождения по набору фактов $\mathcal{A}^{meta,t}$. В рамках алгоритма рассмотрены следующие ситуации, характеризующие дефицит в спецификации:

- отсутствие индивида ЖЦ, без которого невозможно определение пула модели бизнес-процессов;
- отсутствие индивидов процессов, без которых в спецификации не определены «дорожки» (см. раздел 2.2.2);
- отсутствие индивидов событий и действий, без которых спецификация не определяет никакой процесс;
- отсутствие индивидов исходного и целевого события;
- отсутствие индивидов запросов в СВЗ, определяющего индивиды событий бизнес-процесса;

- отсутствие индивидов вычислительных задач в СВЗ, определяющих индивиды действий бизнес-процесса;
- отсутствие индивидов предикатов в СВЗ, определяющих индивиды условий бизнес-процесса;
- отсутствие индивидов конъюнктов и переменных запросов.

Ситуации, связанные с неразличимостью требуют дополнительного пояснения. Необходимо отметить, что проверка наличия связей между элементами спецификаций частично осуществляется табло-алгоритмом на основе того, что концепты в терминологических аксиомах в БЗ $\mathcal{K}^{meta,t}$ определены друг через друга. В результате, в полном непротиворечивом $\mathcal{A}^{meta,t}$ каждый элемент либо не будет связан должным образом с другими, либо однозначно будет связан корректно. Так, например, концепт вычислительной задачи, как один из наиболее сложных случаев в выражении (2.34), определяется как элемент, который:

- определяет действие;
- предшествует переменным, входным для запросов на запись, определяющим события, следующими за действием;
- следует за свободной переменной запроса, являющейся выходной для запроса, определяющего событие, предшествующее рассматриваемой вычислительной задаче. При этом переменная запроса:
 - а) не предшествует никаким предикатам;
 - б) предшествует предикату.

Это определение проиллюстрировано в виде обобщенной спецификации вычислительной задачи на рисунке 3.7 для случаев без предиката (а), с предикатами (б).

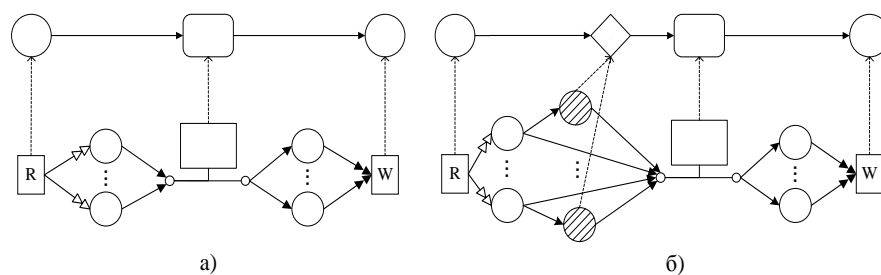


Рис. 3.7. Иллюстрация обобщенной спецификации вычислительной задачи

Замечание 3.4. Определения концептов в онтологических МПЗ о бизнес-процессах и СВЗ позволяют утверждать отсутствие неразличимости для любого элемента спецификации, убедившись лишь в том, что он следует (*Follow*) за любым другим элементом, а в случае СВЗ также определяет (*Defines*) что-либо. Эта ситуация проиллюстрирована на рисунке 3.8.

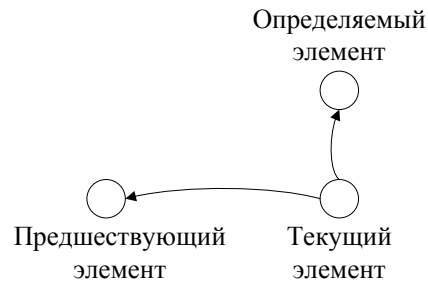


Рис. 3.8. Иллюстрация условий отсутствия неразличимости

В таком случае, в алгоритме на этапах проверки отсутствия неразличимости необходимо и достаточно убедиться в том, что:

- индивид события бизнес-процесса следует за чем-либо, либо является исходным;
- индивид действия следует за чем-либо;
- индивид условия следует за чем-либо (здесь необходимо отметить, что в зависимости от типа условия из выражений (2.17) – (2.21) количество предшествующих элементов существенно);
- индивид предиката следует за чем-либо (по аналогии с предыдущим пунктом количество связей существенно);
- индивид вычислительной задачи следует за чем-либо;
- входные переменные запросов записи следуют за чем-либо;
- запросы выборки, определяющие промежуточные события, следуют за чем-либо.

Таким образом, при соблюдении всех указанных условий созданные спецификации являются онтологически выразительными, а вычислительные модели, формируемые с помощью алгоритма их синтеза (см. раздел 2.4.2) по СВЗ-моделям, определенным в $\mathcal{A}^{meta,t}$, являются разрешимыми.

3.3. Верификация схем программ информационно-аналитической поддержки жизненного цикла космических средств

Еще одним важным аспектом, полностью описанным в [132], является верификация схем программ информационно-аналитической поддержки ЖЦ СОТО в виде получаемых G-сетей. В соответствии с предложенными в работе аспектами квалиметрии G-сети будем её называть *полной корректной*, если принимает значение «1» выражение:

$$V^S = w_1 V^{\text{непр}} + w_2 V^{\text{корр}} + w_3 V^{\text{акт}}, \quad (3.3)$$

где

$V^{\text{непр}}$ – показатель *непротиворечивости* схемы программы;

$V^{\text{корр}}$ – показатель *корректности* схемы программы;

$V^{\text{акт}}$ – показатель *активности* схемы программы;

w_1, w_2, w_3 – веса, характеризующие существенность частных показателей и задаваемые экспертным опросом [103].

Необходимо заметить, что для схем программ информационно-аналитической поддержки ЖЦ СОТО их верификация не отличается от той процедуры, которая описана в [132]. Однако для полноты описания процесса верификации рассмотрим содержательно суть каждого из показателей более подробно.

3.3.1. Непротиворечивость схем программ

Поскольку G-сеть может быть выполнена тогда, когда могут быть выполнены все её переходы, то выполнимость элементарной G-сети (по аналогии с G-моделью) S_j^G , состоящей из единственного перехода $v_j \in V_S$, определяется его выполнимостью. P-переходы $v_j \in V_S^P$ выполняются (с занесением *True* или *False* в свою выходную позицию) лишь тогда, когда значение соответствующей предикатной функции:

$$\varphi_j^P(x) = \text{"True"}. \quad (3.4)$$

При этом множество значений предметных переменных предикатной функции для соответствующего предикатного перехода $v_j \in V_S^P$, при которых

выполняется условие (3.4), определяет область применимости D_{v_j} перехода. Если схема анализа S_G формируется из нескольких элементарных G-сетей, то её выполнимость ограничивается входящими в неё предикатными переходами $v_j \in V_S$, определяющими область применимости D_{S_G} на множестве значений тех атрибутов $x \in X_S$, значения которых формируют истинностные значения всех предикатных функций $p_j(x) = \varphi_j^P(x) \in F_S^P$, входящих в состав сети S_G , для которой выполняется условие:

$$\bigg\&_{j \in I_V^P} [\varphi_j^P(x)] = "True", \quad (3.5)$$

или для предикатных атрибутов:

$$\bigg\&_{k \in I_X^P} [x_k^P = "True"], \quad (3.6)$$

где I_V^P, I_X^P – множества индексов переходов V_S и атрибутов X_S сети S_G соответственно.

Таким образом, выполнимость G-сети определяется предикатами, соответствующими переходам сети. Для того, чтобы сеть могла выполняться, необходимо, чтобы существовала такая область D_{S_G} , для которой верны выражения (3.5), (3.6). Т.е., сеть S_G является выполнимой, если она имеет непустую область применимости $D_{S_G} \neq \{\emptyset\}$.

Указанное свойство G-сетей позволяет обнаруживать возможные ошибки в синтезированной программе.

Случай, когда выражения (3.5), (3.6) принимают значения «False» вследствие того, что пара предикатов p_{j_1} и p_{j_2} на одном и том же подмножестве $D_{S_G^i}$ принимают значения и «истина», и «ложь», приводит к противоречивости G-сети.

Определение 3.2. G-сеть S_G является *непротиворечивой*, когда существует такое непустое множество значений атрибутов D_X , на котором эта сеть является выполнимой [132].

Проверку непротиворечивости G-сети можно осуществлять приведением множества предикатов к дизъюнктивно-нормальной форме (ДНФ):

$$p_j(x) = \bigvee_{i \in I_V} \bigwedge_{i \in I_\wedge} (x_i = x_i^k), \quad (3.7)$$

где каждый предикат есть дизъюнкция импликант, а каждая импликанта получается конъюнкцией элементарных импликант. Элементарная импликанта – это предикат, проверяющий равенство некоторого атрибута $x_i \in X_S$ фиксированному значению $x_i^k \in D_{x_i}$ из области значений атрибута $D_{x_i} = \{x_i^1, x_i^2, \dots, x_i^k, \dots\} = \{x_i^k | k \in I_{D,i}\}$.

С учетом этого для проверки верности выражений (3.5), (3.6) достаточно рассмотреть лишь те $x_i^k \in D_{x_i}$, которые входят в состав элементарных импликант в (3.7). Для этого достаточно попарно выбирать любые $p_{j_1}(x)$ и $p_{j_2}(x)$. Т.к. каждый из них задан в виде ДНФ (3.7), то применение к ним операции логического умножения $\&$ в выражениях (3.5), (3.6) даст также ДНФ. Полученное выражение сможет принять значение «истина», когда хотя бы в одной из ее импликант не содержится таких элементарных импликант, которые бы попарно противоречили друг другу:

$$(x_i = x_i^{k_1}) \wedge (x_i = x_i^{k_2}), \quad (3.8)$$

таких что $k_1 \neq k_2$.

Таким образом, G-сеть S_G непротиворечива тогда, когда хотя бы в одной импликанте в составе ДНФ, полученной в результате применения операции логического умножения всех предикатов $p_j(x) \in F_S^P$, заданных своими ДНФ, не существует противоречивых пар импликант какого-либо из атрибутов $x_i \in X_S$.

3.3.2. Корректность схем программ

G-сети как схемы программ соединяют в себе две структуры, необходимые для описания процесса информационно-аналитической поддержки: информационную и управляющую. Управляющая структура программы анализа задается множеством состояний G-сети, которые однозначно формируют множество разрешенных переходов $V_{S,r}^k$ на каждом k -ом шаге выполнения G-сети.

В то же время информационная структура, описывающая связи между параметрами G-сети должна быть приспособлена для операции выбора и пересылки значений необходимых параметров, участвующих в процессе вычисления. Однако в её рамках может возникать некорректность по входу и выходу некоторой n -й позиции G-сети, возникающая вследствие отсутствия детерминированности выбора или пересылки значения. Эти ситуации представлены на рисунке 3.9.

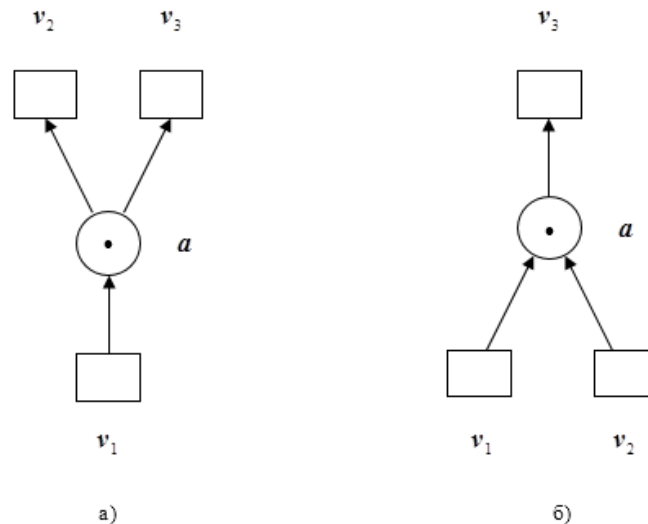


Рис. 3.9. Фрагменты G-сетей с некорректностью в позиции a :

(а) - по выходу и (б) - по входу

В результате возникает необходимость преобразования G-сети, эквивалентного с точки зрения вычисления заданной цели анализа. При этом позиция $a_i \in A_S$ G-сети S_G называется *корректной*, если она одновременно корректна по входу и корректна по выходу, т.е. для любой разметки этой позиции $\mu(a_i) > 0$ четко определено, каким переходом была сформирована любая фишка, и определен дальнейший порядок выбора каждой определяющей эту разметку фишки.

Алгоритм преобразования G-сети относительно некоторой позиции $a_i \in A_S$, предложенный в [132], позволяет достичь корректности вычислений в этой позиции. А корректировка, проведенная для каждой позиции сети, формирует корректную G-сеть в целом.

Суть работы этого алгоритма при некорректности по входу основана на том факте, что для простых G-сетей «ИЛИ»-дуги исходят из позиций, а «И»-дуги – из переходов. Процедура коррекции вначале формирует «И»-дуги дублированием нужным количеством дополнительных позиций для рассматриваемой позиции a_i , а затем «ИЛИ»-дуги для каждой из новых позиций. Для корректной работы рассматриваемого фрагмента G-сети должно быть найдено разбиение области значений предметных переменных предикатов, причем, такое, что логическая сумма всех этих предикатов есть функция, тождественно равная предикату выхода $p_i^-(x)$. Такое разбиение производится применением операций *расщепления* и *сокращения пустых переходов* с помощью преобразования ДНФ (см. выше) к виду совершенной ДНФ (СДНФ) [132]:

$$x \cdot y \vee \bar{x} \cdot y \cdot \bar{z} = x \cdot y \cdot (z \vee \bar{z}) \vee \bar{x} \cdot y \cdot \bar{z} = x \cdot y \cdot z \vee x \cdot y \cdot \bar{z} \vee \bar{x} \cdot y \cdot \bar{z}. \quad (3.9)$$

Каждой импликанте из СДНФ сопоставляется переход v_i^0 G-сети, который не производит никаких вычислений и выполняет работу пустого (холодного) оператора, а каждому такому переходу фиктивная позиция a_{i,i_1}^0 . Процедура сокращения множества пустых переходов заключается в проведении слияния тех пустых переходов, дуги от которых ведут к одним и тем же фиктивным позициям. В результате формируется корректная в соответствии с определенным выше G-сеть.

Аналогично производятся преобразования для корректировки G-сети по входу. Следует отметить, что в корректных G-сетях могут возникнуть ситуации, приводящие к неправильным вычислениям (непредусмотренные остановы и т.п.). Возможность возникновения этих ситуаций исключается посредством учета тупиковых позиций, отсутствие которых характеризует активность G-сети.

3.3.3. Активность схем программ

Непредусмотренные ситуации G-сети выражаются в том, что фишка в фиктивной позиции дальше не может быть выбрана из этой позиции ни по одной из «ИЛИ»-дуг вследствие того, что предикат применимости при входе в рассматриваемую позицию p^+ не равен предикату применимости выхода из нее p^- . Или, в более общем случае, когда предикат p^+ не является логическим следствием предиката p^- . В связи с этим входные и выходные предикаты позиции $a_i \in A_S$ необходимо рассматривать совместно при проведении коррекции. Таким образом, позиция $a_i \in A_S$ G-сети S^G является *тупиковой*, если при выполнении фрагмента G-сети с этой позицией возможен случай, когда пришедшая в нее фишка никогда не сможет быть выбрана оттуда [132]. Тогда решение таких непредсказуемых ситуаций сводится к устранению из G-сети всех её тупиковых позиций, а сама G-сеть без тупиковых позиций называется *активной*.

Необходимо отметить, что понятие тупиковости позиции содержательно связано с известными понятиями ограниченности и безопасности сетей Петри [127]. *Ограниченность* (некоторой величиной $c \geq 0$) позиции $a_i \in A_S$ G-сети означает, что для любой тупиковой ситуации в дереве достижимости сети разметка рассматриваемой позиции не превосходит величины c :

$$\mu(a_i) \leq c, \quad (3.10)$$

где при $c = 0$ позиция $a_i \in A_S$ не является *тупиковой*.

Условия входа и выхода фишки в позиции G-сети задается соответственно выражениями:

$$\begin{aligned} p^+(x) &= \text{"True"}, \\ p^-(x) &= \text{"True"}, \end{aligned} \quad (3.11)$$

где $p^+(x)$ и $p^-(x)$ - предикаты входа и выхода для позиции $a_i \in A_S$.

С учетом этого задача преобразования G-сети с целью исключения тупиковых позиций сводится к задаче ее корректировки со следующими дополнениями [132].

Условия формирования входных «ИЛИ»-дуг в позицию $a_i \in A_S$ задаются следующим образом:

$$\begin{aligned} \forall(j_1, j_2 = 1(1)k^+) [j_1 \neq j_2] &\Rightarrow [p_{j_1}^+ \wedge p_{j_2}^+ = \text{"False"}], \\ \forall_{j=1(1)k^+} p_j^+ &= p^+. \end{aligned} \quad (3.12)$$

А для выходных «ИЛИ»-дуг:

$$\begin{aligned} \forall(j_1, j_2 = 1(1)k^-) [j_1 \neq j_2] &\Rightarrow [p_{j_1}^- \wedge p_{j_2}^- = \text{"False"}], \\ \forall_{j=1(1)k^-} p_j^- &= p^- \vee p^+. \end{aligned} \quad (3.13)$$

Важно отметить, что правая часть содержит и входной, и выходной предикат применимости. В то же время из изложенного выше и определения тупиковой позиции следует, что позиция $a_i \in A_S$ не будет тупиковой, если:

$$p^-(x) \supset p^+(x), \quad (3.14)$$

т.е. предикат выхода $p^-(x)$ только тогда принимает истинное значение, когда и предикат входа $p^+(x)$ принимает истинное значение.

СДНФ предиката выхода для позиции $a_i \in A_S$, сформированная на основе ДНФ из (3.13), удовлетворяет условию (3.14), т.к. на основе свойства операции логического следования « \supset » справедливо:

$$(p^-(x) \vee p^+(x)) \supset p^+(x). \quad (3.15)$$

Тогда проведение процедуры коррекции по входу и выходу для некоторой позиции $a_i \in A_S$ G-сети на основании алгоритма коррекции, приведенного в разделе 3.3.2, с учетом выражений (3.12) (для коррекции по входу) и (3.13) (для коррекции по выходу) исключает рассматриваемую позицию из состава тупиковых. При этом множество пустых переходов необходимо пополнить еще одним переходом с индексом «0» - v_0 , который будет называться *тупиковым*, и еще одной фиктивной позицией, выходной для него, с целью поглощения тупиковых фишек, характеризующих нештатные ситуации в ПрО, возникающие во время анализа.

В результате указанных преобразований может быть сформирована непротиворечивая, корректная, активная программа в виде G-сети.

3.4. Реализуемость единого информационного пространства в рамках функционирования системы информационно-аналитической поддержки жизненного цикла космических средств

Результаты анализа технологий интеграции данных и приложений при формировании *ЕИП* (см. раздел 1.2.4) показывают, что среди существенных аспектов, позволяющих говорить о *реализуемости* ЕИП, и которым удовлетворяет создаваемая СИАП ЖЦ СОТО на модельно-алгоритмическом уровне, можно отметить следующие.

- использование смешанной *декларативно-процедурной интеграции* (технологии ЕИ и ЕАИ), предполагающей в рассматриваемом случае, с одной стороны, представление информации из источников данных в виде онтологических метамodelей (схем источников данных) в однообразном (использующим единое языковое представление) виде (см. приложение 3.3); с другой – семантическую интероперабельность приложений, поддерживаемую на основе опосредованного взаимодействия программных агентов через передачу сообщений (см. раздел 2.4.6);
- реализация *ETL-принципа* OLAP-систем, основанное на том факте, что СВЗ-модели формируются в виде троек «запрос выборки – аналитическая вычислительная задача – запрос записи» при том условии, что запись осуществляется в отдельное хранилище, а также поддерживается автоматический выбор источника данных при исполнении онтологических запросов в соответствии с представленным в приложении 3.3.

В то же время существенно, что ЕИП обладает рядом неотъемлемых черт (см. таблицу 1.3), и тогда в соответствии с предложенным модельно-алгоритмическим аппаратом показатель реализуемости ЕИП может быть формально оценен следующим образом:

$$F^{ds} = w_1 F^{пр} + w_2 F^{инт} + w_3 F^{неиз} + w_4 F^{хрон}, \quad (3.16)$$

где

$F^{пр}$ – показатель *предметной ориентированности* программ информационно-аналитической поддержки ЖЦ СОТО;

$F^{\text{инт}}$ – показатель *интегрированности* информационных ресурсов и программ информационно-аналитической поддержки ЖЦ СОТО;

$F^{\text{неиз}}$ – показатель *неизменчивости* хранилища электронного паспорта;

$F^{\text{хрон}}$ – показатель *поддержки хронологии* изменения ТС СОТО в рамках хранилища электронного паспорта.

w_1, w_2, w_3, w_4 – веса, характеризующие существенность частных показателей реализуемости ЕИП и задаваемые экспертным опросом [103].

Рассмотрим порядок оценивания этих частных показателей более подробно.

3.4.1. Предметная ориентированность программ информационно-аналитической поддержки

Моделирование бизнес-процессов на основе предложенной ранее онтологической МПЗ, очевидно, обеспечивает компоновку моделей процессов в виде совокупности «дорожек» в терминах нотации BPMN, что позволяет разделять модели процессов с точки зрения выполняемых в их рамках задач и рассматриваемых событий. Формально это выражается в том, что каждый отдельно взятый индивид процесса *Process* определяет ЖЦ *LifeCycle* (см. (2.10)). В свою очередь, сам процесс определен множеством индивидов событий *Event*, действий *Action* и условий *Condition* (см. (2.11)). В связи с тем, что на последующих этапах проектирования после задания совокупности бизнес-процессов, их элементы определяются соответствующими запросами *Query* и вычислительными задачами *ComputeTask* в рамках СВЗ-моделей по каждому процессу, а по ним задаются G-модели (см. (2.37), (2.38)) и, соответственно, G-сети, то, в конечном счете, будет сформировано множество предметно-ориентированных программ по количеству процессов и их описанию.

Однако в дополнение к рассмотренному, необходимо заметить, что предметная ориентация процессов информационно-аналитической поддержки связана, в том числе, и с содержанием запросов, предполагаемых для исполнения на этапе функционирования. В таком случае, можно сформулировать следующее утверждение.

Утверждение 3.2. Некоторый процесс pr_v модели бизнес-процессов m_i^{SA} информационно-аналитической поддержки ЖЦ СОТО является *предметно-ориентированным* тогда и только тогда, когда индивиды его событий определены в соответствии с выражениями (2.22) – (2.24), (2.31) таким запросами $RQuery$ в $\mathcal{A}^{meta,t}$, что их идентификаторы ресурсов URI_h не встречаются в запросах, связанных с процессами $pr_{v'} | v' \neq v$ модели бизнес-процессов m_i^{SA} за исключением случая $URI_h = URI_{ep}$, где URI_{ep} – идентификатор хранилища электронного паспорта, а также с учетом того, что каждый запрос обращен к единственному источнику данных.

Содержательно это утверждение означает следующее: каждая спецификация процесса должна быть связана с определённым в общем случае непустым множеством схем источников данных sk , идентифицируемым при помощи URI_h (см. приложение 3.3.4), т.е. каждая программа информационно-аналитической поддержки берет в качестве исходных данные только из собственного набора источников данных. Можно также заметить, что модель бизнес-процессов, где все процессы предметно-ориентированы, обеспечивает конфиденциальность источников данных на этапе функционирования СИАП ЖЦ СОТО, поскольку каждая программа имеет доступ только к своему набору данных.

Проверка предметной ориентированности процессов состоит в выполнении следующих шагов в циклическом переборе фактов из $\mathcal{A}^{meta,t}$, в которой определена модель бизнес-процессов m_i^{SA} :

- получить все индивиды события *Event* каждого процесса *Process*;
- найти для каждого события его запрос *Query*;
- получить и запомнить его уникальный идентификатор URI_h ;
- сравнить URI_h спецификации разных процессов;
- если найдены равные URI_h для разных процессов (за исключением URI_{ep}), то вернуть ошибку;

- иначе подтвердить предметную ориентированность модели бизнес-процессов.

При наличии предметной ориентированности модели бизнес-процессов следом необходимо убедиться в интегрированности информационных ресурсов и программ по этой модели.

3.4.2. Интегрированность системы информационно-аналитической поддержки

Интегрированность в СИАП ЖЦ СОТО аналогично связана с запросами к источникам данных, с одной стороны, а также с *интероперабельностью* процессов в виде передачи сообщений – с другой. Поясним это утверждение более подробно.

В рассматриваемом контексте под *интегрированностью* информационных ресурсов содержательно можно понимать наличие доступа ко всем источникам данных СИАП ЖЦ СОТО, данные которых подлежат обработке и анализу, а также факт, что результаты обработки на этапе функционирования будут записаны в отдельное хранилище, характеризующее моделью ТС СОТО.

В таком случае, проверка интегрированности информационных ресурсов СИАП ЖЦ СОТО состоит в том, чтобы убедиться, что уникальные идентификаторы *URI* онтологических запросов выборки *RQuery* в $\mathcal{A}^{meta,t}$ охватывают все именованные n_h схем источников данных M^{Sk} (см. приложение 3.3.4), а запросы записи связаны с единственным хранилищем электронного паспорта. Такая проверка может быть осуществлена с помощью следующих шагов в цикле перебора фактов из $\mathcal{A}^{meta,t}$, в которой определена модель бизнес-процессов m_i^{SA} :

- получить все индивиды событий *Event* всех процессов *Process*;
- найти для каждого события его запрос *RQuery* или *WQuery*;
- получить и запомнить отдельно уникальный идентификатор URI_h для *RQuery* и *WQuery*;

- сравнить полученное множество идентификаторов URI для $RQuery$ (исключив из него повторяющиеся записи) с множеством именований схем источников данных $N = \{n_h | h = 1 \dots t\}$, где $n_h \subset M_h^{Sk}$;
- если $URI = N \setminus n_{ep}$, где n_{ep} – именование хранилища электронного паспорта, то сравнить URI для $WQuery$ (исключив из него повторяющиеся записи) с n_{ep} ;
- если $URI = n_{ep}$, то подтвердить интегрированность информационных ресурсов;
- иначе вернуть ошибку.

В свою очередь, для проверки *интероперабельности* как частного показателя интегрированности (приложений – EAI) необходимо убедиться в том, что в каждом процессе модели бизнес-процессов имеется факт в \mathcal{A}^{meta} о передаче сообщения в другой процесс в соответствии с выражениями (2.41), (2.42). Такая проверка предполагает выполнение следующих шагов (аналогично) в цикле перебора фактов из $\mathcal{A}^{meta,t}$, в которой определена модель бизнес-процессов m_i^{SA} :

- найти индивиды $Message$ для каждого процесса $Process$;
- при отсутствии сообщения в модели какого-либо из процессов вернуть ошибку;
- для каждого сообщения $Message$ найти индивид события $Event$, для которого предназначено сообщение (связь по роли $MessageTo$);
- убедиться, что это событие определяет процесс, отличный от того, которое определяет текущее сообщение;
- если у всех процессов есть сообщения, которые удовлетворяют пункту выше, то подтвердить интероперабельность процессов;
- иначе вернуть ошибку.

Таким образом, в автоматическом режиме может быть осуществлена проверка интегрированности для СИАП ЖЦ СОТО на основе проверки интегрированности информационных ресурсов и интероперабельности.

3.4.3. Неизменчивость хранилища результатов функционирования системы информационно-аналитической поддержки

Относительно вопроса неизменчивости хранилища электронного паспорта необходимо лишь заметить, что благодаря реализации *ETL-принципа* в рамках модельно-алгоритмического комплекса СИАП ЖЦ СОТО, а также возможности в рамках СВЗ-моделей определять только запросы на запись вида П.(3.4) (см. приложение 3) обеспечивается накопление информации о ТС СОТО в хранилище в том порядке, в каком осуществляются эти запросы на запись, за счет чего обеспечивается сохранность информации о ТС СОТО, поскольку не предполагается возможности удаления данных из хранилища электронного паспорта.

3.4.4. Поддержка хронологии изменения технического состояния объекта информационно-аналитической поддержки

Одним из важных аспектов автоматизированной информационно-аналитической поддержки ЖЦ СОТО является *поддержка хронологии* изменения ТС СОТО для реализации возможности отслеживания хода работ над изделием и того, как с их течением изменяется состояние изделия. Результаты анализа в разделе 1.2 показали, что наличие такого свойства в ИАС позволяет явно наблюдать недостатки производственного или эксплуатационного процессов и, соответственно оптимизировать их.

Предлагаемый подход информационно-аналитической поддержки позволяет обеспечить наличие такого свойства за счет учета ряда аспектов при проектировании соответствующего полимодельного комплекса. Поскольку, как уже было отмечено выше, функционирование СИАП ЖЦ СОТО ориентировано на запись результатов анализа ТС как множества определяемых событий бизнес-процессов на основе совокупности информационных единиц, формализуемых как множество исходных событий модели бизнес-процессов, существует возможность привязки ко времени каждого определяемого (или целевого) события при записи результатов анализа. В таком случае, может быть сформулировано следующее утверждение.

Утверждение 3.3. СИАП ЖЦ СОТО, формируемая на основе модели бизнес-процессов m_i^{SA} , обладает свойством *поддержки хронологии* тогда и только тогда, когда соблюдаются 2 следующих условия:

- в онтологии электронного паспорта O_{ep}^d имеется атрибут времени $Time \in AN_{ep}^d$ с соответствующим типом данных $xsd:dateTimeStamp \in D$, причем в схеме электронного паспорта sk_h^{ep} (т.е. в \mathcal{A}_{ep}^d) этот атрибут связан с каждым концептом C , таким что $C \sqsubseteq Phenomenon$;
- в каждом запросе записи каждой СВЗ-модели имеется конъюнкт вида $C \text{ Time } x$, где значение переменной x – текущее время на момент исполнения запроса.

Содержательно это означает, что атрибут времени должен быть связан со всеми концептами онтологии электронного паспорта, характеризующими явления в ПрО, а исполнение запросов записи предполагает и запись текущего времени, когда соответствующее явление было оценено системой, причем значение времени может быть взято из параметра x^{timer} (см. раздел 2.4.3). Так, например, для модели состояния, предложенной в приложении 3.3.5, атрибут времени должен быть связан с документами, характеризующими ТС СОТО:

$$Document \sqsubseteq \exists Time. xsd:dateTimeStamp. \quad (3.17)$$

Тогда, схему sk_h^{ep} , представленную на рис. П. 3.7 (см. приложение 3), можно преобразовать к виду, представленному на рисунке 3.10.

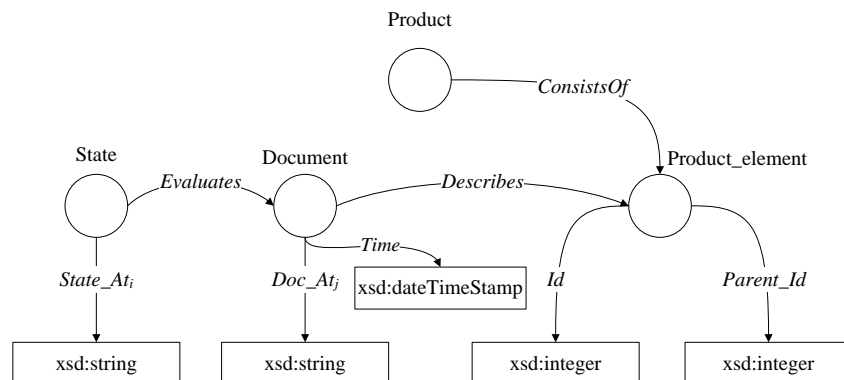


Рис. 3.10. Схема электронного паспорта с учетом временной характеристики

Проверка поддержки хронологии для СВЗ-моделей заключается (как и ранее) в циклическом переборе фактов из $\mathcal{A}^{meta,t}$ со следующими шагами:

- получить все индивиды событий *InterEvent* и *FinalEvent* всех процессов *Process*;
- найти для каждого события его запрос *WQuery*;
- найти индивиды конъюнкта *Conjunct* каждого запроса *WQuery*;
- среди множества конъюнктов найти тот, в котором имеется атрибут *Time*;
- если для всех запросов записи найден атрибут *Time*, то подтвердить поддержку хронологии;
- иначе вернуть ошибку.

Таким образом, при соблюдении указанных условий СИАП ЖЦ СОТО будет обладать свойством поддержки хронологии.

3.5. Методика оценивания функциональной эффективности информационно-аналитической поддержки жизненного цикла космических средств

3.5.1. Оценивание уровня информационно-аналитической поддержки

Существенным вопросом является оценивание эффективности применения СИАП ЖЦ СОТО в ПрО при её функционировании на основе рассматриваемого модельно-алгоритмического комплекса. В ГОСТ о системе информации (см. раздел 1.1.2) отмечается необходимость её совершенствования на основе использования современных ИТ для обеспечения заинтересованных организаций КК актуальной и достоверной информацией об уровне ТС и надежности СОТО. С учетом этого, *уровень информационно-аналитической поддержки* ЖЦ СОТО может быть оценен на основе *оперативности* обеспечения соответствующей информацией и её *достоверности*, что формально может быть задано следующей аддитивной функцией свертки показателей информационно-аналитической поддержки:

$$Q^{ias} = w_1 Ver^{ias} + w_2 Eff^{ias}, \quad (3.18)$$

где

Ver^{ias} – показатель *достоверности* результатов информационно-аналитической поддержки ЖЦ СОТО;

Eff^{ias} – показатель *оперативности* информационно-аналитической поддержки ЖЦ СОТО;

w_1 и w_2 – весовые коэффициенты важности критериев, получаемые в результате экспертного опроса, причем такие что [103]:

$$0 \leq w_i \leq 1; \sum_{i=1}^N w_i = 1, \quad (3.19)$$

где N – количество критериев для общего случая функции свертки.

Рассмотрим более подробно, как могут быть рассчитаны показатели оперативности и достоверности для СИАП ЖЦ СОТО.

Из [132] известно, что при формировании программ на основе G-сетей, обеспечивающих получение результатов в РВ, могут быть рассчитаны временные затраты на вычисления на основе задания следующего множества параметрических отображений:

$$f_{\tau}: V_S \times X_S^+ \rightarrow T^S, \quad (3.20)$$

где

V_S – множество переходов G-сети, каждый из которых соответствует некоторому элементу множества операторов F_S ;

$X_S^+ \subset X_S$ – множество значений входных параметров для каждого оператора-перехода;

$T^S \in \mathbf{R}^+$ – временной ресурс, необходимый для реализации оператора-перехода с соответствующими для него исходными данными.

Поскольку G-сеть в общем случае может представляться совокупностью альтернативных ветвей подалгоритмов (где альтернативная ветвь определяется по решающему переходу (см. [132]), то временная характеристика по каждому альтернативному алгоритму может быть вычислена на основе своих слов вывода р-грамматики $\alpha_k \in L_v$ языка L_v , образующих схему алгоритма из упорядоченного множества имен операторов:

$$\alpha_k = a_k^1 a_k^2 \dots a_k^{l_k}, \quad (3.21)$$

задающих последовательность запуска переходов (без повторения циклических участков) G-сети – $v_k^1, v_k^2, \dots, v_k^{l_k}$.

Тогда условный (без циклов) временной ресурс $\tau_k \in T^S$, необходимый для реализации k -ой ветви алгоритма вычислений, определится:

$$\tau_k = \sum_{i=1}^{l_k} \tau_k^i, \quad (3.22)$$

где

τ_k^i - временная характеристика оператора-перехода v_k^i ;

$l_k = |\alpha_k|$ - длина слова вывода α_k .

Тогда с учетом параллельного и асинхронного исполнения схемы программы, заключающегося в параллельном исполнении элементарных неделимых элементов G-сети в виде её операторов, время исполнения всей G-сети в целом (при наличии значений для всех X_S^+) будет равно времени исполнения наиболее длительной её ветви:

$$T^S = \max_{k=1(1)n_k} \{\tau_k\}, \quad (3.23)$$

где n_k - количество альтернативных ветвей. А время функционирования всей СИАП ЖЦ СОТО для совокупности G-сетей, из которых она состоит:

$$T^{ias} = \sum_{v=1}^p T_v^S, \quad (3.24)$$

где p – количество схем программ СИАП ЖЦ СОТО.

В то же время поскольку в рассматриваемом контексте задача состоит в оценивании оперативности СИАП ЖЦ СОТО, то необходимо учесть максимальное допустимое и минимальное время функционирования системы. С учетом этого, показатель *оперативности* информационно-аналитической поддержки ЖЦ СОТО может быть рассчитан на основе следующего выражения:

$$Eff^{ias} = \frac{T^{max} - (T^{ias} + T^{in})}{T^{max} - T^{min}}, \quad (3.25)$$

где T^{max} , T^{min} – максимальное допустимое и минимальное время получения результатов информационно-аналитической поддержки соответственно, T^{in} – временной ресурс ввода исходных данных (оцифровки документов на бумажном носителе).

В свою очередь, оценивание достоверности, очевидно, связано с достоверностью и полнотой исходных данных, с одной стороны. С другой – она существенно зависит от того, насколько сформированная программа соответствует исходным требованиям (в соответствии с показателем согласованности из раздела 3.2). При наличии согласованной системы требований, а также процедуры автоматического синтеза ВМ и их схем программ можно говорить об инвариантности сформированной программы исходным требованиям, и соответственно, гарантированном заранее известном поведении СИАП ЖЦ СОТО, определяемом по набору полных и достоверных исходных данных с учетом принципа управления по данным. Кроме того, существенную роль играет и свойство корректности программ, позволяющее говорить о детерминированности вычислительного процесса, вычислимости результата и завершаемости программ (в соответствии с описанием из раздела 3.3). Тогда формально показатель *достоверности* может быть рассчитан на основе следующего выражения:

$$Ver^{ias} = w_1 \frac{\sum_{i=1}^p Ver_i^{in}}{p} + w_2 \frac{\sum_{j=1}^p C_j^{in}}{\sum_{j=1}^p C_j^{full}} + w_3 \frac{\sum_{k=1}^p V_k^{ias}}{p} + w_4 C^{cs}, \quad (3.26)$$

где Ver_i^{in} – достоверность каждого набора исходных данных для каждой программы СИАП, C^{in} – полнота исходных данных, представленных исходными фактами ПрО, определяемыми результатами исполнения запросов выборки в виде кортежей $\langle c_1, c_2, c_3 \rangle$, где c_1, c_2, c_3 – полученные значения, V^{ias} – корректность совокупности программ в количестве p единиц, C^{cs} – согласованность спецификаций.

Таким образом, на основе указанных выражений может быть оценен уровень информационно-аналитической поддержки ЖЦ СОТО. Необходимо

также заметить, что в случае оценивания этого показателя для информационно-аналитической поддержки, осуществляемой без СИАП, конкретные значения могут быть получены методом непосредственного экспертного оценивания.

3.5.2. Оценивание эффективности информационно-аналитической поддержки

В результате рассмотрения основных аспектов квалиметрии моделей и уровня информационно-аналитической поддержки ЖЦ СОТО можно перейти к непосредственно способу оценивания её функциональной эффективности. В рамках текущего раздела были рассмотрены такие показатели как согласованность спецификаций бизнес-процессов, СВЗ, G-моделей и G-сетей, реализуемость ЕИП в рамках СИАП ЖЦ СОТО, корректность G-сетей, уровень информационно-аналитической поддержки. Очевидно, совокупность этих показателей и характеризует эффективность процессов информационно-аналитической поддержки, поскольку каждый из них отражает те или иные стороны моделирования и организации корректных вычислений в РВ, обеспечивающих оперативное предоставление достоверных данных о ТС и надежности СОТО, в том числе, на основе обеспечения взаимодействия программ СИАП ЖЦ СОТО, а также обеспечения соответствия хода вычислительного процесса реальным процессам информационно-аналитической поддержки. Таким образом, функциональная эффективность определена уровнем информационно-аналитической поддержки, а с точки зрения охвата ЖЦ СОТО реализуемостью ЕИП. Аналогичному (3.18), показатель функциональной эффективности может быть рассчитан на основе следующей аддитивной функции свертки:

$$E^{ias} = w_1 Q^{ias} + w_2 F^{ds}, \quad (3.27)$$

где, напомним, Q^{ias} – уровень информационно-аналитической поддержки, F^{ds} – реализуемость ЕИП, а w_1, w_2 определены аналогично (3.19).

В результате по совокупности рассмотренных показателей и алгоритмов могут быть верифицированы модели СИАП ЖЦ СОТО, с одной стороны, и оценена эффективность информационно-аналитической поддержки – с другой.

3.6. Выводы по разделу

1. Предложен формальный способ проверки согласованности спецификаций информационно-аналитической поддержки ЖЦ СОТО, основанный на использовании известного табло-алгоритма в рамках аппарата дескрипционных логик, позволяющего в решаемой задаче оценить разрешимость моделей и выполнимость терминологических аксиом онтологических спецификаций бизнес-процессов и СВЗ. В дополнение к нему разработан алгоритм проверки онтологической выразительности спецификаций, позволяющий с одной стороны, осуществить проверку соответствия разрабатываемых онтологических моделей используемой нотации BPMN, а с другой – оценить взаимное соответствие моделей для последующей организации корректных и инвариантных исходным требованиям вычислений по ним.
2. Рассмотрена и показана применимость методов верификации используемых схем программ в виде G-сетей к задаче информационно-аналитической поддержки ЖЦ СОТО с проверкой их непротиворечивости, корректности и активности.
3. Разработан алгоритмический комплекс проверки реализуемости ЕИП в рамках функционирования СИАП ЖЦ СОТО на основе оценивания частных показателей качества полимодельного комплекса: предметной ориентированности программ, интегрированности информационных ресурсов и приложений, неизменчивости хранилища электронного паспорта СОТО, поддержки хронологии изменения ТС СОТО.
4. Разработана методика оценивания функциональной эффективности информационно-аналитической поддержки ЖЦ СОТО, в основу которой положена сформированная структура показателей качества полимодельного комплекса: согласованности спецификаций, реализуемости ЕИП, корректности схем программ и уровня информационно-аналитической поддержки ЖЦ СОТО.

4. ПРАКТИЧЕСКОЕ ПРИМЕНЕНИЕ РЕЗУЛЬТАТОВ ТЕОРЕТИЧЕСКИХ ИССЛЕДОВАНИЙ В ЗАДАЧЕ ИНФОРМАЦИОННО-АНАЛИТИЧЕСКОЙ ПОДДЕРЖКИ ЖИЗНЕННОГО ЦИКЛА РАКЕТЫ-НОСИТЕЛЯ «СОЮЗ-2»

4.1. Описание задачи и элементов системы автоматизированного формирования электронного паспорта ракеты-носителя «Союз-2»

4.1.1. Единый виртуальный электронный паспорт космической ракеты-носителя "Союз-2".

В соответствии с предложенным модельно-алгоритмическим комплексом была разработана специализированная СИАП ЖЦ космических средств на примере РН «Союз-2» для предприятий КК, связанных с этим изделием, названная Единым Виртуальным Электронным Паспортом (ЕВЭП) РН «Союз-2» [149, 178, 181, 220, 221, 255]. Как уже ранее отмечалось, ЕВЭП предназначен для своевременного обеспечения предприятий и организаций, участвующих в проектировании, производстве, испытаниях и эксплуатации космических средств актуальной информацией, необходимой для выполнения работ по обеспечению и повышению уровня их ТС и надежности на всех этапах их ЖЦ. Необходимо отметить, что в рамках исследования при разработке ПК СИАП рассматривались только задачи, связанные с этапами проектирования и производства РН «Союз-2».

В качестве исходной информации в ЕВЭП рассматриваются в общем случае следующие её виды, рассмотренные в таблице 4.1.

Таблица 4.1. Некоторые примеры информации, обрабатываемой в рамках СИАП ЖЦ СОТО

Организационно-техническая информация	Технологическая информация
приказы	конструкторская документация
распоряжения	эксплуатационная документация
технологические графики работ	извещения
сетевые графики	акты
репортажная информация	сведения о регламентных работах
корпоративная корреспонденция	сведения о доработках и ремонтах
технические решения	сведения о технических проверках

технические решения	сведения о проведении испытаний
частные ТЗ	технологические графики работ с фактическими сроками исполнения этапов и информацией по задержкам и срыву сроков
сведения о принимаемых решениях в ходе испытаний	сведения о замечаниях, отказах и неисправностях
донесения и данные о ходе выполнения работ	качественные оценки результатов испытаний
сведения о ресурсах	формализованные данные отчетов об испытаниях
сведения о наличии комплектующих и запасных частей, инструментов и принадлежностей с указанием гарантийных сроков хранения	сведения о результатах работы комиссий

Результатом функционирования СИАП ЖЦ СОТО является совокупность оценок показателей ТС СОТО, формируемая в результате анализа исходной информации, совмещенная с метаданными информационных единиц, входных для ЕВЭП. Такие метаданные формируются в результате процесса атрибутирования документации, упомянутого в приложении 3.3.5. Причем, в случае если документ представлен в электронном виде и имеет свой набор метаданных, то осуществляется импорт такого документа. В случае если документ представлен на бумажном носителе, то он подлежит оцифровке и внесению данных по атрибутам его модели в соответствии с заданным типом документа, что будет продемонстрировано далее. Необходимо также заметить, что совокупность всех показателей ТС РН «Союз-2» вместе с атрибутами документов структурировано в соответствии с конфигурацией изделия, формируемой на этапе его проектирования и также подлежащей её внесению в ЕВЭП.

Для решения функциональных задач ЕВЭП система включает в себя:

- компоненты управления информационными ресурсами;
- компоненты информационной инфраструктуры;
- компоненты доступа к данным;
- компоненты аналитической отчётности.

Компоненты управления информационными ресурсами предназначены для сбора, интеграции и хранения разнородной информации о РН «Союз-2» и входящих в его состав изделий.

Компоненты информационной инфраструктуры предназначены для организации эффективной коммуникационной среды, основанной на единых принципах и механизмах обмена электронными данными между предприятиями и организациями, участвующими в создании, производстве и эксплуатации РН «Союз-2».

Компоненты доступа к данным предназначены для реализации сервисов обработки и представления информации.

Компоненты генерации аналитической отчётности предназначены для формирования различных отчётных документов и протоколов на основе данных информационных ресурсов.

Одной из актуальных задач применения ЕВЭП является задача так называемого формирования заявления о соответствии изделия требованиям заказчиков, интерпретируемого как процесс информационно-аналитической поддержки организаций КК на этапах проектирования и производства, где формирование заявления является последним этапом процесса производства, сигнализирующим об окончании работ над изделием. Эту задачу можно продемонстрировать в виде следующего рисунка (см. рис. 4.1).

При этом в рассматриваемом случае, акцент на формирование заявления о соответствии РН «Союз-2» требованиям заказчиков необходимо понимать как необходимость оценивания ТС в РВ на протяжении этапов проектирования и производства. При этом сам электронный паспорт изделия должен предполагать наличие всей информации об изделии с показателями его ТС от момента начала проектирования изделия и вплоть до выпуска заявления о соответствии изделия.

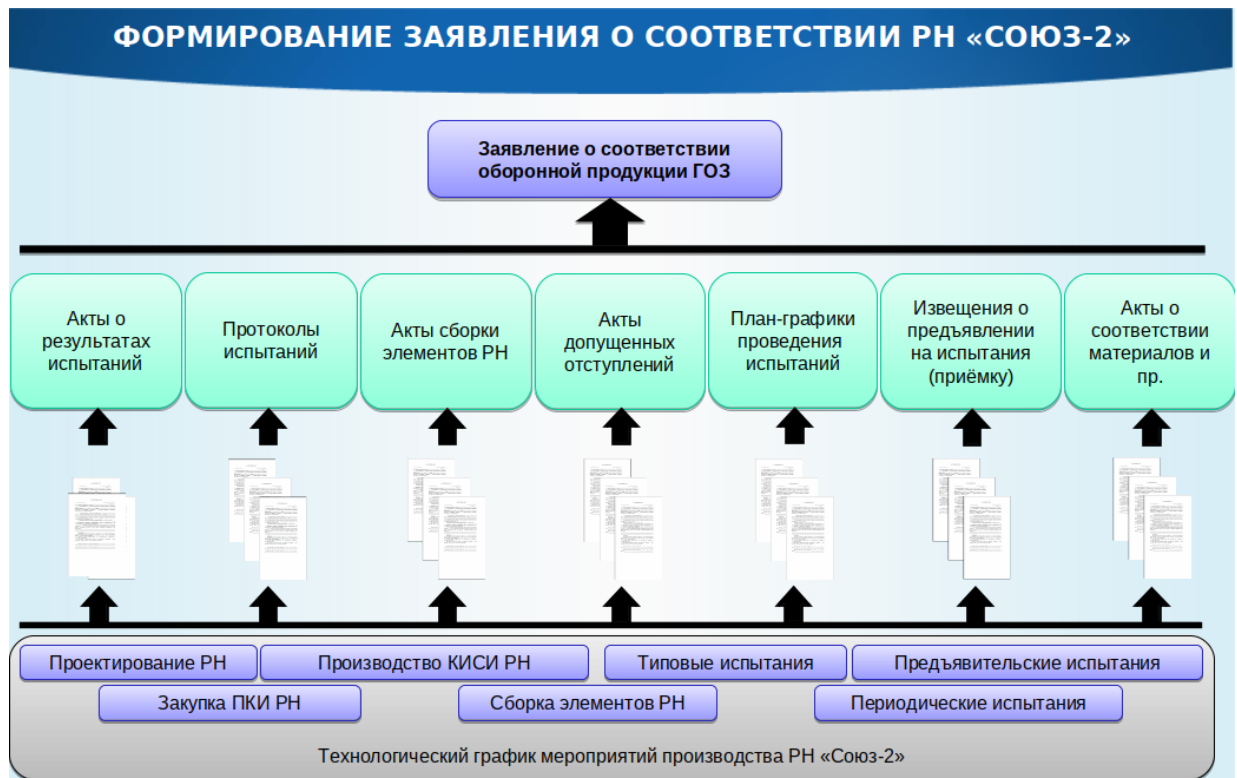


Рис. 4.1. Задача формирования заявления о соответствии изделия требованиям заказчиков

Задача формирования заявления о соответствии РН «Союз-2» непосредственно связана с задачей оценивания соответствия изделия заданной конфигурации на протяжении всего его ЖЦ. Формирование заявления о соответствии – это процесс, реализующий формирование согласованного набора (комплекта) документов, являющихся доказательной базой того, что изделие и все его составные части (на стадиях проектирования и изготовления) соответствуют заданным требованиям. С точки зрения этого процесса каждое действие, связанное с изделием, документируется. Таким образом, формируется некоторая структура документов, соответствующая состоянию изделия на определенный временной срез. Конечной точкой временного отрезка является сформированное заявление о соответствии, являющееся доказательной базой того, что изделие соответствует заданным требованиям.

Электронный паспорт изделия превращает всю совокупность информации о РН «Союз-2» в важнейший информационный ресурс и позволяет решать следующие критические для ЛПР задачи:

- обеспечивать оперативный информационный доступ к электронным документам и совокупности их метаданных, несущих семантику, связанную с прохождением этапов технологических процессов проектирования и производства и выражаемую в численных и строковых значениях атрибутов документов;
 - обеспечивать обмен электронными документами (конструкторскими, эксплуатационными и пр.), а также результатами всех видов испытаний и данными объективного контроля;
 - проводить оценивание уровня ТС, надёжности и качества изделия, всех его составных частей, а также готовности к выполнению целевой задачи;
 - обеспечивать информационную поддержку управления проектом работ над изделием на основе отслеживания хода выполнения работ и его корректировки по необходимости, основываясь на показателях ТС СОТО и обрабатываемых метаданных документов;
 - обеспечивать информационную поддержку управления данными об изделии на основе их последовательного накопления, процесса комплектования электронного паспорта, подготовки метаданных при необходимости;
 - обеспечивать информационную поддержку управления конфигурацией изделия на основе формирования так называемой электронной структуры изделия (ЭСИ), представляющей модель «древовидной» структуры вложенности узлов и агрегатов изделия;
 - обеспечивать информационную поддержку управления качеством на основе оперативного отслеживания показателей качества изделия по метаданным документов;
 - обеспечивать информационную поддержку логистики, связанную с отслеживанием по наборам метаданных текущего местонахождения соответствующих узлов и агрегатов, их гарантийные сроки и др., а также историю их перемещения на основе поддержки хронологии изменения ТС СОТО
- и др.

Таким образом, ЕВЭП является реализующей системой, ориентированной на выполнение задач автоматизированной информационно-аналитической поддержки ЖЦ РН «Союз-2» и предназначен для оперативного обеспечения заинтересованных организаций достоверной информацией об уровне ТС и надежности РН «Союз-2».

4.1.2. Описание порядка функционирования системы информационно-аналитической поддержки ЖЦ РН «Союз-2»

Для рассматриваемой задачи информационно-аналитической поддержки ЖЦ РН «Союз-2» на этапах проектирования и производства предполагается представленный на рисунке 4.3 типовой сценарий функционирования, описывающий порядок функционирования для некоторого единственного бизнес-процесса. Далее дан краткий комментарий по основным элементам изображенного на рисунке сценария.

В качестве основных элементов АПК ЕВЭП используются:

- автоматизированное рабочее место (АРМ) комплектования, в рамках которого КП осуществляет деятельность по сбору, интеграции и систематизации данных из АС предприятий, находящихся в составе КК, и чья информация подлежит интеграции и анализу; процесс комплектования при необходимости предполагает выполнение задач атрибутирования электронных документов в виде специальных электронных карточек регистрации и помещения этих документов в систему управления электронными данными (обозн., СУЭД) (см. рис. 4.2);
- АРМ оцифровки, используемый, очевидно, для оцифровки документов, представленных на бумажном носителе, и их размещение на ресурсе оперативного обмена информацией.
- АРМ пользователя публичного ресурса, предназначенный для визуализации в РВ данных электронного паспорта изделия. В его задачи также входит подготовка электронных документов, их размещение в качестве исходных данных в логических информационных ресурсах СУЭД, доступном для СИАП,

- а в результате проведенного в рамках СИАП анализа опубликование в доступном для других КП публичном ресурсе;
- СИАП, осуществляющая сбор, обработку и анализ данных с занесением результатов оценивания ТС в электронный паспорт РН «Союз-2»;
 - Система управления электронными данными, обеспечивающая хранение и доступ к информации различной природы на основе различных протоколов;
 - Используемые информационные ресурсы являются логическими элементами СУЭД, представляющими собой нормативно-справочные БД, а также соответствующую файловую систему, связанную с хранением электронных документов.

Извещение

ИЗВЕЩЕНИЕ

о предъявлении изделия РН "Союз 2-1в" на испытания (приемку)

Изделие (индекс)	РН "Союз 2-1в"		
Номер документа	Дата документа	09.10.2016	
Основание	< ---	№	26.09.2016
Проверены и подтверждены испытаниями	< ---	№	26.09.2016
Подписано	< ---		26.09.2016
Количество изделий (партии)	1	шт	Поступило в ВП 27.09.2016
Подпись и решение ВП	---		---

Рис. 4.2. Пример электронной регистрационной карточки документа

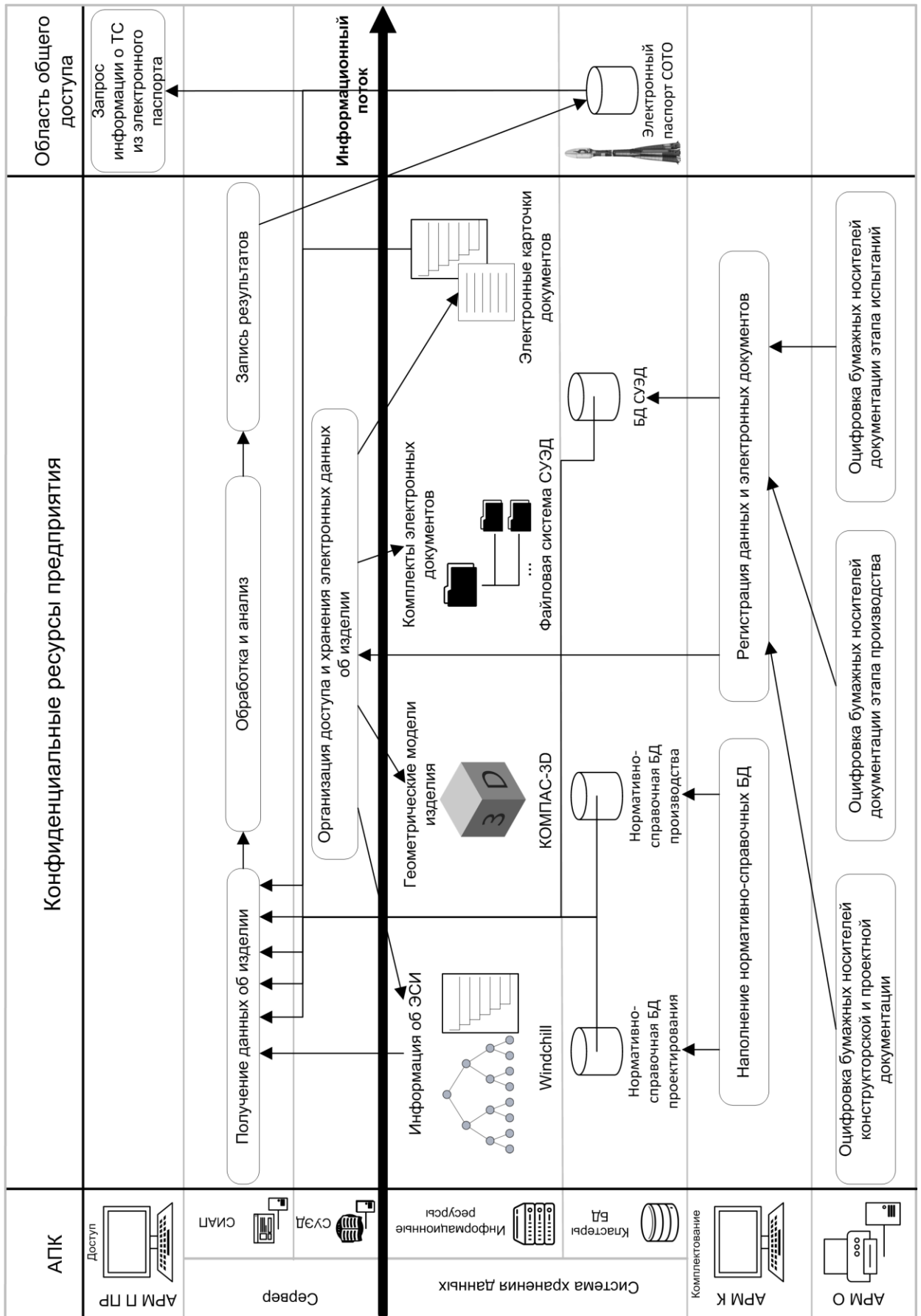


Рис. 4.3. Порядок взаимодействия элементов АПК ЕВЭП

4.1.3. Программное обеспечение доступа к информации из электронного паспорта РН «Союз-2»

На рисунке 4.4 представлен интерфейс ПО АРМ доступа к информации из электронного паспорта РН «Союз-2». Необходимость его демонстрации продиктована тем, чтобы явно отобразить некоторые из показателей ТС изделия, отслеживаемые СИАП, поскольку сама СИАП функционирует в фоновом режиме, а результаты её функционирования визуализируются в ПО доступа с помощью исполнения набора онтологических запросов и последующего отображения полученных данных.

АРМ формирования заявления о соответствии изделия РН "Союз 2-1в"

Выгрузить электронное дело изделия

Ведомость контроля

Поиск по документам

Блок обобщенной информации о ТС СОТО

Информация по изделию РН "Союз 2-1в"

Индекс изделия	Заводск...	Дата изгот...	Дата окончан...	Допущены е отступления (отказы)			
				ТА (КРД)	РА (СОН)	влияет	НЕ влияет
14A15	1Л	не указана	не указана	0	0/0	1	5

Таблица допущенных отступлений

Наименование	№ документа	Дата документа
Карточка разрешения отсека бака "О"	963	13.04.2017
Карточка разрешения маршевого двигателя 2 ступени	0674-14	09.10.2017
Карточка разрешения блока 1 ступени	663-12	18.04.2017
Карточка разрешения маршевого двигателя 2 ступени	0497-14	30.06.2017
Карточка разрешения маршевого двигателя 2 ступени	0473-14	15.07.2017

Документы этапов ЖЦ

Интерактивная 3D-модель

Электронная структура изделия

Интерактивный технологический график

Блок отслеживания комплектности документов

Организационно-штатная структура предприятия | Электронная структура изделия | Заявление о соответствии

Заявление о соответствии

Список документов

Акты о результатах периодических испытаний

Акты о результатах типовых испытаний

Акты о соответствии материалов, полуфабрикатов, ЛКМ и связей

Протоколы предъявительских испытаний

Протоколы приемо-сдаточных испытаний

Извещения о предъявлении на испытания (приемку)

План-графики проведения периодических испытаний

Протокол приемо-сдаточных испытаний блока 2 ступени №433/45 от 16.11.2016

Протокол приемо-сдаточных испытаний отсека бака "О" №657/15/9 от 01.09.2016

Протокол приемо-сдаточных испытаний РН "Союз 2-1в" №657/15/12 от 22.06.2016

Протокол приемо-сдаточных испытаний РН "Союз 2-1в" №657/15/13 от 26.06.2016

Протокол приемо-сдаточных испытаний блока 1 ступени №657/15/1 от 04.09.2016

Рис. 4.4. Интерфейс ПО АРМ доступа к информации из электронного паспорта РН «Союз-2»

Основным объектом, предоставляющим интерфейс доступа к консолидированным данным о СОТО на рассматриваемых стадиях ЖЦ, является его физическая электронная структура - ЭСИ. ЭСИ представляет собой «древовидную» структуру, содержащую детали, сборочные единицы, комплекты и комплексы, образующие состав изделия, а также ассоциированные с каждой

такой составной частью информационные ресурсы СОТО - электронные данные, соответствующие стадиям ЖЦ СОТО. Таким образом, ЭСИ является основой представления инженерных данных об изделии. Пример составной части ЭСИ приведен на рисунке 4.5. ЭСИ выступает в качестве «каркаса», к узлам которого крепятся метаданные, характеризующие ТС СОТО.

