

**А.А. Вичугова, В.Н. Вичугов, Е.А. Дмитриева,  
Г.П. Цапко, С.Г. Цапко**

**МЕТОДОЛОГИЧЕСКИЕ ОСНОВЫ ПРОЕКТИРОВАНИЯ  
СЛОЖНЫХ НАУКОЕМКИХ ИЗДЕЛИЙ И  
ПРИНЦИПЫ ПОСТРОЕНИЯ  
ИНТЕГРИРОВАННОЙ ИНФОРМАЦИОННОЙ СРЕДЫ  
НА БАЗЕ CALS-ТЕХНОЛОГИЙ**

ТПУ

Томск, 2013

УДК 658.512.26:004  
ББК 30.2:32.973-018.2  
М 54

**М 54 Вичугова А.А., Вичугов В.Н., Дмитриева Е.А., Цапко Г.П., Цапко С.Г.** Методологические основы проектирования сложных наукоемких изделий и принципы построения интегрированной информационной среды на базе CALS-технологий: – Томск: ТПУ, 2013. – 180 с.

ISBN 978-5-4387-0347-1

В монографии рассмотрены теоретические и практические положения, направленные на повышение эффективности информационного сопровождения процесса проектирования высокотехнологичной продукции с длительным сроком активного существования. В работе раскрываются особенности разработки радиоэлектронной аппаратуры космического назначения, рассматриваются программные средства поддержки бизнес-процессов разработки бортовой радиоэлектронной аппаратуры космического назначения, на базе которых может быть реализована интегрированная информационная среда. Приводятся методики, модели и алгоритмы, необходимые для автоматизации управления жизненным циклом разнотипных взаимозависимых объектов, которые создаются на этапе проектирования изделий.

Книга предназначена для научных работников и студентов старших курсов технических специальностей, а также инженеров и специалистов, занимающихся системным проектированием сложных систем.

**УДК 658.512.26:004**  
**ББК 30.2:32.973-018.2**

**Рецензенты:** *А.М. Корилов*, заслуженный деятель науки РФ, д.т.н., профессор, заведующий кафедрой АСУ, ТУСУР  
*А.А. Глазунов*, д.ф.-м.н., профессор, директор НИИПММ, ТГУ

*Работа выполнена в порядке реализации постановления Правительства РФ №218 от 09.04.2010 договор №13.G25.31.0017, и договора №2148 от 05.07.2010 ТГУ с ОАО «ИСС» им. акад. М.Ф. Решетнева*

**ISBN 978-5-4387-0347-1**

© А.А. Вичугова, В.Н. Вичугов,  
Е.А. Дмитриева, Г.П. Цапко, С.Г. Цапко, 2013

# СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	6
1 Анализ применения информационных технологий поддержки жизненного цикла изделий к задачам проектирования высокотехнологичной продукции с длительным сроком активного существования.....	12
1.1 Анализ процессов проектирования высокотехнологичной продукции с длительным сроком активного существования.....	12
1.1.1 Особенности высокотехнологичной продукции с длительным сроком активного существования .....	12
1.1.2 Процессы проектирования в отрасли космического приборостроения .....	17
1.2 Анализ объектов проектирования.....	22
1.2.1 Обзор объектов проектирования и анализ их взаимосвязей ...	22
1.2.2 Анализ методов управления продукцией на этапе ее проектирования .....	24
1.2.3 Анализ описания ЖЦ объектов проектирования в существующих государственных стандартах.....	26
1.3 Аналитический обзор информационных технологий поддержки жизненного цикла изделий.....	30
1.3.1 Обзор современных методов и средств информационной поддержки жизненного цикла изделий.....	30
1.3.2 Анализ состава интегрированной информационной среды и взаимодействия ее типовых программных компонент .....	33
1.3.3 Анализ типовых функциональных возможностей систем управления данными для задач проектирования .....	35
1.4 Поиск формальных методов управления связанными объектами	40
1.4.1 Аналитический обзор формальных методов.....	40
1.4.2 Анализ и выбор подходов к разработке программных систем	41
1.4.3 Анализ и выбор методов графического описания и документирования программных систем .....	43
1.4.4 Поиск математических методов для формализации структуры взаимосвязей объектов и динамики изменения их состояний .....	45
1.4.5 Распараллеливание и синхронизация задач при согласовании документов для автоматизации изменения состояний ЖЦ объектов проектирования в ИИС.....	48
1.5 Цель и задачи исследования.....	51
1.6 Резюме обзорных исследований .....	55
2 Разработка методов получения моделей жизненного цикла объектов проектирования .....	57

2.1 Метод формализации структуры взаимосвязей объектов проектирования в статическом состоянии.....	57
2.1.1 Разработка математических положений .....	58
2.1.2 Концептуальное проектирование базы данных СУД.....	61
2.2 Метод формализации динамики изменения жизненного цикла объектов проектирования .....	63
2.2.1 Методика формализации жизненного цикла объектов проектирования.....	64
2.2.2 Модель динамики изменения стадий жизненного цикла объектов проектирования в ИИС .....	67
2.2.3 Объектно-ориентированная модель сущностей проектирования.....	78
2.3 Резюме прикладных исследований .....	83
3 Алгоритмическое обеспечение средств автоматизированного управления жизненным циклом Разнотипных взаимозависимых объектов проектирования в ИИС .....	84
3.1 Формальное описание динамики изменений стадий ЖЦ объектов проектирования.....	84
3.2 Алгоритм согласования документов в ИИС .....	90
3.2.1 Формализация основных понятий процесса согласования.....	90
3.2.2 Последовательное согласование.....	94
3.2.3 Параллельное согласование .....	96
3.3 Использование шаблонов проектирования.....	98
3.4 Метод построения интегрированной информационной среды....	104
3.4.1 Разработка требований к интеграции САПР и СУД в рамках ИИС	104
3.4.2 Разработка структуры хранения типовых справочных данных	106
3.4.3 Разработка алгоритма экспорта ИМИ из САПР в СУД .....	109
3.4.4 Разработка алгоритма формирования ЭСИ в СУД.....	111
3.5 Анализ и поиск методов программной интеграции информационных систем	116
3.5.1 Общий обзор существующих методов интеграции приложений	116
3.5.2 Обзор технологий интеграции приложений методом удаленного вызова процедур .....	117
3.5.3 Выбор объектно-компонентной технологии интеграции приложений методом удаленного вызова процедур.....	119
3.5.4 Средства реализации СОМ-технологии.....	120
3.5.5 Интеграция информационных систем средствами СОМ-технологии.....	121
3.6 Резюме по разработке алгоритмических решений.....	124

4 Реализация разработанных моделей и алгоритмов в виде комплекса программных решений .....	126
4.1 Постановка прикладных задач предметной области на примере процесса проектирования бортовой РЭА .....	127
4.1.1 Специфика процессов проектирования бортовой РЭА .....	127
4.1.2 Программные компоненты ИИС проектирования бортовой РЭА	129
4.1.3 Реализация ИИС проектирования бортовой РЭА КА на базе СУД Enovia SmarTeam и ее интеграций с САПР Altium Designer и SolidWorks	133
4.1.4 Формирование ЭСИ в СУД СУД Enovia SmarTeam на основе ИМИ-файлов, разработанных в САПР Altium Designer и SolidWorks	135
4.1.5 Обоснование необходимости разработки дополнительного программного обеспечения для управления ЖЦ объектов проектирования в СУД Enovia SmarTeam .....	137
4.2 Программная реализация функций управления жизненным циклом объектов проектирования в ИИС на базе интеграций СУД Enovia SmarTeam с САПР Altium Designer и SolidWorks .....	139
4.2.1 Средства расширения функциональных возможностей СУД Enovia SmarTeam .....	139
4.2.2 Выбор среды разработки и языка программирования .....	141
4.2.3 Разработка программных решений .....	143
4.3 Структура и взаимосвязь проектных данных в СУД Enovia SmarTeam	147
4.4 Динамика автоматизированного управления жизненным циклом разнотипных объектов проектирования в СУД Enovia SmarTeam .....	150
4.5 Резюме практической реализации .....	157
ЗАКЛЮЧЕНИЕ .....	160
ЛИТЕРАТУРА.....	163
СПИСОК ОБОЗНАЧЕНИЙ И СОКРАЩЕНИЙ.....	178

## ВВЕДЕНИЕ

**Актуальность работы.** Проектирование высокотехнологичной продукции с длительным сроком активного существования (глубоководные и космические аппараты, искусственные человеческие органы и т.д.) представляет собой целый комплекс процессов со сложной структурой взаимосвязей и временной длительностью. Отличительной особенностью данной деятельности является проведение натурных испытаний на этапе проектирования, что вызывает необходимость создания реального образца изделия для каждой стадии испытаний. Готовность изделия к производству определяется состоянием конструкторской документации (КД) на данное изделие. Под качеством КД понимается отсутствие ошибок в описании характеристик изделия по нормативно-техническим стандартам. Первоисточником данных для КД является совокупность элементов электронной структуры изделия (ЭСИ), формируемой на основе файлов информационных моделей (ИМИ), разработанных в различных системах автоматизированного проектирования (САПР). Проектирование изделия считается завершенным в случае окончательно сформированной ЭСИ и утвержденной КД.

Таким образом, проектирование высокотехнологичной продукции сопровождается большим количеством разнотипных взаимозависимых данных в виде ИМИ, КД и ЭСИ. Необходимость работы с большим количеством этой информации усложняет процесс проектирования и увеличивает его длительность за счет временных затрат на поиск необходимых данных и изменение их в ручном режиме. Учитывая требования современной экономической ситуации к сокращению сроков представления готовых изделий на рынок с условием сохранения их качества задача уменьшения длительности проектирования путем оптимизации процессов, не добавляющих ценности итоговому продукту (итеративный выпуск и проверка документации, поиск данных и т.д.), становится особенно актуальной.

Согласно [1], для обозначения создаваемых при проектировании сущностей (Изделие, ИМИ, КД, ЭСИ) в работе используется термин «объекты проектирования», а термин «управление жизненным циклом объектов проектирования» – для обозначения следующей совокупности действий: структурированное хранение большого объема данных и автоматизированное изменение разнотипных объектов проектирования от исходного состояния к целевому с учетом их взаимозависимости, включая мониторинг работ участников процессов проектирования и обработки информации. Сегодня для решения этих задач применяются технологии информационной поддержки жизненного цикла (ЖЦ) изделий, реализованные в системах управления данными (СУД) и САПР. Объе-

динение СУД и САПР в единый программно-аппаратный комплекс представляет собой интегрированную информационную среду (ИИС).

Идея организации ИИС широко исследуется с 2000-х гг. российскими и зарубежными учеными в работах О. Шиловицкого, Д. Брауна, Яблочникова Е.И., Молочникова В.И. и др. Особое внимание уделяется командному сквозному проектированию, которое означает многопользовательскую работу и оперативный обмен электронными данными при использовании нескольких САПР в разработке изделий. Проведенный анализ существующих стандартов, методов, алгоритмов и программных решений для выполнения проектных работ по разработке изделий в рамках ИИС показал, что они не позволяют в полной мере отразить особенности проектирования высокотехнологичной продукции с длительным сроком активного существования. К этим особенностям относятся итеративность процессов проектирования в соответствии со стадиями испытаний и сложность структуры влияния разнотипных объектов в различных стадиях ЖЦ друг на друга. Также следует учитывать необходимость интеграции различных САПР, в которых выполняется разработка частей проектируемого изделия, при условии агрегации всех характеристик одного элемента ЭСИ в СУД и отсутствия дублирования данных.

Существует ряд государственных и отраслевых стандартов, которые регламентируют основные термины и определения жизненного цикла продукции, например, Р 50-605-80-93, ГОСТ Р 15.201-2000, ГОСТ 2.103-68, ГОСТ Р 5379-2010, ГОСТ Р ИСО 9001-2001 (раздел 7), ГОСТ Р ИСО 9004-2001 (раздел 7), ГОСТ Р ИСО/МЭК 15288-2005. По результатам проведенных исследований и анализа источников литературы [2-51], сделан вывод, что существующие нормативные документы не содержат готовых решений по структуре взаимосвязей разнотипных объектов проектирования и динамике изменения их состояний. Дополнительные ограничения накладывает итеративность процессов проектирования в соответствии со стадиями испытаний и сложность структуры взаимозависимостей состояний ЖЦ разнотипных объектов (Изделие, ИМИ, КД, ЭСИ). Также следует учитывать необходимость интеграции различных САПР, в которых выполняется разработка электрических и механических частей проектируемого изделия, при условии агрегации всех характеристик одного элемента ЭСИ в СУД и отсутствия дублирования данных.

Для обеспечения целостности данных об объектах проектирования, структурированного хранения информации и ее автоматизированной обработки, необходимо наличие правил управления разнотипными объектами проектирования с учетом их взаимозависимости. Поэтому разработка информационного, алгоритмического и программного обеспе-

чения для автоматизированного управления жизненным циклом разнотипных объектов проектирования, которые создаются при разработке высокотехнологичной продукции с длительным сроком активного существования, имеет высокий уровень актуальности.

**Объектом исследования** является процесс автоматизированного управления жизненным циклом разнотипных взаимозависимых объектов в интегрированной информационной среде при проектировании высокотехнологичной продукции с длительным сроком активного существования, с учетом их версионности и взаимовлияния.

**Предметами исследования** являются методы проектирования программного обеспечения с целью структуризации и автоматизированной обработки разнотипных взаимосвязанных данных.

**Настоящее исследование связано с созданием** новых и развитием существующих теоретических и практических положений методологии информационной поддержки жизненного цикла изделий, обеспечивающих повышение эффективности информационного сопровождения процессов проектирования высокотехнологичной продукции за счет автоматизированного управления разнотипными взаимозависимыми объектами в интегрированной информационной среде.

В ходе выполнения исследования решены следующие **задачи**:

1. Выявление особенностей процессов проектирования высокотехнологичной продукции с длительным сроком активного существования с учетом современных требований к данной деятельности и ее результатам, а также анализ актуальных проблем автоматизированного управления данными о множестве разнотипных взаимозависимых объектов с помощью существующих методических и программных PLM-решений.

2. Разработка формальных информационных моделей, описывающих статическую структуру связей разнотипных взаимозависимых объектов и являющихся основой для создания информационного, алгоритмического и программного обеспечения поддержки жизненного цикла изделий на этапе их проектирования.

3. Разработка методических основ, позволяющих формализовать информационную модель, показывающую динамику изменения состояний жизненного цикла разнотипных объектов с учетом их версионности и взаимовлияния.

4. Разработка метода построения интегрированной информационной среды на базе системы управления данными и ее интеграций с системами автоматизированного проектирования, позволяющего

реализовать автоматизированное управление жизненным циклом разнотипных взаимозависимых объектов, представленных множеством версий, с учетом специфических особенностей процессов проектирования высокотехнологичной продукции.

5. Программная реализация разработанных моделей и алгоритмов автоматизированного управления жизненным циклом разнотипных объектов в рамках интегрированной информационной среды на базе системы управления данными и ее интеграций с системами автоматизированного проектирования.

6. Апробация разработанных теоретических и практических положений в условиях действующего предприятия, специализирующегося в области проектирования и производства высокотехнологичной продукции с длительным сроком активного существования.

### **Структура и объем работы.**

**Во введении** кратко описана предметная область, объект и методы исследования, дано определение основных терминов, составлена структура и содержание работы. Обоснована актуальность исследования, показана научная новизна, практическая ценность и основные научные результаты.

**Первая глава** посвящена аналитическому обзору применения современных информационных технологий поддержки ЖЦ изделий к проектированию высокотехнологичной продукции с длительным сроком активного существования на примере отрасли космического приборостроения. На основании выполненного аналитического обзора существующих стандартов и методов управления продукцией было выявлено недостаточное покрытие ими процесса проектирования с точки зрения структурирования и автоматизированного изменения разнотипных взаимосвязанных сущностей, представленных множеством версий в различных стадиях ЖЦ.

Обосновано, что исследуемая предметная область характеризуется рядом специфических особенностей, накладывающих ограничение на применение типовых программных решений по управлению жизненным циклом изделий. Сделаны выводы, что изменения стадий ЖЦ объектов проектирования имеют итеративный взаимосвязанный характер, структуру которого необходимо формализовать. Показано, что специфика разработки высокотехнологичных изделий в части сложной структуры процессов согласования документов обуславливает задачу динамического распараллеливания работ исполнителей при изменении стадий ЖЦ разнотипных объектов проектирования.

Выполнен аналитический обзор и поиск математического аппарата, позволяющего описать поведение взаимозависимых сущностей и сформулировать правила управления связанными объектами. Показано, что применение формальных методов (элементы теории множеств, графов и автоматов, объектно-ориентированный подход) позволяет разработать формальные информационные модели изменений ЖЦ объектов проектирования и алгоритмы управления ими. Обоснован выбор инструментария UML для визуализации формальных информационных моделей и алгоритмов управления ЖЦ объектов проектирования. Показана необходимость сочетания функциональных возможностей информационных систем электронного документооборота и управления данными в части применения методов сетевого планирования, теории распараллеливания и синхронизации операций для распределения потоков работ по согласованию документов и соответствующего изменения состояний ЖЦ связанных с ними объектов проектирования другого типа. На основании результатов проведенного исследования аналитического обзора предметной области сформулированы цели и задачи исследования.

**Во второй главе** выполнена разработка математического и алгоритмического обеспечения для автоматизированного управления ЖЦ объектов проектирования с учетом ранее выявленной специфики рассматриваемых процессов. Решена задача концептуального проектирования базы данных системы управления данными, которая является инструментом автоматизированного управления ЖЦ изделий. С использованием аппарата теории множеств формализованы исследуемые объекты проектирования и связанные с ними сущности. Предложена формальная информационная модель в виде UML-диаграммы классов, иллюстрирующая состав и иерархию объектов проектирования, их основные атрибуты и операции в контексте управления их жизненным циклом в рамках ИИС. С целью разработки набора алгоритмов для автоматического изменения стадий ЖЦ объектов проектирования в СУД была составлена методика, показывающая динамику их изменений в формальном виде. Для этого были использованы базовые положения теорий графов и автоматов, которые являются мощным инструментом для описания динамики поведения многокомпонентных систем. Данная методика носит общий характер и может быть использована практически в любой отрасли разработки высокотехнологичной продукции с длительным сроком активного существования. Работоспособность предложенной методики проверена на примере получения формальных информационных моделей, показывающих динамику изменений стадий ЖЦ объектов проектирования в отрасли космического приборостроения.

**В третьей главе** выполнено преобразование графовых моделей в формальные информационные модели в виде UML-диаграмм, позво-

ляющих алгоритмически описать взаимовлияние объектов и логическую последовательность изменения стадий их ЖЦ в ИИС. Описана логическая последовательность действий пользователя ИИС при изменении стадий ЖЦ объектов проектирования. Проанализирован процесс согласования технических документов в терминах методологии потока работ с учетом параллельного и последовательного вариантов исполнения действий. Предложена модель распараллеливания, и синхронизации операций процесса согласования документов, которая положена в основу разработанного алгоритмического и программного обеспечения автоматизированного управления жизненным циклом разнотипных взаимосвязанных объектов проектирования.

**В заключении** приведены основные научные и практические результаты, достигнутые в ходе исследования и решения поставленных задач.

**В приложение** вынесен список обозначений и сокращений.

# **1 АНАЛИЗ ПРИМЕНЕНИЯ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ ПОДДЕРЖКИ ЖИЗНЕННОГО ЦИКЛА ИЗДЕЛИЙ К ЗАДАЧАМ ПРОЕКТИРОВАНИЯ ВЫСОКОТЕХНОЛОГИЧНОЙ ПРОДУКЦИИ С ДЛИТЕЛЬНЫМ СРОКОМ АКТИВНОГО СУЩЕСТВОВАНИЯ**

В данной главе выполнен обзор современных информационных технологий поддержки жизненного цикла изделий и анализ их применения к процессам разработки высокотехнологичной продукции с длительным сроком активного существования. На примере отрасли космического приборостроения выявлены ключевые особенности рассматриваемой предметной области, предложено понятие объектов проектирования. Показана актуальность развития существующих методов управления продукцией для процессов проектирования. Обоснована необходимость разработки моделей и алгоритмов автоматизированного управления объектами, которые создаются на этапе проектирования. Сделаны выводы о необходимости разработки архитектуры комплекса программных решений для поддержки процессов проектирования высокотехнологичной продукции в рамках единой интегрированной информационной среды (ИИС). В результате выполненного обзора и анализа сформулированы цели и задачи работы. Выполнен поиск и обоснование выбора формальных методов для решения поставленных задач.

Основными направлениями работы являются: разработка моделей и алгоритмов автоматизированного управления жизненным циклом объектов проектирования высокотехнологичной продукции с длительным сроком активного существования и практическая реализация полученных теоретических результатов в виде информационного, алгоритмического и программного обеспечения, которое позволит структурировать и автоматизировать процессы проектирования.

## **1.1 Анализ процессов проектирования высокотехнологичной продукции с длительным сроком активного существования**

### **1.1.1 Особенности высокотехнологичной продукции с длительным сроком активного существования**

Высокотехнологичными традиционно считают отрасли, в отгруженной продукции которых доля затрат на отраслевую науку составляет не менее 4,5-5%, а высоконаукоёмкими – те, в которых она составляет свыше 10% [2]. В настоящее время, согласно [2, 3], под высокотехнологичной продукцией в международной и отечественной статистике понимается валовой выпуск следующей группы отраслей промышленности: аэрокосмическая, радиоэлектронная, производство ЭВМ, офисного оборудования и программного обеспечения (ПО), промышленность

средств связи, медицинская техника, прецизионная и оптико-электронная техника и химико-фармацевтическая отрасль [2].

В рекомендациях по стандартизации Р 50-605-80-93 определены основные термины, применяемые в Системе разработки и постановки продукции на производство, включая разновидности промышленной продукции и ее образцов, виды изделий, стадии жизненного цикла продукции, виды проводимых при этом работ и их участники, а также виды документов, разрабатываемых и используемых в процессе проведения этих работ [4].

Согласно [4], совокупность взаимосвязанных процессов последовательного изменения состояния продукции от формирования исходных требований к ней до окончания ее эксплуатации или применения, обозначается термином «Жизненный цикл» (ЖЦ). В зависимости от типа (гражданский или военный) для ЖЦ продукции приняты следующие стадии: исследование, проектирование, производство, реализация и эксплуатация.

Отличительной характеристикой продукции с длительным сроком активного существования является повышенная продолжительность периода эксплуатации без прекращения функционирования в связи с ремонтом и техническим обслуживанием. При этом условия эксплуатации могут быть экстремальными (высокая/низкая температура, давление, радиация и т.д.). Это обуславливает повышенные требования к надежности продукции, что, в свою очередь, накладывает ряд особенностей на процессы ее проектирования и производства.

В случае высокотехнологичной продукции с длительным сроком активного существования стадия проектирования является основным источником временных затрат, поскольку на этом этапе выполняются следующие мероприятия:

- формирование требований к продукции;
- разработка образца продукции с целью проверки предлагаемых решений и уточнения отдельных характеристик [4];
- тестирование образца продукции на соответствие требованиям;
- разработка конструкторской документации (КД), необходимой для производства продукции.

Высокие требования к надежности высокотехнологичной продукции с длительным сроком активного существования обуславливают особый порядок разработки КД, в том числе ее проверку и согласование. В бумажном документообороте согласование реализуется с помощью подписи согласующего лица. Подпись электронного документа выполняется с использованием электронной подписи (ЭП). ЭП представляет собой набор знаков, генерируемый по алгоритму, определенному в [5, 6]. При помощи ЭП можно подписывать отдельные файлы

или фрагменты баз данных. ЭП реализуется внешней по отношению к СУД специализированной программой, сертифицированной ФАП-СИ/ФСБ, например, такой как продукты серии КРИПТОН/Crypton, Sbersign, средства цифровой подписи Microsoft Windows, КАРМА и т.д. После того, как документ подписан ЭП, он не подлежит изменению [7]. Однако, зачастую в процессе проектирования или во время испытаний возникает необходимость изменения КД, в том числе подписанных ЭП. Таким образом, при разработке одного изделия возрастает количество связанных с ним документов.

Как правило, рассматриваемая в рамках настоящей работы высокотехнологичная продукция с длительным сроком активного существования, обладает высокой стоимостью и не является продуктом массового потребления, поэтому объемы производства не велики. Изделия с одинаковыми техническими характеристиками и функциональным назначением принято объединять в серии. Преемственность технических решений как в рамках одной серии, так и между ними, приводит к необходимости структурированного накопления информации о предыдущих проектах и возможности ее использования в новых разработках. Например, на основании анализа работ [8-11], можно сделать вывод, что на российских предприятиях космической отрасли промышленности разработка каждой серии КА осуществляется по проектной модели деятельности, основные положения которой изложены в [13-15].

Также, в связи со специальным назначением рассматриваемой высокотехнологичной продукции с длительным сроком активного существования, к ней выдвигаются особые требования: к изделию в целом и отдельным его комплектующим. Поэтому процесс проектирования изделий, помимо разработки КД, также включает проведение натурных испытаний. Это связано с необходимостью оценить способность продукции безотказно функционировать в жестких условиях реальной эксплуатации. В ГОСТ Р 15.201-2000 описаны цели проведения предварительных и приемочных испытаний опытных образцов продукции [11]. Согласно [16-18], этапу предварительных испытаний (ПРИ) предшествуют конструкторские, к которым относят испытания лабораторных образцов (ЛОИ) и конструкторско-доводочные испытания (КДИ).

Целью конструкторских испытаний является проверка правильности предложенных конструкторско-технологических решений. По результатам конструкторских испытаний, как правило, возникает необходимость в доработке конструкции, изменении технологии, корректировке КД, доукомплектовании и доработке испытательных стендов, после чего испытания повторяются. Целью предварительных испытаний является демонстрация заказчику достижения технических характеристик,

предъявленных в техническом задании (ТЗ), в условиях, близких к натурным, а также подтверждение выполнения задач, поставленных и нерешенных на этапе конструкторской отработки [16-18].

Этапы испытаний принято называть стадиями, например, стадия ЛОИ, КДИ, ПРИ. Натурные испытания каждой стадии проводятся на реальном экземпляре изделия. Поэтому в рамках каждой стадии испытаний должен быть создан экземпляр изделия, а следовательно, должен существовать комплект КД, необходимый для производства изделия. В основе документов на следующую стадию испытаний лежат данные предыдущих стадий. Таким образом, проектирование является стадийным процессом, при этом в рамках разработки одного изделия создается множество связанных с ним документов, представленных в бумажном и электронном видах. В настоящее время для структурированного хранения информации об изделии в виде электронных документов различного формата применяются информационные системы управления данными (СУД).