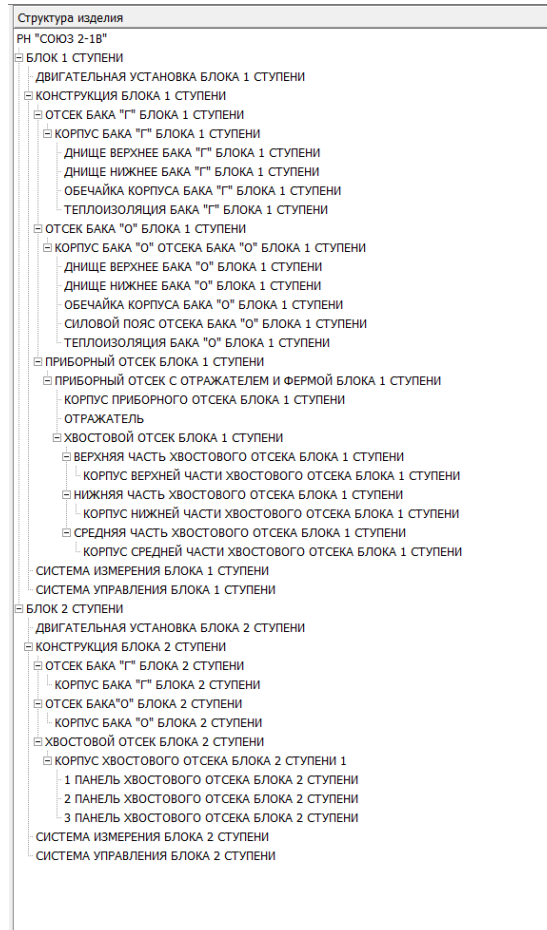


Рис. 4.5. Пример составной части ЭСИ изделия РН «Союз-2»

Другим элементом ПО является электронное дело изделия, формируемое в результате наполнения ИС. Ведение электронного дела изделия на всех этапах ЖЦ с использованием поддержки хронологии, упоминавшейся ранее, позволяет организовать сбор, хранение, обмен, доступ ко всей необходимой информации об изделии на основе использования ЭСИ.

Для распределения потоков информации о ТС СОТО используется организационно-штатная структура (ОШС) СОТО. ОШС представляет собой иерархическую структуру предприятия, элементы которой тем или иным образом

связаны с ЭСИ в технологическом или организационном процессе. Благодаря уточнению таких взаимосвязей обеспечивается возможность отслеживания ответственных единиц, порядка выполнения их работ, их производительность и эффективность.

В составе ПО реализована интерактивная интеллектуальная 3D- модель СОТО. Модель функционирует на основе потоковых вычислений, соответствующих принципам ОВМ. 3D-модель является специальной мнемосхемой, визуализирующей обобщенные показатели ТС СОТО, полученные в результате семантического анализа данных в составе СИАП. Изменение состояния элементов изделия отражается на 3D-модели с помощью цветовой индикации в РВ в результате появления новой информации об изделии.

Еще одним элементом ПО является интерактивный интеллектуальный технологический график, формальные основы проектирования которого были предложены в [213]. График позволяет в режиме РВ оценивать состояния происходящих процессов на основе потоковой модели вычислений. Визуальные средства технологического графика позволяют ЛПР наглядно оценивать текущий этап ЖЦ СОТО, степень готовности изделия, наличие/отсутствие отказов и неисправностей в составе СОТО, соответствие текущего этапа запланированному директивному времени. Этапность процессов логически взаимосвязана с наборами документов, полнота метаданных которых в соответствии с заранее заданной моделью и совокупностью оценок показателей, полученных на их основе, является определяющей для перехода от одной задачи процесса к другой.

В рамках интерфейса ПО представлен также блок визуализации известных нештатных ситуаций на производстве, предназначенный для отслеживания совокупности наиболее критичных показателей ТС изделия рассматриваемого этапа ЖЦ, к числу которых можно отнести: дату окончания гарантийного срока узла ЭСИ, количество карточек регистрации дефектов изделий, рекламационных актов, сообщений о неисправности, технических актов, так

называемых влияющих и невливающих на эксплуатационную пригодность карточек разрешения.

Еще одной существенной частью системы является отслеживание степени готовности пакетов документов, связанных в технологическом процессе с определенными этапами (рубежами) проектирования и производства. Эта задача предполагает оценивание степени наполненности ХД необходимыми метаданными по различным типам документов, связанным с этапом подготовки конструкторской документации, проектной документации, проведения периодических и типовых испытаний, производственного процесса по крупным составным частям изделия (поддеревьям ЭСИ). Причем готовность всех необходимых пакетов документов и определяет готовность к формированию заявления о соответствии. В то же время в любой момент времени доступна информация о текущем положении дел по контролю выпуска документов в надлежащие сроки (см. рис. 4.6).

Конечной целью ПО ставится автоматическое формирование заявления о соответствии изделия на основе полученных в результате мониторинга ТС данных. Заявление о соответствии является формальным доказательством того, что ТС изделия соответствует заданным ограничениям, а само изделие заданному уровню качества и надежности.

ВЕДОМОСТЬ			
контроля исполнения документов при производстве изделия РН "Союз 2-1в" по состоянию на 26.09.2016			
Мероприятия	С начала производства	За 26.09.2016	
По плану	36	4	
Исполнено	32	1	
Не исполнено	1	0	
Не исполнено			
Наименование документа	Плановая дата представления	Задержка (сут)	Исполнитель
1 Акт о результатах типовых испытаний хвостового отсека блока 1 ступени №	23.09.2016	3	Участок №1.3

Рис. 4.6. Пример электронной карты-ведомости контроля исполнения документов при производстве РН «Союз-2»

4.2. Программный комплекс и порядок автоматизированного онтолого-управляемого проектирования системы информационно-аналитической поддержки жизненного цикла космических средств

4.2.1. Выбор программных средств, удовлетворяющих требованиям решения задачи информационно-аналитической поддержки

Разработка прикладного ПО элементов инфраструктуры ЕВЭП тесно связана с программными компонентами интегрируемых информационных систем (модели данных, протоколы информационного взаимодействия и пр.). В процессе функционирования ведомственных АС возможно изменение характеристик ПО, включение в их состав новых уникальных модулей, использование новых протоколов передачи данных. Кроме того, вновь подключаемые к ЕВЭП ИС, также могут обладать набором новых (ранее неизвестных) программных компонентов, алгоритмов обработки и интерфейсов. Сами бизнес-процессы, составляющие основу обработки и анализа данных о ТС СОТО также могут изменяться. Эта ситуация характерна для задачи управления изменениями.

Поэтому к множеству требований к функциональной и программной архитектуре ЕВЭП добавляется необходимость наличия технологий оперативной и эффективной разработки новых программных решений и сопровождения старых. Использование таких технологий должно опираться на отработанные методики, апробированные инструментальные средства, возможность организации совместной работы программистов и экспертов ПрО, т.е., в конечном счёте, – на автоматизацию процесса разработки прикладного ПО и ИС в целом. Таким образом, ПК САПР ПО должен обеспечивать возможность оперативного изменения спецификаций, повторного синтеза программ и их верификации, возможность наращивания функционала, оперативного подключения новых организационных единиц.

Рассмотренный в разделе 2 модельно-алгоритмический комплекс удовлетворяет этим требованиям с точки зрения реализации принципа сквозного

моделирования на основе системы уточняющих спецификаций. В соответствии с этим принципом имеется возможность уточнять аспекты функционирования СИАП, возвращаясь не к исходному коду (и зачастую даже не к схемам программ или ВМ), а к описанию требований, представленных онтологическими спецификациями.

Однако при этом возникает вопрос о выборе средств проектирования и языков описания онтологий. На сегодняшний день существует большое разнообразие соответствующих инструментов и языков [41, 48, 176]. Наиболее известные и часто используемые из них представлены на рисунке 4.7. Существенными критериями при выборе языка описания онтологий были: формальность языка, широкие выразительные возможности, основанные на количестве доступных диалектов дескрипционной логики, популярность языка и высокая степень его документированности, наличие примеров реализации систем на основе такого языка, авторитет разработчика языка. В соответствии с указанными критериями был выбран OWL, развиваемый консорциумом W3C, как один из наиболее развитых языков и активно используемый в сфере Семантического Web. Элементы теоретического описания онтологий соотносятся с синтаксисом языка OWL так, как это показано в приложении 2.3.8.2. К критериям выбора средств проектирования онтологий можно отнести: возможность графического представления онтологий, наличие реализующих табло-алгоритм для разных диалектов дескрипционных логик решателей, масштабируемость поддерживаемого функционала, возможность визуального проектирования, отечественная или открытая реализация. В соответствии с указанными критериями был выбран ПК Protege, а в качестве решателя использовался Fact++, разработанный на язык C++ и позволяющий быстро и эффективно в автоматическом режиме верифицировать проектируемые онтологии.

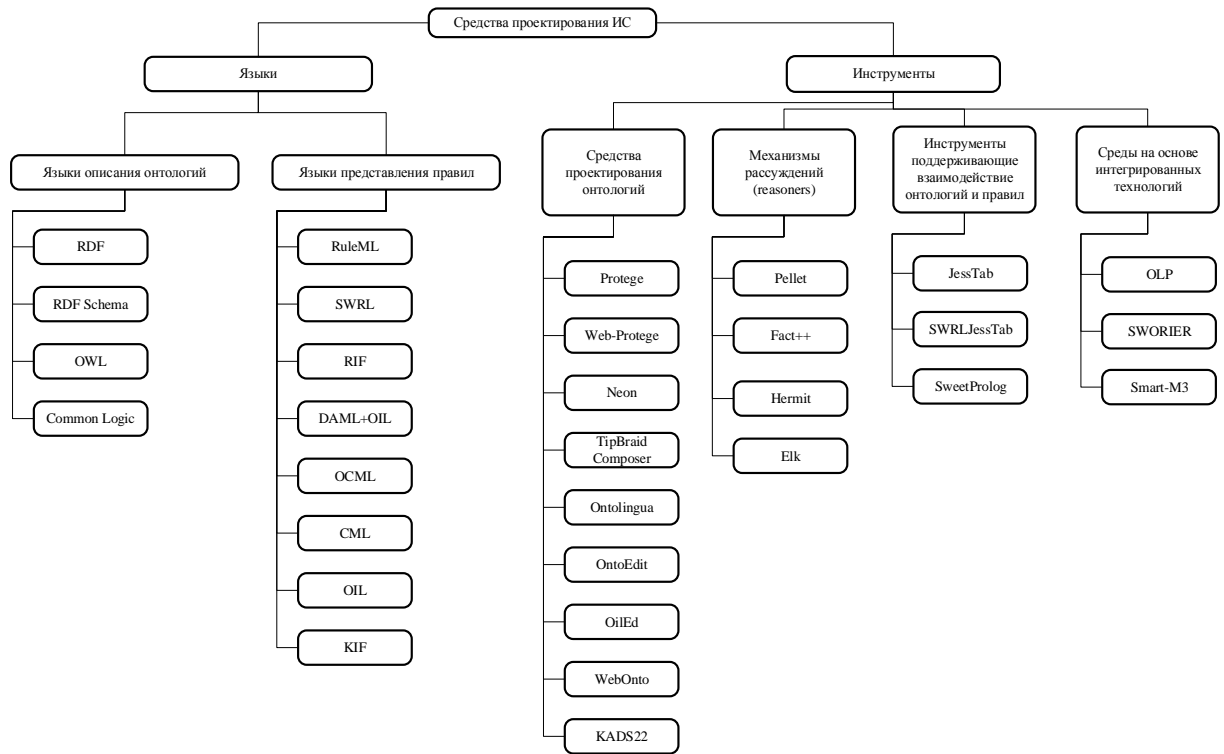


Рис. 4.7. Языки и инструменты разработки онтологических моделей

Для реализации онтологических отображений источников данных, представляющих собой в рассматриваемом случае реляционные БД и XML-файлы использовались соответственно инструменты Ontop и XS2OWL, реализующие методы, описанные в приложении 3 [224, 228]. Заметим, что существует множество систем трансляции с языков XML, HTML и др. на синтаксис RDF, что позволяет использовать онтологические онтобразжения для различных типов (моделей) источников данных.

Кроме того, в связи с тем, что в рассматриваемом модельно-алгоритмическом комплексе используется методология автоматизированного анализа на основе G-моделей, в рамках задач проектирования СИАП использовался ПК, связанный с G-моделями и представленный системой подготовки исходных данных, а также исполнительной системой, позволяющей по спроектированным G-моделям организовывать вычислительный процесс. Таким образом, в совокупности с онтологическими инструментами обобщенная схема компонент ПК представлена на рисунке 4.8 [141]. Полученный ПК был также использован при решении смежных задач [154-157].

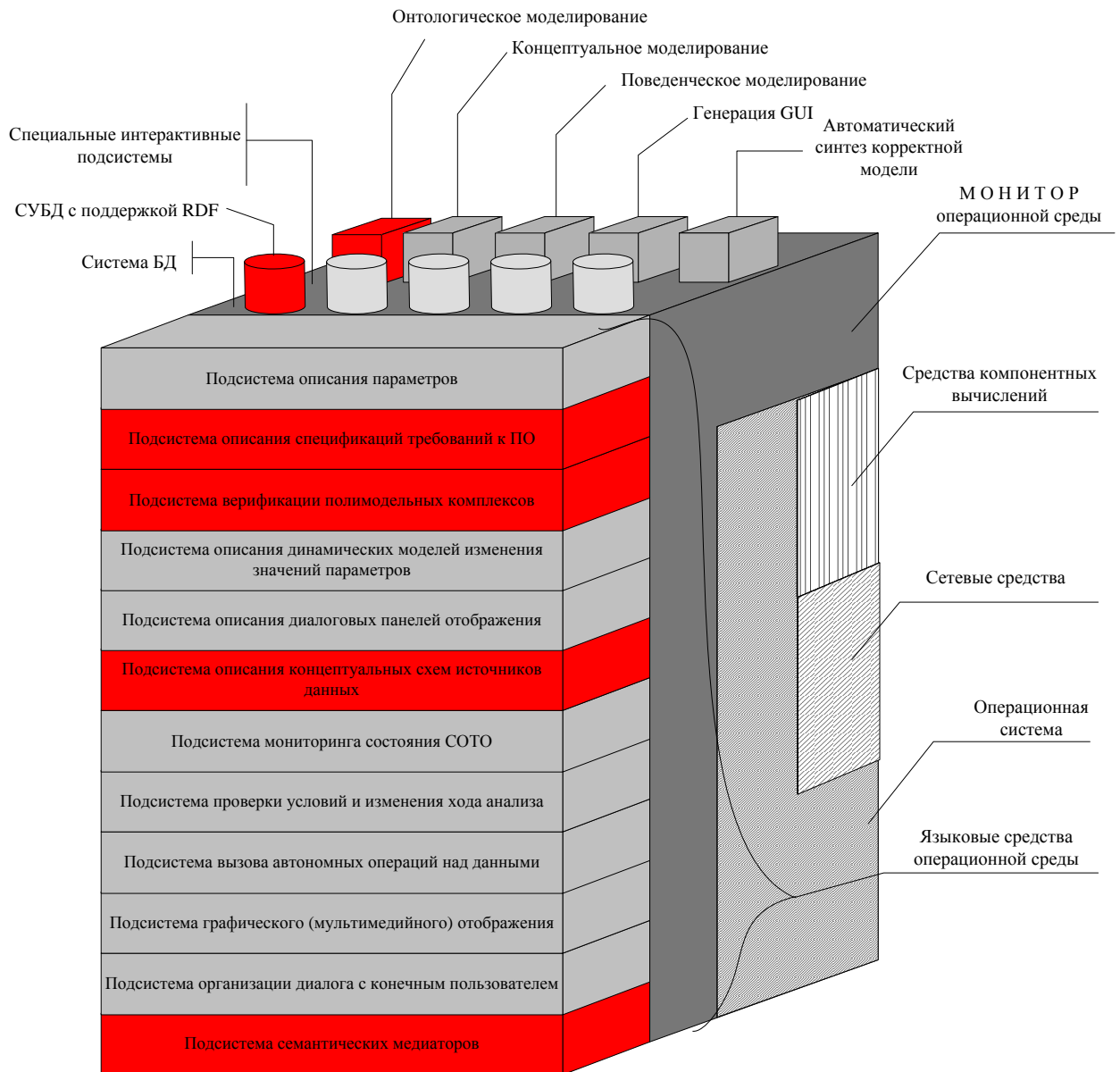


Рис. 4.8. Схема основных компонент ПК проектирования моделей информационно-аналитической поддержки ЖЦ СОТО

4.2.2. Порядок проектирования системы информационно-аналитической поддержки жизненного цикла сложных объектов

Использование САПР ПО, представленной ранее, предполагает при решении задачи формирования ПК СИАП ЖЦ СОТО порядок проектирования, рассмотренный в таблице 4.2.

Таблица 4.2. Основные этапы проектирования СИАП ЖЦ СОТО

Этап №	Содержательное пояснение
1	Разработка бизнес-аналитиками (представителями заказчика) спецификации ПрО верхнего уровня на основе онтологического подхода. Выявление объектов, явлений и процессов в ПрО. Определение словаря ПрО.
2	Разработка бизнес-аналитиками (представителями заказчика) спецификации существующих в действительности бизнес-процессов обработки и анализа информации на основе нотации BPMN исходя из предположения, что задачи, подлежащие автоматизации и выполняемые вручную находятся в разных «дорожках» в терминах BPMN (по ролям исполнения). На данном этапе осуществляется извлечение и формализация знаний о процессах с последующим анализом возможности их автоматизации. Элементы бизнес-процессов имеют только содержательное описание, а сами бизнес-процессы характеризуют структуру событий и действий технологических и организационных процессов, подлежащих автоматизации.
3	Уточнение спецификаций объектов и явлений ПрО с заданием их атрибутов (признаков) и связей между ними специалистами (экспертами) ПрО за счет задания предметных онтологий, характеризующих инфлогическое представление о ПрО как совокупности взаимосвязанных фактов о ПрО. При наличии существующих БД или квазиструктурированных источников данных онтологии строятся в автоматическом режиме как семантические отображения на модели данных, представленные в источниках независимо от реализующей технологии СУБД. При необходимости осуществляется корректировка полученных спецификаций. Отдельно в обязательном порядке задается концептуальная схема электронного паспорта в виде соответствующей предметной онтологии, осуществляется синтез соответствующего RDF-хранилища.
4	Уточнение спецификации бизнес-процессов в виде спецификаций СВЗ для каждого отдельно взятого бизнес-процесса с использованием графической нотации, близкой к IDEF5, в предположении, что каждому событию может быть сопоставлен запрос к СУБД источника данных на основе заданных отображений (предметных онтологий), где уточняются факты из предметных онтологий как ограничения запроса. Каждому действию бизнес-процесса сопоставляется некоторая модель решения вычислительной задачи, при чем параметры запроса являются входными и выходными по отношению к модели решения вычислительной задачи в соответствии со структурой бизнес-процесса. Такая модель СВЗ позволяет осуществить переход от описания ПрО к спецификации создаваемого прикладного ПО, уточняя входные и выходные параметры решаемых задач, порядок и условия их применения, при этом отчуждаясь непосредственно от моделирования вычислений. Совокупность созданных спецификаций образует БЗ рассматриваемой ПрО.
5	По окончании определения поведенческих, функциональных и информационных требований инициализация процесса верификации онтологических спецификаций, осуществляемого в автоматическом режиме и задействующего решатель (reasoner), в основе которого лежат табло-алгоритм, алгоритм проверки онтологической выразительности и пр. При нахождении ошибок осуществляется корректировка спецификаций и повторное их согласование с возвращением на этапы 1, 2, 3, 4.
6	Задание для каждой вычислительной задачи модели её решения в виде соответствующей G-модели, входы и выходы которой заранее predeterminedены в модели СВЗ при том условии, что для каждого элемента-переменной в СВЗ задается соответствующий параметр G-модели. На этом этапе инженеры по знаниям проектируют непосредственно модели решения в терминах параметров и операторов, оперируя словарем и ограничениями, заданными на предыдущих этапах и пользуясь заранее подготовленным набором операторов вычисляемых отношений.
7	В соответствии с имеющимися спецификациями СВЗ и G-моделями, спроектированными на предыдущих этапах, в автоматическом режиме осуществляется синтез структуры обобщенной G-модели для каждого отдельно взятого бизнес-процесса, где для спецификаций запросов создается оператор исполнения запроса, параметры запроса и текст запроса в синтаксисе SPARQL, и далее осуществляется конкатенация (склеивание) полученных элементарных G-моделей.
8	По обобщенным G-моделям в автоматическом режиме осуществляется синтез соответствующей G-сети на основе формального вывода, реализуемого алгоритмом, в основу которого положены r-грамматики. Осуществляется проверка разрешимости имеющихся G-моделей.
9	По окончании проектирования инициализируется процедура верификации полученных G-сетей, осуществляемая в автоматическом режиме. При нахождении ошибок осуществляется корректировка G-моделей с возвращением на этап 6.
10	Полученные программы информационно-аналитической поддержки устанавливаются на АРМ в соответствии с местами их предполагаемой эксплуатации.

Необходимо отметить, что, если моделей бизнес-процессов несколько, и они взаимосвязаны, то для каждой из них создается собственная G-сеть и соответствующая программа, в результате чего формируется сеть интеллектуальных агентов в виде совокупности программ, взаимодействующих по принципу модели классной доски. Запросы на запись в одной программе формируют утверждения для электронного паспорта, которые инициируют запуск ветки вычислительного процесса в другой программе.

При этом, в рамках процесса проектирования, также параллельно выполняются следующие этапы:

- информационное концептуальное моделирование – введение параметров, групп и атрибутов параметров, задание правил сегментации области значений вводимых параметров (для осуществления качественного анализа и перехода от непрерывнозначных показателей свойств данных к дискретным);
- проектирование и генерация графического пользовательского интерфейса, предназначенного для организации диалога с пользователем и визуализации данных и результатов функционирования.

Таким образом, основываясь на указанном порядке проектирования соответствующего полимодельного комплекса, может быть получена СИАП ЖЦ СОТО необходимого вида.

4.3. Некоторые примеры реализации элементов системы информационно-аналитической поддержки жизненного цикла ракеты-носителя «Союз-2»

4.3.1. Пример модели бизнес-процессов формирования конфигурации и заявления о соответствии ракеты-носителя «Союз-2» требованиям заказа

В рамках разработки СИАП этапов проектирования и производства РН "Союз-2" при решении задачи формирования заявления о соответствии на основании регламента выполнения работ отдела контроля качества и ГОСТ 2.102-2013, 1410-002-2010 о документации, выпускаемой в процессе непосред-

ственно проектирования и производства изделия, была разработана спецификация в виде соответствующей модели бизнес-процессов информационно-аналитической поддержки, упрощенный для наглядности вариант которой представлен на рисунке 4.9.

Проиллюстрированная модель демонстрирует процессы появления различных типов документов и действий по их анализу в соответствии с регламентом. Содержательно процесс этапа проектирования описывает порядок формирования электронной структуры изделия на основе проводимого 3D-моделирования РН "Союз-2", порядок проведения анализа по этапности подготовки инструкций по эксплуатации, чертежей, расчетов, технических и технологических условий, связанных с узлами изделия, прочих документов, и, в конечном счете, проекта изделия, декларирующего полную готовность базовой (нулевой) конфигурации к этапу производства.

В свою очередь, процесс этапа производства описывает порядок анализа этапности появления документации о производстве изделия. К числу анализируемых документов можно отнести технические акты, акты сборки, карточки разрешения, сообщения о неисправности, карточки регистрации дефекта, извещения и акты о проведении испытаний и др. Необходимо обратить внимание на тот факт, что выпускаемые на этом этапе документы и их метаданные связываются с узлами электронной структуры изделия, полученной на этапе проектирования. Кроме того, анализ некоторых документов непосредственно связан с наличием и содержанием ряда документов этапа проектирования. Окончание этапа производства характеризуется появлением заявления о соответствии изделия требованиям заказчика.

Таким образом, модель бизнес-процессов содержательно отражает порядок выполнения работ информационно-аналитической поддержки этапов проектирования и производства РН "Союз-2". При этом указанная модель была использована как спецификация требований для СИАП, и на ее основе был разработан соответствующий ПК.

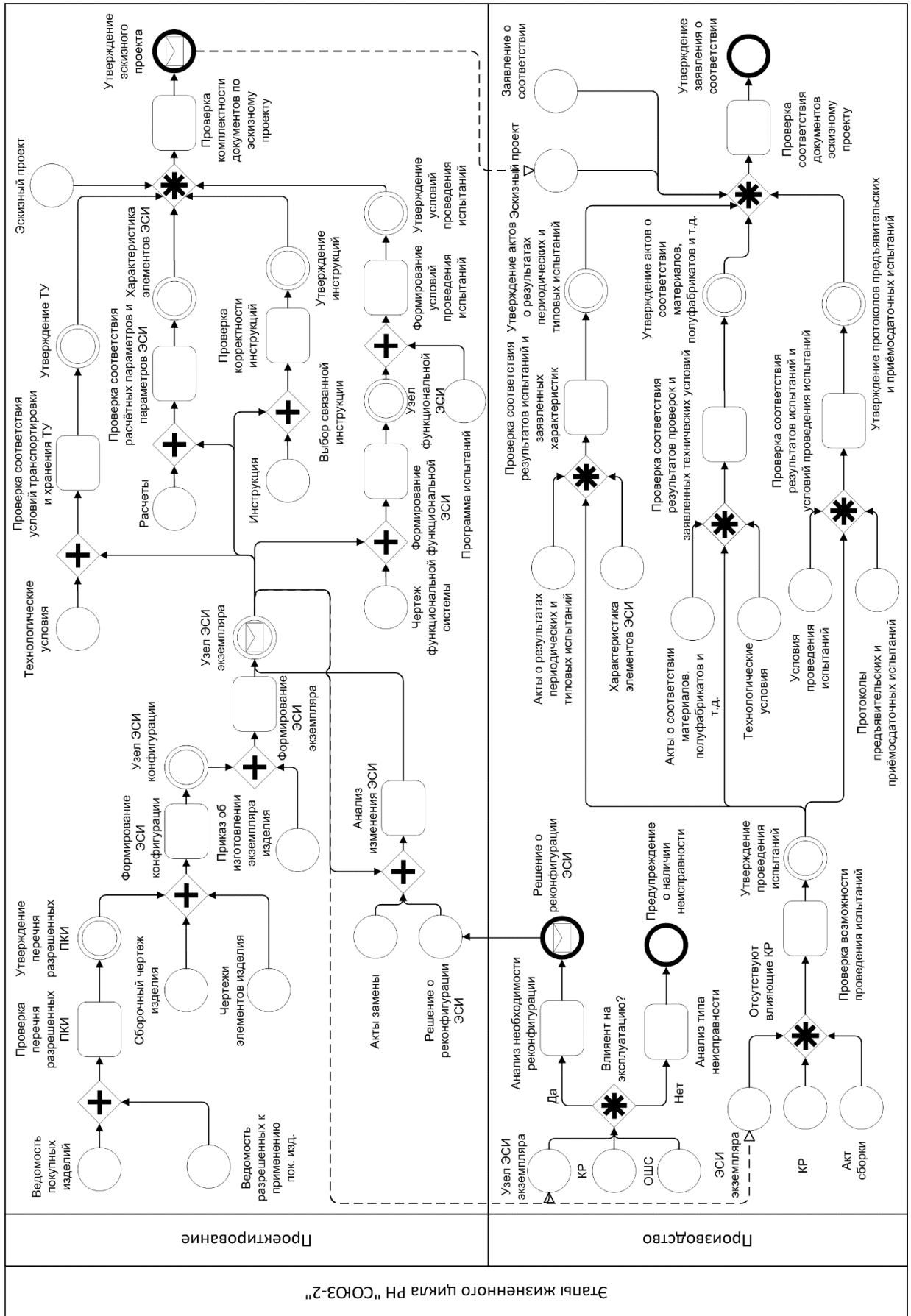


Рис. 4.9. Пример модели бизнес-процессов информационно-аналитической поддержки ЖЦ РН «Союз-2» на этапах проектирования и производства

4.3.2. Пример формирования совокупности моделей информационно-аналитической поддержки ЖЦ РН «Союз-2» для этапов проектирования и производства

Для сохранения наглядности приведем пример моделирования спецификаций и соответствующих ВМ и их схем программ для части модели бизнес-процессов, которая представлена на рисунке 4.10.

Приведенный пример характеризует ситуацию, когда на этапе производства в результате нахождения дефекта произведенного узла или агрегата РН был сформирован документ, называемый "Карточкой разрешения" и отражающий суть дефекта и характер его влияния на эксплуатационную пригодность соответствующей детали изделия. При этом анализ этого документа существенно влияет на дальнейший ход производства, затрачиваемые на разрешение данной ситуации ресурсы и, безусловно, само изделие. Оперативность реакции ЛППР на данную ситуацию также имеет высокую степень важности.

Содержательно бизнес-процесс декларирует два возможных сценария развития ситуации, инициализируемых в зависимости от содержания документа. В одном случае, если документ отражает отсутствие влияния на эксплуатационные характеристики, то в электронном паспорте делается пометка о допущенном отступлении от нормы, и процесс продолжается дальше.

В другом – при существенном влиянии на надежность и качество изделия, делается запись о необходимости замены узла изделия, и отправляется соответствующее сообщение для этапа проектирования, в рамках которого ожидается поступление акта замены детали, подтверждающего её замену в электронной структуре изделия.

Соответствующие уточняющие бизнес-процессы спецификации в виде СВЗ-моделей для проектирования и производства представлены соответственно на рисунках 4.11, 4.12. Сформированные по ним G-модели на рисун-

ках 4.13, 4.14. Необходимо отметить, что некоторые элементы представленных моделей имеют содержательное наименование с целью демонстрации сути моделируемых процессов.

По сформированным G-моделям на основе синтеза их G-сетей могут быть получены соответственно две программы СИАП ЖЦ РН «Союз-2».

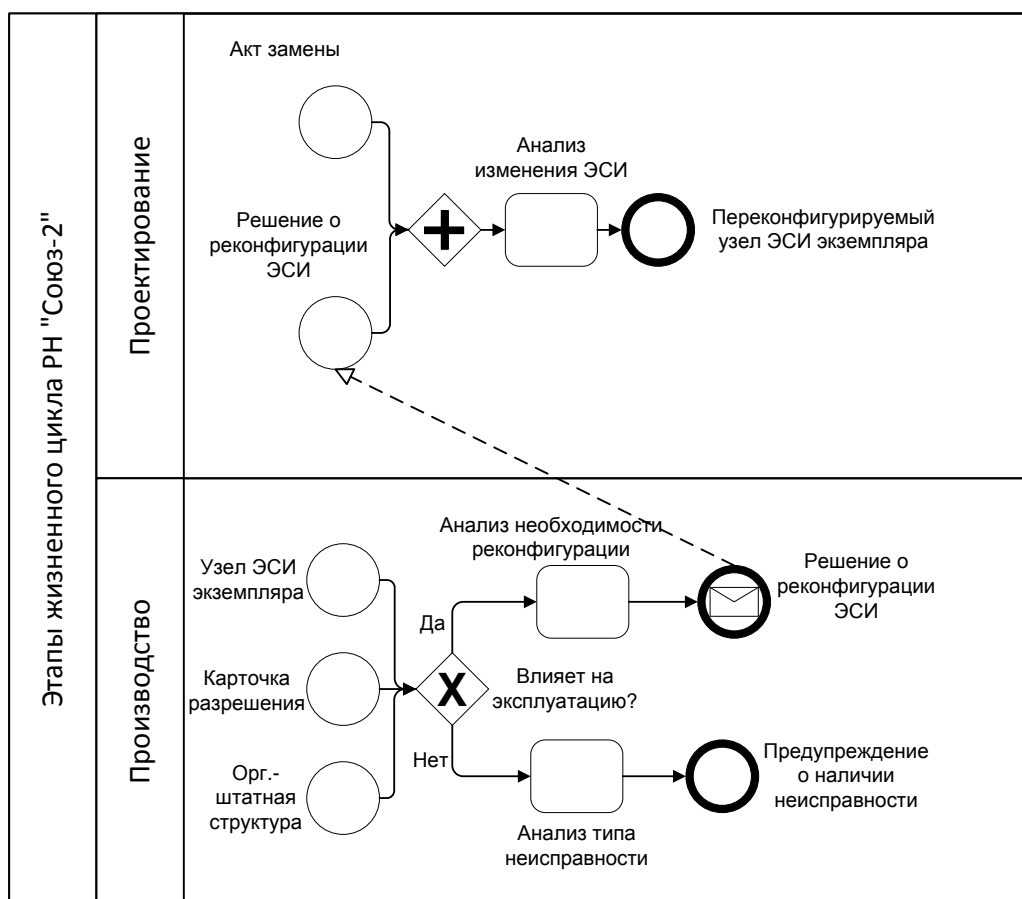


Рис. 4.10. Пример модели бизнес-процесса информационно-аналитической поддержки ЖЦ РН «Союз-2» при обработке метаданных документа «Карточка разрешения»

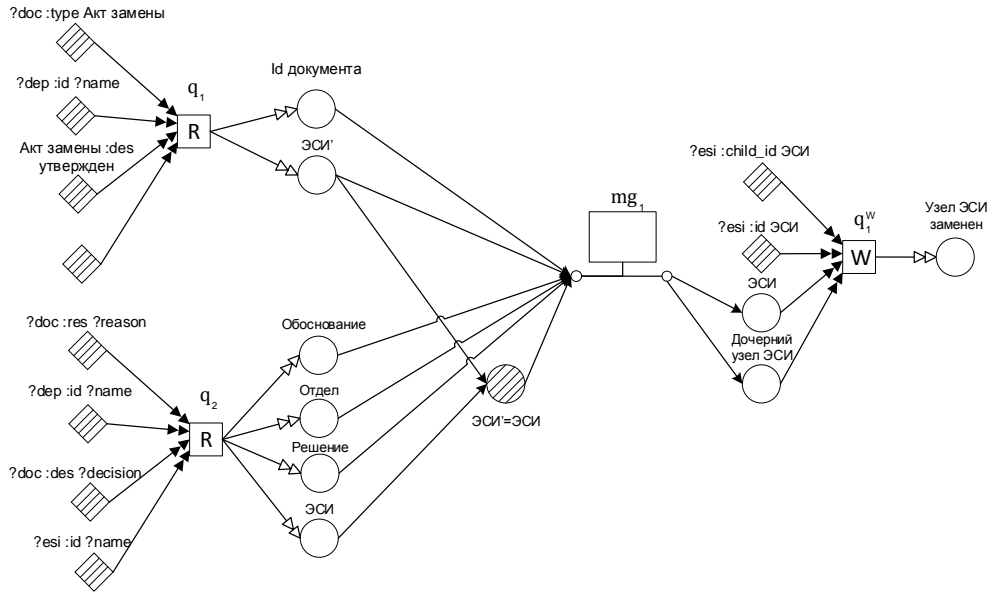


Рис. 4.11. Пример СВЗ-модели анализа изменения конфигурации изделия по метаданным документа «Акт замены»

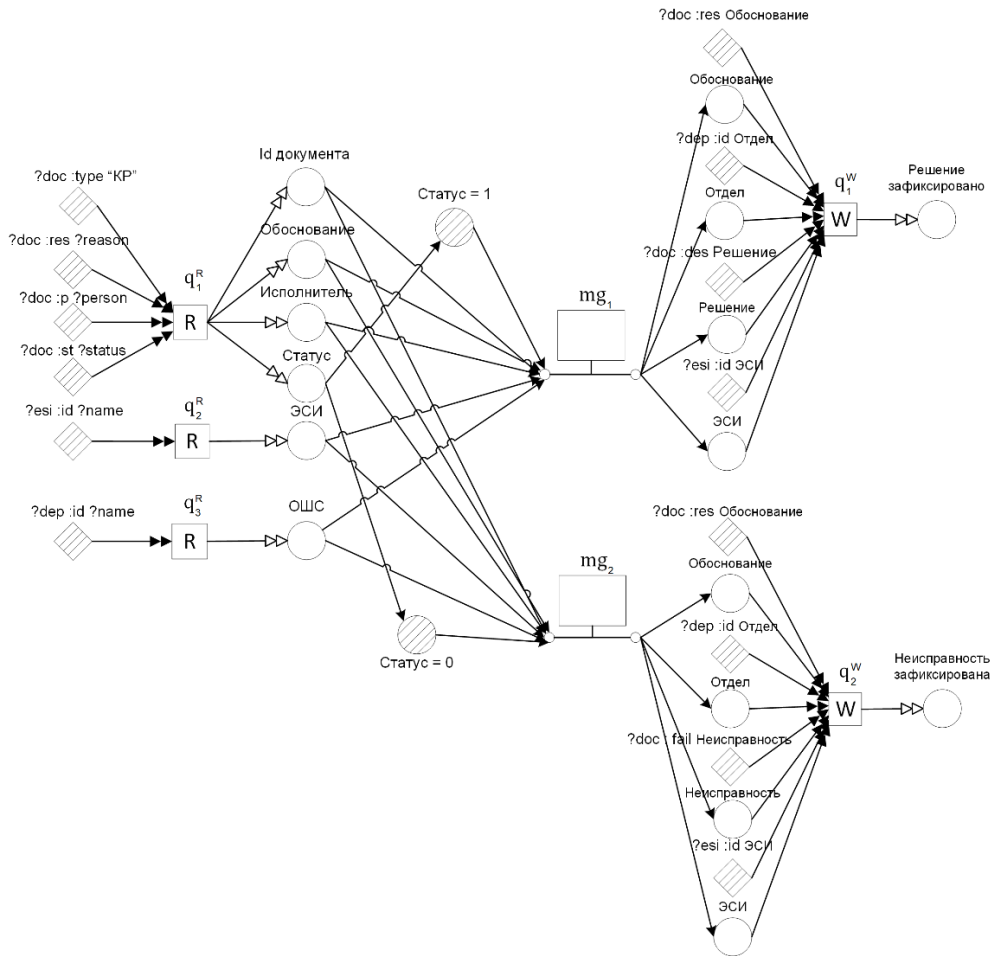


Рис. 4.12. Пример СВЗ-модели оценивания ТС СОТО по метаданным документа «Карточка разрешения»

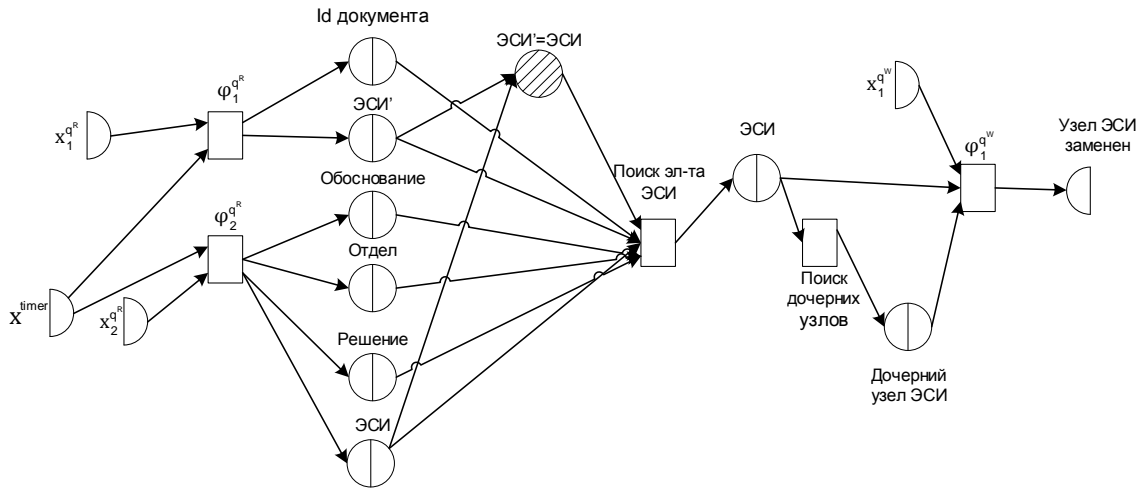


Рис. 4.13. Пример G-модели анализа изменения конфигурации изделия по метаданным документа «Акт замены»

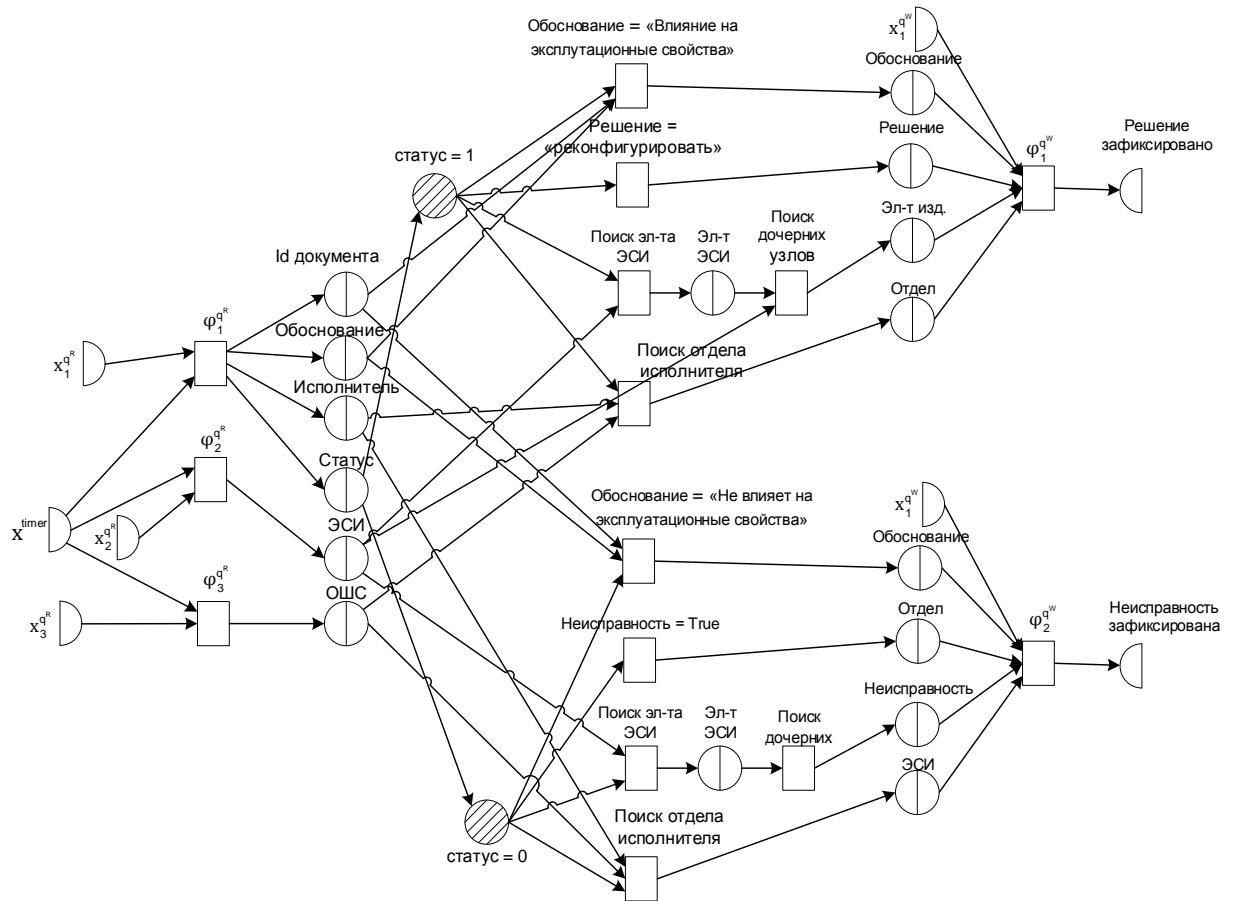


Рис. 4.14. Пример G-модели оценивания ТС COTO по метаданным документа «Карточка разрешения»

4.3.3. Пример онтологических спецификаций источников данных и электронного паспорта «Союз-2»

Информационные ресурсы ЕВЭП могут быть представлены с помощью онтологических графов концептов и их ролей (см. пояснение в приложении 3 и рис. П. 3.1). В соответствии с решаемой задачей в СИАП ЖЦ СОТО для источника данных в виде реляционной БД СУЭД (см. приложение 4.2) было сформировано онтологическое отображение, схема которого представлена на рисунке 4.16 (см. приложение 4.1). В свою очередь, схема электронного паспорта, представленного RDF-хранилищем, продемонстрирована на рисунке 4.17. В качестве примера наполнения данными на рисунке 4.15 приведена схема онтологии для XML-файлов пакета документов, экспортируемого из используемой на этапе проектирования ИС Windchill.

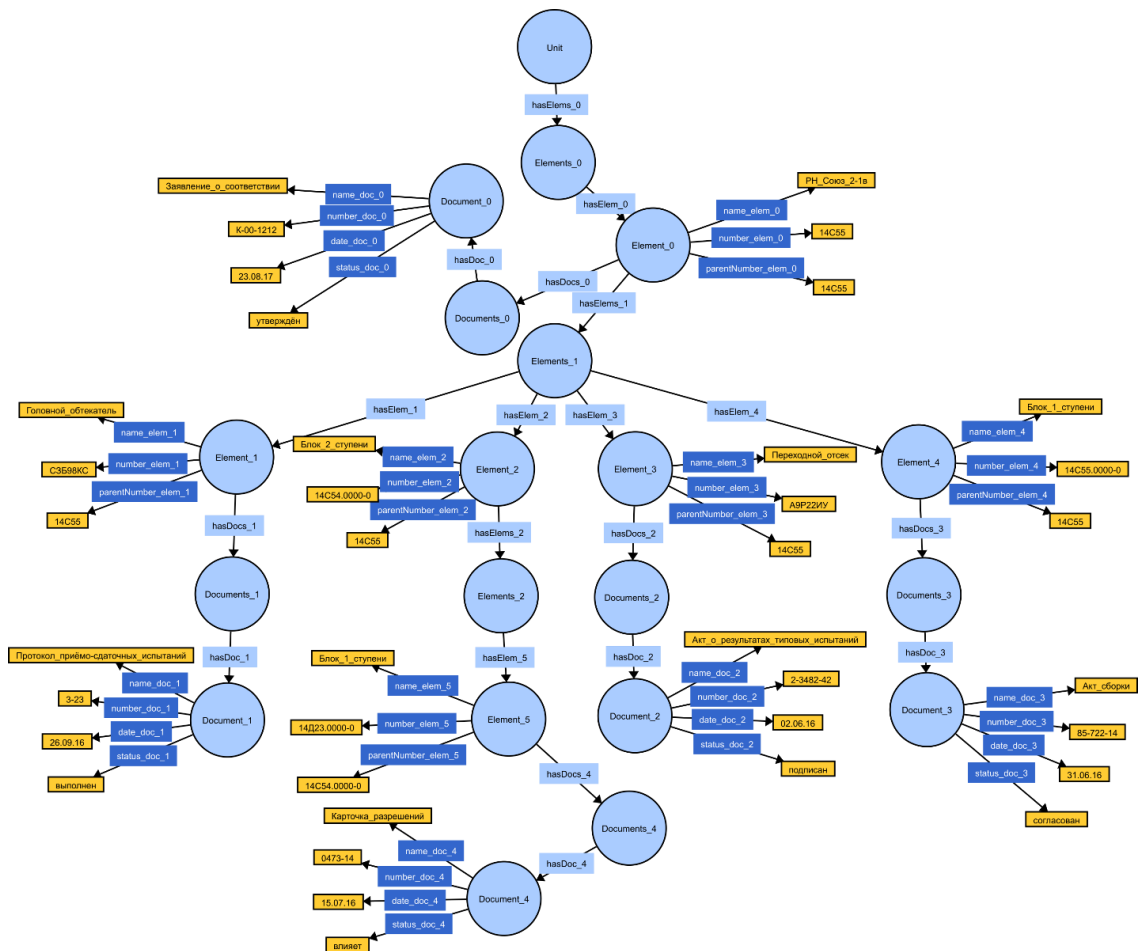


Рис. 4.15. Пример онтологической схемы метаданных об изделии

4.4. Экспериментальные расчеты эффективности применения системы информационно-аналитической поддержки жизненного цикла ракеты-носителя «Союз-2»

В результате внедрения СИАП ЖЦ РН «Союз 2» на предприятия КК при решении задачи формирования заявления о соответствии изделия требованиям заказчика, были проведены экспериментальные расчеты по оцениванию функциональной эффективности информационно-аналитической поддержки ЖЦ рассматриваемого изделия.

Проведение эксперимента заключалось, в частности, в замере временных характеристик функционирования СИАП и выполнения аналитических работ представителями организаций КК на примере информационно-аналитической поддержки ЖЦ одного экземпляра изделия РН «Союз 2-1в» и их последующего сравнения. Так, результаты замеров для четырех последовательных этапов работ над изделием соответственно при максимально допустимых значениях, при выполнении работ представителями организаций КК и СИАП составили (дней):

- этап формирования базовой конфигурации – 8, 7, 1;
- этап подготовки конструкторской и проектной документации – 16, 12, 5;
- этап производства экземпляра изделия – 24, 24, 12;
- этап проведения итоговых испытаний и формирования заявления о соответствии – 32, 28, 17.

Замечание 4.1. При подсчете времени функционирования СИАП использовался рассмотренный ранее способ из раздела 3.5.1, однако в дополнение к рассчитанному времени необходимо добавить время, затрачиваемое на внесение метаданных документов в систему и оцифровку документов, представленных на бумажном носителе.

С учетом этого, результаты замеров приведены на рисунке 4.18.

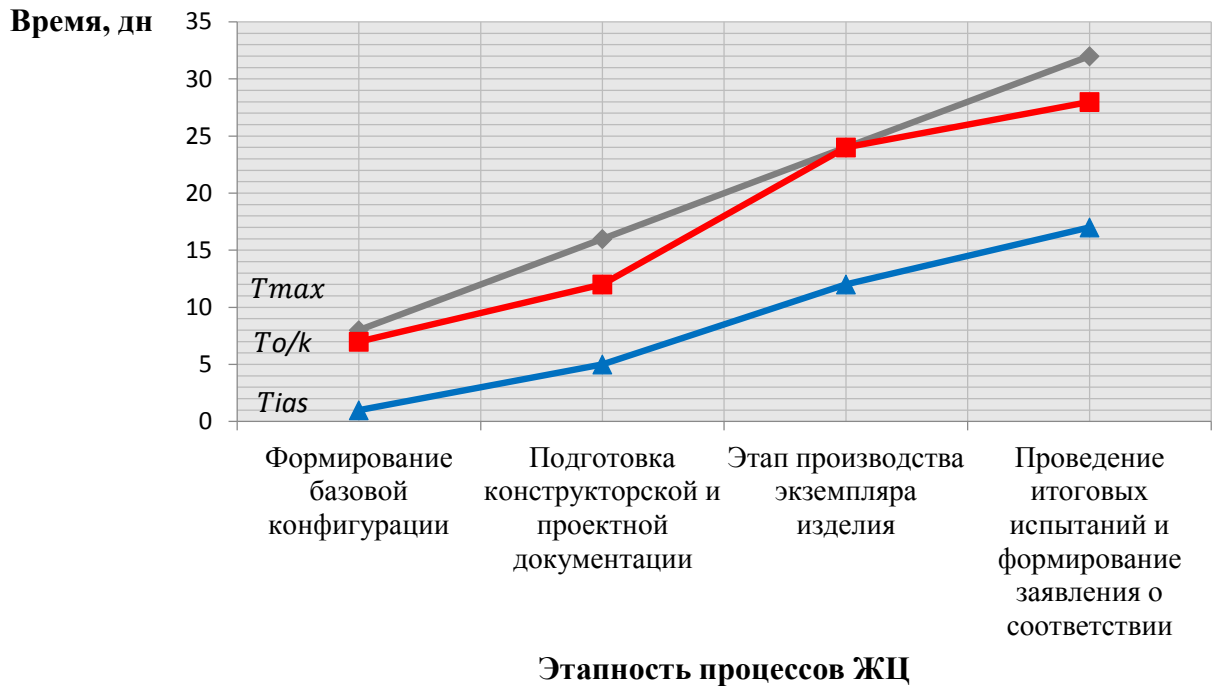


Рис. 4.18. График зависимости времени, затраченного на информационно-аналитическую поддержку от стадии ЖЦ изделия

Тогда для задачи формирования заявления о соответствии оперативность информационно-аналитической поддержки может быть рассчитана согласно выражению (3.25):

- для представителей организаций КК:

$$Eff^{ias} = \frac{32 - 28}{32} \approx 0,125,$$

- для СИАП:

$$Eff^{ias} = \frac{32 - 17}{32} \approx 0,468.$$

При оценивании достоверности результатов выполнения работ представителями организаций КК и СИАП методом экспертного опроса и в результате подсчета было получено количество информационных единиц как исходных данных для информационно-аналитической поддержки на разных этапах ЖЦ изделия, что продемонстрировано на рисунке 4.19, где в тройках столбцов правый отражает объем данных, учтенный представителями организаций КК,

средний – объем данных СИАП, полученный из источников данных, левый – общий объем данных, который необходимо учесть на каждом этапе ЖЦ.



Рис. 4.19. Столбчатая гистограмма сравнения количества оцениваемых информационных единиц на разных этапах ЖЦ изделия

В то же время в результате расчетов показателей, предложенных в разделах 3.2, 3.3, 3.4 была подтверждена согласованность спецификаций бизнес-процессов (из рис. 4.9), их СВЗ и соответственно их G-моделей и G-сетей, реализуемость ЕИП в СИАП, корректность схем программ. В связи с отсутствием возможности объективно оценить эти показатели для существующей технологии выполнения работ представителями организаций КК, их значения были получены в результате экспертного опроса. Так, общий показатель согласованности был задан значением 0,7, общий показатель корректности – 0,8. Относительно реализуемости ЕИП была подтверждено наличие неизменчивости хранилища паспорта изделия и предметной ориентированности процессов информационно-аналитической поддержки. В соответствии с выражением (3.16) и заданными весами $w_1 = 0,4$, $w_2 = 0,3$, $w_3 = 0,2$ и $w_4 = 0,1$ показатель

реализуемости ЕИП для существующей технологии может быть рассчитан следующим образом:

$$F^{ds} = 0,4 * 1 + 0,3 * 0 + 0,2 * 1 + 0,1 * 0 = 0,6.$$

В соответствии с полученными результатами, очевидно, показатели реализуемости ЕИП, корректности программ и согласованности спецификаций равны 1 для СИАП ЖЦ СОТО. При этом в работе сделано допущение о том, что исходные данные являются достоверными, а вопрос степени их достоверности будем рассмотрен в рамках дальнейших исследований. Тогда, для задачи формирования заявления о соответствии достоверность результатов информационно-аналитической поддержки может быть рассчитана согласно выражению (3.26) с учетом заданных в результате экспертного опроса весовых коэффициентов:

- для представителей организаций КК:

$$Ver^{ias} = 0,25 * 1 + 0,35 * \frac{2389}{2475} + 0,2 * 0,8 + 0,2 * 0,7 \approx 0,88,$$

- для СИАП:

$$Ver^{ias} = 0,25 * 1 + 0,35 * \frac{2461}{2475} + 0,2 * 1 + 0,2 * 1 \approx 0,99.$$

В свою очередь, уровень информационно-аналитической поддержки в соответствии с (3.18) и назначенными экспертным опросом весами $w_1 = 0,6$ и $w_2 = 0,4$ для задачи формирования заявления о соответствии равен:

- для представителей организаций КК:

$$Q^{ias} = 0,6 * 0,88 + 0,4 * 0,125 \approx 0,578,$$

- для СИАП:

$$Q^{ias} = 0,6 * 0,99 + 0,4 * 0,468 \approx 0,781.$$

Таким образом, при применении СИАП достигается повышение уровня информационно-аналитической поддержки на 20%.

Тогда, показатель функциональной эффективности информационно-аналитической поддержки ЖЦ РН «Союз 2-1в» в задаче формирования заявления о соответствии согласно выражению (3.27) и назначенными экспертным опросом весами $w_1 = 0,7$, $w_2 = 0,3$ равен:

- для представителей организаций КК:

$$E^{ias} = 0,7 * 0,578 + 0,3 * 0,6 \approx 0,584,$$

- для СИАП:

$$E^{ias} = 0,7 * 0,781 + 0,3 * 1 \approx 0,846.$$

Совокупность показателей, характеризующих эффективность информационно-аналитической поддержки ЖЦ РН «Союз 2-1в» в сравнении двух рассматриваемых способов представлена на рис. 4.20.

Таким образом, из рис. 4.20 и представленных выше расчетов видно, что в результате внедрения СИАП достигается повышение функциональной эффективности информационно-аналитической поддержки ЖЦ РН «Союз 2-1в» на 26%.

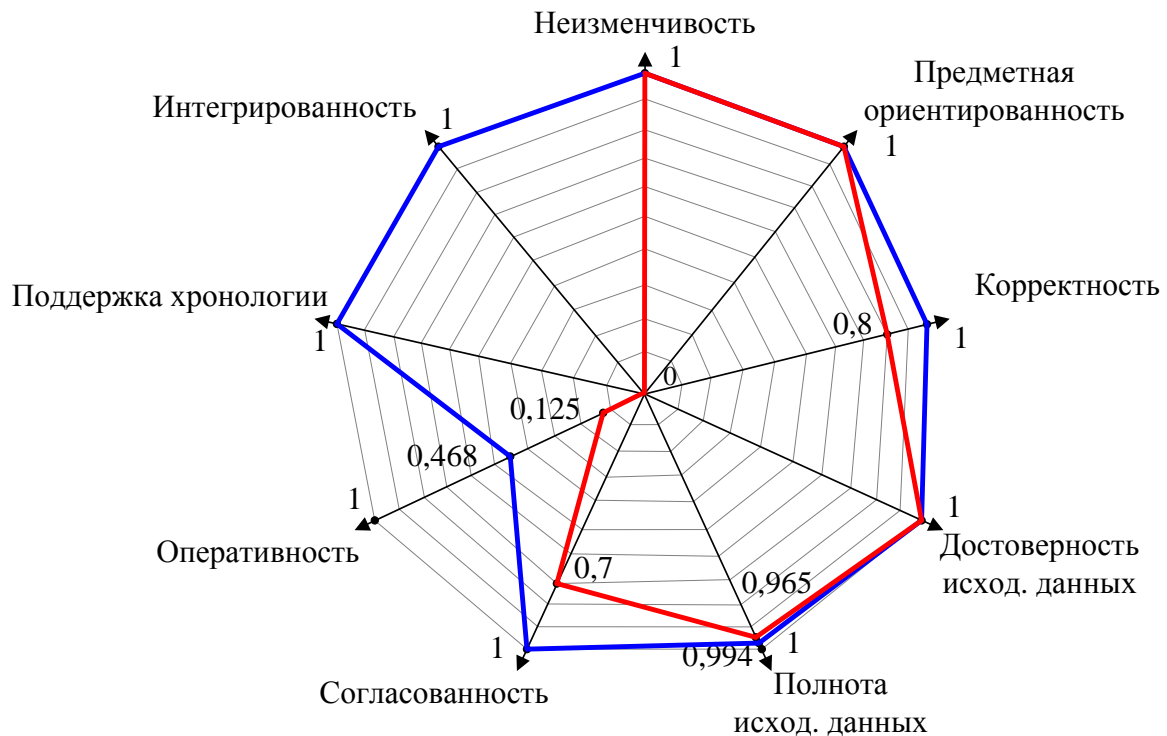


Рис. 4.20. Лепестковая диграмма сравнения эффективности рассматриваемых способов информационно-аналитической поддержки

4.5. Выводы по разделу

1. В соответствии с предложенным в работе модельно-алгоритмическим комплексом был разработан и внедрен на предприятия КК ПК информационно-аналитической поддержки ЖЦ РН «Союз-2», охватывающий этапы проектирования и производства изделия от формирования его базовой конфигурации и вплоть до формирования заявления о соответствии экземпляра изделия требованиям, предъявляемым заказчиками.
2. Практическое применение результатов теоретических исследований задачи информационно-аналитической поддержки ЖЦ РН «Союз-2» позволяет сделать вывод о том, что сформированный модельно-алгоритмический комплекс, созданный ПК автоматизированного онтолого-управляемого проектирования, соответствующая реализующая система позволяют создать СИАП ЖЦ РН «Союз-2», отвечающую представленным в диссертационной работе требованиям. При этом разработанные модели и алгоритмы могут быть применены и в других ПрО благодаря ориентации на информационно-аналитическую поддержку изделий как СОТО с использованием перспективных ИИТ: экспертному комплексному моделированию, онтологическому моделированию, концептуальному программированию и др.
3. Проведены экспериментальные расчеты по оцениванию функциональной эффективности применения СИАП ЖЦ РН «Союз-2» на основе разработанной структуры показателей, указывающие на то, что при использовании разработанного ПО, соответствующего предъявляемым в диссертационном исследовании требованиям, достигается существенное повышение функциональной эффективности информационно-аналитической поддержки ЖЦ изделия РН «Союз-2» на 26%. При этом наблюдается сокращение времени, затрачиваемого на информационное взаимодействие задействованных организационных единиц, повышается оперативность проводимого анализа и достоверность его результатов.

ЗАКЛЮЧЕНИЕ

В диссертационной работе решена важная научно-техническая задача автоматизации информационно-аналитической поддержки процессов создания и применения КСр. Полученные результаты имеют существенное значение для совершенствования системы информации о техническом состоянии и надежности космических комплексов и входящих в их состав изделий в ракетно-космической отрасли РФ. В рамках исследования **получены следующие результаты:**

- разработана онтологическая система, позволяющая на конструктивной основе специфицировать и совместно учитывать информационные, поведенческие и функциональные требования, предъявляемые к облику СИАП ЖЦ КСр;
- разработана онтологическая МПЗ о бизнес-процессах информационно-аналитической поддержки ЖЦ КСр, позволяющая формализовать разноаспектные знания аналитиков и экспертов ПрО об организационных и технологических процессах этапов ЖЦ КСр;
- разработана онтологическая МПЗ о СВЗ, уточняющая спецификации бизнес-процессов в виде сценариев интеграции данных на основе семантических онтологических отображений и порядка их обработки и анализа с использованием принципов OLAP-систем, ETL и OBDA;
- разработан алгоритм структурно-параметрического синтеза ВМ в виде G-моделей, обеспечивающий в автоматизированном режиме переход от специфицирования требований к конструктивизму задания отношений вычислимости и получению схем программ, управляющая и информационная структура которых позволяет организовать параллельные потоковые и асинхронные вычисления по ним;
- разработан алгоритмический комплекс верификации предложенных моделей и связанная с ним методика, позволяющие оценить согласованность онтологических моделей, реализуемость ЕИП, корректность программ СИАП,

функциональную эффективность информационно-аналитической поддержки ЖЦ КСр.

Результаты диссертационного исследования позволяют получить следующий **практический эффект** при решении задачи автоматизации информационно-аналитической поддержки ЖЦ КСр:

- повышение функциональной эффективности информационно-аналитической поддержки ЖЦ КСр на 26% и совместно с ним повышение оперативности информационно-аналитической поддержки и достоверности её результатов;
- применение ЭВМ на основе автоматизированного проектирования уже на этапе постановки задачи и конструктивная декомпозиция этапов разработки ПО с последовательным уточнением моделей без перехода на алгоритмический уровень;
- повышение степени соответствия разрабатываемого ПО исходным требованиям и постановке задачи, а также достоверности результатов его функционирования на основе формальных спецификаций;
- потенциальное снижение стоимости, сроков разработки и сложности масштабирования и модифицирования ПО информационно-аналитической поддержки ЖЦ СОТО;
- конструктивизация процессов извлечения и представления экспертных знаний на основе использования графических нотаций, совместного представления бизнес-процессов, декомпозиции процесса проектирования, использования полимодельного и многоязычного принципов, комплекса средств верификации моделей.

Рекомендации и перспективы дальнейшей разработки темы. Совершенствование и разработка новых методов комплексного моделирования экспертных знаний на основе комбинированных моделей с использованием онтологий и синтеза оптимальных программ при решении прикладных задач автоматизации управления ЖЦ СОТО в условиях неопределенности и стохастичности, существенной структурной динамики исследуемых объектов.

СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

АПК	Аппаратно-Программный Комплекс
АРМ	Автоматизированное Рабочее Место
АС	Автоматизированная Система
АСУ	АС Управления
АСУ П	АСУ Предприятием
АСУ ТП	АСУ Технологическими Процессами
БД	База данных
БЗ	База знаний
ВМ	Вычислительная Модель
ДНФ	Дизъюнктивно-Нормальная Форма
ЖЦ	Жизненный Цикл
ИАС	Информационно-Аналитическая Система
ИИ	Искусственный Интеллект
ИИС	Интегрированная Информационная Среда
ИИТ	Интеллектуальные Информационные Технологии
ИПИ	Информационная Поддержка Изделия
ИС	Информационная система
ИТ	Информационные Технологии
КК	Космический комплекс
КСр	Космическое средство
МАС	МультиАгентные Системы

МПЗ	Модель Представления Знаний
ООП	Объектно-ориентированная парадигма
ПИ	Параллельный Инжиниринг
ПО	Программное Обеспечение
ППР	Поддержка Принятия Решений
ПрО	Предметная Область
РВ	Реальное Время
РКТ	Ракетно-Космической Техника
РН	Ракета-Носитель
САПР	Система Автоматизированного Проектирования
СВЗ	Согласованные Вычислительные Задачи
СДФ	Совершенная ДНФ
СОА	Сервис-Ориентированная Архитектура
СОТО	Сложный Организационно-Технический Объект
СПМ	Структурно-Потоково-Многоуровневое (распознавание)
СУБД	Система Управления БД
СУЭД	Система Управления Электронными Данными
ТЗ	Техническое Задание
ТС	Техническое Состояние
УДИ	Управление данными об изделии
ХД	Хранилище Данных
ЭВМ	Электронно-Вычислительная Машина
ЭС	Экспертная Система

ЯЗ	Язык Запросов
ЯМД	Язык Манипулирования Данными
ЯОД	Язык Описания Данных
ЯПЗ	Язык Представления Знаний
BI	Business Intelligence
BPMN	Business Process Modeling Notations
CAD	Computer Aided Design
CAE	Computer Aided Engineering
CALS	Continuous Acquisition and Life cycle Support
CASE	Computer-Aided Software Engineering
CAM	Computer Aided Manufacturing
CNC	Computer Numerical Control
CPC	Collaborative Product Commerce
CRM	Customer Requirement Management
CSM	Component Supplier Management
EAI	Enterprise Application Integration
EII	Enterprise Information Integration
EDPC	Event-Driven Process Chain
ERP	Enterprise Resource Planning
ETL	Extraction – Transformation – Loading
FIFO	First In – First Out
FTP	File Transfer Protocol
GBWL	Graph-Based Workflow Language

IDEF	Integrated Computer Aided Manufacturing
JSON	JavaScript Object Notation
MES	Manufacturing Execution Systems
MRP	Manufacturing Requirement Planning
OBDA	Ontology-Based Data Access Systems
ODBC	Open Database Connectivity
OLAP	On-Line Analytical Processing
OMG	Object Management Group
OOSE	Object-Oriented Software Engineering
OWL	Ontology Web Language
PDM	Product Data Management
RDF	Resource Description Framework
SADT	Structured Analysis and Design Technique
S&SM	Sales and Service Management
SCADA	Supervisory Control and Data Acquisition
SCM	Supply Chain Management
SDL	Specification and Description Language
SKBL	Scenario Knowledge Base Language
SQL	Structured Query Language
STEP	Standard for The Exchange of Product model data
TQM	Total Quality Management
UML	Unified Modeling Language
WebDAV	Web Distributed Authoring and Versioning

WfMC	Workflow Management Coalition
WPDL	Workflow Process Definition Language
XML	eXtensible Markup Language
YAWL	Yet another Workflow Language

СПИСОК ЛИТЕРАТУРЫ

1. *Автамонов П.Н.* и др. Применение технологии поддержки принятия решений на различных этапах жизненного цикла космических средств в составе системы информации о техническом состоянии и надежности / *Автамонов П.Н., Бахмут А.Д., Крылов А.В., Охтилев М.Ю., Охтилев П.А., Соколов Б.В.* // Материалы V Всероссийской научно-технической конференции «Актуальные проблемы ракетно-космической техники» (V Козловские чтения); под общ. ред. А.Н. Кирилина/СамНЦ РАН – Самара: ЦСКБ-Прогресс, 2017. С. 222-233.
2. *Автамонов П.Н., Охтилев М.Ю., Соколов Б.В., Юсупов Р.М.* Актуальные научно-технические проблемы разработки и внедрения взаимосвязанного комплекса унифицированных интегрированных систем поддержки принятия решений (СППР) в АСУ объектами военно-государственного управления // Известия Южного федерального университета. Технические науки. Вып. №3(152) – Ростов-на-Дону: Южный федеральный университет, 2014. С. 14-27.
3. *Айзерман М.А., Браверман Э.М., Розоноэр Л.И.* Метод потенциальных функций в теории обучения машин. Главная редакция физико-математической литературы изд-ва «Наука», М., 1970. - 384с.
4. *Акимов А.А.* Проектирование информационно-аналитической системы мониторинга на базе технологий OLAP и Data mining // Надежность и качество: Тр. XVI Междунар. симп. Т. 1. – Пенза: Изд-во ПГУ, 2011. С. 235-239.
5. *Александрян Р.А., Мирзаханян Э.А.* Общая топология: учебное пособие для вузов. – М.: Высшая школа, 1979. – 336 с.
6. *Андреев Д.А., Воронов М.В.* Метод построения онтологии технологических действий // Вестник СГТУ. No 3 (67). – Саратов: СГТУ, 2012. С. 160 – 168.
7. *Андрюшкевич С.К., Ковалев С.П.* Интеллектуальный мониторинг распределенных технологических объектов с использованием информационных моделей состояния // Известия Томского политехнического университета. Инжиниринг георесурсов. – Томск: ГОУ ВПО ТПУ, 2010. С.35-39.

8. *Атаева О.М., Серебряков В.А.* Основные понятия формальной модели семантических библиотек и формализация процессов интеграции в ней // Программные продукты и системы. № 4 (112). – Тверь: НИИ ЦПС, 2015. С. 180-187.
9. *Ахметов Р.Н., Васильев И.Е., Капитонов В.А.* Концепция создания и применения перспективной АСУ подготовки и пуска ракеты космического назначения "Союз-2": новые подходы к интеграции, интеллектуализации, управлению / Ахметов Р.Н., Васильев И.Е., Капитонов В.А., Охтилев М.Ю., Соколов Б.В. // Авиакосмическое приборостроение. № 4. – М: ООО «Научтехлитиздат», 2015. С. 3-54.
10. *Бардзинь Я.М.* Язык спецификаций SDL/PLUS и его применения / Бардзинь Я.М., Калкинъш А.А., Стродс Ю.Ф., Сыцко В.А. – Рига, 1988. 313 с.
11. *Барсегян А.А.* и др. Методы и модели анализа данных: OLAP и Data Mining / Барсегян А.А., Куприянов М.С., Степаненко В.В., Холод И.И.. – СПб: БХВ-Петербург, 2004. 336 с.
12. *Бахмут А.Д., Крылов А.В., Охтилев П.А.* Подход к организации интеллектуальной системы поддержки принятия решений по управлению сложной организационно-технической системы на основе модифицированной сети Петри // Проблемы управления и моделирования в сложных системах: Труды XIX Международной конференции / Под ред.:акад. Е.А.Федосова, акад. Н.А.Кузнецова, проф.В.А.Виттиха. – Самара: ООО «Офорт», 2017. С.379-386.
13. *Бахмут А.Д., Крылов А.В., Охтилев П.А., Кириллов И.С.* Применение модифицированной сети Петри при решении задачи автоматизированной поддержки принятия решений по управлению сложной организационно-технической системы // Научная сессия ГУАП: сб. докл.: В 3 ч. Ч. II. Технические науки / СПб.: ГУАП. СПб., 2017. С.185-191.
14. *Бахмут А.Д., Крылов А.В., Охтилев П.А., Охтилев М.Ю.* Концептуальная модель процессов манипулирования данными при проектировании интеллектуальных информационных систем // Научная сессия ГУАП: сб. докл.: В 3 ч. Ч. II. Технические науки / СПб.: ГУАП. СПб., 2018. С. 228-231.

15. *Бахмут А.Д., Крылов А.В., Охтилев П.А., Соколов Б.В.* Модели представления знаний при проактивном управлении сложными организационно-техническими объектами // Научная сессия ГУАП: сб. докл.: В 3 ч. Ч. II. Технические науки / СПб.: ГУАП. СПб., 2018. С. 232-241.
16. *Бахмут А.Д., Крылов А.В., Охтилев П.А., Соколов Б.В.* Системный анализ моделей и методов, используемых при автоматизированном проектировании информационно-аналитических систем поддержки жизненного цикла космических комплексов и их изделий// Научная сессия ГУАП: сб. докл.: В 3 ч. Ч. II. Технические науки / СПб.: ГУАП. СПб., 2018. С. 228-231.
17. *Бездушный А.А.* Математическая модель системы интеграции данных на основе онтологии // Журнал «Вестник НГУ», серия «Информационные технологии» — Новосибирск, 2008. - Т.6, вып.2. - С 15-40.
18. *Бездушный А. Н.* и др. Интегрированная система информационных ресурсов РАН и технология разработки цифровых библиотек / А.Н. Бездушный, А.Б. Жижченко, М.В. Кулагин, В.А. Серебряков // Программирование. №4. – М.: НИИ механики МГУ, 2000. С. 3-14.
19. *Бездушный А.Н.* и др. Предложения по наборам метаданных для научных информационных ресурсов / Кулагин М.В., Серебряков В.А., Бездушный А.А., Нестеренко А.К., Сысоев Т.М // Журнал «Вычислительные Технологии». Т. 10, Вып. 7. – Новосибирск, 2005. – С. 29-48.
20. *Белов В.С.* Информационно-аналитические системы. Основы проектирования и применения: учебное пособие, руководство, практикум. – М.: Московский государственный университет экономики, статистики и информатики, 2005. 111 с.
21. *Биряльцев Е.В., Гусенков А.М.* Интеграция реляционных баз данных на основе онтологий // Учетные записки Казанского государственного университета. Серия Физико-математические науки. Т. 149, кн. 2. – Казань, 2007. С. 13-34.

22. *Бистерфельд О.А.* Методология функционального моделирования IDEF0 : учебно-методическое пособие / О.А. Бистерфельд ; Ряз. гос. ун-т им. С.А. Есенина. — Рязань, 2008. — 48 с.
23. *Божинский И.А.* Методы и технологии интеграции информационных систем и распределенных баз данных // Радиоэлектроника и информатика. №2. — Харьков: Харьковский национальный университет радиоэлектроники, 2015. С. 30-36.
24. *Бова В.В.* Онтологическая модель интеграции данных и знаний в интеллектуальных информационных системах // Вычислительная техника и информатика, Известия ЮФУ. Технические науки. — Ростов-на-Дону: ЮФУ, 2015. С. 225 – 237.
25. *Боргест Н.М.* Онтология проектирования: теоретические основы. Часть 1. Понятия и принципы. Учеб. пособие. / Н.М. Боргест. — Самара: Изд-во СГАУ, 2010. 92 с.
26. *Бородаенко Д.С.* Отображение реляционных данных на семантическую модель RDF при помощи динамического преобразования запросов // Доклады БГУИР. №2 (48). — Минск: БГУИР, 2010. С.84 – 89.
27. *Бубарева О.А., Попов Ф.А.* Математическая модель процесса интеграции информационных систем на основе онтологий // Современные проблемы науки и образования. 32. — Пенза: Издательский Дом "Академия Естествознания", 2012. 263 с.
28. *Бурлуцкий С. Г., Езерский В. В., Хахаев И. А.* Электронный паспорт как основа информационного обеспечения автоматизированных систем поддержки принятия решений // Информационно-управляющие системы, № 1, 2015. С.100-104.
29. *Бурматов С.В.* Информационная поддержка жизненного цикла изделий как основа системы менеджмента безопасности авиационной деятельности авиационного комплекса России // Научный вестник МГТУ ГА. №178. —М.: Моск. гос. техн. ун-т ГА, 2012. С. 65-70.

30. Буч Г., Рамбо Д., Якобсон И. Язык UML. Руководство пользователя. 2-е изд.: Пер. с англ. Мухин Н. – М.: ДМК Пресс, 2006. 496 с.
31. В.Н. Калинин. Теоретические основы системных исследований. Учебник. - СПб.ВКА имени А.Ф. Можайского, 2016. - 293 с.
32. Вапник В.Н., Червоненкис А.Я. Теория распознавания образов (статистические проблемы обучения), М.: Наука, 1974. – 416 с.
33. Вахитов, А.Р., Новосельцев, В.Б. Преимущества дескриптивной логики при обработке знаний / А.Р. Вахитов, В.Б. Новосельцев // Известия Томского политехнического университета. 2008. -Т. 313. -№ 5. -С. 73-76.
34. Вендоров А.М. CASE-технологии. Современные методы и средства проектирования информационных систем. – М.: Финансы и статистика, 1998. 176с.
35. Верхаген К., Дёйн Р., Трун Ф. Распознавание образов: состояние и перспективы. Пер. с англ. М.:Радио и связь, 1985. -104 с.
36. Вишняков В.А., Бородаенко Д.С. Метод семантического доступа к данным на основе отображения реляционных БД на модель RDF // Экономика и управление. № 2. – СПб: ЧОУ ВО СПУТУЭ, 2011. С.79-83.
37. Вовченко А.Е. Анализ и сравнение систем интеграции неоднородных информационных ресурсов / Вовченко А.Е., Калининченко Л.А. // Электронные библиотеки: перспективные методы и технологии, электронные коллекции: Труды 10 Всероссийской научной конференции "RCDL-2008". - Дубна: ОИЯИ, 2008. - С. 246-251.
38. Волков А.А., Лосев Ю.Г., Лосев К.Ю. Информационная поддержка жизненного цикла объектов строительства // Информационные системы и логистика в строительстве. Вестник МГСУ. №11. –М.: НИУ МГСУ, 2012. С.253-258.
39. Волкова В. Н., Козлов В. Н. Моделирование систем и процессов: учебник для академического бакалавриата / В. Н. Волкова, Г. В. Горелова, В. Н. Козлов, Ю. И. Лыпарь, Н. Б. Паклин, А. Н. Фирсов, Л. В. Черненькая ; под общ. ред. В.

Н. Волковой и В. Н. Козлова. — М.: Издательство Юрайт, 2014. 588 с. — Серия: Бакалавр. Академический курс.

40. *Гаазе-Рапопорт М.Г., Поспелов Д.А.* Структура исследований в области искусственного интеллекта //Толковый словарь по искусственному интеллекту /Авторы – составители А.Н. Аверкин, М.Г. Гаазе-Рапопорт, Д.А. Поспелов. — М.: Радио и связь, 1992. — С. 5-20.

41. *Гаврилова Т. А., Кудрявцев Д. В., Муромцев Д. И.* Инженерия знаний. Модели и методы: Учебник. — СПб.: Издательство «Лань», 2016. — 324 с.: ил. — (Учебники для вузов. Специальная литература).

42. *Гаврилова Т.А., Хорошевский В.Ф.* Базы знаний интеллектуальных систем. — СПб.:Питер, 2000. — 384 с.

43. *Гаранина Н.О., Ануреев И.С., Боровикова О.И.* Онтология процессов, ориентированная на верификацию. Моделирование и анализ информационных систем. 25(6). 2018;:607-622.

44. *Головков В., Портнов А., Чернов В.* RDF — инструмент для неструктурированных данных // Открытые системы. СУБД. № 09. — М.: ООО «Издательство «Открытые системы», 2012. Интернет-ресурс. URL: www.osp.ru. (дата обращения: 02.10.2018)

45. *Горелик А.Л., Гуревич И.Б., Скрипкин В.А.* Современное состояние проблемы распознавания: Некоторые аспекты. М.:Радио и связь, 1985. — 160 с.

46. *Городецкий В.И., Грушинский М.С., Хабалов А.В.* Многоагентные системы (обзор). // Новости искусственного интеллекта. №2. 1998. С. 64 – 116.

47. *Городецкий В.И., Серебряков С.В.* Методы и алгоритмы коллективного распознавания: обзор // Труды СПИИРАН. Вып. 3. т. 1. — СПб.: Наука, 2006 г. С. 13 – 18.

48. *Горшков С.* Введение в онтологическое моделирование. ООО «Трини-Дата», 2016. Интернет-ресурс. URL: trinidata.ru. (дата обращения: 01.10.2018)

49. ГОСТ 34.321-96 Информационные технологии. Система стандартов по базам данных. Эталонная модель управления данными.

50. ГОСТ Р 50922-2006. Защита информации. Основные термины и определения.
51. ГОСТ Р 58125-2018 Системы космические. Система технологического обеспечения разработки и постановки на производство изделий космической техники. Организация и управление технологической подготовкой производства.
52. ГОСТ Р 53802-2010. Системы и комплексы космические. Термины и определения.
53. ГОСТ Р ИСО 10303-1-99 Системы автоматизации производства и их интеграция. Представление данных об изделии и обмен этими данными. Часть 11. Методы описания. Справочное руководство по языку Express.
54. ГОСТ Р ИСО 18629-1-2010 Системы промышленной автоматизации и интеграция. Язык спецификаций процесса.
55. ГОСТ Р ИСО/МЭК 12207-2010. Информационная технология (ИТ). Системная и прикладная инженерия. Процессы жизненного цикла программных средств.
56. ГОСТ Р ИСО/МЭК 7498-1-99 Информационная технология. Взаимосвязь открытых систем. Базовая эталонная модель. Часть 1. Базовая модель.
57. ГОСТ РВ 52006-2003. Создание изделий военной техники и материалов военного назначения. Термины и определения. 01.07.2003.
58. ГОСТ РО 1410-002-2010. Ракетно-космическая техника. Система информации о техническом состоянии и надежности космических комплексов и входящих в их состав изделий.
59. Графический язык моделирования бизнес-процессов BPMN Версия 2.0. Пер. с англ. комп. EleWise [Электронный ресурс]. URL: elma-bpm.ru (дата обращения 31.10.2017). 298 с.
60. Гуцин А.Н. Основы представления знаний: учеб, пособие / А.Н. Гуцин; Балл. гос. техн. ун-т. — СПб., 2007. -31 с.
61. *Давлетбакова З.Л.* Построение модели обработки пространственной информации на основе методов нечеткой логики // Современные проблемы

науки и образования: электрон. науч. журн. 2013. № 6 (50). URL: www.science-education.ru (дата обращения: 10.10.2018)

62. *Данеев А.В., Воробьев А.А., Лебедев Д.М.* Исследование динамики поведения сложных организационно-технических систем в условиях воздействия неблагоприятных факторов // Вестник Воронежского института МВД России. Вып. №2 – Воронеж: ГОУ ВПО Воронежский институт МВД РФ, 2010. С. 163-171.

63. *Дейт К. Дж.* Введение в системы баз данных, 8-е издание.: Пер. с англ. — М.: Издательский дом "Вильямс", 2005. — 1328 с.: ил.

64. *Дли М.И., Стоянова О.В., Белозерский А.Ю.* Модели представления данных сложных производственных проектов в автоматизированных информационных системах промышленных предприятий // Программные продукты и системы. №4 (112). 2015. С. 210-218.

65. *Демидов А. А., Захаров Ю. Н.* Информационно-аналитические системы поддержки принятия решений в органах государственной власти и местного самоуправления. Основы проектирования и внедрения: Учебное пособие. — СПб.: НИУ ИТМО, 2012. — 100 с.

66. *Дмитриев А.К., Юсупов Р.М.* Идентификация и техническая диагностика. — МО СССР, 1987. —521 с.

67. *Дмитриева Е.О.* Инструменты *bi* для поддержки принятия решений для руководителей высшего и среднего уровней с помощью решения SAS (опыт ВТБ24) // Вестник университета, №6. — М.: ГУУ, 2017. С. 10-14.

68. *Доросинский Л.Г., Зверева О.М.* Информационные технологии поддержки жизненного цикла изделия. — Ульяновск: Зебра, 2016. — 243 с.

69. *Евгенов Г. Б.* Интеллектуальные системы проектирования : учеб. пособие / Г. Б. Евгенов. - М. : Изд-во МГТУ им. Н. Э. Баумана, 2009. — 334 с.

70. *Егорова А.А., Козлов С.А.* Информационные системы: методы и средства проектирования // Научный вестник МГТУ ГА. Вып. 105 — Москва.: ФГОУВПО МГТУ ГА, 2006. С. 84 – 92.

71. *Журавлева Е.В.* Теория процессов для моделей мира СМА В.Н. Брюшинкина // РАЦИО.ru. № 12. 2014. С. 45-55.
72. *Золин Е.* Дескрипционная логика (лекции). Интернет-ресурс. URL: <http://pcs.math.msu.su/~zolin/dl/>. Дата обращения: 15.12.2018.
73. *Ильин В.Д.* Система порождения программ. –М.: Наука, 1989. 264 с.
74. *Инюшкина О.Г.* Проектирование информационных систем (на примере методов структурного системного анализа): учебное пособие / О.Г. Инюшкина, Екатеринбург: «Форт-Диалог Исеть», 2014. 240 с.
75. *Казаков И. А., Малых А. А., Манцивода А. В.* Погружение реляционных баз данных в объектные онтологии: реализационные аспекты. Изв. Иркутского гос. ун-та. Сер. Математика. Т. 10. – Иркутск: ФГБОУ ВО «ИГУ», 2014. С. 27 – 43.
76. *Калиниченко Л. А.* Поддержка баз данных с онтологическими зависимостями на основе дескриптивных логик // Системы высокой доступности. Т. 8. № 1. –М.: Радиотехника, 2012. С. 3-12.
77. *Калянов Н.К.* Теория бизнес-процессов: формальные модели и методы // Экономика, статистика и информатика. Вестник УМО. №4. – М.: Московский ГУ экономики, статистики и информатики, 2016. С. 19 – 21.
78. *Капитонов В.А., Кононенко А.В., Тихомиров С.А.* Среда автоматизированного анализа измерительной информации космических ракетноносителей // Вестник Самарского государственного аэрокосмического университета С.П. Королева. №4 (42). – Самара: Самарский университет, 2013. С.154-163.
79. *Карабегов А.В., Тер-Микаэлян Т.М.* Введение в язык SDL. –М: Радио и связь, 1993. – 184 с.
80. *Каргин В.А., Майданович О.В., Охтилев М.Ю.* Автоматизированная система информационной поддержки принятия решений по контролю в реальном времени состояния ракетно-космической техники // Известия высших учебных заведений. Приборостроение. Т. 53. № 11. –СПб: СПб НИУ ИТМО, 2010. С. 20-23.

81. Катюха Р.В., Логунов С.В., Рогов Д.А. Методика многопараметрической идентификации состояния элементов космических аппаратов, реализуемая на основе применения аппарата нечеткой логики // Труды Военно-космической академии им. А.Ф. Можайского. 2014. № 642. С. 132-135.
82. Клещев А.С., Артемьева И.Л. Математические модели онтологий предметных областей. Часть 2. Компоненты модели // Научно-техническая информация: Серия 2. № 3. 2001. С. 19-29.
83. Кнут Д. Искусство программирования, том 1. Основные алгоритмы. — 3-е изд. — М.: «Вильямс», 2006. — 720 с.
84. Когаловский М. Р. и др. Глоссарий по информационному обществу / Под общ. ред. Ю.Е. Хохлова. — М.: Институт развития информационного общества, 2009. 160 с.
85. Когаловский М.Р. Метаданные в компьютерных системах // Программирование. Т. 39, №4. — МАИК/Наука «Интерпериодика», 2013. С. 28-46.
86. Когаловский М.Р. Методы интеграции данных в информационных системах. — М.: ФГБУН Ист-т проблем рынка РАН, 2010. 9 с.
87. Когаловский М.Р. Перспективные технологии информационных систем / Когаловский М.Р., - 2-е изд., (эл.) - М.: ДМК Пресс, 2018. - 287 с..
88. Когаловский М.Р. Системы доступа к данным, основанные на онтологиях // Программирование. №4. — М.: РАН, 2012. С. 55-77.
89. Когаловский М.Р., Калиниченко Л.А. Концептуальное и онтологическое моделирование в информационных системах // Программирование. №5. — М.: РАН, 2009. С.3-25.
90. Концепция формирования единого информационного пространства России и соответствующих государственных информационных ресурсов. Одобрена решением Президента РФ от 23.11.1995. № Пр -1694.
91. Корнаков А.Н. Модель сложной организационно-технической системы // Перспективы науки и образования. №2. (14). 2015. С. 44 – 50. Интернет-ресурс. URL: pnojurnal.wordpress.com.

92. *Королев Е. Н., Бескоровайная М. А., Фиртыч О. А.* Особенности проектирования реляционных баз данных с помощью CASE-средств в нотации UML // *Объектные системы, №2(2).* 2010, – Шахты: Олейник Павел Петрович. С. 86-89.
93. *Корчагин А.Б.* и др. Модели данных информационного взаимодействия / *Корчагин А.Б., Лисьих И.Г., Никифоров Д.А., Р.Л. Сиваков.* // *International Journal of Open Information Technologies.* vol. 5. no.3, 2017.. С. 49 – 54.
94. *Котов В.Е., Сабельфельд В.К.* Теория схем программ. –М.: Наука, Гл. ред. физ.-мат. лит., 1991. – 248 с.
95. *Кочегаров И.И.* Система управления жизненным циклом изделий на основе универсальной технологической платформы // *Труды международного симпозиума «Надежность и качество», Т.2.* – Пенза: Пензенский ГУ, 2010. С. 424-426.
96. *Крылов А.В., Охтилев П.А., Бахмут А.Д.* Комбинированная прецедентная модель поддержки принятия решения и её применение в экспертных системах для сложных организационно-технических объектов // *Проблемы управления и моделирования в сложных системах: Труды XIX Международной конференции / Под ред.:акад. Е.А.Федосова, акад. Н.А.Кузнецова, проф.В.А.Виттиха.* – Самара: ООО «Офорт», 2017. С.428-435. ISBN 978-5-473-01154-8
97. *Крылов А.В., Охтилев П.А., Бахмут А.Д.* Использование прецедентной методологии при построении экспертных систем управления сложными организационно-техническими объектами // *Научная сессия ГУАП: сб. докл.: В 3 ч. Ч. II. Технические науки / СПб.: ГУАП. СПб., 2017. С.255-261. ISBN 978-5-8088-1218-5 (Ч. II)*
98. *Крючков Л.Н.* Структура стандартов STEP. Обзор языка Express. – СПб: Балт. техн. ун-т, 2009. 11 с.
99. *Кудрявцев Д. В.* Практические методы отображения и объединения онтологий // *Труды 11-й Национальной конференции по искусственному интеллекту с международным участием (КИИ-08), семинар «Знания и Онтологии*

ELSEWHERE», 29 сентября-3 октября 2008. г. Дубна, Россия. Том 3. – М.: URSS, 2008. – С. 164-173.

100. *Лавров А.А., Липатов А.А., Когель Д.М.* Технология активных объектов – новая реализация // 11-я Национальная конференция по искусственному интеллекту с международным участием – КИИ-2008.

101. *Ланин В.* Мультиагентная система для интеллектуального анализа документов // XIV International Conference “Knowledge-Dialogue-Solution” KDS-2008. С.166-172.

102. *Ларин М.В.* Электронные документы: вопросы теории и практики // Вестник РГГУ. Серия: документоведение и архивоведение. Информатика. Защита информации и информационная безопасность. – М.: Изд. РГГУ, 2015. №2(145). С. 53-63.

103. *Ларичев О.И.* Теория и методы принятий решений, а также Хроника событий в Волшебных Странах: Учебник. – М.: Логос, 2000. – 296 с.

104. *Левашова Т.В.* Принципы управления онтологиями, используемые в среде интеграции знаний// Труды СПИИРАН/ Под ред. Р.М. Юсупова. – СПб.: СПИИРАН.2002. Вып.1, т.2. С. 51–68.

105. *Логинов Н. В.* “Методы стохастической аппроксимации” // Автоматика и телемеханика, 1966, № 4. С. 185 – 204.

106. *Люггер Джордж Ф.* Искусственный интеллект: стратегии методы решения сложных проблем, 4-е издание: Пер. с англ. – М.:Издательский дом «Вильямс», 2003. -864с. :ил. – Парал. тит. англ. ISBN 5-8459-0437-4 (рус.)

107. *Ляпунов А.А.* Проблемы кибернетики. –М.: –М.: Наука, Гл. ред. физ.-мат. лит., 1958. – 268 с.

108. *Мансуров Н.Н., Майлингова О.Л.* Методы формальной спецификации программ: языки MSC и SDL. –М.: АО “Диалог-МГУ ”, 1998. – 125 с.

109. *Марков Н.Г.* Инструментальные средства для создания единого информационного пространства промышленных компаний // Информационное общество, вып.3, 2014. с.53-62

110. *Масленникова Н.Ю., Слинкова О.К.* Понятие и сущность мониторинга с позиции системного подхода // *Science Time*. 2014 г. Вып.№6(6). С. 110-121.
111. *Маслобоев А.В., Шишаев М.Г.* Мультиагентная система интеграции распределенных информационных ресурсов инноваций // Программные продукты и системы. №4. – Тверь: ЗАО НИИ "Центрпрограммсистем"2007. С. 30-32.
112. *Микони С.В., Соколов Б.В., Юсупов Р.М.* Применение алгебраического подхода в квалиметрии моделей и полимодельных комплексов //Сборник докладов VI научно-практической конференции «Имитационное моделирование. Теория и практика». Том 1. –Казань: Изд-во «Фэн», 2013, с.68-79.
113. *Миронов А.М.* Теория процессов. – Переславль-Залесский: Университет города Переславля, 2008. — 345 с.
114. *Митрофанова О.А.* Онтологии как системы хранения знаний / О.А. Митрофанова, Н.С. Константинова // Всероссийский конкурсный отбор обзорно-аналитических статей по приоритетному направлению «Информационно-телекоммуникационные системы», 2008. – 54 с.
115. *Могилев А. В.* Технологии поиска и хранения информации. Технологии автоматизации управления / А. В. Могилев, Л. В. Листрова. – СПб.: БХВ-Петербург, 2012. – 320 с.
116. *Муромцев Д. И.* и др. Опыт использования онтологий верхнего уровня при проектировании базы знаний музея оптических технологий / Муромцев Д. И., Баландин Е. А., Катков Ю. В., Починок И. Н. // Материалы Всерос. конф. „Знания— Онтологии — Теории" (30НТ-09). Новосибирск, 2009. Т. 1. С. 165—172.
117. *Мюллер Р.Д.* Базы данных и UML: Проектирование. М.: Лори, 2002. 420с.
118. *Найханова Л.В.* Основные аспекты построения онтологий верхнего уровня и предметной области. // В сборнике научных статей "Интернет-порталы: содержание и технологии". Выпуск 3. / Редкол.: А.Н. Тихонов (пред.) и др.; ФГУ ГНИИ ИТТ "Информика". - М.: Просвещение, 2005. - С. 452-479.

119. *Найханова Л.В.* Основные типы семантических отношений между терминами предметной области // Известия высших учебных заведений. Поволжский регион. Технические науки: науч.-практ. журн. – Пенза: Информационно-издательский центр ПензГУ. – 2008. – № 1. – С. 62-71
120. *Нариньяни А.С.* Модель или алгоритм: новая парадигма информационной технологии // Информационные технологии, 1997. – С. 17-22.
121. *Науменко А.П.* Теория и методы мониторинга и диагностики: Материалы лекций. – Омск: ОмГТУ, 2017. – 154 с.
122. *Нейбург Э.Д., Максимчук Р.А.* Проектирование баз данных с помощью UML. – М: Вильямс, 2002. 288 с.
123. *Немыкин С.А., Охтилев М.Ю., Охтилев П.А., Крылов А.В., Соколов Б.В.* Анализ и обоснование подходов к созданию систем поддержки принятия решений в асу объектами военно-государственного управления // Материалы 9-й конференции «Информационные технологии в управлении» (ИТУ-2016). – СПб: ЗАО «Концерн ЦНИИ «Электприбор», 2016. С. 185-193.
124. *Неуймин Я.Г.* Модели в науке и технике. История, теория, практика. – Л.:Наука, 1984. – 190 с.
125. *Нечитайлов Ю.В.* Дескрипционные логики как фундамент для разрешения ряда алгоритмических вопросов // Логико-философские штудии. Том 15 (№ 4). – СПб: СПбФО, 2017. С. 204 –226.
126. *Новиков Б.А., Графеева Н.Г., Михайлова Е.Г.* BIG DATA: новые задачи и современные подходы // Компьютерные инструменты в образовании. №4. – СПб: ФГАОУ «ЛЭТИ», 2014. С.10-18.
127. *Норенков И.П.* Основы автоматизированного проектирования: учеб. для вузов. -4-е изд., перераб. и доп. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2009. 430 с.
128. *Орлик С.* Программная инженерия. Программные требования. Перевод Swebok 2004. 2010 г. Интернет-ресурс. URL: <http://swebok.sorlik.ru>. Дата обращения: 10.01.2019.

129. Орлов С. А., Цилькер Б. Я. Технологии разработки программного обеспечения: Учебник для вузов. 4.е изд. Стандарт третьего поколения. – СПб.: Питер, 2012. 608 с.
130. Охтилев М.Ю. Использование топологического пространства вычислений при оценивании технических состояний объектов управления // Изв. вузов. Приборостроение. 2000. – Т.43, № 7. – С.27 – 30.
131. Охтилев М.Ю. Один из подходов к проблеме автоматизации анализа измерительной информации в реальном времени // Программирование. 2001. № 6. С.329–335.
132. Охтилев М.Ю. Основы теории автоматизированного анализа измерительной информации в реальном времени. Синтез системы анализа. – СПб.: ВИКУ им. А.Ф.Можайского, 1999. 160 с.
133. Охтилев М.Ю. Системы искусственного интеллекта и их применение в автоматизированных системах мониторинга состояния сложных организационно-технических объектов. – СПб: ГУАП, 2018. – 261 с.
134. Охтилев М. Ю., Мустафин Н. Г., Миллер В. Е., Соколов Б. В.. Теоретические основы проактивного управления сложными объектами // Приборостроение. – СПб.: НИУ ИТМО, 2014. Т. 57, № 11. С.7-15.
135. Охтилев М.Ю., Соколов Б.В. Новые информационные технологии мониторинга и управления состояниями сложных технических объектов в реальном масштабе времени // Тр. СПИИРАН, 2:2. – СПб: ФГБУН СПИИРАН, 2005. С. 249–265.
136. Охтилев М.Ю., Соколов Б.В. Теоретические и прикладные проблемы разработки и применения автоматизированных систем мониторинга состояния сложных технических объектов //Тр. СПИИРАН. Отделение информационных технологий и вычислительных систем. Спб., 2002. Вып. 1. Т. 1. С.167 - 180.
137. Охтилев М.Ю., Соколов Б.В., Юсупов Р.М. Интеллектуальная информационная технология комплексной автоматизации процессов мониторинга и

управления подвижными объектами // Гироскопия и навигация. № 2 (53). – СПб: ЦНИИ «Электроприбор», 2006. С. 120.

138. *Охтилев М.Ю., Соколов Б.В., Юсупов Р.М.* Интеллектуальные технологии мониторинга состояния и управления структурной динамикой сложных технических объектов. – М.: Наука, 2006. – 410 с. – (Информатика: неограниченные возможности и возможные ограничения).

139. *Охтилев М.Ю., Чуприков А., Ничипорович О.П.* и др. Унифицированная информационная технология мониторинга динамически изменяющихся состояний космических средств и средств нку на основе измерительной информации и представления его результатов на индивидуальных и коллективных средствах отображения / Охтилев М.Ю., Чуприков А., Ничипорович О.П., Черников А.Д., Заозерский С.А., Николаев Д.А., Луговая Т.С., Воеводина В.Н., Фирсанов А.В., Харлан А.А. // Авиакосмическое приборостроение. № 5. – М: ООО «Научтехлитиздат», 2007.С. 20-24.

140. *Охтилев П. А.* Интеллектуальный комплекс автоматизированного проектирования систем информационно-аналитической поддержки жизненного цикла сложных объектов // Изв. вузов. Приборостроение. 2018. Т. 61, № 11. С. 963—971.

141. *Охтилев П.А.* О методологии автоматизированного проектирования интеллектуальных информационно-аналитических систем для сложных объектов // Региональная информатика (РИ-2018). XVI Санкт-Петербургская международная конференция «Региональная информатика (РИ-2018)». Санкт-Петербург, 24-26 октября 2018 г. Материалы конференции. \ СПОИСУ. – СПб, 2018. С. 585-587.

142. *Охтилев П.А.* Программный комплекс первичной обработки информации при оценивании состояний сложных технических объектов в составе автоматизированной системы поддержки принятия решений. // 68 международная научная конференция ГУАП. Сб. докл.: в 2ч. Ч.1 Технические науки / СПб.:ГУАП., 2015. С.255 - 258: ил. ISBN 978-5-8088-1032-7 (Ч.1)

143. *Охтилев П.А.* Обзор возможных подходов к оцениванию состояний сложных организационно-технических объектов//Научная сессия ГУАП: сб. докл.: В 3 ч. Ч. 2 Технические науки / СПб.: ГУАП, СПб., 2016. С.246-255.
144. *Охтилев П.А., Автамонов П.Н., Бахмут А.Д., Крылов А.В., Охтилев М.Ю., Соколов Б.В.* Применение технологии поддержки принятия решений на различных этапах жизненного цикла космических средств в составе системы информации о техническом состоянии и надёжности // Вестник Самарского университета. Аэро-космическая техника, технологии и машиностроение. 2017. Т. 16, № 3. С. 173-184.
145. *Охтилев П.А., Бахмут А.Д., Коромысличенко В.Н., Крылов А.В.* и др. Концептуальное моделирование основных компонентов интеллектуальных систем поддержки принятия решений при управлении сложными объектами на всех этапах их жизненного цикла / Бахмут А.Д., Коромысличенко В.Н., Крылов А.В., Охтилев М.Ю., Охтилев П.А., Соколов Б.В., Устинов А.В., Зянчурин А.Э. // Труды XX Международной конференции «Проблемы управления и моделирования в сложных системах». Под ред. Е.А. Федосова, Н.А. Кузнецова, С.Ю. Боровика. – Самара: ООО «Офорт», 2018. С. 299-306.
146. *Охтилев П.А., Бахмут А.Д., Коромысличенко В.Н., Крылов А.В.* и др. Обзор состояния исследований задач автоматизированного проектирования интеллектуальных систем поддержки принятия решений при управлении сложными объектами на всех этапах их жизненного цикла / Бахмут А.Д., Коромысличенко В.Н., Крылов А.В., Охтилев М.Ю., Охтилев П.А., Соколов Б.В., Устинов А.В., Зянчурин А.Э. // Труды XX Международной конференции «Проблемы управления и моделирования в сложных системах». Под ред. Е.А. Федосова, Н.А. Кузнецова, С.Ю. Боровика. – Самара: ООО «Офорт», 2018. С. 307-313.
147. *Охтилев П.А., Бахмут А.Д., Крылов А.В.* Обзор и применение моделей представления знаний в интеллектуальной системе мониторинга структурных состояний сложных организационно-технических объектов // Научная сессия

ГУАП: сб. докл.: В 3 ч. Ч. II. Технические науки / СПб.: ГУАП. СПб., 2017. С.266-277. ISBN 978-5-8088-1218-5 (Ч. II)

148. *Охтилев П.А., Бахмут А.Д., Крылов А.В.* Подход к оцениванию состояния сложного организационно-технического объекта на основе обобщенных вычислительных моделей // Десятая Всероссийская мультikonференция по проблемам управления : материалы 10-й Всероссийской мультikonференции (с. Дивноморское, Геленджик, Россия, 11–16 сентября 2017 г.) : в 3 т. / [редкол.: И. А. Каляев (отв. ред.) и др.]. – Ростов-на-Дону ; Таганрог : Издательство Южного федерального университета, 2017. Т. 1. С.201 – 204.

149. *Охтилев. П.А., Бахмут А.Д., Крылов А.В.* Полимодельное описание интеллектуальной системы мониторинга технического состояния ракеты-носителя «Союз-2» // Труды Военно-космической академии имени А.Ф. Можайского. Вып. 664. – СПб: ФГБВОУ ВО ВКА им. А.Ф. Можайского МО РФ, 2018. С. 10-19.

150. *Охтилев П.А., Бахмут А.Д., Крылов А.В., Охтилев М.Ю.* Применение технологии имитационно-аналитического моделирования к оцениванию структурных состояний сложных организационно-технических объектов на основе обобщенных вычислительных моделей // Труды Восьмой всероссийской научно-практической конференции по имитационному моделированию и его применению в науке и промышленности «Имитационное моделирование. Теория и практика» (ИММОД-2017) [Электронный ресурс] URL: simulation.su (дата обращения 27.10.2017), Санкт-Петербург, 2017 г. С. 145-150.

151. *Охтилев П.А., Бахмут А.Д., Крылов А.В., Охтилев М.Ю., Соколов Б.В.* Подход к оцениванию структурных состояний сложных организационно-технических объектов на основе обобщенных вычислительных моделей // Научно-емкие технологии в космических исследованиях Земли. 2017. Т. 9. № 5. С. 73–82.

152. *Охтилев П.А., Бахмут А.Д., Крылов А.В., Охтилев М.Ю., Соколов Б.В.* Применение технологии поддержки принятия решений при управлении жизненным циклом космических средств // Труды II Международной научной

конференции по проблемам управления в технических системах (CTS'2017), СПб.: СПбГЭТУ «ЛЭТИ», 2017 г. С. 43-46.

153. *Охтилев П.А., Крылов А.В., Бахмут А.Д.* Системный анализ моделей представления знаний при решении задачи синтеза системы мониторинга структурных состояний сложных организационно-технических объектов // Проблемы управления и моделирования в сложных системах: Труды XIX Международной конференции / Под ред.: акад. Е.А. Федосова, акад. Н.А. Кузнецова, проф. В.А. Виттиха. – Самара: ООО «Офорт», 2017. С.455-467.

154. *Охтилев П.А., Бахмут А.Д., Крылов А.В., Палицын А.Б., Автамонов П.Н.* «Редактор распределения ресурсов сети сложных объектов в составе программного комплекса поддержки принятия решений в логистических задачах управления сложными организационно-техническими объектами» // Свидетельство о государственной регистрации программы для ЭВМ № 2019613148 от 12.03.2019 г.

155. *Охтилев П.А., Бахмут А.Д., Крылов А.В., Палицын А.Б., Автамонов П.Н.* «Редактор распределения доступных средств в составе программного комплекса поддержки принятия решений в логистических задачах управления сложными организационно-техническими объектами» // Свидетельство о государственной регистрации программы для ЭВМ № 2019613397 от 15.03.2019 г.

156. *Охтилев П.А., Бахмут А.Д., Туманова А.В., Охтилев М.Ю., Ключарев А.А.* «Автоматизированная система планирования учебного процесса вуза» // Свидетельство о государственной регистрации программы для ЭВМ № 2018614290 от 04.04.2018 г.

157. *Охтилев П.А., Бахмут А.Д., Туманова А.В., Охтилев М.Ю., Ключарев А.А.* «Автоматизированная система отчетности учебного процесса вуза» // Свидетельство о государственной регистрации программы для ЭВМ № 2018614289 от 04.04.2018 г.

158. *Паращенко Д.А., Царев Ф.Н., Шалыто А.А.* Технология моделирования одного класса мультиагентных систем на основе автоматного программирования на примере игры «Соревнование летающих тарелок»
159. *Паронджанов В. Д.* Дружелюбные алгоритмы, понятные каждому. Как улучшить работу ума без лишних хлопот. — М.: ДМК-пресс, 2010. — 464 с.
160. *Перегудов Ф. И., Тарасенко Ф. П.* Основы системного анализа: Учеб 2-е изд., доп. – Томск: Издательство НТЛ, 1997. – 396 с.: ил.
161. *Погорелов В.И.* Система и её жизненный цикл: введение в CALS-технологии: учебное пособие / В.И. Погорелов. – СПб: Балт. Гос. Техн. Ун-т., 2010. 182 с.
162. *Пономарева О.А.* Построение онтологии эталонной модели бизнес-процессов // Технологии разработки информационных систем - ТРИС-2016: материалы VII Международной научно-технической конференции. Том 2. Под науч. редакцией: Ю. И. Рогозов Т. 2. – Таганрог: Издательство ЮФУ, 2016. С. 70-74.
163. *Попова О.Р., Увайсов С.У.* CALS Современные технологии управления жизненным циклом продукта // Труды международного симпозиума «Надежность и качество». – Пенза: Пензенский ГУ, 2012. – 462 с.
164. *Поспелов Д.А.* Искусственный интеллект. – В 3-х кн. Кн. 2. Модели и методы: Справочник. – М.: Радио и связь. 1990г. – 304 с.
165. *Потрясаев С.А.* Комплексное моделирование сложных процессов на основе нотации BPMN // Известия высших учебных заведений. Приборостроение. Т. 59. № 11. – СПб: СПб НИУ ИТМО, 2016. С. 913-920.
166. Программа "Цифровая экономика Российской Федерации. Утверждена распоряжением Правительства Российской Федерации от 28 июля 2017 г. № 1632-р.
167. *Пустовалова Н. В., Авдеенко Т. В.* Построение согласованной модели требований для процесса программной инженерии // Труды СПИИРАН. Вып. 44. – СПб: СПИИРАН, 2016. С. 31–49.

168. *Пьявченко Т.А.* Управление технологическими процессами на основе SCADA // Политематический сетевой электронный научный журнал Кубанского государственного аграрного университета, №128(04). – Краснодар: ФГБОУ ВО Кубанский ГАУ, 2017. С.2-22.
169. *Рассел С., Норвиг П.* Искусственный интеллект: современный подход, 2-е изд.: Пер. с англ. — М. : Издательский дом "Вильямс", 2006. — 1408 с.: ил. — Парал. тит. англ.
170. РД IDEF 0 – 2000. Методология функционального моделирования IDEF0. Руководящий документ. – М.: Госстандарт России. 2000 г. 75 с.
171. Россия 4.0: четвертая промышленная революция как стимул глобальной конкурентоспособности [Электронный ресурс] // ИТАР-ТАСС. URL: <http://tass.ru> (дата обращения: 16.02.2018).
172. *Рыбалов Ю.В.* Автоматизированная информационно-аналитическая система по искусственным сооружениям на автомобильных дорогах // САПР и ГИС автомобильных дорог. № 2(5). –Томск: ООО «ИндорСофт», 2015. С. 126 – 135.
173. *Рыбина Г.В., Паронджанов С.С.* Моделирование процессов взаимодействия интеллектуальных агентов в многоагентных системах // Искусственный интеллект и принятие решений. №3. – М.: ФГУ «ФИЦ «Информатика и управление», 2008. С.3-15.
174. *Серебряков В.А.* Семантическая интеграция данных, 2012. Интернет-ресурс. URL: <http://sp.ctmc.msu.ru>. (дата обращения: 22.03.2018).
175. *Симанков В.С., Луценко Е.В.* Адаптивное управление сложными системами на основе теории распознавания: Монография. Краснодар: ТУ КубГТУ, 1999. - 318с.
176. *Скобелев П.О.* Онтология деятельности для ситуационного управления предприятиями в реальном времени // Онтология проектирования. № 1 (3). – Самара: ФГАОУ Самарский университет им. С.П. Королева, 2012. С. 6-38.
177. *Смирнов А.В., Кашевник А.М., Пономарев А.В., Савосин С.В.* Онтологический подход к организации взаимодействия сервисов интеллектуального

пространства при управлении гибридными системами // Искусственный интеллект и принятие решений. № 4. 2014. с. 42-51.

178. *Соколов Б.В.* и др. Интеллектуальные информационные технологии управления жизненным циклом сложных технических объектов / Б.В. Соколов, А.И. Птушкин, М.Ю. Охтилев, О.В. Майданович // Региональная информатика (РИ-2010): сб. тр. XII С.-Петерб. Международной конф. «Регионал», 2010.

179. *Соколов Б.В.* и др. Комплексное моделирование сложных объектов: основные особенности и примеры практической реализации / Охтилев М.Ю., Павлов А.Н., Плотников А.М., Потрясаев С.А., Соколов Б.В., Юсупов Р.М. // Седьмая всероссийская научно-практическая конференция “Имитационное моделирование. Теория и практика” (ИММОД-2015): Труды конф., 21-23 окт. 2015 г., Москва: в 2 т./ Ин-т проблем упр. им.В.А.Трапезникова Рос. Акад. наук; под общ. ред. С.Н.Васильева, Р.М.Юсупова.-Т.1. Пленарные доклады – М.: ИПУ РАН, 2015. С.58-81.

180. *Соколов Б.В., Крылов А.В., Охтилев М.Ю., Охтилев П.А., Потрясаев С.А.* Логико-динамическая модель и алгоритмы комплексного планирования функционирования автоматизированной системы управления активными подвижными объектами // Проблемы управления и моделирования в сложных системах: Труды XIX Международной конференции / Под ред.: акад. Е.А.Федосова, акад. Н.А.Кузнецова, проф. В.А.Виттиха. – Самара: ООО «Офорт», 2017. С.508-513.

181. *Соколов Б.В., Охтилев М.Ю.* Методологические и технологические основы создания и использования нового поколения систем управления жизненным циклом космических средств // Труды XVI-ой международной молодежной конференции «Системы проектирования, технологической подготовки производства и управления этапами жизненного цикла промышленного продукта (CAD/CAM/PDM - 2016)». – СПб: ООО «Аналитик», 2016. С. 383-387.

182. *Соколов Б.В., Юсупов Р.М.* Комплексное моделирование рисков при выработке управленческих решений в сложных организационно-технических системах // Проблемы управления и информатики – 2006, №1. С. 1 – 22.
183. *Соколов Б.В., Юсупов Р.М.* Комплексное моделирование функционирования автоматизированной системы управления навигационными космическими аппаратами // Институт кибернетики им. В.М. Глушкова НАН Украины. - 2002. № 5. С. 103-117.
184. *Соколов Б. В., Юсупов Р. М.* Концептуальные и методические основы квалиметрии моделей и полимодельных комплексов // Труды СПИИРАН. Вып. 2, т. 1. — СПб.: СПИИРАН, 2004. С. 10 – 35.
185. *Соловьев И.В.* Общие принципы управления сложной организационно-технической системой // Перспективы науки и образования. 2014 г. №2 (8). Электронный журнал. URL: pnojurnal.wordpress.com. С. 21 – 27.
186. *Стоянова О.В., Дли М.И., Белозерский А.Ю.* Модели представления данных сложных производственных проектов в автоматизированных информационных системах промышленных предприятий // Программные продукты и системы, №4. – Тверь: ЗАО НИИ "Центрпрограммсистем", 2015. С. 210-218.
187. *Ступников С.А.* и др. Методы унификации нетрадиционных моделей данных / Скворцов Н.А., Будзко В.И., Захаров В.Н., Калиниченко Л.А. // Системы высокой доступности. Том 10. № 1. –М.: Радиотехника, 2014. С. 18-39.
188. *Судов Е.В.* и др. Концепция развития CALS-технологий в промышленности России / Судов Е.В., Левин А.И., Давыдов А.Н., Барабанов В.В. - М.: НИЦ CALS-технологий "Прикладная логистика", 2002. - 127 с.
189. *Ткаченко Н.И., Спириин Н.А.* Применение сервис-ориентированной архитектуры при интеграции систем управления технологическими процессами // Известия Томского политехнического университета, Т.317, №5. –Томск: ГОУ ВПО ТПУ, 2010. С. 61-67.
190. Толковый словарь по искусственному интеллекту /Авторы – составители А.Н. Аверкин, М.Г. Гаазе-Рапопорт, Д.А. Поспелов. – М.: Радио и связь, 1992. – 256 с.

191. *Троцкий Д.В., Городецкий В.И.* Сценарная модель знаний и язык описания процессов для оценки и прогнозирования ситуаций // Труды СПИИРАН. Вып. 8. – СПб: ФГБУН СПИИРАН, 2009. С. 94 – 127.
192. *Ту Дж., Гонсалес Р.* Принципы распознавания образов. Пер. с англ. И.Б. Гуревича. Под ред. Ю.И. Журавлева. М.: Мир, 1978. -411 стр.
193. *Тыгу Э.Х.* Концептуальное программирование. – М.: Наука. Главная редакция физико-математической литературы, 1984. 256 с. (Проблемы искусственного интеллекта)
194. *Тюгашев А.А.* Об автоматизированной спецификации, верификации и синтезе управляющих программ реального времени на базе логического и алгебраического подходов // Труды Международного симпозиума «Надежность и качество». Т.1. – Пенза: Пензенский ГУ, 2006. С.143-146.
195. *Урманов О.* ИПИ: шанс не отстать навсегда. Российский космос. №5 (29). 2008.
196. *Устимов К.О., Федоров Н.В.* Автоматизация построения имитационной модели бизнес-процессов на основе методологии IDEF0 и раскрашенных сетей Петри // Горный информационно-аналитический бюллетень (научно-технический журнал). – М.: ООО «Горная книга», 2013. С. 90 – 94.
197. *Ушаков Д.М., Телерман В.В.* Системы программирования в ограничениях (обзор) // Системная информатика: Сб. науч. тр. Новосибирск: Наука, 2000. Вып.7: Проблемы теории и методологии создания параллельных и распределенных систем. С. 275—310.
198. *Федоров И.Г.* Адаптация онтологии Бунге–Ванда–Вебера к описанию исполняемых моделей бизнес-процессов // Прикладная информатика, Том 10. №4 (58). – М.: Издательский дом «Синергия», 2015. С. 82-92.
199. *Федоров И.Г.* Анализ концептуальной модели бизнес-процесса с использованием онтологии Бунге-Ванда-Вебера // Экономика, Статистика и Информатика №6, 2014. С. 216-221.
200. *Фёдоров И.Г.* Моделирование бизнес-процессов в нотации BPMN 2.0: Монография. – М.: МЭСИ, 2013. – 255 стр.

201. *Фу К.* Структурные методы в распознавании образов./под ред. Айзермана М.:Мир, 1977. -320 стр.
202. *Халл Э., Джексон К., Дик Дж.* Инженерия требований. – М.: ДМК Пресс, 2017. – 224 с.
203. *Хаф Л.* Методологии разработки программного обеспечения. Часть 3. Rational Unified Process // КомпьютерПресс. № 1.Электронный журнал, 2004. URL: compress.ru (дата обращения: 15.12.2018)
204. *Хомоненко А.Д., Цыганков В.М., Мальцев М.Г.* Базы данных. Учебник для высших учебных заведений / Под ред. А. Д. Хомоненко. — 6-е изд., доп. – СПб.: КОРОНА-Век, 2009. 736 с.
205. *Цуканова О. А.* Методология и инструментарий моделирования бизнес-процессов: учебное пособие–СПб.: НИУ ИТМО, 2015. –100 с.
206. *Цуканова О. А.* Онтологическая модель представления и организации знаний. Учебное пособие для вузов. – М.: Горячая линия – Телеком, 2015. – 272 с.: ил.
207. *Черняк Л.* Интеграция данных: синтаксис и семантика // Открытые системы. СУБД. Электронный журнал. № 10. URL: www.osp.ru. (дата обращения: 26.12.18)
208. *Чуваков А.В.* Разработка информационной системы поддержки принятия решений при управлении сложными техническими системами // Международный научный журнал «Символ науки» – 2015, №6. С. 65 - 72
209. *Шалыто А.А.* SWITCH-технология. Алгоритмизация и программирование задач логического управления. – СПб: Наука, 1998. Интернет-ресурс. URL: <http://is.ifmo.ru> (дата обращения: 22.03.2018)
210. *Шибанов С.В., Горин А.А.* Программные интерфейсы баз данных: от пассивных запросов к активному взаимодействию // Надежность и качество: тр. Междунар. симп.: в 2 т. Т. 1. – Пенза: Изд-во ПГУ, 2011. С. 266-267.
211. *Шильников П.С.* Онтологии управления качеством компьютерных данных об изделии // Онтология проектирования. №2 (24). Т.7. – Самара: Самарский университет, 2017. С. 216 – 226.

212. *Шильников П.С.* Представление данных корпоративных информационных систем. – М.: МГТУ им. Н. Э. Баумана, 2010. – 150 с.
213. *Шмелев В.В.* Решение задачи оптимального планирования технологического процесса методом динамического программирования // Труды Военно-космической академии им. А.Ф. Можайского. № 645. – СПб: ВКА им. А.Ф. Можайского, 2014. С. 63-67.
214. *Шундеев А.С., Першин И.С.* Интеграция данных: от баз данных к информационным ресурсам // Материалы всерос. конф. с междунар. участием "Знания – онтологии – теории "(ЗОНТ – 09), 20–22 окт. 2009. – Новосибирск: Институт математики им. С. Л. Соболева СОРАН, 2009. С. 41–45.
215. *Юсупов Р. М., Соколов Б. В., Птушкин А.И., Иконникова А. В., Потрясаев С. А., Цивирко Е. Г.* Анализ состояния исследований проблем управления жизненным циклом искусственно созданных объектов // Тр. СПИИРАН, 16. – СПб: ФГБУН СПИИРАН, 2011. С. 37–109.
216. *Юсупов Р. М., Мусаев А. А.* Особенности оценивания эффективности информационных систем и технологий // Тр. СПИИРАН, 51. – СПб: ФГБУН СПИИРАН, 2017. С. 5–34.
217. *Яблочников Е.И., Фомина Ю.Н., Саломатина. А.А.* Компьютерные технологии в жизненном цикле изделия / Учебное пособие – СПб: СПбГУ ИТМО, 2010. – 180 с.
218. *Ambler S.W.* Agile Database Techniques: Effective Strategies for the Agile Software Developer/ John Wiley & Sons publ/, 2003. 416 pp.
219. *Auer S.* Mapping XML to OWL ontologies. Marktplatz Internet: von E-Learning bis E-Payment, 13. Leipziger Informatik-Tage, LIT 2005. P. 21 – 23.
220. *Bakhmut Aleksey D., Kljucharjov Alexander A., Krylov Aleksey V., Okhtilev Michael Yu., Okhtilev Pavel A., Ustinov Anton V., and Zyanchurin Alexander E.* Models, Algorithms and Monitoring System of the Technical Condition of the Launch Vehicle “ Soyuz-2 ” at All Stages of Its Life Cycle // Springer International Publishing AG, part of Springer Nature 2019 R. Silhavy (Ed.): CSOC 2018, AISC 764, 2019. pp. 288 –297.

221. *Bakhmut Aleksey D., Koromyslichenko Vladislav N., Krylov Aleksey V., etc.* (2019) *Methods of Conceptual Modeling of Intelligent Decision Support Systems for Managing Complex Objects at All Stages of Its Life Cycle / Aleksey D. Bakhmut, Vladislav N. Koromyslichenko, Aleksey V. Krylov, Michael Yu. Okhtilev, Pavel A. Okhtilev, Boris V. Sokolov, Anton V. Ustinov, and Alexander E. Zyanchurin // In: Abraham A., Kovalev S., Tarassov V., Snasel V., Sukhanov A. (eds) Proceedings of the Third International Scientific Conference “Intelligent Information Technologies for Industry” (IITI’18). IITI’18 2018. Advances in Intelligent Systems and Computing, vol 875. Springer, Cham pp. 171-180.*
222. *Bakhmut Aleksey D., Krylov Aleksey V., Krylova Margaret A., Okhtilev Michael Yu., Okhtilev Pavel A., and Sokolov Boris V.* *Proactive Management of Complex Objects Using Precedent Methodology // Springer International Publishing AG, part of Springer Nature 2019 R. Silhavy (Ed.): CSOC 2018, AISC 764, 2019. pp. 298–307.*
223. *Berners-Lee T.* *The Semantic Web / T. Berners-Lee, J. Hendler, O. Lassila // Scientific American. – May, 2001. – P. 28–37. – 240 p.*
224. *Bikakis N., Gioldasis N., Tsinaraki Ch., Christodoulakis S.* *Querying XML Data with SPARQL DEXA '09 Proceedings of the 20th International Conference on Database and Expert Systems Applications, 2009. P. 372 - 381*
225. *Booch G.* *Object-Oriented Analysis And Design With Application, second edition. The Benjamin/Cummings Publishing Company, Inc. 1994. 589 pp.*
226. *Business Process Model and Notation (BPMN). Version 2.0.2. OMG Document Number: formal/2013-12-09 [Электронный ресурс]. URL: www.omg.org (дата обращения 31.10.2017). – 532 pp.*
227. *Calof J., Richards G., Smith J.* *Foresight, Competitive Intelligence and Business Analytics — Tools for Making Industrial Programs More Efficient // Forsite, Vol.9. №1. 2015. P. 68-81.*
228. *Calvanese D., Cogrel B., Komla-Ebri S., Kontchakov R., Lanti D., Rezk M., Rodriguez-Muro M., Xiao G.* *Ontop: Answering SPARQL queries over relational databases. Semantic Web, 8(3):471–487, 2017.*

229. *Cygniak R.* A relational algebra for SPARQL. Technical Report HPL-2005-170, HP Labs, 2005. – 20 pp.
230. *Codd E.F., Codd S.B., Salley C.T.* Providing OLAP (On-Line Analytical Processing) to user-analysts: An IT mandate. Technical report, 1993.
231. *Cudré-Mauroux P., Enchev I., Fundatureanu S., Groth P.T., Haque A., Harth A., Keppmann F. L., Miranker D., Sequeda J., Wylot M.* NoSQL Databases for RDF: An Empirical Evaluation // *The Semantic Web – ISWC, 2013. Vol. 2. P. 310-325.*
232. ERP-CRM-SCM-EAM-ECM. Интернет-ресурс. URL: osnova.ru. (дата обращения: 22.03.2018)
233. *Fensel D., Van Harmelen F., Horrocks I.* OIL: An ontology infrastructure for the semantic web. *IEEE intelligent systems* 16 (2), pp. 38-45.
234. *Fiodorov I.G.* Overcoming expressiveness deficit of business process modeling languages // *Business Informatics. 2016. № 3. (37). P. 62–71.*
235. *Gehlert A., Pfeiffer D., Becker J.* The BWW-Model as Method Engineering Theory // *Americas Conference on Information Systems (AMCIS). AMCIS 2007 Proceedings, 83, 2007.* Интернет-ресурс. URL: aisel.aisnet.org (дата обращения: 29.01.2018)
236. IDEF5 Method Report. Information Integration for Concurrent Engineering. Knowledge Based Systems, Inc., 1994. – 175 pp.
237. IEEE Standard 1320.2-1998 for Conceptual Modeling Language – Syntax and Semantics for IDEF1X97 (IDEF object).
238. IEEE Swebok V 3.0 Guide to the Software Engineering. Body of Knowledge. A Project of the IEEE Computer Society. Inetrnet resource. URL: www.computer.org (дата обращения: 22.03.2018).
239. ISO 9001:2015 Quality management systems – Requirements.
240. ISO/IEC 12207:2008. System and software engeneering - Software life cycle processes. Интернет-ресурс. URL: www.iso.org. (дата обращения: 22.03.2018).