Также следует отметить еще одну особенность процессов разработки высокотехнологичной продукции с длительным сроком активного существования: использование дорогостоящего оборудования и специального программного обеспечения в качестве инструментов создания, например, системы автоматизированного проектирования (САПР) и инженерного анализа. Это обусловлено технологичностью конструкции, которая означает взаимосвязанное решение конструкторских и технологических задач на всех стадиях проектирования, направленных на повышение производительности труда, достижение оптимальных трудовых и материальных затрат, сокращение времени на производство и эксплуатацию изделия.

Таким образом, проектирование высокотехнологичной продукции с длительным сроком активного существования представляет собой целый комплекс взаимосвязанных процессов, например, функциональное проектирование, конструкторское, технологическое и проведение испытаний. При этом в выполнении каждого этапа работ принимают участие множество специалистов различного профиля: конструкторы, технологи, разработчики программного обеспечения (ПО) и другие сотрудники, которые, как правило, находятся в территориально распределенных зонах. При этом взаимодействие участников процессов проектирования и производства поддерживается с помощью современных информационных технологий передачи и представления данных.

Для структурирования свойств исследуемых объектов, можно выделить следующие категории характеристик:

1. характеристики ЖЦ, к которым относятся:
  - высокая длительность ЖЦ;

- многокомпонентный состав процессов проектирования;
  - большое количество участников процессов проектирования;
  - стадийность проектирования согласно набору испытаний изделия;
  - итеративность процессов проектирования;
2. организационные и продуктовые характеристики, к которым относятся:
- проектная модель деятельности по разработке изделий;
  - малые объемы производства изделий;
3. технологичность конструкции:
- функционально-узловой принцип конструирования – объединение функционально-законченных схем в сборочные единицы и их модульная компоновка;
  - большой объем операций ручной сборки комплектующих в единый модуль;
  - ограниченный перечень комплектующих, разрешенных к применению;
  - вариативность установки комплектующих;
  - специальные требования (к надежности, тепловому режиму, радиационной стойкости, электромагнитным и массогабаритным характеристикам и т.д.);
4. характеристики информационного обеспечения, к которым относятся:
- большой объем инженерной информации, создаваемой при проектировании изделия в рамках каждой стадии испытаний;
  - необходимость внесения изменений в КД на каждой стадии испытаний;
  - высокий уровень применения ИТ (САПР, СУД).

Для конкретизации объектов исследования настоящей работы из множества высокотехнологичных отраслей промышленности, в рамках которых выполняется проектирование и производство продукции с длительным циклом активного существования, целесообразно выбрать предметную область, обладающую вышеуказанными характеристиками и рассматриваемой спецификой, например, космическое приборостроение. Обоснование этого выбора приведено далее.

### **1.1.2 Процессы проектирования в отрасли космического приборостроения**

Согласно Закону РФ «О космической деятельности», космическая деятельность включает создание (в том числе разработку, изготовление, испытания), а также использование и передачу космической техники, космической технологии, иной продукции и услуг, необходимых для ее осуществления. Особенностью космической деятельности является использование уникального оборудования, дорогостоящих образцов космической техники, сложных наземных и орбитальных систем и технологий [19].

Продукция космической отрасли промышленности, к которой, в том числе, относятся космические аппараты (КА), отличается длительным временем эксплуатации (десятки лет) в экстремальных условиях (температура, давление, радиация и т.д.). Например, для геостационарных космических аппаратов связи общепринятый гарантированный срок эксплуатации составляет 15 лет. Это обуславливает высокие требования к надежности данных изделий, что вызывает необходимость проведения большого объема наземных и летных испытаний для получения сертификата на новый КА. Выполнение таких работ требует длительных сроков: от 5-7 до 20 лет для пилотажно-навигационного комплекса и агрегатов [20].

Высокая стоимость космической техники является одной из основных причин того, что на сегодняшний день задача обеспечения максимально длительного срока эксплуатации космических аппаратов на орбите является наиболее актуальной. При этом функциональная насыщенность КА возрастает, например, один современный КА связи по своим основным функциям (количеству предоставляемых потребителям каналов связи) во много раз превосходит своих предшественников десятилетней давности.

Назначение космического комплекса, в состав которого входит КА [19], определяет типовую серию КА. Состав каждой серии ограничен несколькими моделями КА, которые отличаются друг от друга некоторыми свойствами. КА не является продуктом массового потребления, объемы производства не велики. Как правило, разработка каждой серии КА осуществляется по проектной модели деятельности. Увеличение срока эксплуатации и функциональной насыщенности делают каждый космический аппарат уникальным, но при его проектировании и изготовлении широко применяются ранее разработанные и проверенные многочисленными испытаниями на Земле и в космосе технические решения и технологии.

В связи с этим перед предприятиями-разработчиками космической техники стоят следующие задачи: в современных условиях жесткой

конкуренции создать уникальное изделие в кратчайшие сроки, максимально используя отработанные технические и технологические изделия. Таким образом, задача повышения эффективности работы, сокращения сроков и материальных затрат при создании космической техники является актуальнейшей для российских космических предприятий.

Практически все современные КА строятся на основе двух составных частей: модуля служебных систем (МСС) и модуля полезной нагрузки (МПН). В МСС, который также называют «космическая платформа», входят все двигатели и горючее для них, система энергоснабжения, система управления движением, ориентации и стабилизации, система терморегулирования, бортовой компьютер и другие вспомогательные системы. МПН состоит из отсека для установки оборудования выполняющего функции, для которых данный КА был создан. Обычно платформы оптимизируются под массу выводимой полезной нагрузки, что, в свою очередь, определяет массу всего КА и мощность системы энергоснабжения. Таким образом, в связи с особенностями условий функционирования КА, требования к его массогабаритным характеристикам являются решающим ограничением при проектировании.

Функционирование и выполнение целевых задач КА обеспечивается за счет средств вычислительной техники, часть которых расположена непосредственно на борту КА и называется бортовой аппаратурой [19]. Структурированную совокупность радиоэлектронных средств (РЭС), объединенных в сборочные единицы и устройства, предназначенные для преобразования и обработки электромагнитных сигналов в диапазоне частот колебаний от инфранизких до сверхвысоких, необходимых для выполнения различных функций КА на борту принято называть бортовой радиоэлектронной аппаратурой (РЭА).

Рассматривая бортовую РЭА как составляющую КА, следует определить ее место в данном изделии. В [21] предусмотрено иерархическое деление РЭС по уровням, в зависимости от функциональной и конструктивной сложности. В настоящее время для обозначения прибора бортовой РЭА принято использовать термин «радиоэлектронное устройство» (РЭУ), которым обозначают совокупность функционально и конструктивно законченных сборочных единиц, предназначенную для решения технических задач [21].

Иерархическое разделение бортовой РЭА КА по уровням РЭС согласно [21] соответствует понятиям видов изделий по [22]: РЭУ считается изделием, радиоэлектронный функциональный узел (РЭФУ), наряду с печатной платой являются сборочными единицами, а рамка и текстолитовая подложка печатной платы представляют собой детали.

Самый нижний уровень иерархии РЭС принято называть термином «прочие изделия» или «электрорадиоизделия» (ЭРИ). На рисунке 1.1 приведено упрощенное представление компонентов КА с точки зрения проектирования бортовой РЭА согласно [22].



Рис. 1.1. Основные компоненты КА с точки зрения проектирования РЭА

Итак, конечным результатом производственного процесса с точки зрения приборостроения, является создание РЭУ путем такого объединения РЭС, которое позволит обеспечить требуемую функциональность и конструктивное решение [23]. Таким образом, проектирование бортовой РЭА включает, как минимум, три взаимосвязанных аспекта:

- электрическое проектирование, которое обеспечивает выполнение заданных функций РЭУ;
- механическое проектирование, цель которого состоит в размещении РЭС, выбранных для реализации функций РЭУ, в соответствии с требованиями к массогабаритным характеристикам изделия;
- технологическое проектирование, в рамках которого разрабатываются документы, определяющие технологический процесс производства изделия.

Также имеет место разработка ПО, которое определяет поведение изделия в зависимости от различных условий и выполняется с использованием программируемых логических интегральных схем (ПЛИС). Определение вида, состава и количества ПЛИС осуществляется на этапе электрического и механического проектирования, которое принято называть конструкторским. Каждый этап проектирования представляет собой целый комплекс работ и выполняется соответствующим специалистом с использованием специальных инструментов, аппаратного и программного обеспечения, в частности, САПР.

Таким образом, под проектированием понимается разработка такой конструкции бортовой РЭА, которая может быть реализована технологически и представляет собой совокупность ЭРИ и деталей с различными физическими свойствами и формами, находящихся в определенной пространственной, механической и электромагнитной взаимосвязи.

Анализ работ [24-26] позволяет отметить еще одну важную особенность относительно выбора ЭРИ при проектировании прибора КА. В бортовой РЭА космического назначения допускается применять только специальные компоненты космического назначения (категории «SpAce», по терминологии американских и европейских стандартов), которые будут гарантированно выполнять свои функции не менее 15 лет в условиях космического вакуума и космической радиации. В связи с тем, что ракетно-космическая техника относится к сфере оборонной промышленности, ЭРИ должны входить в перечень средств, разрешенных к применению и соответствовать требованиям необходимых спецификаций и специальных стандартов, например, таких, как МЛ, ESA/SCC и т.д. Основанием для применения импортных ЭРИ является невозможность реализации требуемых тактико-технических характеристик аппаратуры на отечественной элементной базе. При этом выбор импортных ЭРИ должен производиться из числа включенных в «Перечень изделий и материалов иностранного производства, разрешенных к применению» для аппаратуры конкретного изделия, выпускаемый Министерством Обороны РФ. Кроме того, при выборе ЭРИ на этапе проектирования следует также учитывать возможные сроки поставки комплектующих, поскольку элементная база космического назначения производится только для конкретного заказа. В связи с этим сроки поставки комплектующих могут составлять несколько месяцев, что отражается на длительности процессов проектирования и производства.

Анализируя существующую классификацию машиностроения по отраслям, приведенную в [27], можно сделать вывод, что космическое приборостроение сочетает в себе типовые характеристики разных направлений машиностроительного комплекса. Проектирование и производство бортовой РЭА относится к приборостроительной отрасли точного машиностроения, тогда как ракетно-космическую промышленность принято считать подотраслью тяжелого машиностроения [27]. Отличительной чертой космического приборостроения является включение испытаний в процесс проектирования. В связи с повышенными требованиями к надежности изделий, проектирование КА включает испытания, которые называются наземная экспериментальная отработка (НЭО). В настоящее время продолжительность НЭО бортовой РЭА может превышать продолжительность проектирования в 2-3 раза, т.к. вы-

полняется последовательная разработка нескольких образцов изделия, каждый из которых подвергается различным видам испытаний.

НЭО условно делят на три уровня: бортового оборудования, систем и КА. Каждый из этих уровней, в свою очередь, разделяется на конструкторские и предварительные испытания, включая стадии ЛОИ, КДИ, ПРИ, описанные ранее. Кроме того, на уровне КА выполняются конструкторские испытания силовых узлов и зачетные испытания (статические, динамические) [16]. Замена испытаний материальных опытных образцов виртуальными испытаниями позволит в разы сократить время и затраты на НЭО путем исключения некоторых этапов (например, ЛОИ или КДИ). Но это потребует разработки большего количества компьютерных моделей проектируемого изделия. Также на каждой стадии испытаний выполняется разработка большого количества технических документов и другой информации об изделии, которую необходимо структурировать, обеспечить ее хранение и возможность оперативного поиска с целью использования в новых проектах.

Кроме того, высокие требования к надежности КА обуславливают особый порядок разработки КД, в том числе ее проверку, согласование и внесение изменений. В связи с большим объемом информации в бумажном и электронном видах необходимо применение соответствующих методов и средств информационных технологий для структурированного хранения данных об изделии с целью сокращения времени проектирования. В контексте длительного и дорогостоящего периода разработки рассматриваемой продукции эта задача является весьма актуальной. Кроме того, накопление информационного капитала в виде данных по предыдущим проектам и использование их наиболее удачных решений в новых разработках направлено на сокращение количества ошибок в КД. В связи особыми требованиями к надежности КА, высокой стоимостью его проектирования и эксплуатации наличие ошибок в КД является недопустимым.

В настоящее время срок проектирования бортовой РЭА составляет от 6 до 12 месяцев, при том, что контрактные сроки создания всего КА (включая процессы проектирования, изготовления и испытаний) при коммерческих заказах составляют 24-30 месяцев. В связи с этим задача сокращения сроков проектирования и повышения качества результатов этого процесса становится особенно важной.

Таким образом, управление процессом проектирования высокотехнологичной продукции с длительным сроком активного существования, включая автоматизированное формирование создаваемых при этом взаимосвязанных разнотипных объектов, их структурированное хранение и автоматическое изменение, позволит сократить временные и материальные затраты на осуществление данной деятельности.

Подводя итог вышеприведенному описанию особенностей процессов проектирования бортовой РЭА КА, можно сделать вывод, что данная продукция является высокотехнологичной и обладает длительным сроком активного существования. Поэтому теоретические и практические положения настоящей работы, показанные на примере бортовой РЭА КА, применимы также и к другим отраслям промышленности, продукция которых соответствует критериям, определенным в п.1.1, например, глубоководные аппараты, искусственные человеческие органы и т.д.

Рассматривая процессы проектирования высокотехнологичной продукции с длительным сроком активного существования, следует выполнить подробный анализ сущностей, которые создаются и используются в рамках этой деятельности, а также проанализировать существующие методы управления рассматриваемыми объектами.

## **1.2 Анализ объектов проектирования**

### **1.2.1 Обзор объектов проектирования и анализ их взаимосвязей**

В приборостроительной промышленности проектирование относится к ключевым видам деятельности, конечным результатом которой является продукция, ценная для потребителя. Единица промышленной продукции называется изделием. Изделия делятся по видам (сборочная единица, деталь, комплекс и комплект), и могут включать составные части [22].

Главным результатом проектирования является КД, необходимая для производства новой продукции. Виды и состав КД регламентируется Государственными стандартами Единой системы конструкторской документации (ЕСКД). В связи с широким распространением вычислительной техники и информационных технологий (ИТ), в ряд существующих стандартов, которые описывают состав КД и порядок работы, внесены соответствующие изменения. В частности, согласно [30, 31] состав комплекта КД дополняется электронной моделью изделия (ЭМИ), которая используется для получения чертежей и должна содержать полный набор конструкторских, технологических и физических параметров согласно [32], необходимых для выполнения расчетов, математического моделирования, разработки технологических процессов и др. [31].

Однако, проектирование включает в себя не только разработку массогабаритных свойств изделия посредством его электронной модели. Также должны быть определены другие характеристики, необходимые для производства и эксплуатации изделия, например, электромагнитные параметры и т.д. Для моделирования, расчетов и анализов применяются системы автоматизированного проектирования (САПР).

Разрабатываемый в САПР виртуальный образец продукции называется информационной моделью изделия (ИМИ). Согласно [33], ИМИ является источником данных для формирования КД, необходимой для производства продукции. Фактически, ИМИ представляет собой файл формата САПР для описания каких-либо свойств изделия. Например, твердотельная ИМИ, созданная в САПР *SolidWorks*, используется на этапе механического проектирования изделия; ИМИ для описания электрических свойств изделия представляет собой набор файлов проекта, разработанного в САПР *Altium Designer* и т.д.

Кроме того, ИМИ отражает состав изделия с точки зрения иерархии его составных частей, которую принято называть электронной структурой. Таким образом, проектирование включает в себя описание в различных САПР электронной структуры, электронной и информационной моделей изделия. В связи с важностью этих понятий, следует рассмотреть их определение с точки зрения регламентирующих стандартов ЕСКД.

Электронная структура изделия (ЭСИ) – это конструкторский документ, содержащий состав сборочной единицы, комплекса или комплекта и иерархические отношения (связи) между его составными частями и другие данные в зависимости от его назначения [33]. ИМИ – это совокупность данных и отношений между ними, описывающая различные свойства реального изделия, интересующие разработчика модели и потенциального или реального пользователя [33]. Таким образом, электронная модель изделия является частным случаем его информационной модели, которая отражает данные о проектируемой сущности с той или иной точки зрения (например, ИМИ электрического или конструкторского проектирования, теплового моделирования и т.д.). В свою очередь, электронная структура изделия определяется его совокупностью информационных моделей.

Таким образом, ИМИ является первичным документом для получения ЭСИ и различных конструкторских документов (КД), входящих в основной и полный комплекты согласно [30]. Например, спецификация, один из самых важных конструкторских документов, необходимых для производства изделия, формируется на основе данных из ЭСИ и информационных моделей, разработанных в различных САПР. Изменение ИМИ влечет изменение ЭСИ и, соответственно, КД. Поскольку функции интерактивного проектирования и инженерных расчетов реализуются в ИМИ, основным назначением КД становится статическое представление информации об изделии в соответствии с [30].

Обобщая понятия создаваемых в процессе проектирования сущностей (изделие, ИМИ, ЭСИ, КД), следует определить единый термин для их обозначения. В соответствии с [1], целесообразно использовать по-

нятие объекта проектирования. Таким образом, далее в работе под термином «объекты проектирования» подразумевается проектируемое изделие, его информационные модели и конструкторские документы, необходимые для его производства. С учетом описанных взаимозависимостей разнотипных объектов проектирования друг от друга, можно выделить два вида задач:

- структурированное хранение связанных сущностей;
- динамическое создание и изменение характеристик в соответствии с заданными условиями, что, фактически, является функцией управления.

Для структурированного хранения вышеуказанных сущностей и взаимосвязанного управления ими необходимо проанализировать существующие методы управления продукцией на этапе ее проектирования и оценить их достаточность для решения поставленных задач.

### **1.2.2 Анализ методов управления продукцией на этапе ее проектирования**

Существует ряд государственных и отраслевых стандартов, которые регламентируют основные термины и определения жизненного цикла продукции, например, Р 50-605-80-93, ГОСТ Р 15.201-2000, ГОСТ 2.103-68, ГОСТ Р 5379-2010, ГОСТ Р ИСО 9001-2001 (раздел 7), ГОСТ Р ИСО 9004-2001 (раздел 7), ГОСТ Р ИСО/МЭК 15288—2005 [3, 11, 34-37]. Однако, указанные стандарты описывают общие положения процессов ЖЦ и требования к характеру их выходных результатов, но не содержат готовых решений по структуре взаимосвязей разнотипных объектов проектирования и динамике изменения их состояний на протяжении всего процесса проектирования и испытаний.

Поэтому для обеспечения целостности данных об объектах проектирования, структурированного хранения информации и ее автоматизированной обработки, необходимо непротиворечивое представление информационных сущностей, которые создаются и используются при проектировании. Таким образом, возникает задача управления объектами проектирования с учетом их взаимозависимости. Большинство современных методов управления продукцией, такие как, например, управление затратами, сбытом, качеством, MRP-II, теория ограничений, бережливое производство и другие методы производственного менеджмента, основные положения которых изложены в работах Масаки Имаи, Тайити Оно, Дж. Орлиски, Элияху М. Голдрата, фокусируются на экономических показателях [38].

В настоящее время широко применяется методология параллельного проектирования, разделенного по стадиям с использованием и накоплением знаний, называемая RGD (от англ. RelAtionAl GenerAtive

Design) [39]. В основе методологии RGD лежит критический подход к проектированию, базовые положения которого изложены в работах английских дизайнеров А. Дюнне и Ф. Раби [40, 41]. В настоящее время RGD является основой PLM-решений, поскольку предусматривает структурированное накопление опыта предыдущих разработок и использование его для текущих проектов [42]. Анализируя существующие методы проектирования, основные положения которых изложены в работах Альтшуллера Г.С., Половинкина А.И., Диксона Дж. и других авторов, [43], можно сделать вывод, что RGD относится к итерационным эвристическим методам.

Основные принципы методологии RGD заключаются в следующем:

- процесс проектирования разделяется на стадии;
- каждой стадии соответствует специализация пользователей по ролям, правам доступа, представлениям данных (т.е. по видам моделей);
- при построении моделей на следующей стадии проектирования наследуются только те данные, которые необходимы для работы на этой стадии;
- ограничение по ролям обеспечивает для каждого пользователя ролевой группы видимость только предыдущих стадий, которые специально определены как необходимые на текущей стадии;
- связь с информацией, созданной на предыдущих стадиях проектирования, реализуется с помощью ассоциативных ссылок.

Таким образом, обеспечивается возможность отслеживания любых изменений, выполненных на предыдущих стадиях, конфиденциальность информации и возможность работы с максимально облегченным представлением моделей на каждой стадии. При этом, гарантируется целостность проекта в целом, т.к. все причинно-следственные связи отслеживаются по ссылкам [39].

Анализ основных положений методологии RGD позволяет сделать вывод о возможности ее применения для задач управления жизненным циклом объектов проектирования, включая учет стадий испытаний. Гибкая политика управления доступом к информации, принятая в RGD, объясняет тот факт, что эта методологии лежит в основе практически всех современных САПР и СУД. Однако, несмотря на то, что RGD предлагает концептуальные понятия управления большими массивами связанных данных с помощью ИС, она не содержит готовых структурированных моделей и алгоритмов управления продукцией и связанными с ней информационными сущностями, которые могут быть использованы в проектировании высокотехнологичных изделий с дли-

тельным сроком активного существования без их адаптации к специфике данной области деятельности. Для того, чтобы сделать обоснованные выводы о необходимости разработки таких моделей и алгоритмов, следует детально рассмотреть существующие государственные стандарты, которые описывают ЖЦ объектов проектирования.

### **1.2.3 Анализ описания ЖЦ объектов проектирования в существующих государственных стандартах**

На основании анализа работ [18, 23-26], можно представить типовой процесс разработки высокотехнологичной продукции с длительным сроком активного существования в виде следующей последовательности действий обработки объектов проектирования:

1. Разработка ИМИ в САПР.
2. Разработка КД с использованием САПР, а также текстовых, табличных и графических редакторов.
3. Получение твердой копии КД, т.е. представление документов в бумажном виде;
4. Проверка КД при ее согласовании.
5. При наличии ошибок в КД выполняется поиск источника ошибок (неверные данные возникли при моделировании в САПР или при оформлении КД). В зависимости от источника ошибок возврат к действию 1 или 2.
6. В случае отсутствия ошибок КД согласовывается и утверждается.

Графическая иллюстрация вышеприведенной последовательности действий представлена в виде алгоритма на рисунке 1.2.

Одним из главных недостатков описанной последовательности действий (рисунок 1.2) является наличие большого количества бумажных документов, на обработку которых (включая процессы создания, проверки и согласования) уходит значительное количество временных, материальных и трудовых ресурсов. При этом создаются дополнительные документы, описывающие внесенные в КД изменения. ГОСТ 2.503-90 регламентирует правила внесения изменений в КД с помощью выпуска извещений об изменениях. Таким образом, количество бумажных документов при каждой итерации их разработки и согласования все более возрастает.

Для устранения данной ситуации с целью оптимизации процесса проектирования был разработан ряд государственных стандартов, которые регламентируют понятие «электронный документ» и применение этого понятия в процессе проектирования. Например, ГОСТ 2.051-2006 устанавливает общие требования к выполнению электронных конструкторских документов изделий машиностроения и приборостроения, включая описание основных терминов [44].

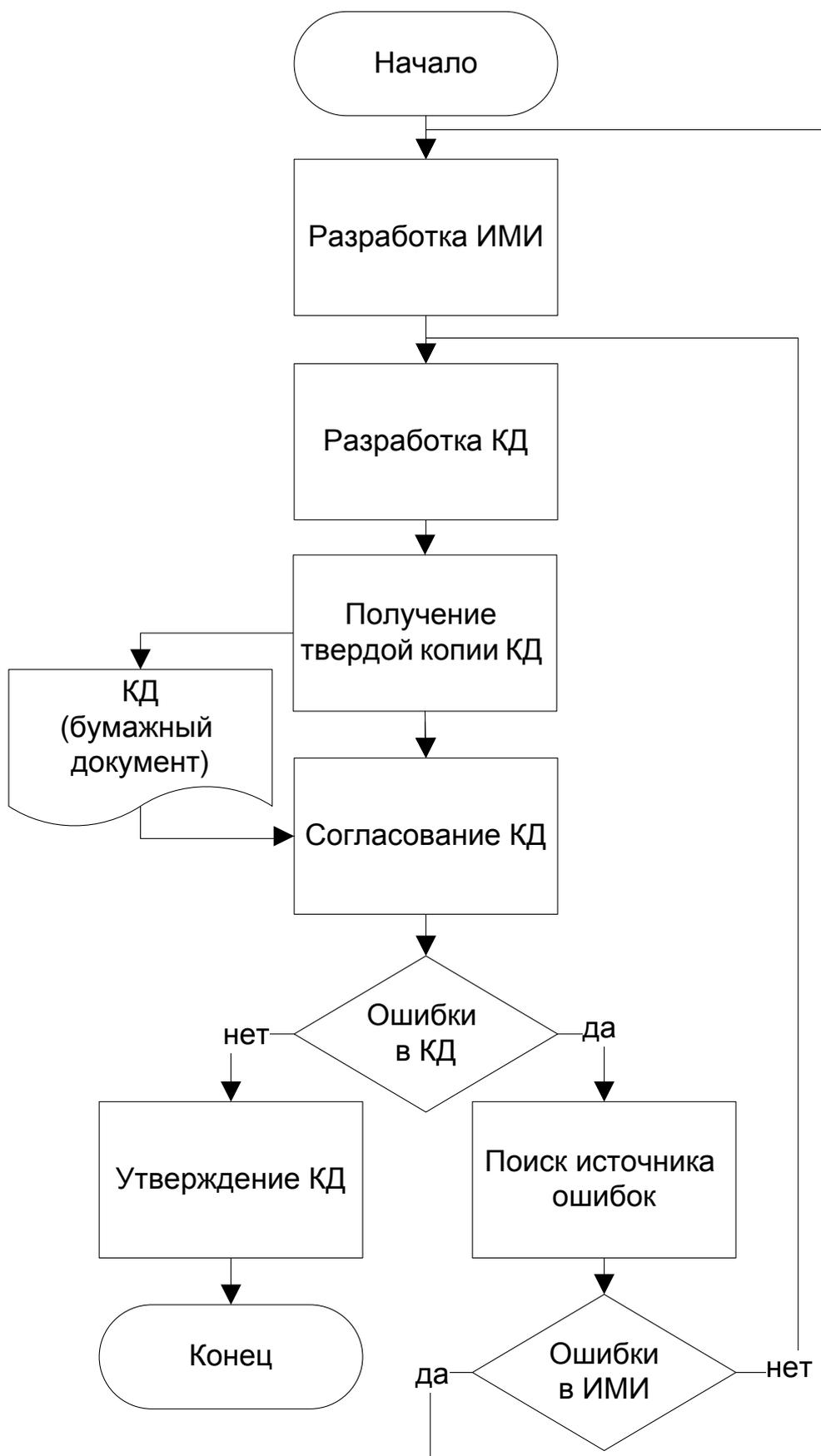


Рис. 1.2. Последовательность действий типowego процесса проектирования

Рассматривая ГОСТ 2.051-2006 с точки зрения управления ЖЦ объектов проектирования, следует отметить, что в нем приводится зна-

чение терминов «версия документа» и «статус версии документа», а также сказано, что изменение содержательной части электронного документа вызывает появление его новой версии, которая замещает изменяемую [44]. Однако, данный стандарт не содержит готовых алгоритмических решений по управлению жизненным циклом объекта проектирования типа «документ».