241. ISO 10303-11:2004. «Industrial automation systems and integration — Product data representation and exchange — Part 11: Description methods. The EXPRESS language reference manual». Интернет-ресурс. URL: www.iso.org. (дата обращения: 22.03.2018).
242. ISO/IEC 19501:2005 Information technology – Open Distributed Processing – Unified Modeling Language (UML) 1.4.2 (2005).
243. ITU Recommendation Z.100: Specification and Description Language. 1993. – 204 p.
244. *Jacobson I.* Object-Oriented Software Engineering. ASM press. 1992, – 528p.
245. *Kiwelekar, A.W., Joshi, R.K.* An object oriented metamodel for bunge-wand-weber ontology. In: In Proc. of SWeCKa 2007, Workshop on Semantic Web for Collaborative Knowledge Acquisition at IJCAI 2007. 2007. – 8 pp.
246. *Klaus Schwab.* The Fourth Industrial Revolution. What It Means and How to Respond [Internet-source] // Foreign Affairs. Tampa, 2015. URL: www.foreignaffairs.com (дата обращения: 16.02.2018).
247. *Kramer Th., Marks B., Schlenoff C., Balakirsky St., Kootbally Z., Pietromartire A.* Software Tools for XML to OWL Translation // NIST Interagency/Internal Report (NISTIR). National Institute of Standards and Technology, 2015. 32 pp.
248. *Kupriyanovsky V.* Industries transformation in the digital economy – the design and production/ Vasily Kupriyanovsky, Sergei Sinyagov, Dmitry Namiot, Nikita Utkin, Danila Nikolaev, Andrey Dobrynin // International Journal of Open Information Technologies. Vol.5, no.1, 2017. P. 50-70.
249. *McGuinness Deborah L., Nardi Daniele, Patel-Schneider Peter F.* The description logic handbook: Theory, implementation, and applications. Edited by Franz Baader. – 505 pp.
250. *Manyika J.* et al. Big data: The next frontier for innovation, competition, and productivity. McKinsey Global Institute, June, 2011. 156 pp.
251. *Michel F., Djimenou L., Faron-Zucker C., Montagnat J.* xR2RML: Non-Relational Databases to RDF Mapping Language, 2014. – 34 pp.

252. *Murthy S.* Automatic construction of decision trees from data: A multidisciplinary survey. *Data Mining and Knowledge Discovery* // Kluwer Academic Publishers, Boston. Manufactured in The Netherlands. 1998. 49 p.
253. Object Management Group. About the Unified Modeling Language Specification Version 2.5.1 URL: www.omg.org (дата обращения: 15.12.2018)
254. Object Management Group. Business Process Model and Notation. Интернет-ресурс. URL: www.bpmn.org. (дата обращения: 22.03.2018).
255. *Okhtilev Pavel A., Bakhmut Aleksey D., Krylov Aleksey V., Okhtilev Michael Yu., Sokolov Boris V.* Application of decision-making support technology for management of space vehicle life cycle // 2017 IEEE II International Conference on Control in Technical Systems (CTS). IEEE Publ, 2017, P.41-44.
256. OWL 2 Web Ontology Language. Overview. W3C Recommendation, 27 October 2012. Интернет-ресурс. URL: www.w3.org. (дата обращения: 22.03.2018).
257. *Pan Jeff Z.* Description Logics: Reasoning support for the Semantic Web. A thesis submitted to the University of Manchester for the Degree of Doctor of Philosophy in the Faculty of Science and Engineering. 2004. 215 pp.
258. *Paschhenko A., Okhtilev P., Potrysaev S., Ipatov Y., Sokolov B.* Methodology and Structure Adaptation Algorithm for Complex Technical Objects Reconfiguration Models // *Advances in Intelligent Systems and Computing*. Vol.574. Cybernetics and Mathematics Applications in Intelligent Systems. Proceedings of the 6th Computer Science On-line Conference, 2017. Vol 2. P.319-328.
259. *Peter Pin-Shan Chen.* The Entity-Relationship Model-Toward a Unified View of Data. *ACM Transactions on Database Systems*, Volume 1, Number 1, 1976. P. 9-36.
260. *Poggi A., Lembo D., Calvanese D., De Giacomo G., Lenzerini M., Rosati R.* Linking data to ontologies // *Data Semantics*, 2008, С.133-173.
261. Predicts 2007: SOA Advances, 17 November 2006. Интернет-ресурс. URL: www.intuit.ru. (дата обращения: 22.03.2018).

262. *Rasmey Heang and Raghul Mohan*. Literature review of business intelligence. P.1-10; Wixom, B. and Watson, H. 2010. The BI-Based Organization. International Journal of Business Intelligence Research, Vol. 1(1), pp.12-24.
263. Resource description framework: concepts and abstract syntax. W3C Recommendation, 25 February 2014. Интернет-ресурс. URL: www.w3.org. (дата обращения: 22.03.2018).
264. *Rosemann M. and Wyssusek B*. Enhancing the Expressiveness of the Bunge-Wand-Weber Ontology (2005). AMCIS 2005 Proceedings. Paper 438. P. 2802 – 2810.
265. *Rosemann M., Vessey I., Weber R., Wyssusek B*. On the Applicability of the Bunge–Wand–Weber Ontology to Enterprise Systems Requirements // ACIS 2004 Proceedings. 78, 2004. – 11 p.
266. *Rudolph S*. Foundations of Description Logics. Karlsruhe Institute of Technology Germany. 2011. – 72 pp.
267. *Selic B., Gullekson G., Ward P.T*. Real-Time Object-Oriented Modeling. John Wiley & Sons. Inc. 1994. – 525 pp.
268. Service Oriented Architecture (SOA). Chapter 1. Интернет-ресурс. URL: msdn.microsoft.com. (дата обращения: 22.03.2018).
269. Service-Oriented Architecture Standards - The Open Group. Интернет-ресурс. URL: www.opengroup.org. (дата обращения: 22.03.2018).
270. Simatic IT. Интернет-ресурс. URL: siemens.com. (дата обращения: 22.03.2018).
271. The Open Group: Making Standards Work. Интернет-ресурс. URL: www.opengroup.org. (дата обращения: 22.03.2018).
272. The World Wide Web Consortium (W3C). Интернет-ресурс. URL: www.w3c.org. (дата обращения: 22.03.2018).
273. Trace Mode. Интернет-ресурс. URL: adastra.ru. (дата обращения: 22.03.2018).
274. *Vicente García Díaz*. Progressions and Innovations in Model-Driven Software Engineering 1 edition. IGI Global publ, 2013. – 388 p.

СПИСОК ИЛЛЮСТРАЦИЙ И ТАБЛИЦ

Рис. 1.1. Уровни организации информационного взаимодействия	23
Рис. 1.2. Структура задачи диссертационного исследования.....	28
Рис. 1.3. Основные CALS-технологии, реализуемые при формировании интегрированной информационной среды.....	32
Рис. 1.4. Требования, предъявляемые к автоматизированной информационно-аналитической поддержке ЖЦ СОТО.....	60
Рис. 1.5. Коммутативная диаграмма задачи автоматизированной информационно-аналитической поддержки ЖЦ КСр как СОТО.....	63
Рис. 2.1. Треугольник Г. Фреге	72
Рис. 2.2. Диаграмма онтологической системы информационно-аналитической поддержки ЖЦ СОТО.....	79
Рис. 2.3. Схема взаимосвязей концептов и ролей онтологической системы информационно-аналитической поддержки ЖЦ СОТО	83
Рис. 2.4. Пример отношения следования между событием и действием	95
Рис. 2.5. Пример бизнес-процесса информационно-аналитической поддержки ЖЦ СОТО.....	96
Рис. 2.6. Пример связи по входу и выходу некоторого запроса.....	107
Рис. 2.7. Пример модели согласованных вычислительных задач.....	109
Рис. 2.8. Графическая интерпретация структурно-потокowo-многоуровневого распознавания образов.....	111
Рис. 2.9. Операторная схема (а) и схема (б) G-модели.....	114
Рис. 2.10. Пример отношения следования индивидов переменных и вычислительной задачи	115
Рис. 2.11. Блок-схема алгоритма формирования G-модели по модели согласованных вычислительных задач (часть 1)	119
Рис. 2.12. Блок-схема алгоритма формирования G-модели по модели согласованных вычислительных задач (часть 2)	120
Рис. 2.13. Пример G-модели, сформированной по СВЗ-модели.....	122

Рис. 2.14. Схема оценивания технического состояния СОТО	123
Рис. 2.15. Пример G-сети.....	127
Рис. 2.16. Блок-схема алгоритма трансляции описания запроса в СВЗ-модели на язык SPARQL.....	131
Рис. 2.17. Схема порядка формирования моделей информационно-аналитической поддержки ЖЦ СОТО	133
Рис. 2.18. Пример модели бизнес-процессов информационно-аналитической поддержки ЖЦ СОТО.....	137
Рис. 3.1. Дерево поиска табло-алгоритма для примера модели бизнес-процесса	152
Рис. 3.2. Взаимосвязь показателей онтологической выразительности	153
Рис. 3.3. Показатели онтологической выразительности, отражающие связь онтологии и примитивы нотации моделирования	154
Рис. 3.4. Примеры наличия дефицита, избыточности и неразличимости несогласованных спецификаций	155
Рис. 3.5. Блок-схема алгоритма проверки онтологической выразительности (часть 1)	157
Рис. 3.6. Блок-схема алгоритма проверки онтологической выразительности (часть 2)	158
Рис. 3.7. Иллюстрация обобщенной спецификации вычислительной задачи	160
Рис. 3.8. Иллюстрация условий отсутствия неразличимости.....	161
Рис. 3.9. Фрагменты G-сетей с некорректностью в позиции <i>a</i> : (а) - по выходу и (б) - по входу.....	165
Рис. 3.10. Схема электронного паспорта с учетом временной характеристики	175
Рис. 4.1. Задача формирования заявления о соответствии изделия требованиям заказчиков	185
Рис. 4.2. Пример электронной регистрационной карточки документа	188
Рис. 4.3. Порядок взаимодействия элементов АПК ЕВЭП.....	189

Рис. 4.4. Интерфейс ПО АРМ доступа к информации из электронного паспорта РН «Союз-2»	190
Рис. 4.5. Пример составной части ЭСИ изделия РН «Союз-2».....	191
Рис. 4.6. Пример электронной карты-ведомости контроля исполнения документов при производстве РН «Союз-2».....	193
Рис. 4.7. Языки и инструменты разработки онтологических моделей.....	196
Рис. 4.8. Схема основных компонент ПК проектирования моделей информационно-аналитической поддержки ЖЦ СОТО	197
Рис. 4.9. Пример модели бизнес-процессов информационно-аналитической поддержки ЖЦ РН «Союз-2» на этапах проектирования и производства....	201
Рис. 4.10. Пример модели бизнес-процесса информационно-аналитической поддержки ЖЦ РН «Союз-2» при обработке метаданных документа «Карточка разрешения».....	203
Рис. 4.11. Пример СВЗ-модели анализа изменения конфигурации изделия по метаданным документа «Акт замены».....	204
Рис. 4.12. Пример СВЗ-модели оценивания ТС СОТО по метаданным документа «Карточка разрешения».....	204
Рис. 4.13. Пример G-модели анализа изменения конфигурации изделия по метаданным документа «Акт замены».....	205
Рис. 4.14. Пример G-модели оценивания ТС СОТО по метаданным документа «Карточка разрешения».....	205
Рис. 4.15. Пример онтологической схемы метаданных об изделии	206
Рис. 4.16. Онтология, сформированная по БД СУЭД	207
Рис. 4.17. Онтология электронного паспорта РН «Союз-2».....	208
Рис. 4.18. График зависимости времени, затраченного на информационно-аналитическую поддержку от стадии ЖЦ изделия	210
Рис. 4.19. Столбчатая гистограмма сравнения количества оцениваемых информационных единиц на разных этапах ЖЦ изделия	211
Рис. 4.20. Лепестковая диграмма сравнения эффективности рассматриваемых способов информационно-аналитической поддержки.....	213

Таблица 1.1. Требования, предъявляемые к OLTP и OLAP-системам.....	35
Таблица 1.2. 12 требований Э. Кодда к OLAP-системам.....	36
Таблица 1.3. Характерные черты модели ИИС.....	40
Таблица 1.4. Виды программной поддержки интеграции информационных ресурсов.....	40
Таблица 1.5. Сравнительный анализ моделей решения задачи информационно-аналитической поддержки ЖЦ СОТО	62
Таблица 1.6. Соотношение этапов ЖЦ ПО, элементов модельно-алгоритмического комплекса СИАП ЖЦ СОТО и требований, предъявляемых к ним	67
Таблица 2.1. Связь элементов нотации BPMN и концептов и ролей онтологии бизнес-процессов информационно-аналитической поддержки ЖЦ СОТО ...	89
Таблица 2.2. Описание элементов нотации моделей согласованных вычислительных задач.....	99
Таблица 2.3. Матрица инцидентности $R_s +$	127
Таблица 2.4. Матрица инцидентности $R_s -$	128
Таблица 3.1. Правила табло-алгоритма	145
Таблица 3.2. Табло-алгоритм, описанный на мнемоязыке	146
Таблица 3.3. Пример работы табло-алгоритма при проверке выполнимости фактов, описывающих модель бизнес-процесса.....	149
Таблица 4.1. Некоторые примеры информации, обрабатываемой в рамках СИАП ЖЦ СОТО	182
Таблица 4.2. Основные этапы проектирования СИАП ЖЦ СОТО	198

СПИСОК ИЛЛЮСТРАЦИЙ И ТАБЛИЦ В ПРИЛОЖЕНИЯХ

Рис. П. 1.1. Использование АС на различных этапах ЖЦ изделия.....	276
Рис. П. 1.2. Схема алгоритма управления конфигурацией СОТО	280
Рис. П. 1.3. Пример организации единого информационного пространства организаций – участников ЖЦ СОТО.	281
Рис. П. 1.4. Использование ИИС для задач на различных этапах жизненного цикла СОТО	282
Рис. П. 1.5. Компоненты системы управления качеством	283
Рис. П. 1.6. Концептуальная схема информационных ресурсов в стандарте STEP.....	292
Рис. П. 1.7. Схема типовой сущности, описываемой языком EXPRESS	292
Рис. П. 1.8. Общий вид динамической системы	297
Рис. П. 1.9. Проблема наблюдения состояния динамической системы	299
Рис. П. 1.10. Структура теоретических основ и методов автоматизированного анализа информации о ТС СОТО на основе G-моделей.....	306
Рис. П. 1.11. Некоторые примеры операций на многомерном гиперкубе.....	311
Рис. П. 1.12. Классификация методов интеграции информационных ресурсов по Клаусу Диттриху	313
Рис. П. 2.1. Описательные возможности CASE-систем.....	334
Рис. П. 2.2. Типовая структура экспертной системы.....	342
Рис. П. 2.3. Классификация моделей представления знаний.....	344
Рис. П. 2.4. Подходы к представлению знаний	345
Рис. П. 2.5. Принцип функционирования производственной системы	348
Рис. П. 2.6. Схема производственно-фреймового представления знаний (P – некоторый алгоритм)	349
Рис. П. 2.7. Семиотические аспекты языков программирования.....	354
Рис. П. 2.8. Уровни интеграции информационных ресурсов	359
Рис. П. 2.9. Графическая интерпретация метода обобщенного портрета	365
Рис. П. 2.10. Графическая интерпретация статистического метода распознавания для тривиального случая	366

Рис. П. 2.11. Графическая интерпретация метода потенциальных функций	368
Рис. П. 2.12. Общий вид таблицы обучения при распознавании на основе вычисления оценок	369
Рис. П. 3.1. Типовой граф триплета модели данных RDF	382
Рис. П. 3.2. Иерархия языков Семантического Веб по уровням представления знаний	384
Рис. П. 3.3. Соотнесение элементов дескрипционной логики DL-Lite _A и элементов схемы реляционной БД.....	391
Рис. П. 3.4. Соотнесение элементов дескрипционной логики DL-Lite _A и элементов схемы XML.....	393
Рис. П. 3.5. Схема архитектуры системы интеграции данных и приложений для задачи информационно-аналитической поддержки ЖЦ СОТО	396
Рис. П. 3.6. Блок-схема алгоритма выбора источника данных и исполнения запроса.....	398
Рис. П. 3.7. Типовая схема электронного паспорта СОТО на основе онтологического подхода.....	402
Рис. П. 4.1. Схема БД СУЭД.....	406
Таблица П. 1.1. Назначение АС на различных этапах ЖЦ СОТО.....	274
Таблица П. 1.2. Примеры внедрения CALS-технологий и используемых при этом методов моделирования.....	287
Таблица П. 1.3. Стандарты описания технологий информационной поддержки жизненного цикла изделий.....	289
Таблица П. 1.4. Состав моделей прикладного протокола STEP	294
Таблица П. 1.5. Анализ возможностей применения G-моделей для задачи проектирования СИАП ЖЦ СОТО.....	307
Таблица П. 2.1. Некоторые критерии, определяемые для спецификаций требований	318
Таблица П. 2.2. Типы требований к информационным системам	319
Таблица П. 2.3. Согласование требований различных уровней.....	321

Таблица П. 2.4. Процессы ЖЦ ПО согласно SWEBOOK	323
Таблица П. 2.5. Классификация уровней представления знаний и их моделей	336
Таблица П. 2.6. Типовые этапы наполнения базы знаний	342
Таблица П. 2.7. Системный анализ МПЗ	351
Таблица П. 2.8. Сравнительная таблица модели и алгоритма	352
Таблица П. 2.9. Сравнительный анализ методов распознавания образов.....	373
Таблица П.2.10. Сопоставление диалектов дескрипционной логики.....	375
Таблица П.2.11. Трансляция синтаксических конструкций языка OWL в дескрипционную логику.....	376
Таблица П. 4.1. Текст SQL-запроса, позволяющего по реляционной БД сформировать схему её онтологии в терминах OWL.....	404

1. ТЕХНОЛОГИЧЕСКИЕ АСПЕКТЫ ИНФОРМАЦИОННОЙ ПОДДЕРЖКИ ЖИЗНЕННОГО ЦИКЛА СЛОЖНЫХ ОБЪЕКТОВ

1.1. Сложные организационно-технические объекты

КК и входящие в их состав изделия относятся к классу СОТО по системно-кибернетическому определению сложных систем [31] (см. определение П. 1.1). Составные элементы СОТО также образуют системы (организации, отделы, само изделие и т.д.). Такие объекты сочетают в себе как технические, так и организационные аспекты. Это означает, что в составе системы элементы могут быть представлены как аппаратно-программными и информационными комплексами, так и людьми, и система может иметь некоторую сложную структуру отношений между элементами [91, 142, 143, 153].

Определение П. 1.1. Под *СОТО* понимают искусственную, самоорганизующуюся, динамическую организационно – техническую полиструктуру, состоящую из взаимосвязанных элементов и предназначенную для некоторой целенаправленной деятельности, осуществляемой человеком [135, 183, 185].

Функционирование СОТО характеризуется сверхбольшими объемами информационных ресурсов различной природы, неполнотой и недоопределенностью информации о состоянии, высокой стоимостью ошибок, наличием директивного времени на принятие решения при управлении, наличием различного рода взаимосвязанных структур, изменяющихся во времени (структурная динамика): организационных структур предприятий, структур изделий, функциональных структур и пр.

К СОТО в настоящее время, безусловно, можно отнести существующие и проектируемые территориально распределенные производственные и сервисные объекты, входящие в состав корпораций и крупных предприятий, многочисленные транспортные и логистические системы, городское хозяйство и

государственные структуры, органы военного управления, системы в космической отрасли, высококонкурентные сферы экономики и пр., так как все они имеют целый ряд особенностей, характерных для СОТО [182].

СОТО представляют собой заданную совокупность взаимосвязанных программных, аппаратных и организационных (информационных, математических, лингвистических и других) средств и включают [62]:

- информационные ресурсы, содержащие информацию, необходимую для реализации основных процессов;
- средства и системы информатизации, программные средства (операционные системы, системы управления БД (СУБД) и др.), автоматизированные системы управления (АСУ), системы связи и передачи данных, осуществляющие приём, обработку, хранение и передачу информации;
- технические средства и системы, обрабатывающие информацию;
- персонал, обеспечивающий деятельность СОТО;
- систему нормативно-технических документов, регламентирующих применение и функционирование сложных объектов.

В последние годы много говорится о необходимости автоматизации процессов мониторинга состояния и управления СОТО. Этой теме посвящено много теоретических и прикладных работ [137, 138]. Однако, выработка управляющих решений до сих пор большей частью осуществляется человеком на вербальном уровне. А специальное ПО имеет узкую специализацию и помогает автоматизировать решение лишь некоторых частных задач.

Таким образом, КК и входящие в их состав изделия (в т.ч. КСр) необходимо рассматривать как интерпертацию СОТО в критическом приложении.

1.2. Метаданные и большие данные

СОТО характеризуется сверхбольшими объемами информации различной природы, которая характеризует состояние КК и входящих в их состав изделий на всех этапах их ЖЦ. Такая информация представлена на различных носителях и формируется в различных источниках:

- различного вида документы, характеризующие процессы, мероприятия и результаты их проведения в ходе проектирования, производства, эксплуатации и утилизации изделий;
- телеметрируемая информация, связанная с наличием киберфизических систем в составе СОТО: это и автоматизированное производство, и множество АС, используемых при эксплуатации изделия;
- неструктурированные данные, характеризующие организационные аспекты функционирования СОТО.

Необходимо также отметить, что такая информация характеризуется недоопределенностью, неполнотой. Кроме того, часть информации зачастую слабо структурирована или не структурирована вообще. В процессе сбора, обработки и анализа этой информации в целях ППР при управлении ЖЦ СОТО необходимо учитывать её полностью, обнаруживая связи между информационными единицами.

В таких условиях необходимым является применение концепции «Big Data» (большие данные). «По существу, концепция больших данных не имеет точного определения – это ... объединение различных направлений исследований, как практических, так и теоретических», связанных с аналитической обработкой данных в целях получения существенной, ценной для задачи информации [126]. Большие данные принято характеризовать пятью «V»-свойствами:

- *Volume*. Большие объемы данных. При этом, говоря об объеме, подразумевают большие объемы, трудно обрабатываемые в контексте решаемой задачи;
- *Variety*. Многообразие данных в различных контекстах:
 - степени структурированности;
 - семантического многообразия моделей данных и языков доступа к ним;
 - неоднородности доступа и хранения в автономных распределенных системах;
 - динамики (многообразных потоков данных);

- надежности и достоверности в контексте выбора источников данных;
- *Velocity*. Своевременность получения результатов аналитической обработки. Балансирование между задачами поиска точного и быстрого результата, что особенно актуально для задач анализа в режиме РВ, когда результат необходимо получить за предсказуемое время;
- *Veracity*. Степень доверия к результатам анализа. Необходимость оценивания достоверности получаемых в результате анализа данных.

В работе [126] отмечается, что существуют 2 общих подхода аналитической обработки в контексте Big Data:

- использование относительно низкоуровневых объектно-ориентированных языков, что существенно затрудняет проверку соответствия полученного программного кода и проверку корректности результатов;
- использование декларативных языков описания сценариев аналитической обработки, ориентированных на организацию последовательных/параллельных моделей вычислений. «Фактически высокоуровневые декларативные языки спецификации сценариев аналитической обработки ... являются расширениями реляционных языков дополнительными операторами и предикатами, упрощающими решение типичных аналитических задач».

Для реализации потенциала расширенных декларативных языков описания сценариев аналитической обработки требуется существенный пересмотр концепций и алгоритмов их оптимизации и выполнения. В частности, необходимы новые подходы в определении представления запросов и сценариев, а также новые методы оптимизации, учитывающие наличие большого количества операций, определяемых пользователем.

При синтезе систем, основанных на Big Data, необходимо также учитывать неоднородность и распределенность источников данных. В связи с этим актуальным становится применение межмашинных технологий (Machine to Machine — M2M) за счет которых может быть организованы автоматический приём, передача данных между вычислительными устройствами с целью их дальнейшей обработки.

Среди прочего к методам и техникам, применяемым в Big Data, согласно [250], относятся: распознавание образов, прогнозная аналитика, имитационное моделирование, визуализация аналитических данных в виде интерактивных моделей (рисунков, диаграмм, 3D-графики).

Хранение и обработка данных в компьютерных системах, обмен данными между ними и доступ к ним пользователей невозможны без явно представленных свойств этих данных. Такие описания необходимы программным средствам, выполняющим указанные функции, а также пользователям для оценки возможности применения имеющихся данных в различных ситуациях, интерпретации и анализа их содержания, формулировки запросов. Описания такого рода называются метаданными и являются особым видом информационных ресурсов [20, 85]. Строгое определение для метаданных на сегодня отсутствуют. Наиболее распространенным является определение, согласно которому: «метаданные – это данные о данных».

Использование метаданных продиктовано необходимостью структурированного описания структурированных, слабоструктурированных и неструктурированных данных с целью доступа к ним и манипулирования ими. Такими метаданными могут выступать схемы БД, концептуальные модели ПрО или *онтологии* (см. приложение 2.3.5) [85].

Для решаемой задачи наиболее интересны языки семантических метаданных, характеризующие описание данных в ПрО. Выбор таких языков будет обсуждаться далее. В век «больших данных» (Big Data) специалистам бывает довольно трудно разобраться в имеющихся разнообразных и многочисленных информационных ресурсах [41]. Для представления сведений о ресурсах (в частности, о документах или БД) и их интеграции было предложено понятие метаданных, описывающих содержимое ресурса в виде специального краткого набора данных, описывающих также и связи с другими ресурсами. Метаданные используются для автоматизированного анализа содержимого ресурса, построения поисковых индексов и позволяют обеспечить достаточно высокую точности и эффективности поиска разнородной информации.

1.3. Анализ существующих технологий, применяемых при интеграции и анализе информационных ресурсов предприятий

Современные предприятия характеризуются наличием различного рода АС. К числу таких систем можно отнести [186, 232, 246, 273, 270]:

- АС управления (АСУ) технологическими процессами. Примеры реализации (Trace Mode, Master SCADA, Simatic WinCC (Siemens), Citect SCADA, InTouch);
- АСУ производством. Примеры реализации (T-Factory, 1С:MES Оперативное управление производством, Simatic IT (Siemens));
- АСУ предприятием. Примеры реализации (BAAN, R/3 (SAP), Oracle Applications, SAS System, 1С: Управление производственным предприятием);
- АСУ данными о продукции. Примеры реализации (1С: Интегратор, 1С: PDM Управление инженерными данными, Lotsia PDM Plus, T-FLEX DOCs, TechnologiCS, Windchill, Smar-Team, IDPM CADISON PDM, SolidWorks Enterprise PDM);
- АСУ взаимоотношениями с клиентами. Примеры реализации (Microsoft Dynamics CRM, Oracle CRM On Demand, NetSuite CRM);
- АСУ основными фондами предприятия. Примеры реализации (Mincom Ellipse, Maxim, EAM модуль в составе SAP ERP, EAM модуль в составе Oracle eBusiness Suite);
- АСУ цепями поставок. Примеры реализации (SAP Advanced Planner & Optimizer (APO), модуль в iBAAN, модуль в Ахарта, модуль в QAD);
- АСУ корпоративной информацией. Примеры реализации (Documentum, OpenText, Directum, Optima workflow);

и др.

Вопрос интеграции и анализа данных подобных систем рассматривается в области исследований, называемой Business Intelligence (BI) [67, 227, 262]. В рамках данного направления также разрабатываются *системы оперативных аналитических вычислений (OLAP*, см. раздел 1.2.3), системы оперативной обработки транзакций, а также системы управления ЖЦ изделий, выступающие

как интегрирующие платформы для обозначенных выше АС. Лидерами на рынке таких систем являются Teamcenter (Siemens PLM Software), Agile PLM (Oracle), решения от Dassault Systemes. Данные решения являются дорогостоящими и сложно адаптируемыми, поэтому область их внедрения ограничивается на сегодняшний день крупными предприятиями и концернами. Следует отметить, что указанные системы являются проприетарными программными продуктами, технологические и методологические решения которых (модели, методы, алгоритмы) не раскрываются. Кроме того, внедрение указанных систем связано с необходимостью регулярных выплат лицензионных отчислений зарубежным компаниям за использование, техническую поддержку, расширение и адаптацию функционального наполнения прикладных решений под задачи отечественных организаций и предприятий.

Среди основных подходов при создании интегрированных *корпоративных ИС* можно отметить два основных направления: интеграция приложений (под которой в рамках теории ИИ можно понимать интеграцию знаний) и интеграция данных [174] (см. раздел 1.2.4). Первые попытки интеграции приложений были основаны на применении протоколов TCP и UDP транспортного уровня сетевой модели OSI и механизма вызовов удаленных процедур (RPC) [56, 95]. Описание взаимодействия приложений осуществлялось с помощью языка определения интерфейсов IDL, а с помощью специального протокола XDR обеспечивалось унифицированное представление данных. Стандарты были определены консорциумом OpenGroup [271]. С развитием объектно-ориентированной парадигмы (ООП) консорциумом Object Management Group (OMG) была предложена технология CORBA, являющаяся расширением RPC. Однако, применение этих подходов было сопряжено с рядом трудностей, таких как: высокая сложность внедрения, сложность адаптации, высокая стоимость, необходимость в высококвалифицированных специалистах по данным технологиям, поскольку для включения нового приложения в существующую интегрированную среду было необходимо модифицировать исходные коды программного обеспечения (ПО), что представляется трудоемким процессом.

Другим подходом интеграции приложений стали Web-сервисы (стандартизованы консорциумом W3C, [272], в рамках которых сформировалось понятие сервис-ориентированной архитектуры (COA), предполагающей модульный подход к построению информационной корпоративной инфраструктуры [189]. COA хорошо зарекомендовала себя при построении крупных корпоративных программных систем. Целый ряд разработчиков и интеграторов предлагают инструменты и решения на основе COA (например, платформы Microsoft .NET, IBM WebSphere, SAP NetWeaver, Diasoft и др.) [268, 269]. По данным Gartner Group (всемирно известной консалтинговой компании в области информационных технологий) [261]: "К 2008 году COA станет господствующей архитектурой построения информационных систем, что приведет к окончанию 40-летней эры господства архитектуры монолитных приложений". Отметим, что этот прогноз в большой степени оправдался. Важным элементом COA является интеграционная шина предприятия (ESB), позволяющая динамически наполнять интегрируемые решения новыми сервисами. Кроме того, Web-сервисы задействуют следующие стандартизованные протоколы: XML - расширяемый язык разметки, предназначенный для хранения и передачи структурированных данных; SOAP - протокол обмена сообщениями на базе XML; WSDL - язык описания внешних интерфейсов веб-сервисов на базе XML; UDDI – универсальный интерфейс распознавания, описания и интеграции.

Заметим, что появление технологии Web-сервисов активизировало еще одно направление интеграции корпоративных приложений, основанное на давно известной концепции *потоков работ* (workflow) (см. приложение 1.5.8). В рамках этого направления появились сценарные языки высокого уровня абстракции, позволяющие на концептуальном уровне описывать потоковые параллельные событийно-ориентированные бизнес-процессы организаций и предприятий, однако вопрос формализации и строгого использования таких бизнес-процессов остается открытым [254].

Важным элементом ИАС поддержки ЖЦ организаций и их изделий, являются *CALS-технологии* (см. раздел 1.2.1.2). CALS-технологии представляют собой современную организацию процессов разработки, производства, послепродажного сервиса, эксплуатации изделий путем информационной поддержки процессов их ЖЦ на основе стандартизации методов представления данных на каждой стадии ЖЦ и безбумажного электронного обмена данными [161]. При этом программные продукты, основанные на CALS-технологиях, являются комплексными АС и должны предоставлять лингвистическое, информационное, математическое, программное, методическое и техническое обеспечение. Предполагается, что достижение существенных результатов от внедрения CALS возможно благодаря системе стандартов, к которым относятся стандарты функционального моделирования IDEF, стандарт структур, конфигураций изделий STEP, стандарт форматов данных PLIB и др. Важным элементом CALS является электронное описание изделия (являющееся в перспективе «цифровым двойником», инвариантным реальному объекту) [28]. В соответствии с международным стандартом STEP, описание информационных моделей подсистем и их взаимодействия в организациях и предприятиях предлагается проводить с помощью визуального языка EXPRESS [241] (см. приложение 1.7.2). Стоит отметить, что описание функционального взаимодействия на языке Express по синтаксису находится близко к языкам программирования высокого уровня, что не обеспечивает возможности проектирования таких систем прикладными специалистами (экспертами) Про.

Отдельно необходимо отметить комплекс работ по автоматизации и интеллектуализации процессов создания современных АС и ИС (в том числе и ИАС, входящих в их состав), проводимых в РФ и за рубежом. Для совершенствования процесса управления состоянием находящихся в эксплуатации изделий и в развитие главной идеи новой технологии управления ЖЦ сложных объектов (изделий), получившим на западе условное наименование - Closed Loop Lifecycle Management (CL2M, управление ЖЦ с обратной связью), свя-

занной с обеспечением принятия управленческих решений в течение ЖЦ изделия на основе глубоких знаний в Европейском Союзе в 2005 – 2009 гг. был выполнен другой (наряду с проектом PROMISE) комплекс исследований под общим названием DYNAMITE – Dynamic Decisions in Maintenance, направленный на разработку перспективного метода, основанного на принятии решений о необходимости и объеме обслуживания (ремонта) изделия в реальном времени в процессе его работы [215].

Другим важным вопросом является доступ к комплексу распределенных БД и банков данных для множества ИС в составе организаций. Использование БД и банков данных позволяют хранить в структурированном виде большие объемы информации, характеризующие состояние СОТО на всех этапах их ЖЦ, а также осуществлять контроль целостности, избыточности и непротиворечивости хранимой информации. При этом, в составе системы интеграции могут присутствовать различные по своей структуре модели данных. Современные подходы к проектированию распределённых БД предполагают, что проектирование начинается с синтеза концептуальной схемы БД. Важным свойством концептуальных схем является возможность описания структуры сущностей ПрО независимо от используемых моделей данных и конкретных технологических решений. При этом проектирование БД и банков данных на основе концептуальных схем в настоящее время не обеспечивает возможность проектирования, наполнения и взаимодействия с БД в терминах ПрО, что обеспечило бы возможность проектирования систем непосредственно экспертами на доступном им тезаурусе.

Одним из распространенных на сегодняшний день подходов к проектированию концептуальных схем БД является использование ER-моделей, предложенных П. Ченом в 1976 году [259]. Проектирование при помощи ER-моделей предполагает возможность синтеза схем БД по одной из известных моделей данных (реляционной, сетевой, иерархической и др.), что является неоспоримым преимуществом для систем интеграции, т.к. нередко в составе ИС присутствуют БД, основанные на различных моделях. В то же время ER-

модель обладает ограниченными возможностями формализации ограничений целостности данных, а отсутствие методологии хода проектирования приводит к разночтениям спроектированных концептуальных схем.

Другим подходом является использование стандарта IDEF1X, разработанного Т. Рэмеем на основе подхода П. Чена [34, 237]. IDEF1X позволяет проектировать модели данных, соответствующие классу реляционных, что затрудняет использование ее в системах интеграции ввиду разнообразия используемых моделей. Основным преимуществом IDEF1X, по сравнению с другими методами проектирования реляционных БД, является строгая стандартизация процесса моделирования, которая позволяет избежать возможности различных трактовок спроектированной модели.

Распространенным подходом к проектированию концептуальных схем БД является использование нотации Unified Modeling Language (UML), разработанной и развиваемой консорциумом OMG [92, 242]. Использование UML при проектировании концептуальных схем БД в контексте единого терминологического представления может быть преимуществом только в том случае, если проектирование ИС так же опиралось на нотацию UML. При этом известно, что нотация UML требует адаптации для проектирования концептуальных схем БД [117, 122, 218].

Взаимодействие с комплексом БД и банков данных неразрывно связано с задачей разработки методов манипулирования данными, т.е. моделированием процессов получения, записи, удаления и изменения данных в комплексе распределенных БД для интеграции информации из них. При этом необходимо учитывать, что модели данных, задействованные в информационном поле организаций, могут отличаться. Эта задача предполагает использование соответствующих требованиям проекта языков манипулирования данными.

На практике широкое распространение получил язык запросов Structured Query Language (SQL) [63]. Важным преимуществом языка SQL является его декларативность, позволяющая представлять запросы в виде задаваемых огра-

ничений к запрашиваемым данным, при этом запросы в автоматическом режиме могут быть интерпретированы в схемы программ исполнения запросов. Однако, во-первых, язык SQL позволяет моделировать взаимодействие только с реляционными БД, вследствие чего отсутствует возможность его использования для манипулирования данными в БД на различных моделях, а, во-вторых, несмотря на возможность автоматической генерации схем программ запросов, выразительные возможности языка SQL затрудняют его применение экспертами ПрО.

Для унификации доступа к БД компанией Microsoft была разработана технология Open Database Connectivity (ODBC) [210]. ODBC позволяет обеспечить доступ к данным из различных источников на основе стандартизованных протоколов: БД, файловые хранилища, электронные таблицы, электронная почта, системы документооборота и др. С одной стороны, это является значительным преимуществом в рамках задачи моделирования процессов манипулирования данными из гетерогенных источников, с другой - применение технологии ODBC ограничено только терминами языков программирования высокого уровня, что значительно затрудняет использование данной технологии экспертами.

Таким образом, существующие подходы и методы проектирования систем интеграции гетерогенной информации из распределённых источников данных затрудняют решение задачи автоматизированного проектирования интегрированных моделей распределённых БД и банков данных, а также моделей процессов манипулирования данными в терминах ПрО непосредственно экспертами. Ввиду этого, необходимо разработать принципиально новую технологию, основанную на методах теории ИИ, использовании МПЗ, языков концептуального моделирования, ориентированных на задачу интеграции информационных ресурсов.

Также получила широкое распространение в рамках задачи моделирования интеллектуальных систем интеграции и анализа информации из распределённых источников данных теория *мультиагентных систем* [111, 176] (см.

приложение 2.3.6). Важно, что суть теории мультиагентных систем сводится к взаимодействию распределенных интеллектуальных автономных агентов, решающих частные задачи и обладающих лишь нужной частью информации о состоянии объектов, явлений и процессов в ПрО. Это свойство отвечает требованиям моделирования и унифицированного представления разнородных информационных ресурсов, данные из которых составляют цифровое пространство организаций и их изделий. Таким образом, применение мультиагентных систем и их моделирование может существенно повысить удобство проектирования прикладного ПО рассматриваемых систем с опорой на СОА.

В настоящее время назрела острая необходимость создания такой технологии моделирования аналитической деятельности, которая позволила бы на глубинном, модельно-алгоритмическом уровне объединить все ранее выполненные разработки, усилив, тем самым, их достоинства за счет компенсации соответствующих им недостатков альтернативными подходами к созданию тех или иных подсистем ИАС, созданным с использованием разнотипного математического аппарата.

В рамках перечисленных ранее разработок, выполненных в академических, учебных, промышленных и коммерческих организациях, построено большое количество моделей и алгоритмов, базирующихся на математическом аппарате теории систем, исследования операций, ИИ, теории принятия решений, теории управления, теории прогнозирования и т.д. До последнего времени при синтезе ИАС объединение разнотипного модельно-алгоритмического обеспечения, построенного с использованием перечисленных ранее теорий, проводилось, как правило, на программно-техническом уровне, когда выходные данные (результаты), полученные в одной модели, передавались в качестве входных данных для другой модели.

Однако для корректности использования даже на таком уровне разнотипных моделей, описывающих различные аспекты функционирования объектов в ПрО, необходимо было обеспечить согласование причинно-следствен-

ных связей (отношений), описывающих специфику моделируемой ПрО в каждой из моделей. Такого рода процедур при создании ИАС, к сожалению, до сих пор не проводилось, что не обеспечивало, в конечном счете, достоверность результатов вычислений. Поэтому на практике к результатам, полученным с помощью существующих ИАС, относятся с недоверием. Существующие средства интеграции при этом не ориентированы на семантические аспекты интегрируемых информационных ресурсов и не используются совместно со средствами анализа данных, чтобы обеспечить их согласованность на глубинном уровне с целью проведения в рамках соответствующих ИАС всестороннего анализа, гарантирующего полноту, достоверность и целостность искомой информации.

1.4. Некоторые аспекты концепции информационной поддержки изделий

Современные исследования показывают, что в основу интеграции информационных ресурсов предприятий должна быть положена CALS-технология комплексной компьютеризации различных сфер деятельности предприятий, позволяющая консолидировать данные разнообразных АС с целью их обобщения, последующего анализа и предоставления этой информации ЛПР для оптимизации процессов, связанных с ЖЦ изделий.

Основная задача, решаемая путем применения CALS-технологий, – экономия времени и средств при одновременном повышении качества. Так, в США применение CALS-технологий сопровождается следующими типовыми показателями [68, 195]:

- в процессах проектирования и инженерных расчетах: сокращение времени проектирования на 50%; снижение затрат на изучение выполнимости проектов – 15-40%;
- в процессах организации поставок: уменьшение количества ошибок при передаче данных на 98%; сокращение времени поиска и извлечения данных –

- 40%; сокращение времени планирования – 70%; сокращение стоимости информации – 15-60%;
- в производственных процессах: сокращение производственных затрат на 15-60%; повышение показателей качества – 80%;
 - в процессах эксплуатационной поддержки изделий: сокращение времени на изменения технической документации на 30%; сокращение времени планирования поддержки – 70%; снижение стоимости технической документации – 10-50%.

Исходное положение CALS-технологий – на всех этапах ЖЦ изделия необходимо достигать поставленных целей с максимальной эффективностью. Так, например, на этапах проектирования и производства нужно обеспечить выполнение ТЗ при заданной степени надежности изделия и минимизации материальных и временных затрат. Понятие эффективности охватывает не только снижение себестоимости продукции и сокращение сроков проектирования и производства, но и обеспечение удобства освоения и снижения затрат на будущую эксплуатацию изделий [127].

Достижение поставленных целей на современных предприятиях, выпускающих сложные промышленные изделия, оказывается невозможным без широкого использования АС, основанных на применении современных ИТ и предназначенных для ввода, обработки и использования всей необходимой информации о свойствах изделий и сопровождающих их процессов. Специфика задач, решаемых на различных этапах ЖЦ изделий, обуславливает разнообразие применяемых АС и САПР (см. таблицу П. 1.1) [68, 127, 161, 188, 195].

Таблица П. 1.1. Назначение АС на различных этапах ЖЦ СОТО

Этап ЖЦ	Тип автоматизированной системы	Назначение
Проектирование	CAE (Computer Aided Engineering)	Функциональная САПР.
	CAD (Computer Aided Design)	Конструкторская САПР.
	CAM (Computer Aided Manufacturing)	САПР технологических процессов.
	PDM (Product Data Management)	Система управления проектными данными. Выполняет функции координации работы систем CAE/CAD/CAM в части управления проектными данными.

Производство	SCM (Supply Chain Management)		Система управления цепочками поставок, иногда называемая системой Component Supplier Management (CSM). На этапе производства эта система управляет поставками необходимых материалов и комплектующих.	
	АСУ П (Автоматизированная Система Управления Предприятием)	ERP (Enterprise Resource Planning)	Система планирования и управления предприятием. Наиболее развитые системы ERP выполняют различные бизнес-функции, связанные с планированием производства, закупками, сбытом продукции, анализом перспектив маркетинга, управлением финансами, персоналом, складским хозяйством, учетом основных фондов и т. п.	
		MRP-2 (Manufacturing Requirement Planning)	Система планирования производства и требований к материалам. Ориентирована, главным образом, на бизнес-функции, непосредственно связанные с производством.	
		MES (Manufacturing Execution System)	Производственная исполнительная система. Ориентирована на решение оперативных задач управления проектированием, производством и маркетингом.	
	АСУ ТП (Автоматизированная Система Управления Технологическими Процессами)	SCADA (Supervisory Control and Data Acquisition)	Выполняет диспетчерские функции (сбор и обработку данных о состоянии оборудования и технологических процессов), также используется для разработки ПО для встроенного оборудования.	
		CNC (Computer Numerical Control)	Выполняет непосредственное программное управление технологическим оборудованием на базе контроллеров (специализированных компьютеров, называемых промышленными), которые встроены в технологическое оборудование.	
Реализация и эксплуатация	CRM (Customer Requirement Management)		Выполняет функции управления отношениями с заказчиками и покупателями, используется для анализа рыночной ситуации, определения перспектив спроса на планируемые изделия.	
	S&SM (Sales and Service Management)		Выполняет маркетинговые задачи, а также используется для решения проблем обслуживания изделий.	
На всех этапах	E-Commerce (системы электронного бизнеса)	CPC (Collaborative Product Commerce)	Система управления данными в ИИС.	Объединяют в ЕИП запросы заказчиков и данные о возможностях организаций, специализирующихся на предоставлении различных услуг и выполнении процедур и операций по проектированию, изготовлению, поставкам заказанных изделий. Обеспечивают взаимодействие предприятий, т. е. формируют ИИС для остальных систем.
		PLM (Product Lifecycle Management)	Система управления ЖЦ изделий.	
		OLAP (OnLine Analytical Processing)	Система аналитических вычислений в режиме РВ.	

Таким образом, для каждого этапа ЖЦ есть соответствующая АС. Взаимосвязь этих систем между собой и по отношению к этапам ЖЦ СОТО проиллюстрирована на рис. П. 1.1[127].

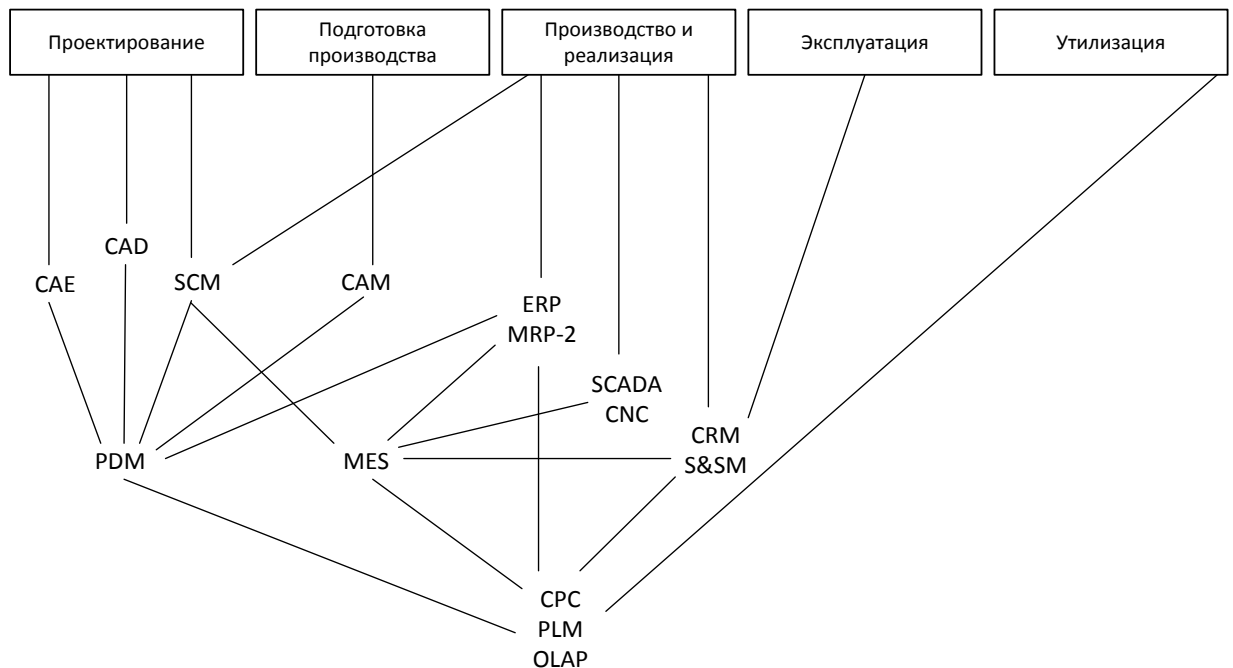


Рис. П. 1.1. Использование АС на различных этапах ЖЦ изделия

Из рис. П. 1.1 и таблицы П. 1.1 нетрудно заметить, что решающую роль в интеграции данных АС, формировании ЕИП и обеспечении взаимодействия организаций с целью повышения их эффективности играют системы класса CPC, PLM, OLAP. На сегодняшний день полноценное автоматизированное управление предприятиями в рамках PLM и CPC представляется трудно реализуемым. Однако, как будет показано далее, достижима автоматизация в рамках соответствующей *OLAP-системы* в соответствии с рассматриваемой в работе задачей по оцениванию ТС и надежности СОТО на всех этапах его ЖЦ.

Стоит отметить еще один немаловажный аспект концепции CALS. В ИИС протекают информационные процессы, сопровождающие и поддерживающие ЖЦ изделия на всех его этапах. Здесь реализуется главный принцип ИПИ: информация, однажды возникшая на каком-либо этапе ЖЦ, сохраняется

в ИИС и становится доступной всем участникам этого и других этапов (в соответствии с имеющимися у них правами пользования этой информацией) [188]. Это позволяет избежать дублирования, перекодировки и несанкционированных изменений данных, избежать связанных с этими процедурами ошибок и сократить затраты труда, времени и финансовых ресурсов.

Развитие CALS-технологий должно привести к появлению так называемых виртуальных предприятий (ВП) электронного бизнеса. ВП – новая организационная форма выполнения масштабных наукоемких проектов, связанных с разработкой, производством и эксплуатацией сложной продукции [188]. ВП создается посредством объединения на контрактной основе предприятий и организаций, участвующих в ЖЦ продукции и связанных общими бизнес-процессами. Информационное взаимодействие участников ВП осуществляется на основе общих ХД через общую корпоративную или глобальную компьютерную сеть. Срок жизни ВП определяется длительностью проекта или ЖЦ продукции. Для ВП проблема информационного взаимодействия и применения CALS-технологий особенно актуальна.

Важным понятием концепции ИПИ является принцип параллельного инжиниринга (ПИ) – Concurrent Engineering [9, 68], означающий выполнение процессов разработки и проектирования одновременно с моделированием процессов изготовления и эксплуатации. Сюда же относится одновременное проектирование различных компонентов сложного изделия. При ПИ многие проблемы, которые могут возникнуть на более поздних стадиях ЖЦ, выявляются и решаются на стадии проектирования. Такой подход позволяет улучшить качество изделия, сократить затраты и время его вывода на рынок. Отличиями ПИ от традиционного подхода является параллельная работа с изделием на основе информационного обмена, итеративность процесса приближения к необходимому результату.

Таким образом, ИПИ предполагает повышение эффективности предприятий на основе таких ключевых принципов, как создание ВП, обеспечение оперативного информационного обмена, формирование ИИС разнообразных АС,

использование интегрирующей OLAP-системы, сохранность результатов обработки информации для ЛПП, реализация принципа ПИ.

1.5. Задачи информационной поддержки изделий

1.5.1. Управление проектом (Project Management)

Определение П. 1.2. Под *проектом* в данном контексте понимается уникальный процесс, состоящий из совокупности скоординированных и управляемых видов деятельности с начальной и конечной датами, предпринимаемый для достижения цели, соответствующей конкретным требованиям, и включающий ограничения по срокам, стоимости и ресурсам [239, серия ISO 9000].

В результате документального оформления целей руководитель проекта (по терминологии исследования операций, ЛПП) должен собрать всю информацию о доступных ресурсах (материальных, финансовых, людских, временных и т.д.), факторах и среде, влияющих на проект, и т.д.

По мере продвижения проекта по этапам ЖЦ формируется информация, которая характеризует ход реализации проекта (соблюдение или нарушение сроков, расходование и текущие потребности в ресурсах и т.д.), а также расстояние (в некоторой выбранной метрике) до поставленной цели (обратная связь). На основе этой информации ЛПП вырабатывает корректирующие решения.

1.5.2. Управление данными об изделии (Product Data Management)

Управление данными об изделии (УДИ) является системообразующим элементом ИИС, поскольку снабжает информацией все этапы ЖЦ. Реализуется посредством PDM-систем. Цель УДИ – обеспечение полноты, целостности и актуальности информации об изделии в любой момент времени и доступность ее всем участникам ЖЦ.

К числу основных функций PDM можно отнести:

- управление структурой и составом изделия;
- управление технологическими данными об изделии;
- управление ролями исполнителей;

- управление вспомогательными данными;
- регистрация статусов документов и их изменений.

Согласно ISO 10303 система PDM строится на основе стандартизованной объектной модели данных и оперирует следующими основными понятиями [241]: версия, конфигурация, экземпляр изделия, структура изделия, контекст представления данных (конструкторский, технологический, эксплуатационный и т.д.), электронный технический (конструкторский, технологический, эксплуатационный) документ, состояние (статус) документа/структуры/свойства/процесса/ресурса, электронно-цифровая подпись, поток работ, ресурс, свойство (характеристика), единица измерения, категория.

1.5.3. Управление конфигурацией изделия (Configuration Management)

Определение П. 1.3. Конфигурация – взаимосвязанные функциональные и физические характеристики продукции или услуги, установленные в информации о конфигурации продукции [239, серия ISO 9000].

Конфигурация зачастую представляется в виде некоторого древовидного или сетевого графа и предполагает, что атрибуты, описывающие компоненты, обладают определенными значениями [188]. С вершинами графа могут быть связаны бизнес-процессы, документы и свойства (характеристики) объектов (изделий, процессов, ресурсов, документов). Структура изделия определяется отношениями входимости, установленными между его составными частями. Входимость – понятие, характеризующее использование составных частей изделия в составе конечного изделия или его составных частей [68]. Управление конфигурацией предполагает наличие правил и процедур её формирования, изменения и документирования. Пример управления конфигурацией представлен на рис. П. 1.2 [161]. Конфигурация создается для хранения и отслеживания значений связанных атрибутов, характеризующих состояние узлов изделия, с целью оценивания степени соответствия заданным показателям качества. Формируется на этапе конструкторской подготовки производства, имеющей целью создание комплекта конструкторских документов (про-

екта изделия). В ходе этих процессов порождается и помещается в ИИС множество информационных объектов, содержащих данные о структуре, составе и всех компонентах изделия.

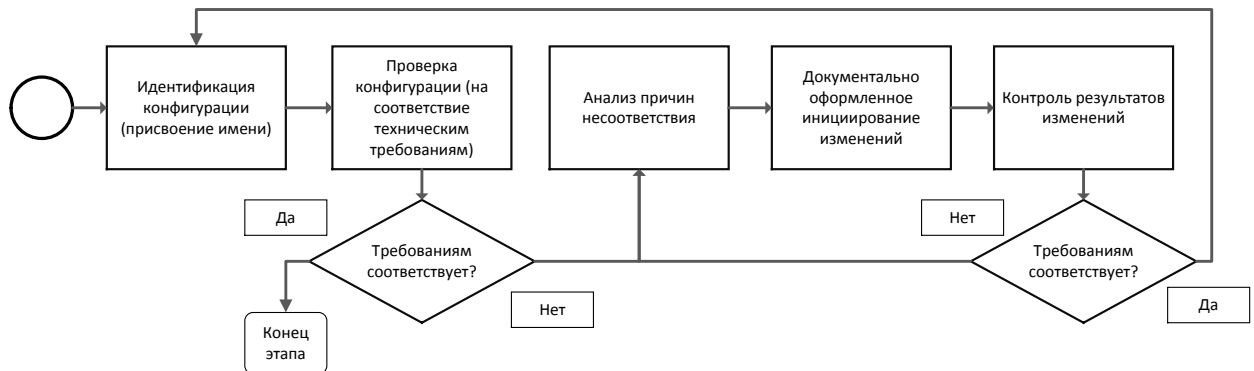


Рис. П. 1.2. Схема алгоритма управления конфигурацией COTO

Относительно понятия конфигурации выделяют несколько её представлений в зависимости от этапа ЖЦ [68]:

- базовая конфигурация (утвержденный тип изделия);
- проектная конфигурация (набор проектных расчетных характеристик разрабатываемого изделия);
- функциональная конфигурация (набор требований, предъявляемых заказчиком к изделию, сгруппированный в соответствии с его функциональной структурой);
- физическая конфигурация (исполнение изделия, набор фактических (измеренных) характеристик экземпляра изделия).

1.5.4. Управление ИИС, в том числе информационными потоками (Information Management)

ИИС характеризуется определенной степенью распределенности между вовлеченными в ЖЦ организациями и используемыми ими ИС, в отличие от традиционного использования БД. Эта особенность требует создания специальной инфраструктуры, обеспечивающей накопление, хранение и передачу данных между всеми заинтересованными участниками ЖЦ изделия. Отметим, что состояние телекоммуникационных средств и систем позволяет говорить о

том, что основой инфраструктуры ВП, а также предприятий с географически распределенной структурой может служить глобальная сеть Интернет, в которой данные передаются с помощью протокола TCP/IP (Transmission Control Protocol / Internet Protocol). В общем случае пример организации ИИС представлен на рис. П. 1.3.

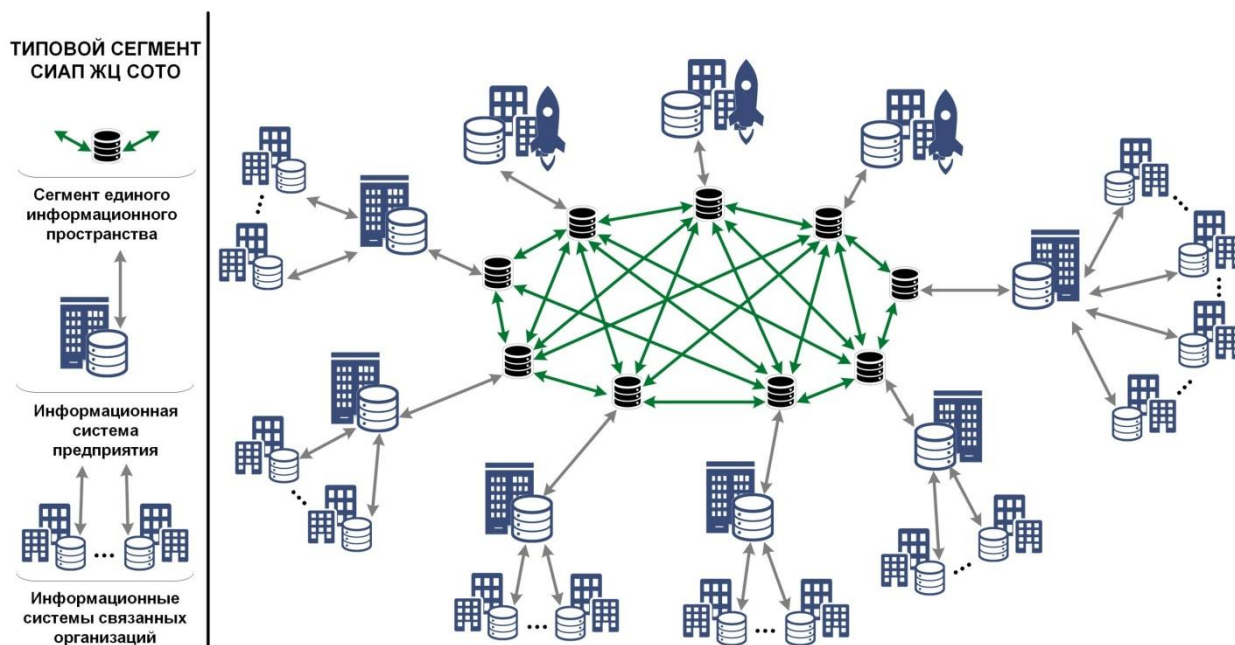


Рис. П. 1.3. Пример организации единого информационного пространства организаций – участников ЖЦ СОТО.

Логическая архитектура и использование ИИС для задач на всех этапах ЖЦ в общем виде представлен на рис. П. 1.4, взятом из работы [68]. На рисунке представлена структура ИИС во взаимодействии с процессами ЖЦ продукции предприятия. Из схемы видно, что в этих процессах используется информация, содержащаяся в ИИС, а информационные объекты, порождаемые в ходе процессов, возвращаются в ИИС для хранения и последующего использования в других процессах. Это отображено на схеме двойными стрелками.

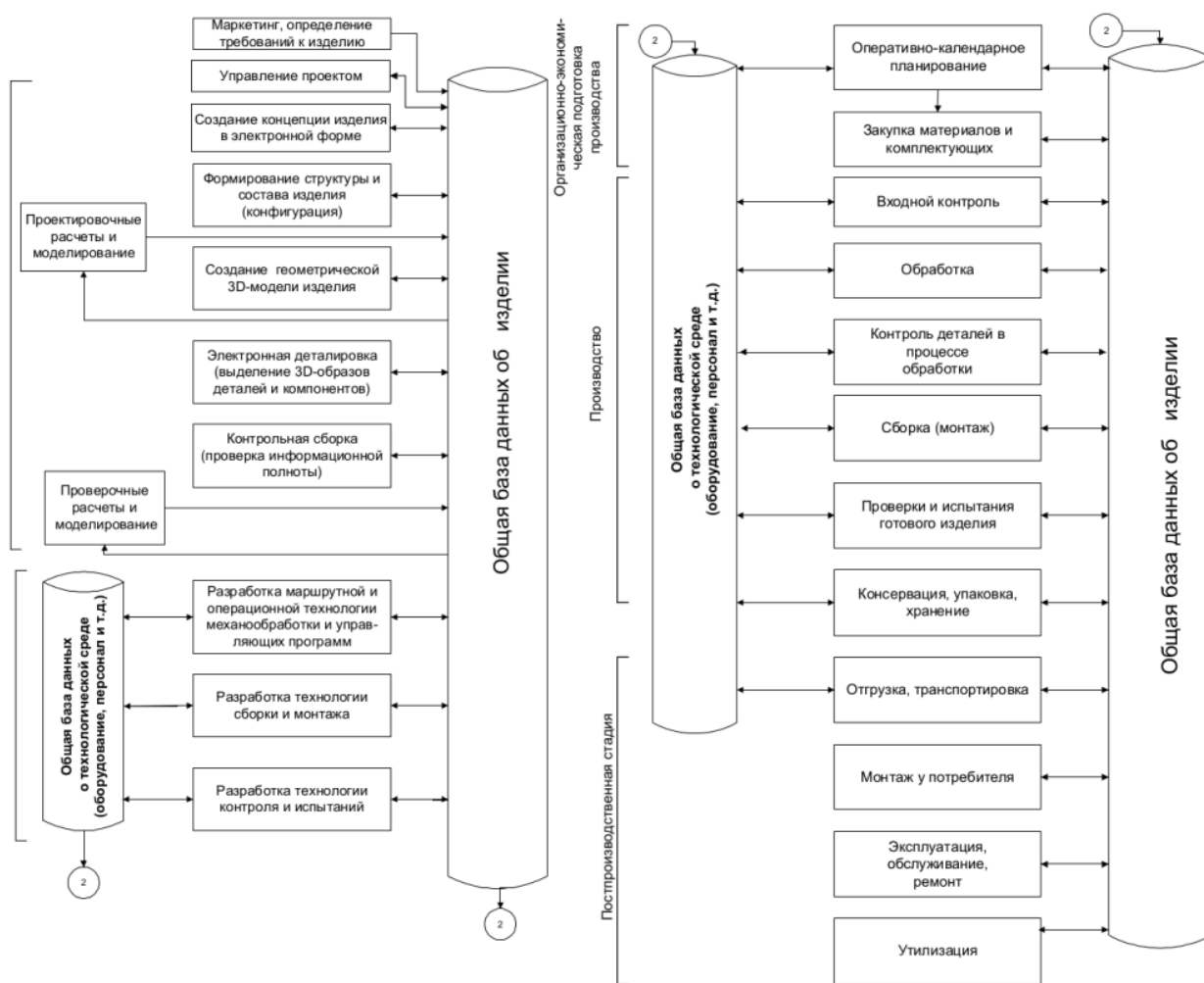


Рис. П. 1.4. Использование ИИС для задач на различных этапах жизненного цикла СТО

1.5.5. Управление качеством (Quality Management)

Современные предприятия вкладывают существенные ресурсы в повышение качества производимой продукции (см. определение П. 1.4). Эта ситуация привела к появлению ряда системных подходов к управлению качеством, одним из которых является концепция «всеобщего управления качеством» (TQM – Total Quality Management) [68]. Основной принцип TQM заключен в положении, что качество продукции напрямую зависит от качества организации процессов ЖЦ, а также их взаимодействия.

Определение П. 1.4. Качество продукции – степень соответствия характеристик, присущих продукции, требованиям потребителя [239, ГОСТ Р ИСО 9000 - 2001].

В соответствии с международным стандартом ИСО 9000:2000, управление качеством должно базироваться на использовании ИС, поддерживающих автоматизированную обработку данных и документирование процессов обеспечения качества на всех стадиях ЖЦ изделия [68, 239].

С точки зрения управления качеством характеристики изделия, выражаемые через атрибуты конфигурации, а также технологические и бизнес-процессы должны удовлетворять техническим требованиям и техническим условиям, при этом проверка соответствия может быть автоматизирована.

В соответствии с современными стандартами к количественным показателям качества относят 13 свойств промышленной продукции: функциональная пригодность, надежность, безотказность, долговечность, ремонтпригодность, сохраняемость, эргономичность, эстетичность, технологичность в производстве, технологичность при применении, ресурсопотребление, безопасность, экологичность [68]. Надежность может рассматриваться, как свойство изделия сохранять свое качество (в определенных пределах) во времени.

Для обеспечения полноты оценивания качества и надежности изделий в литературе [217] предлагается к рассмотрению система управления качеством, включающая несколько компонент (см. рис. П. 1.5).

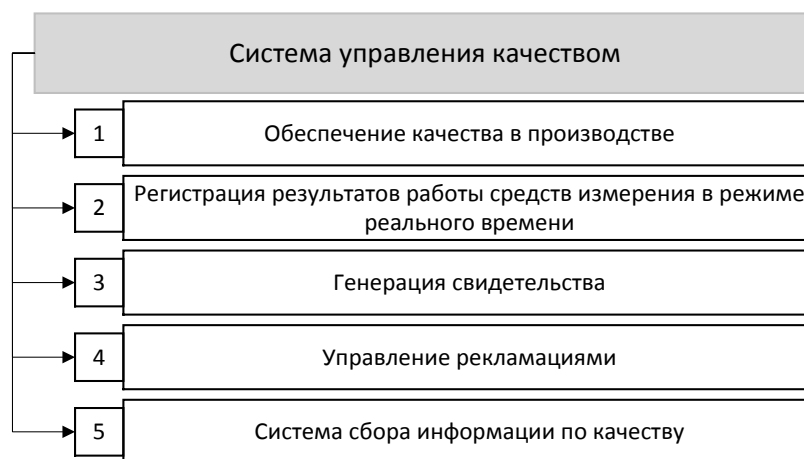


Рис. П. 1.5. Компоненты системы управления качеством

При более детальном рассмотрении компонент системы управления качеством можно определить следующее их назначение.

1. Обеспечение качества в производстве.

Обеспечение качества в производстве имеет особое значение, поскольку позволяет уменьшить затраты за счет уменьшения процента брака. Предполагается, что реализация принципа ПИ на основе быстрого обмена информацией между участниками процесса, а также обеспечения её достоверности за счет устранения человеческого фактора позволит существенно повысить качество производимых изделий (см. раздел 1.2.1.2). Поддержка этого процесса может осуществляться на основе специализированных электронных регистрационных карт документов. Конечный контроль качества продукции может осуществляться на основе анализа информации, представленной в таких картах.

2. Регистрация результатов работы средств измерения в режиме РВ.

Для оптимизации процесса проверки качества изделий электронные средства измерения могут регистрировать данные в режиме РВ при условии наличия соответствующего режима работы в ИС.

3. Генерация свидетельства.

Отслеживание данных по качеству начинается у поставщика, продолжается в производстве и заканчивается у потребителя (заказчика). ИС должна поддерживать оценивание качества на протяжении всего ЖЦ и формировать по требованию электронный документ о показателях качества – свидетельство. Иными словами, такое свидетельство можно проинтерпретировать как электронный паспорт изделия (см. определение 1.3).

4. Управление рекламациями.

Одним из основных производственных и эксплуатационных документов для регистрации несоответствий качества изделий является рекламационный акт. В задаче управления рекламациями первым шагом является сбор рекламационной информации с целью обеспечения ЛПР достоверной информации о снижении показателей качества изделий от заданных требований. Одним из вариантов такого сбора является использование регистрационных карт, о которых говорилось выше.

5. Система сбора информации по качеству.

Эффективность «предупреждающих» мероприятий, обеспечивающих качество продукции, базируется на степени достоверности информации, получаемой со всех этапов ЖЦ. При этом важно такую информацию не только собрать, но и объединить в едином процессе её анализа с целью своевременного предоставления ЛПР обобщенной информации о состоянии изделия. Система сбора и анализа информации, таким образом, помогает улучшить качество продукции, снизить количество возникающих дефектов и рекламаций, уменьшить затраты на производство изделия.

1.5.6. Интегрированная логистическая поддержка постпроизводственных этапов ЖЦ

В области ИПИИ-технологий под логистикой понимают материальное и информационное обеспечение постпроизводственных стадий ЖЦ изделия. Интегрированная логистическая поддержка (ILS – Integrated Logistic Support) – это комплекс методов и средств, направленных на сокращение затрат на поддержку ЖЦ. К основным из них относятся [217]:

- процедуры поддержки материально-технического обеспечения;
- планирование процессов технического обслуживания и ремонта изделия;
- разработка и обеспечение обслуживающего персонала интерактивными электронными техническими руководствами по эксплуатации и ремонту изделия.

1.5.7. Информационная поддержка обеспечения надежности изделий

Существует два основных пути обеспечения надежности изделия. Первый предполагает задание технических характеристик «с запасом», что ведет к увеличению материалоемкости, ухудшению эксплуатационных характеристик, повышению стоимости. Другой путь – моделирование состояния изделия и расчет значений (оценивание) надежностных параметров с помощью специализированных программных средств. На сегодняшний день существует несколько основных подходов к оцениванию надежности изделий. Наиболее популярные из них [217]:

- методология анализа видов и последствий отказов FMEA/FMECA (Failure Modes and Effects Analysis), (пример реализующей программы – RAM Commander). Производится оценивание надежности на основе интенсивности отказов на N часов по методу Монте-Карло. Предполагает построения дерева отказов для каждого из узлов и агрегатов изделия.
- системы сбора и анализа информации об отказах и определения необходимых корректирующих действий FRACAS (Failure Reporting and Corrective Actions System) (пример реализующей программы – FRACAS FavoWeb). Эта система накапливает знания по отказам/дефектам, их анализу и корректирующим действиям, чтобы оценивать прогресс в устранении причин отказов/дефектов, а также связанных с ними процессов. Непрерывный мониторинг данных через данную систему позволяет оценить, были ли предшествующие отказы и дефекты устранены через корректирующие действия. Позволяет своевременно реагировать на отказы.

1.5.8. Управление потоками работ (Workflow Management)

Понятие «поток работ» (англ., Workflow) используется при описании подходов к формализации и управлению бизнес-процессами предприятия, а также программных средств, которые их реализуют.

Определение II. 1.5. По определению ассоциации Workflow Management Coalition (WfMC), *бизнес-процесс* – это одна или более связанных между собой процедур или операций (функций), которые совместно реализуют некую бизнес-задачу или политическую цель предприятия, как правило, в рамках организационной структуры, описывающей функциональные роли и отношения [188].

Бизнес-процесс представляет собой поток заданий и функций, соотносимых зачастую именно с организационными аспектами функционирования предприятий, а также правила, управляющие последовательностью выполнения этих функций. Технология Workflow призвана это автоматизировать.

Термин «поток работ» употребляется применительно к автоматизации бизнес-процессов, при которой документы, информация или задания передаются в определенном порядке для выполнения необходимых действий от одного участника к другому в соответствии с набором процедурных правил. Система управления потоком работ описывает этот поток, реализуя таким образом процессный подход к управлению, который концентрирует внимание на правилах взаимодействия участников процесса. Такой акцент при автоматизации делается в связи с тем, что практика предприятий показывает, что наибольшие потери эффективности и снижение качества изделий происходит вследствие отсутствия формализации организационных процессов и высокой доли в них человеческого фактора.

1.5.9. Управление изменениями производственных и организационных структур (Change Management)

Объединяет в себе другие задачи в рамках CALS-технологий в том смысле, что предполагает системный подход к изменению технологических и бизнес-процессов, состава документации и её содержания, порядка взаимодействия смежных организаций и систем. С позиций ИТ эти аспекты предполагают возможность оперативного изменения в CALS-системе информационных моделей, моделей данных, моделей потоков работ и данных, на основе которых она функционирует.

1.6. Некоторые примеры внедрения CALS-технологий

В работе [188] приводятся некоторые примеры компаний, успешно внедривших CALS-технологии в свои информационные процессы. В таблице П. 1.2 для рассматриваемых компаний представлены области применения CALS и используемые методы при моделировании.

Таблица П. 1.2. Примеры внедрения CALS-технологий и используемых при этом методов моделирования

ПрО	Примеры компаний	Область применения	Используемые методы, технологии и стандарты реализации CALS
Авиационная промышленность	Airbus	Системы снабжения	
	AlliedSignal	Управление качеством	

	American Airlines	Эксплуатация самолетов	<ul style="list-style-type: none"> – Проектирование информационных моделей на основе языка Express. – подход SADT (Structured Analysis & Design Technique) на основе использования стандарта IDEF0 моделирования процессов. – Использование стандарта STEP, регламентирующего правила информационного обмена. – Использование языка XML для структурированного описания документов. – Использование UML для моделирования схем БД, процессов. – Реализация потоков работ на основе XPDL (Process Definition Language) консорциума WorkFlow Management Coalition (WMC). – Реализация потоков работ на основе BPMN (Business Process Model and Notation). – ISO 15926 OIL & GAS, ориентированный на разделение доступа к данным. – ISO 15926 интеграция ресурсов посредством языка Express. – ISO 10303 стандарт STEP. – ISO 9000:2000 Системы управления качеством. – ISO 8879. Стандарт создания, управления и распространения научной и технической информации в электронном виде. – ISO 13584 Parts Library (PLIB) - серия международных стандартов для представления и обмена доступными для компьютерной интерпретации данными о поставляемых компонентах и комплектующих изделиях (узлах, деталях). – FIPS 183 регламентирует порядок и правила функционального моделирования бизнес-процессов на основе методологии IDEF0. – ГОСТ Р ИСО 10303 – серия российских ГОСТов о языке Express. – ГОСТ Р 5 ***.01-02. Техническая документация в электронном виде. – Рекомендация по стандартизации (PC) Р 50. 1. 028-2001. Язык моделирования IDEF0.
	Bell Helicopter Textron	Поддержка обслуживания продукции	
	Boeing	Виртуальное предприятие	
	General Motors	Автоматизация офиса и завода	
	Hughes Aircraft	УДИ ВП	
	Lockheed Aeronautical	Ускорение закупок	
	Lockheed Martin	Разработка документации	
	McDonnell Douglas	Программа C-17	
	Northrop Grumman	Документация по бомбардировщикам	
	Pratt & Whitney	Обработка больших данных	
	Raytheon	Программа «Patriot»	
	Rockwell International	Документация по бомбардировщикам	
	Rolls Royce	Параллельная разработка	
	United Airlines	Обслуживание самолетов	
Автомобильная промышленность	John Deere	Интеграция предприятия	
	Land Rover	Параллельная разработка	
Телекоммуникации	AT&T	Защита данных пользователей сети	
	GTE Government Systems	Управление данными систем беспроводной связи	
Энергетика	Electric Power Research Institute	Электронная коммерция	
	Tokyo Electric Power	Интеграция предприятий	
Проектирование и строительство	Bechtel	Документация для инженерных войск	
	Sverdrup	Развитие системы San Francisco Bay Area Rapid Transit	
Государственные организации, космическая и военная отрасль	General Dynamics	взаимодействие систем беспроводной связи	
	НАСА	Центр управления космическими полетами, космические аппараты	
	ВМФ США	Центр слежения за военными кораблями, 3D-моделирование эсминцев	

Примеры реализации в России	Воронежский механический завод	Интеграция разработки и производства техники для космической, железнодорожной, авиационной отраслей
	НПП «Аэро-сила»	Интеграция разработки и производства авиационных двигателей, воздушных винтов и винтовых преобразователей
	АНТК "Туполев"	Интеграция разработки и производства авиационной техники
	Комсомольское-на-Амуре Авиационное Производственное Объединение	Создание интерактивного электронного технического руководства изделия
	Тульское констр. бюро приборостр-я Рособоронэкспорт	Создание интерактивного электронного технического руководства изделия

1.7. Стандарты информационной поддержки жизненного цикла COTO

К настоящему моменту для решения рассмотренных ранее задач CALS-технологий разработано немало стандартов, в которых осуществляются попытки формализации их решения, разработки методик, регламентов, языков описания соответствующих моделей. Получившие наибольшее распространение в России и за рубежом представлены в таблице П. 1.3 [20, 68, 188, 127, 248].

Таблица П. 1.3. Стандарты описания технологий информационной поддержки жизненного цикла изделий

Информационные модели		Стандарт представления информации	Содержание стандарта
Модель ЖЦ продукта и выполняемых в его ходе бизнес-процессов		IDEF – Integrated definition	Функциональное моделирование ЖЦ и выполняемых бизнес-процессов
		ISO 10303 AP208	
Модель изделия	Конструкторская	ISO 10303 (STEP) PC P 50. *. 031-2001 ГОСТ Р ИСО 10303	Структура, конфигурация, геометрия изделия, другие информационные модели
	Производственная	ISO 13584 (PLIB)	Формат данных о библиотеках деталей у поставщиков
		ISO 9000:2000	Управлением качеством
		MIL-STD-1388-1/2 Logistic Support Analysis (LSA) Record	Формат данных о процессах материально-технического снабжения

	Эксплуатационная	<u>MIL-M-87268</u> Manuals Interactive Electronic Technical General Content, Style, Format and User-Interaction Requirements (IETM)	Требования к электронным руководствам: содержание, стиль, формат, интерфейс с пользователем
Форма представления данных (цифровизация)	Эксплуатационная	<u>MIL-D-87269</u> Data Base Revisable Interactive Electronic Technical Manuals <u>ISO 8879</u>	Требования к электронным руководствам: содержание, стиль, формат, интерфейс с пользователем
		<u>ISO 8879</u> (SGML) – Standard Generalized Markup Language	Требования к оформлению БД и электронных справочников по изделиям
		<u>MIL-PRF-28001C</u> Markup Requirements and Generic Style Specification for Electronic Printed Output and Exchange of Text	Требования к оформлению электронных документов (рекомендации по применению SGML для электронных документов)
		<u>MIL-PRF-28002C</u> Requirements for Raster Graphics Representations in Binary Format	Требования к оформлению растровых изображений в двоичном формате в электронной документации
		<u>MIL-PRF-28003</u> Color Graphics Metafile (CGM)	Требования к представлению иллюстраций для технической документации в электронном виде
		<u>ISO 10744</u> Hytime Hypermedia/Time Based Structuring Language	Требования к мультимедийной информации в электронных документах
			<u>ISO 7372-86</u> «Trade data interchange. Trade data elements directory. First edition. 1986-07-01» <u>ISO 9735-88</u> «EDI for administration, commerce and transport (EDIFACT). Syntax rules. 1988-07-15» <u>ГОСТ 6.20.2-91</u> <u>ГОСТ 6.20.1.90</u>
	Модель среды	<u>ISO 15531</u> (MANDATE)	Форма представления и методы использования информации о производстве и используемых производственных ресурсах, их характеристиках и ограничениях с точки зрения управления производством

Стоит отметить, что широкое распространение при реализации CALS-технологий получила группа стандартов STEP (Standard for The Exchange of Product model data), собравший воедино многие аспекты, характерные для CALS-систем, специальный язык описания информационных моделей EXPRESS, описания интерфейсов обмена данными.

1.7.1. Стандарт STEP в системах информационной поддержки жизненного цикла изделий

Одним из наиболее распространенных форматов обмена данными, применяющихся в рамках CALS-технологий, описан в стандарте STEP [53, 98, 127, 212]. Информация об изделии, формируемая при его проектировании,

производстве, эксплуатации, техническом обслуживании и утилизации, используемая для решения различных задач в течение ЖЦ изделия во многих вычислительных системах, должна быть представлена в едином машинно-ориентированном формате, от которого требуется сохранение полноты и совместимости информации при обмене между различными вычислительными системами. Целью стандарта является создание механизма, позволяющего описывать и хранить данные об изделии на протяжении всего ЖЦ изделия независимо от конкретной системы на основе свойства интероперабельности. В контексте стандарта под *интероперабельностью* понимают совместимость по данным в пределах пространства STEP.

Стандарт STEP регламентирует на основе создания специализированных прикладных протоколов (или использования готовых концептуальных моделей ПрО): логическую структуру БД, совокупность информационных объектов, хранимых в БД, их связи и атрибуты, а также протоколы обмена данными. Типовые информационные объекты, такие как данные о составе изделия, материалах, геометрических изделиях, независимые от характера описания изделия, называются в стандарте «интегрированными ресурсами», на основе которых строятся схемы БД об изделии для разных ПрО автомобилестроения, судостроения, аэрокосмической промышленности и т.д. Типовая концептуальная модель представлена на рис. П. 1.6[161].

Таким образом, стандарт STEP определяет правила создания единой информационной модели ПрО, готовую модель данных, описание которых базируется на использовании объектно-ориентированного декларативного языка EXPRESS для описания иерархий понятий. При этом стоит отметить, что стандартом STEP не определяется программная реализация сформированной на языке EXPRESS модели, и вводится требование, чтобы ЭВМ «владела» понятиями информационной модели. В качестве визуальной нотации для языка EXPRESS может быть использована IDEF1X или EXPRESS-G (см. рис. П. 1.7).

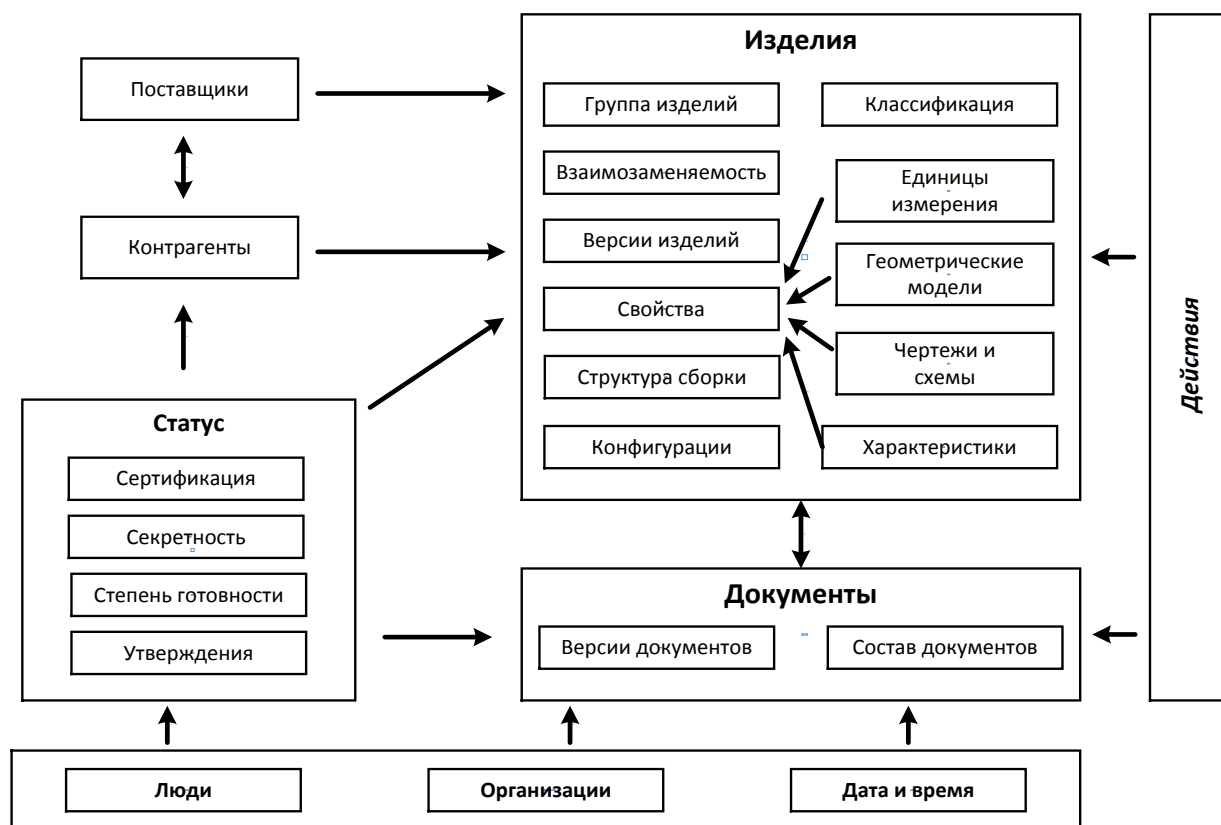


Рис. П. 1.6. Концептуальная схема информационных ресурсов в стандарте STEP

- объекты (<i>ENTITY</i>);
- типы (<i>Type</i>);
- константы (<i>Constant</i>);
- правила (<i>Rule</i>);
- функции (<i>Function</i>) и процедуры (<i>Procedure</i>), необходимые для проверки правил и для вычисления значений производных атрибутов.

Рис. П. 1.7. Схема типовой сущности, описываемой языком EXPRESS

1.7.2. Язык специфицирования EXPRESS

EXPRESS – это название формального языка спецификации информационных требований к моделям изделия и концептуальных схем данных [53, 127, 241, 212]. Язык EXPRESS применяется для определения информационных требований других стандартов комплекса ИСО 10303. В языке EXPRESS объекты, называемые концептами, определяются через их атрибуты, характеризующие признаки или характеристики объектов. Такая постановка определяет

возможность назвать формируемые на EXPRESS модели онтологиями (см. приложение 2.3.5). Атрибуты могут быть представлены простыми данными (например, целым числом) или другими объектами.

В данном языке EXPRESS-схема (концептуальная схема данных) является основой для структурирования и описания совокупности условий, устанавливающих область ее определения. На основе таких схем может производиться обмен данными между АС при условии, что они поддерживают стандарт STEP. Также заложенный в стандарте подход предполагает генерацию объектно-ориентированных БД на основе схем, куда заносятся данные об изделии, и из которых можно получить доступ к этим данным.

Необходимость программной реализации для языка EXPRESS определила необходимость формализации синтаксиса языка. В качестве формализма в стандарте STEP выбрана в соответствии с классификацией Н. Хомского контекстно-свободная формальная грамматика в нотации Вирта (Wirth Syntax Notation). Эта нотация обеспечивает однозначное представление слов языка. Грамматика определяется как четверка (V_t, V_n, P, S) , где V_t – алфавит, символы которого называются терминальными (терминалами), из них строятся цепочки, порождаемые грамматикой. V_n – алфавит, символы которого называются нетерминальными (нетерминалами), используются при построении цепочек; они могут входить в промежуточные цепочки, но не должны входить в результат порождения. V_t и V_n не имеют общих символов. Полный алфавит (словарь) грамматики V определяется как объединение V_t и V_n . P – набор порождающих правил (выводов), каждый элемент которых состоит из пары (a, b) , где a находится в V^+ , а b в V^* , т.е. цепочки, построенные из символов алфавита V . Правило записывается $a = b$. S принадлежит V_n и называется начальным символом (аксиомой). Этот символ – отправная точка в получении любого предложения языка.

Стандарт STEP предполагает наличие расширяемого набора прикладных протоколов, содержащих стандартизованные концептуальные схемы различ-

ных ПрО, описанных на языке EXPRESS. Создание прикладного ПО, связанного с CALS-технологиями, предполагает внедрение соответствующего прикладного протокола, на основе которого система будет функционировать. Можно отметить, что в общем случае такое ПО будет основано на конечных автоматах с магазинной памятью вследствие использования формальных грамматик [212]. Каждый прикладной протокол должен включать в себя три типа моделей (см. таблицу П. 1.4) [212]:

- функциональную модель ПрО (ААМ);
- концептуальную информационную модель ПрО (АРМ);
- модель интерпретации ресурсов STEP (АИМ).

Таблица П. 1.4. Состав моделей прикладного протокола STEP

Обозн.	Наименование	Содержание	Средства представления
АММ	Application Activity Model Функциональная модель ПрО	Структура функций, которые создают и используют информацию о производстве изделия в контексте ПрО, охватываемой данным протоколом. Функциональная модель строится в предположении, что в данной ПрО уже достигнут требуемый уровень компьютерной автоматизации.	IDEF0
АРМ	Application Reference Model Справочная модель ПрО	Концептуальная информационная модель (концептуальная схема данных), описывающая структуру информационных понятий ПрО и их связи, обеспечивающие реализацию ААМ. АРМ строится полностью в терминалах ПрО.	EXPRESS, ранее использовались; IDEF1X, диаграммы Чена, NIAM
АИМ	Application Interpreted Model Интерпретированная модель ПрО	Концептуальная схема данных, идентичная АРМ, но выраженная с помощью сущностей, имеющих в информационных ресурсах STEP. Процесс выражения содержимого АРМ, заключающийся в подборе и адаптации содержащихся в информационных ресурсах сущностей называется интерпретацией.	EXPRESS

Для реализации возможности обмена данными между АС используются SDAI (Standard Data Access Interface) – схемы, т.е. программные интерфейсы к источникам данных прикладных систем (в том числе к библиотекам моделей CAE/CAD/CAM -систем) с переводом моделей в STEP-файлы, которые используются в STEP-средах для организации обмена между приложениями через общую БД STEP.

1.7.3. Анализ возможности использования STEP в решении задачи информационно-аналитической поддержки

Стоит отметить, что в конечном счете модели, создаваемые посредством языка STEP, по существу, являются концептуальными моделями ПрО, в которых к единому виду приведены все подсистемы ПрО. Это действительно обеспечивает единое представление, благодаря которому информация о состоянии изделия может быть интегрирована в единое хранилище для всех этапов его ЖЦ.

Существует несколько успешных примеров внедрения CALS- технологий на предприятиях на основе стандарта STEP (см. приложение 1.6). Проведенный анализ показал, что большей частью CALS используется в крупных иностранных компаниях, а отечественных – единицы. К числу примеров внедрения в отечественных компаниях можно, например, привести [29, 38]. Эта ситуация может быть объяснена тем, что предлагаемый в STEP подход требует изменения существующей на предприятиях инфраструктуры АС, что требует существенных ресурсов и длительного срока внедрения (в среднем, 3-5 лет). Так, например, такие известные зарубежные системы, реализующие CALS, как Teamcenter (Siemens PLM Software), Agile PLM (Oracle), Dassault Systemes (и ее подсистемы CATIA, ENOVIA-VPLM, ENOVIA-SmarTeam, DELMIA) не предполагают интеграцию с существующими на предприятиях АС, а полностью заменяют собой все автоматизируемые задачи CALS целым комплексом программных продуктов [64, 217]. Кроме того, в целом, систем, полноценно реализующих концепцию CALS, практически нет. Распространены ИС, выполняющие частные задачи. К числу таких ИС можно отнести 1С: Интегратор, 1С: PDM Управление инженерными данными, Lotsia PDM Plus, T-FLEX DOCs, TechnologiCS, Windchill, SmarTeam, IDPM CADISON PDM, SolidWorks Enterprise PDM.

В то же время методика, предлагаемая в STEP и предлагающая к использованию язык EXPRESS, имеет ряд существенных недостатков:

- для успешного внедрения такого подхода необходимо, чтобы АС, используемые на всех этапах ЖЦ изделия, поддерживали стандарт на уровне протоколов и использования моделей;
- в связи с тем, что многие АС не поддерживают этот стандарт, зачастую предлагается использовать STEP-трансляторы, которые являются узкоспециализированными интерфейсами для конкретных АС и усложняют возможность масштабирования и модификации соответствующих АС и порядка их взаимодействия (в соответствии с принципами управления изменениями);
- затруднено внесение изменений в прикладное ПО интеграции данных, поскольку при внесении изменений в используемые концептуальные модели появляется необходимость изменения соответствующего прикладного протокола и перекодирования ПО, его использующего. В то же время специфика предприятий зачастую не позволяет использовать готовые прикладные протоколы STEP;
- отсутствие возможности задействовать существующие на предприятиях связанные с АС БД в процессах интеграции данных;
- стандарт STEP не поднимает вопрос об оценивании ТС и надежности изделий, рассматриваемый в рамках CALS-технологий;
- невозможность перейти от сформированных моделей непосредственно к исполняемому коду, поскольку EXPRESS не является исполняемым языком программирования, а перейти от функциональных моделей IDEF0 к коду также невозможно.

Таким образом, можно сделать вывод о том, что на текущем этапе развития стандарта STEP, он не может быть использован в решении задач информационно-аналитической поддержки ЖЦ СОТО по указанным выше причинам.

1.8. Задача мониторинга состояния в информационно-аналитической поддержке жизненного цикла сложных объектов

1.8.1. Задача мониторинга состояния

Современные СОТО относятся к классу динамических систем в традиционном смысле теории системного анализа. Эта особенность связана с необходимостью рассмотрения СОТО, как системы, непрерывно изменяющейся во времени в результате влияния на неё внешней среды. В связи с этим в работе будет использоваться ряд определений из [31, 160].

Определение П. 1.6. Динамической системой называется система, удовлетворяющая принципу причинности [31].

В общем случае, динамическая система может быть представлена следующим образом (рис. П. 1.8).



Рис. П. 1.8. Общий вид динамической системы

Проблема наблюдения состояния динамической системы является одной из четырех фундаментальных проблем системных исследований. Её теоретическое и прикладное значение связано с тем, что концепции динамической системы и её состояния позволяют разработать эффективные математические инструменты для проникновения в глубинную сущность причинно-следственных связей, присущих реальному миру, строго сформулировать и решить задачу восстановления скрытых причин явлений на основе фиксации доступных для измерения и регистрации характеристик этих явлений (реакции системы). В различных ситуациях используются различные определения понятия состояния. В общем случае может быть использовано следующее.

Определение П. 1.7. Совокупность характеристик системы, знание которой при известном входном воздействии является достаточным для однозначного определения выходной ситуации в будущем, называется *состоянием динамической системы* [31].

Для технических систем в рамках научных исследований, связанных с ракетно-космической отраслью, используются другие определения.

Определение П. 1.8. *Техническое состояние* – совокупность изменяющихся в процессе производства, испытаний, эксплуатации свойств (качеств) объекта управления (ОУ), характеризующих его функциональную пригодность в заданных условиях применения. ТС определяется посредством оценивания его параметров, множество значений которых образует некоторое пространство параметров, а определение ТС заключается в указании некоторой точки в этом пространстве [66].

Определение П. 1.9. *Состояние системы* – совокупность свойств, признаков (параметров) системы, которые отражают наиболее существенные стороны функционирования системы [138].

Задача наблюдения (или оценивания) состояния системы неразрывно связана с понятиями идентификации системы, мониторинга состояния. Основной целью решения проблемы наблюдения является нахождение алгоритма наблюдения, позволяющего по известной реакции системы определить искомое состояние в заданный момент времени. АС, основанные на таких алгоритмах, можно называть системами мониторинга.

Определение П. 1.10. *Мониторинг* – процесс информационного обеспечения управления сложной системой [138].

Определение П. 1.11. *Мониторинг* – процесс систематического сбора информации о параметрах состояния некоторой технической системы, социально-экономической среды либо состояния выполнения различного рода соглашений [84].

Определение П. 1.12. Система мониторинга – это ИАС непрерывного наблюдения, сбора, обработки и исследования информации о состоянии объекта, его функционировании и развитии в течение определенного периода времени, создаваемая и регулируемая субъектами мониторинга с целью обеспечения полной, своевременной и достоверной информации и соответствующей организации эффективного функционирования и развития ОУ [110].

Цель мониторинга – получение обобщенных оценок совокупности параметров состояния, значения которых в явном виде указывают либо степень работоспособности рассматриваемого объекта, либо место и вид возникшей неисправности, либо являются оценками прогнозируемых явлений и процессов с заданной точностью и интервалом прогноза и т.п. Необходимость оценивания состояния связана с необходимостью организации эффективного функционирования и развития ОУ и его элементов. Качество реализации функций мониторинга обеспечивается соблюдением научно обоснованных требований и принципов, к числу которых следует отнести: целенаправленность, достоверность, оперативность, систематичность, комплексность, валидность, плановость [132].

Соблюдение в ходе мониторинга данных принципов способствует наиболее полному и всестороннему исследованию, в результате которого представление и знание об объекте становится адекватным и истинным.

Для случая детерминированной внешней среды проблему наблюдения (решения задачи мониторинга состояния) можно проиллюстрировать следующим образом (рис. П. 1.9).

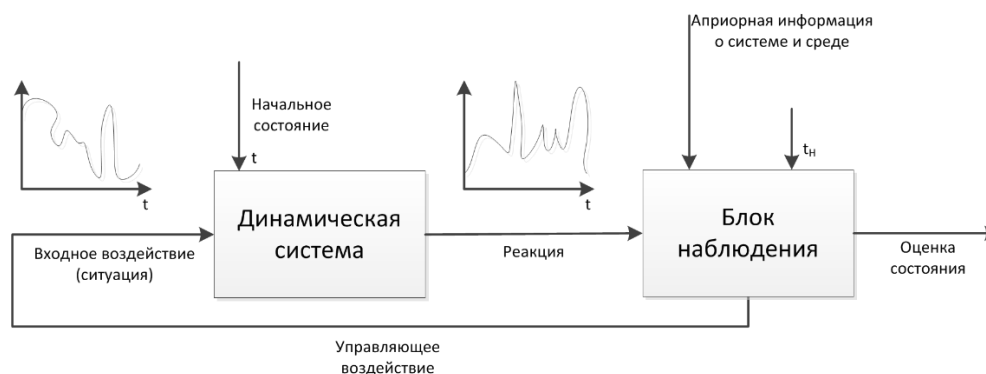


Рис. П. 1.9. Проблема наблюдения состояния динамической системы

Для рассматриваемой проблемы полагается известной сама динамическая система, множество управляющих воздействий и среда функционирования, множество моментов времени наблюдения, некоторый текущий момент времени наблюдения t_H . Задача состоит в определении состояния динамической системы по её реакции на входные воздействия.

В задаче мониторинга классически выделяют три основных этапа: сбор информации, её обработка и анализ [136]. Наибольшие сложности вызывает этап анализа информации, связанный с необходимостью извлечения семантики из показателей, характеризующих состояния СОТО, что требует привлечения эвристических экспертных знаний, трудно поддающихся формализации.

1.8.2. Системный анализ моделей, методов и алгоритмов мониторинга состояния СОТО

В настоящее время предлагается целый ряд различных методов, моделей и алгоритмов мониторинга состояния СОТО [7, 9, 61, 66, 78, 80, 81, 96, 97, 121, 130, 132, 135, 136, 138, 175].

Так, например, в работе [7] предлагается рассмотрение системы мониторинга как информационно-аналитической надстройки над системами класса АСУ ТП, системами учета и диспетчерскими системами для производственных предприятий. Система основана на построении информационной модели состояния в виде многомерного массива значений, характеризующих те или иные аспекты функционирования предприятия, и для указанного массива решается конкретная многокритериальная оптимизационная задача в зависимости от целей функционирования системы. При этом создается отдельная подсистема сбора данных из различных источников и подсистема генерации программного кода, предполагающего «сшивку» массива значений, характеризующего состояние объекта мониторинга.

В работе [78], где система мониторинга ориентирована на анализ информации о состоянии ракет-носителей, используется специально разработанный

алгоритм допускового контроля значений параметров, характеризующих состояние ракеты. Алгоритм предполагает использование экспертной БЗ, представленной допусковыми интервалами, и расчетов отклонения реальных значений от допустимых.

В работах [96, 97] рассматривается прецедентная методология оценивания состояния, предполагающая создание базы прецедентов, где прецедент представляется совокупностью значений параметров, характеризующих СОТО, а также решением по управлению, связанным с ним. На основе методов распознавания образов осуществляется поиск ближайшего прецедента к наблюдаемому текущему по базе прецедентов и применяется найденное решение. При этом выбираемая мера близости прецедентов зачастую имеет эвристический характер, адекватность которой для множества всех состояний оценить невозможно, а применяемое решение требует адаптации, которую сложно осуществить, поскольку не ясно, как близость прецедентов влияет на коэффициент адаптации.

В работе [81] отмечается, что традиционно при решении задачи параметрической идентификации строится уравнение модели объекта мониторинга и в зависимости от критерия идентификации применяются методы максимального правдоподобия, байесовских оценок, фильтра Калмана, наименьших квадратов, осредненных невязок, стохастической аппроксимации и др. Делается вывод о том, использование такого подхода сопряжено с рядом трудностей:

- модель объекта «вход-выход», как правило, не имеет явной интерпретации;
- отсутствует прямая возможность работы с переменными «вход-выход», которые имеют качественную природу;
- отсутствует возможность обнаружения информации о структурных свойствах объектах, выражаемых посредством продукционных выражений «ЕСЛИ – ТО».

В качестве альтернативы предлагается метод структурной идентификации, основанный на использовании экспертных знаний о функционировании

объекта мониторинга и представимый в виде нечеткой логической БЗ с использованием лингвистических термов.

По литературным данным во многих ранее разработанных и современных системах идентификации состояния объектов используются детерминистские математические модели «прямого счета», которые однозначно и достаточно просто определяют, что делать с объектом, если у него наблюдаются определенные внешние параметры [175].

При этом не ставится и не решается вопрос о том, как связаны эти параметры с теми или иными состояниями объекта. Эта позиция соответствует точке зрения, состоящей в том, что «по-умолчанию» принимается их взаимно-однозначная связь. Поэтому термины: «параметры объекта» и «состояния объекта» рассматриваются как синонимы, а понятие «состояние объекта» в явном виде вообще не вводится. Однако очевидно, что в общем случае связь между наблюдаемыми параметрами объекта и его состоянием имеет динамичный и вероятностный характер.

Таким образом, традиционные системы мониторинга, по сути дела, являются системами параметрической идентификации, т.е. системами, которые оперируют не состояниями объекта, а лишь его наблюдаемыми параметрами. Решение об управляющем воздействии принимается при таких системах «вслепую», т.е. без формирования целостного образа объекта и окружающей среды в их текущем состоянии.

В настоящее время существует и развивается методология структурно-функционального моделирования и проектирования программ автоматизированного анализа ТС СОТО на основе G-моделей, в рамках которой существенно пересмотрены взгляды на задачу мониторинга ТС и предложен формальный аппарат, обеспечивающий возможность конструктивно анализировать ТС сложных объектов, обнаруживая существенные семантические связи их характеристик и наиболее полно и достоверно формируя выводы об их функционировании (см. приложение 1.8.3).

1.8.3. Методология автоматизированного анализа состояния сложных объектов на основе G-моделей

В работах [130 – 136, 138] рассматривается методология автоматизированного анализа информации о состоянии СОТО, основанная на так называемых *G-моделях* и используемая для информационного обеспечения ЛПР по управлению СОТО. Разработка соответствующего теоретико-методического аппарата стала логическим следствием ряда предпосылок в различных ПрО, к числу которых можно отнести следующие:

- наличие жестких временных ограничений на получение результатов информационного обеспечения;
- высокие требования к достоверности и точности результатов анализа;
- оперативность обработки больших потоков информации;
- разнообразие типов информации, используемой для принятия решения о состоянии ОУ – как по физической природе информации, так и по её носителю;
- широкий диапазон как задач, решаемых при анализе информации (контроль функционирования объекта анализа, диагностирование неисправностей, прогнозирование поведения, оценка технических характеристик и т.п.), так и типов самих объектов анализа;
- территориальная распределенность элементов СОТО;
- многочисленность разнородных классов потребителей результатов информационного обеспечения, состав которых меняется во времени;
- сложность формализации семантики процессов информационного обеспечения;
- малые сроки «постановки на информационное обслуживание» новых объектов анализа и, соответственно, небольшая стоимость этого процесса;
- невысокие требования к программистской квалификации персонала, как конечных пользователей результатов автоматизированного анализа, так и специалистов (экспертов) по управлению СОТО;

- требования по унификации, модульности и масштабируемости построения систем информационного обеспечения, реализация которых позволяет быстро комплектовать специализированные программно-аппаратные комплексы в зависимости от конкретных условий и целей применения СОТО;
- устойчивая (надежная) работа системы информационного обеспечения при возникновении различных нештатных (непредусмотренных специалистами по соответствующим объектам) ситуаций, способных возникать на этапе применения.

Основные аспекты методологии автоматизированного анализа информации о состоянии СОТО представлены на рис. П. 1.10 [138]. Далее будет дано краткое описание рассматриваемой методологии в соответствии с рисунком.

Одним из средств *концептуального моделирования* и программирования ПрО являются обобщенные ВМ (ОВМ) [193]. К классу ОВМ, в частности, относится МПЗ, называемая G-моделью [132]. G-модель является основой для создания имитационно-аналитического полимодельного описания и возникла как результат обобщения таких существующих подходов в рамках систем ИИ, как концептуальное программирование, программирование в ограничениях, мультиагентное моделирование и др. G-модель представляет собой структуру параметров, связанных концептами вычислимых функциональных отношений в виде операторов и предикатов, обуславливающих выполнимость того или иного оператора в вычислительном процессе. ПК системы мониторинга ТС СОТО может быть синтезирован посредством формального вывода схем программ вычислений в виде *G-сетей* по G-моделям. При этом такой ПК позволяет осуществлять оценивание состояния СОТО в режиме РВ с применением метода структурно-потокowo-многоуровневого (СПМ) распознавания образов. Целью функционирования ПК является получение обобщенных оценок совокупности параметров состояния системы, соотносимых с целями анализа ОУ.

Теоретический аппарат G-моделей позволяет:

- в рамках парадигмы декларативного построения ИС формировать G-модели как своеобразный вид ВМ или ОВМ, представляющие собой семантическую

сеть, вершинам которой сопоставляются денотаты ПрО, а дугам – отношения между ними.

- при переходе от задач автоматизированной обработки информации, имеющих чисто синтаксический характер, к задачам ее анализа, связанных уже с семантическим аспектом представления анализируемых данных использовать концепцию инвариантности (наличия соответствия) реального процесса и состояния вычислительного процесса. Её необходимость объясняется тем, что при наблюдении за состоянием СОТО на самом деле производится наблюдение за нужными параметрами вычислительного процесса.
- формализовать процесс анализа информации в РВ по заданной цели анализа (или их множеству) как процесс рекуррентного доопределения ТС на основе модели программы анализа в виде G-сети.
- использовать для построения модели процесса анализа информации метод, учитывающий специфику ПрО, который основан на описании класса задач анализа информации в виде G-моделей, на доказательстве разрешимости конкретной задачи с помощью грамматик специального вида – р-грамматик и на управлении процессом анализа на базе G-сетей.
- на основе G-сетей моделировать различные по мощности вычисления. Т.е. G-сети представляют собой универсальную моделирующую систему, порождающую рекурсивно-перечислимые классы языков, и эквивалентную по своей мощности машине Тьюринга.
- для исключения возможных ошибок в ходе вычислений по программе анализа на основе G-сети использовать методы *верификации* модели, за счет чего минимизировать время анализа и повысить достоверность результатов.

Отметим, что комплексный подход автоматизированного оценивания ТС СОТО на основе G-моделей позволил найти ему применение в целом ряде ПрО, таких как атомная энергетика, экология, логистика, военные приложения, космическая отрасль (системы подготовки и пуска ракет-носителей, мониторинг активного участка полета ракеты), ситуационные центры и др. [1, 9, 80, 137, 139].

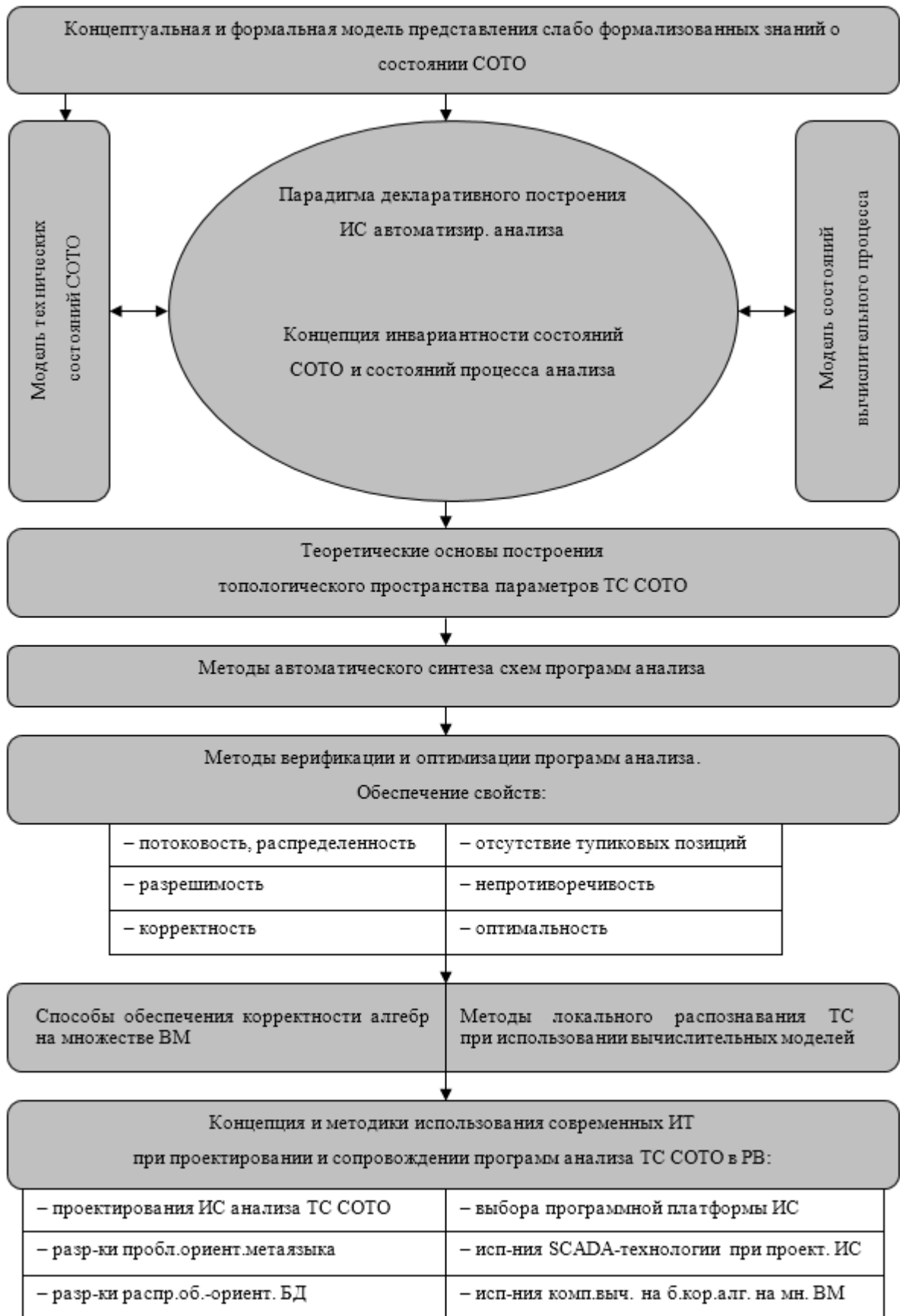


Рис. П. 1.10. Структура теоретических основ и методов автоматизированного анализа информации о ТС СОТО на основе G-моделей

1.8.4. Анализ возможности применения методологии автоматизированного анализа на основе G-моделей в задаче информационно-аналитической поддержки

Отметим, что особенностью СИАП ЖЦ СОТО является необходимость оценивания состояния СОТО на всех этапах его ЖЦ. Эта особенность не позволяет напрямую воспользоваться аппаратом G-моделей для решения поставленной в работе задачи в силу многогранности условий и требований к информационно-аналитической поддержке ЖЦ СОТО, которые G-модели не покрывают. Анализ возможности применения G-моделей в этой части представлен в таблице П. 1.5.

Таблица П. 1.5. Анализ возможностей применения G-моделей для задачи проектирования СИАП ЖЦ СОТО

Аспект ПрО	Возможности методологии автоматизированного анализа состояния СОТО на основе G-моделей	Оц.
Одной из основных задач СИАП ЖЦ СОТО является оценивание состояния СОТО и процессов, протекающих в нем.	Ориентирована на решение задачи мониторинга состояния техническим систем. Остается открытым вопрос анализа организационных аспектов, как слабо формализованных процессов функционирования СОТО.	+/-
Проектирование СИАП ЖЦ СОТО связано с необходимостью извлечения и представления экспертных знаний по решению аналитических задач, возникающих при оценивании состояния СОТО на различных этапах его ЖЦ.	Позволяет моделировать решения аналитических задач на основе теории концептуального программирования, теории распознавания образов, потокового доопределения значений вычисляемых параметров по наборам заданных функциональных вычислимых отношений.	+
Автоматизация решения аналитических задач в составе системы информации о ТС и надежности СОТО непосредственно связана с возможностью синтеза ПК СИАП ЖЦ СОТО на основе формальных моделей, спроектированных экспертами ПрО.	Позволяет синтезировать схемы программ решения аналитических задач по концептуальным моделям с помощью контекстно-свободной р-грамматики. При этом такие схемы, с точки зрения организации вычислений, сопоставимы по мощности с машиной Тьюринга и покрывают класс частично-рекурсивных функций.	+
СИАП ЖЦ СОТО в общем случае должна представлять собой территориально распределенный ПК, где его компоненты позволяют решать задачи, связанные с конкретными этапами ЖЦ, а также в условиях распределенного выполнения задач должна обеспечиваться конфиденциальность обрабатываемых данных.	Позволяет проектировать распределенные мультиагентные системы, где каждому интеллектуальному агенту сопоставлена некоторая своя G-модель (собственная модель решения аналитической задачи). С учетом возможности вложения G-моделей друг в друга, любой агент может решать определенный набор аналитических задач.	+
ПК СИАП ЖЦ СОТО должен функционировать таким образом, чтобы обеспечивать заинтересованные организации в составе СОТО актуальной и достоверной информацией о его состоянии.	Методология предполагает возможность организации потоковых, асинхронных и параллельных вычислений по имитационно-аналитическим моделям. При этом доказывается возможность функционирования в режиме РВ за счет свойства завершаемости вычислений по алгоритмам за прогнозируемое время. Таким образом, при корректности сформиро-	+

	ванных экспертных моделей обеспечивается актуальность и достоверность предоставляемой в результате анализа информации.	
Поскольку ПК СИАП ЖЦ СОТО должен функционировать на основе экспертных знаний, в силу зачастую слабой формализованности и противоречивости этих знаний, а также их разобшенности по этапам ЖЦ необходимо обеспечивать их совместное представление и верификацию проектируемых моделей, основанных на этих знаниях.	В рамках методологии предлагаются методы верификации создаваемых схем программ, однако с рассматриваемых в работе позиций отсутствует общее представление модели процессов и соответствующие методы её верификации, отсутствуют методы верификации проектируемых моделей по спецификациям требований решаемых задач.	+/-
Задача формирования электронного паспорта СОТО определяет необходимость иметь актуальную, полную и достоверную информацию о состоянии СОТО на всех этапах его ЖЦ, хронологию изменения его состояния с целью выявления качественных и надежных показателей работы с изделием.	G-модели предполагают «сеансовый» режим функционирования, когда ПК осуществляет обработку и анализ телеметрируемой информации технических объектов. ПК заканчивает своё выполнение в результате окончания передачи исходных данных. При этом результаты обработки не сохраняются и не систематизируются в некотором едином представлении для их последующей обработки. Кроме того, сами результаты функционирования ПК в виде структуры значений параметров плохо поддаются последующему анализу в силу отсутствия наглядности при большом количестве параметров, которыми может характеризоваться СОТО на всех этапах ЖЦ. Иначе говоря, G-модели не позволяют в полной мере создать структурированную хронологию изменения состояния СОТО, которая бы обеспечивала наблюдаемость изменения существенных характеристик СОТО при прохождении им этапов ЖЦ.	-
Работа с изделием связана с необходимостью учитывать различные организационные аспекты, влияющие на состояние изделия и выражающиеся формированием документов об изделии или о ходе работы с ним, а также необходимостью оперативной передачи этой документации организационным единицам, связанным с последующими этапами или стадиями ЖЦ СОТО. Необходимо также учитывать, как влияет появление (или отсутствие) определенных документов и их содержание на дальнейший ход работы с изделием. При этом такие документы зачастую предоставляют либо слабоструктурированную информацию, либо, наоборот, имеют сложную структуру.	G-модели ориентированы, в первую очередь, на мониторинг состояния технических систем, информация о которых представляется набором значений параметров. В связи с этим методология опирается на использование БД с моделями данных, достаточными для хранения таких наборов. В то же время в рассматриваемой работе некоторые исходные данные могут быть представлены взаимосвязанными структурированными наборами значений, представляющими при заданной интерпретации некоторые декларативные выражения, факты ПрО, отражающие некоторые организационные аспекты СОТО. Такие факты подлежат предварительному семантическому разбору, но ни такая задача, ни способы представления и хранения таких данных в методологии не рассматриваются.	-
Оценивание состояния изделия связано с информацией об изделии, представленной как различными документами, так и большими объемами разнородных данных, формируемых в результате функционирования различных ИС, представленных на территориально распределенных предприятиях, в связи с чем необходимо интегрировать эти данные, обрабатывать по единым принципам и передавать между организациями, иначе говоря, формировать ЕИП КК. Такие данные при этом хранятся средствами БД различных типов: реляционных, сетевых, иерархических, объектных.	В общем случае G-модели позволяют решать задачи интеграции, обработки и анализа данных из различных источников за счет мультиагентности и потоковости вычислений с принципом управления по данным. Это позволяет осуществлять интеграцию данных и приложений в соответствии с поставленной задачей. Однако, с учетом отмеченного в предыдущем пункте, отсутствие возможности представления и обработки сложно структурированных семантических выражений, представленных в ПрО средствами различных оцифрованных документов, их метаданных и БД различных типов, не позволяет в полной мере осуществить такого рода интеграцию.	+/-

Работы с изделием на всех этапах ЖЦ СОТО связаны с различными технологическими и организационными процессами, прохождение этапов которых определяет переход СОТО в то или иное состояние. В связи с этим методы проектирования СИАП ЖЦ СОТО должны предполагать возможность формализованного описания таких процессов и их наблюдаемость с целью организации корректного хода вычислений с точки зрения последовательности выполняемых операций, их взаимосвязей и полноты описания этих процессов.	Выразительные возможности языка представления знаний (ЯПЗ), предложенных в G-моделях, не позволяют наглядным образом моделировать реальные процессы.	–
Ход задания (проектирования) спецификаций процессов работы над изделием должен предполагать его декомпозицию так, чтобы эксперты в своих областях, наблюдающие «сверху», могли описывать общие закономерности процессов в ПрО, а специалисты ниже уровнем могли описывать свои частные аналитические задачи.	Методология позволяет распределить процесс проектирования между его участниками, однако такое разделение предполагает моделирование сети задач, что не позволяет описывать их иерархию, основываясь на общих принципах декомпозиции сложных систем. В том же случае, когда используются вложенные G-модели, отсутствует возможность визуально наблюдать и верифицировать проектируемые модели с точки зрения задаваемой спецификации процессов.	–
Итого: 4 «+», 3 «+/-», 4 «-».		

Результаты анализа возможности использования методологии автоматизированного анализа состояния СОТО на основе G-моделей показали, что G-модели обладают рядом необходимых свойств для решения поставленной задачи. Предпосылки, исходя из которых разрабатывался указанный методологический аппарат, актуальны и для рассматриваемой темы. Тем не менее, ряд особенностей требует расширения возможностей теоретического аппарата для решения поставленной задачи в части моделирования процессов решения аналитических задач, обработки и хранения сложно структурированных семантических выражений как информации о происходящем в ПрО, многоуровневой декомпозиции процесса проектирования СИАП ЖЦ СОТО.

1.9. Системы оперативных аналитических вычислений в режиме реального времени

1.9.1. Согласование данных в интегрированном хранилище

В процессе создания ИАС и её информационного хранилища выделяют проблему обеспечения необходимого качества данных, их достоверности, согласованности, соответствия установленным ограничениям и бизнес-правилам [11, 20]. Качество загружаемых и содержащихся в ХД данных достигается

системой организационных и программно-технических мероприятий. При этом учитывают два аспекта: структурный и смысловой. Структурный аспект заключается в представлении данных из источников в едином формате. Смысловой аспект состоит в содержательной интерпертации знаковых структур данных – даже при наличии согласованных форматов могут быть разные толкования близких понятий, встречаемых в различных источниках данных. Также должна соблюдаться целостность и полнота при доступе к данным этих источников. Одним из распространенных подходов при решении этой проблемы является использование витрин данных, как способа подготовки данных для анализа. При этом данные могут быть агрегированы, обеспечивается единая хронология, согласуются различные форматы, устраняются противоречивость и неоднозначность данных.

Однако существенным является тот факт, что при решении аналитических задач возникает необходимость построения сложных зависимостей между параметрами, значения которых будут обрабатываться на этапе исполнения. К числу таких задач можно отнести различные классификационные, регрессионные задачи, поиск ассоциативных правил, построение деревьев решений, предиктивную аналитику и др. Стоит отметить, что зачастую в рамках OLAP используют методы Data Mining, позволяющие искать закономерности в «сырых» данных. В то же время ряд ПрО предполагает наличие набора эвристических правил обработки информации, формируемых экспертами за время их работы. В связи с этим эти закономерности могут быть заранее известны. В таком случае в рамках OLAP-системы необходимо предложить соответствующий аппарат, позволяющий по запросу определенного набора данных осуществлять их обработку в соответствии с этими правилами.

Множественность измерений предполагает многомерную модель данных. Традиционно такое представление в OLAP-технологиях называют многомерным *гиперкубом* [11]. Цель такого концептуального представления связана с необходимостью одновременного анализа по нескольким измерениям различных источников данных, который Э. Кодд называет

многомерным анализом. Измерением называется последовательность значений по одному из анализируемых параметров. При этом результат выполнения запроса к источникам данных представляет собой некоторое пересечение по осям измерений, называемое мерой. Традиционно выделяют следующие операции, выполняемые над гиперкубом (см. рис. П. 1.11) [11]:

- a) выбор некоторой конкретной меры;
- b) вращение – изменение расположения измерений (например, изменение расположения строк и столбцов);
- c) срез – выбор подмножества многомерного массива по одному или нескольким критериям;
- d) консолидация – переход к агрегированному представлению некоторого подмножества многомерного массива;
- e) детализация – обобщение данных по некоторым критериям.

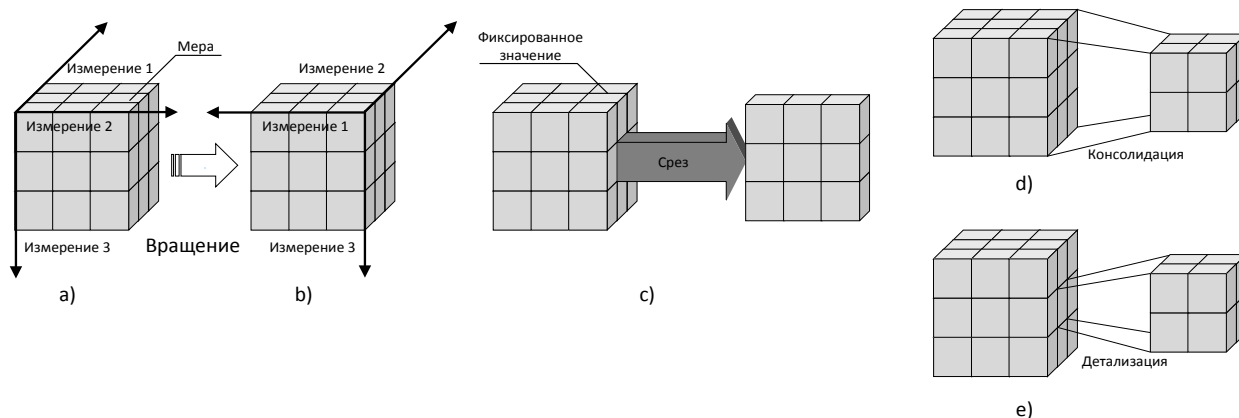


Рис. П. 1.11. Некоторые примеры операций на многомерном гиперкубе

Таким образом, неотъемлемой задачей OLAP-систем является задача глубинного согласования на структурном и семантическом уровнях обрабатываемых и анализируемых данных, что коррелирует с ранее рассмотренными аспектами необходимости совместного анализа агрегируемых данных для обеспечения полноты и достоверности результатов анализа.