Аналогичным образом стандарты ГОСТ 2.052-2006 и ГОСТ 2.053-2006 описывают общие требования к выполнению электронной структуры и электронной модели изделий машиностроения и приборостроения. Согласно [44], ЭСИ представляет собой электронный конструкторский документ в виде обменного файла согласно ИСО 10303-21-2002 или базы данных с организацией доступа согласно ИСО 10303-22, формируемый автоматизированным способом.

На основе ЭСИ могут быть сформированы в электронной и бумажной форме текстовые конструкторские документы, виды которых указаны в [30]. С точки зрения управления ЖЦ объектов проектирования, ГОСТ 2.053-2006 содержит сведения о возможности существования нескольких версий ЭСИ и указания об обозначении литер и разновидности [33]. Однако в данном стандарте, как и других регламентирующих материалах по описанию ЖЦ продукции и стадиям разработки документов [4, 11, 34-37], отсутствуют алгоритмы изменения состояний ЭСИ, которые могут быть использованы в качестве непосредственной основы программного обеспечения, автоматизирующего этот процесс.

Таким образом, основные стандарты, регламентирующие общие требования к выполнению электронных конструкторских документов изделий машиностроения и приборостроения описывают только принципиальную зависимость ИМИ, ЭСИ и КД друг от друга, но не содержат сведений о порядке согласования этих объектов проектирования. Алгоритмическая последовательность действий по созданию ИМИ, ЭСИ и КД, составленная по результатам проведенного анализа содержания ГОСТ 2.051-2006, ГОСТ 2.051-2006 и ГОСТ 2.051-2006, представлена на рисунке 1.3.

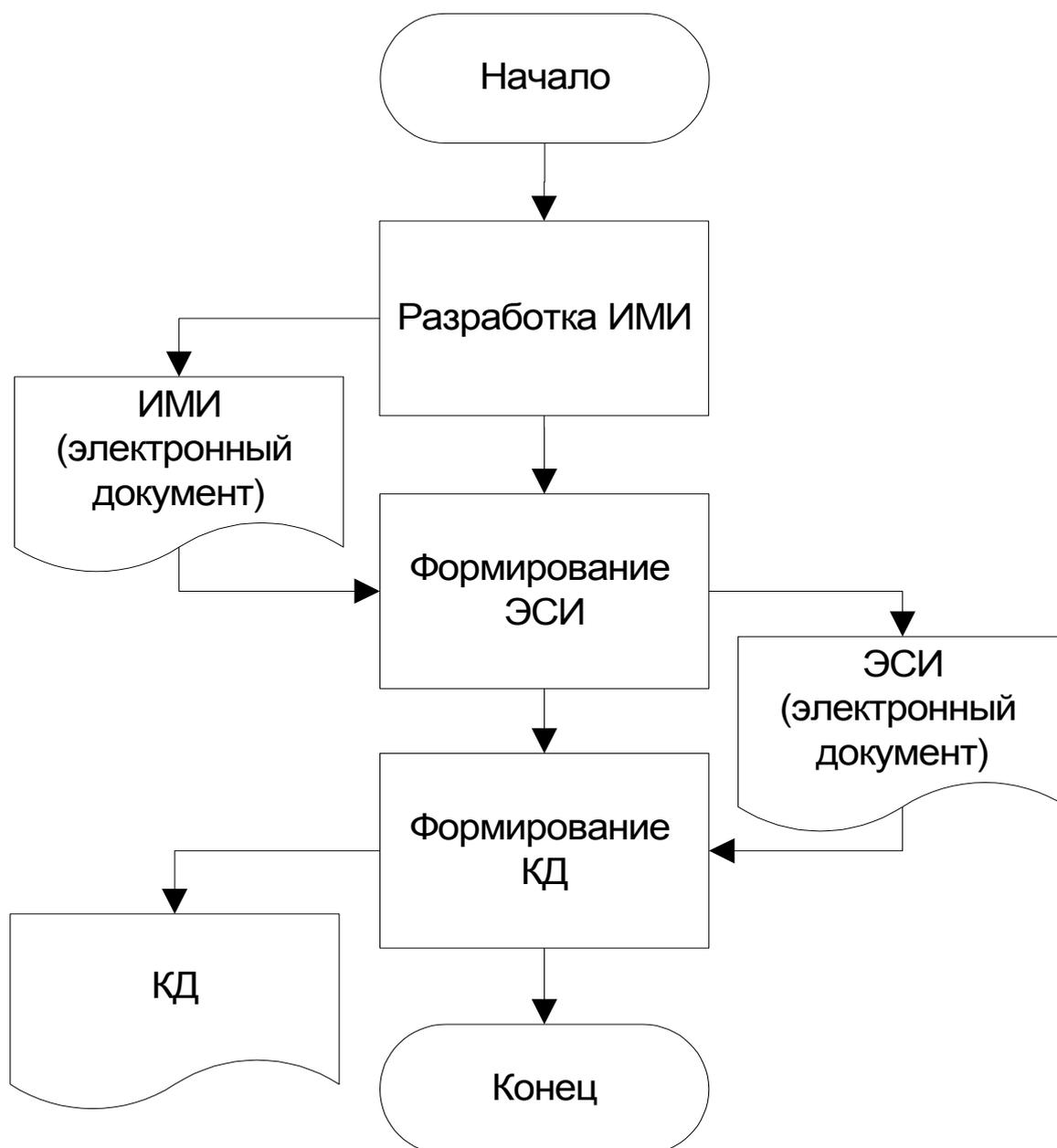


Рис.1.3. Получение ИМИ, ЭСИ и КД (согласно ГОСТ 2.051-2006, ГОСТ 2.051-2006 и ГОСТ 2.051-2006)

Итак, по результатам проведенного анализа существующих государственных стандартов, регламентирующих процессы разработки изделий машиностроения и приборостроения, выявлено отсутствие готовых моделей и алгоритмов управления жизненным циклом объектов проектирования, которые могут являться непосредственной основой соответствующего программного обеспечения. Для подтверждения этого вывода следует проанализировать назначение существующих информационных систем поддержки ЖЦ изделий, а также оценить возможность их интеграции в ИИС с учетом выявленной специфики процессов проектирования высокотехнологичной продукции с длительным сроком активного существования.

### **1.3 Аналитический обзор информационных технологий поддержки жизненного цикла изделий**

#### **1.3.1 Обзор современных методов и средств информационной поддержки жизненного цикла изделий**

Исследование методов управления ЖЦ изделий с помощью современных информационных технологий приводится в работах Пестрецова С.И., Бакаева В.В., Яблочникова Е.И., Молочникова В.И., Д. Брауна, А. Мейера и других авторов [7, 45-51]. История систем и технологий информационной поддержки ЖЦ изделий начинается с разработки САПР в 60-х гг. XX в. трудами таких исследователей и практиков как, например, П. Хэнретти, И. Сазерленда, С. Гейзберга, Г.К. Горанского, В.Д. Цветкова, Н.М. Капустина, С.П. Митрофанова и др. [52].

По мере усложнения изделий и информационных технологий их проектирования и производства, возникает необходимость накопления и структурированного хранения информации о продукции на каждом этапе ее ЖЦ. Таким образом, в 80-х гг. XX века начинает формироваться методология CALS (от англ. *Continuous Acquisition And Life cycle Support* - непрерывная информационная поддержка поставок и жизненного цикла). В настоящее время термин CALS замещается аббревиатурой PLM (от англ. *Product Lifecycle MAnAgement* – управление жизненным циклом изделия) [48].

Анализируя работы Д. Брауна, А. Мейера, Яблочникова Е.И., Молочникова В.И., Пестрецова С.И., Бакаева В.В., Д. Левина и других авторов по тематике PLM [7, 45-5451], можно выделить следующие базовые положения PLM-методологии:

- деятельность по работе с изделием на каждом этапе его ЖЦ осуществляется в соответствии с процессным подходом к управлению предприятием;
- обмен информацией между участниками процессов ЖЦ выполняется в электронном виде;
- в силу географической распределенности участников процессов ЖЦ изделий, передача данных осуществляется с использованием компьютерных сетей;
- все ИС, базы данных (БД) и прочие программные приложения должны обеспечивать однозначность и непротиворечивость хранения и представления информации, в том числе, отсутствие дублирования данных;
- информация хранится централизованно на защищенных сетевых ресурсах, а не на локальных компьютерах участников процессов ЖЦ изделий;

- доступ к информации и разрешение на ее обработку предоставляется участникам процессов ЖЦ изделий в зависимости от их функций в данной деятельности.

Все вышеперечисленные положения укладываются в понятие единого информационного пространства (ЕИП). ЕИП представляет собой комплекс организационно-управленческих, методических и технических решений для всестороннего информационного сопровождения продукции на каждом этапе ее ЖЦ, от разработки концепции до утилизации, через стадии маркетинговых исследований, проектирования, производства/изготовления, реализации и эксплуатационной поддержки [55, 56].

Технологии непрерывной информационной поддержки ЖЦ изделий направлены на оптимизацию сквозного проектирования и производства с точки зрения снижения временных и трудовых ресурсов на данную деятельность. Это достигается за счет автоматизации передачи и структурированного хранения данных об изделии в унифицированных форматах. При этом существенно облегчается решение проблем ремонтпригодности, адаптации к меняющимся условиям эксплуатации, специализации проектных организаций и т. д. в связи с возможностью многократного воспроизведения частей проекта в новых разработках [57].

Анализ исследований О.Шиловицкого [58] и В.Н. Малюха [59, 60] позволяет сделать вывод, что PLM – не единственная методология непрерывной информационной поддержки ЖЦ продукции. В работах В.Н. Малюха приводится аналитический обзор концепции BIM (от англ. *Building Information Modeling*, формирование и управления данными в архитектуре и строительстве в течение всего жизненного цикла зданий и сооружений). Однако, несмотря на некоторые свойства, ключевым отличием данных методологий является понятие объект проектирования: геометрические объекты в BIM, как правило, имеют целевое назначение и классификацию, типичные для архитектуры и строительства (стены, перекрытия, колонны, дверные и оконные проемы и т.д.) [60]. В дискретном производстве высокотехнологичной продукции с длительным сроком активного существования, которым является машиностроение и его отрасли, в частности, ракетно-космическая и приборостроительная, проектирование осуществляется с применением САПР и СУД.

Набор средств и настроек для хранения и обработки информации об изделии на различных этапах его ЖЦ принято называть системой управления ЖЦ изделия, или PLM-системой. Информацию об изделии, содержащуюся в PLM-системе, принято называть цифровым макетом изделия [61]. Анализируя смысл данного термина, описанный в работах [62, 63], применительно к объекту исследования настоящей работы, можно сделать вывод, что совокупность ИМИ, КД, ЭСИ входит в поня-

тие цифрового макета, наряду с другими данными, описывающих изделие, его создание и обслуживание.

Внедрение PLM-систем и технологий на промышленном предприятии – это целый комплекс мероприятий различного характера, включая исследование и анализ области внедрения, разработку моделей деятельности «как есть» и «как должно быть» в соответствии с положениями реинжиниринга, описанными в работах М. Хаммера, Дж. Чампи, Б. Андерсена, А.В. Шеера и других классиков процессного подхода к управлению [8, 45, 46, 108-110].

Таким образом, внедрение PLM-технологий включает принятие соответствующих организационно-управленческих решений, а также изменение комплекса программного обеспечения, в связи с разработкой и внедрением ИС для поддержки различных этапов ЖЦ изделий и интеграции этих программных продуктов между собой в рамках ИИС.

На основании аналитического обзора работ [7, 45-63] можно сделать вывод, что сегодня PLM является зрелой методологией, в рамках которой уже разработаны примеры лучших практик информационного сопровождения проектирования и производства на машиностроительных предприятиях аэрокосмической, автомобильной, оборонной, энергетической и других отраслей промышленности.

В то же время, несмотря на значительную развитость PLM-решений, остались до сих пор нерешенными некоторые ключевые вопросы, один из которых – это обмен информацией между разными ИС в рамках единого PLM-комплекса. До сих пор количество форматов данных, используемых в ИС растет, при этом выделяют следующие группы стандартов представления информации [64]:

- международные стандарты представление информации о продукте: ISO/IEC 10303 (STEP) и ISO 13584 PLib;
- стандарты представления текстовой и графической информации: ISO 8879 SGML; ISO/IEC 10179 (DSSSL); ISO/IEC IS 10744 (HyTime); ISO/IEC 8632; ISO/IEC 10918 (JPEG); ISO 11172 MPEG2 (MPEG); ISO/IECS 13522 (MHEG).

Другие существующие государственные и международные регламентирующие документы, которые принято называть CALS-стандартами, например, стандарты серии ISO 10303 и их аналоги, принятые в РФ (семейство рекомендаций Р 50.1, семейство стандартов ГОСТ Р ИСО 13584, ГОСТ 2.051-2006, ГОСТ 2.0523-2006, ГОСТ 2.0523-2006, а также ГОСТ Р 53791-2010, ГОСТ 2.103.68 и т.д.), в основном, определяют общий подход, способы представления и интерфейсы доступа к данным различного типа, вопросы защиты информации и ее электронной авторизации (цифровой подписи). Однако, они не содержат готовых моделей и алгоритмов для управления жизненным

циклом объектов проектирования, которые могут быть использованы в качестве основы для разработки соответствующего программного обеспечения.

Данная ситуация показывает необходимость дальнейшего развития PLM-решений в этом направлении и обуславливает актуальность настоящей работы в части создания новых моделей и алгоритмов взаимодействия различных ИС, которые обеспечивают информационное сопровождение процесса разработки высокотехнологичной продукции с длительным сроком активного существования за счет автоматизированного управления жизненным циклом объектов проектирования в рамках интегрированной информационной среды.

### **1.3.2 Анализ состава интегрированной информационной среды и взаимодействия ее типовых программных компонент**

Техническую сторону ЕИП представляет интегрированная информационная среда (ИИС), которая реализуется посредством программной интеграции ИС, обеспечивающих автоматизацию и поддержку различных этапов ЖЦ продукции. Центральное место в организации ИИС играет информационная система управления данными (СУД) о продукции, называемая также PDM (от англ. Product Data Management). В соответствии с методологией PLM [7, 49, 56-58], кроме структурированного хранения информации об изделии, для реализации полноценного управления данными СУД должна поддерживать интеграцию с другими программными средствами в рамках ИИС.

Основную функциональную роль в процессах разработки изделий играют САПР, которые принято классифицировать по отраслевому и целевому назначению. Применительно к объекту исследований, можно представить следующую последовательность использования различных видов САПР при проектировании высокотехнологичной продукции с длительным сроком активного существования на примере приборостроения:

- с помощью ИС механического проектирования, MCAD (от англ. Mechanical Computer Aided Design), создаётся геометрическая модель изделия;
- описание электронных характеристик изделия формируется с применением ИС электрического проектирования, ECAD (от англ. Electronic Computer Aided Design);
- инженерный анализ электронных и механических характеристик изделия выполняется в ИС инженерного расчета, CAE (от англ. Computer Aided Engineering) на основе моделей из MCAD- и ECAD-систем;

- на этапе технологического проектирования информация из MCAD-систем используется в качестве входных данных для ИС технологической подготовки к производству, САМ (от англ. *Computer Aided MANufacturing*).

Для технической реализации задач взаимного сопряжения программных компонентов ИИС необходимо наличие соответствующей телекоммуникационной среды, обеспечивающей непрерывный обмен данными согласно коммуникационным стандартам протоколов передачи данных модели OSI, основные положения которой изложены в [65, 66].

Для обеспечения информационной интеграции программных компонентов ИИС существует ряд международных стандартов электронного представления данных, (например, IGES, ISO 10303 STEP, ISO 13584 PLIB, ISO 15531 MANDATE и т.д), которые регламентируют представление данных с помощью стандартизированных языков разметки (например, ISO 8879 SGML, HTML, 3DXML, JSON, 3DPDF, JSON, DWF, eDrA wings, JT) [67]. Конфиденциальность и аутентичность информации в ИИС обеспечивается с помощью ряда различных криптографических стандартов (шифрования, хеширования, беспроводных коммуникаций, инфраструктуры открытых ключей и т.д.) и технологий электронной подписи (простая, усиленная, квалифицированная, неквалифицированная) [67].

Соответствие ИС, которые подлежат интеграции в рамках ИИС, вышеупомянутым международным стандартам представления данных не является единственным условием их программной взаимосвязи. ИС должны поддерживать технологии расширения их стандартных функциональных возможностей, например, *API* (от англ. *ApplicAtion ProgrAm InterfAce*) или аналогичных инструментов и т.д. Это особенно важно при необходимости объединения в рамках ИИС нескольких САПР, которые используются для разработки информационных моделей изделия, описывающих его с разных точек зрения в рамках сквозного проектирования.

Интегрированное объединение различных ИС в рамках единую среду для поддержки ЖЦ изделий согласно методологии PLM принято называть PLM-решением. Сегодня наиболее популярными PLM-решениями являются программные продукты крупнейших производителей программного обеспечения, таких как DAssAult Systemes, Siemens, SAP, PTC, Лоция Софт, АСКОН, Oracle, UniGraphics и др.: ENOVIA, SmarTeam, Вертикаль, Lotsia PLM, Oracle PLM, Windchill, SAP PLM, TeamCenter, T-FLEX DOCs, ЛОЦМАН:PLM, 1С: PDM 2.0. [69-74].

В задачи настоящей работы не входит создание самостоятельного программного обеспечения для информационной поддержки локальных

этапов ЖЦ изделий, таких как проектирование или управление данными. Существует множество готовых программных продуктов от крупнейших российских и мировых компаний-разработчиков ИС, соответствующих положениям RGD-методологии. Однако, несмотря на множество отраслевых решений для автоматизации локальных этапов ЖЦ изделий и набора интегрированных между собой программных средств, задача управления разнотипными взаимозависимыми объектами проектирования в едином информационном пространстве остается актуальной в связи с отсутствием готовых структурированных моделей и алгоритмов построения ИИС.

Таким образом, применительно к объекту исследования настоящей работы, возникает задача разработки метода построения ИИС на базе интеграций программных средств (САПР и СУД), используемых при создании высокотехнологичных изделий с целью управления жизненным циклом объектов проектирования.

Кроме того, в связи с центральной ролью СУД в ИИС, необходимо выработать правила хранения и управления информацией с точки зрения динамики изменения стадий жизненного цикла объектов проектирования. Для этого, прежде всего, следует рассмотреть типовую базовую функциональность существующих систем управления данными и возможность их адаптации к задачам управления жизненным циклом объектов проектирования. При этом следует учитывать выявленную специфику процессов разработки высокотехнологичной продукции с длительным сроком активного существования на примере космического приборостроения относительно стадий испытаний НЭО.

### **1.3.3 Анализ типовых функциональных возможностей систем управления данными для задач проектирования**

С помощью СУД осуществляется хранение и обработка больших массивов инженерно-технической информации, необходимых на этапах проектирования, производства или строительства, а также поддержка эксплуатации, сопровождения и утилизации технических изделий. В СУД агрегируются данные любых форматов и типов, и предоставляются пользователям уже в структурированном виде в соответствии с особенностями конкретного предприятия.

На основании работ [72-75], можно сделать вывод, что основным отличием СУД от систем электронного документооборота является их направленность на проектную специфику деятельности. Разработка изделия выполняется в рамках проекта, который является хранилищем всех связанных с изделием информационных сущностей (ЭСИ, ИМИ, КД). В СУД предусмотрено логическое разделение информационных сущностей и представление их в виде структурированной совокупности

однотипных объектов, например, дерево проектов, дерево изделий, дерево документов, дерево элементов, дерево материалов и т.д. При этом все информационные сущности, относящиеся к одному изделию, связаны между собой.

На рисунке 1.4 показана взаимосвязь объектов проектирования из иерархических деревьев различных типов. Например, Проект\_2 из существующего множества проектов, сопровождается набором документов, объединенных в дерево документов. В рамках Проекта\_2 осуществляется разработка Изделия\_1, которое состоит из множества Элементов, образующих структуру изделия. Информация об Изделии\_1 фиксируется в связанном с ним Документе\_n.

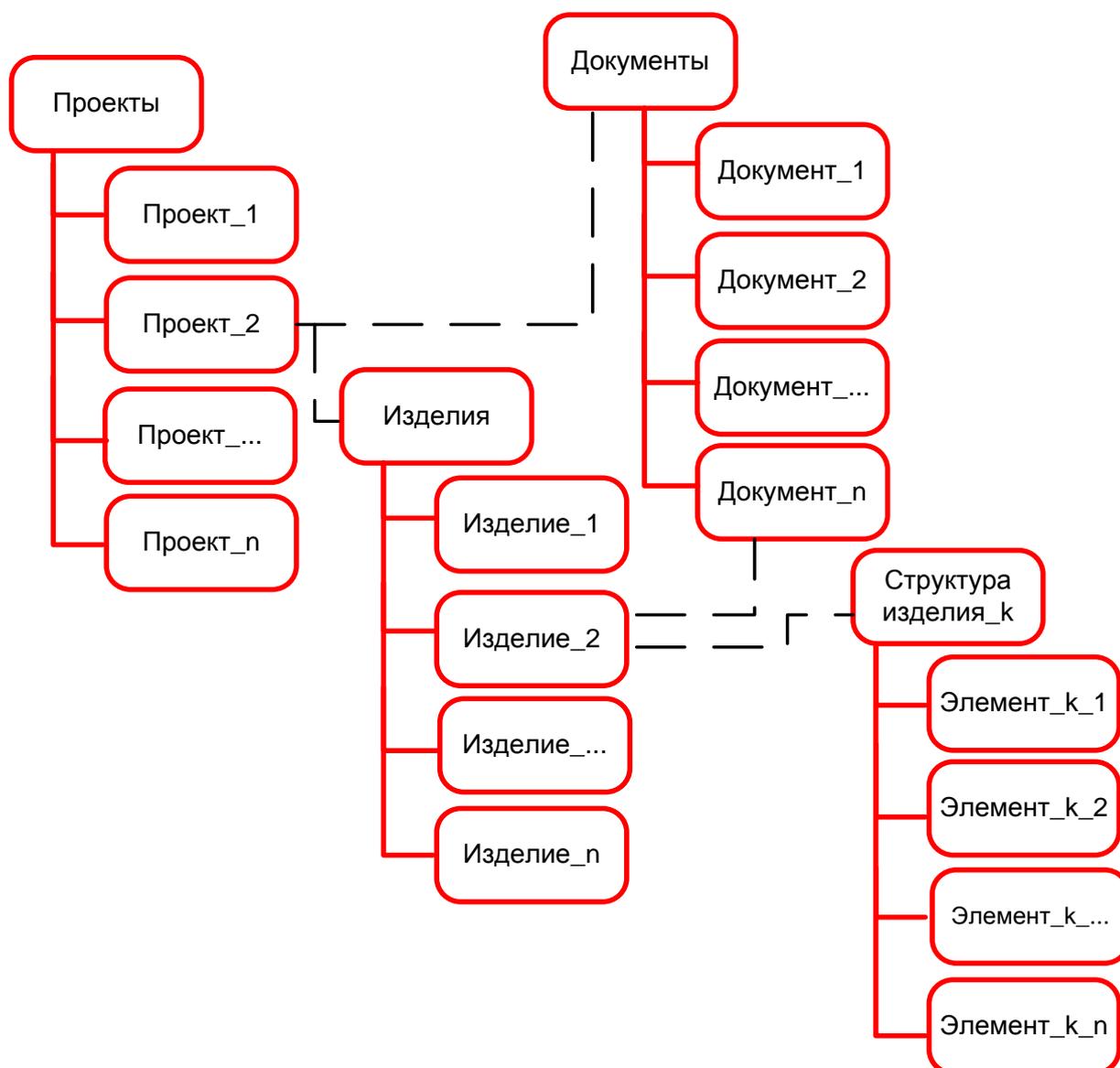


Рис.1.4. Схематичное представление иерархии и взаимосвязи информационных сущностей в СУД в виде структурированных деревьев

В СУД выделять физические сущности: изделие и составляющие его части, которые называют элементами [75, 75]. Количество и вариан-

ты соединения элементов между собой определяют структуру изделия. Таким образом, представление ЭСИ в СУД реализуется объектами типа «элемент ЭСИ». Документ является информационной сущностью, которая определяет готовность изделия к производству, т.е. стадию его ЖЦ.

Документом является информационная модель изделия и формируемая на ее основе конструкторская документация (КД). На рисунке 1.5 показана взаимосвязь между объектами проектирования в терминах СУД.