1.9.2. Стандарт CWM в OLAP-системах

При проектировании OLAP-систем зачастую предлагается использовать стандарт CWM, разработанный консорциумом OMG и используемый для единообразного доступа к метаданным разнородных источников данных, которые в общем случае семантически взаимосвязаны [11]. CWM определяет интерфейсы, которые могут быть использованы для обмена метаданными между ХД и ИАС. При этом с помощью языка UML формируются предметно-ориентированные метамодели, определяющие структуры сущностей и их взаимосвязи, определенные в источниках данных. Доступ к метаданным происходит посредством обмена файлами метаданных на языке XML.

На практике стандарт не получил широкого распространения по смежным со стандартом STEP причинам (см. раздел 1.2.1). Построение метамодели ХД требуется осуществлять вручную, при изменении схем или структур источников данных проводить моделирование заново и поддерживать модели всегда в актуальном состоянии.

1.10. Системный анализ методов интеграции информационных ресурсов

Прежде всего, необходимо заметить, что существующие методы предполагают различные уровни интеграции (см. рис. П. 1.12) [207].

Из рис. П. 1.12 видно, что по мере продвижения снизу вверх консолидируемые данные обретают семантику, в результате чего становятся доступной информацией для понимания человеком. В этом заключается отличие понятий консолидации и интеграции. Соотнесем эту классификацию с известными технологиями интеграции данных.

Системы интеграции данных могут обеспечивать интеграцию данных на физическом, логическом и семантическом уровне [86]. Интеграция данных на физическом уровне является простейшей консолидацией на основе единых форматов. Интеграция данных на логическом уровне предусматривает возможность доступа к данным в терминах единой глобальной схемы.

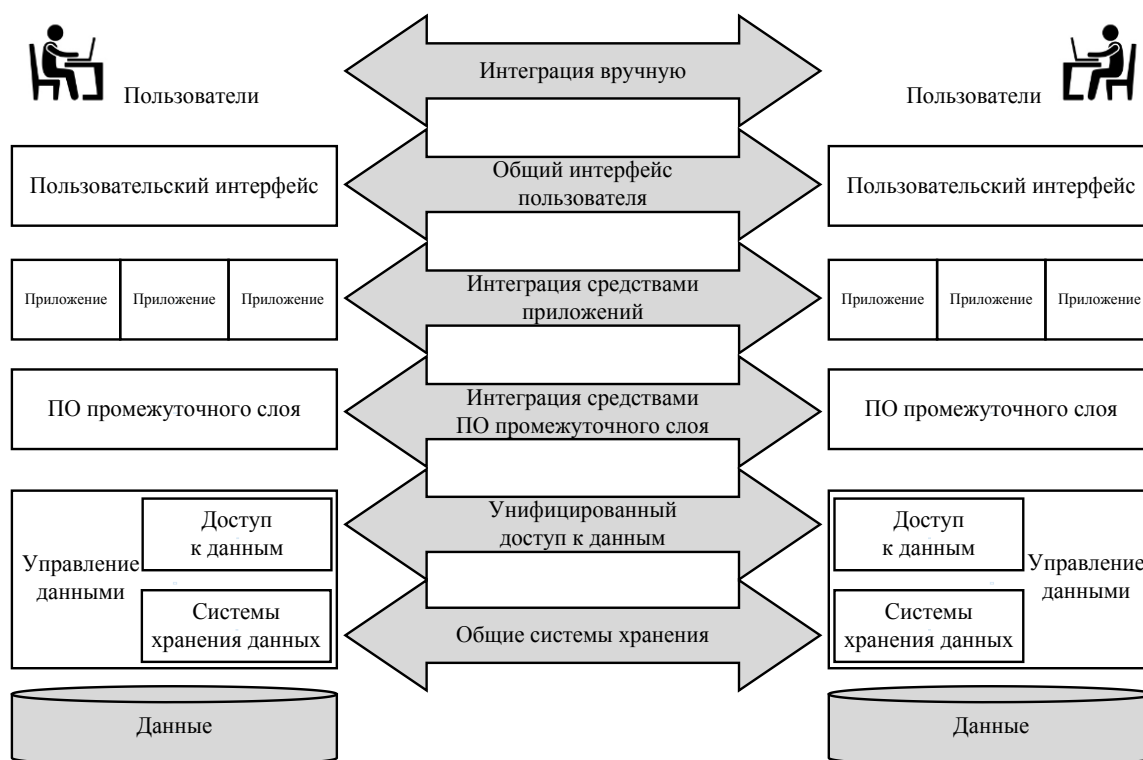


Рис. П. 1.12. Классификация методов интеграции информационных ресурсов по Клаусу Диттриху

Поддержку единого представления данных с учетом их семантических свойств обеспечивает интеграция метаданных, предполагающая создание метамоделей единой системы понятий всех источников данных [21, 38, 45]. Трудности ее разработки в конкретных ситуациях могут быть связаны с наличием семантической неоднородности.

К настоящему моменту было разработан ряд методов интеграции. Рассмотрим наиболее известные из них.

1.10.1. Консолидация данных

Данные собираются из нескольких исходных БД и интегрируются в одно постоянное место хранения на основе принципа “мгновенных снимков” первичных данных [23]. К недостаткам подхода можно отнести большие времена отставания при обновлении, высокую стоимость реализации, отсутствие

наглядности процесса изменений данных, необходимость в значительных вычислительных ресурсах и ресурсах памяти. С другой стороны, подход обладает простотой в реализации.

1.10.2. Распределение данных

Метод распределения данных осуществляет их копирование из одного места в другое на основе механизма репликации [23]. Обычно используется в оперативном режиме и зависит от определенных событий. Обновления в исходной БД могут передаваться в конечную БД синхронно или асинхронно. К преимуществам подхода можно отнести гарантированную доставку данных в систему назначения, двусторонний обмен данными между первичными и конечными системами. Технологиями, поддерживающими распределение данных, являются EAI и тиражирование корпоративных данных (Enterprise Data Replication (EDR)).

1.10.3. Федерализация данных

Подход, заключающийся в реализации взаимных связей между каждой из БД [21, 23] на основе создания единого виртуального представления. Основными недостатками является необходимость кодирования узкоспециализированных методов трансляции запросов от одной система управления БД (СУБД) в терминах другой, а также рассогласованность и низкое качество «сырых» данных. Основное преимущество подхода отсутствие необходимости физической консолидации исходных источников данных.

1.10.4. Хранилища данных

Копии фрагментов информации из нескольких СУБД сохраняются в единой БД, как правило, с предварительной обработкой с целью их согласования [21, 23]. Успех построения модели ПрО для соответствующего ХД зависит от того, насколько каждая из этих моделей удовлетворяет требованиям адекватности, поскольку отсутствует формализация процесса интеграции и формирование соответствующей модели происходит эвристически в диалоговом режиме с владельцами источников данных.

1.10.5. Медиаторы

Программные компоненты – посредники, обеспечивающие поддержку виртуальных БД на основе семантической интеграции [21, 47, 214]. Медиатор транслирует каждый запрос в соответствующие подзапросы к источникам данных, затем агрегирует ответы в формат исходного запроса. Средствами посредников поддерживаются унифицированные метаописания интегрируемых источников данных, разрабатываемые для конкретной ПрО. Процесс функционирования медиаторов формализован на основе использования онтологических спецификаций источников (см. приложение 2.3.5). Иногда для них создается интегрированная *онтология* используемых источников. В зависимости от реализуемого подхода в таких случаях говорят об архитектурах Global as View (GAV) (использование терминов локальных онтологий в глобальной) и Local as View (LAV) (использование терминов глобальной онтологии при обращении к локальным) [39, 40]. Достоинство такого подхода заключается не только в том, что основой пользовательского интерфейса является при этом высокоуровневая семантическая модель согласованных данных, но и возможность рассуждений в терминах онтологии, служащей концептуальной моделью.

1.10.6. Анализ существующих методов интеграции данных и приложений

Проведенный анализ показал, что интеграция информационных ресурсов в ЕИП является стратегическим направлением в CALS-технологиях. Однако современное состояние интеграционных технологий, моделей и методов в настоящее время не позволяет эффективно решать множество практических задач из-за функциональной ограниченности существующих подходов. Стоит привести несколько примеров существующих систем интеграции информационных ресурсов.

Примерами интеграции данных средствами ПО могут служить системы Oracle Data Integration Suite из состава Oracle Fusion Middleware, MapForce компании Altova, Informatica PowerCenter. Среди крупных производителей систем интеграции данных: IBM, SAP/BusinessObjects и Microsoft. Среди небольших компаний: Ab Initio, iWay Software, Syncsort, DMExpress, Embarcadero

Technologies, Sunopsis [207]. В этих системах широко используется ряд официальных международных стандартов. Среди них стандарты БД ISO/IEC SQL, ISO/IEC SQL/MED, стандарт объектных данных консорциума ODMG, стандарты CORBA и UML консорциума OMG, стандарты платформы XML консорциума W3C, стандарт Дублинского ядра консорциума OCLC и многие другие [86].

Результаты анализа показывают, что существующие системы в основной своей массе представлены только за рубежом, а их решения так или иначе сводятся лишь к консолидации данных на том или ином уровне абстракции. В связи с этим можно говорить о необходимости разработки отечественных систем интеграции информационных ресурсов. Кроме того, важно отметить, что в рамках рассматриваемой в работе задачи СИАП ЖЦ СОТО также должна поддерживать решения задачи интеграции данных и приложений. На основе проведенного анализа можно выделить следующие требования к системе:

- необходимость обеспечения ИИС (ЕИП), характеризующегося чертами предметной ориентированности, интегрированности, неизменчивости, поддержки хронологии;
- необходимость совместной реализации принципов ЕП и ЕАІ на основе создания одноранговой сети автоматизированных бизнес-процессов, поддерживающих обмен сообщениями и доступ к источникам данных в соответствии с их предметной ориентацией;
- использование комбинированных декларативно-процедурных спецификаций процессов сбора и обработки интегрируемых ресурсов;
- повышение гносеологического уровня в моделях интеграционных процессов с возможностью оперирования терминами ПрО для всех источников данных;
- реализация автоматизированных бизнес-процессов на основе принципов семантических посредников (медиаторов) с использованием онтологических спецификаций источников данных.

2. МОДЕЛИ, МЕТОДЫ И ТЕХНОЛОГИИ ПРОГРАММНОЙ ИНЖЕНЕРИИ ИНФОРМАЦИОННЫХ СИСТЕМ

2.1. Инженерия требований при разработке информационных систем

2.1.1. Специфицирование требований к информационным системам

Программные требования устанавливают соглашения между пользователями (заказчиками) и разработчиками (исполнителями) в отношении того, что будет делать система [128, 238]. Зачастую это делается на естественном языке и оценка степени соответствия программы предъявляемым к ней требованиям производится на основании тестирования. В то же время существуют формальные методы и подходы, используемые для описания программных требований, предназначенные для однозначной их интерпретации при разработке ПО. Когда свойства системы могут быть записаны на некотором формальном языке, анализ на соответствие этим свойствам может быть произведен при наличии адекватной формальной системы, модели программы, методов *верификации* программ с возможностью определения значений этих свойств, чего тестирование обеспечить обычно не может [194]. Стандарт IEEE 830 «Recommended Practice for Software Requirements Specifications» является классическим примером стандарта на содержание, структурирование и методы описания программных требований [167].

Определение П. 2.1. Под *спецификацией* понимается точное формальное определение программной системы или ее части, под описанием – неформальная спецификация, иллюстрирующая тот или иной аспект системы [9, 65, 92, 194].

Описания обычно используются на ранних этапах разработки ПО или для документирования, спецификации – на стадии детального проектирования. Для визуального проектирования спецификации используются различные нотации. Нотация – это графический язык для описания моделей [30, 225].

Независимо от способа описания требований, выделяют критерии, которым должна удовлетворять спецификация требований. Вкратце некоторые из

этих критериев можно сформулировать следующим образом [167, 202] (см. таблицу П. 2.1).

Таблица П. 2.1. Некоторые критерии, определяемые для спецификаций требований

Требование	Содержательное пояснение
Атомарность	каждое утверждение (формулировка требования) должно представлять собой один несводимый к другим элемент иерархии требований, пригодный для установки связей с ним
Уникальность	каждое требование должно иметь собственный уникальный идентификатор
Выполнимость	требование должно быть технически реализуемо
Однозначность	требование должно быть понятно сформулировано (исключать неоднозначное толкование)
Проверяемость	должна существовать возможность проверки реализации каждого конкретного требования
Абстрактность	формулировка не должна навязывать определенные технические решения, характерные для более низких уровней требований (спецификаций)
Полнота	спецификация содержит всю необходимую информацию для своей корректной реализации в программе
Непротиворечивость	не существует требований, противоречащих друг другу
Отсутствие избыточности	каждое требование сформулировано только один раз (нет повторов)
Адекватность	на основе спецификации может быть разработана программа, выполняющая именно те функции, которые нужны заказчику
Трассируемость	всегда можно определить, с какими другими требованиями связано текущее, а также как оно реализуется в рамках этапов проектирования, кодирования и тестирования

Формализация и соблюдение указанных требований может обеспечить возможность синтеза программы, гарантированно удовлетворяющей поставленным в спецификации требованиям, на основе некоторой формальной процедуры. В этом случае фактически процесс синтеза программы напоминает механизм логического вывода. В 1968 г. Э. Бишоп отметил, что возможна цепочка чисто формальных преобразований: постановка задачи (спецификация), конструктивное доказательство (извлечение программы), программа [73, 194]. При этом необходимо и модели программ проверять на их корректность.

В рамках инженерии требований обычно выделяют функциональные и нефункциональные требования [128, 167, 238]. Функциональные – задают “что” система должна делать; нефункциональные – с соблюдением “каких условий” (см. таблицу П. 2.2).

Таблица П. 2.2. Типы требований к информационным системам

Группа	Тип	Описание
Функциональные требования	Бизнес-требования	Определяют высокоуровневые цели организации или клиента (потребителя) – заказчика разрабатываемого ПО.
	Пользовательские требования	Описывают цели/задачи пользователей системы, которые должны достигаться/выполняться пользователями при помощи создаваемой программной системы. Эти требования часто представляют в виде вариантов использования (Use Cases).
	Функционал. требования	Определяют функциональность (поведение) программной системы, которая должна быть создана разработчиками для предоставления возможности выполнения пользователями своих обязанностей в рамках бизнес-требований и в контексте пользовательских требований.
Нефункциональные требования	Бизнес-правила	Связаны с корпоративными регламентами, политиками, стандартами, законодательными актами, внутрикорпоративными инициативами, учетными практиками, алгоритмами вычислений и т.д. Они подразумевают организацию структуры бизнеса, контролируют или влияют на поведение бизнеса. Бизнес-правила часто определяют распределение ответственности в системе, отвечая на вопрос, кто будет осуществлять конкретный вариант, сценарий использования или диктуют появление некоторых функциональных требований.
	Внешние интерфейсы	Конкретизация аспектов взаимодействия с другими системами, операционной средой (например, запись в журнал событий операционной системы), возможностями мониторинга при эксплуатации и т.д.
	Атрибуты качества	Описывают дополнительные характеристики продукта в различных “измерениях”, важных для пользователей и/или разработчиков. Атрибуты касаются вопросов портируемости, интероперабельности (прозрачности взаимодействия с другими системами), целостности, устойчивости и т.п.
	Ограничения	Формулировки условий, модифицирующих требования или наборы требований, сужая выбор возможных решений по их реализации. В частности, к ним могут относиться параметры производительности, влияющие на выбор платформы реализации и/или развертывания (протоколы, серверы приложений, БД, ...), которые, в свою очередь, могут относиться, например, к внешним интерфейсам.
Системные требования		Иногда классифицируются как составная часть группы функциональных требований. Описывают высокоуровневые требования к ПО, содержащему несколько взаимосвязанных подсистем и приложений. При этом, система может быть как целиком программной, так и состоять из программной и аппаратной частей.

2.1.2. Согласование требований

При проектировании крупных систем необходимо применять системно-кибернетический подход и использоваться агрегацию и декомпозицию, которые позволяют создать систему взаимосвязанных решаемых задач. В контексте разработки требований, создание и анализ связей необходимы для понимания того, как требования высокого уровня – общие цели, задачи трансформируются в требования к их реализации. Иначе говоря, связи нужны между различными информационными уровнями представления [202].

В контексте бизнеса может быть интересна:

- стратегия бизнеса, которая конкретизируется как
 - задачи бизнеса, которые воплощаются как
 - организация автоматизированных бизнес-процессов.

В контексте системного проектирования интерес может фокусироваться на том, как:

- пользовательские требования (stakeholder requirements) удовлетворяются
 - системными и функциональными требованиями, которые уточняются в виде
 - моделей программ, которые позволяют организовать
 - вычисления.

К. Вигерс, известный специалист в инженерии требований отмечал, что необходимо комбинировать разные типы требований в процессе создания спецификаций [167]. Схожий аспект отмечается в задачах системного анализа и моделирования ПрО. В работах [133, 137, 165, 183] рассматривается принцип полимодельности (комплексного моделирования) в контексте того, что разноаспектность совместно решаемых задач в рамках многих ПрО требует привлечения при проектировании разнотипных моделей. В то же время в спецификации требований к конечному ПО моделирование помогает инженеру уточнять и детализировать реализацию системы путем разбиения ее на компоненты при движении вниз от одного уровня требований к другому [202]. Требования на каждом уровне являются «снимком» ПрО в том аспекте и в тех терминах, в

которых это интересует проектировщика. При этом детализация «снимка» увеличивается при движении вниз от уровня к уровню. Поэтому для описания различных аспектов системы обычно используется несколько различных взаимосвязанных моделей. Эта ситуация также характеризует «многоязычность» формальных моделей при проектировании для использования соответствующих синтаксиса и семантики на каждом уровне [73, 202]. Критичным является сохранение нужной степени детализации для каждого языка и соответствующей модели.

Таким образом, полимодельность можно понимать в двух аспектах – разнотипность моделируемых задач и разнотипность моделей по уровням представления. При условии их согласованности может иметь место следующий порядок моделирования (см. таблицу П. 2.3).

Таблица П. 2.3. Согласование требований различных уровней

Этап проект-я	Область	Точка зрения	Цель
Пользовательские требования, бизнес-правила	Область проблем	Представитель заинтересованной стороны – бизнес-аналитик	Определяет – что пользователь желает достичь с помощью создаваемой системы. Следует избегать формулировки конкретных решений.
Системные, функциональные требования, ограничения, внешние интерфейсы	Область специфики задач	Эксперт-непрограммист	Определяет, как система будет удовлетворять пользовательским требованиям.
Анализ требований	Область верификации требований	Система верификации требований	Определяет разрешимость и реализуемость требований.
Концептуализация вычислительных задач	Область решения	Инженер по знаниям	Определяет вычислительные задачи в рамках ограничений спецификаций.
Спецификация программы	Область алгоритмизации	Порождающая система вывода	Определяет, как в соответствии со спецификациями верхних уровней организовать программу, вычисляющую решение задачи.
Верификация программы	Область верификации программ	Система верификации программ	Определяет корректность сформированных программ.

Природа моделей на разных уровнях различна. На верхнем уровне используются пользовательские сценарии, характеризующие поведение будущей системы с точки зрения аспектов Про. Также определяется словарь Про

[129]. При переходе к системным требованиям могут использоваться различные типы функциональных моделей. Так, например, функциональные модели могут включать в себя различные UML- или IDEF-диаграммы.

2.1.3. Анализ требований

Одной из важных задач является проверка корректности спецификаций по обозначенным в приложении 2.1.1 критериям. Проверка должна начинаться уже на этапе разработки спецификаций требований, поскольку при продвижении по ЖЦ разработки ПО стоимость ошибок и неверно определенных требований существенно возрастает [165]. В связи с этим существует этап анализа требований, включающий процедуры верификации и валидации [128, 238, ISO/IEC 12207]. Несмотря на то, что зачастую эти процедуры сводятся к проведению, в лучшем случае, автоматических тестов, когда ПО уже готово, существуют направления исследований, связанных с автоматизацией *верификации* моделей (*квалиметрией* моделей) при условии, что имеет место формализация спецификаций [23, 137, 184]. Общеизвестным средством решения слабо формализуемых задач, к которым относится инженерия требований, являются методы, развиваемые в рамках ИИТ. Кроме того, в [73] отмечается, что верификация может быть использована и для согласования взаимосвязанных спецификаций. В работе [167] предлагается согласовывать требования с помощью онтологического моделирования.

2.2. Методы разработки и жизненный цикл информационных систем

2.2.1. Модели жизненного цикла разработки программного обеспечения

В общем случае, ЖЦ определяется некоторой моделью и описывается в форме, задаваемой некоторой методологией [128, 238]. При этом разделяют ИС, как программную реализацию, и АС, как совокупность аппаратной и программной реализации (см. определения П. 2.2, П. 2.3). В рамках работы будем говорить об ИС. Модель ЖЦ определяет концептуальный взгляд на его организацию, принципы перехода между фазами ЖЦ. Методология в контексте

программной инженерии задает комплекс работ, их содержание и ролевую ответственность специалистов на всех этапах выбранной модели ЖЦ, также определяет саму модель и рекомендует практики, позволяющие максимально эффективно воспользоваться соответствующей методологии моделью (см. определение П. 2.4).

Определение П. 2.2. Модель ЖЦ ПО – структура, состоящая из процессов, работ и задач, включающих в себя разработку, эксплуатацию и сопровождение программного продукта, охватывающая жизнь системы от установления требований к ней до прекращения ее использования [128, 238, ГОСТ 12207].

Определение П. 2.3. Жизненный цикл АС – совокупность взаимосвязанных процессов создания и последовательного изменения состояния АС, от формирования исходных требований к ней до окончания эксплуатации и утилизации комплекса средств автоматизации АС [128, 238, ГОСТ 34.601-90].

Как уже ранее было отмечено, был разработан стандарт разработки ПО ISO/IEC12207 «Software Life Cycle Processes» (англ., процессы ЖЦ ПО) на основе SWEBOOK. Данный стандарт определяет ЖЦ как структуру декомпозиции работ. Модель согласно стандарту определяет организацию последовательности работ. Совокупность моделей, процессов, техник и организации задается методологией. Стандарт описывает 17 процессов ЖЦ. (см. таблицу П. 2.4) [128, 238].

Таблица П. 2.4. Процессы ЖЦ ПО согласно SWEBOOK

Тип	Процесс	Описание	
Основные процессы ЖЦ	Заказ		
	Поставка		
	Разработка	Определение процесса (подготовка процесса)	
		Анализ системных требований (анализ требований к системе)	
		Проектирование системы (проектирование системной архитектуры)	
		Анализ программных требований (анализ требований к программным средствам)	
		Проектирование программной архитектуры	
Детальное проектирование программной системы (техническое проектирование программных средств)			

		Кодирование и тестирование (программирование и тестирование программных средств)
		Интеграция программной системы (сборка программных средств)
		Квалификационные испытания программных средств
		Интеграция системы в целом (сборка системы)
		Квалификационные испытания системы
		Установка (ввод в действие)
		Обеспечение приемки программных средств
	Эксплуатация	Процесс разработки определяет рабочие задачи оператора службы поддержки и включает следующие работы: определение процесса, операционное тестирование, эксплуатация системы, поддержка пользователя.
	Сопровождение	Процесс включает следующие работы: определение процесса, анализ проблем и изменений, внесение изменений, проверка и приемка при сопровождении, миграция, вывод программной системы из эксплуатации.
Вспомогательн. процессы ЖЦ	Документирование	
	Управление конфигурацией	
	Обеспечение качества	
	Верификация	
	Аттестация	
	Совместный анализ	
	Аудит	
	Решение проблем	
Организац. процессы ЖЦ	Управление	
	Создание инфраструктуры	
	Усовершенствование	
	Обучение	

В то же время применительно к моделям ЖЦ ПО упоминают следующие: каскадная (водопадная), эволюционная, спиральная (модель Боэма) [128, 238]. При детальном изучении этих моделей, можно отметить, что все они акцентируют внимание на максимально возможное снижение количества ошибок в конечном ПО, как с точки зрения ошибок в реализации алгоритмов, так и с точки зрения несоответствия исходно поставленной задаче. Предлагается тестировать ПО, итерационно устранять ошибки, создавать n-ое количество прототипов, последовательно приближаясь к конечной точке разработки ПО. При этом приходится возвращаться к документации, спецификациям, моделям, и, что самое важное, к исходному коду программы. Это значительно

усложняет, замедляет и снижает эффективность процесса разработки ПО. Возникающие ошибки зачастую связаны с тем, что участники проектирования взаимодействуют друг с другом по принципу «испорченного телефона». Например, в пятерке «заказчик – бизнес-аналитик – руководитель проекта – программист – тестировщик» все указанные представители имеют совершенно разную специализацию и тезаурус, и, как следствие, в ходе их диалога происходит обеднение и искажение информации о решаемой задаче. В связи с этим видится перспективным такой автоматизированный процесс разработки ПО, при котором (как представлено в таблице П. 2.3) каждый из представителей, оперируя своими терминами и знаниями, создает собственную спецификацию, уточняя то, что известно ему, с помощью некоторой формальной системы на ЭВМ, языковые средства которой по своим синтаксическим и семантическим свойствам ориентированы в точности на описываемые им аспекты ПО. При этом создаваемые спецификации верифицируются в автоматическом режиме как в отдельности, так и по их совокупности в соответствии с рассматриваемыми ранее критериями, а код генерируется автоматически. Этот подход позволил бы существенно сократить количество ошибок и сохранить высокую степень точности описания решаемой задачи, а значит и качества ПО. В SWEBOOK уже отмечены предпосылки, говорящие о необходимости создания такой системы.

Дерево процессов ЖЦ представляет собой структуру декомпозиции ЖЦ на соответствующие процессы (группы процессов). Декомпозиция процессов строится на основе двух важнейших принципов, определяющих правила разбиения ЖЦ на составляющие процессы [128, 238].

– Модульность:

- задачи в процессе являются функционально связанными;
- связь между процессами – минимальна;
- если функция используется более, чем одним процессом, она сама является процессом;

- если процесс Y используется процессом X и только им, значит процесс Y принадлежит (является его частью или его задачей) процессу X , за исключением случаев потенциального использования процесса Y в других процессах в будущем.
- Ответственность:
- каждый процесс находится под ответственностью конкретного лица (управляется и/или контролируется им), определенного для заданного ЖЦ, например, в виде роли в проектной команде;
 - функция, чьи части находятся в компетенции различных лиц, не может рассматриваться как самостоятельный процесс;
 - общая иерархия (декомпозиция) составных элементов ЖЦ по порядку описывается следующим образом: группа процессов, процессы, работы, задачи.

Такое описание с позиций решаемой задачи говорит о наличии следующих спецификаций: описание словаря ПрО, как совокупности спецификаций источников, данные которых подлежат интеграции, описание совокупности моделей бизнес-процессов и процессов решения взаимосвязанных аналитических задач и порядка их взаимодействия, а также описание функциональных моделей, конкретизирующих вышеуказанные процессы с точек зрения порядка сбора данных (интеграционных процессов) и их обработки (алгоритмов вычисления). При этом, как уже было отмечено, такие спецификации должны быть верифицируемы относительно их корректности и согласованности на основе некоторого формального аппарата.

2.2.2. Методы разработки информационных систем

При разработке ПО в соответствии с той или иной моделью ЖЦ используют различные методологии, определяющие порядок разработки системы и предлагающие в своем составе различные методы (см. определения *П. 2.4*, *П. 2.5*). К числу получивших распространение можно отнести следующие методологии [128, 202, 203, 238]:

- Rational Unified Process (RUP);
- Enterprise Unified Process (EUP);
- Microsoft Solutions Framework (MSF);
- Agile-практики (eXtreme Programming (XP), Feature Driven Development (FDD), Dynamic Systems Development Method (DSDM), SCRUM и др.);
- ООП.

Определение П. 2.4. Методология в программной инженерии – это набор методов, применяемых в процессе всего ЖЦ создания ПО и объединенных единой философской концепцией [30, 225].

Определение П. 2.5. Метод с позиции программной инженерии – это четко определенный процесс создания набора моделей с помощью ясно специфицированных нотаций; эти модели описывают различные аспекты ПО [30, 202, 225].

Методы используют различные представления – от текстового описания и диаграмм, до строгой математики. Методы могут быть ориентированы на представление структур (структурные модели), объектов (объектно-ориентированные модели), функций (функциональные модели), логических выражений (логические модели), формальных выражений (формальные модели), смешанных представлений (структурно-функциональные, структурно-логические и др.) [202]. Во многом, выбор того или иного представления зависит от возможной степени формализации рассматриваемой проблемы. В этом смысле их разделяют на хорошо и слабо структурированные [103]. Хорошо структурированные – те, в которых существенные зависимости выяснены настолько, что могут быть выражены в числах или символах, получающих в конце концов численные оценки. Слабо структурированные – те, которые содержат как качественные, так и количественные элементы, причем качественные, малоизвестные и неопределенные стороны проблем имеют тенденцию доминировать. В том случае, когда проблема носит характер слабо структурированной, говорят о необходимости использования смешанных моделей и использования ИИТ (см. приложение 2.3). Существенно и то, что выбор метода зависит от

категориального аппарата, который удобен для описания ПрО. Например, удобно ли представление в виде совокупности объектов или функций.

К числу известных методов можно отнести диаграммы ДеМарко, Иордона, Румбаха, Шлера и Меллора, метод Буча, метод контролируемого формулирования требований (Controlled Requirements Expression (CORE)), метод структурного анализа и проектирования (Structured Analysis and Design Technique (SADT)), метод определения требований, основанное на перспективах (Viewpoint-oriented Requirement Definition (VORD)), метод Real-Time Object-Oriented Modeling (ROOM), [30, 127, 202, 225, 253, 267].

В последние несколько десятков лет особой популярностью пользуется ООП (метод Буча) в качестве концептуальной основы при проектировании ПО [30, 225, 253]. В основе метода Буча лежит возможность рассматривать разрабатываемую систему с разных точек зрения. Точка зрения – это определенный способ видения системы, исходя из которого строится определенная модель системы. Точка зрения включает в себя набор графических нотаций и их семантику. Метод получил воплощение в стандартизованном языке UML в результате наработок комитета OMG. «UML – это стандартный инструмент для разработки «чертежей» ПО. Его можно использовать для визуализации, спецификации, конструирования и документирования артефактов программных систем» [30]. UML объединяет большое количество различных графических нотаций и представляет собой целую совокупность различных диаграмм, предназначенных для отражения тех или иных точек зрения. К ним относятся диаграммы: классов, интерфейсов, пакетов, объектов, компонент, вариантов использования, взаимодействия, деятельности (активностей), состояний, артефактов, размещения. Для ряда диаграмм предлагается генерация элементов кода на объектно-ориентированных языках, что позволяет частично сократить объем работы программистов.

Важность появления ООП может быть описана через разницу императивного (алгоритмического) и объектно-ориентированного подходов [30]. В императивном программировании основным строительным блоком является

процедура или функция, и внимание разработчика сконцентрировано на вопросах передачи управления процедурам и декомпозиции больших алгоритмов на меньшие. Системы на основе алгоритмов тяжело адаптируемы при изменении требований и сопровождении. В ООП в качестве строительного блока выступает объект или класс. В самом общем смысле объект – это сущность, обычно извлекаемая из словаря ПрО, а класс – описание множества однотипных объектов. Каждый объект идентифицируем по своему наименованию, состоянию (по данным) и поведению (с ним можно что-то делать или он сам может что-то делать с другими объектами). Такое представление гораздо ближе к человеческому пониманию и позволяет более гибко оперировать классами и объектами при проектировании сложных систем.

Анализ рассмотренных методов показывает, что все они основываются на использовании перспектив (точек зрения), поскольку предполагают, что требования к системе не должны рассматриваться только с одной стороны. По существу предлагается два основных типа перспектив [202]:

- точки зрения заинтересованных сторон;
- точки зрения экспертов, обладающих знаниями в ПрО.

Сформулированные выводы коррелируют с рассмотренными ранее утверждениями о необходимости разноаспектного моделирования (специфирования) систем. При этом ООП должна быть в основе такого описания. В то же время можно отметить, что многие существующие методы недостаточно формализованы и используются в большей степени аналитиками для визуального описания без прямой привязки к созданию ПО, а для предлагаемых диаграмм, характеризующих точки зрения, не вполне ясно, как их соотносить между собой, чтобы получить согласованную систему спецификаций. Кроме того, можно отметить, что зачастую характерна подмена понятий концептуальной модели ПрО и концептуальной модели программной системы. Например, UML предназначен для спецификации программных систем. Перспективной видится возможность согласовать между собой концептуальные модели

ПрО и ПО, так чтобы обеспечить наличие взаимосвязанной совокупности моделей, согласующих представление о ПрО и ПО в соответствии с обозначенным в разделе 1.3.1. Такие научно обоснованные попытки уже совершались: вычислительные модели Э.Х. Тыгу, Н-модели Нариньяни А.С., модели СР-L проекта СР, G-модели Охтилева М.Ю. и др. [73, 120, 132, 193] (см. приложение 2.3).

2.2.3. Автоматизированное проектирование программного обеспечения

Проектирование, при котором все проектные решения или их часть получают путем взаимодействия человека и ЭВМ, называют автоматизированным, в отличие от ручного (без использования ЭВМ) или автоматического (без участия человека на промежуточных этапах) [127]. Система, реализующая автоматизированное проектирование, представляет собой САПР. Отдельный вид САПР – это системы автоматизированной разработки ПО. Их название трактуют двояко – Computer Aided System Engineering и Computer Aided Software Engineering. *CASE-системы* часто отождествляют с инструментальными средами разработки ПО, называемыми средами быстрой разработки приложений (RAD — Rapid Application Development), позволяющими существенно сократить труд программистов. Примерами известных инструментальных сред RAD являются Visual Basic, Delphi, PowerBuilder фирмы Microsoft, Borland, PowerSoft. В то же время они предполагают использование методов концептуального моделирования (см. раздел 1.3.2.2) и, соответственно, принципов системного моделирования (см. определение П. 2.6), анализа и синтеза моделей [127, 180, 222, 258]:

- моделирование решает две задачи: создание моделей сложных систем, анализ свойств систем на основе исследования их моделей (имитация);
- синтез также подразделяют на две задачи: синтез структуры проектируемых систем (структурный синтез), выбор численных значений параметров элементов систем (параметрический синтез).

Определение П. 2.6. Системное (комплексное) моделирование – полимодельное многокритериальное описание и исследование заданной ПрО с использованием комбинированных методов, алгоритмов и методик, позволяющих на конструктивной основе обеспечить эффект взаимного усиления достоинств каждой из применяемых моделей принадлежащих заданным классам, при этом для решаемой задачи необходимо проводить согласование применяемых моделей, а также согласование критериальных функций по выбору альтернатив [208].

Среди преимуществ моделирования в CASE-системах можно отметить следующие [202]:

- использование точно определенной терминологии, однозначность которой поддерживается в рамках разработки всей системы;
- наглядное представление системных спецификаций и архитектуры системы;
- рассмотрение различных аспектов взаимодействия системы с различных точек зрения;
- реализация системного подхода;
- возможность проверить достоверность некоторых аспектов поведения системы с помощью динамических моделей;
- синтез и совершенствование системы посредством уточнения архитектуры с поддержанием генерации исходного кода;
- позволяет свободно общаться различным организациям между собой, используя стандартные нотации.

При проектировании предполагается один из стилей проектирования: нисходящее, восходящее, эволюционное. В рассматриваемой задаче с учетом обозначенного в разделе 1.1 характерно использование принципов нисходящего проектирования на основе поэтапного уточнения описания ПрО и проектируемого ПО. Реализация нисходящего проектирования приводит к спиральной (или каскадной) модели разработки ПО, на каждом витке спирали блоки предыдущего уровня детализируются, используются обратные связи.

Верхний уровень проектирования АС (или ИС) называют *концептуальным проектированием* (см. раздел 1.3.2.2). Традиционно, считается, что его выполняют в процессе предпроектных исследований, формулировки ТЗ, разработки эскизного проекта и прототипирования [127]. При концептуальном проектировании создают спецификации (модели), о которых говорилось в приложении 2.1.1. Различают функциональные, информационные, поведенческие и структурные модели. Функциональная модель системы описывает совокупность выполняемых системой функций. Информационная модель отражает структуры данных – их состав и взаимосвязи. Поведенческая модель описывает информационные процессы (динамику функционирования), в ней фигурируют такие категории, как состояние системы, событие, переход из одного состояния в другое, условия перехода, последовательность событий, осуществляется привязка ко времени. Структурная модель характеризует состав подсистем, их взаимосвязи.

Исходя из назначения САПР, они предлагают средства формального синтеза проектных решений, выполняемого в автоматическом режиме [127]. Задачи синтеза структур проектируемых объектов относятся к трудно формализуемым. Существует ряд общих подходов к постановке этих задач, однако практическая реализация большинства из них не очевидна. Поэтому имеются лишь «островки» автоматического выполнения процедур синтеза среди «моря» проблем, ждущих автоматизации. Однако имеются и примеры успешной автоматизации структурного синтеза в ряде ПрО. Структурный синтез заключается в преобразовании описаний проектируемого объекта: исходное описание содержит информацию о требованиях к свойствам объекта, об условиях его функционирования, ограничениях на элементный состав и т. п., а результирующее описание должно содержать сведения о структуре, т. е. о составе элементов и способах их соединения и взаимодействия. Исходное описание, как правило, представляет собой ТЗ на проектирование, описание на некотором формальном языке, являющимся входным для САПР.

Таким образом, задачу структурного синтеза проектных решений представляют как множество элементов и совокупность правил объединения этих элементов в допустимые структуры (проектные решения) [127]:

$$ЗС = \langle \mathcal{A}, \mathcal{HT}, \mathcal{AK}, \mathcal{P} \rangle, \quad \text{П.}(2.1)$$

где \mathcal{A} — алфавит исчисления (алфавит представлен базовыми элементами, из которых синтезируется структура); \mathcal{HT} — множество букв, не совпадающих с буквами алфавита \mathcal{A} и служащих для обозначения переменных; \mathcal{AK} — множество аксиом исчисления, под которыми понимаются задаваемые исходные формулы (слова) в алфавите \mathcal{A} (например, соответствия функций и элементов); \mathcal{P} — множество правил вывода новых формул в алфавите \mathcal{A} из аксиом и ранее выведенных корректных формул. Каждую формулу можно интерпретировать как некоторую структуру, поэтому синтез — это процесс вывода формулы, удовлетворяющей исходным требованиям и ограничениям. В случае, когда не существует аналитического решения этой задачи используют методы ИИТ. Тогда \mathcal{A} есть БД, \mathcal{P} — БЗ. Так, например, могут быть использованы морфологические таблицы, И-ИЛИ графы и деревья, системы логического исчисления и др.

Согласно [73], CASE-система предполагает необходимость определения некоторой цели в ПрО, декомпозиции её на задачи, а для получения программы — специфицирования получившихся задач. «Желательно, чтобы система поддерживала формирование замысла и составление спецификации задач, а выбор подходящей структуры данных и алгоритма полностью брала на себя, оставляя нам совещательную функцию. Весь процесс разработки можно представить как процесс пошагового символического воплощения замысла». Автор [73] конкретизирует наличие у системы возможностей, идеи которые уже упоминались ранее (см. рис. П. 2.1). При этом эффективность воплощения зависит от выразительности языковых средств, использования текста или графики, других выразительных возможностей. Известно, что графические образы ближе человеку, в связи с этим используются различные графические нотации.

CASE-системы могут быть использованы для задач интеграции, упоминавшихся в разделе 1.2. В работе [127] отмечается, что синтаксические и семантические аспекты интеграции базируются на унифицированных языках и форматах данных, технологиях типа Open Data Base Connectivity (ODBC) для доступа к общему банку данных.

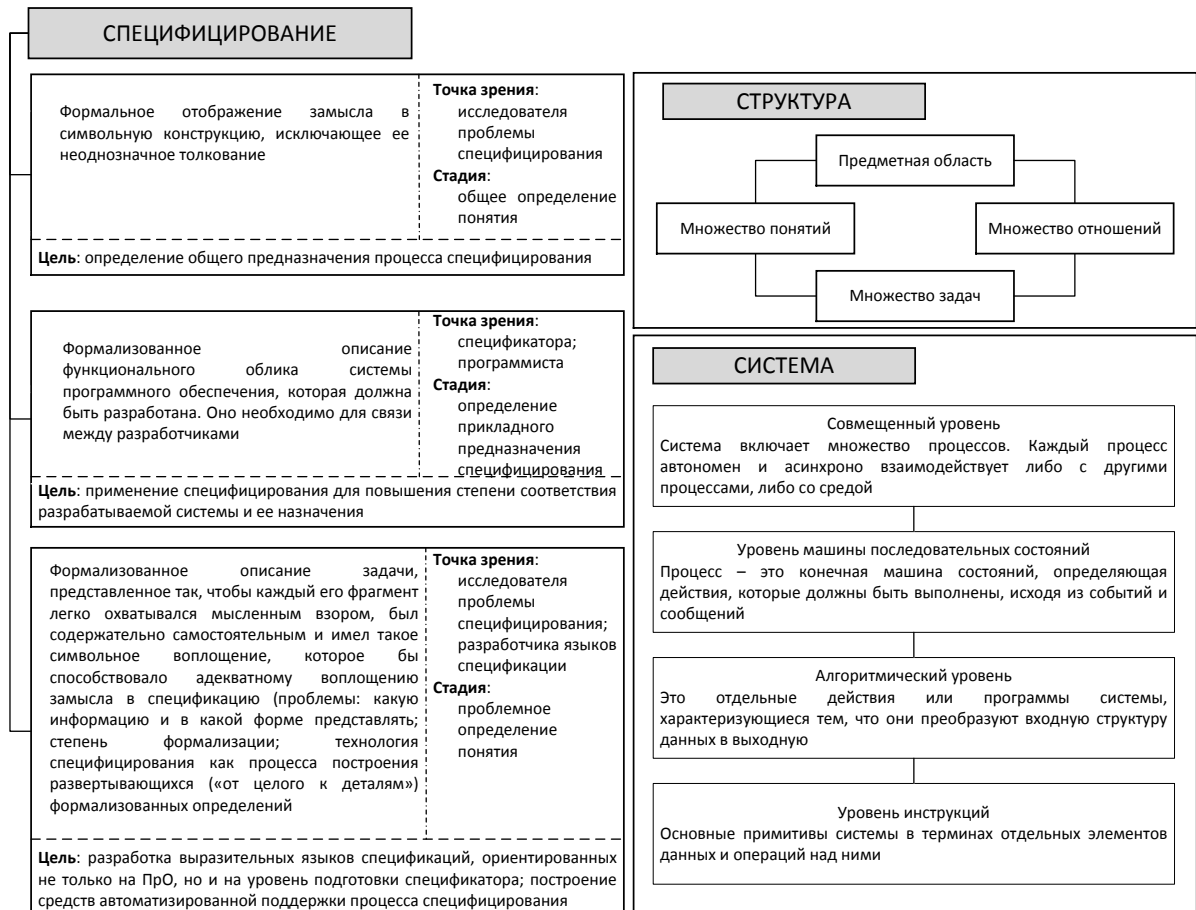


Рис. П. 2.1. Описательные возможности CASE-систем

Семантическая интеграция подразумевает автоматическое распознавание разными системами смысла передаваемых между ними данных. Теоретической базой для создания технологий интеграции ПО в САПР являются: объектно-ориентированная методология, методология автоматизированного проектирования, в соответствии с которой осуществляются типизация проектных процедур и маршрутов проектирования в различных ПрО, выявление типичных входных и выходных данных процедур, построение информационных моделей приложений и их обобщение. При этом могут быть использованы такие

технологии, как Integrated Computer Aided Manufacturing (IDEF) и её модели IDEF0, IDEF3 (функциональные), IDEF1X и UML для описания информационных моделей, некоторые UML-диаграммы для описания поведения и функционирования и др. [127, 170, 205]. Многие PDM-системы создавались фирмами, первоначально специализировавшимися на разработке САПР. В качестве примеров можно привести Optegra и ProPDM (PTC), iMAN (EDS), WorkCenter (Autodesk), BaanPDM (BAAN), Euclid Design Manager (Matra Datavision) и др. [127].

Таким образом, понятие CASE-систем является важным для рассматриваемой задачи. С одной стороны, отмечается необходимость использовать нисходящую модель проектирования и решать задачу формального вывода проектного решения, представляющего собой совокупность спецификаций и исходный код, сгенерированный на их основе. С другой, в результате анализа было выявлено, что перспективные CASE-системы характеризуются возможностью проектирования трёх типов формальных моделей, предполагающих иерархическую взаимосвязь: информационную модель как описание словаря ПрО, поведенческую модель как спецификацию отношений между элементами словаря, функциональную модель как спецификацию решаемых задач.

2.3. Интеллектуальные технологии в программной инженерии

2.3.1. Классификация моделей по уровням представления знаний

Любую модель можно представить как совокупность носителя модели и ее сигнатуры (см. (2.5)) [69, 133].

Множество отношений модели образует ее сигнатуру. Если классифицировать модели по типу носителя, то все модели подразделяются по их ПрО (физика, химия, электроника и т.д.). Если же в качестве критерия классификации выступает тип сигнатуры, то модели классифицируются по эпистемологическим (познавательным) уровням – уровням представления знаний (см. таблицу П. 2.5) [132, 133].

Таблица П. 2.5. Классификация уровней представления знаний и их моделей

№ п/п	Уровень знаний	Наименование модели	Модель	Носитель модели	Сигнатура модели
0.	Исходная система	Модель данных объекта	$B = \langle A; R^k \rangle$	Словарь свойств объектов (множество атрибутов) $A = \{a_1, \dots, a_k\}$	Схема отношения (содержание понятия) $R^k \subset A \times \dots \times A$ $R^k = \{a_{i1}, \dots, a_{ik}\}$ Ключ отношения $K^p \subseteq R^r, p \leq k$
1.	Система данных	Множество экземпляров объектов	$E = \langle D_1, \dots, D_k; r^k \rangle$	Домены атрибутов (допустимые множества значений атрибутов) $D_i = \text{dom}(a_i)$	Отношение со схемой R^k $r^k \subset D_1 \times \dots \times D_k$ Множество кортежей (денотатов понятия) $r^k = (d_1^j, \dots, d_k^j)$
2.	Порождающая система	Метод объекта:	$P = \langle V_1, \dots, V_r; F \rangle$	Свойства объекта и переменные метода	Функциональное отображение $F \subset V_1 \times \dots \times V_r$
		1. Математическая негеометрическая модель		Числовые и переменные, функции, мат. и лог. отношения, операторы и операции	Системы уравнений и неравенств
		2. Геометрическая модель		Точки, линии, поверхности, тела	Объединения, пересечения, вычитания, преобразования, расчеты параметров
		3. Модель экспертных знаний (продукционное правило)		Числовые и нечисловые переменные: Входные $V_i u(i) = 0,$ Выходные $V_i u(i) = 1,$ где $u(i)$ – определитель вход-выход	Отображение входных переменных в выходные
3.	Структурированная система:				
	Исходная	Мультиагентная система (сеть агентов)	$SB = \langle B, A; C \rangle$	Объекты и их свойства $B = \{b_1, \dots, b_l\},$ $A = \{a_1, \dots, a_n\}$	Семантическая сеть объектов $C \subset B \times \dots \times A$
	данных	База данных (реляционная)	$SD = \langle B, G; J \rangle$	Объекты и мощность отношений между ними $B = \{b_1, \dots, b_l\},$ $G = \{O, P, Z, N\}$	Структура базы данных $J \subset B \times B \times G$
	Порождающая	База знаний (метод объекта)	$SP = \langle F, V; K \rangle$	Правила и переменные модели $F = \{f_1, \dots, f_l\},$ $V = \{v_1, \dots, v_n\}$	Семантическая сеть правил $K \subset F \times F \times V \times \dots \times V$
4.	Метасистема	Модель метасистемы (И-ИЛИ граф)	$MB = \langle B, M; L \rangle$	Объекты и типы связей между ними $B = \{b_1, \dots, b_l\}$ $M = \{AND, OR\}$	И-ИЛИ граф $L \subset B \times B \times M$

Самый нижний уровень в представленной классификационной схеме, обозначаемый как уровень 0, это система, различаемая КП как таковая. На

этом уровне исследуемая система определяется через множество свойств и носит название *исходная система*. Другими словами, на уровне 0 рассматриваются свойства исследуемой или проектируемой системы [132, 133].

На более высоких эпистемологических уровнях фиксируются отличия относительно переменных соответствующей исходной системы. В системах более высоких уровней используются все знания систем более низких уровней и, кроме того, содержатся дополнительные знания, недоступные низшим уровням [132, 133].

После того, как исходная система дополнена данными, т.е. фактическими состояниями (значениями) переменных, рассматривают новую систему (исходную систему с данными) как определенную на эпистемологическом уровне 1. Системы этого уровня получили название *системами данных*.

Уровень 2 представляет собой уровень БЗ генерации значений переменных, определяющих свойства исходной системы. На этом уровне задаются функциональные связи переменных, в число которых входят переменные, определяемые соответствующей исходной системой и, возможно, некоторые дополнительные. Правило преобразования БЗ на этом уровне обычно представляет собой однозначную функцию, сопоставляющую каждому элементу множества переменных, рассматриваемых в этом правиле в качестве выходного, единственное значение из множества допустимых [132, 133].

Поскольку задачей генерации свойств является реализация процесса, при котором состояния (значения) переменных могут порождаться по множеству параметров при определенных начальных или граничных условиях, системы уровня 2 называются *порождающими системами* (generative system).

На эпистемологическом уровне 3 исследуемые системы, определенные как порождающие или системы более низкого уровня, называются *подсистемами* общей системы. Эти подсистемы могут соединяться в том смысле, что они имеют некоторые общие переменные. Системы этого уровня называются *структурированными системами* (structured system) [132, 133].

На эпистемологических уровнях 4 и выше исследуемые системы состоят из набора систем, определенных на уровнях 1, 2 или 3, и некоторой *метахарактеристики* (правила, отношения, процедуры), описывающей замены в системах более низкого уровня. Это уровни, необходимые для формирования концептуальных И-ИЛИ графов [132, 133].

Модели первых двух уровней достаточно хорошо изучены в теории реляционных БД. Исходная система предназначается для описания некоторого объекта или эквивалентного ему понятия. Носителем модели являются атрибуты различных типов, неструктурированное множество которых с описанием идентификаторов, типов и наименований на языке деловой прозы образуют словарь. Сигнатурой модели уровня 0 является схема реляционного отношения, определяющего объект или содержание понятия.

Система данных представляет собой некоторое множество экземпляров объекта или выборку из объема понятия, определяемого исходной системой. Носителем модели уровня 1 являются домены атрибутов, а сигнатуру представляет реляционное отношение со схемой, описанной на уровне 0.

Порождающая система связана с методом объекта, определяющим его поведение. В зависимости от типа моделей можно выделить математические модели негеометрического, геометрического и экспертного типов. Носителем негеометрической модели являются числовые константы и переменные, функции, математические и логические операторы, отношения и операторы. Сигнатуру таких моделей образуют системы уравнений и неравенств. Экспертные модели отличаются от математических тем, что в качестве носителя, помимо числовых, могут появляться и нечисловые переменные. Сигнатура таких моделей строится обычно на базе, на основании которых при определенных условиях входные переменные отображаются в выходные [132, 133].

Описанную иерархию моделей можно связать в единую технологическую систему множество различных пакетов расчетных программ посредством интерфейсных модулей. Однако такие попытки обречены на неудачу,

поскольку такая «технология» есть не что иное, как «латание дыр» или «лоскутное одеяло». Уровень порождающих систем (уровень 2) является основной ареной борьбы моделей с алгоритмами. До недавнего времени казалось, что у прикладного специалиста – КП – нет другой возможности, кроме как написать нужную ему модель, т.е. систему уравнений или неравенств, и искать в вычислительной математике подходящий алгоритм решения по ней. Не найдя такого, подбирать аппроксимацию своей модели другой, для которой алгоритм есть [132, 133].

Представителями альтернативного, гармоничного построения таких иерархических систем на единой методологической основе являются, в частности, интегрированный программный продукт СПРУТ, интеллектуальный решатель математических задач UniCalc, которые позволяют исключить разработку алгоритма из деятельности прикладного специалиста [132, 133]. При этом форма представления знаний (метод объекта – см. таблицу П. 2.5) может иметь произвольную форму.

2.3.2. Интеллектуальные системы

Системы, ядром которых является БЗ (см. определение П. 2.7) или модель ПрО, описанная на проблемно-ориентированном языке представления знаний (ЯПЗ) так называемого сверхвысокого уровня, называют интеллектуальными [41, 42, 45, 60, 106, 169]. Чаще всего интеллектуальные системы применяются для решения сложных задач, где основная трудность решения связана с использованием слабоформализованных знаний специалистов-практиков и где логическая (или смысловая) обработка информации превалирует над вычислительной. Кроме того, важной целью ставится освобождение специалистов от рутинной работы в пользу большей ориентации на творческую. Изучение методов проектирования таких систем находится в поле теории ИИ (или инженерии знаний) и предполагает разработку моделей и методов извлечения, структурирования формализации (представления) знаний для их обработки в интеллектуальных ИС. В России в качестве известных исследователей в области теории ИИ можно назвать таких ученых, как М. Г. Гаазе-Рапопорт, Д.А.

Поспелов, Ю. И. Журавлев, А. С. Нариньяни, Э.Х. Тыугу, В. И. Городецкий, В. Ф. Хорошевский, Т.А. Гаврилова и др.

Определение П. 2.7. База знаний — семантическая модель, предназначенная для представления в ЭВМ знаний, накопленных человеком в определенной области [41].

Важность этой теории с позиций ИТ определяется возможностью извлечения и формализации знаний для их автоматического применения в различных прикладных задачах, тиражирования, проверки на полноту и непротиворечивость. Существенна цепочка «данные – информация – знания», предполагающая, во-первых, необходимость определения семантики (смысла) на данных на основе интерпретации синтаксических выражений, а во-вторых, позволяющая говорить о гносеологической цепочке представления знаний при выборе языков моделирования: факт, обобщенный факт, эмпирический закон, теоретический закон (см. приложение 2.3.1) [41, 164]. Так, например, Э.Х. Тыугу говорил о программировании, как о методе доказательства теорем на основе предположения, что синтаксис языка программирования предопределяет возможности по описанию решения математических задач, а корректный и завершаемый процесс вычислений есть непосредственно вывод (доказательство) теоремы [193].

В процессе наполнения БЗ возникают задачи, связанные с извлечением эмпирического знания, которое характеризуется [41]:

- модальностью – определением знаний в различных категориях;
- противоречивостью – диалектичность знаний;
- неполнотой – невозможностью полного описания ПрО.

Эта ситуация требует привлечения различных методов *верификации* проектируемых моделей, что коррелирует с выводами, сделанными в разделах 1.3.1, 1.3.2. Кроме того, существенную роль играют те языковые средства, которые используются при концептуальном проектировании. Если в их основе лежит формальный аппарат, который в рамках ИИТ называют МПЗ, допол-

ненной соответствующей графической нотацией, то это позволяет в определенной мере объективизировать знания экспертов для решения задач, которые ставятся перед создаваемой системой. Графические представления, в свою очередь, как метод компактной организации информации могут использоваться как специализированные категорийные нотации для профессиональной интеллектуальной деятельности, включая моделирование предприятий, сложных организационных структур, рынка, бизнес-процессов [41]. Одним из проблемных вопросов в использовании графических языков является семантическая *интероперабельность* – различные пользователи могут по-разному интерпретировать значения символов в нотациях, различные языки моделирования могут иметь элементы со схожими названиями, но с разным смыслом, а также, наоборот, при схожести смысла понятий могут различаться по формулировкам. Для устранения данной проблемы используются унифицированные метамоделли (или онтологии). Визуальные языки хорошо дополняют технологии онтологического (или концептуального) моделирования, делая содержимое онтологий понятным и наглядным для экспертов и аналитиков. Кроме того, для решения сложных задач обычно нужна не отдельная диаграмма, а комплекс диаграмм, охватывающих разные типы знаний.

Системы ИИ основаны на знаниях, отделенных от процедурной части программ и представленных в одной из характерных для них форм [40, 60, 127]. Реально функционирующие современные CASE-системы с БЗ чаще всего относятся к классу ЭС. ЭС является типичной системой ИИ, которая реализует нисходящую модель проектирования, основанную на создании концептуальных моделей на основе извлечения экспертных знаний. Трудности формализации процедур структурного синтеза привели к популярности применения экспертных систем в САПР, поскольку в них вместо выполнения синтеза на базе формальных математических методов осуществляется синтез на основе опыта и эвристических рекомендаций, полученных от экспертов.

Предпосылки к созданию такой ЭС, которая предполагает синтез программ, связаны с наличием проблем в диалоге эксперта и программиста: различные тезаурусы эксперта и программиста, ограниченный доступ программиста к информации, избыточное количество организационных аспектов между экспертом и программистом, человеческий фактор, слабая формализованность знаний эксперта (эмпирические знания) [106]. Можно выделить следующий список общих задач ЭС: интерпретация, прогнозирование, диагностика, проектирование, планирование, мониторинг, инструктирование, управление [41, 42].

Типовую ЭС можно представить следующим образом [164] (см. рис. П. 2.2):

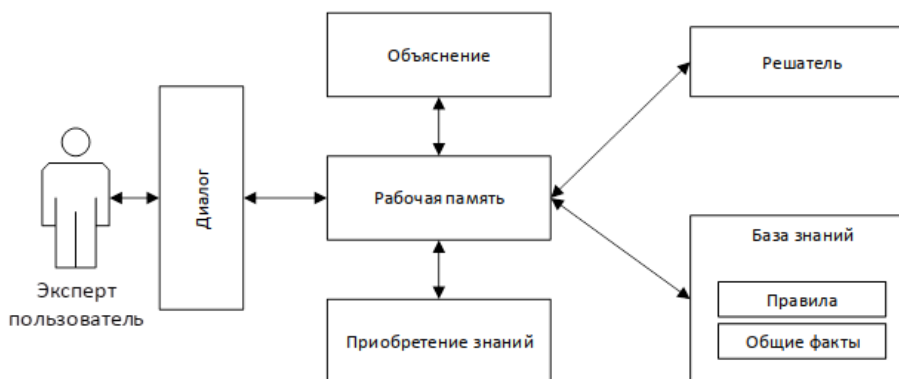


Рис. П. 2.2. Типовая структура экспертной системы

Важными элементами ЭС являются БЗ и решатель. При этом в рамках БЗ определен набор правил, а решатель необходим для применения того или иного правила к поступающим входным данным [169]. Возможные синтаксические утверждения в составе БЗ существенно зависят от используемой МПЗ и механизмов вывода. При этом процесс наполнения БЗ может предполагать следующий набор действий (см. таблицу П. 2.6) [169].

Таблица П. 2.6. Типовые этапы наполнения базы знаний

№	Этап	Содержательное пояснение
1	Идентификация знаний	Инженер по знаниям должен очертить круг вопросов, которые должна поддерживать БЗ, и виды фактов, которые будут доступными применительно к каждому конкретному экземпляру задачи.

2	Сбор относящихся к делу (релевантных) знаний	Инженер по знаниям может уже быть экспертом в рассматриваемой проблемной области или ему может потребоваться общаться с настоящими экспертами для выявления всего, что они знают — извлечение знаний. На этом этапе знания еще не представлены формально.
3	Определение словаря ПрО	В иной формулировке этот этап можно определить как преобразование важных понятий уровня проблемной области в имена формального уровня. Результатом выбора наиболее подходящих средств представления становится словарь, известный как онтология проблемной области.
4	Регистрация общих знаний о проблемной области	Инженер по знаниям записывает аксиомы для всех терминов словаря. Тем самым он закрепляет (в возможной степени) смысл этих терминов, позволяя эксперту проверить их содержание. На этом этапе часто обнаруживаются неправильные трактовки или пропуски в словаре, который необходимо исправить, возвратившись на этап 3 и снова пройдя данную итерацию в текущем процессе проектирования.
5	Составление описания данного конкретного экземпляра задачи	Этап сводится к написанию простых атомарных высказываний об экземплярах понятий, которые уже являются частью онтологии. Для систем ИИ определения экземпляров задачи поставляются «датчиками» и БЗ снабжается дополнительными высказываниями таким же образом, как традиционные программы снабжаются входными данными.
6	Передача запросов процедуре логического или формального вывода и получение ответов	На этом этапе применяется процедура логического или формального вывода к аксиомам и фактам о конкретной задаче для получения фактов, которые составляют цель функционирования системы.

Таким образом, в рамках рассматриваемой задачи можно говорить, что СИАП ЖЦ СОТО необходимо относить к категории интеллектуальных систем, предполагающих функционирование на основе БЗ. При этом, как указывалось ранее, наполнение БЗ должно осуществляться пользователями разных специализаций, с одной стороны, а с другой, должны использоваться разнотипные модели как с точки зрения иерархии спецификаций, так и с точки зрения разнообразия решаемых аналитических задач. В связи с этим требуется использование нескольких МПЗ, ориентированных на поведенческие, функциональные и информационные аспекты ПрО, причем можно подчеркнуть необходимость использования такого единого формального аппарата в рамках этих МПЗ, который позволит согласовать совокупность формируемых экспертами моделей. Системный анализ существующих МПЗ приведен в приложении 2.3.3. Результаты анализа показали, что *G-модель* как комбинированная декларативно-процедуральная МПЗ обладает рядом преимуществ по сравнению с другими моделями при решении задачи автоматизированного анализа

СОТО. В то же время её выразительных возможностей недостаточно для моделирования бизнес-процессов и спецификаций интеграции данных и приложений в рамках задач информационно-аналитической поддержки ЖЦ СОТО.

2.3.3. Модели представления знаний в теории искусственного интеллекта

В рамках работы, с позиции системного подхода, ставится задача синтеза интеллектуальной СИАП ЖЦ СОТО, позволяющей осуществить формальное описание структуры и состава СОТО и характера его функционирования. Одной из задач при проектировании СИАП является задача выбора и обоснования выбора МПЗ [147, 153].