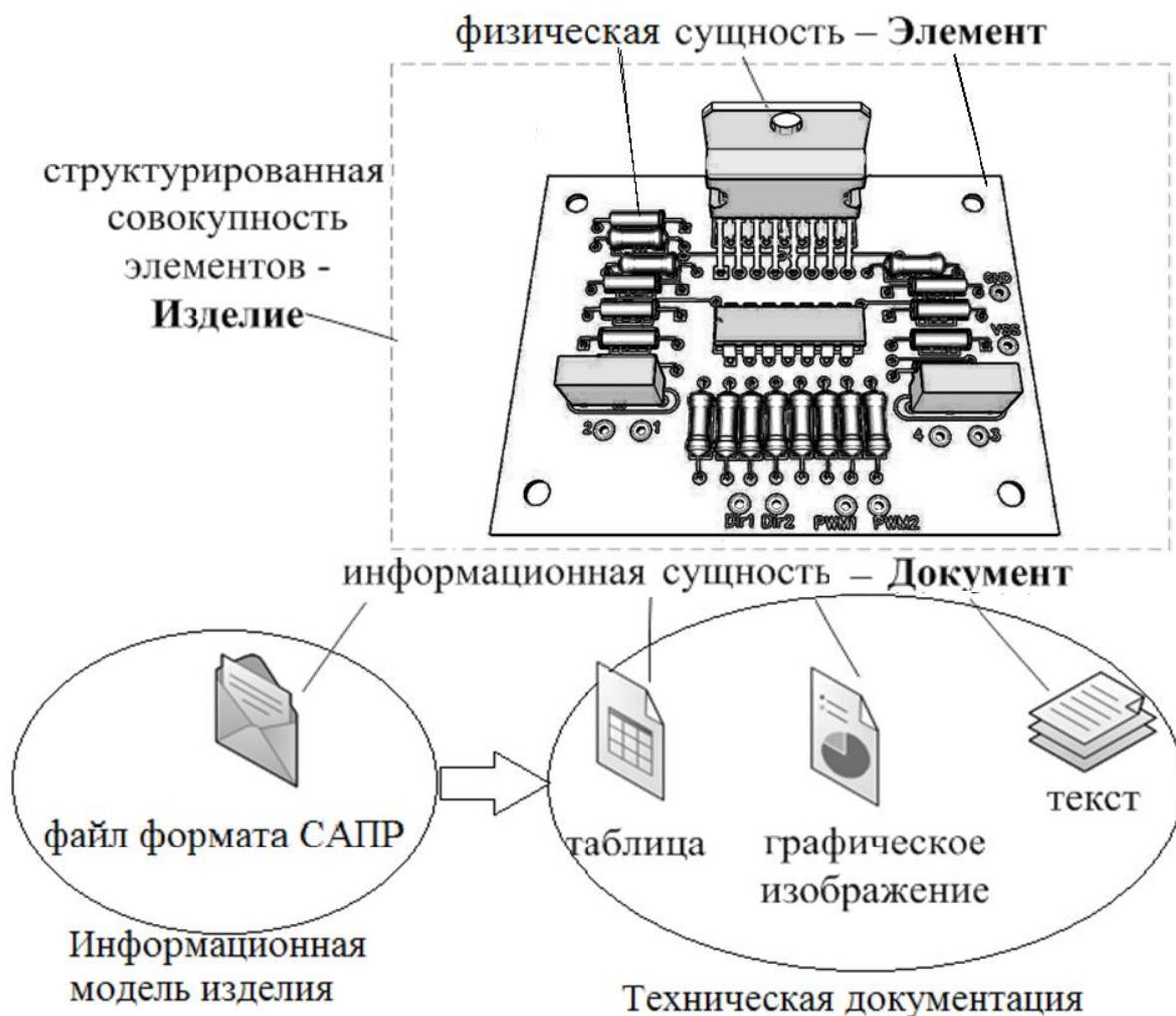


Рис.1.5. Представление объектов проектирования в терминах СУД

Для анализа возможности применения СУД к задачам управления жизненным циклом объектов проектирования, следует детально рассмотреть типовые этапы ЖЦ документа и элемента в данной категории ИС. На основании проведенного анализа маршрутов ЖЦ объекта типа «Документ» в наиболее популярных СУД (Enovia SmarTeam, Lotsia PLM, Windchill, TeamCenter, T-FLEX DOCs, ЛОЦМАН:PLM, 1С: PDM) [49, 71-75], был выявлен типовой набор состояний документа и их логическая последовательность. В таблице 1.1 показаны типовые этапы ЖЦ документа в терминах СУД.

**Типовые этапы ЖЦ документов в терминах СУД**

Стадия ЖЦ	Описание
Новый	На начальном этапе своего ЖЦ документ в стадии «новый» находится локально на компьютере его создателя и не доступен для других пользователей СУД
Опубликован	Документ перемещается на сервер хранения данных и становится доступен для просмотра и изменения уполномоченным пользователям СУД
Выписан	При редактировании опубликованного документа он копируется на компьютер пользователя, осуществляющего эту операцию, и блокируется от просмотра и изменения другими пользователями ИС. По завершении редактирования автоматически создается новая версия документа и размещается на сервере.
Выпущен	Документ находится на сервере и доступен для просмотра уполномоченным пользователям СУД без возможности внесения изменений
Устаревший	По истечении срока действия документ считается устаревшим и перемещается в архив.

На начальном этапе своего ЖЦ документ в стадии «новый» находится локально на компьютере его создателя и становится доступным для других пользователей ИС только на следующем этапе ЖЦ. При этом документ переходит в стадию ЖЦ «опубликован» и перемещается на сервер хранения данных, поскольку СУД предназначена для одновременной работы нескольких пользователей. В случае редактирования опубликованного документа он считается «выписанным». При этом документ копируется на компьютер пользователя, осуществляющего эту операцию, и блокируется от изменения другими пользователями ИС. По завершении редактирования автоматически создается новая версия документа и размещается на сервере. Если необходимо зафиксировать содержимое документа без возможности его изменения и/или показать завершенность текущей версии, стадия ЖЦ меняется на «выпущенный». По истечении срока действия документ считается «устаревшим» и перемещается в архив.

Объекты типов «Элемент» и «Документ» в СУД автоматически связаны между собой, поэтому их ЖЦ должен рассматривать в зависимости друг от друга. В таблице 1.2 показаны типовые этапы ЖЦ элемента в терминах в СУД.

Стадии ЖЦ элементов в терминах СУД

Стадия ЖЦ	Описание
Новый	На начальном этапе ЖЦ (при создании) элемент находится в стадии «новый»
Разработка	По этому элементу запущены процессы проектирования, что включает разработку ТД
Готов к производству	Элемент готов к запуску в производство, т.е. разработаны все связанные с ним документы, необходимые для производства
Неактивный	Элемент не находится в стадии разработки или изготовления/производства

На начальном этапе своего ЖЦ элемент создается в СУД в стадии «новый». Как только по данному элементу изделия запущены процессы проектирования, он переходит в состояние «в разработке». При этом процессы проектирования означают разработку КД. Элемент считается готовым к производству в случае завершения всех процессов проектирования. Это означает, что все, связанные с данной физической сущностью документы разработаны и согласованы. Если элемент не находится в стадии разработки или изготовления/производства и не является «Новым», состояние его ЖЦ меняется на «Неактивный».

Однако, несмотря на наличие в СУД типовых функциональных возможностей по управлению состоянием документа и элемента, основные положения которого приведены в таблицах 1.1, 1.2, анализ работ [49, 71-75] выявил ряд актуальных проблем управления ЖЦ объектов проектирования типов «документ» и «элемент»:

- типовые этапы ЖЦ документа в СУД не учитывают атрибут подлинности документа согласно [30];
- отсутствует взаимозависимость состояний ЖЦ элемента и документов, связанных с ним;
- отсутствует взаимозависимость документов ИМИ и КД;
- последовательность этапов ЖЦ элемента характеризует изменение состояния проектируемого изделия, а не его составных частей;
- в связи с тем, что составные части изделия (его элементы), как правило, являются покупными компонентами, необходимо предусмотреть проверку возможности их использования, например, наличие на складе. В случае космического приборостроения данная ситуация осложняется ограничениями на применение ЭРИ [24].

Таким образом, возникает задача разработки алгоритмического и программного обеспечения среды автоматизированного управления жизненным циклом объектов проектирования, с учетом взаимозависимости

мости стадий их ЖЦ, положений PLM-методологии, а также требований ЕСКД, в частности [29, 30]. Для этого следует описать структуру взаимосвязей объектов проектирования и динамику изменения стадий их жизненного цикла в виде формализованных моделей и алгоритмов. Это, в свою очередь, потребует проведения теоретических и практических исследований в области формальных методов управления связанными объектами.

## **1.4 Поиск формальных методов управления связанными объектами**

### **1.4.1 Аналитический обзор формальных методов**

Анализ особенностей процессов проектирования высокотехнологичной продукции с длительным сроком активного существования на примере отрасли космического приборостроения позволяет сделать вывод о тесной взаимосвязи стадий ЖЦ объектов проектирования и их зависимости друг от друга, что соответствует методологии проектирования RGD. Поэтому для разработки правил автоматического изменения стадий ЖЦ объектов проектирования в ИИС необходим соответствующий математический аппарат, который позволит формализовать поведение взаимосвязанных сущностей. Такой математический аппарат для спецификации, разработки и верификации программного и аппаратного обеспечения, лежит в основе группы техник, называемых формальными методами [77]. Формальные методы чаще всего находят применение в проектировании высокоточных систем, в случае высоких требований к безопасности и надежности, как, например, в области разработки КА. Формальные методы занимают приложением довольно широкого класса фундаментальных техник теоретической информатики: разные исчисления логики, формальных языков, теории автоматов, формальной семантики, систем типизации и алгебраических типов данных [78].

Применительно к объекту исследований настоящей работы, следует определить набор формальных методов, который позволит разработать алгоритмическое и программное обеспечение среды управления жизненным циклом объектов проектирования. При этом необходимо решить следующие задачи:

- анализ и выбор подходов к разработке программных систем;
- анализ и выбор методов графического описания и документирования программных систем;
- поиск математических методов формализации статического состояния многосвязных объектов и их динамического поведения;
- анализ возможности применения идей распараллеливания и синхронизации операций к задаче изменения стадий ЖЦ объектов проектирования.

### 1.4.2 Анализ и выбор подходов к разработке программных систем

В основе современных методов проектирования программного обеспечения лежат положения методологии структурного анализа, которые реализованы в исследованиях Дугласа Росса, Йордана-ДеМарко и Гейна-Сарсона, Джексона-Орра и Дж.Д Варнира [79]. Анализ работ Буча Г., Рамбо Д., Джекобсона А., Маклакова С.В., Дурнова П.А., Вендрова А.М., Йордон Эд. [79-82] позволяет сделать вывод, что в настоящее время принято выделять следующие два подхода к разработке многокомпонентных программных систем:

1. Функционально-модульный подход, который является частью более общего структурного подхода. В его основу положен принцип алгоритмической декомпозиции, структура системы описывается в терминах иерархии ее функций и передачи информации между отдельными функциональными элементами.

2. Объектно-ориентированный подход, который использует объектную декомпозицию. При этом структура системы описывается в терминах множества объектов и связей между ними, а поведение системы описывается в терминах обмена сообщениями между объектами. Такой подход позволяет получить достаточно качественное представление о системе, но зависит от цели моделирования и, соответственно, подходов к моделированию, т.к. по своей природе все методы объектно-ориентированного моделирования являются эмпирическими и не описываются строго формализованными процедурами [84]. **Ошибка! Источник ссылки не найден.**

Несмотря на критику объектно-ориентированного подхода, изложенную в работах А. Степанова, Р. Гэбриэла, П. Грэма, Р. Столмана, Т. Потока, Е. Дейкстры [83-86], в настоящее время он считается основным при разработке программного обеспечения.

Объектно-ориентированный подход состоит в описании структуры и поведения проектируемой системы. Выделение частей производится таким образом, чтобы каждая имела минимальный по объёму и точно определённый набор выполняемых функций. Базисом объектно-ориентированного подхода, отражающими его суть, являются понятия класса и объекта, абстракции, инкапсуляции, наследования и полиморфизма [79-94]. Эти понятия лежат в основе принципа подстановки Б. Лисков [90, 91]

Анализируя работы Г. Буча, Д. Лоффредо, Зинченко Р.Е, Дрождина В.В [92-94], а также практики разработки программных систем, в частности, [95], можно сделать вывод, что положения объектно-ориентированного подхода являются базисом представления комплекса программных решений в семантике ИТ-архитектуры согласно [95]. Учитывая значения понятий ИИС и PLM, описанные ранее в настоящей

работе, следует отметить, что структура ИИС, помимо требований прикладной области, должна проектироваться с учетом рекомендации по разработке ИТ-архитектуры. Таким образом, в связи с необходимостью сопряжения различных программных компонентов ИИС между собой с целью непротиворечивого представления объекта проектирования с различных точек зрения в разных программных системах, в основу программной архитектуры ИИС на базе интеграций САПР и СУД будут положены положения объектно-ориентированного подхода. Для практической реализации программного сопряжения информационных систем в рамках ИИС необходимо выполнить аналитический обзор и поиск методов интеграции приложений. В связи с прикладным характером этой задачи целесообразно решить ее в отдельной главе настоящей работы.

Поскольку центральное место в организации ИИС занимает СУД, возникает задача ее настройки к выявленным особенностям космического приборостроения. Информация в СУД хранится в базе данных (БД), схема которой, согласно [96], определяет правила работы с информацией.

В работах К. Дж. Дейта, М. Р. Когаловского, С. Д. Кузнецова [97-101] подчеркивается отличие терминов «модель данных» и «схема базы данных». Модель данных есть теория (инструмент) моделирования, а схема базы данных есть результат моделирования. Большинство современных ИС, как правило, реализованы согласно реляционной или объектно-реляционной модели данных.

В классической теории баз данных [101], модель данных есть формальная теория представления и обработки данных в СУБД, которая включает, по меньшей мере, три аспекта:

- аспект структуры (методы описания типов и логических структур данных в БД);
- аспект манипуляции (методы манипулирования данными);
- аспект целостности (методы описания и поддержки целостности БД).

Разработка концептуальной модели данных является начальным этапом проектирования БД и включает в себя описание информационных объектов (понятий предметной области) и связей между ними, а также описание ограничений целостности. Работы П. Чена, К. Дж. Дейта, М. Р. Когаловского, С. Д. Кузнецова показывают, что на этапе концептуального проектирования модели БД применяются в том числе и базовые положения объектно-ориентированного подхода [97].

В настоящей работе семантическое моделирование и объектно-ориентированный подход будут использованы на этапе концептуальной разработки модели данных СУД для управления жизненным циклом

объектов проектирования в рамках ИИС. Последующие этапы проектирования БД (разработка логической и физической модели) не подлежат рассмотрению, поскольку на этих этапах выполняется привязка к конкретной СУБД, что не может считаться общим решением. Кроме того, прикладная направленность настоящей работы ориентирована на управление ЖЦ объектов проектирования, поэтому будет рассмотрен участок модели данных СУД, который непосредственно описывает представление объектов проектирования и правила манипулирования этой информацией.

Для концептуального проектирования программной архитектуры ИИС необходимо найти способ описания статического состояния исследуемых объектов, а также их динамического поведения. Это может быть реализовано с помощью формального аппарата графических методов описания и документирования программных систем.

#### **1.4.3 Анализ и выбор методов графического описания и документирования программных систем**

Для создания моделей анализа и проектирования программных систем используют языки визуального моделирования. Появившись сравнительно недавно, в период с 80-х гг. XX в., эти языки уже имеют представительную историю развития. В настоящее время различают три поколения языков визуального моделирования, в том числе, нотации моделирования на основе методов структурного анализа, которые реализованы в исследованиях Дугласа Росса (метод структурного анализа и проектирования SADT), Йордана-ДеМарко и Гейна-Сарсона (метод моделирования потоков данных DFD), Питера Чена, Джексона-Орра и Дж.Д. Варнира (модели представления структур данных) [102]. Чрезмерное количество языков визуального моделирования [103] привело к необходимости разработки унифицированных языков 3-го поколения. В качестве стандартного языка третьего поколения был принят UML (от англ. *Unified Modeling Language*), создававшийся в 1994-1997 годах с участием Г. Буча, Дж. Рамбо, А. Джекобсона [94]. **Ошибка! Источник ссылки не найден.** В основе UML лежат положения объектно-ориентированного анализа Э. Йордона, структурного анализа Меррора, Шлеера, а также методов объектно-ориентированного проектирования Г. Буча.

UML был создан для определения, визуализации, проектирования и документирования в основном программных систем, но может применяться также для моделирования бизнес-процессов, системного проектирования и отображения организационных структур. UML включает в себя множество графических диаграмм, которые позволяют описывать структуру и поведение проектируемой системы.

Статическую структуру проектируемой системы и ее состав в UML описывают диаграммы компонентов и классов. Диаграмма компонентов показывает разбиение программной системы на структурные компоненты (файлы, библиотеки, модули, исполняемые файлы и т.д.) и связи между ними. Компоненты связываются через зависимости, когда соединяется требуемый интерфейс одного компонента с имеющимся интерфейсом другого.

Диаграмма классов описывает структуру системы, показывая её классы, их атрибуты и операторы, а также отношения между классами. Это позволяет показать типизацию данных в виде классов и связи между ними на этапе концептуальной разработки объектно-ориентированной модели БД.

Для моделирования динамического поведения системы в UML предназначены диаграммы последовательностей и состояний. Диаграммы последовательностей позволяют показать взаимодействия объектов (разных классов), упорядоченные по времени их проявления. Диаграммы состояний описывают процесс изменения состояний объектов только одного класса, которые могут быть вызваны внешними воздействиями со стороны других объектов или извне. Таким образом, с помощью диаграммы состояний можно описать возможные последовательности состояний и переходов, которые в совокупности характеризуют поведение элемента модели в течение его ЖЦ. Диаграмма состояний представляет собой ориентированный граф для конечного автомата, в котором вершины обозначают состояния, а ребра показывают переходы между двумя состояниями. При этом поведение моделируется как последовательное перемещение по графу состояний от вершины к вершине по связывающим их дугам с учетом их ориентации.

При моделировании поведения проектируемой системы следует представить процесс изменения ее состояний и детализировать особенности алгоритмической и логической реализации операций, выполняемых системой или ее пользователями. В настоящее время, согласно исследованиям М. Хаммера, А. Белайчука, Ю. Вагнер, И. Озеровой [108-110], наиболее распространенным способом моделирования деятельности, с точки зрения процессного подхода к управлению, считается нотация BPMN (от англ. *Business Process Model And NotAtion*). Принятые в BPMN правила использования графических элементов и связей между ними, как и UML-диаграммы, также основаны на положениях теории графов: событие или процесс представляет собой вершину, а переход между ними является направленным ребром (дугой). Диаграммы UML и BPMN для описания деятельности позволяют показать не только логику выполнения процесса, но и состояния объектов. Таким образом, для формализации операций пользователей ИИС, связанных с изменением

состояний ЖЦ объектов проектирования, также целесообразно использовать инструментарий UML.

В связи с наличием в UML мощного инструментария для проектирования программных систем с целью минимизации количества используемых нотаций, было решено выбрать метод UML в качестве основного средства представления формальных моделей состава, структуры и операций изменения разнотипных взаимозависимых объектов при проектировании высокотехнологичной продукции в интегрированной информационной среде.

Подводя итог краткому описанию средств формального аппарата описания программных систем и бизнес-процессов, которые применимы к направлениям исследований настоящей работы, можно сделать вывод, что в основе рассмотренного инструментария лежит теория графов. В свою очередь, теория графов относится к математическим методам описания сложных систем. Таким образом, возникает задача анализа и выбора математических методов для формализации структуры разнотипных взаимозависимых объектов в статическом и динамическом состояниях.

#### **1.4.4 Поиск математических методов для формализации структуры взаимосвязей объектов и динамики изменения их состояний**

Поскольку одним из направлений настоящей работы является описание структуры взаимосвязей разнотипных взаимозависимых объектов и динамики изменений их жизненного цикла с учетом специфики процессов проектирования высокотехнологичной продукции с длительным сроком активного существования, возникает задача поиска соответствующих методов формализации. Искомые методы и подходы должны предоставлять математический аппарат, с помощью которого возможно выполнить описание структуры исследуемых объектов в статическом состоянии, а также динамику их изменения.

Анализ ЖЦ объектов проектирования на примере изделий отрасли космического приборостроения, основные этапы которого описаны в работах [16-18, 111, 112], показывает дискретность состояний ЖЦ исследуемых объектов. Таким образом, для описания стадий ЖЦ объектов проектирования целесообразно применение методов дискретной математики.

Из множества разделов дискретной математики [113] следует выбрать те, методы которых позволяют формализовать структуру взаимосвязей объектов проектирования и динамику изменения их ЖЦ. Такая формулировка задач соответствует положениям системного подхода, в основе которого лежит рассмотрение объекта как целостного множества элементов в совокупности отношений и связей между ними. Основопо-

ложниками системного подхода являются Л. фон Берталанфи, А.А. Богданов, Г. Саймон, П. Друкер, А. Чандлер. Согласно [114], общие формальные свойства систем могут быть аксиоматизированы в языке теории множеств, основные понятия которой изложены в работах Б. Больцана «Парадоксы бесконечного» 1850, Г. Кантора 1879—1897, Д. Гильберта, Б. Рассела «Начала математики» 1910—1913, 1904—1908 гг. Э. Цермело и др. [114].

Учитывая ранее описанные взаимозависимости различных типов объектов проектирования, целесообразно формализовать эти объекты (изделие, ИМИ, КД, ЭСИ) и связи между ними с помощью базовых понятий теории множеств. Полученные таким образом отношения между объектами проектирования будут отражать типизацию объектов, что соответствует логической структуре представления данных в ИС.

Таким образом, применение аппарата теории множеств позволит структурировать исследуемые объекты и связи между ними с целью дальнейшего использования полученных выражений в качестве базиса для разработки алгоритмического и программного обеспечения. Кроме того, теоретико-множественные отношения являются основой реляционной алгебры в теории БД [106], элементы которой будут применяться на этапе концептуального проектирования архитектуры ИИС.

Однако, теория множеств позволяет формализовать структуру исследуемых объектов в статичном состоянии, а управление ЖЦ требует инструмента, который поможет представить динамику изменения объектов. Одним из таких инструментов является теория автоматов [115], используя базовые положения которой можно представить стадии ЖЦ объектов в виде состояний, переходы между которыми выполняются по определенным правилам. Для визуализации изменения состояний целесообразно применять теорию графов, которая также относится к дискретной математике [114]. Теория автоматов и теория графов тесно связаны между собой, поскольку позволяют описать дискретный процесс изменения объекта под воздействием определенных сигналов.

Несмотря на то, что основные понятия теории графов получили свое развитие задолго до появления теории множеств как самостоятельной научной дисциплины, формальное определение графа принято представлять в теоретико-множественных терминах. Граф можно рассматривать как графическую нотацию для бинарного отношения двух множеств. Бинарное отношение состоит из таких списков элементов, которые содержат только два элемента некоторого множества [106].

Родоначальником теории графов считается Леонард Эйлер, но как отдельная математическая дисциплина теория графов была впервые представлена в работе венгерского математика Кенига в 30-е годы XX столетия. Основные исследования в теории графов и ее практических

приложениях были выполнены в работах Р. Акофа, Р. Беллмана, Дж. Данцига, Г. Куна, Т. Саати, Р. Чермена, А. Кофмана, Р. Форда и др. [117].

Некоторые положения теории графов легли в основу теории автоматов, развитие которой связано с созданием сложных технических средств с программным управлением в 50-х гг. XX в. [115]. В связи с ограниченным количеством состояний исследуемых объектов, можно сделать вывод о применимости модели конечного автомата к описанию динамики изменения их ЖЦ. Поскольку у каждого типового объекта проектирования (Изделие, ИМИ, КД, ЭСИ) переход между стадиями ЖЦ зависит от текущей стадии и операции, применяемой к объекту в текущем состоянии, целесообразно использование модели автомата Мили [115].

В настоящее время теории графов и автоматов широко применяются в программировании, электротехнике, социологии, автоматическом управлении и других предметных областях, где необходимо представить динамическое изменение состояний сложной системы. Поэтому в данной работе для формализации динамики изменений жизненного цикла объектов проектирования будут использованы элементы теорий графов и автоматов.

Представляя динамику изменений стадий жизненного цикла объектов проектирования в терминах теории графов [117], можно отметить следующие положения:

- стадия ЖЦ каждого исследуемого объекта представляет собой состояние автомата;
- совокупность состояний автомата образует конечное множество вершин графа;
- переход между состояниями автомата осуществляется под воздействием входных сигналов и зависит от текущего состояния и входных сигналов;
- порядок изменения состояний автомата (стадий ЖЦ) отражается с помощью направленных ребер графа, связывающих его вершины;
- поскольку фактические действия по изменению стадий ЖЦ объектов проектирования в ИИС выполняются автоматизировано (людьми) или автоматически (функциями программного обеспечения), то ребра графа отражают работы на протяжении ЖЦ объектов;
- с целью повышения наглядности графа рационально визуально разделить его вершины, соответствующие различным типам исследуемых объектов, с помощью раскраски их разными

цветами (однако, здесь не идет речь о правильной раскраске графа и нахождении его хроматического числа).

На основе данных положений можно сделать вывод о целесообразности формализации структуры изменений стадий ЖЦ объектов проектирования в виде раскрашенного ориентированного графа и методики получения этой модели. Данная методика и модель, в свою очередь, будут являться базисом для построения графических UML-диаграмм, которые иллюстрируют алгоритмы, необходимые при разработке программного обеспечения среды управления ЖЦ объектов проектирования.

#### **1.4.5 Распараллеливание и синхронизация задач при согласовании документов для автоматизации изменения состояний ЖЦ объектов проектирования в ИИС**

Специфика разработки высокотехнологичной продукции с длительным сроком активного существования в части сложной структуры процессов согласования документов обуславливает задачу распараллеливания операций пользователей СУД по изменению стадий ЖЦ объектов проектирования. В настоящее время теоретические и прикладные исследования в области распараллеливания и синхронизации операций наиболее развиты в сфере разработки операционных систем и многопроцессорных вычислений. Однако, концептуальные положения данной теории могут быть использованы, в том числе для решения некоторых задач настоящей работы.

Согласно [119], идея распараллеливания основана на том, что большинство задач может быть разделено на набор меньших задач, которые могут быть решены одновременно. Как правило, параллельные операции требуют синхронизации действий. Синхронизация процессов представляет собой приведение двух или нескольких процессов к такому их протеканию, когда конкретные стадии разных процессов совершаются в определенном порядке, либо одновременно. Синхронизация требуется в любых случаях, когда параллельно протекающим процессам необходимо взаимодействовать.

В настоящей работе под процессом понимается последовательность изменения стадий ЖЦ объектов проектирования, которая выполняется по определенному алгоритму или маршруту. В терминологии современных ИС маршрут прохождения данных во время их ЖЦ принято называть бизнес-процессом или потоком работ [104]. Наиболее распространенным считается англоязычный термин *WorkFlow*, в основе которого лежит один из методов структурного анализа – *IDEF3*. Анализируя исследования М. Хаммера, А. Белайчука, Ю. Вагнер, И. Озеровой [108-110], можно сделать вывод, что базисом *WorkFlow* является один из ме-

тодов *IDEF3*, PFD (от англ. Process Flow Description), графическая нотация которого представляет собой описание логической последовательности действий с помощью графических элементов.

Задача синхронизации выполнения взаимосвязанных работ различными исполнителями во времени относится к области сетевого планирования в рамках дисциплины управления проектами [13, 14]. Методы сетевого планирования ориентированы, в первую очередь, на нахождение оптимальной последовательности выполнения работ в условиях временных и материальных ограничений. Однако, в настоящей работе процесс согласования документов при проектировании продукции рассматривается не с точки зрения оптимальной загрузки исполнителей и/или синхронизации по времени выполнения, а на предмет равнозначности результата выполнения операции.

Рассматривая методологию Workflow в терминах сетевого планирования теории управления проектами [13, 14], следует отметить, что поток работ представляет собой ориентированный граф – последовательность вершин (узлов) и направленных связей между ними. Узел является ключевой точкой бизнес-процесса, в которой задаются параметры работы (исполнитель, контроллер, сроки и т.д.). Для определения направления движения потока работ от одного узла к другому используется стрелка с надписью, указывающей действие, которое будет выполняться (возврат к предыдущему либо переход к следующему узлу).

Для отражения сложного характера реальной деятельности в линейном порядке расположения узлов на диаграмме Workflow необходимо свести к минимуму сложные логические ветвления. Следует стремиться представить деятельность как взаимосвязанную совокупность инкапсулированных друг в друга процессов, число узлов в которых уменьшается по мере вложенности. Подобное упрощение приводит к элементарному процессу с единственным узлом – заданием для одного исполнителя.