Выделяют несколько видов МПЗ. В современной литературе приводится их обобщенная классификация [39] (см. рис. П. 2.3). Выбор той или иной МПЗ определяется чаще спецификой решаемой задачи. Однако на практике при моделировании ЭС, основанных на знаниях экспертов, целесообразно пользоваться классическими МПЗ (см. рис. П. 2.4) [60].



Рис. П. 2.3. Классификация моделей представления знаний



Рис. П. 2.4. Подходы к представлению знаний

МПЗ можно классифицировать двумя способами: с одной стороны, их делят на формальные (логические, формальные продукционные) и эвристические (фреймы, продукционные, семантические сети), с другой – на процедурные и декларативные.

Эвристические модели чаще выбирают при моделировании ЭС, поскольку их возможности ориентированы на моделирование тех или иных форм экспертных знаний.

Процедурные модели основаны на предпосылке, что интеллектуальная деятельность есть знание проблемной среды, вложенное в программы, то есть знание о том, как можно использовать те или иные сущности. Декларативные модели основаны на предпосылке, что знание неких сущностей («знать что») не имеет глубоких связей с процедурами, используемыми для обработки этих сущностей [60, 164]. При использовании декларативных моделей реализуется ООП, с одной стороны, с другой – повышается степень проблемной ориентированности ЯПЗ. [132, 133, 137].

На практике необходима реализация смешанных моделей (декларативно-процедуральных). Такой подход позволяет рассматривать объекты через их модели с декларативной точки зрения, а их поведение – через программы с императивной позиции. ООП реализует указанные принципы. Однако она является лишь промежуточным этапом на пути использования моделей в интеллектуальных технологиях. Другой декларативно-процедуральной моделью является G-модель [132, 133, 137].

В работах [60, 106, 132, 164] в рамках теории ИИ выделено 4 основных МПЗ.

Логические МПЗ:

$$M = \langle T, P, A, B \rangle, \quad \text{П.(2.2)}$$

где T – множество базовых элементов различной природы, P – множество синтаксических правил, A – множество аксиом, B – множество правил вывода.

Данная МПЗ оперирует фактами и утверждениями, представляемыми в виде логических высказываний на основе исчисления предикатов. Позволяет порождать логические выводы, оценивать истинность или ложность высказываний. Опираясь на теорию нечетких множеств, можно построить логическую модель для более широкого класса задач.

Сетевые МПЗ (иногда их называют семантические):

$$H = \langle I, C_1, C_2, \dots, C_n, \Gamma \rangle, \quad \text{П.(2.3)}$$

где I – множество информационных единиц, C_1, C_2, \dots, C_n – множество типов связей между информационными единицами, множество Γ задает между информационными единицами связи из заданного набора типов связей:

- классифицирующие сети (на основе отношений структуризации);
- функциональные сети (ВМ [193]);
- сценарии (на основе каузальных отношений).

Семантические сети являются одной из наиболее выразительных МПЗ, так как в них имеются средства для выполнения всех пяти требований, предъявляемых к знаниям: внутренняя интерпретируемость, структурированность,

связность, семантическая метрика, активность [164]. Но такая универсальность семантических сетей имеет и негативную сторону. Если допускать в них произвольные типы отношений и связей, не являющиеся отношениями в математическом смысле (например, ассоциативные связи), то сложность работы с таким образом организованной информацией резко возрастает. Поэтому вводятся ограничения на характер структур и типов информационных единиц, находящихся в вершинах семантической сети, и на характер связей, задаваемых ее дугами. Выделенная совокупность сущностей, понятий и ситуаций в сети ПрО называется ее состоянием [60]. Вершины сети могут быть различны и представлены событиями, фактами, суждениями, атрибутами, комплексами признаков, и даже процедурами (преобразование информации). Значения признаков, атрибутов предполагаются известными. Если вершины семантической сети обладают структурой, то такую сеть называют иерархической. В иерархической сети можно устанавливать отношения не только между элементами, но и между подсетями (пространствами, подсистемами, образами).

Производственные МПЗ – сеть производств, основанная на правилах вывода из логических моделей и описании знаний в виде семантической сети. Правила представляют собой форму: «Если ‘условие’ – то ‘действие’». Производственную модель можно представить кортежем:

$$N = \langle A, P, C, I \rangle, \quad \text{П.(2.4)}$$

где N – имя продукции, A – сфера применения продукции, P – условия применимости продукции (если P истинно, ядро продукции активизируется), C – ядро продукции, I – постусловия продукции, актуализирующиеся при реализации продукции (например, процедура).

БД в производственной МПЗ содержит факты и утверждения о рассматриваемой ПрО, в БЗ хранятся производственные правила. Механизмом вывода является интерпретатор правил (см. рис. П. 2.5). При этом задается имя продукции для выделения ее из списка производств. Производственные системы обладают свойством асинхронности и естественного параллелизма.

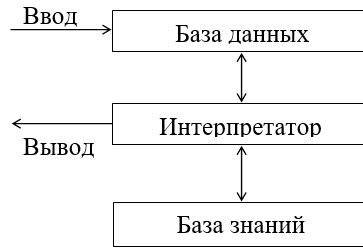


Рис. П. 2.5. Принцип функционирования продукционной системы

Фреймовые МПЗ, в которых фрейм представляется в виде жесткой информационные структуры, называемой протофреймом:

$$\begin{aligned} &Имя_слота_1(значение_слота_1) \\ &Имя_слота_2(значение_слота_2) \\ &..... \\ &Имя_слота_3(значение_слота_3) \end{aligned}, \quad \text{П.}(2.5)$$

Фрейм представляет собой структуру семантически связанных данных. По сути фрейм можно рассматривать как семантическую сеть. Верхний уровень фрейма представляет некоторое понятие, последующие уровни – терминальные слоты с их значениями. Значением слота может быть практически что угодно, в том числе ссылки на другие фреймы, а также процедуры, реализующиеся в виде алгоритмов в вычислительном процессе. Конкретизация фрейма может происходить постепенно с уточнением знаний об объекте. Заполненный фрейм называется фреймом-экземпляром. С формальной точки зрения фрейм представляется конструкцией:

$$F = \langle N, S_1, S_2, S_3 \rangle, \quad \text{П.}(2.6)$$

где N – имя объекта, S_1 – множество слотов, содержащих факты, определяющие декларативную семантику фрейма, S_2 – множество слотов, обеспечивающих связи с другими фреймами, S_3 – множество слотов, обеспечивающих преобразования, определяющие процедурную семантику фрейма.

Выделяют виды фреймов: экземпляры, образцы, классы, роли, сценарии.

Использование каждой из МПЗ при моделировании СОТО связано с рядом проблем, которыми характеризуются такие модели в приложении к особенностям функционирования СОТО [153].

Таким образом, на основе проведенного анализа в [153] следует сделать вывод о том, что необходимо привлечение комбинированных МПЗ так, чтобы включаемые МПЗ нивелировали недостатки друг друга и дополняли преимущества. В работе [132] предложен подход совместного использования семантической сети и фреймового представления знаний. При этом учитывается возможность применимости алгоритмов в составе фреймов к входным данным с помощью условных выражений (см. рис. П. 2.6). Кроме того, при сравнении по ряду критериев оказалось, что комбинация фреймовой и продукционной моделей позволяет комплексно подходить к моделированию процессов, происходящих в СОТО.

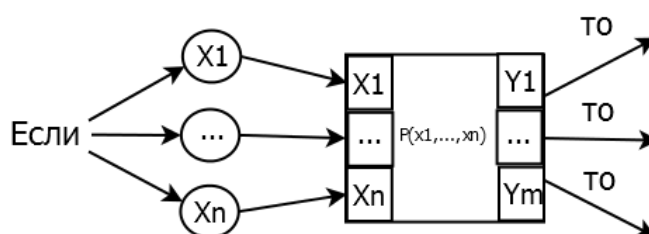


Рис. П. 2.6. Схема продукционно-фреймового представления знаний
(P – некоторый алгоритм)

При таком подходе может быть сформирована продукционная БЗ [15, 40, 60, 164]. На основе такой БЗ может моделироваться принятие решений, связанное с процессами мониторинга и управления СОТО, соответствующее некоторому сложному дереву решений [252], расширенному за счет способности фреймов моделировать сложные процессы. Кроме того, такой подход реализует декларативно-процедуральное представление знаний о функционировании объекта анализа, где знания становятся активными элементами управления. Существенно и то, что такая МПЗ выставляет на первое место модель как средство описания ПрО. Алгоритмы же становятся лишь инвариантными автоматическими блоками в составе таких моделей (см. далее). Примером такой унифицированной МПЗ может служить *G-модель* (см. приложение 1.8.3).

Был проведен системный анализ классических МПЗ и продукционно-фреймовой G-модели по некоторым существенным для задачи показателям [60, 147, 153].

К таким показателям можно отнести отношения, в которые вступают информационные единицы системы: отношения структуризации (иерархичность), функциональные отношения, каузальные отношения (причинно-следственные), семантические отношения, отношения релевантности (семантическая метрика, ассоциативная связь).

Кроме того, к дополнительным показателям также можно отнести возможность моделирования различных типов знаний: поверхностные знания (эмпирические ассоциации, причинно-следственные отношения), глубинные знания (абстракции, образы, аналогии, структура ПрО), жесткие знания (четкие рекомендации при заданных начальных условиях), мягкие знания (множественные расплывчатые решения) [20].

Таким образом, был проведен анализ МПЗ по следующим показателям возможности моделирования (см. таблицу П. 2.7):

1. декларативно-процедурального;
2. отношений структуризации;
3. функциональных отношений;
4. каузальных отношений;
5. семантических отношений;
6. отношений релевантности;
7. поверхностных знаний;
8. глубинных знаний;
9. жестких знаний;
10. мягких знаний;
11. дедуктивного вывода;
12. индуктивного вывода.

Таблица П. 2.7. Системный анализ МПЗ

Вид модели	1	2	3	4	5	6	7	8	9	10	11	12	+
Продукционная модель	+	-	-	+	-	-	+	-	+	+	+	-	6
Формальная логическая модель	-	-	+	+	-	-	+	-	+	-	+	-	5
Семантическая сеть (сетевая модель)	+	+	-	-	+	+	-	+	+	+	-	-	7
Фреймовая модель	+	+	+	-	-	+	-	+	+	+	-	-	7
Продукционно-фреймовая модель (G-модель)	+	-	+	+	+	-	+	+	+	+	+	-	9

В результате проведенного анализа было выявлено, что G-модель обладает рядом преимуществ по сравнению с другими МПЗ при решении задачи автоматизированного анализа СОТО. В то же время существенно отметить, что, как было выяснено в разделе 1.2.2, использование G-моделей может быть положено в основу системы только в части моделирования решения аналитических вычислительных задач. Процессы ЖЦ СОТО характеризуются большим количеством таких задач, порядок выполнения которых должен быть согласован. Кроме того, аппарат необходимо дополнить в части задач интеграции информационных ресурсов. Также необходимо отметить, что G-модели не могут быть использованы напрямую для моделирования бизнес-процессов.

2.3.4. Роль моделей и алгоритмов в интеллектуальных системах

Моделирование процессов в составе СОТО при помощи ЭВМ связано с представлением информации о них в форме, реализуемой на ЭВМ. Таким образом, необходимо рассмотреть два фундаментальных для ИТ понятия, позволяющих говорить о представлении информации: модель и алгоритм [120, 132, 153, 182].

Определение П. 2.8. Под *моделью* можно понимать определенное множество абстрактных объектов или несколько множеств абстрактных объектов разной природы, различающихся условно приписываемыми им именами, в совокупности с заданной системой отношений между элементами этих множеств [132].

Определение П. 2.9. Модель в общем смысле – это создаваемый с целью получения и (или) хранения информации специфический объект (в форме мысленного образа, описания знаковыми средствами либо материальной системы), отражающей свойства, характеристики и связи объекта-оригинала произвольной природы, существенные для задачи, решаемой субъектом [124].

Следует отметить, что уже довольно продолжительное время наблюдается тенденция, в которой модель начинает преобладать над алгоритмом [120]. Дело в том, что информация, которой оперирует ЭВМ, разделяется на процедурную (императивную) и декларативную [164]. Процедурная информация о веществе в программах, которые выполняются в процессе решения задач, а декларативная информация – в данных, с которыми эти программы работают. Процедурная информация представляется в форме алгоритма, где под алгоритмом можно понимать точное предписание последовательности действий, необходимых для получения искомого результата [83]. При этом алгоритм определяет цепочку причинно-следственных связей с помощью вычислительных операций. Декларативная информация представляется в моделях как формальное описание исследуемого объекта, процесса или явления. Считается, что модель определяет, ЧТО надо вычислять, а алгоритм – КАК надо вычислять [73]. Исходя из вышеизложенного, можно выделить ряд преимуществ модели над алгоритмом (см. таблицу Таблица П. 2.8) [120, 132].

Таблица П. 2.8. Сравнительная таблица модели и алгоритма

Модель	Алгоритм
Принципиально декларативна	В определенном смысле антидекларативен
Симметрична по отношению к параметрам, поскольку все они неявным образом определяются друг через друга	Разделяет параметры на входные и выходные, явным образом определяя вторые через первые
В неявной форме определяет решение всех задач, связанных с объектом модели	Определяет в явной форме задает решение только одной задачи, отношение которой к реальному объекту не всегда очевидно
Может быть недоопределенной	Алгоритм и недоопределенность - несовместимые понятия
В общем случае определяет пространство решений	Традиционный (не интервальный) алгоритм позволяет получать только отдельные точечные решения

Поскольку модель ориентирована на конкретную Про и позволяет конструктивно описывать функционирование анализируемого объекта или системы, можно говорить о том, что модель является более высокоуровневым средством представления по сравнению с алгоритмом. Модель не просто определяет порядок преобразования входных данных в выходные, как это происходит в алгоритме, но позволяет оперировать абстрактными сущностями и отношениями между ними, приближая таким образом ход вычислений в ЭВМ к тем подходам, которые свойственны человеческому мышлению. В задаче поиска в пространстве состояний алгоритмы поиска каждого состояния представляют собой черный ящик с неразличимой внутренней структурой. Уже сейчас наблюдается активный переход от алгоритмов к модели. Этот подход развит в ООП, в работах Э.Х. Тыугу в виде ВМ, в программировании в ограничениях в виде Н-моделей, в языке СIP-L, в G-моделях [73, 100, 120, 132, 169, 193, 197]. Приведенный порядок указывает на то, что основным средством для интеллектуальных систем становится модель, в то время как алгоритм становится лишь инвариантной составной частью модели.

В то же время необходимо учитывать, что сложность современных ПК требует постадийного разделения труда при их разработке. «В этих условиях необходимо многоязычие, образность, простые правила перевода с одного языка на другой и стабильность, позволяющие воплощать замысел, не искажая его при переходе от одной стадии разработки к следующей за ней» [73]. Эту ситуацию можно выразить иначе: кроме того, что модели могут использоваться вместо алгоритмов при проектировании, необходима взаимосвязанная иерархия моделей (а точнее, на системном уровне иерархия МПЗ). Ситуация, когда такая система моделей отсутствует, некоторому специалисту в одном лице приходится совмещать функции постановщика задачи, инженера по знаниям, и конструктора программ. Таким образом, необходимо разделить эти функции на три отдельных этапа концептуализации. При этом выразительные средства используемых для этих трёх этапов языков должны располагать к стилю мышления соответствующих специалистов, занимающихся тем или

инным этапом, поскольку символическое представление сильно связано с понятийным (семантическим пониманием) в соответствии с известным треугольником Г. Фреге (см. рис. 2.1) [164, 198, 199, 200]. В этом случае говорят и синтаксисе, семантике и прагматике (см. рис. П. 2.7) [73].

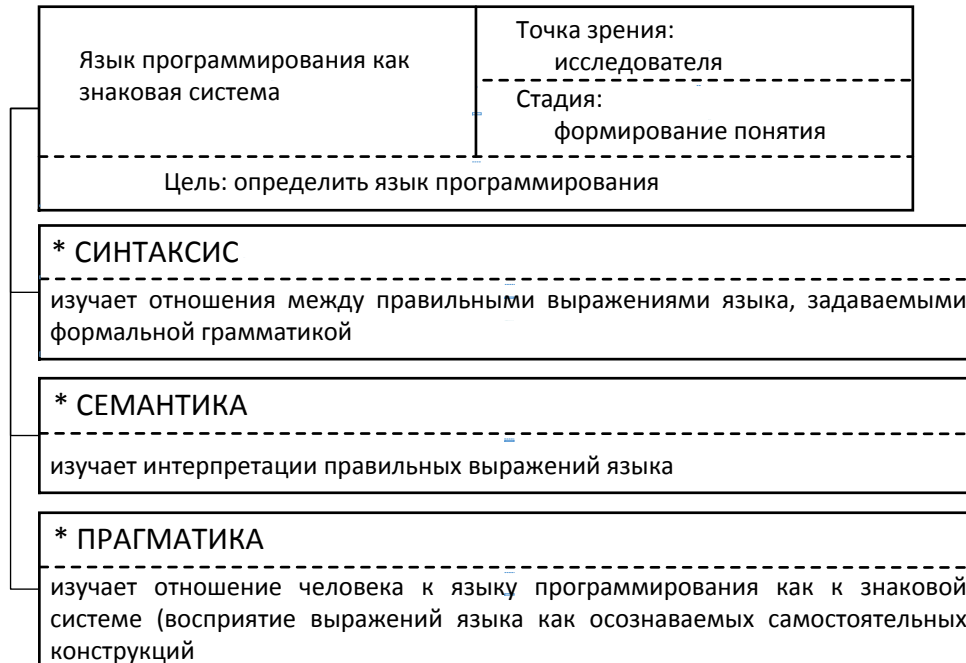


Рис. П. 2.7. Семиотические аспекты языков программирования

В результате можно говорить о необходимости учета следующих свойств создаваемой системы:

- использование декларативно-процедуральных МПЗ;
- применение ЭВМ уже на этапе постановки задачи;
- программирование в терминах ПрО решаемых задач;
- использование языков, отвечающих на вопрос КАК, а не ЧТО;
- использование программ как некоторой формы хранения знаний о решении задач;
- автоматический синтез программы решения каждой задачи;
- использование многоязычного моделирования в ориентации на пользователей разных этапов проектирования (описания процессов, описание порядка согласования задач, описание самих задач).

2.3.5. Онтологическое моделирование

С 80-х годов 20 века и по настоящее время высокими темпами развивается направление, связанное с использованием онтологического моделирования в ИТ. *Онтологии* используются как способ формального описания используемых понятий и их взаимосвязей при проектировании ИС (см. определение П. 2.10). Это, с одной стороны, определяет их важную роль в теории ИИ, с другой – что онтологическое моделирование должно предварять разработку любой ИС в рамках концептуального однозначного определения терминологического словаря той ПрО, для которой создается ПО с целью сохранения семантики и последующего его совместного использования [25].

Определение П. 2.10. Онтология – это точная спецификация концептуализации, при этом под концептуализацией понимается структура реальности, рассматриваемая независимо от словаря ПрО конкретной ситуации (определение Т. Груббера) [25, 41, 42].

К причинам проектирования онтологий можно отнести [25, 41, 42, 82, 89, 207, 264, 274]:

- возможность совместного использования людьми или программными агентами общего понимания структуры информации;
 - возможность повторного использования знаний о ПрО;
 - явно формируемые ограничения и допущения в ПрО;
 - возможность анализировать извлекаемые знания на корректность и противоречивость;
 - возможность осуществлять логический вывод новых знаний на совокупности утверждений о ПрО;
 - возможность аннотирования концептуальных моделей ПрО
- и др.

В работах [25, 41, 114, 211] отмечается возможность создания онтологий предприятия, как формализованное описание знаний о бизнес-ресурсах компании, проектах, задачах, заказах и планах, их атрибутах и отношениях. Отделяя знания о ПрО от системного кода, онтология не только позволяет вносить

изменения в систему специалистам без навыков программирования, но и дает возможность вносить эти изменения без прерывания работы системы. При этом выполняется онтологический анализ – процесс выявления и определения концептов реальной ПрО, их фундаментальных свойств и отношений. На основе готовой и заполненной онтологии осуществляют автоматический семантический поиск по структуре концептов с целью нахождения релевантной запросу информации. Так, например, большие надежды возлагают на идею Семантического Web-пространства, предложенного Т. Бернерсом-Ли и основанного на представлении ресурсов глобальной сети с помощью онтологий для обеспечения прозрачного доступа ко всей информации [223].

Во многих работах предполагается, что при создании онтологий необходимо опираться на *онтологическую систему* [25, 41, 69, 104]. Такая система представляет собой взаимосвязанную группу онтологий: представления (концептуализации формализмов представления знаний), верхнего уровня (повторно используемых в разных ПрО), предметную (повторно используемую внутри одной ПрО), задач/действий (описывающую совокупность задач ПрО), прикладную (используемую конкретной прикладной программой).

Основная идея онтологической системы в том, чтобы согласовать понятия разных аспектов и разных уровней представления между собой для более точного описания ПрО. Цель онтологии представления – описать область представления знаний, создать язык для спецификации других онтологий более низких уровней. Например, описание понятий языка OWL средствами RDF/RDFS. Многие онтологии верхнего уровня похожи друг на друга. Они содержат одни и те же концепты: сущность, явление, процесс, объект, роль, пространство, время, материя, событие, действие и т. п. Онтологии верхнего уровня могут способствовать интеграции данных, обеспечению интероперабельности. Предметная онтология (онтология домена) обобщает понятия, использующиеся в некоторых задачах домена, абстрагируясь от самих задач. Предметная онтология – описывает реальные предметы, участвующие в какой-либо деятельности (например, производстве). Онтология задач содержит

термины, которые используются при решении задач в ПрО. Назначение прикладной онтологии в том, чтобы описать концептуальную модель конкретной задачи или приложения. Анализ приведенной литературы показал, что несмотря на заявления о необходимости формализации и создания систем на основе совокупности взаимосвязанных онтологий, ни реализации, ни конструктивного теоретического аппарата эта идея пока не получила.

Стоит отметить, что в ряде работ [176, 198, 199, 200, 234, 235] предпринимались попытки по описанию процессов в ПрО, что существенно для рассматриваемой в работе задачи. При этом оказалось, что такое описание в онтологических языках возможно, но имеющиеся реализации имеют частный характер. В тоже время, все они ориентированы на описание процессов, как смену состояний, а не как последовательность действий в соответствии с представлением бизнес-процессов.

Онтологии могут быть использованы для аннотирования моделей. В таком случае концепты онтологии связаны с элементами моделей [89]. Этот аспект при использовании онтологической системы позволяет говорить о возможности создания системы взаимосвязанных языков, где переходы между ними могут осуществляться на основе онтологий, связывающих понятия одного языка с другим. Такие попытки уже предпринимались, однако носили несистемный характер и не использовали концепцию онтологической системы [41].

Онтология может использоваться для описания и спецификации схем БД или БЗ [17, 25, 41]. Этот факт говорит о возможности интегрировать информацию, находящуюся в различных источниках, — электронные документы, БЗ и БД, структурированные данные с использованием языков разметки. При этом процесс интеграции состоит в спецификации содержимого разнородных источников данных в виде онтологии, получения ответов на запросы, адресованные интегрирующей системе и основанные на спецификации источников. В таком случае, концептуальный уровень предполагает концептуальное пред-

ставление источников и интегрируемых данных вместе с явным декларативным описанием отношений между их компонентами, а логический – содержит представление источников в терминах логической модели данных. В работе [207] отмечается, что используемые на практике средства интеграции информационных ресурсов не оправдывают своих ожиданий в силу высокой стоимости и низкой эффективности. Иной подход, зародившийся в идее семантической паутины, может стать перспективным направлением, в частности, создания «семантических предприятий», где квазиструктурированные данные («semi-structured») составляют основу информации, подвергаемой обработке. Например, многие документы, используемые в бизнесе, нельзя признать полностью свободными от структурирования, но их структуры не являются таким строгими, как БД. При том, что отчеты и им подобные документы предназначены для чтения человеком, они в достаточной степени формальны, чтобы при анализе можно было выделить содержательные метаданные. Известные технологии ODBC, концепция COA, существующие прикладные протоколы могут выступать поддерживающими для будущей технологии семантической интеграции на основе онтологий. То, как данные представляются в семантической сети, можно рассматривать как новый шаг в управлении данными, и вполне естественно воспользоваться этими преимуществами в корпоративных ИС. Степень интегрированности информации с позиций онтологического подхода автор [207] представляет следующим образом (см. рис. Рис. П. 2.8).

Важно отметить, что на сегодняшний день уже существует множество языков описания онтологий: RDF, RDFS, DAML+OIL, OWL, LOOM, OKBC, OCML, F-Logic, XOL, SHOE, UPML [25, 233, 256, 263, 272]. Это позволяет говорить о том, что уже подготовлена почва для использования онтологий в сложных задачах аннотирования моделей, интеграции информационных ресурсов, моделирования процессов и задач.

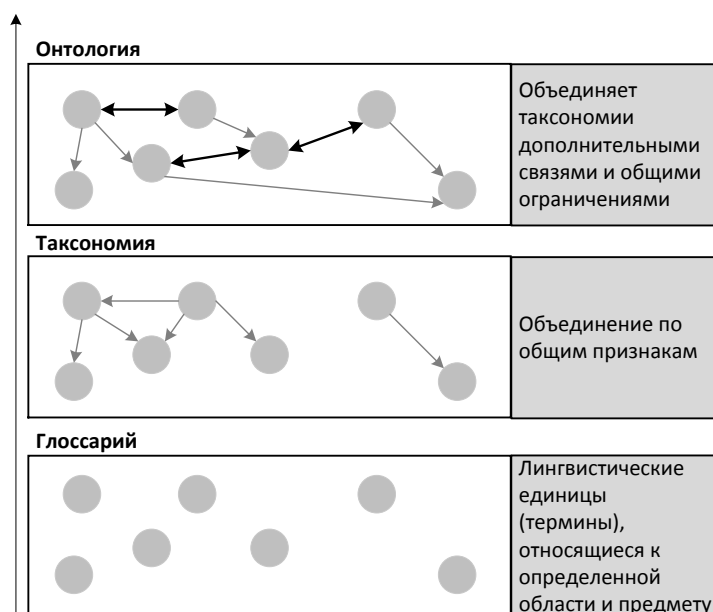


Рис. П. 2.8. Уровни интеграции информационных ресурсов

2.3.6. Мультиагентные системы

В соответствии с определением 1.11 ЖЦ представляет собой совокупность взаимосвязанных процессов. Автоматизация анализа ТС, качества и надежности СОТО, в таком случае, предполагает распределенный комплекс программ, взаимодействующих между собой. Этот аспект требует рассмотрения *МАС*, как возможный вид архитектуры взаимодействующих агентов [41].

Агентно-ориентированный подход находит применение в таких областях, как распределенное решение сложных задач, совмещенное проектирование изделий, реинжиниринг бизнеса и построение ВП, имитационное моделирование интегрированных производственных систем и др.

В слабом смысле понятие агента предполагает у него наличие следующих свойств [41, 46, 106]:

- активность – способность к организации и реализации действий;
- автономность – относительная независимость от окружающей среды или наличие некоторой «свободы воли»;
- общительность – решение своих задач совместно с другими агентами на основе протоколов коммуникации;

– целенаправленность – наличие собственных источников мотивации, интенциональных характеристик.

В сильном определении предполагается также наличие у него знаний, намерений, убеждений, желаний, целей и обязательств [46].

Важным аспектом МАС является кооперативное распределенное решение задач в виде сети слабо связанных между собой решателей, которые совместно работают в целях решения задач, выходящих за рамки индивидуальных возможностей (эмерджентность). Различные узлы сети при этом имеют неодинаковый опыт, знания, точки зрения и разные ресурсы [106]. Цель функционирования МАС – удовлетворение различных информационных и вычислительных потребностей конечных пользователей [46]. Для реализации МАС необходимо определить формализмы и математические методы для описания рассуждений агентов, методы их кооперации, их архитектуру (одноуровневую или иерархическую), языки описания моделей агентов, языки и протоколы взаимодействия.

Известно, что коллективы даже простейших автоматов, в которых каждый автомат преследует только свои примитивные цели, в целом способны решать очень сложные задачи [46]. Однако использование идеи коллективного поведения приводит к массе проблем реализации указанных выше свойств. Так, например, создание МАС в соответствии с определением в сильном смысле с помощью исчисления предикатов первого порядка (язык STRIPS), с помощью модальных логик и формальных грамматик не привело к успешным результатам из-за неразрешимости некоторых задач, проблем описания окружения и невыразимость указанных свойств агентов [169].

Реализация самого агента предполагает либо функционирование на основе знаний (ИИТ), либо на основе поведения (реакция системы на события). В работе [46] отмечается, что на практике необходимо использовать гибридное описание. Отмечается также необходимость реализации функционирования агентов на основе альтернатив планов в ориентации на достижение целей.

В настоящее время не существует языка программирования, который в полной мере отвечает требованиям, предъявляемым к МАС их назначению и свойствам [46]. В результате системы создают на основе совокупности различных языков, что вызывает сложности при переводе с одного языка на другой. В то же время определяют следующие требования к языкам: обеспечение переносимости кода и доступность на многих платформах, поддержка сетевого взаимодействия, многопоточная обработка, поддержка символьных вычислений, истинная объектная ориентированность, языковая поддержка свойств агента.

Можно отметить, что в настоящее время активно развивается теория дескрипционных (описательных логик), выразительная мощь которых сочетает в себе преимущества исчисления предикатов и модальных логик. Дескрипционные логики предполагают использование ООП, позволяют описывать свойства объектов, явно задавать и наполнять БЗ, часто используются для формализации онтологий [72]. Можно предположить, что они способны стать основой для формального описания МАС.

2.3.7. Обзор методов распознавания образов

При решении задач построения как автоматических, так и автоматизированных систем управления кибернетика использует в качестве образца процессы интеллектуальной деятельности человека, реализуя в виде устройств, алгоритмов и программ те или другие стороны механизма человеческого мышления [45].

Чем чаще человеку приходится сталкиваться с определенным набором стандартных изменений некоторой ситуации, тем естественнее он вырабатывает и хранит в своей памяти соответствующую классификацию этих ситуаций, связывая с каждым классом определенное решение. В процессах функционирования и автоматических, и автоматизированных систем управления достаточно четко очерчиваются три этапа. На первом решается задача распознавания вновь возникшей ситуации или явления, на втором на основании результатов решения предыдущей задачи и в соответствии с алгоритмом в некотором

смысле оптимального управления вырабатывается решение, и, наконец, на третьем этапе реализуется принятое решение, осуществляется собственно управление.

В настоящее время еще не разработан строгий формальный аппарат, предназначенный для построения математических моделей достаточно сложных явлений и процессов, адекватных моделируемым явлениям или процессам. Один из способов обойти проблему математического моделирования, т.е. без помощи математических моделей количественно прогнозировать развитие процессов на основе накопленной информации о процессах, относящихся к соответствующим классам, и состоит в построении систем распознавания этих процессов.

Теория распознавания образов получила бурное развитие и сформировалась в отдельное научное направление прикладной математики. Это объясняется рядом причин, среди которых можно выделить, в частности, следующие [132]. С одной стороны, многие задачи обработки и анализа информации могут быть сведены к задаче распознавания. С другой, - наличие большого количества плохо формализуемых реальных систем, для которых стоит актуальная задача их изучения и моделирования поведения. При этом для таких систем не удается, как правило, синтезировать математические модели традиционными в математике средствами – из-за невозможности точного описания модели изучаемых явлений (процессов) или неспособности реализовать необходимые вычисления по синтезированной модели с помощью существующих вычислительных средств.

В реальной ситуации состояние СОТО характеризуется значениями его наблюдаемых параметров. Таким образом, можно говорить о том, что оценивание состояния СОТО есть ни что иное, как автоматизированное распознавание его состояния через текущие значения его параметров. В таком случае, об оценивании состояния СОТО можно говорить в терминах классической задачи теории распознавания образов, которая по своей содержательной сути сводится к отнесению распознаваемого объекта к одному из классов состояний. В

рамках традиционной прикладной интерпретации задачи оценивания состояния СОТО распознаваемый объект представляется в виде набора значений его параметров, а его состояние – это класс, к которому необходимо отнести распознаваемый объект.

Системы распознавания образов классифицируют самым различным образом. Одним из вариантов классификации является разделение систем распознавания с точки зрения применяемого математического аппарата. В связи с этим выделяют 4 основных типа систем распознавания:

- детерминированные,
- вероятностные,
- логические,
- комбинированные.

На основе этих основных типов выделяют несколько типов методов распознавания образов [32, 35, 45, 47, 190, 201]:

- методы на основе принципа разделения (R – модели);
- статистические методы;
- методы на основе потенциальных функций;
- методы на основе вычисления оценок;
- методы на основе исчисления высказываний (алгебры логики);
- структурные методы распознавания образов;
- методы структурно-потокково-многоуровневого распознавания образов.

Некоторые авторы (например, [35]) классифицируют эти методы несколько иначе, однако, в целом подход остаётся именно таким, как это указано выше. Каждый указанный тип требует более детального рассмотрения.

Следует также отметить, что системы распознавания образов классифицируют также с точки зрения приоритетной информации о распознаваемом объекте. При таком подходе системы распознавания образов подразделяют на экстенциональные и интенциональные [175].

Отличительной особенностью интенциональных методов является то, что в качестве элементов операций при построении и применении алгоритмов

распознавания образов они используют различные характеристики признаков и их связей. Такими элементами могут быть отдельные значения или интервалы значений признаков, средние величины и дисперсии, матрицы связей признаков и т. п., над которыми производятся действия, выражаемые в аналитической или конструктивной форме. При этом объекты в данных методах не рассматриваются как целостные информационные единицы, а выступают в роли индикаторов для оценки взаимодействия и поведения своих атрибутов.

В экстенциональных методах, в отличие от интенционального направления, каждому изучаемому объекту в большей или меньшей мере придается самостоятельное диагностическое значение. По своей сути эти методы близки к клиническому подходу, который рассматривает людей не как проранжированную по тому или иному показателю цепочку объектов, а как целостные системы, каждая из которых индивидуальна и имеет особенную диагностическую ценность. Такое бережное отношение к объектам исследования не позволяет исключать или утрачивать информацию о каждом отдельном объекте, что происходит при применении методов интенционального направления, использующих объекты только для обнаружения и фиксации закономерностей поведения их атрибутов. Основными операциями в распознавании образов с помощью обсуждаемых методов являются операции определения сходства и различия объектов. Объекты в указанной группе методов играют роль диагностических прецедентов.

Методы распознавания образов, основанные на принципе разделения – это R – модели [45] или модели, основанные на решающих функциях. Распознаваемый объект представляется в виде некоторой точки, представляющей собой набор оценок значений параметров СОТО в пространстве признаков. Для этого пространства строится решающая функция (или в графической интерпретации гиперповерхность для n -мерного пространства признаков), разделяющая эти значения на классы [190]. Гиперповерхность строится по правилу, минимизирующему количество неправильных опознаний, т.е. с минимизацией риска ошибки распознавания. Распознавание происходит на основе расчета

расстояния (близости) от точки – распознаваемого объекта до разделяющей гиперповерхности, поэтому об этой задаче говорят также как о задаче поиска экстремума. Для таких методов используются различные функционалы качества и алгоритмы поиска экстремума. Расстояние рассчитывается различными методами [45] (расстояние Махаланобиса, Евклидово расстояние, взвешенное Евклидово расстояние, Хеммингово расстояние и пр.). Функционал качества решающего правила обычно связывают с ошибкой классификации. Примером такого подхода может служить метод обобщенного портрета, представленный на рис. П. 2.9. Также примерами методов на основе принципа разделения могут служить метод стохастической аппроксимации [105], методы эволюционного моделирования (метод группового учёта аргументов и др.), алгоритм Гаусса-Зейделя [175].

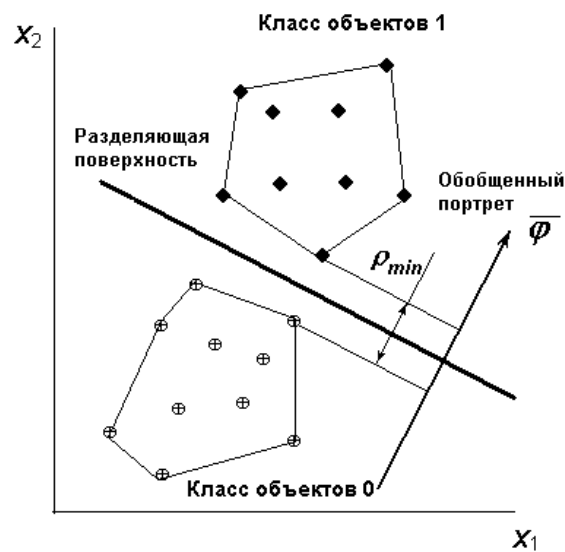


Рис. П. 2.9. Графическая интерпретация метода обобщенного портрета

К достоинствам методов, основанных на принципе разделения можно отнести ясную математическую постановку задачи распознавания как задачи поиска экстремума, а также многообразие формализованных алгоритмов расчета расстояния и функционалов качества. К недостаткам методов можно отнести высокую сложность расчетов при большом количестве признаков, а

также заранее заданную структуру решающей функции, что ограничивает возможный круг решаемых задач распознавания для конкретной реализованной системы распознавания образов.

Статистические методы распознавания образов применяются тогда, когда между признаками объектов и классами, к которым они относятся существует вероятностная зависимость [32, 45]. Распознаваемый объект представляется как реализация многомерной случайной величины. При этом предполагается, что для такого метода априорно известно распределение значений признаков. Эти методы базируются на байесовской схеме принятия решений, апеллирующей к априорным вероятностям принадлежности объектов к тому или иному распознаваемому классу и условным плотностям распределения значений вектора признаков. Для тривиального случая 2 классов состояний и одного признака процесс распознавания сводится к разделению имеющегося распределения некоторой точкой, и далее выдвигаются гипотезы о принадлежности распознаваемого объекта к тому или иному классу. Рассчитывается вероятность правильного распознавания на основе ошибок первого и второго рода. Данные методы сводятся к определению отношения правдоподобия в различных областях многомерного пространства признаков. Пример тривиального случая показан на рис. П. 2.10. В зависимости от различных исходных данных в статистических методах применяются различные критерии: Байеса, минимаксный, Неймана-Пирсона и др.

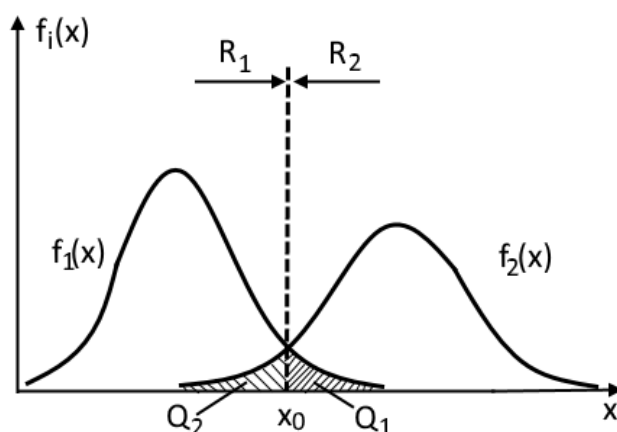


Рис. П. 2.10. Графическая интерпретация статистического метода распознавания для тривиального случая

Примерами статистических методов могут служить метод Байеса, методы дискриминантного анализа. К достоинствам статистических методов распознавания можно отнести многообразие критериев для различных исходных условий задачи. К недостаткам можно отнести необходимость постулирования существования априорного распределения, необходимость априорного знания формы распределения, а также сложность вычислений для полимодальных распределений.

Методы, основанные на применении потенциальных функций, заключаются прежде всего в формировании обучающей выборки прецедентов – пар «распознаваемый объект; информация о том, к какому классу этот объект относится», а также построении решающей функции, называемой «потенциалом» [32]. Каждому прецеденту присваивается определенный «заряд» (для случая дихотомии заряд может быть положительным либо отрицательным). Таким образом, образуется некоторое поле притяжения, формируемое системой зарядов в пространстве признаков. Решающая функция должна убывать по мере отдаления от прецедента (идеального образа). То есть, потенциал – это функция, монотонно уменьшающаяся с увеличением расстояния. В итоге, распознавание осуществляется притяжением к одному из заряженных полей, где поле – это класс состояний. Графическая интерпретация данного метода представлена на рис. П. 2.11. Примерами методов на основе потенциальных функций могут служить общая рекуррентная процедура [3], методы с использованием функций Эрмита.

К достоинствам данного подхода можно отнести относительную простоту решения задач с нелинейным разбиением множества объектов. К недостаткам следует отнести так называемую «наивность» метода – необходимость в соизмеримости количества точек разных классов в признаковом пространстве, а также проблему наложения полей одних точек на другие. Также можно обозначить высокую сложность вычислений для большой обучающей выборки.

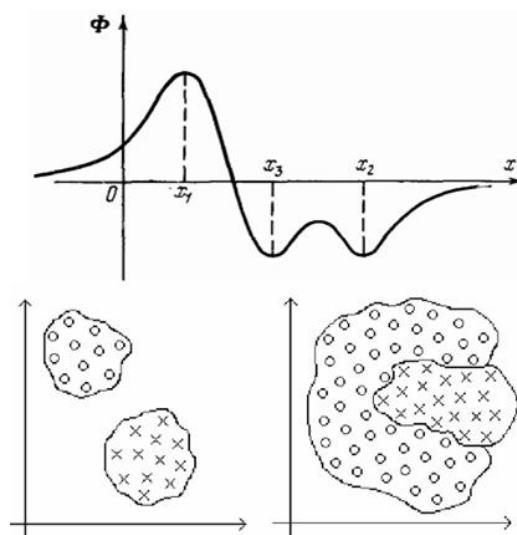


Рис. П. 2.11. Графическая интерпретация метода потенциальных функций

Принцип действия методов на основе вычисления оценок состоит в вычислении приоритетов (оценок сходства), характеризующих «близость» распознаваемого и эталонных объектов по системе ансамблей признаков, представляющей собой систему подмножеств заданного множества признаков [45, 105]. В отличие от всех ранее рассмотренных методов алгоритмы вычисления оценок принципиально по-новому оперируют описаниями объектов: при распознавании учитываются различные сочетания значений признаков распознаваемого объекта. Поскольку не всегда известно, какие сочетания признаков наиболее информативны, то в методах вычисления оценок степень сходства объектов вычисляется при сопоставлении всех возможных или определенных сочетаний признаков, входящих в описания объектов. Используемые сочетания признаков (подпространства) авторы [45] называют опорными множествами или множествами частичных описаний объектов. Вводится понятие обобщенной близости между распознаваемым объектом и объектами обучающей выборки (с известной классификацией). Критерием качества служит ошибка распознавания. В процессе распознавания формируется информация об объекте в виде вектора оценок значений признаков, а также формируется информация о классах состояний, часто представляемая в форме «таблицы

обучения» [45] (рис. П. 2.12). Распознавание объекта осуществляется сравнением сформированного вектора с каждой строкой таблицы обучения. В результате сравнения строится матрица оценок, элементы которой отражают степень подобия информации об объекте и информации из таблицы обучения. Примерами методов распознавания образов на основе вычисления оценок могут служить некоторые методы экспертной оценки или специальные эвристические алгоритмы. Общих строго формализованных алгоритмов для данного подхода не существует, в связи с этим конкретная реализация данного подхода зависит от конкретной задачи.

Объекты	Признаки и их значения				Классы		
	X_1	X_2	X_j	X_N			
ω_1	$a_{1,1}$	$a_{1,2}$...	$a_{1,j}$...	$a_{1,N}$	Ω_1
ω_2	$a_{2,1}$	$a_{2,2}$...	$a_{2,j}$...	$a_{2,N}$	
...	
ω_{r_i}	$a_{r_i,1}$	$a_{r_i,2}$...	$a_{r_i,j}$...	$a_{r_i,N}$	
.....							
$\omega_{r_{i-1}+1}$	$a_{r_{i-1}+1,1}$	$a_{r_{i-1}+1,2}$...	$a_{r_{i-1}+1,j}$...	$a_{r_{i-1}+1,N}$	Ω_i
$\omega_{r_{i-1}+2}$	$a_{r_{i-1}+2,1}$	$a_{r_{i-1}+2,2}$...	$a_{r_{i-1}+2,j}$...	$a_{r_{i-1}+2,N}$	
...	
ω_{r_i}	$a_{r_i,1}$	$a_{r_i,2}$...	$a_{r_i,j}$...	$a_{r_i,N}$	

Рис. П. 2.12. Общий вид таблицы обучения при распознавании на основе вычисления оценок

К достоинствам данного подхода можно отнести возможность решения задач прогнозирования за счёт полного перебора различных сочетаний значений признаков. К недостаткам можно отнести отсутствие учёта структурных связей между признаками (или элементами) распознаваемого объекта.

Логические методы распознавания образов базируются на аппарате алгебры логики и позволяют оперировать информацией, заключенной не только в отдельных признаках, но и в сочетаниях значений признаков [35, 175]. В рамках этих методов значения признаков представляются как элементарные собы-

тия. В самом общем виде логические методы можно охарактеризовать как разновидность поиска по обучающей выборке логических закономерностей и формирование некоторой системы логических решающих правил (например, в виде конъюнкций элементарных событий), каждое из которых имеет собственный вес. Таким образом, распознаваемый образ представляется как некоторое логическое высказывание из элементарных событий. Непосредственно распознавание заключается в формировании логического выражения, отражающего свойства класса состояний и логического выражения со значениями признаков распознаваемого объекта; в результате их сравнения происходит определение объекта к одному из классов. Группа логических методов разнообразна и включает методы различной сложности и глубины анализа. Для дихотомических (булевых) признаков популярными являются так называемые древообразные классификаторы, метод тупиковых тестов, алгоритм «Кора» и др.

К достоинствам данного подхода можно отнести детерминированность логического представления, а также учёт структурных связей между признаками распознаваемого объекта. К недостаткам можно отнести высокую вычислительную сложность вследствие полного перебора, а также ограничение применимости данных методов только для ситуации с высокой достоверностью значений признаков, поскольку ошибка классификации в рамках данных методов не учитывается.

В рамках методов структурного распознавания образов, которое зачастую также называют лингвистическим, чётко определено понятие образа. Образ определяется как набор подобразов, сходящихся к непроемным элементам, а также отношения между ними [35]. Лингвистические методы распознавания образов основаны на использовании специальных грамматик, порождающих языки, с помощью которых может описываться совокупность свойств распознаваемых объектов. Для различных классов объектов выделяются непроемные (атомарные) элементы и возможные отношения между ними. Грамматикой называют правила построения объектов из этих непроемных

элементов [190, 201]. Таким образом, каждый объект представляет собой совокупность производных элементов, «соединенных» между собой теми или иными способами или, другими словами, «предложением» некоторого «языка». Путем синтаксического анализа (грамматического разбора) «предложения» определяется его синтаксическая «правильность» или, что эквивалентно, может ли некоторая фиксированная грамматика, описывающая класс, породить имеющееся описание объекта. Таким образом, в рамках лингвистических методов «распознать» - значит определить, является ли образ («предложение») синтаксически верным. Примерами методов структурного распознавания образов могут служить деревья решений, конечные автоматы, автоматы с магазинной памятью, машина Тьюринга и др.

К достоинствам методов структурного распознавания образов можно отнести возможность распознавания «искаженных» объектов на основе стохастических грамматик, наличие большого множества формализованных методов и алгоритмов. К недостаткам данного подхода следует отнести прежде всего интуитивный подбор производных элементов, а также сложность формализации для конкретной задачи.

Другим методом распознавания образов, характеризующемуся как интенциональными, так и экстенциональными признаками, является структурно-потокково-многоуровневое (СПМ) распознавание, предложенное в методологии автоматизированного анализа на основе G-моделей [130, 132]. Термином СПМ подчеркивается наличие трех основных качеств этого типа распознавания:

- структурность распознавания означает, что основным носителем знаний о ПрО является топологическая структура параметров ТС;
- потоковость означает, что процесс распознавания предполагает наличие потоковой асинхронной модели вычислений, обеспечивающей непрерывное получение результатов распознавания в РВ;

- многоуровневость означает, что процесс распознавания включает в себя несколько уровней, причем таких, что результаты распознавания на более низких уровнях используются как исходные данные распознавания на более высоких уровнях.

Для СПМ-распознавания могут быть сформулированы следующие задачи:

а. Синтез системы СПМ-распознавания ТС – как задача формирования программы (схемы программы) вычислений для оперативного потокового многоуровневого вычисления заданной цели анализа – может быть сформулирована следующим образом:

- на основе сформированного пространства параметров ТС, несущего в себе данные о ПрО в виде G-модели и учитывающей отношение обучения, синтезировать все программы анализа в виде G-сетей, правильно вычисляющих значения параметров цели анализа по отношению классификации;
- из синтезированного множества программ выбрать такую, которая может быть выполнена за минимальное время.

б. Непосредственно анализа информации по синтезированной схеме программы, обеспечивающей потоковое многоуровневое получение достоверных результатов в РВ:

- по синтезированной схеме программы анализа, соответствующей отношению классификации, построить потоковое многоуровневое вычисление значений параметров цели анализа по отнесению предъявленных для распознавания объектов к одному или нескольким классам, которые дают наибольшую достоверность.

Если СПМ-распознавание применяется для построения и реализации потоковой многоуровневой программы анализа информации, то традиционное распознавание – для построения и реализации операторов такой программы. Это значит, что если в качестве какой-либо ВМ, используется некоторая процедура распознавания, то для нее может быть применен богатый методологический аппарат решения традиционной задачи распознавания. Это означает,

что в составе операторов ВМ могут использоваться любые известные методы распознавания образов, имеющие свой реализующий алгоритм. При этом в ходе вычислительного процесса по схеме программы происходит постепенное доопределение (уточнение) значений параметров, сужающее область допустимых классов состояний для цели анализа. В связи с этим задачи распознавания, решаемые в рамках ВМ, следует называть локальными задачами распознавания [132].

Указанные факты можно отнести к существенному преимуществу СПМ-распознавания перед традиционным для решаемой в работе задачи. Разнообразие аспектов рассматриваемой ПрО требует использования различных методов распознавания, адекватно распознающих состояния СОТО на том или ином этапе ЖЦ при решении частных аналитических задач. Это определяет возможность в системе СПМ-распознавания как применения всех рассмотренных методов, так и методов, не вошедших в работу. Возможность поэтапного «сжатия» пространства допустимых классов состояний методом СПМ-распознавания позволяет в условиях неполной информации о состоянии СОТО добиваться нахождения частичных результатов анализа, в то время как другие методы в подобной ситуации не дают никакого решения.

Говоря о сравнении рассмотренных методов, можно сказать, что все они, за исключением методов на основе вычисления оценок, относятся к интенциональному типу. Сравнение этих методов представлено в таблице П. 2.9.

Таблица П. 2.9. Сравнительный анализ методов распознавания образов

Тип методов	Методы	Достоинства	Недостатки
Интенциональные методы	Основанные на принципе разделения	Учёт связей между признаками распознаваемого объекта; Ясная математическая постановка задачи; Многообразие формализованных алгоритмов.	Высокая вычислительная сложность для большого пространства признаков; Сложность построения гиперповерхности для большого пространства признаков; Сложность соблюдения требуемого качества распознавания при высокой вычислительной сложности;
	Статистические		
	На основе потенциальных функций		
	На основе исчисления высказываний		

	Структурного распознавания		Заранее заданная структура решающей функции.
Экстенциональные методы	На основе вычисления оценок	Нет ограничений по структуре распознаваемого объекта.	Высокая вычислительная сложность; Поиск оптимальных метрик сравнения.
Комбинированные методы	СПМ-распознавание	Учёт связей между признаками распознаваемого объекта; Ясная математическая постановка задачи; Распознавание в режиме РВ; Возможность поэтапного доопределения в пространстве классов состояний; Возможность использовать в своем составе любые другие методы распознавания образов.	Сложность повторного использования моделей, основанных на указанном методе распознавания, в силу их предметной ориентации.

Таким образом, обзор методов распознавания образов показывает, что в настоящее время теоретически разработан целый ряд различных методов распознавания образов. В литературе приводится развернутая их классификация. Однако для большинства этих методов их программная реализация существует только для сравнительно простых систем распознавания (таких как, например, распознавание номерных знаков автомобиля), где пространство признаков невелико, количество состояний конечно, а распознаваемый объект имеет строгое и простое описание. Кроме того, на практике чаще встречаются задачи со слабо структурированной информацией, для которых полностью формализованных алгоритмов не разработано. Таким образом, следует говорить о том, что в рамках СИАП ЖЦ СОТО необходим целый ряд моделей распознавания, которые зачастую должны быть комбинированными (экстенционально – интенциональными) и которые бы отражали не только параметрическую классификацию состояний объекта, но также и его состояния с точки зрения связей или степени связности между его параметрами. Таким методом является СПМ-распознавание, применимое для рассматриваемой задачи.

2.3.8. Онтологии и дескрипционная логика

2.3.8.1. Выразительные возможности различных языков дескрипционной логики

В таблице П.2.10 представлено сопоставление различных диалектов дескрипционной логики с точки зрения их выразительных возможностей, т.е. отмечается факт наличия или отсутствия той или иной языковой конструкции (операции, концепта или роли) в соответствующем диалекте. Таблица заимствована в качестве примера из работ [17, 18, 19], а также диссертационной работы Бездушного А.А. о построении онтологической модели интеграции данных.

Таблица П.2.10. Сопоставление диалектов дескрипционной логики

Характеристика	Нотация	RDFS	$\mathcal{AL}\mathcal{E}$	$\mathcal{AL}\mathcal{N}$	$\mathcal{AL}\mathcal{C}\mathcal{N}\mathcal{R}$	$\mathcal{SH}\mathcal{I}\mathcal{P}^{(D)}$ OWL-Lite	$\mathcal{SH}\mathcal{O}\mathcal{I}\mathcal{N}^{(D)}$ OWL-DL	$\mathcal{S}\mathcal{R}\mathcal{O}\mathcal{I}\mathcal{Q}^{(D)}$ OWL 1.1
Конструкторы концептов и ролей								
\top	\mathcal{AL}		+	+	+	+	+	+
\perp	\mathcal{AL}		+	+	+	+	+	+
$\neg A$	\mathcal{AL}		+	+	+	+	+	+
$\neg C$	\mathcal{C}				+		+	+
$C_1 \sqcap C_2$	\mathcal{AL}	неявно	+	+	+	+	+	+
$C_1 \sqcup C_2$	\mathcal{U}						+	+
$\forall R.C$ Ограничения (allValuesFrom)	\mathcal{AL}	неявно range domain	+	+	+	+	+	+
$\exists R(\exists R.T)$ Простые экзистенциальные ограничения	\mathcal{AL}		+	+	+	+	+	+
$\exists R.C$ Экзистенциальные ограничения (someValuesFrom)	\mathcal{E}		+			+	+	+
$\geq nR, \leq nR$ Ограничения мощности	\mathcal{N}			+	+	только 0 и 1	+	+
$\geq n.R.C, \leq n.R.C$ Сложные ограничения мощности	\mathcal{Q}							+
$\{a_1 \dots a_n\}$ Номиналы, перечисление концепта (oneOf, value)	\mathcal{O}						+	+
Примитивные типы данных	(\mathcal{D})	+				+	+	+
R Обратные роли (inverseOf)	\mathcal{I}					+	+	+

Сложные роли, рефлексивность отрицание ролей	\mathcal{R}				только $R_1 \sqcap R_2$			+
Высказывания терминологии								
$C_1 \sqsubseteq C_2$ Вложение концептов, GCI (subClassOf)	\mathcal{AL}	+	+	+	+	+	+	+
$R_1 \sqsubseteq R_2$ Вложение ролей (subpropertyOf)	\mathcal{H}	+				+	+	+
Funct(R), или $\top \sqsubseteq \leq 1R$ Функциональные роли (Functional)	\mathcal{F}					+	+	+
Trans(R) Транзитивные роли (Transitive)	\mathcal{S}					+	+	+
Высказывания множества фактов								
$C(a), R(a,b)$ Принадлежность концепту, роли	ABox	+	+	+	+	+	+	+

2.3.8.2. Трансляция языка веб-онтологий OWL в дескрипционную логику

В таблице П.2.11 представлено отображение языка OWL в дескрипционную логику для ознакомления, каким образом формальным выражениям могут быть сопоставлены синтаксические выражения на языке OWL. Таблица заимствована в качестве примера из работ [17, 18, 19], а также диссертационной работы Бездушного А.А. о построении онтологической модели интеграции данных.

Таблица П.2.11. Трансляция синтаксических конструкций языка OWL в дескрипционную логику

Обозначение синтаксиса OWL	Выр-е DL ($\mathcal{SHOIN}^{(D)}$)	Пример	Пояснение
Конструкторы			
A	A	Человек	Атомарный концепт
d (типы данных XML)	d	xsd:integer (целое)	Примитивный тип данных
Thing	\top	\top	Верхний концепт
Nothing	\perp	\perp	Нижний концепт
intersectionOf (C_1, \dots, C_n)	$C_1 \sqcap \dots \sqcap C_n$	Человек \sqcap Мужчина	Пересечение концептов
unionOf (C_1, \dots, C_n)	$C_1 \sqcup \dots \sqcup C_n$	Мужчина \sqcup Женщина	Объединение концептов
complementOf (C)	$\neg C$	\neg Мужчина	Отрицание
oneOf (a_1, \dots, a_n)	$\{a_1, \dots, a_n\}$	{Вася, Петя}	Перечисление концепта
restriction (R allValuesFrom (C))	$\forall R.C$	\forall сын.Доктор	Универсальное ограничение

restriction (R someValuesFrom (C))	$\exists R.C$	Эсын.Юрист	Экзистенциальное ограничение
restriction (R value (a))	$\exists R.\{a\}$	Эсын. {Петя}	Существование значения роли
restriction (R minCardinality (n))	$\geq nR$	≥ 2 сын	Минимальная мощность
restriction (R maxCardinality (n))	$\leq nR$	≤ 1 сын	Максимальная мощность
restriction (R cardinality (n))	$\geq nR \sqcap \leq nR$	≥ 1 сын $\sqcap \leq 2$ сын	Конкретная мощность
Аксиомы			
SubClassOf (C ₁ , C ₂)	$C_1 \sqsubseteq C_2$	Человек \sqsubseteq Животное	Вложение концептов
EquivalentClasses (C ₁ , ..., C _n)	$C_1 \equiv C_2 \dots C_{n-1} \equiv C_n$	Муж \equiv Супруг	Эквивалентность концептов
Class (A partial C ₁ , ..., C _n)	$A \sqsubseteq C_1 \sqcap \dots \sqcap C_n$	Человек \sqsubseteq Животное \sqcap Разумное	Вложение концепта в пересечение концептов
Class (A complete C ₁ , ..., C _n)	$A \equiv C_1 \sqcap \dots \sqcap C_n$	Муж \equiv Мужчина \sqcap Супруг	Определение концепта пересечением концептов
DisjointClasses (C ₁ , ..., C _n)	$C_i \sqsubseteq \neg C_j, i < j \leq n$	Мужчина $\sqsubseteq \neg$ Женщина	Неперекрываемость концептов
SubPropertyOf (R ₁ , R ₂), ObjectProperty (R ₁ super(R ₂))	$R_1 \sqsubseteq R_2$	сын \sqsubseteq ребенок	Вложение ролей
EquivalentProperties (R ₁ , R ₂)	$R_1 \equiv R_2$	цена \equiv стоимость	Эквивалентность ролей
ObjectProperty (R ₁ inverseOf(R ₂))	$R_1 \equiv R_2^{-}$	предок \equiv потомок	Обратные роли
ObjectProperty (R Transitive)	Trans (R)	Trans (потомок)	Транзитивность роли
ObjectProperty (R Symmetric)	$R \equiv R^{-}$	связан \equiv связан	Симметричность роли
ObjectProperty (R Functional), DatatypeProperty (R Functional)	$\top \sqsubseteq \leq 1R$	$\top \sqsubseteq \leq 1$ отец	Функциональная роль
ObjectProperty (R InverseFunctional)	$\top \sqsubseteq \leq 1R^{-}$	$\top \sqsubseteq \leq 1$ сын	Обратно функциональная роль
ObjectProperty (R ₁ range (C)), DatatypeProperty (R ₂ range (D))	$\top \sqsubseteq \forall R_1.C$ $\top \sqsubseteq \forall R_1.D$	$\top \sqsubseteq \forall \text{сын.Мужчина}$	«Область значений» роли
ObjectProperty (R domain (C)), DatatypeProperty (R domain (C))	$\exists R.\top \sqsubseteq C$, или $\top \sqsubseteq \forall R.C$	$\exists \text{сын}.\top \sqsubseteq \text{Человек}$	«Область определения» роли
SameIndividuals (a ₁ ... a _n)	$\{a_1\} \equiv \{a_2\} \dots \{a_{n-1}\} \equiv \{a_n\}$	{Петя} \equiv {Петр}	Эквивалентность объектов
DifferentIndividuals (a ₁ ... a _n)	$\{a_1\} \sqsubseteq \neg\{a_{j+1} \dots a_n\}$, $i \leq n$	{Петя} $\sqsubseteq \neg$ {Вася}	Различие объектов

3. МОДЕЛИ И АЛГОРИТМЫ ДОСТУПА К ГЕТЕРОГЕННЫМ ИНФОРМАЦИОННЫМ РЕСУРСАМ НА ОСНОВЕ ОНТОЛОГИЙ

3.1. Использование концептуальных семантических моделей для доступа к разнородным источникам данных

Зачастую информация, представленная в БД и ресурсах данных современных АС, извлекается посредством плановых и неплановых запросов [118, 119]. При этом, возникают сложности с использованием различных инструментальных средств и диалектов, необходимых для проектирования запросов, в рамках создаваемых программ вследствие того, что данные представлены в различных форматах, задействованы различные технологии управления данными, используются различные модели данных. Кроме того, несмотря на декларативность языков манипулирования данными (ЯМД), их разнообразие и чрезмерность технических деталей затрудняет понимание сути выражений в запросах. Поэтому, чтобы совместно использовать данные информационных ресурсов необходимо свести разнообразно представленную информацию к общепонятному (каноническому) виду. В качестве способа извлечения и представления знаний о таких данных могут выступать онтологии.

Информацию часто структурируют посредством БД для ее последующей автоматизированной обработки, а доступ к ней осуществляется с использованием СУБД. Для исследователей в области ИИТ СУБД представляют особый интерес, поскольку это первые крупные программы, в которых знания выступают в явной форме – в виде схем БД [193]. Схемы БД представляют собой концептуальные семантические модели той реальной действительности, информация о которой хранится в БД. В общем случае, модели данных, используемые в БД, можно разделить на две группы: реляционные и сетевые. С учетом этого, необходимы знания о структурах данных в БД указанных типов для доступа к ним.