Применяя вышеизложенные положения к задаче управления ЖЦ объектов проектирования, следует отметить, что операция согласования ИМИ и КД должна выполняться параллельно несколькими пользователями СУД. Таким образом, процесс согласования документа детализируется на множество дочерних элементарных заданий по числу исполнителей (согласующих лиц), что показано на рисунке 1.5.

Результатом выполнения каждого из элементарных заданий (рисунок 1.5) является отметка о согласовании документа данным исполнителем. Данная отметка может быть реализована в качестве значения «согласовано» или «не согласованно» в свойстве объекта элементарного задания по согласованию.

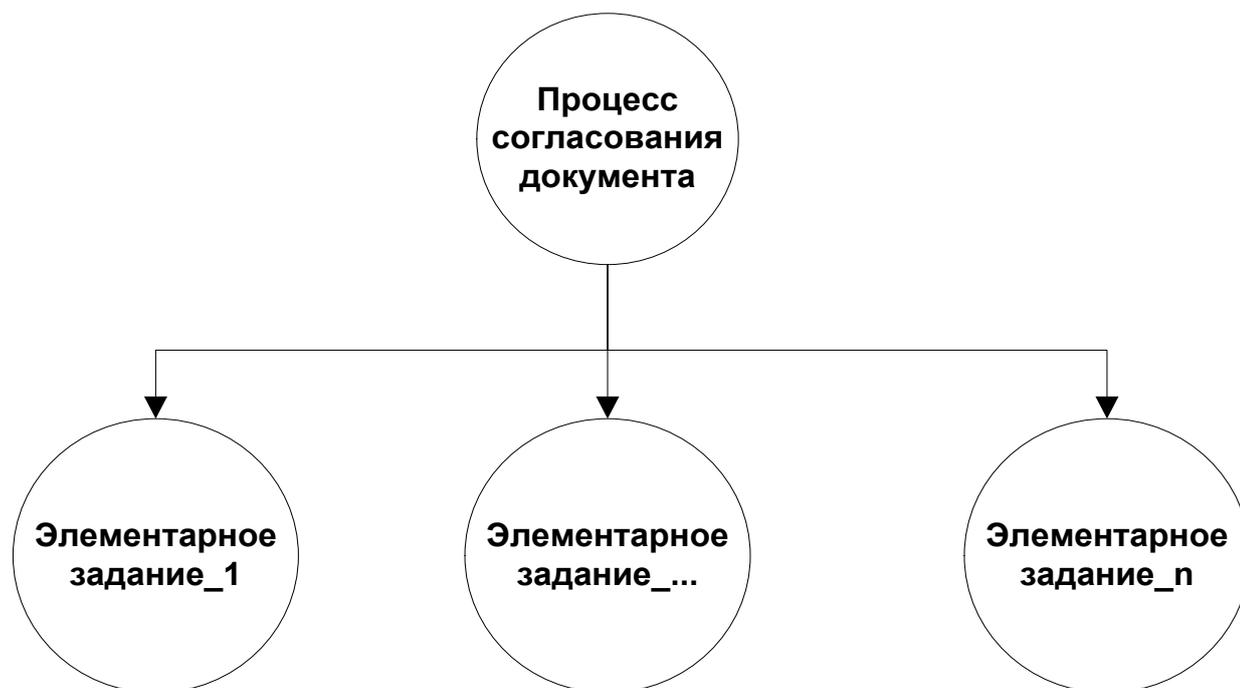


Рис.1.5. Распараллеливание процесса согласования документа

После того, как все исполнители элементарных заданий их выполнили, необходимо синхронизировать результаты, т.е. проверить, что все участники процесса согласования поставили отметки «согласовано». Только в этом случае документ считается согласованным и меняется его стадия ЖЦ, а также связанных с ним объектов проектирования. Если документ не согласован кем-либо из исполнителей, то разработчик его корректирует и публикует в СУД новую версию, которая доступна всем согласующим лицам для просмотра и внесения замечаний. Все эти итерации совершаются в рамках единого процесса согласования документа. Когда, наконец, все замечания согласующих лиц устранены и документ согласован каждым из них, документ считается согласованным.

Таким образом, для реализации процессов согласования документов в рамках ИИС при проектировании высокотехнологичной продукции с длительным сроком активного существования необходимо использовать некоторые положения теории управления проектами, а также теории распараллеливания и синхронизации операций. Решено использовать следующие положения теории управления проектами:

- представление работ в виде графа как при сетевом планировании;
- задание временных сроков выполнения и исполнителей работ.

Из теории распараллеливания и синхронизации операций решено использовать следующие положения:

- декомпозиция одной сложной задачи на несколько простых;
- синхронизация результатов выполнения простых задач.

Современные информационные системы электронного документооборота, модули которых могут входить в СУД, как правило, поддерживают автоматизацию процессов согласования документов [69, 70, 75]. При этом процесс согласования документа может выполняться по логически сложному маршруту, который определяет пользователь. Функциональные возможности гибкой настройки ЖЦ документа присутствуют практически в каждой современной системе электронного документооборота. Однако, проведенный анализ наиболее популярных систем электронного документооборота и СУД, не выявил готовых программно-методических решений по автоматизированному изменению состояния ЖЦ объекта типа «Изделие» или «элемент ЭСИ» в зависимости от результата согласования связанных с ним документов. Поэтому для автоматизированного управления ЖЦ разнотипных взаимозависимых объектов в ИИС, необходимо разработать программное обеспечение, сочетающее некоторые возможности систем электронного документооборота и управления данными с целью осуществления вышеописанного порядка согласования документов и автоматического изменения состояний ЖЦ связанных с согласуемым документами объектов проектирования других типов («Изделие», «элемент ЭСИ»). Вышеуказанные положения будут использованы в качестве концептуальной основы разрабатываемого программного обеспечения. Для этого, прежде всего, следует формализовать алгоритм действий пользователей ИИС с использованием средств UML.

### **1.5 Цель и задачи исследования**

На основании проведенного анализа существующих стандартов, регламентирующих общие требования по выполнению электронных конструкторских документов изделий машиностроения и приборостроения, а также программных методов и средств информационной поддержки ЖЦ изделий, можно представить процесс проектирования в виде последовательности действий, включающей разработку ИМИ и автоматическое/автоматизированное создание ЭСИ и КД. Однако, данные методические материалы не содержат подробных указаний по реализации процесса согласования электронных документов и изменению их состояний на этапе проектирования в рамках ИИС. Поэтому следует объединить порядок действий по получению ИМИ, ЭСИ и КД, регламентируемых в стандартах (рисунок 1.2), с типовым порядком проектирования высокотехнологичной продукции (рисунок 1.1). Однако, следует устранить выявленные недостатки обоих случаев: большое количество бумажной документации (рисунок 1.1.) и неопределенность момента получения твердой копии КД. Полученную последовательность действий можно представить следующим образом:

1. Разработка ИМИ в САПР.
2. Согласование ИМИ при отсутствии ошибок, в случае ошибок – возврат к шагу 1.
3. Автоматическое формирование ЭСИ.
4. Автоматическое формирование КД (в виде электронного документа).
5. Получение твердой копии КД, т.е. представление документов в бумажном виде;
6. Согласование КД при отсутствии ошибок, иначе – возврат к шагу 1.
7. В случае отсутствия ошибок КД утверждается.
8. Получение твердой копии КД (документ в бумажном виде).

Графическая иллюстрация описанной последовательности действий по разработке объектов проектирования в ИИС показана на рисунке 1.6. Представленная на рисунке 1.6 схема позволяет показать общий порядок создания разнотипных объектов проектирования в ИИС, но не содержит прямых описаний последовательности стадий их ЖЦ. Кроме того, возникает вопрос о возможности автоматического формирования корректной и полной ЭСИ в СУД на основе множества ИМИ, разработанных в различных не интегрированных между собой САПР. Также следует отметить, что существующие стандарты, методы и средства не предлагают готовых алгоритмических решений по реализации сложной структуры процессов согласования, которые имеют место при проектировании высокотехнологичной продукции с длительным сроком активного существования.

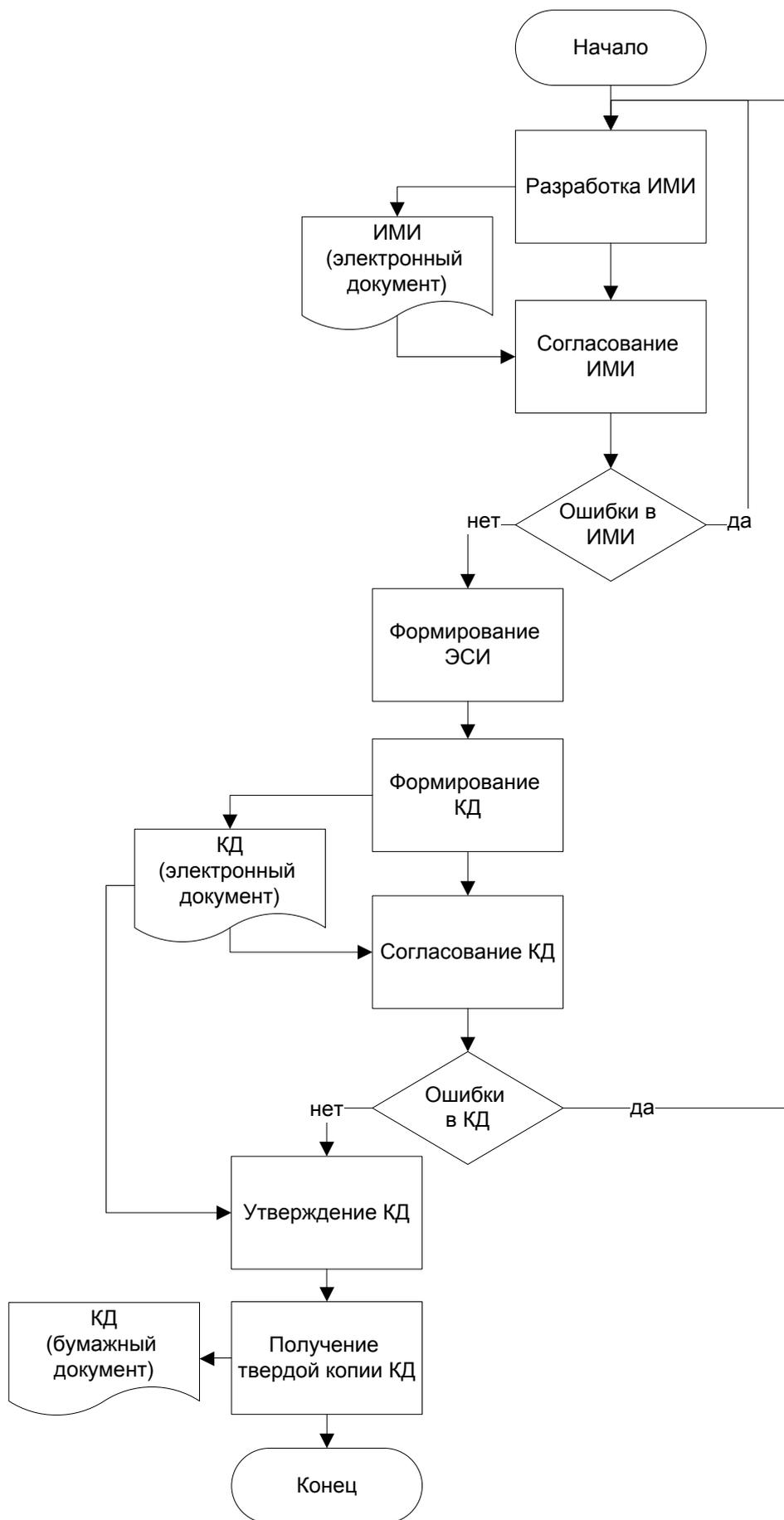


Рис.1.6. Последовательность действий получения ИМИ, ЭСИ и КД в ИИС

Выполненный аналитический обзор предметной области позволяет сделать вывод, что необходим способ автоматического изменения объектов проектирования от исходного состояния к целевому. Это должно включать структурированное хранение данных, контроль за обработкой информации и действиями участников процессов проектирования. Для обобщения вышеперечисленных аспектов предложено использовать термин «автоматизированное управление жизненным циклом разнотипных взаимозависимых объектов».

На основании материала, представленного в данной главе сделан вывод, что существующие методологии и стандарты информационного сопровождения жизненного цикла изделий, а также типовые функциональные возможности поддерживающих их программных систем не позволяют в полной мере реализовать автоматизированное управление жизненным циклом объектов проектирования. Поэтому необходимо развитие существующих методологий и средств управления жизненным циклом изделий с учетом особенностей процессов их проектирования.

Для разработки моделей и алгоритмов автоматизированного управления жизненным циклом разнотипных взаимозависимых объектов в ИИС необходимо выполнить следующие действия:

1. представить структуру взаимозависимостей разнотипных объектов проектирования и формализовать динамику изменений стадий их ЖЦ;
2. разработать метод построения ИИС, который обеспечит непротиворечивость представления информационных сущностей, создаваемых и используемых в различных САПР и СУД;
3. разработать комплекс программных решений для построения ИИС на базе интеграций САПР и СУД.

Таким образом, **целью настоящей работы** является создание новых и развитие существующих теоретических и практических положений методологии информационной поддержки жизненного цикла изделий, обеспечивающих повышение эффективности процессов проектирования высокотехнологичной продукции путем сокращения длительности процесса проектирования за счет автоматизированного управления разнотипными взаимозависимыми объектами в интегрированной информационной среде.

На основании выполненного анализа жизненного цикла объектов проектирования и возможностей применения существующих информационных систем и технологий к особенностям процессов разработки высокотехнологичной продукции с длительным сроком активного существования, сделан вывод, что для достижения поставленной цели работы необходимо решить следующие **задачи**:

1. разработать информационные модели, описывающие статическую структуру связей разнотипных взаимозависимых объектов и являющихся основой для создания информационного, алгоритмического и программного обеспечения поддержки жизненного цикла изделий на этапе их проектирования;

2. сформулировать методические основы, позволяющие формализовать информационную модель, показывающую динамику изменения состояний жизненного цикла разнотипных объектов с учетом их версионности и взаимовлияния;

3. разработать метод построения интегрированной информационной среды на базе системы управления данными и ее интеграций с системами автоматизированного проектирования, позволяющий реализовать автоматизированное управление жизненным циклом разнотипных взаимозависимых объектов, представленных множеством версий, с учетом специфических особенностей процессов проектирования высокотехнологичной продукции;

4. выполнить программную реализацию разработанных моделей и алгоритмов автоматизированного управления жизненным циклом разнотипных объектов в рамках интегрированной информационной среды на базе системы управления данными и ее интеграций с системами автоматизированного проектирования;

5. выполнить практическую апробацию разработанных теоретических и практических положений в условиях действующего предприятия, специализирующегося в области проектирования и производства высокотехнологичной продукции с длительным сроком активного существования.

## **1.6 Резюме обзорных исследований**

В данной главе выполнен обзор применения современных информационных технологий поддержки жизненного цикла изделий к проектированию высокотехнологичной продукции с длительным сроком активного существования на примере отрасли космического приборостроения. Исследованы проблемы процессов проектирования данной прикладной области с точки зрения методологий управления продукцией и стандартов, регламентирующих ее жизненный цикл. Выполнен анализ методов управления связанными объектами. В результате проведенных обзорных исследований:

1. показано, что организация ИИС позволит сократить временные и трудовые затраты на проектирование и производство рассматриваемой категории объектов с помощью однозначного, непротиворечивого структурированного хранения информации и автоматизированного управления данными;

2. обосновано, что исследуемая предметная область характеризуется рядом специфических особенностей, которые накладывают ограничение на применение типовых программных решений по управлению ЖЦ изделий;

3. сделан вывод, что изменения стадий ЖЦ объектов проектирования соответствуют методологии проектирования RGD и имеют итеративный взаимосвязанный характер, структуру которого необходимо формализовать;

4. показано, что применение формальных методов управления связанными объектами, такими как элементы теории множеств, графов, автоматов, а также объектно-ориентированный подход, позволяет разработать модели изменений ЖЦ объектов проектирования и алгоритмы управления ими;

5. обоснован выбор инструментария UML для графической формализации моделей и алгоритмов управления ЖЦ объектов проектирования в ИИС;

6. показана необходимость сочетания функциональных возможностей информационных систем электронного документооборота и управления данными в части применения методов сетевого планирования, теории распараллеливания и синхронизации операций для распределения потоков работ по согласованию документов и соответствующего изменения состояний жизненного цикла связанных с ними объектов проектирования другого типа;

7. на основании результатов проведенного исследования сформулированы цели и задачи настоящей работы.

Все приведенные выводы, полученные из аналитического обзора применения современных информационных технологий поддержки ЖЦ изделий к управлению процессами проектирования высокотехнологичной продукции с длительным сроком активного существования, будут заложены в основу практической реализации выполненных теоретических исследований. Практическая часть настоящей работы представляет собой алгоритмическое и программное обеспечение автоматизированного управления жизненным циклом объектов проектирования в рамках интегрированной информационной среды и будет подробно описана в следующих разделах настоящей работы.

## **2 РАЗРАБОТКА МЕТОДОВ ПОЛУЧЕНИЯ МОДЕЛЕЙ ЖИЗНЕННОГО ЦИКЛА ОБЪЕКТОВ ПРОЕКТИРОВАНИЯ**

В данной главе выполнена формализация понятий рассматриваемой предметной области, описана структура взаимосвязей объектов проектирования и динамика изменения стадий их жизненного цикла. Предложены семантико-математические, структурные и формально-логические модели, которые иллюстрируют состав объектов проектирования. Описана динамика ЖЦ объектов проектирования. Формализованы зависимости жизненного цикла объектов проектирования друг от друга. Разработаны алгоритмы автоматизированного управления ЖЦ объектов проектирования в ИИС.

Разработанные модели и алгоритмы предназначены для использования в качестве основы программного обеспечения интегрированной информационной среды автоматизированного управления жизненным циклом проектирования объектов.

### **2.1 Метод формализации структуры взаимосвязей объектов проектирования в статическом состоянии**

Метод формализации структуры взаимосвязей объектов проектирования должен обеспечивать возможность использовать модели, полученные в результате его применения, в качестве алгоритмической основы для разработки программного обеспечения средств управления ЖЦ объектов проектирования в ИИС.

Разрабатываемый метод должен включать математические положения и возможность использования полученных математических выражений в качестве основы для описания предметной области с применением существующих нотаций моделирования. Кроме того, согласно цели настоящей работы, необходимо установить соответствие положений разрабатываемого метода и подходов к разработке программного обеспечения.

Таким образом, метод формализации структуры взаимосвязей разнотипных объектов проектирования должен включать следующие составные части:

- математическая основа;
- концептуальное проектирование структуры программного обеспечения с точки зрения объектно-ориентированной типизации данных;
- формальные модели объектно-ориентированной типизации данных в рамках структуры разрабатываемого программного обеспечения.

### 2.1.1 Разработка математических положений

Согласно положениям, описанным в [7, 45-63, 70-75], СУД выполняет роль инструмента, реализующего автоматизированное управление жизненным циклом изделий. Поэтому возникает задача концептуального проектирования базы данных СУД. Концептуальное проектирование предполагает структуризацию данных в соответствии с базовыми положениями объектно-ориентированного подхода [97-101]. В свою очередь, в основе объектно-ориентированного подхода лежит теория систем и теория множеств [114]. Таким образом, для реализации цели настоящего исследования, связанной с разработкой моделей и алгоритмов управления высокотехнологичной продукцией на этапе ее разработки необходимо формализовать отношения между объектами проектирования с применением теории множеств.

С использованием аппарата теории множеств, основные положения которой изложены в работах Б. Больцана, Б. Рассела, Э. Цермело и др., формализованы исследуемые объекты проектирования (изделие, ИМИ, КД, ЭСИ) и связанные с ними сущности.

Пусть  $OP$  – множество объектов проектирования,  $PRJ$  – множество проектов,  $PROD$  – множество изделий,  $ITM$  – множество элементов ЭСИ,  $DM$  – множество ИМИ,  $KD$  – множество конструкторских документов.

Для запуска изделия в производство необходим набор конструкторских документов, называемый комплект КД [30]. В соответствии с [30, 33] в настоящее время к таким документам также относят ЭМИ и ИМИ. Таким образом, ИМИ и КД фактически являются документами, поэтому следует определить множество документов  $DOC = \{DM, KD\}$ .

Рассматривая высокотехнологичную продукцию на примере отрасли космического приборостроения, следует детализировать множество элементов ЭСИ. В [22] устанавливаются следующие виды изделий: детали, сборочные единицы, комплексы и комплекты. Между этими видами изделий существуют следующие отношения:

- в состав сборочной единицы могут входить детали, комплекты и другие сборочные единицы;
- в состав комплекта могут входить детали, сборочные единицы и другие комплекты;
- в комплекс могут входить детали, сборочные единицы, комплекты, а также другие комплексы.

Кроме того, согласно выделению разделов спецификации по [120], изделие и его составные части также принято делить на:

- стандартные изделия (изделия, примененные по межгосударственным, государственным и отраслевым стандартам, а также стандартам предприятий);
- прочие изделия (изделия, примененные по техническим условиям, и импортные покупные изделия, примененные по сопроводительной технической документации зарубежных изготовителей/поставщиков);
- материалы (все материалы, непосредственно входящие в специфицируемое изделие, иногда их называют «изделия из материалов»).

В рамках данной классификации ЭРИ, которые являются основой РЭА, относятся к прочим изделиям. Таким образом, множество ЭСИ имеет неоднородный характер, поэтому можно записать

$$ITM = \{ASM, PART, CT, CX, STPT, OTPT, MAT\},$$

где  $ASM$  означает множество сборочных единиц,  $PART$  – множество деталей,  $CT$  – множество комплектов;  $CX$  – множество комплексов,  $STPT$  – множество стандартных изделий,  $OTPT$  – множество прочих изделий,  $MAT$  – множество изделий из материала.

Совокупность элементов ЭСИ образуют изделие: одному элементу множества изделий соответствует несколько элементов множества ЭСИ. Поэтому можно записать, что изделие  $A$  является сюръективной функцией относящихся к нему элементов множества ЭСИ:

$$f_A : ITM_A \rightarrow \{prod_A\}, \quad (1)$$

где  $prod_A \in PROD$  означает изделие  $A$ , конкретный элемент множества изделий  $PROD = \{prod_A, prod_A, \dots, prod_x\}$ ,  $prod_A \subseteq PROD$ ,  $ITM_A$  – множество элементов электронной структуры изделия  $A$ ,  $ITM_A \subseteq ITM$ ,  $ITM_A = \{itm_{i,A}\}$ ,  $itm_{i,A} \in ITM_A$  – конкретный элемент множества  $ITM_A$ ,  $i = \overline{1, n}$ ,  $n$  – количество элементов электронной структуры изделия  $A$ .

Используемый в настоящей работе термин «объекты проектирования» означает изделие, элементы его электронной структуры и описывающая их информация в виде файлов информационных моделей и конструкторских документов. Поэтому можно записать

$$OP = \{PROD, ITM, DM, KD\}. \quad (2)$$

В рамках одного проекта  $prj \in PRJ$  может создаваться одно или несколько изделий, каждое из которых описывается файлами информационных моделей  $dm \in DM$  и конструкторских документов  $kd \in KD$ . Учитывая (1) и (2), можно записать выражение, описывающее частный случай множества объектов проектирования изделия  $A$  в рамках проекта  $Z$ :

$$OP_{Z,A} = \{prj_Z, prod_{Z,A}, \{itm_{i,A}\}, \{dm_{j,A}\}, \{kd_{k,A}\}\},$$

где  $n$  – количество элементов электронной структуры изделия  $A$ ,  $i = \overline{1, n}$ ,  $m$  – количество информационных моделей изделия  $A$ ,  $j = \overline{1, m}$ ,  $p$  – количество конструкторских документов по изделию  $A$ ,  $k = \overline{1, p}$ ,  $prj_z \in PRJ$  – проект  $Z$ ,  $prod_{z,A} \in PROD$  – разрабатываемое в рамках проекта  $Z$  изделие  $A$ .

Первоисточником данных для одного элемента ЭСИ является один или несколько объектов ИМИ, поэтому можно записать, что множество элементов ЭСИ является сюръективной функцией множества ИМИ:

$$g_A : DM_A \rightarrow \{itm_A\}. \quad (3)$$

В свою очередь, описание состава и иерархии элементов ЭСИ представляет собой КД. Таким образом, множество КД является сюръективной функцией множества ИТМ:

$$y_A : ITM_A \rightarrow \{kd_A\}. \quad (4)$$

где  $\{kd_A\} \subseteq KD$  означает конструкторский документ на изделие  $A$ .

Выражения (3) и (4) показывают зависимость объектов проектирования одного типа от объектов другого типа. Специфика разработки высокотехнологичной продукции с длительным сроком активного существования, рассматриваемая на примере космического приборостроения, определяет неоднородный состав множества информационных моделей изделия, разработанных в различных САПР. Оно включает подмножества информационных моделей, разработанных в различных САПР: при электрическом и при механическом проектировании.

Таким образом, можно записать

$$DM = \{MCADDM, ECADDM\},$$

где  $MCADDM$  – множество файлов ИМИ, разработанных в САПР механического моделирования (MCAD), а  $ECADDM$ , соответственно, в САПР электрического моделирования (ECAD).

По технологии проектирования РЭА, этап электрического проектирования, на котором решаются задачи схмотехнического и радиотехнического характера предшествует этапу механического моделирования, на котором рассматриваются конструктивные и теплотехнические вопросы. Но не все составные части изделия подлежат электрическому моделированию: некоторые детали создаются только на этапе механического конструирования, как, например, рамка для закрепления электронных схем. В свою очередь, все электронные элементы ЭСИ, помимо своих электрических характеристик, моделирование которых осуществляется с помощью ECAD-САПР, имеют конструктивное воплощение (корпус), пространственные параметры которого являются объектом моделирования в MCAD-САПР.

Таким образом, в рамках разработки одного изделия  $A$  количество ИМИ, созданных в MCAD-САПР превышает количество ИМИ, создан-

ных в ЕСAD-САПР. Поскольку множества  $MCADDM_A \subseteq MCADDM$  и  $ECADDM_A \subseteq ECADDM$  являются конечными, можно записать выражение, показывающее сравнение их мощностей:

$$|MCADDM_A| > |ECADDM_A|. \quad (5)$$

Учитывая (5), можно сделать вывод, что полный состав изделия определяется его информационными моделями, разработанными в MCAD-САПР. Поэтому множество элементов ЭСИ изделия  $A$ ,  $ITM_A \subseteq ITM$ , является сюръективной функцией множества информационных моделей изделия  $A$ , разработанных в MCAD-САПР

$$w_A : MCADDM_A \rightarrow ITM_A. \quad (6)$$

Практическая реализация выражений (1)-(6) будет положена в основу программного обеспечения средств автоматизированного управления жизненным циклом объектов проектирования в рамках ИИС, включая проектирование базы данных СУД и разработку требований к программной интеграции САПР и СУД.

### 2.1.2 Концептуальное проектирование базы данных СУД

Зависимости между разнотипными объектами проектирования, формализованные в виде выражений (1)-(6) позволяют выделить типы рассматриваемых сущностей и структурировать их в виде классов в соответствии с положениями объектно-ориентированного подхода, описанного в работах [87-94]. Таким образом, классы являются контейнерами информации о рассматриваемых типовых объектах. Анализируя выражение (2) с точки зрения системного подхода, можно сделать вывод, что для реализации автоматизированного управления жизненным циклом объектов проектирования в рамках ИИС в модели данных СУД необходимо предусмотреть следующие классы: «Проекты», «Изделия», «элемент ЭСИ», «Документы». Учитывая свойство сюръективности функций, показанных в выражениях (1), (3), (4), (6), можно сделать вывод о наличии связи «один-ко-многим» при проектировании базы данных. Таким образом, между классами «Проекты», «Изделия», «элемент ЭСИ», «Документы» существуют следующие отношения:

- в рамках одного проекта может выполняться разработка одного или более изделий;
- для одного изделия может существовать множество вариантов его электронной структуры;
- к одному проекту может относиться множество документов;
- ЭСИ и ее составные части описывается одним или более документом;
- к одному изделию может относиться один или более документ.

В связи с неоднородным составом множества ЭСИ, класс «элемент ЭСИ» является абстрактным и детализируется на следующие классы, согласно выделению составных частей изделия по [22]: «Стандартное изделие», «Комплект», «Деталь», «Сборочная единица», «Прочее изделие», «Изделие из материала». При этом между данными классами определены следующие отношения:

- к одному экземпляру класса «Комплект» может относиться множество экземпляров классов «Деталь», «Сборочная единица», «Стандартное изделие», «Комплект»;
- к одному экземпляру класса «Сборочная единица» может относиться множество экземпляров классов «Деталь», «Сборочная единица», «Прочее изделие», «Изделие из материала», «Стандартное изделие».

Выражение (5), а также особенности приборного проектирования с электрической и механической точек зрения позволяют сделать вывод о необходимости объявления классов, объекты которых будут использоваться для хранения соответствующих информационных моделей. Кроме того, одной из особенностей механического проектирования является компонентный состав модели и понятие сборки. Сборка представляет собой структурированную совокупность отдельных конструктивных элементов, с учетом их габаритных размеров, положений в пространстве и ограничений подвижности. Фактически, сборка представляет собой файл, в котором содержатся ссылки на файлы с описанием отдельных элементов, а также вышеперечисленные параметры зависимостей между ними. Таким образом, сборка состоит из множества подборок и деталей.

Аналогично организован компонентный состав модели изделия с точки зрения электрического проектирования: проект модели представляет собой набор ссылок на структурированную совокупность электрических схем, печатных плат и программного кода. Следовательно, применительно к рассматриваемой специфике проектирования высокотехнологичных изделий на примере отрасли космического приборостроения, потомками абстрактного класса «ИМИ» являются абстрактные классы «MCAD» и «ECAD». Они, в свою очередь, детализируются на следующие классы:

- «MCAD-Сборка» – для хранения сборки, с которой ассоциируется множество экземпляров классов сборки «MCAD\_Сборка» и деталей «MCAD\_Деталь»;
- «ECAD-Проект» – для хранения проекта, с которым ассоциируется множество экземпляров класса входящих в него файлов электрических схем, печатных плат, программного кода «ECAD\_Файл».

Различные типы документов (ИМИ, КД) позволяют сделать вывод, о необходимости объявить класс «Документы» абстрактным суперклассом, потомками которого будут классы «КД» и «ИМИ».

На рисунке 2.1. изображена UML-диаграмма классов, которая иерархию и взаимосвязь вышеописанных сущностей проектирования. На данном (начальном) этапе концептуального проектирования модели данных СУД перечень операций и атрибутов классов пока не определен и будет представлен далее.

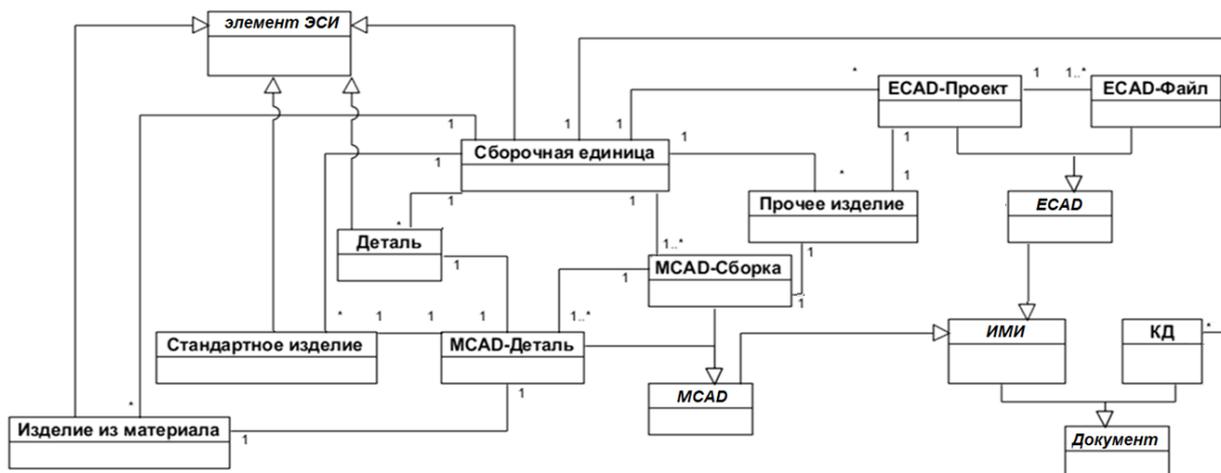


Рис. 2.1. Иерархия и взаимосвязи типовых объектов проектирования

## 2.2 Метод формализации динамики изменения жизненного цикла объектов проектирования

Метод формализации динамики изменения стадий жизненного цикла разнотипных взаимозависимых объектов должен обеспечивать возможность использовать методики, модели и алгоритмы, полученные в результате его применения, в качестве алгоритмической основы для разработки программного обеспечения средств управления процессом проектирования высокотехнологичной продукции в ИИС.

Разрабатываемый метод должен включать следующие составные части:

- методика представления динамики изменений стадий жизненного цикла разнотипных взаимозависимых объектов в виде раскрашенного графа;
- формальная модель в виде раскрашенного графа, которая показывает динамику изменений стадий ЖЦ объектов проектирования в ИИС и является основой получения алгоритмического базиса для разработки соответствующего программного обеспечения;
- алгоритмы изменения стадий ЖЦ объектов проектирования в ИИС.

### 2.2.1 Методика формализации жизненного цикла объектов проектирования

Поскольку теория графов представляет собой мощный инструмент для описания многокомпонентных систем, целесообразно использовать ее положения для формализации жизненного цикла объектов проектирования. Однако, построение модели изменений ЖЦ объектов исследования настоящей работы представляет собой отдельную нетривиальную задачу. Поскольку процесс разработки модели включает структурированную последовательность действий, следует определить ее в виде методики.

Предлагаемая методика формализации ЖЦ разнотипных взаимозависимых объектов с применением теории графов состоит из следующих шагов:

1. Составить перечень типов разнотипных взаимозависимых объектов.
2. Для каждого типового объекта определить последовательность изменений стадий его ЖЦ (от исходного состояния к целевому).
3. Для каждого типового объекта определить операции перехода между стадиями ЖЦ.
4. Каждую пару значений «типовой объект проектирования, его стадия ЖЦ» представить в виде состояния конечного автомата Мили. Получается конечное множество состояний автомата  $Q$ , элементы которого описаны следующим образом:

$$q_{i+1} = f(q_i, x_i, q_j, x_j), \quad (7)$$

где  $q_i \in Q$  – состояние автомата Мили в момент  $i$ , которое означает текущую стадию ЖЦ рассматриваемого объекта;  $x_i \in X$  – сигнал, поданный на вход состояния  $q_i$ , описывающий операцию изменения стадий ЖЦ объекта;  $f(q_i, x_i)$  – отображение функции перехода между состояниями, показывающее зависимость изменения стадии ЖЦ рассматриваемого объекта от его текущего состояния, поданного сигнала и состояния связанного с ним объекта  $q_j \in Q$ , на вход которого подан сигнал  $x_j \in X$ . При отсутствии зависимостей изменения состояний от стадий ЖЦ объектов разных типов, состояние автомата описывается более упрощенно:

$$q_{i+1} = f(q_i, x_i). \quad (8)$$

5. С целью обобщенного представления выражений (7)-(8), показывающих изменение стадий ЖЦ исследуемых разнотипных взаимозависимых объектов, следует формализовать функционирование рассматриваемого автомата Мили в виде совмещенной матрицы переходов/выходов.
6. Для повышения наглядности, вершины графа с одинаковым типом объекта проектирования раскрасить одним цветом, при этом не

ставится задача правильной раскраски графа согласно принятому определению [117, 117].

7. В соответствии с логической последовательностью процесса проектирования выбрать начальную и конечную вершины графа.

8. Связать вершины графа в направлении логической последовательности изменения стадий ЖЦ типовых объектов от начальной вершины к конечной, учитывая зависимости, описанные в виде (7), (8), а также ранее составленную совмещенную матрицу переходов/выходов состояний рассматриваемого автомата Мили.

Далее необходимо проанализировать полученный оргграф, выполнив следующие действия:

1. Проверить полученный оргграф на наличие контуров и висячих вершин. Движение по контурам означает повтор операций по изменению стадий ЖЦ объектов проектирования. Вершины внутри контура показывают возвратные состояния объектов. Висячие вершины представляют собой конечные, т.е. целевые состояния объектов.

2. С использованием существующих методов (например, алгоритм Дейкстры, Флойда, Форда-Беллмана, Джонсона) [125] найти кратчайший путь от начальной вершины к конечной, исключив висячие вершины (кроме начальной и конечной), петли и контуры. Для этого граф должен быть взвешенным, для чего следует поставить в соответствие его дугам некоторое число, называемое вес.

3. Вершины полученного кратчайшего пути показывают невозвратные стадии ЖЦ объектов проектирования при движении от исходного состояния к целевому. В случае проектирования высокотехнологичных изделий с длительным сроком активного существования полученный кратчайший путь представляет оптимальный ход выполнения процесса разработки с точки зрения сокращения временных (и, соответственно, материальных) затрат. Это необходимо принять во внимание на этапе организационного планирования процесса проектирования и разработке его методического и информационного обеспечения.

4. Для формирования алгоритмической основы программного обеспечения управления ЖЦ объектов проектирования разработать формальные графические схемы (UML-диаграммы), описывающие последовательность изменений объектов и действий пользователей ИИС. Исходные данные для разработки UML- диаграмм состояний, последовательностей и классов в части операций над объектами различных типов содержатся в ранее составленных выражениях типа (7)-(8), а также в совмещенной матрице переходов/выходов состояний рассматриваемого автомата Мили.

На рисунке 2.2 представлена графическая схема алгоритма действий разработанной методики формализации динамики изменений стадий жизненного цикла разнотипных взаимозависимых объектов в ИИС.



Рис. 2.2 Алгоритм действий разработанной методики формализации динамики изменений стадий жизненного цикла разнотипных взаимозависимых объектов

Работоспособность предложенной методики проверена на примере формализации модели динамики изменений ЖЦ объектов проектирования в отрасли космического приборостроения.

### **2.2.2 Модель динамики изменения стадий жизненного цикла объектов проектирования в ИИС**

В соответствии с разработанной методикой следует, прежде всего, составить перечень типов объектов проектирования. Для этого целесообразно воспользоваться выражениями (1)-(8) и UML-диаграммами (рис. 2.1), разработанными на этапе концептуального проектирования базы данных СУД. Итак, в перечень типов объектов проектирования входят следующие сущности: «Изделие», «ИМИ», «ЭСИ», «КД».

Рассмотрим логическую последовательность проектирования в ИИС:

1. До начала проектирования изделие находится в стадии ЖЦ «Новый».
2. Как только в рамках одной стадии испытаний запущен процесс проектирования, изделие переходит в стадию ЖЦ «В разработке».
3. Запуск процесса проектирования изделия означает начало разработки его информационной модели, которая в начале своего жизненного цикла находится в стадии «Новый» (файл ИМИ, разработанный в САПР, находится на локальном компьютере разработчика).
4. Для предоставления общего доступа к файлу ИМИ его следует опубликовать в СУД, при этом стадия жизненного цикла этого объекта меняется на «Опубликован».
5. Для изменения ИМИ объект переходит в стадию ЖЦ «На редактировании» и становится недоступным от изменений.
6. В связи с итеративностью процесса проектирования и необходимостью накопления истории в СУД сохраняется каждая новая версия файла ИМИ, разработанного или отредактированного в САПР, при этом стадия жизненного цикла ИМИ меняется на «Новая версия».
7. Чтобы объявить окончание разработки ИМИ, последнюю версию этого объекта следует перевести в стадию ЖЦ «Выпущен».
8. При отсутствии замечаний ИМИ переходит в стадию ЖЦ «Согласован».
9. При наличии в СУД документов ИМИ, которые характеризуют изделие с конкретной точки зрения (например, электрическое или механическое проектирование) и находятся в стадии ЖЦ

- «Согласован», следует на их основе программным образом сформировать элементы ЭСИ в стадии ЖЦ «Новая версия».
10. На основе совокупности элементов ЭСИ возможна автоматическое формирование документов КД в стадии ЖЦ «Новый», при этом созданные объекты находятся на локальном компьютере разработчика.
  11. Для предоставления общего доступа к сформированной конструкторской документации ее следует опубликовать в СУД (стадия жизненного цикла КД меняется на «Опубликован»).
  12. Поскольку КД является визуальным представлением специализированных данных ИМИ, именно этот объект, наряду с информационной моделью, предоставляется для согласования уполномоченным лицам. При наличии замечаний в ИМИ (или КД, полученной на ее основе), необходимо изменить первоисточник, т.е. ИМИ. При этом происходит возврат на этап разработки в САПР новой версии ИМИ и формирование новой версии КД на ее основе.
  13. Чтобы объявить окончание разработки ИМИ и КД, следует программным образом изменить стадию жизненного цикла КД на «Выпущен».
  14. После окончательного выпуска КД рационально отметить итоговый вариант ЭСИ, изменив программным образом стадию ЖЦ ее последней версии на «Выпущен».
  15. После электронной подписи согласованных ИМИ, на основе которых была выпущена КД, ИМИ переходит в стадию ЖЦ «Оригинал», поскольку является первичным документом для генерации КД.
  16. Наличие ИМИ в стадии ЖЦ «Оригинал» разрешает перевод КД в стадию ЖЦ «Согласован».
  17. После электронной подписи согласованных конструкторских документов комплект КД переходит в стадию ЖЦ «Подлинник».
  18. При наличии подлинников КД процесс проектирования в рамках одной стадии испытаний считается выполненным, и изделие переходит в стадию ЖЦ «Готово к производству».

Согласно разработанной методике, представленной в пункте 2.2.1 настоящей работы, формализуем для каждого типового объекта последовательность стадий его ЖЦ. В таблице 2.1 показаны исследуемые типовые объекты (Изделие, ИМИ, ЭСИ, КД) и стадии их ЖЦ.

Таблица 2.1

**Стадии жизненного цикла объектов проектирования**

№	Тип объекта	Стадия ЖЦ
1	Изделие	Новый
2		В разработке
3		Готово к производству
4	Информационная модель изделия (ИМИ)	Новый
5		Опубликован
6		На редактировании
7		Новая версия
8		Выпущен
9		Согласован
10		Оригинал (подписан ЭП)
11	Электронная структура изделия (ЭСИ)	Новая версия
12		Выпущен
13	Конструкторская документация (КД)	Новый
14		Опубликован
15		Новая версия
16		Выпущен
17		Согласован
18		Подлинник (подписан ЭП)

Далее определим набор операций по изменению стадий ЖЦ, т.е. фактически конечное множество входных сигналов автомата Мили (табл. 2.2).

Таблица 2.2

**Операции перехода в стадии жизненного цикла**

№	Стадия ЖЦ	Операция перехода в эту стадию (входной сигнал)	
		Название	Обозначение
1	Новый	Создать	$x_1$
2	Опубликован	Опубликовать	$x_2$
3	На редактировании	Изменить ИМИ	$x_3$
4	Новая версия	Создать версию	$x_4$
5	Выпущен	Выпустить	$x_5$
6	Согласован	Согласовать	$x_6$
7	Оригинал / Подлинник (подписан ЭП)	Подписать ЭП	$x_7$
8	В разработке / Готово к производству	Перевести в следующее состояние	$x_8$

Далее, для каждого объект представим его стадии ЖЦ в виде состояния конечного автомата Мили, описав при этом функцию перехода в состояние согласно выражениям (7)-(8). Подобное формальное описание зависимостей перехода в стадии ЖЦ объектов представлено в таблице 2.3.

## Формальное описание зависимостей перехода в стадии ЖЦ

№	Тип объекта	Стадия ЖЦ	Состояние	Функция перехода в состояние
1	Изделие	Новый	$q_1$	
2		В разработке	$q_2$	$f(q_1, x_1, q_4, x_8)$
3		Готово к производству	$q_3$	$f(q_{18}, x_7, q_2, x_8)$
4	Информационная модель изделия (ИМИ)	Новый	$q_4$	$f(q_1, x_1)$
5		Опубликован	$q_5$	$f(q_4, x_2)$
6		На редактировании	$q_6$	$f(q_5, x_3) = f(q_7, x_3) = f(q_8, x_3) = f(q_{14}, x_3) = f(q_{16}, x_3)$
7		Новая версия	$q_7$	$f(q_5, x_4) = f(q_6, x_4)$
8		Выпущен	$q_8$	$f(q_5, x_5) = f(q_7, x_5)$
9		Согласован	$q_9$	$f(q_8, x_6)$
10		Оригинал (подписан ЭП)	$q_{10}$	$f(q_9, x_7, q_{17}, x_7)$
11	Электронная структура изделия (ЭСИ)	Новая версия	$q_{11}$	$f(q_9, x_1)$
12		Выпущен	$q_{12}$	$f(q_{16}, x_5, q_{11}, x_5)$
13	Конструкторская документация (КД)	Новый	$q_{13}$	$f(q_{11}, x_1)$
14		Опубликован	$q_{14}$	$f(q_{13}, x_2)$
15		Новая версия	$q_{15}$	$f(q_{14}, x_4)$
16		Выпущен	$q_{16}$	$f(q_{14}, x_5) = f(q_{15}, x_5)$
17		Согласован	$q_{17}$	$f(q_{16}, x_6)$
18		Подлинник (подписан ЭП)	$q_{18}$	$f(q_{10}, x_7, q_{17}, x_7)$

Для отображения динамики изменения ЖЦ исследуемых разнотипных объектов построим в виде автомата Мили построим совмещенную матрицу переходов/выходов (табл. 2.4), представляющую собой таблицу, строки которой показывают текущее состояние автомата, а столбцы – состояние перехода. Отношение вида  $a/(b+c)$  на пересечении строки  $i$  и столбца  $j$  означает, что переход из состояния  $q_i$  в состояние  $q_j$  выполняется под воздействием сигнала  $a$ , выходящего из состояния  $q_i$ . При этом выходами состояния  $q_j$  являются сигналы  $b$  и  $c$  (рис. 2.3). Пустое значение в ячейке таблицы на пересечении строки  $i$  и столбца  $j$  показывает отсутствие непосредственного перехода из состояния  $q_i$  в состояние  $q_j$ .

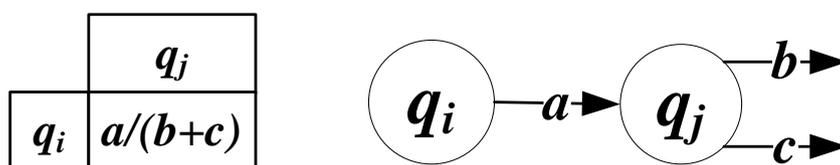


Рис. 2.3 – Принцип составления совмещенной матрицы выходов/переходов автомата Мили

В настоящей работе, в отличие от традиционной модели автомата Мили, предлагается не вводить дополнительное множество выходных сигналов состояний, а оперировать только с одним множеством опера-

ций перехода между состояниями. В табл. 2.4 показана совмещенная матрица переходов/выходов автомата Мили, составленная для рассматриваемого случая описания динамики ЖЦ объектов проектирования на примере отрасли космического приборостроения.

Таблица 2.4

## Совмещенная матрица переходов/выходов автомата Мили

	$q_1$	$q_2$	$q_3$	$q_4$	$q_5$	$q_6$	$q_7$	$q_8$	$q_9$	$q_{10}$	$q_{11}$	$q_{12}$	$q_{13}$	$q_{14}$	$q_{15}$	$q_{16}$	$q_{17}$	$q_{18}$
$q_1$		$x_8$		$x_1 / (x_2 + x_8)$														
$q_2$			$x_8$															
$q_3$																		
$q_4$					$x_2 / (x_3 + x_4 + x_5)$													
$q_5$						$x_2 / x_3$	$x_2 / x_4$	$x_2 / x_5$										
$q_6$							$x_3 / x_4$											
$q_7$						$x_4 / x_3$		$x_4 / x_5$										
$q_8$						$x_5 / x_3$			$x_5 / x_6$									
$q_9$										$x_6 / x_1$								
$q_{10}$												$x_1 / x_1$						
$q_{11}$																		
$q_{12}$													$x_2 / (x_3 + x_4 + x_5)$					
$q_{13}$																		
$q_{14}$						$x_2 / x_3$									$x_2 / x_4$	$x_2 / x_5$		
$q_{15}$															$x_4 / x_5$			
$q_{16}$												$x_5$					$x_7 / x_7$	
$q_{17}$										$x_7 / x_7$								$x_7 / x_7$
$q_{18}$			$x_7 / x_8$															

Анализ таблиц 2.1, 2.3-2.4 позволяет сделать вывод о количестве вершин графа. С целью повышения его наглядности, вершины графа с одинаковым типом объекта проектирования следует раскрасить одним цветом. На рисунке 2.4 представлена цветовая дифференциация вершин графа по рассматриваемым типам объектов проектирования (подчеркнем, что здесь не имеется ввиду правильная раскраска согласно принятому определению [117, 117]). Для повышения наглядности принята цветовая схема раскраски вершин графа по типам объектов проектирования, показанная в таблице 2.5 и на рисунке 2.4.

Таблица 2.5

### Цветовая дифференциация вершин графа по типам объектов

Тип объекта	Цвет соответствующей вершины графа
Изделие	Желтый
Информационная модель изделия (ИМИ)	Зеленый
Электронная структура изделия (ЭСИ)	Оранжевый
Конструкторская документация (КД)	Голубой

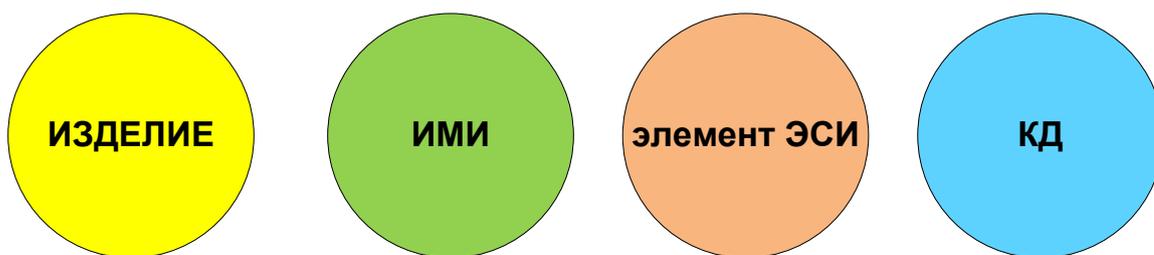


Рис. 2.4. Цветовая дифференциация вершин графа по типам объектов

В соответствии с вышеизложенной логической последовательностью разработки изделия, за начальную точку графа следует принять вершину с парой значений «Изделие, Новое», а за конечную – «Изделие, Готово к производству». Далее, связав вершины графа в направлении движения от начальной вершины к конечной, согласно таблице 2.4, получаем модель изменений стадий ЖЦ объектов проектирования в ИИС в виде раскрашенного орграфа (рис. 2.5).

Для подробного анализа полученного орграфа составим его матрицу смежности, показанную в таблице 2.6. С целью компактного представления вершин орграфа, использовано их обозначение в виде состояний автомата Мили, указанного в таблицах 2.3-2.4.