Для полноценного управления данными в БД необходимо спроектировать ее семантическую модель, и далее в соответствии с ней осуществлять загрузку данных, запросы к СУБД, манипулирование данными (преобразование формы данных, дополнение и изменение данных) [14]. Для этих целей используются языки описания данных (ЯОД) (знания о структурах данных), языки запросов (ЯЗ), ЯМД, языки данных (задание самих данных; для отделения ЯОД в качестве языка описания знаний о данных). В качестве единой теоретической основы для них может выступать дескрипционная логика как способ создания OBDA-системы (см. раздел 1.3.3) [69, 88]. Важное предположение заключается в том, что система БД может уже существовать и разрабатываться независимо от онтологии, которую предстоит создать и связать с этой системой. Процесс создания OBDA-системы начинается с разработки онтологии ПрО, данные которой хранятся в БД. В качестве наиболее значимых возможностей OBDA-систем рассматриваются:

- предоставление развитых формальных выразительных средств для представления БД и спецификации запросов;
- обеспечение декларативности запросов в терминах такого представления;
- наличие механизмов для рассуждений на онтологии, а также для обработки сформулированных в терминах онтологии запросов данных в системе БД; последнее влечет необходимость отображения запроса, сформулированного в терминах онтологии, в запрос, специфицированный средствами ЯМД;
- способность осуществлять рассуждения и обработку запросов данных с приемлемой производительностью;
- возможность использовать в запросах явно не определенные в системе БД (скрытые) отношения. Может быть проверено качество данных. В результате может обнаруживаться неожиданная неполнота или противоречивость данных в БД.

Таким образом, сформулированный в терминах такой концептуальной схемы запрос данных должен отображаться в среду СУБД, которая будет, в конечном счете, обрабатывать конвертированные запросы. Важно, чтобы

накладные расходы на выполнение такого отображения не были значительными, а также чтобы используемая для их представления логика позволяла выражать основные модельные элементы, свойственные традиционно используемым языками концептуального моделирования. Наиболее распространенными из них в настоящее время являются ER-диаграммы и диаграммы классов языка UML.

Использование семантических предметных посредников совместно с онтологиями позволяет расширить спектр возможных источников, которые могут быть виртуально интегрированы в единую систему, а также сделать спецификацию соответствия терминов ПрО более краткой [17, 27, 37, 187].

3.2. Среда описания ресурсов как унифицированная модель данных для интеграции

Другим стимулом для применения онтологий в контексте интеграции данных распределенных гетерогенных информационных источников является модель данных, определенная для ЯОД, называемого Средой Описания Ресурсов (англ., Resource Description Framework (RDF)), которая специально спроектирована консорциумом W3C для представления распределенной в сети информации [36, 48, 187, 215, 218, 226]. RDF позиционируется как унифицирующая модель данных, предназначенная для описания информационных ресурсов таким образом, чтобы любые совместимые со стандартами RDF сторонние системы могли единым образом автоматически проинтерпретировать смысл этих описаний (принцип семантической интероперабельности). Так, например, в сети Интернет успешно развивается проект Linked Open Data (LOD), ставящий задачей накопление открытых взаимосвязанных данных в Web на основе RDF [8]. В [44] отмечается, что «реляционная модель, десятилетиями служившая основой технологии работы с данными, более не является главенствующей — на сцену выходят новые задачи, требующие учета и выявления существенно большего количества взаимосвязей (степени связности), большей выразительности и производительности. Среди новых методов обработки данных выделяется открытая и стандартизованная модель RDF: переход от

SQL-БД к системам RDF обещает технологический скачок, сравнимый с переходом от Кобола к SQL». Это связано с тем, что в доля структурированных данных становится всё меньше, всё более остро встает вопрос решения аналитических задач по таким данным, а для реляционных БД «замечена деградация производительности при аналитической обработке и длинных SQL-запросах».

RDF имеет абстрактный синтаксис, который отражает модель данных на основе графов, а формальная семантика с детально описанным понятием отношения следствия предоставляет базис для достоверных логических рассуждений над RDF-данными [187, 257, 263, 272]. Предполагается, что RDF должен отвечать следующим целям:

- наличие простой модели данных;
- наличие формальной семантики и доказуемого логического вывода;
- использование расширяемых словарей на основе уникального идентификатора ресурса (англ., Uniform Resource Identifier (URI)) в соответствии с соглашением об уникальности имен в дескрипционной логике;
- использование синтаксиса на основе XML;
- поддержка использования типов данных XML-схемы;
- возможность любому делать объявления о любом ресурсе.

RDF может использовать значения, представленные в соответствии с типами данных XML-схемы, что способствует обмену информацией между RDF и другими XML-приложениями.

Модель данных в основе выражений в RDF предполагает создание коллекций триплетов, каждый из которых состоит из субъекта, предиката и объекта:

$$\langle \textit{Subject}, \textit{Predicate}, \textit{Object} \rangle. \quad \text{П.}(3.1)$$

Набор таких триплетов называется RDF-графом. Триплет может быть проиллюстрирован с помощью диаграммы узлов и направленных дуг (см. рис. П. 3.1). Можно заметить, что такие триплеты по своей структуре в точности соответствуют утверждениям в системах фактов \mathcal{A} онтологий в соответствии с (2.3) и (2.4).

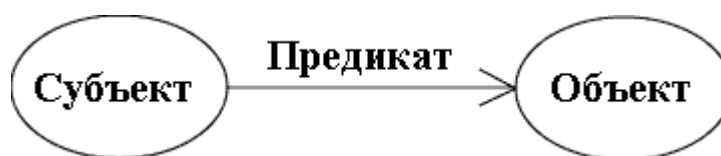


Рис. П. 3.1. Типовой граф триплета модели данных RDF

URI-ссылка и, так называемый, литерал (обозначение индивида) используются как идентификаторы узлов и предикатов.

В Semantic Web принято, чтобы эти идентификаторы имели вид URI – универсальных идентификаторов ресурсов, стандартизированных для сети Интернет. Таким образом, все идентификаторы в онтологических моделях получают следующий вид [48]:

http://имя-хоста/онтология#идентификатор_ресурса. П.(3.2)

Литералы используются, чтобы идентифицировать значения типов данных с помощью лексического представления. Все, что представлено литералами, также может быть представлено с помощью URI, но интуитивно, часто более уместно использовать литералы. Литерал может быть объектом RDF-объявления, но не субъектом или предикатом.

Таким образом, графовая модель RDF позволяет хранить утверждения, связанные с онтологиями. При этом, посредством использования URI она определяет глобальную идентификацию ресурсов в открытом информационном пространстве, и за формируемыми словарями и определяемыми предикатами (читай, ролями онтологии) может скрываться определённая семантика, фактически определяющая новые информационные модели [187]. Все эти особенности отражаются на подходах к отображению данной модели в каноническую модель предметных посредников. С моделью RDF неразрывно связаны модель RDF-Schema (RDFS), предоставляющая средства определения схем данных для RDF-описаний, а также широко распространённый ЯЗ SPARQL (среди прочих альтернатив, таких как DQL, N3QL, R-DEVICE, RDFQ, RDQ, RDQL, SeRQL), основанный на использовании триплетов в запросах. Фактически, они вместе определяют ЯОД и ЯМД. В результате, модель RDF, как

способ описания \mathcal{A} , является базисом для других моделей, характеризующих \mathcal{T} . В частности, язык Ontology Web Language (OWL) с определёнными семантикой и набором правил вывода в одном из своих синтаксисов основывается на RDF, и данные, представленные в модели OWL, обретают вид RDF-ресурсов. Для поиска данных в источниках, представленных RDF-хранилищами, служат специализированные поисковые механизмы, обращение к которым возможно с помощью языка SPARQL [39]. В качестве примера простейшего SPARQL-запроса можно привести следующий:

*SELECT * WHERE { ?a ?b ?c }.* П.(3.3)

Выполнение SPARQL-запроса – поиск ребер графа, удовлетворяющих триплетам в запросе. При этом запрос может быть обращен к нескольким источникам в соответствии с указанными URI, и, в таком случае, поиск осуществляется по гиперграфу фактов, где подграфы источников объединяются по заданным в запросе ограничениям. Далее будет дано более подробное пояснение. Синтаксически запрос записи на ЯЗ SPARQL аналогично представляется тройками литералов конкретных значений:

INSERT DATA {x y z}. П.(3.4)

OWL имеет несколько диалектов в соответствии с диалектами дескрипционной логики: редко используемый, упрощенный OWL Lite, часто применяемый OWL DL (Description Logic), и богатый по выразительным возможностям OWL Full [48]. Одно из различий между OWL DL и OWL Full состоит в том, что для OWL DL гарантируется вычислимость любого логического выражения, а для OWL Full – нет. Далее для создания предметных онтологий O^d будет использоваться разновидность диалекта OWL DL – DL-Lite [17].

Замечание П. 3.1. Так называемый стек языков Семантического Веб [76, 256], опирающийся на аппарат дескрипционной логики, по существу определяет гносеологические уровни представления данных, информации и знаний, начиная от представления элементов данных, их идентификаторов URI и модели данных RDF и вплоть до моделей логических рассуждений средствами

языка OWL и механизмов логического вывода (см. рис. П. 3.2, приложение 2.3.1).

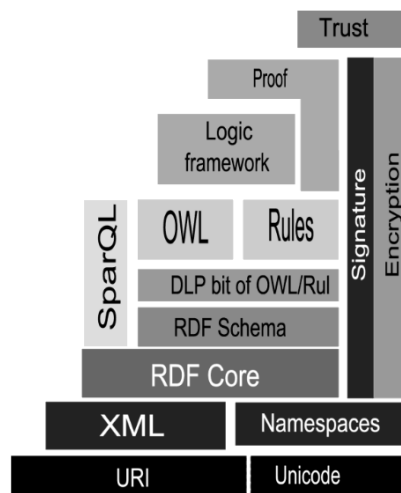


Рис. П. 3.2. Иерархия языков Семантического Веб по уровням представления знаний

3.3. Система интеграции данных и приложений на основе предметных онтологий

3.3.1. Обоснование необходимости разработки новой системы интеграции данных и приложений

На практике информационные ресурсы, которые подлежат интеграции и совместной обработке в рамках интеграции приложений, могут быть представлены с помощью структурных моделей данных типа XML, JSON, в сетевых БД, часто называемых NoSQL (Not only SQL) БД, их разновидностях в виде графовых, объектно-ориентированных или документо-ориентированных схем БД [187]. Однако наиболее распространенным случаем является использование реляционных БД, позволяющих хранить и запрашивать большие объемы данных, характерных и для рассматриваемой Про [36]. При этом семантический доступ к таким информационным ресурсам затруднен.

В соответствии с рассмотренным выше, задание семантических онтологических отображений на такие источники данных предполагает использова-

ние модели данных RDF и задания отображения концептуальных схем источников на модель RDF на основе триплетов «субъект-предикат-объект». Для реализации такого отображения необходимо разработать модель адаптации схем источников данных на структуру RDF и обеспечить обработку SPARQL-запросов на доступ и обновление данных [17, 36, 75].

На сегодняшний день существует немало примеров разработанных систем, использующих онтологический подход для интеграции данных. К их числу можно отнести SYNTHESIS, разрабатываемый в ИПИ РАН, AutoMed, Infomaster, Information Manifold, PICSEL, SIRUP [37]. Они относятся к классу систем, реализующих LAV или GAV подходы. В [27] предлагается алгоритм построения результирующей онтологии из нескольких исходных. Вопрос меры близости концептов по их содержательному значению предлагается решать с помощью генетического алгоритма и определения весовых коэффициентов близости. Такой подход оставляет высокую меру субъективности и при применении в автоматическом режиме зачастую неверно соотносит концепты по их эквивалентности, что и отражено в результатах исследования. В [8] предлагается вручную связывать концепты онтологий локальных источников данных. В работах [17, 18] рассматривается формальная модель интеграции данных на основе построения результирующей онтологии по множеству источников данных в виде реляционных БД, для которых построены виртуальные онтологические отображения. Пользователь не взаимодействует с источниками напрямую, а обращается к выделенному компоненту-посреднику, который отвечает за обслуживание пользовательских запросов и взаимодействие с источниками. При этом предполагается, что определения терминов в источниках данных не пересекаются. Пользователь формулирует свои запросы в терминах глобальной схемы данных (схемы данных посредника) (GLAV-подход), которая проектируется для системы интеграции исходя из интересующих пользователя аспектов ПрО. Основной задачей ставится переформулировка запросов в системе интеграции данных на основе онтологий, т. е. построение на основе пользовательского запроса, заданного в терминах глобальной схемы данных,

распределенного запроса к источникам данных (из SPARQL в SQL). Результаты анализа показали, что описание онтологии для обращения к источникам данных на диалекте DL-Lite_A позволяет переформулировать запросы с приемлемой вычислительной сложностью. В [26, 36] предлагается смежный подход, однако к его недостаткам можно отнести необходимость внедрения в источники данных специализированных хранимых процедур и таблиц, хранящих RDF-триплеты, что нарушает требования, предъявляемые к OLAP-системам (см. разделы 1.2.3, 1.2.4, 1.4.1).

Кроме того, проблема семантической интеграции данных на основе представленных GAV, LAV и GLAV подходов с проектированием обобщенной канонической модели Про предполагает, что однозначным образом могут быть интерпретированы различные лексико-семантические отношения между определениями сущностей и атрибутов различных таблиц. Интеграция не связанных между собой БД приводит к необходимости преодоления неоднозначности, связанной с синонимией и антонимией понятий и их атрибутов, их мимонимией и холонимией (входимости концептов) и ряда других сложных семантических несоответствий, которые подробно рассмотрены в [21, 24]. Также проблема отображения (интеграции) онтологий заключается в том, что [99, 176]:

- сущности (классы, свойства, связи, объекты) имеющие одинаковые имена могут иметь разный смысл и разные имена;
- сложность трансляции онтологических запросов в запросы исходных СУБД в GLAV-подходе;
- трансляцию между двумя локальными онтологиями предлагается осуществлять с помощью последовательного обобщения, поиска наиболее общего концепта (гиперонима). При этом возникает вопрос меры обобщения, поскольку при чрезмерном обобщении информацию атрибутов исходных концептов можно потерять;

– в рамках рассмотренных подходов рассматривается только проблема ЕП, но никак не рассматривается проблема интеграции приложений ЕАИ, т.е. совместной обработки данных, взятых из различных источников, различными приложениями.

Таким образом, существующие системы позволяют осуществлять интеграцию данных на основе онтологий, предоставляя семантический доступ к источникам данных. Однако все они, так или иначе, вводят ряд существенных ограничений, вмешиваются в структуру источников данных, не позволяют в полной мере решить вопрос семантической неоднородности терминов онтологий, а также не решают вопрос интеграции приложений. В то же время в рамках рассматриваемой Про требуется, не затрагивая источники данных, иметь доступ к данным в них, совместно обрабатывать эти данные в рамках приложений, связанных с организационными единицами, в которых находятся эти источники данных, обеспечивая, с одной стороны их конфиденциальность, а с другой – интегрированность для оценивания состояния СОТО на основе этих данных на всех этапах ЖЦ. В связи с этим требуется разработка такой модели (архитектуры) системы, которая на формальной основе обеспечивала бы возможность реализации этих свойств в соответствии с требованиями, представленными в разделе 1.4.1, описанием CASE-технологий в разделе 1.2.

3.3.2. Метод интеграции данных на основе правил отображения в онтологиях

В рамках настоящей работы был выбран подход проектирования онтологических отображений на источники данных, предложенный D. Calvanese и его соавторами, основанный на создании и использовании так называемого T-mapping для реляционных БД (см. далее) [228, 260]. Существенно, что указанный подход исходит из предположения, что задание отображения не должно влечь за собой изменения в структуре моделей и данных источников данных, а сам онтолого-управляемый доступ осуществляется пользователями с помощью онтологических запросов в терминах концептуального представления независимо от технологических решений в СУБД, причем обеспечивается автоматическая конвертация онтологического запроса в запрос, синтаксически

соответствующий ЯЗ, определенному для данной СУБД. Существенно также и то, что предложенный метод задает компромисс выразительных возможностей проектирования онтологических запросов (обеспечивается полнота при их конвертации в реляционные) и позволяет отвечать на запросы с вычислительной сложностью LogSpace от объема запрашиваемых данных и за полиномиальное время, в отличие от OWL-языков, приводящих к NP-сложности и не позволяющие хранить большие объемы данных в \mathcal{A} .

Суть метода D. Calvanese заключается в следующем. Предметная онтология O^d формируется на основе специального диалекта дескрипционной логики $\text{DL-Lite}_{\mathcal{A}}$. Введение этого диалекта связано с необходимостью представления непосредственно данных в онтологии, как это происходит в реляционных БД, поскольку в традиционных диалектах индивидами концептов являются только абстрактные сущности – объекты. Эта проблема называется несоответствием импеданса. Отличие $\text{DL-Lite}_{\mathcal{A}}$ от рассмотренного в разделе 2.1.3, заключается в том, что в дополнение к множеству концептов CN и ролей RN определено множество абстрактных атрибутов $AN = \{At_1, \dots, At_k\}$, которые определяют свойства (признаки) объектов. Определенный в разделе 2.1.3 алфавит D – множество констант, соответствующих известным типам данных, а $\text{val}(D_i)$ – непосредственно значение заданного типа данных. Причем отмечается, что $\delta(At)$ задает множество объектов, соотносимых со значениями отношением At , а $\rho(At)$ – множество значений, соотносимых с объектами, причем $\delta(At)$ и $\rho(At)$ не пересекаются. Семантика задается следующим образом. В области интерпретации Δ^I выделяются две непересекающиеся подобласти Δ^I_o и Δ^I_v , задающие области определений объектов и области значений соответственно. Тогда:

$$At^I \subseteq \Delta^I_o \times \Delta^I_v, \quad \text{П.(3.5)}$$

$$(\rho(At))^I = \{v \mid \exists o. (o, v) \in At^I\}, \quad \text{П.(3.6)}$$

$$(\delta(At))^I = \{o \mid \exists v. (o, v) \in At^I\}, \quad \text{П.(3.7)}$$

$$val(D_i) \cap val(D_j) = \emptyset, \quad \text{П.(3.8)}$$

причем следующие выражения являются концептами для \mathcal{T} :

$$\begin{aligned} C ::= & |\top_D| \rho(At) |\delta(At)| At |\neg At| D_i \sqsubseteq D_j | \\ & At_i \sqsubseteq At_j | funct(At) | funct(R), \end{aligned} \quad \text{П.(3.9)}$$

где $funct(R)$ и $funct(At)$ – функциональные роли и функциональные атрибуты соответственно, такие что:

$$(o_1, o_2) \in R^I; (o_1, o_3) \in R^I \Rightarrow o_2 = o_3, \quad \text{П.(3.10)}$$

$$(o, v_1) \in At^I; (o, v_2) \in At^I \Rightarrow v_1 = v_2. \quad \text{П.(3.11)}$$

В заданном диалекте набор фактов \mathcal{A} задается также, как определено в (2.3) и (2.4), однако дополнительно определено утверждение вида:

$$a At b, \quad \text{П.(3.12)}$$

где a – объект, b – тип данных, причем $a^I \in \Delta_O^I$ и $b^I = val(b), b^I \in \Delta_V^I$ – значение.

Замечание П. 3.2. Определения, заданные в диалекте DL-Lite_A, без иных дополнений позволяют создавать графовые БД в рамках онтологий вида $O^d = \langle CN^d, RN^d, AN^d, \mathcal{I}^d \rangle$, для которых набор терминологических аксиом \mathcal{T}^d определяет концептуальную схему БД, задающую структуру и ограничения моделируемой ПрО, а набор фактов \mathcal{A}^d является непосредственно ХД на основе модели данных, заданной в П.(3.1), при условии, что модель интерпретации \mathcal{I}^d выполнима относительно \mathcal{T}^d и \mathcal{A}^d .

К онтологии O^d можно осуществлять запросы на выборку данных или их обновление (вставку). Обновление по существу относится к ситуации добавления нового факта в \mathcal{A}^d и может рассматриваться в контексте проверки выполнимости концепта, представленной в разделе 3.2.1. Запрос на выборку – есть процедура извлечения фактов о ПрО из \mathcal{A}^d , и может быть задан следующим образом [17, 18, 72, 228, 260]. Пусть определено счетное множество индивидных переменных $Var = \{x_0, x_1, \dots\}$. Атомы (или атомарные запросы) –

это выражения вида $u: C$, $u R v$, $u At v$, $u = v$, где C – концепт, R – роль, At – атрибут, u, v – индивиды (из IN), переменные (из Var) или константы $val(D_i)$.

Определение П. 3.1. Конъюнктивный запрос к онтологии – это выражение вида $q(\vec{x}) \leftarrow \bigwedge_{i=1}^n A_i(\vec{x}, \vec{y})$, где $\vec{y} = (y_1, \dots, y_m)$ – конечный набор переменных из Var или констант (литералов), причем любой конъюнкт представляется атомом A_i , т.е. триплетом вида П.(3.1). Переменные в \vec{y} называются связными, $\vec{x} = (x_1, \dots, x_n)$ – список свободных переменных запроса q .

Число свободных переменных запроса q называется его размерностью и обозначается $ar(q)$. Запрос q истинен в БЗ \mathcal{K}^d , если $\mathcal{K}^d \models q$, то есть для любой интерпретации \mathcal{I}^d из $\mathcal{I}^d \models \mathcal{K}^d$ следует $\mathcal{I}^d \models q$. Также определен объединенный запрос UCQ , представимый в дизъюнктивной нормальной форме (ДНФ) запросов $q(\vec{x})$ [169]. Пример запроса представлен в П.(3.2).

Предложенный D. Calvanese и его соавторами подход предполагает задание отображения онтологического запроса из определения П. 3.1 в SQL-запрос к реляционной БД на основе набора фактов \mathcal{A}^d , заданного посредством диалекта DL-Lite_A, который по своей сути является виртуальным отображением реальных данных, представленных в реляционной БД и может быть определен как:

$$\mathcal{A}^d = \langle \mathcal{M}, \mathcal{DB} \rangle, \quad \text{П.(3.13)}$$

где \mathcal{DB} – реляционная БД, а \mathcal{M} – набор утверждений T-mapping, состоящий из двух подмножеств \mathcal{M}_t и \mathcal{M}_a правил вида:

$$\mathcal{M}_t: \Phi \rightsquigarrow D, \quad \text{П.(3.14)}$$

где Φ – реляционный SQL-запрос $ar(1)$ к \mathcal{DB} , D – некоторый тип данных, и

$$\mathcal{M}_a: \Phi \rightsquigarrow \psi, \quad \text{П.(3.15)}$$

где Φ – реляционный SQL-запрос $ar \geq 0$ к \mathcal{DB} , ψ – конъюнктивный запрос из определения П. 3.1 арности $ar' \geq 0$.

При этом содержательно элементы утверждений \mathcal{M} и элементы реляционной БД соотносятся так, как это представлено на рис. П. 3.3.

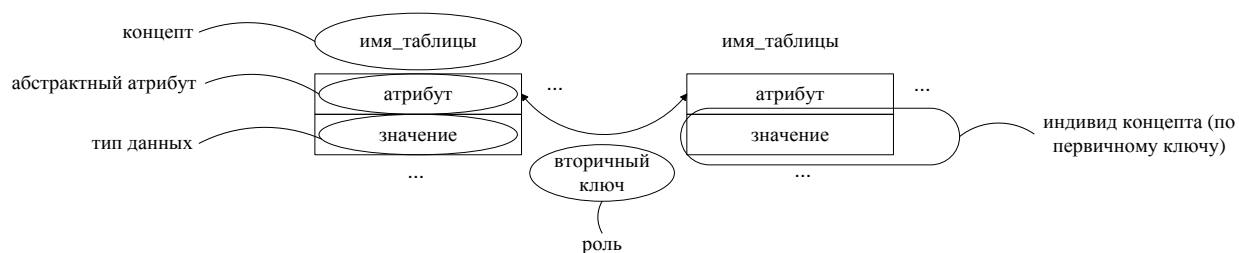


Рис. П. 3.3. Соотнесение элементов дескрипционной логики DL-Lite_A и элементов схемы реляционной БД

Для такой онтологии интерпретация \mathcal{I}^d называется её моделью, если она удовлетворяет всем утверждениям из \mathcal{M} относительно \mathcal{DB} . При этом налагаются следующие ограничения:

- типы данных реляционной БД и онтологии должны совпадать;
- реляционная БД должна соответствовать первой нормальной форме, чтобы у каждой таблицы был первичный ключ, поскольку индивидами концептов (таблиц) являются строки таблицы как объекты, идентифицируемые по записям атрибута, являющегося первичным ключом [11, 115];
- набор правил \mathcal{M} должен предполагать наличие всех необходимых SQL-запросов, обеспечивающих доступ к необходимым данным в \mathcal{DB} . Соответственно должны быть в наличии и все соответствующие им триплеты для проектирования онтологических запросов.

Процедура исполнения онтологического запроса на основе T-mapping, определенная в рамках некоторого медиатора (семантического предметного посредника) предполагает [228, 260]:

- предварительное автоматизированное построение набора правил \mathcal{M} по схеме БД (существующие СУБД предполагают возможность получения в запросе наборов таблиц, их атрибутов, первичных и вторичных ключей). Важно отметить, что изменение структуры в схеме БД требует перепроектирования \mathcal{M} ;
- разделение пользовательского запроса $q(\vec{x})$ на атомарные подзапросы;

- подстановку реляционных запросов Φ из \mathcal{M} с помощью перебора всех возможных вхождений переменных в атомарные подзапросы $q(\vec{x})$ на основе построения специализированного дерева вывода запроса [228, 260];
- сбор атомарных подзапросов $q(\vec{x})$ в виде их дизъюнкции (операция UNION в терминах SQL);
- исполнение сформированного запроса;
- возврат ответа в соответствии с переменными \vec{x} .

Необходимо отметить, что аналогичный подход на основе отображений предлагается для квазиструктурированных источников данных в составе АС, для которых выбирают сетевые и иерархические модели данных, используя такие ЯОД, как, например, XML (и другие языки разметки) или БД типа NoSQL [219, 247, 251]. В таком случае, задание отображения проводится более простыми методами по сравнению с T-mapping, однако по своей сути аналогичными, поскольку эти модели данных приближены к графовому представлению RDF (см. рис. П. 3.1) [224]. Пример соотнесения элементов XML и дескрипционной логики DL-Lite_A приведен на рисунке П. 3.4. Для реализации отображений NoSQL БД часто используется стандарт xR2RML, при этом создается две схемы: набора терминологических аксиом и набора фактов (с использованием языка RDF). В работе [231] приведены примеры для множества известных NoSQL СУБД (Cassandra, HBase, CouchDB, Hive), имеющих поддержку RDF-представления и исполнения SPARQL-запросов (например, с помощью системы Apache Jena). Существенно, что зачастую именно эти модели данных выбираются для хранения таких метаданных, как электронные документы, спецификации 3D-моделей, совокупности параметров АСУ, SCADA и других АС. В связи с этим необходимо осуществлять интеграцию данных из таких источников, обеспечивая доступ к их семантике с помощью онтологических отображений.

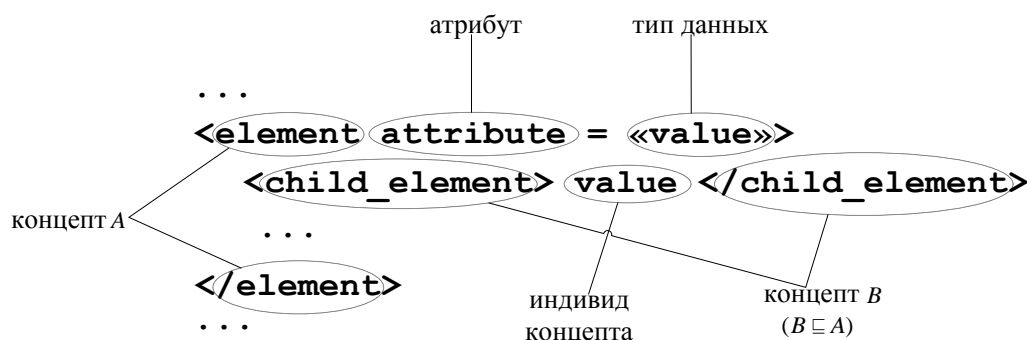


Рис. П. 3.4. Соотнесение элементов дескрипционной логики DL-Lite_A и элементов схемы XML

Таким образом, обеспечивается онтолого-управляемый доступ к информационным ресурсам источников данных в терминах концептуальной схемы без вмешательства в структуру схем источников данных и сами данные.

3.3.3. Архитектура системы интеграции данных и приложений, ориентированная на решение задачи информационно-аналитической поддержки ЖЦ СОТО

Рассмотренные в предыдущих разделах аспекты онтологических отображений на источники данных позволяют выявить следующие утверждения.

D. Calvanese отмечает, что предложенный метод решает вопрос только онтолого-управляемого доступа к БД, что позволяет решать вопрос интеграции данных на основе концептуального терминологического представления Про с помощью онтологий O^d , но никак не затрагивает вопросы интеграции приложений (EAI), тесно связанной с задачей интеграции данных [228, 260].

Наличие возможности создавать онтологические отображения на источники данных позволяют в терминах таких концептуальных схем $Sk = \{sk_h \mid h = 1, \dots, t\}$ (см. постановку задачи, раздел 1.4.3) осуществлять запросы к источникам данных. При этом совокупность запросов позволяет в рамках единого вычислительного процесса осуществлять сбор данных из разных источников, их обработку и анализ, и в этих же терминах запись результатов обработки. При этом отсутствует необходимость в рамках онтологических описа-

ний преодолевать семантическую неоднородность данных, поскольку этот вопрос может решаться непосредственно в рамках вычислительных задач, явно определяя функциональные (вычислимые) отношения между ними необходимого вида. При этом задано ограничение, что один запрос может быть обращен к одному источнику данных в отличие от традиционных онтологических запросов, предполагающих возможность выборки из нескольких источников одновременно.

С учетом замечания П. 3.2 может быть создана специальная онтология со своей схемой sk_h^{ep} , куда могут записываться факты \mathcal{A}^d , как результаты обработки информации, собираемой из различных источников данных. Иначе говоря, информация организационных единиц предприятий, неявно характеризующая состояние СОТО в виде соответствующих электронных документов, их метаданных, данных из различных БД, данных результатов функционирования АС может быть обработана и проанализирована по их совокупности с целью выявления показателей ТС и надежности (эксплуатационной пригодности по определению 1.1), которые в свою очередь могут записываться в единое хранилище, по своей концептуальной схеме sk_h^{ep} соответствующее представлению электронного паспорта изделия.

При условии наличия программы вычислений, обеспечивающей вычисление задач выборки, обработки, анализа данных и записи результатов в потоковом режиме, хранилище электронного паспорта может накапливать информацию о состоянии на протяжении всего ЖЦ СОТО, обеспечивая на каждом этапе потребителей актуальной информацией о состоянии СОТО, с одной стороны, и поддерживая хронологию изменения состояния, с другой. Заметим, что множество операторов модели программы можно разделить на 3 подмножества:

$$M^{Op} = \{M^{qR}, M^{ctT}, M^{qW}\}, \quad \text{П.}(3.16)$$

где M^{q^R} – конечное множество операторов выбора источника данных данных и исполнения запроса выборки, M^{ct^T} – конечное множество операторов обработки промежуточных данных, M^{q^W} – конечное множество операторов исполнения запросов записи результатов обработки.

Кроме того, такой подход обеспечивает соответствие требованиям OLAP-систем (см. раздел 1.2.3), т.е. достигается сохранность и целостность данных источников, автоматический выбор источника данных для запроса (см. ниже), формирование запросов в терминах ПрО, реализация *ETL-принципа*, обеспечение вычислительной системы витринами данных как метамоделями источников данных.

Для обеспечения конфиденциальности собираемых и обрабатываемых данных, а также для содержательного разделения процессов обработки и анализа в соответствии с реальными задачами, специфичными для каждой организационной единицы ЖЦ СОТО программ вычислений должно быть несколько, соответственно количеству организационных единиц. При этом должна обеспечиваться их интегрированность (*интероперабельность* процессов), которая может достигаться взаимодействием программ через хранилище электронного паспорта, т.е. результаты анализа ТС СОТО одного этапа ЖЦ могут быть использованы в качестве основы для инициализации решения задач на другом этапе в соответствии с реальными процессами информационно-аналитической поддержки ЖЦ СОТО (см. раздел 2.4.6).

С учетом отмеченных фактов, типовая архитектура системы интеграции данных и приложений представлена на рис. П. 3.5.

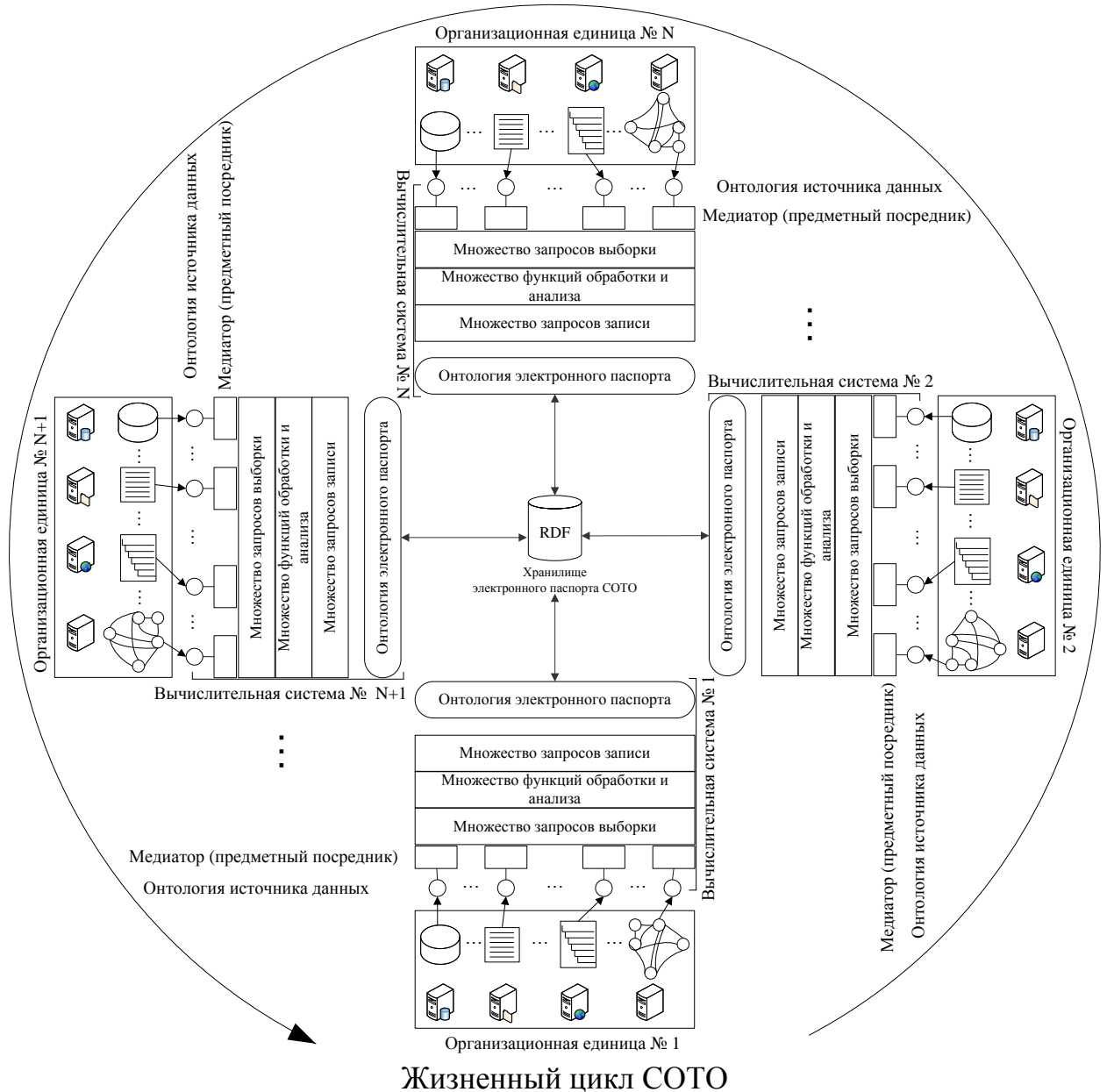


Рис. П. 3.5. Схема архитектуры системы интеграции данных и приложений для задачи информационно-аналитической поддержки ЖЦ COTO

3.3.4. Алгоритм выбора источника данных и исполнения онтологических запросов

Для реализации рассмотренной выше архитектуры, в частности, необходимо обеспечить ПК такой вычислительной процедурой исполнения онтологических запросов, которая позволила бы осуществлять эти запросы к соответ-

ствующему источнику данных (СУБД или файлу) за счет их конвертации в термины необходимого ЯЗ на основе медиаторов, как это было рассмотрено выше, и затем исполнять конвертированные запросы.

Тогда, пусть заранее имеется конечное множество спецификаций (концептуальных схем источников данных) M^{Sk} в паре с их наименованиями n в соответствии с URI (см. П.(3.2)) и с наименованиями посредника-медиатора (вычислительной процедуры, определенной для заданного типа источника данных – реляционной БД, NoSQL БД, XML и т.д.), т.е. $S_h = \{sk_h, n_h, mt_b\}$, причем $Mt = \{mt_b | b = 1, \dots, n\}$ по количеству типов используемых моделей данных. Пусть также имеется некоторый онтологический запрос $query$, определенный в рамках вычислительной системы в соответствии с определением П. 3.1. Заметим, что исполнение онтологического запроса к RDF-хранилищу предполагает, что $S_h = \{sk_h^{ep}, n_h, \emptyset\}$, поскольку для традиционного онтологического запроса отсутствует необходимость в конвертации [228]. В таком случае, алгоритм выбора источника данных может быть представлен в виде следующей схемы (см. рис. П. 3.6).

Суть работы алгоритма заключается в нахождении схемы sk_h , связанной по URI с онтологическим запросом (функцией $getURI$) и запуске одной из процедур-медиаторов mt_b_exec для конвертации и исполнения запроса. Очевидно, что размерность вектора \vec{x} аналогична арности запроса $query$. Также очевидно, что оператор SWITCH по ветви “default” должен исполнить обычный онтологический запрос без его конвертации процедурой $sparql_exec$ известным способом (описанным в [229]) к некоторому RDF-хранилищу, поскольку для него не может быть найдено соответствующее онтологическое отображение к источнику данных. В случае, если наименование источника данных n_h не было найдено, алгоритм, очевидно, должен исполнить обычный онтологический запрос и завершить работу с возвратом результата исполнения запроса.

В то же время для запросов записи результатов не предполагается выбора источника данных, поскольку запись осуществляется в единственное хранилище.

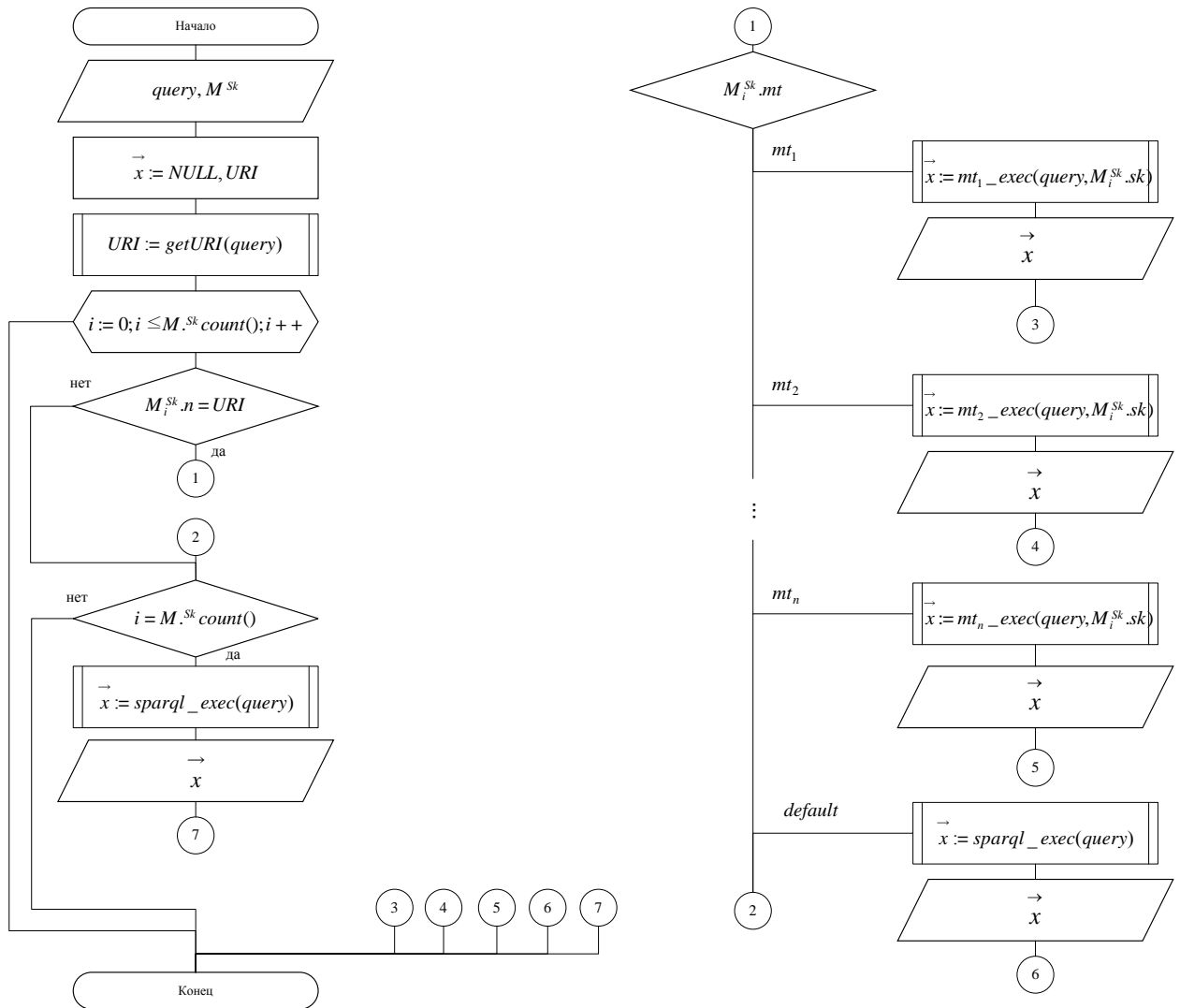


Рис. П. 3.6. Блок-схема алгоритма выбора источника данных и исполнения запроса

3.3.5. Модель состояния СОТО на основе онтологического подхода

Рассмотренная архитектура, как уже было отмечено, предполагает наличие отдельного информационного ресурса, предназначенного для сохранения результатов сбора, обработки и анализа данных. В то же время в соответствии с рассмотренным в разделе 1.2.1, задача информационной поддержки ЖЦ

СОТО предполагает решение подзадачи информационной поддержки управления конфигурацией изделия, где информационное представление конфигурации (часто называемое электронной структурой изделия) по существу определяется некоторой моделью состояния СОТО, причем совокупность конкретных значений, заданных в рамках модели, называется электронным паспортом СОТО. Из определения П. 1.3, и описания конфигурации в приложении 1.5.3 известно, что конфигурация представляется в виде некоторого древовидного или сетевого графа и предполагает наличие атрибутов, явно задающих качественные характеристики и эксплуатационную пригодность изделия (оценки ТС). С вершинами графа могут быть связаны метаданные документов, показатели функционирования связанных с изделием АС и результаты анализа этих данных в виде значений показателей соответствующих свойств (характеристик) объектов (изделий, процессов, ресурсов, документов). Стоит отметить, что для *OLAP-систем* зачастую характерно в составе хранилищ результатов обработки собирать и данные из источников (например, сами метаданные документов) для обеспечения полноты описания состояния СОТО [11].

Утверждение П. 3.1. Использование диалекта дескрипционной логики $DL-Lite_A$ с учетом замечания П. 3.2 позволяет задать концептуальную схему электронного паспорта СОТО sk_n^{ep} в соответствии с определением понятия конфигурации изделия в приложении 1.5.3.

Состояние системы определяется как множество всех свойств образующих ее объектов в определенный момент времени. Совокупность всех допустимых состояний системы, задаваемое областями определения концептов, рассматривают как пространство состояний, иначе называемое фазовым пространством. Таким образом, исполнение программы можно трактовать как движение фазовой точки в многомерном фазовом пространстве, где каждая ось отображает значения одного из свойств системы. Поскольку анализ движения точки в многомерном пространстве состояний является нетривиальным, принято выделять переменные состояния [198, 199, 200, 234]. В зависи-

мости от решаемой задачи и типа изделия модель состояния СОТО будет отличаться наборами конкретных атрибутов и понятий, связанных с рассматриваемым изделием. Т.е. для конкретной системы СИАП ЖЦ СОТО модель должна задаваться экспертами в виде соответствующей схемы sk_n^{ep} как набора терминологических аксиом предметной онтологии O_{ep}^d . Однако можно выделить некоторые общие черты, характерные для конфигураций. Рассмотрим случай, когда состояние изделия оценивается посредством набора документов и их атрибутов (метаданных). В соответствии с ГОСТ Р ИСО 15489-1–2007, в аспекте цифровизации документов «метаданные – это данные, описывающие контекст, содержание, структуру документов», а оцифровка документов с доступом к семантике их данных есть процесс атрибутирования, т.е. выявления метаданных [102]. Тогда модель состояния (электронного паспорта) СОТО может быть задана следующим образом.

Пусть в рамках онтологии O_{ep}^d задан набор концептов $CN_{ep}^d = \{Product, Product_element, Document, State\}$, где *Product* определяет конечное множество изделий, связанных с ПрО *SubjectArea*, *Product_element* – конечное множество составных элементов (узлов и агрегатов изделия), связанных с объектами ПрО *Object*, *Document* – конечное множество документов, появление которых характеризует выполнение этапов работ с изделием и содержательно отражающих результаты этих работ, причем в таком контексте можно говорить, что документы отражают явления в ПрО $Document \sqsubseteq Phenomenon$, *State* – конечное множество статусов документов, совокупность атрибутов которых определяют показатели ТС СОТО. Пусть также задано множество ролей $CN_{ep}^d = \{ConsistsOf, Describes, Evaluates\}$, где *ConsistsOf* – определяет отношение входимости узлов в изделие, *Describes* – задает отношение между документами и узлами изделия, *Evaluates* – задает отношение между статусами документов и самими документами. Пусть также задано множество атрибутов $AN_{ep}^d = \{Id, Parent_Id, Doc_At, State_At\}$, где *Id* – идентификатор узла изделия, *Parent_Id* – идентификатор родительского

узла, пара атрибутов $\{Id, Parent_Id\}$ позволяет формировать структуру изделия, Doc_At – множество атрибутов документов (различны в разных ПрО), $State_At$ – множество атрибутов статусов документов (различны в разных ПрО). Пусть также задано множество типов данных $D = \{xsd:integer, xsd:string\}$ (см. приложение 2.3.8.2). Тогда набор терминологических аксиом \mathcal{T}_{ep}^d может быть задан следующими выражениями:

$$Product \equiv \exists ConsistsOf.Product_element, \quad \text{П.}(3.17)$$

$$Product_element \equiv \exists Id.xsd:integer \quad \text{П.}(3.18)$$

$$\sqcup \exists Parent_Id.xsd:integer,$$

$$Document \equiv \exists Describes.Product_element \sqcap \quad \text{П.}(3.19)$$

$$\exists Doc_At_j.xsd:string,$$

$$State \equiv \exists Evaluates.Document \sqcap \quad \text{П.}(3.20)$$

$$\exists State_At_i.xsd:string.$$

В соответствии с указанными выражениями схема электронного паспорта представлена на рисунке П. 3.7. Необходимо отметить, что данная схема представлена не как общезначимая, а как один из вариантов описания модели состояния СОТО и предназначена для того, чтобы показать принципиальную возможность описания конфигурации изделия на основе её структуры и совокупности атрибутов. Очевидно, что необходимо задать конкретные атрибуты в выражениях, что можно иначе представить электронную структуру изделия. Также необходимо заметить, что предложенная структура (и при дальнейшей детализации информационных сущностей, их атрибутов) соответствует схеме «звезды», используемой в рамках *OLAP-систем* и представима в виде *гиперкуба*, операции на котором позволяют получать информацию о состоянии СОТО в интересующем пользователя аспекте (см. раздел 1.2.3).

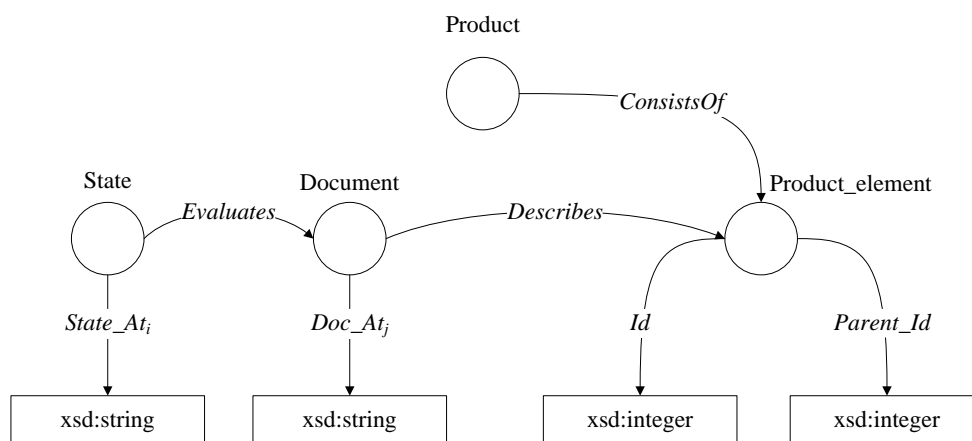


Рис. П. 3.7. Типовая схема электронного паспорта СОТО на основе онтологического подхода

Таким образом, концептуальная схема в RDF-хранилище может соответствовать структуре конфигурации изделия sk_h^{ep} в совокупности с атрибутами, характеризующими ТС СОТО. Наполнение ХД есть наполнение системы фактов \mathcal{A}_{ep}^d посредством исполнения запросов записи типа П.(3.4).

3.4. Выводы по разделу

Обоснована необходимость разработки и предложена типовая архитектура системы интеграции данных и приложений в качестве основы для СИАП ЖЦ СОТО, позволяющая обеспечить интегрированность, конфиденциальность, сохранность данных из интегрируемых источников, реализуемость ETL-принципа в соответствии с принципами функционирования OLAP-систем в рамках CASE-технологий, что позволяет осуществлять сбор и обработку данных, формируемых в рамках организационных единиц СОТО на всех этапах ЖЦ, и поддерживать хронологию изменения состояния СОТО как результатов обработки данных в рамках единого хранилища электронного паспорта.

Разработан алгоритм выбора источника данных для онтологических запросов, позволяющий, с одной стороны, исполнять пользовательские запросы, описанные в рамках единой нотации ЯЗ независимо от модели данных и СУБД

источников с использованием метода онтолого-управляемого доступа к данным, а с другой избавиться от необходимости согласования онтологий разных источников в части их семантической неоднородности.

Предложена типовая модель состояния СОТО как концептуальная схема информационного ресурса электронного паспорта СОТО, ориентированная на представление конфигурации изделия и его состояния. Обоснована принципиальная возможность использования для этой цели диалекта дескриптивной логики DL-Lite_A в ориентации на описание совокупности свойств, атрибутов и характеристик СОТО, явно определяющих его ТС и эксплуатационную пригодность.

4. НЕКОТОРЫЕ ЭЛЕМЕНТЫ СИСТЕМЫ ИНФОРМАЦИОННО-АНАЛИТИЧЕСКОЙ ПОДДЕРЖКИ ЖИЗНЕННОГО ЦИКЛА РАКЕТЫ-НОСИТЕЛЯ «СОЮЗ-2»

4.1. Проектирование онтологии в OWL-формате на основе реляционной базы данных

Для автоматической генерации онтологии по структуре реляционных БД, выступающих источниками данных для СИАП ЖЦ СОТО, был создан специального вида запрос, учитывающий синтаксис OWL и RDF, и осуществляющий перевод:

- каждой таблицы БД в соответствующий концепт онтологии;
- каждого первичного ключа таблицы в отношения принадлежности значений атрибута первичного ключа (индивидов) «концепту таблицы»;
- каждого атрибута таблицы в роль, связывающую концепты со значениями атрибутов.

При этом каждому элементу онтологии назначается уникальное имя в соответствии с URI. Множество значений любого атрибута связывается с соответствующим типом данных.

Текст запроса представлен ниже.

Таблица П. 4.1. Текст SQL-запроса, позволяющего по реляционной БД сформировать схему её онтологии в терминах OWL

```
SELECT '<?xml version="1.0"?>
<rdf:RDF xmlns="http://www.evep.org/"
  xml:base="http://www.evep.org/"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
  <owl:Ontology rdf:about="http://www.evep.org/">' ||
array_to_string(array(
/*ObjectProperty*/
```

```

SELECT '<owl:ObjectProperty rdf:about="http://www.evep.org/" || CASE WHEN tc.constraint_schema != 'public'
THEN tc.constraint_schema || '.' ELSE " END || tc.table_name || '#ref-' || kcu.column_name || "'>' || E'\n' ||
E'\t' || '<rdfs:domain rdf:resource="http://www.evep.org/" || CASE WHEN ccu.table_schema != 'public' THEN
ccu.table_schema || '.' ELSE " END || ccu.table_name || "'/>' || E'\n' ||
E'\t' || '<rdfs:range rdf:resource="http://www.evep.org/" || CASE WHEN tc.constraint_schema != 'public' THEN
tc.constraint_schema || '.' ELSE " END || tc.table_name || "'/>' || E'\n' ||
'</owl:ObjectProperty>'
FROM information_schema.table_constraints AS tc
    JOIN information_schema.key_column_usage AS kcu ON tc.constraint_name = kcu.constraint_name
    JOIN information_schema.constraint_column_usage AS ccu ON ccu.constraint_name = tc.con-
straint_name
WHERE tc.constraint_type = 'FOREIGN KEY'
UNION
/*DatatypeProperty*/
SELECT '<rdfs:Datatype rdf:about="http://www.evep.org/" || sch || tbl || '#' || clm || ':' || type || "'>' || E'\n' ||
'<owl:DatatypeProperty rdf:about="http://www.evep.org/" || sch || tbl || '#' || clm || "'>' || E'\n' ||
E'\t' || '<rdfs:domain rdf:resource="http://www.evep.org/" || sch || tbl || "'>' || E'\n' ||
E'\t' || '<rdfs:range rdf:resource="http://www.evep.org/" || sch || tbl || '#' || clm || ':' || type || "'>' || E'\n' ||
'</owl:DatatypeProperty>'
FROM (
SELECT CASE WHEN tbl.table_schema != 'public' THEN tbl.table_schema || '.' ELSE " END as sch, tbl.ta-
ble_name as tbl, clm.column_name as clm,
    CASE WHEN clm.data_type = 'integer' THEN 'integer'
    WHEN clm.data_type = 'text' THEN 'string'
    WHEN clm.data_type = 'bigint' THEN 'long'
    WHEN clm.data_type = 'character varying' THEN 'string'
    WHEN clm.data_type = 'timestamp without time zone' THEN 'dateTimeStamp'
    WHEN clm.data_type = 'smallint' THEN 'integer'
    ELSE 'other'
    END as type
FROM information_schema.tables tbl
    , information_schema.columns clm
WHERE tbl.table_schema NOT IN ('information_schema','pg_catalog')
    AND clm.table_schema = tbl.table_schema AND clm.table_name = tbl.table_name
) T
UNION
/*Class*/
SELECT '<owl:Class rdf:about="http://www.evep.org/" || CASE WHEN table_schema != 'public' THEN ta-
ble_schema || '.' ELSE " END || table_name || "'>' || E'\n'
FROM information_schema.tables
WHERE table_schema NOT IN ('information_schema','pg_catalog')
), ' ') || '</rdf:RDF>'

```

4.2. Схема базы данных системы управления электронными данными

Схема реляционной БД СУЭД, для которой была построена онтология (см. рис. 4.16), представлена ниже.

СУЭД предназначена для выполнения задач атрибутирования электронных документов посредством идентификации существенных с точки зрения ТС СОТО метаданных (реквизитной и содержательной частей документа) в виде специальных электронных карточек регистрации и помещения этих метаданных и документов в систему управления электронными данными. Так, например, в схеме БД можно увидеть такие таблицы (и их атрибуты) как этап ЖЦ, заводской номер экземпляра изделия, департамент и т.д., играющие существенную при формировании электронного паспорта изделия.

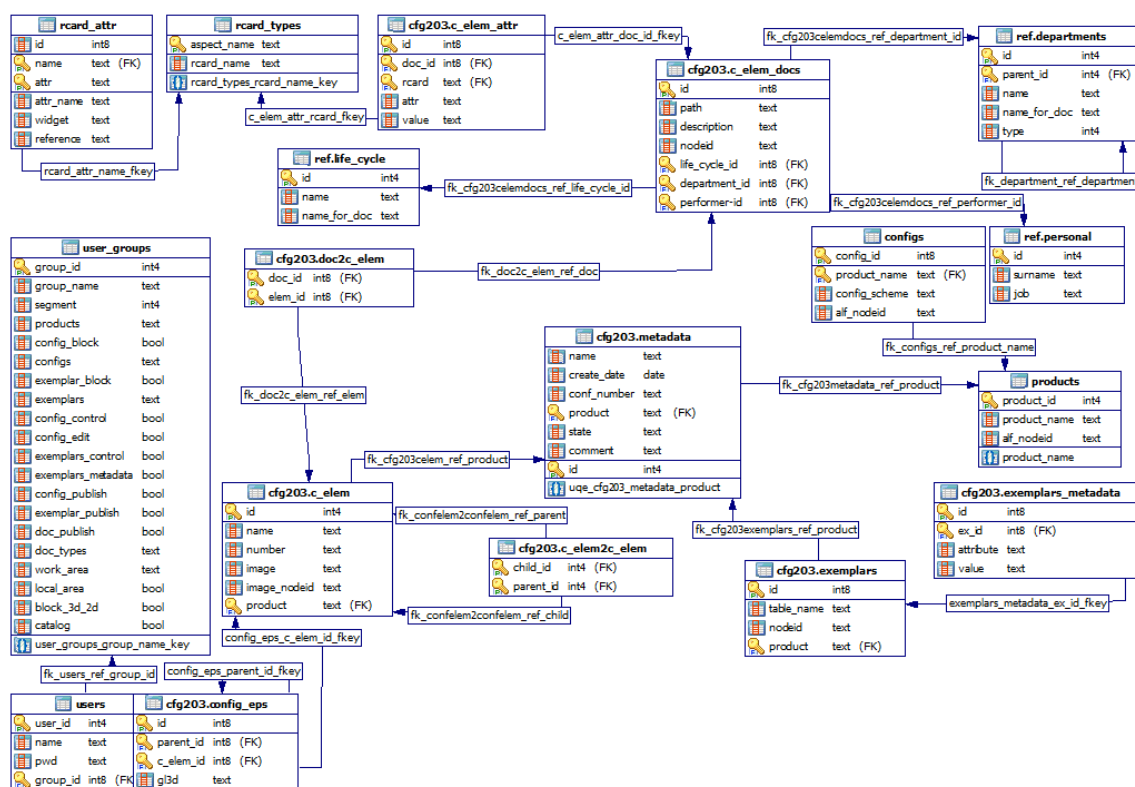


Рис. П. 4.1. Схема БД СУЭД

ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ

С

CALS (Непрерывная информационная поддержка
жизненного цикла продукции) · 33

CALS-технология · 33, 270

CASE-система · 50, 332

Е

ETL-принцип · 39, 171, 176, 397

Г

G-модель · 58, 112, 115, 305, 345, 351

G-сеть · 127, 306

О

OLAP · 26, 37, 38, 267, 278, 401, 403

А

активность · 164, 169

Б

база данных · 27, 271

база знаний · 342

бизнес-процесс · 72, 97, 288

большие данные · 264

В

верификация · 36, 47, 144, 307, 319, 324, 342

выполнимость · 77, 79, 145

вычислительная задача · 117

Г

гиперкуб · 40, 312, 403

гомоморфизм · 145

Д

дерево достижимости · 169

дефицит · 155

динамическая система · 299

достоверность · 179

Е

единое информационное пространство · 24, 171

Ж

жизненный цикл АС · 325

жизненный цикл изделия · 32

жизненный цикл ПО · 325

И

избыточность · 155

интеграция данных · 42

интегрированная информационная среда · 32

интегрированность · 172, 174

интероперабельность · 27, 59, 73, 140, 174, 175, 293,
343, 397

интерпретация · 77

информационная система · 25

информационно-аналитическая поддержка · 19, 27, 28

информационно-аналитическая система · 26

К

качество продукции · 284
 квалитетрия · 144, 324
 конфигурация · 281
 концептуальное моделирование · 47, 51, 74, 306, 334
 конъюнктивный запрос · 392
 корпоративная информационная система · 26, 268
 корректность · 164, 167
 космическое средство · 22

М

метаданные · 24, 266
 метод · 329
 методология · 329
 модель · 78, 353
 мониторинг · 35, 300
 мультиагентная система · 136, 273, 361

Н

надежность · 22
 неизменчивость · 172
 неоднозначность · 155
 непротиворечивость · 164, 165
 неразличимость · 155

О

ограниченность позиции G-сети · 169
 онтологическая выразительность · 145, 154
 онтологическая система · 59, 81, 143, 358
 онтология · 53, 74, 266, 317, 357
 онтолого-управляемое моделирование · 74
 оперативность · 179

П

параллельный инжиниринг · 279, 286
 поддержка хронологии · 172, 176

полная корректная G-сеть · 164
 полный непротиворечивый · 148
 поток работ · 269
 предметная ориентированность · 171, 173
 проект · 280
 противоречивый · 148

Р

реализуемость ЕИП · 171

С

СВЗ-модель · 99, 101
 система информации о техническом состоянии и
 надежности · 22, 26
 система мониторинга · 35, 301
 система фактов · 78
 системное моделирование · 50, 333
 сложный организационно-технический объект · 262
 согласованность · 145
 состояние динамической системы · 300
 спецификация · 47, 49, 73, 319
 СПМ-распознавание · 113, 373

Т

табло-алгоритм · 146
 терминологическая аксиома · 78
 техническое состояние · 35, 300
 тупиковая позиция G-сети · 169

У

уровень информационно-аналитической поддержки ·
 178

Э

эквивалентность · 77, 155
 электронный паспорт · 23