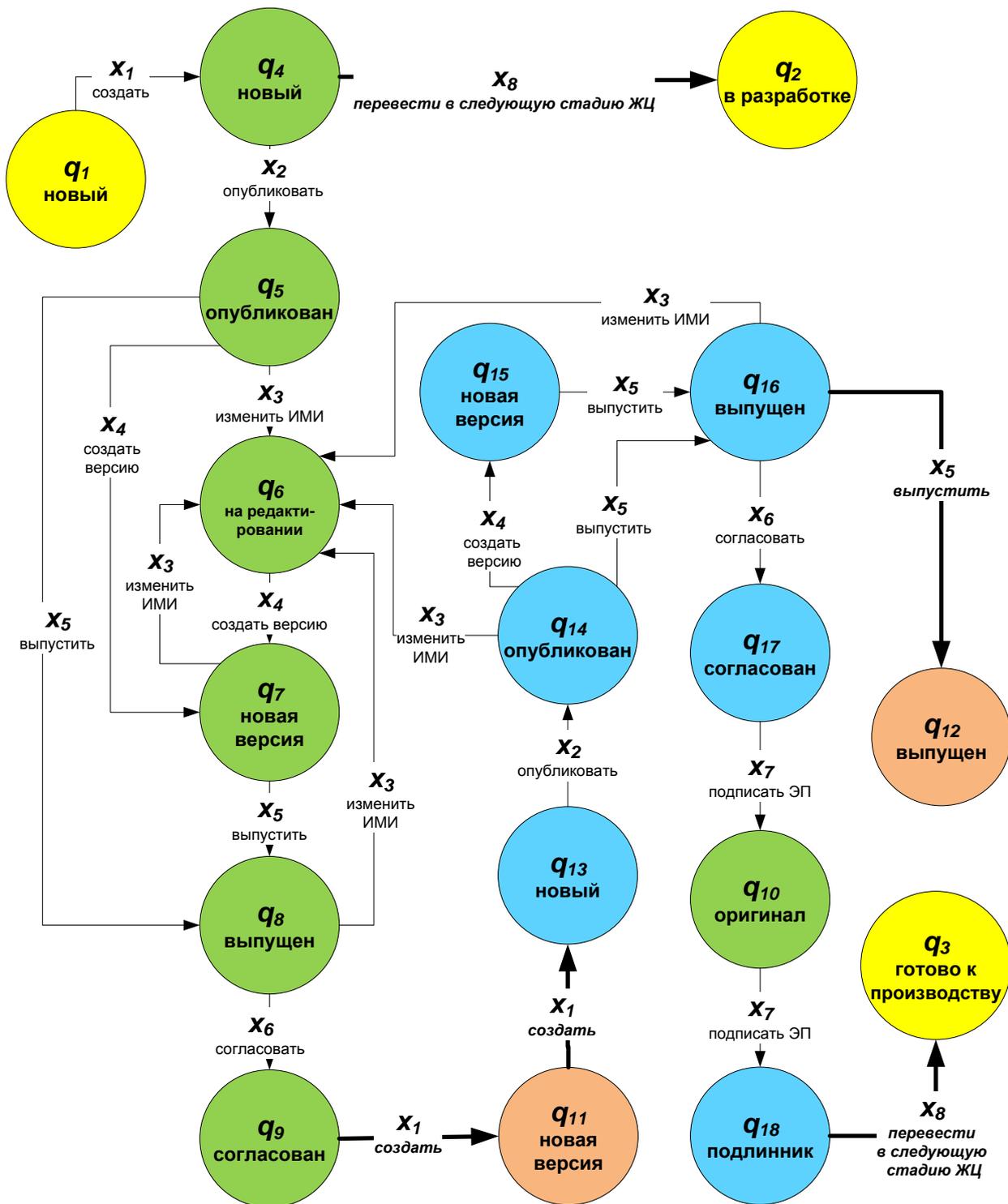


Рис.2.5. Модель динамики изменений стадий жизненного цикла объектов проектирования в ИИС в виде раскрашенного орграфа

Анализируя полученный орграф (рис. 2.5), в нем можно выделить следующие замкнутые контуры:

- «ИМИ новая версия» - «ИМИ на редактировании» - «ИМИ новая версия»;
- «ИМИ выпущен» - «ИМИ на редактировании»- «ИМИ новая версия»;

- «ИМИ новая версия» - «ИМИ выпущен» - «ИМИ новая версия»;
- «КД опубликован» - «ИМИ на редактировании»- «ИМИ новая версия» - «ИМИ выпущен» - «ИМИ согласован» - «ЭСИ новая версия» - «КД новый» - «КД опубликован»;
- «КД выпущен» - «ИМИ на редактировании»- «ИМИ новая версия» - «ИМИ выпущен» - «ИМИ согласован» - «ЭСИ новая версия» - «КД новый» - «КД опубликован» - «КД выпущен».

Движение по контурам орграфа означает формирование новой версии проектируемых сущностей.

Таблица 2.6

**Матрица смежности полученного раскрашенного орграфа**

	$q_1$	$q_2$	$q_3$	$q_4$	$q_5$	$q_6$	$q_7$	$q_8$	$q_9$	$q_{10}$	$q_{11}$	$q_{12}$	$q_{13}$	$q_{14}$	$q_{15}$	$q_{16}$	$q_{17}$	$q_{18}$
$q_1$				1														
$q_2$				-1														
$q_3$																		-1
$q_4$	-1	1			1													
$q_5$				-1		1	1	1										
$q_6$					-1		1	-1						-1		-1		
$q_7$					-1	1		1										
$q_8$					-1	1	-1		1									
$q_9$								-1			1							
$q_{10}$																	-1	1
$q_{11}$								-1					1					
$q_{12}$																-1		
$q_{13}$											-1			1				
$q_{14}$						1							-1		1	1		
$q_{15}$														-1		1		
$q_{16}$						1						1		-1	-1		1	
$q_{17}$										1						-1		
$q_{18}$			1							-1								

Анализ матрицы смежности (табл. 2.6), позволяет сделать следующие выводы:

Граф является связным, поскольку в его матрице смежности отсутствуют нулевые строки и столбцы.

Граф содержит висячие вершины, у которых в строке или столбце матрицы смежности существует только единственное значение, по модулю равное 1. В разработанном раскрашенном орграфе можно выделить следующие висячие вершины:

- $q_1$  – «Изделие новый» (начальная вершина орграфа);
- $q_2$  – «Изделие в разработке»;
- $q_{12}$  – «ЭСИ выпущен»;

–  $q_3$  – «Изделие готово к производству» (конечная вершина орграфа).

Перечисленные висячие вершины представляют собой целевые стадии ЖЦ объектов проектирования. Анализируя инцидентные им дуги, можно сделать вывод, что перевод объекта в целевое состояние следует осуществлять автоматически программным образом, а не с помощью операции, выполняемой пользователем.

В соответствии с ранее предложенной методикой, определение оптимального маршрута выполнения процесса разработки с точки зрения сокращения временных (и, соответственно, материальных) затрат сводится к задаче нахождения кратчайшего пути в орграфе. Для этого следует задать вес его дуг. В связи с однотипностью перехода между вершинами орграфа, которые обозначают операции изменения стадий ЖЦ объектов проектирования, было решено задать каждой дуге орграфа вес, равный 1. При наличии большого количества вершин в графе для нахождения кратчайшего пути целесообразно воспользоваться существующими методами и средствами решения данной задачи, которые отличаются наиболее простой реализацией, например, набор вычислений в математическом пакете MathCad с использованием алгоритмы Дейкстры [126]. При этом были исключены найденные контуры, висячие вершины (кроме конечной и начальной) и инцидентные им дуги.

Полученный кратчайший путь анализируемого орграфа (рис. 2.6) иллюстрирует линейный порядок создания и обработки объектов проектирования, который реализуется при разработке высокотехнологичной продукции в ИИС.

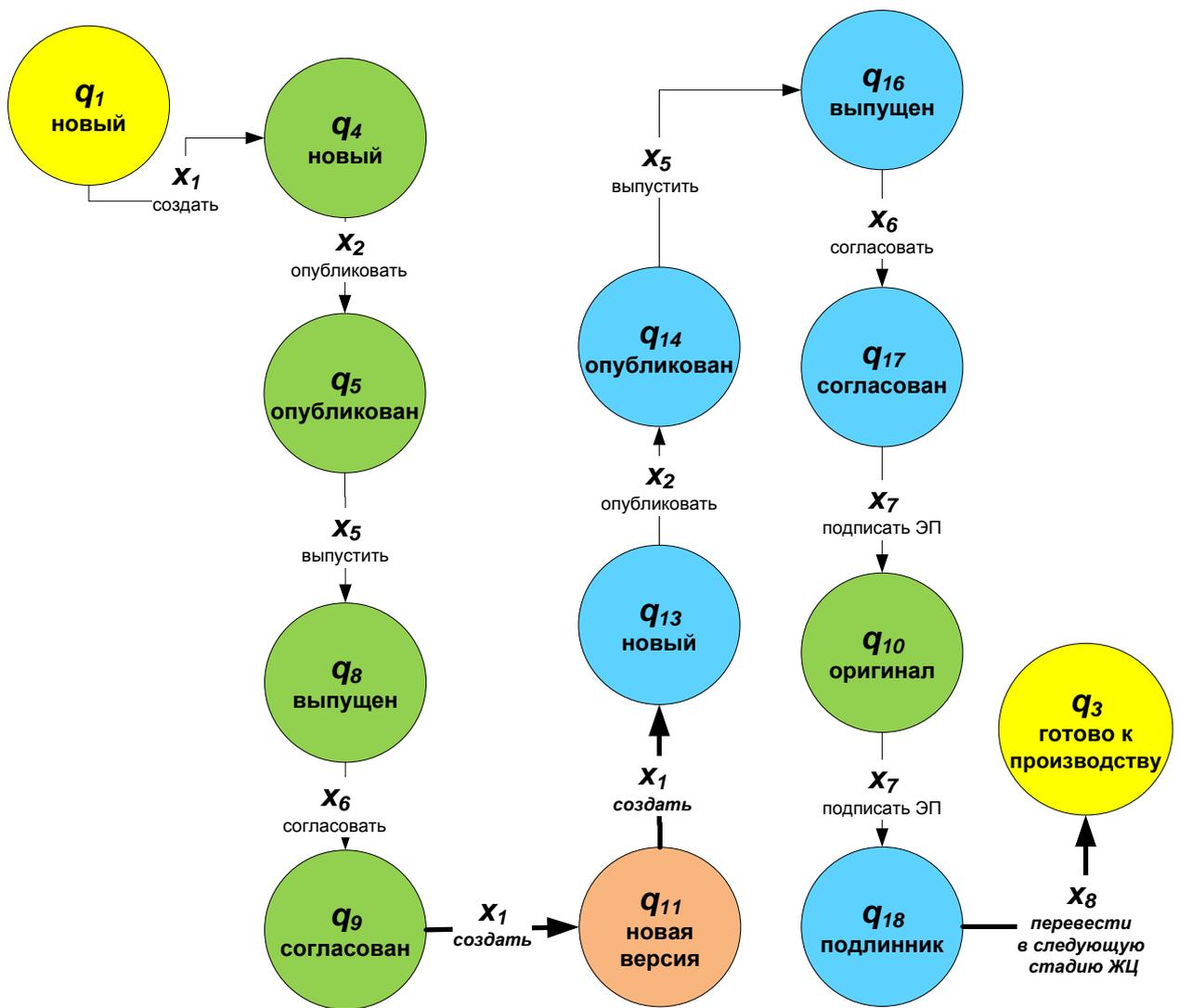


Рис.2.6. Линейный порядок создания и обработки объектов проектирования в ИИС

Анализ полученного кратчайшего пути орграфа позволяет сделать вывод, что переход от вершины одного цвета к другой практически всегда означает операцию, выполняемую автоматически программным образом, например «создать элемент ЭСИ», «создать КД», «перевести в следующую стадию ЖЦ». При этом переход между вершинами одного цвета, который означает последовательное изменение стадий ЖЦ объекта проектирования в рамках одного типа, выполняется, как правило, с помощью стандартной команды в интерфейсе пользователя СУД.

Анализ составленной модели орграфа, описывающей динамику изменения ЖЦ разнотипных объектов проектирования, позволяет сделать вывод, что объекты типов «ЭСИ» и «КД», являются зависимыми сущностями, поскольку состояние их жизненного цикла определяется жизненным циклом информационной модели изделия. С учетом данно-

го замечания и выражений (3)-(4), целесообразно записать следующие отображения:

$$q: \{s_{kd_A}\} \cap \{s_{ITM_A}\} \rightarrow \{s_{prod_A}\}, \quad (9)$$

где  $\{s_{prod_A}\} \subseteq S_{PROD}$  – это набор стадий ЖЦ изделия  $A$ , элемент множества возможных стадий ЖЦ изделий  $S_{PROD}$ ,  $\{s_{kd_A}\} \subseteq S_{KD}$  – стадии ЖЦ конструкторского документа на изделие  $A$ , подмножество возможных стадий ЖЦ конструкторской документации  $S_{KD}$ ,  $\{s_{ITM_A}\} \subseteq S_{ITM}$  – стадии ЖЦ множества элементов ЭСИ изделия  $A$ , подмножество возможных стадий ЖЦ элементов ЭСИ  $S_{ITM}$ ;

$$z: \{s_{DM_A}\} \rightarrow \{s_{kd_A}\}, \quad (10)$$

где  $\{s_{DM_A}\} \subseteq S_{DM}$  – набор стадий ЖЦ множества информационных моделей изделия  $A$ , подмножество возможных стадий ЖЦ документов ИМИ  $S_{DM}$ .

Разработанная модель динамики изменений ЖЦ объектов проектирования (рис. 2.5) в виде раскрашенного орграфа и автомата Мили позволяет определить основные операции, которые могут быть выполнены над объектами. На основании этого будет продолжено концептуальное проектирование объектно-реляционной модели данных СУД с помощью задания операций классов, описывающих анализируемые типы объектов.

### 2.2.3 Объектно-ориентированная модель сущностей проектирования

Анализ разработанной модели, показывающей динамику изменения состояний ЖЦ рассматриваемых типов объектов (рис. 2.5), позволяет сделать вывод, что объекты всех типов имеют атрибут «Стадия ЖЦ» и операции «Создать» и «Изменить стадию ЖЦ». Поскольку КД и ИМИ фактически являются документами, обобщение объектов этих типов представлено классом «Документ». Таким образом, согласно принципу подстановки Б. Лисков [90, 91], возможно обобщение классов «Изделие», «ЭСИ», «Документ» в виде потомков абстрактного класса «Объекты проектирования».

Поскольку сущность «Изделие» не включает составных частей, у класса «Изделие» не предусмотрены потомки. В свою очередь, составные части изделия реализуются подклассами абстрактного класса «элемент ЭСИ» («Стандартное изделие», «Деталь», «Сборочная единица»,

«Прочее изделие», «Изделие из материала»), которые характеризуют составные части изделия по [22].

В связи с тем, что в состав сборочной единицы могут входить разные сущности (детали, прочие и стандартные изделия, а также изделия из материала), необходима операция определения состава и иерархии компонентов сборочной единицы. Поэтому класс «Сборочная единица» поддерживает операцию «Сформировать структуру», а также «Сформировать КД», поскольку некоторые виды конструкторских документов (например, спецификация, перечень элементов), формируются на отдельные сборочные единицы [32, 30].

Согласно разработанной модели динамики изменения стадий ЖЦ объектов проектирования (рис. 2.5), ИМИ могут быть экспортированы в САПР. Для этого в суперклассе «ИМИ» и его потомках следует предусмотреть соответствующую операцию. Операции по изменению стадий ЖЦ объектов проектирования фактически представляют собой изменение некоторых характеристик рассматриваемых сущностей. Согласно [103, 104 **Ошибка! Источник ссылки не найден.**], эти характеристики принято называть атрибутами.

Как правило, типовые функциональные возможности современных СУД включают операции по изменению типового набора стадий ЖЦ объектов вида «Документ» и «элемент ЭСИ». Однако, на основании выполненного анализа наиболее популярных СУД (Enovia SmarTeam, Lotsia PLM, Windchill, TeamCenter, T-FLEX DOCs, ЛОЦМАН:PLM, 1С: PDM) [49, 71-75], был сделан вывод, что существующие решения не отражают необходимую специфику управления ЖЦ объектов проектирования, в том числе взаимозависимость рассматриваемых сущностей, характеристики подлинности документов, процессы согласования и утверждения. Поэтому необходимо определить набор атрибутов, которые будут в полной мере характеризовать ЖЦ рассматриваемых объектов проектирования.

При этом следует учесть возможность существования нескольких версий объекта, статус его согласования, электронной подписи и типовые этапы ЖЦ в СУД. Существование нескольких версий объектов подклассов «элемент ЭСИ» и «Документ» позволит реализовать накопление интеллектуального капитала за счет структурированного хранения результатов предыдущих проектных работ и возможности их повторного использования, в том числе с внесением необходимых изменений.

Таким образом, в полной мере состояние объекта описывает не только типовая стадия ЖЦ, определенная в качестве состояний в таблицах 2.1 и 2.3 в таблицах, но и другие характеристики. В таблице 2.7 показан предложенный набор атрибутов для определения состояния жизненного цикла разнотипных взаимозависимых объектов проектирования

в СУД Enovia SmartTeam для классов-контейнеров информации о рассматриваемых типовых объектах (Проект, Изделие, ЭСИ, КД, ИМИ).

Помимо анализируемых типов объектов проектирования (Изделие, ИМИ, ЭСИ и КД) в таблице 2.7 также показан класс «Проект» и приведены возможные значения атрибута, который показывает ЖЦ данной сущности. Это сделано на основании проектно-ориентированной структуры СУД, подробно описанной ранее на примере рисунка 1.3. Фактически, предложенный атрибут «статус выполнения» представляет собой стадию ЖЦ объекта типа «Проект».

Следует также отметить, что из набора классов в суперклассе «ЭСИ» подробно рассмотрен класс «Сборочная единица» в связи с многокомпонентным составом этой сущности. Согласно [16, 18, 111], испытаниям в процессе НЭО подвергается не отдельная составная часть изделия, а единая сборочная единица по [22], например, РЭУ или РЭФУ. В связи с этим для класса «Сборочная единица» предложен атрибут «стадия испытаний».

Практическая реализация набора атрибутов и их возможных значений осуществляется в виде задания настроечных параметров типовых объектов в интерфейсе администратора СУД. При отсутствии необходимого атрибута у какого-либо класса-контейнера информации о типовом объекте, следует добавить, изменив модель данных используемой информационной системы,

С учетом состава и иерархии объектов проектирования (выражения (1)-(8), рис. 2.1), динамики изменения стадий их ЖЦ (рис. 2.5), а также вышеприведенных выводов по характеристикам классов объектов проектирования, их атрибутам и операциям с точки зрения управления ЖЦ, была составлена формальная модель. Ее фрагмент показан на рисунке 2.7 в виде UML-диаграммы классов. Следует подчеркнуть, что предложенная модель показывает лишь основные атрибуты и операций классов-контейнеров информации о типовых объектах проектирования в контексте управления их жизненным циклом в рамках ИИС.

Таблица 2.7

**Классы-контейнеры информации об объектах проектирования и перечень атрибутов по состояниям их ЖЦ**

Класс	Атрибут	Значение атрибута	Смысл значения атрибута
Проект	статус выполнения (стадия ЖЦ)	выполнение	осуществляются работы по проекту
		приостановлен	выполнение работ по проекту приостановлено
		выполнен	все работы по проекту выполнены
Изделие	стадия испытаний	*	разработка изделия для нескольких стадий испытаний
		ЛОИ, КДИ, ПРи и т.д.	разработка для конкретной стадии испытаний
	статус готовности	в разработке	выполняется проектирование
		спроектировано	проектирование изделия выполнено
ЭСИ	версия	численное значение	при создании новой версии численное значение атрибута инкрементируется, например, а.0, а.1, и т.д.
	этап ЖЦ в СУД (стадия ЖЦ)	в разработке	выполняется проектирование
		разработано	проектирование сборочной единицы выполнено
		деактивировано	сборочная единица не используется в изделии
Сборочная единица (подкласс ЭСИ)	стадия испытаний	ЛОИ, КДИ, ПРи и т.д.	разработка для конкретной стадии испытаний
Документ (КД, ИМИ)	версия	численное значение	при создании новой версии численное значение атрибута инкрементируется, например, а.0, а.1, и т.д.
	статус блокировки	логическое значение (0 или 1)	показывает, возможность изменения документа (1 - документ заблокирован, изменения невозможны)
	этап ЖЦ в СУД (стадия ЖЦ)	новый	документ локально находится на компьютере пользователя
		опубликован	документ находится на сервере и доступен другим пользователям
		на редактировании	при редактировании опубликованного документа он копируется на компьютер пользователя, осуществляющего эту операцию, и блокируется от просмотра и изменения другими пользователями, по завершении редактирования автоматически создается новая версия документа и размещается на сервере
		выпущен	документ находится на сервере, не доступен для внесения изменений и актуален для применения в производстве, значение атрибута «версия» меняется на буквенное, например А, В и т.д., и называется «релиз»
		не актуален (устарел)	документ находится на сервере, не доступен для внесения изменений и не актуален для применения в производстве
	статус электронной подписи (ЭП)	не подписан ЭП	документ не подписан электронной подписью
		подписан ЭП	документ подписан электронной подписью, считается подлинником (КД) или оригиналом (ИМИ), не подлежит изменению
	статус согласования	не просмотрен	решение о согласовании документа еще не принято (документ не просмотрен согласующим)
согласован		документ согласован всеми согласующими лицами	
не согласован		документ не согласован хотя бы одним согласующим лицом	



Полученная модель в виде UML-диаграммы классов (рис. 2.7) будет использована при разработке программного обеспечения средств управления жизненным циклом разнотипных взаимозависимых объектов проектирования в ИИС. В основе программного обеспечения, как правило, лежат алгоритмические выражения в виде графических схем. Поэтому возникает задача разработки алгоритмов изменения стадий ЖЦ объектов проектирования в ИИС, решение которой представлено далее.

### **2.3 Резюме прикладных исследований**

В данной главе выполнена разработка математического и алгоритмического обеспечения для управления жизненным циклом объектов проектирования с учетом специфики процессов создания высокотехнологичной продукции. В результате проведенных прикладных исследований:

- разработан оригинальный метод формализации структуры взаимосвязей разнотипных взаимозависимых объектов в статическом состоянии с использованием формального аппарата теории множеств, положений объектно-ориентированного подхода и классической теории баз данных;
- выполнено концептуальное проектирование базы данных СУД, предложен набор атрибутов для настройки модели данных СУД с целью определения состояния ЖЦ экземпляров классов, которые являются контейнерами информации о типовых объектах проектирования (Проект, Изделие, ЭСИ, КД, ИМИ);
- разработана UML-диаграмма классов разнотипных взаимозависимых объектов проектирования, которая показывает структуру взаимосвязей исследуемых сущностей и уровни иерархии составляющих их типов данных;
- разработан оригинальный метод формализации динамики изменения стадий жизненного цикла объектов проектирования, включающий методику представления жизненного цикла разнотипных взаимозависимых объектов проектирования в виде автомата Мили и раскрашенного орграфа, формализованную модель в виде UML-диаграммы состояний и алгоритмы изменения стадий жизненного цикла объектов проектирования в ИИС.

Все разработанные модели являются основой алгоритмического обеспечения средств управления жизненным циклом разнотипных взаимозависимых объектов в рамках ИИС, которое подробно описано в следующих разделах настоящей работы.

### **3 АЛГОРИТМИЧЕСКОЕ ОБЕСПЕЧЕНИЕ СРЕДСТВ АВТОМАТИЗИРОВАННОГО УПРАВЛЕНИЯ ЖИЗНЕННЫМ ЦИКЛОМ РАЗНОТИПНЫХ ВЗАИМОЗАВИСИМЫХ ОБЪЕКТОВ ПРОЕКТИРОВАНИЯ В ИИС**

В данной главе решены следующие задачи, поставленные при разработке алгоритмического обеспечения среды автоматизированного управления жизненным циклом разнотипных взаимозависимых объектов проектирования:

- преобразование полученной модели динамики изменений стадий ЖЦ объектов проектирования в формальные модели, которые позволят алгоритмически описать взаимовлияние объектов и логическую последовательность изменения стадий их ЖЦ в ИИС;
- описание логической последовательности действий пользователя ИИС при изменении стадий ЖЦ объектов проектирования;
- разработка метода построения ИИС на базе интеграций САПР и СУД.

#### **3.1 Формальное описание динамики изменений стадий ЖЦ объектов проектирования**

Анализ методов графического описания и документирования программных систем, выполненный в первой главе настоящей работы, позволяет сделать вывод, что функциональные возможности UML-инструментария позволяют решить вышеперечисленные задачи. Для этого решено использовать UML-диаграммы состояний и последовательностей. На рисунках 3.1 – 3.4 представлены UML-диаграммы состояний, которые описывают процесс изменения стадий ЖЦ исследуемых типовых объектов проектирования. В основе этих диаграмм лежат модели ЖЦ объектов в виде автомата Мили, представленные в таблице 2.3.

Однако, поскольку указанная модель автомата Мили иллюстрирует изменение рассматриваемых типовых объектов от исходного состояния к целевому, она не включает некоторые стадии их ЖЦ. Например, набор стадий ЖЦ объекта типа «Документ» (ИМИ и КД), дополняется еще одним состоянием, которое означает, что документ является неактуальным, устаревшим. Это состояние не является целевым, поэтому оно не было рассмотрено при составлении модели ЖЦ типовых объектов проектирования.

Документ может быть объявлен устаревшим только после того, как он прошел стадию ЖЦ «выпущен» и стал доступен уполномоченным пользователем в среде СУД. Состояние «устаревший» является конечным, но не целевым. Конечным целевым состоянием объекта класса

«Документ» является переход в стадию ЖЦ «подписан ЭП», который означает, что проектирование завершено.

Аналогичное состояние можно выделить и у объекта проектирования типа «элемент ЭСИ». При исключении элемента из состава изделия, он переходит в состояние «деактивирован», которое является конечным, но не целевым. Учитывая выявленную зависимость состояний элемента ЭСИ от связанных с ним документов ИМИ согласно разработанной модели динамического изменения стадий ЖЦ объектов проектирования (рис. 2.5), можно сделать вывод, что исключение элемента из состава изделия возможно при отсутствии элемента в новой версии ИМИ. И наоборот, документ становится не актуальным при деактивации связанного с ним элемента ЭСИ. Конечным целевым состоянием объекта класса «элемент ЭСИ» является переход в стадию ЖЦ «выпущен», который означает, что проектирование данного элемента завершено.

С целью получения полной формальной модели ЖЦ каждого типового объекта все их состояния следует отразить при составлении автомата Мили (табл. 2.1-2.3). Полученные таблицы состояний и переходов между ними являются основой для разработки графических моделей ЖЦ исследуемых объектов в виде UML-диаграмм состояний. Также эта информация представляет собой исходные данные при практической настройке правил изменения ЖЦ типовых объектов в СУД,

На рисунках 3.1 и 3.2 представлены UML-диаграммы состояний классов «Документ» и «элемент ЭСИ», которые являются контейнерами информации об объектах проектирования соответствующих типов. Кроме набора стадий ЖЦ объектов также показаны операции перехода между ними. На рисунке 3.3 в виде UML-диаграммы последовательностей представлен перечень операций, которые выполняются при деактивации объектов проектирования типа «Документ» и «элемент ЭСИ». С помощью сообщений, которые обозначают вызов процедуры выполнения операций конкретного класса, показано взаимодействие объектов классов «Документ» и «элемент ЭСИ».

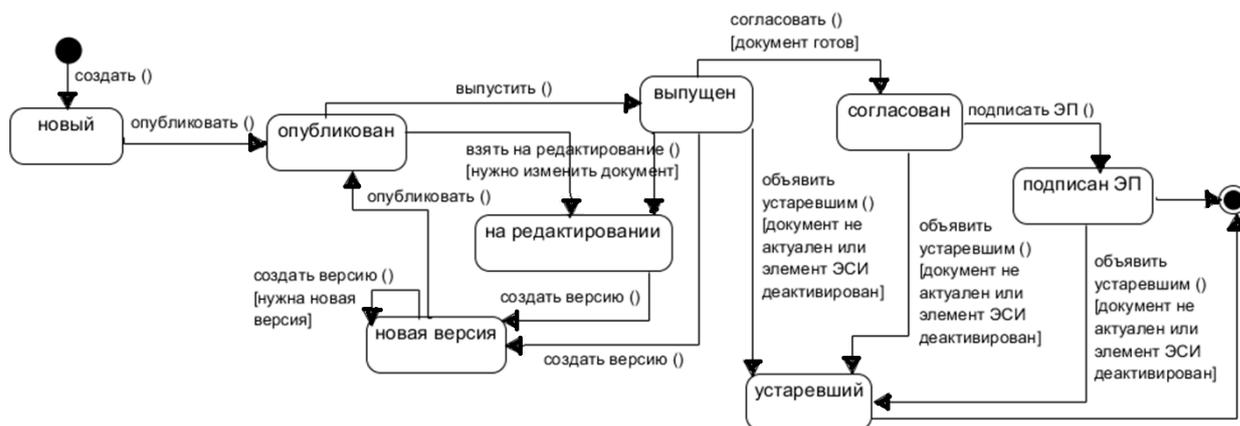


Рис. 3.1. UML-диаграмма состояний объекта класса «Документ»



Рис. 3.2. UML-диаграмма состояний объекта класса «элемент ЭСИ»



Рис. 3.3. UML-диаграмма последовательностей, которая показывает деактивацию объектов проектирования типа «Документ» и «элемент ЭСИ»

На рисунке 3.4 представлена UML-диаграмма объекта класса «Изделия», которая показывает набор стадий ЖЦ этой сущности и операции перехода между ними. Дополнительно к представленным в таблице 2.1 добавлено состояние «разработка остановлена», означающее прекращение работ по проектированию изделия, при том, что изделие еще не находится в целевом состоянии («готово к производству»).

Переход в состояние «разработка остановлена» возможен только из состояния «в разработке». В случае необходимости продолжения разработки возможен обратный переход из состояния «разработка остановлена» в состояние «в разработке». Иначе, стадия ЖЦ изделия «разработка остановлена» является конечной (но не целевой). Конечным целевым состоянием объекта класса «Изделие» является переход в стадию ЖЦ «готово к производству», который означает, что все КД в рамках этой стадии испытаний находятся в целевом конечном состоянии (подписаны ЭП).

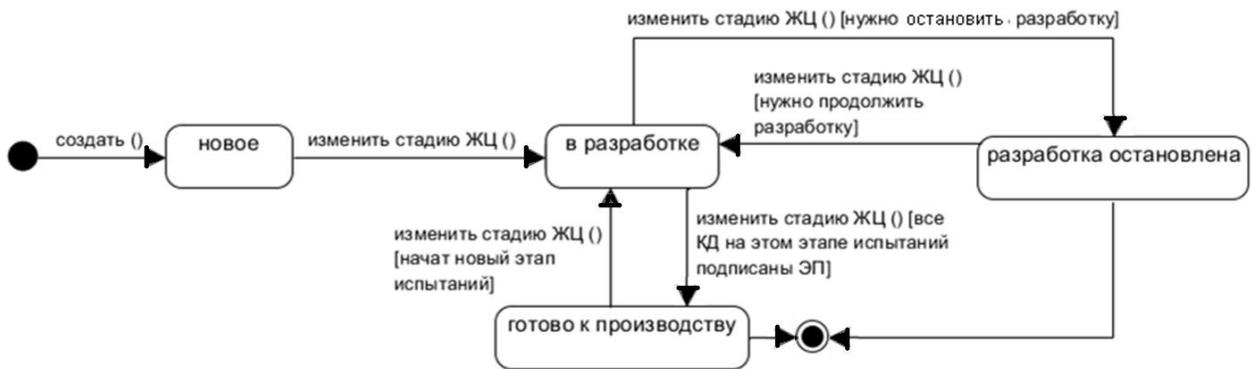


Рис. 3.4. UML-диаграмма состояний объекта класса «Изделие»

В случае создания нового образца изделия в рамках нового этапа испытаний (например, стадия КДИ после ЛОИ) объект класса «Изделие» из состояния «готово к производству» может вернуться обратно в состояние «в разработке». Согласно выявленной взаимозависимости объектов проектирования разных типов, при остановке разработки изделия обработка связанных с ним документов нецелесообразна. Поэтому следует защитить опубликованные в СУД документы от изменения до тех пор, пока разработка изделия не возобновится. На рисунке 3.5 представлена UML-диаграмма последовательностей, которая показывает блокировку опубликованного в СУД объекта класса «Документ» на основании синхронных сообщений от объектов класса «Изделия», которые находятся в состоянии «разработка остановлена».

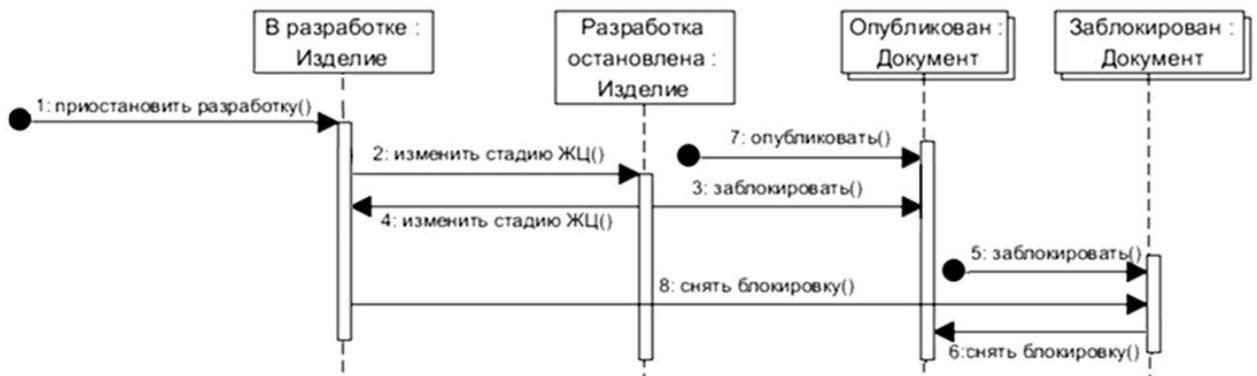


Рис. 3.5. UML-диаграмма последовательностей, которая показывает блокировку объекта класса «Документ»

На рисунке 3.6 представлена UML-диаграмма последовательностей, которая показывает взаимодействия разных типов объектов проектирования в соответствии с линейным порядком их создания и обработки в ИИС. Данная последовательность получена на основании кратчайшего пути (рис. 2.6) модели динамики изменения стадий ЖЦ объектов проектирования в виде раскрашенного орграфа (рис. 2.5), в который включены также целевые состояния объектов проектирования, показанные на рисунке 2.5. висячими вершинами.

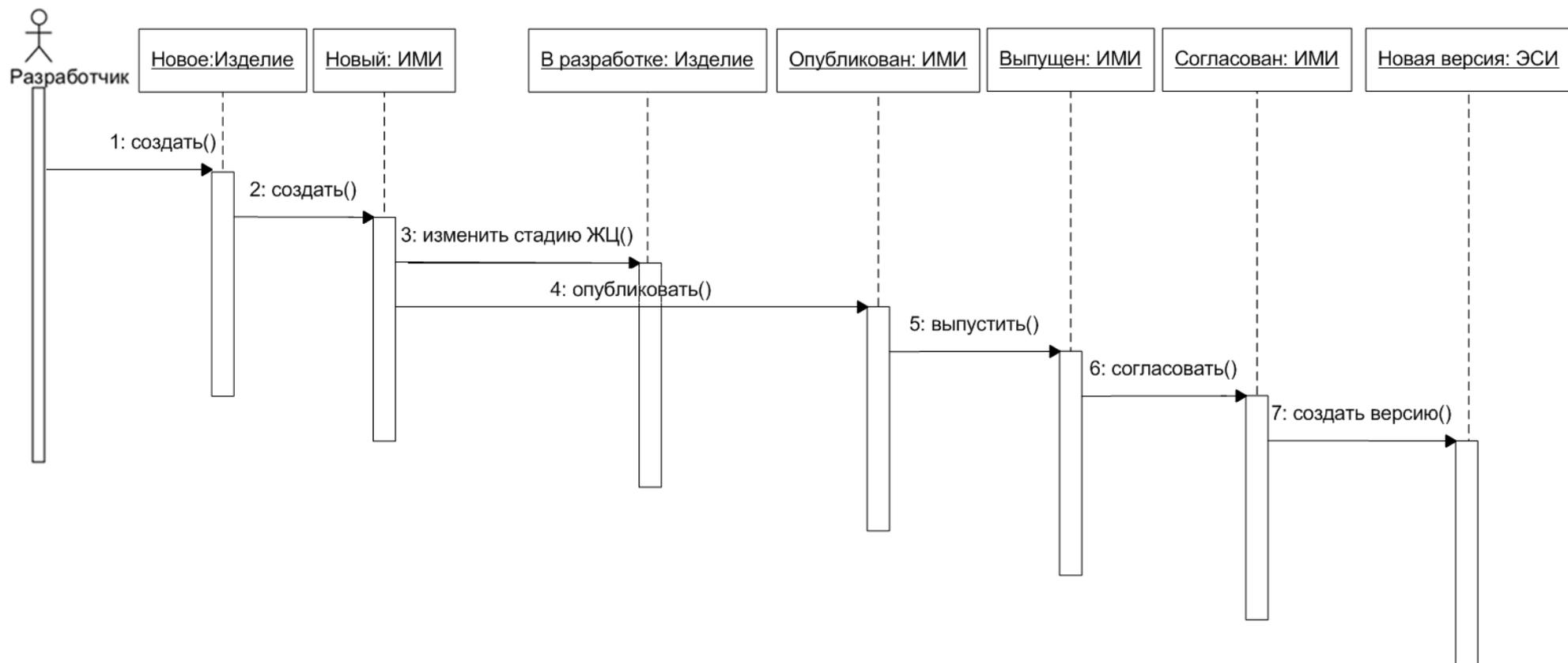


Рисунок 3.6, а) – UML-диаграмма последовательностей, которая показывает взаимодействия разных типов объектов проектирования в соответствии с линейным порядком их создания и обработки рамках в ИИС

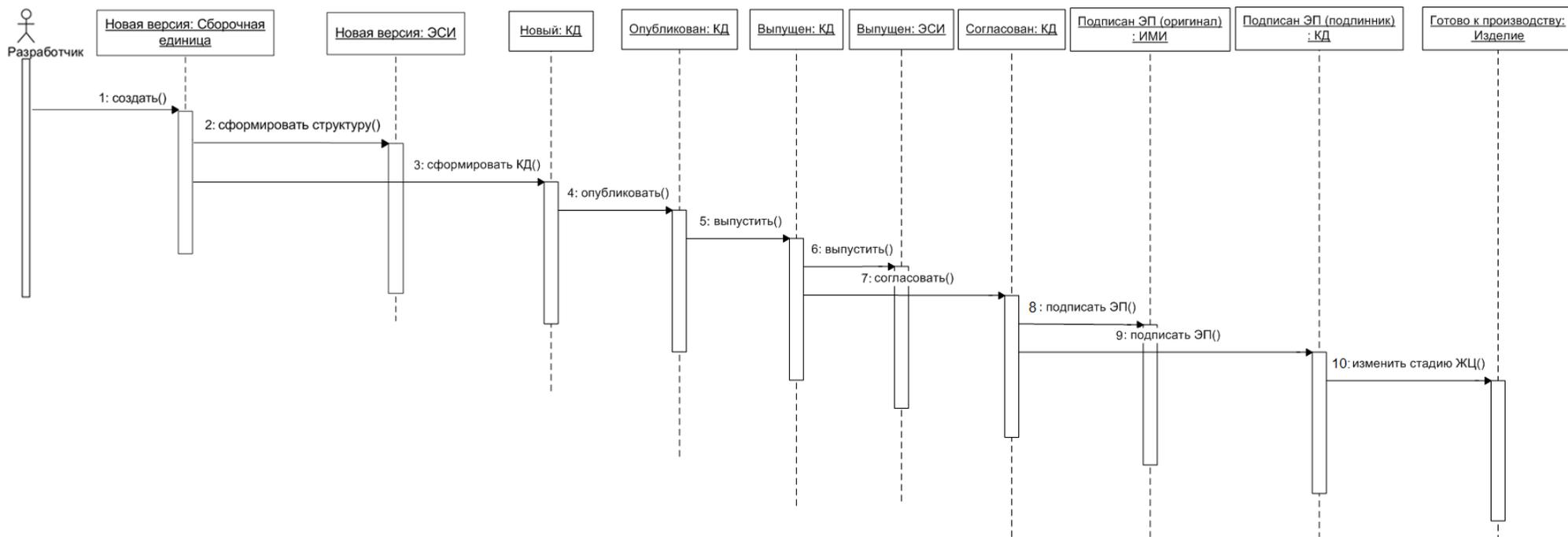


Рисунок 3.6, б) –UML-диаграмма последовательностей, которая показывает взаимодействия разных типов объектов проектирования в соответствии с линейным порядком их создания и обработки рамках в ИИС

Рассматривая последовательность изменения стадий ЖЦ объектов проектирования разных типов, следует отметить ключевую роль согласования документов: сообщение о согласовании документа ИМИ вызывает операцию формирования ЭСИ, а сообщение о согласовании КД – операцию подписи документа с помощью ЭП. В случае не согласования документа создаются его новые версии. Таким образом, согласование документов представляет собой сложный итеративный процесс с участием нескольких пользователей ИИС. Поэтому, в связи с важностью данной операции следует рассмотреть ее более детально.

## **3.2 Алгоритм согласования документов в ИИС**

### **3.2.1 Формализация основных понятий процесса согласования**

Фактически, согласование документа представляет собой совокупность нескольких процессов. Во-первых, это проверка содержимого, результаты которой определяют дальнейшие действия: одобрение в виде подписи (отметки «согласовано») при отсутствии замечаний или же отказ в согласовании с указанием причин. Как правило, согласование технических документов, к которым относятся ИМИ и КД, выполняется несколькими исполнителями. При этом проверка документа может выполняться параллельно, т.е. одновременно несколькими согласующими лицами или последовательно. При последовательном согласовании движение потока работ между согласующими лицами зависит от результатов проверки предыдущих исполнителей.

Для отражения составного характера согласования необходимо представить эту деятельность как взаимосвязанную совокупность инкапсулированных друг в друга процессов, число функций в которых уменьшается по мере вложенности. В терминах методологии Workflow функция процесса называется узлом. Подобное упрощение приводит к элементарному процессу с единственным узлом – заданием для одного исполнителя.

Таким образом, получаем два типа работ по согласованию:

- общий процесс согласования, в котором определены исполнители и порядок выполнения работы (последовательно или параллельно);
- элементарное согласование – задание для единственного исполнителя.

Объекты обоих типов зависят друг от друга, поэтому для описания динамики их поведения целесообразно применить ранее разработанную методику формализации ЖЦ разнотипных взаимозависимых объектов. В таблице 3.1 определены операции, которые изменяют состояния рассматриваемых разнотипных процессов согласования.

Таблица 3.1

**Операции изменения состояний процессов согласования**

Операция	Сигнал
Создать	$y_1$
Запустить	$y_2$
Выполнить	$y_3$
Согласовать	$y_4$
Не согласовать	$y_5$
Завершить	$y_6$
Прекратить	$y_7$

Представим исследуемые взаимозависимые процессы согласования в виде автомата Мили (табл. 3.2).

Таблица 3.2

**Представление процессов согласования в виде автомата Мили**

Тип процесса	Цвет вершины	Состояние		Функция перехода
Согласование документа	Серый	Новое	$p_1$	$f(y_1)$
		Запущено	$p_2$	$f(p_1, y_2)$
		Завершено	$p_3$	$f(p_6, y_4)$
		Прекращено	$p_4$	$f(p_6, y_5)$
Элементарное согласование	Белый	Запущено	$p_6$	$f(p_6, y_3)$
		Выполнено	$p_7$	$f(p_6, y_3)$

Далее, согласно разработанной методике построения формальных моделей ЖЦ разнотипных взаимозависимых для описания порядка взаимодействия между состояниями, построим совмещенную матрицу переходов/выходов автомата Мили (табл. 3.3).

Таблица 3.3

**Совмещенная матрица переходов/выходов состояний автомата Мили по процессам согласования**

	$p_1$	$p_2$	$p_3$	$p_4$	$p_5$	$p_6$
$p_1$		$y_1/y_2$				
$p_2$					$y_2/(y_1 + y_2)$	
$p_3$						
$p_4$						
$p_5$						$(y_1 + y_2)/y_3$
$p_6$			$y_3/y_4$	$y_3/y_5$		

На рисунке представлена иллюстрация динамики изменения состояний автомата Мили по исследуемым процессам согласования в виде раскрашенного графа. Как и в случае с разнотипными объектами проектирования, здесь речь идет не о правильной раскраске графа. Цветовое выделение вершин использовано с целью визуального разделения их по типам процессов.

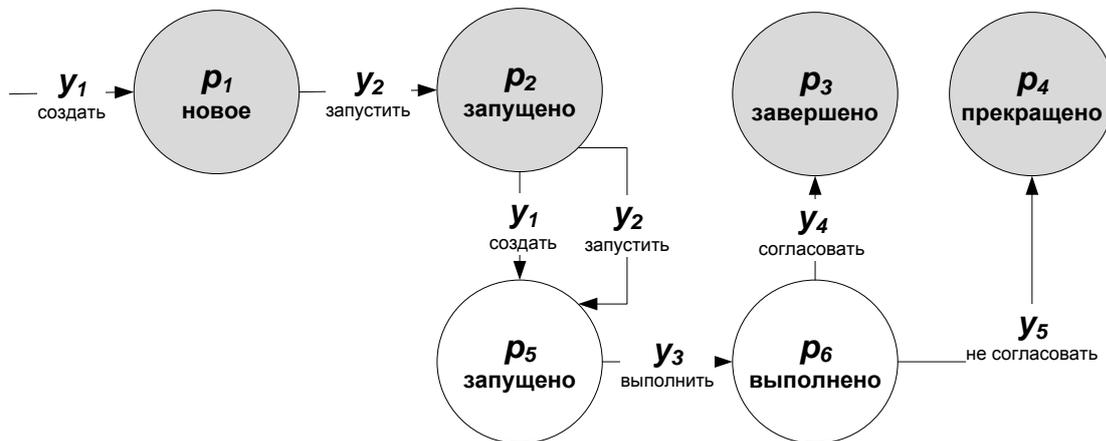


Рис. 3.7. Граф динамики ЖЦ разнотипных процессов согласования

Далее, согласно разработанной методике получения формальных моделей ЖЦ разнотипных взаимозависимых объектов, для разработки программного обеспечения процесса согласования, необходимо соответствующие алгоритмические основы. Для этого решено использовать инструментарий UML. На рисунке 3.8 приведена UML-диаграмма состояний, соответствующая графу динамики ЖЦ разнотипных процессов согласования, показанному на рис. 3.7.

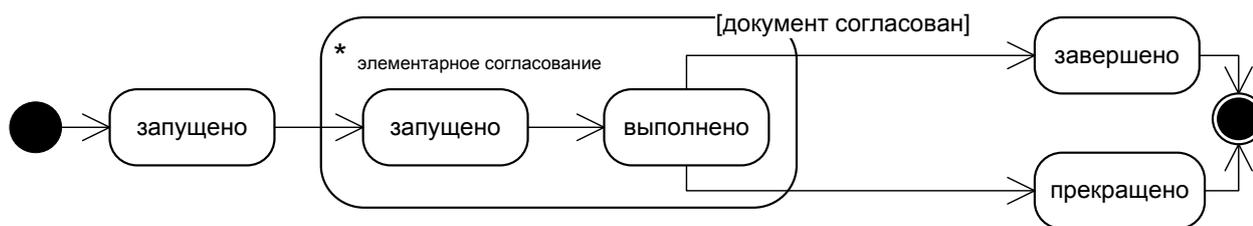


Рис. 3.8. Диаграмма состояний разнотипных процессов согласования

Таким образом, работоспособность ранее разработанной методики получения формальных моделей, описывающих ЖЦ разнотипных взаимозависимых объектов, проверена также и на процессе согласования документов в ИИС, предусматривающем изменение состояний разных сущностей при взаимодействии нескольких пользователей.

Итак, согласование документов является одним из видов работ, выполняемых пользователями ИИС в среде СУД. В соответствии с используемым объектно-ориентированным подходом для обозначения множества действий пользователя следует предусмотреть суперкласс «Работа», потомком которого будет базовый класс «Согласование». Поскольку число согласующих лиц и порядок их работы зависит от типа согласуемого объекта (ИМИ, КД), необходим класс-контейнер для хранения этих параметров согласования. Для этого предлагается класс «Задание параметров на согласование», связанный с классом «Элементарное согласование».

Класс «Элементарное согласование» представляет собой элементарное задание на согласование, каждое из которых получает и исполняет конкретное согласующее лицо. Экземпляров объектов класса «Элементарное согласование» создается столько, сколько согласующих лиц указал разработчик документа в экземпляре класса «Задание параметров на согласование». Если значение атрибута «статус элементарного согласования» у каждого из объектов класса «Элементарное согласование», связанных с согласуемым документом, истинно, то документ переходит в стадию ЖЦ «согласован», иначе документ является не согласованным. Операцию изменения атрибута «статус согласования» у объекта класса «Документ» запускает операция «изменить статус согласования», реализуемая в классе «Запуск на согласования».

Предлагаемая структура классов для работ по согласованию документов представлена на рисунке 3.9.

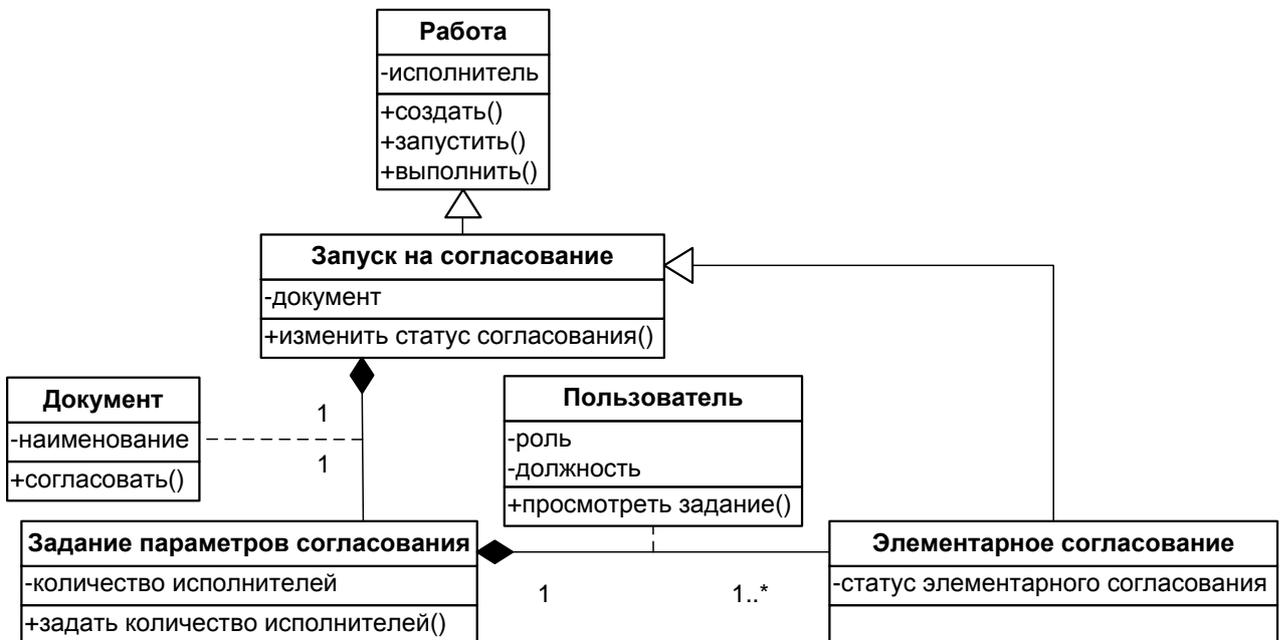


Рис.3.9. UML-диаграмма классов действий пользователей ИИС в части согласования документов

На рисунке 3.10 показана UML-диаграмма деятельности пользователей ИИС при согласовании объектов проектирования, которая иллюстрирует динамическую генерацию множества элементарных заданий на согласование. В качестве согласуемого объекта может выступать экземпляр класса «Документ» (КД или ИМИ), который программным образом переходит в стадию ЖЦ «Согласована» в том случае, если все согласующие лица согласовали объект, т.е. элементарные задания на согласование выполнены со статусом «согласовано».

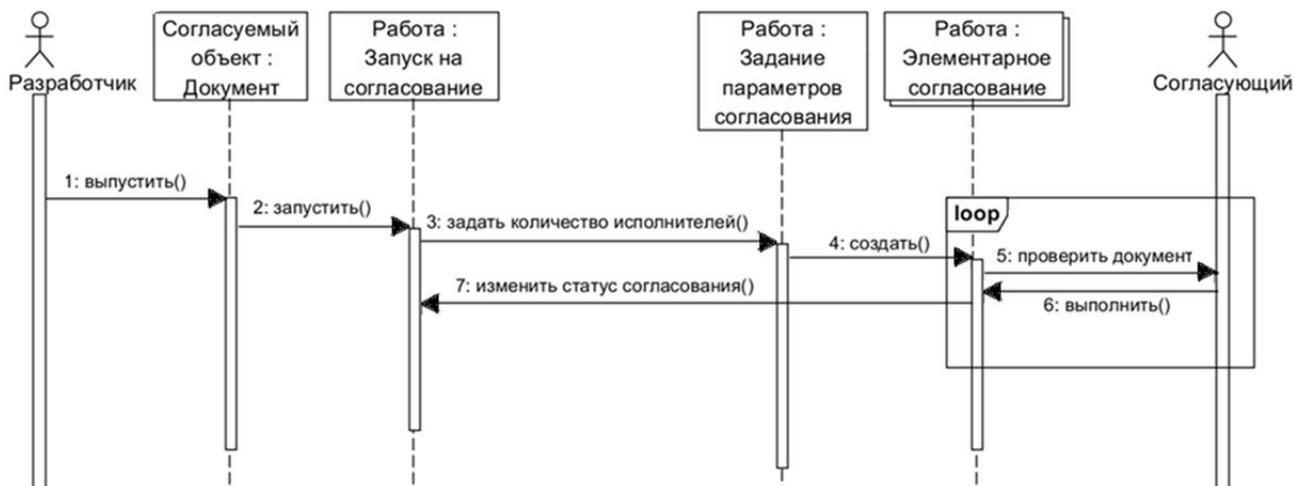


Рис. 3.10. UML-диаграмма последовательностей, которая показывает взаимодействие объектов класса «Работа» и пользователей ИИС при согласовании документов

Вышеизложенные положения являются общими и соответствуют как последовательному, так и параллельному выполнению процесса согласования несколькими пользователями. Процесс согласования документа состоит из конечного числа элементарных согласований, выполняемых единственным исполнителем. Однако, характер формирования элементарных заданий зависит от заданного порядка их следования. Поэтому целесообразно более подробно рассмотреть алгоритмические основы процесса согласования технических документов в ИИС при их последовательном и параллельном выполнении.

### 3.2.2 Последовательное согласование

При последовательном согласовании результат каждого элементарного согласования определяет формирование задания для последующего исполнителя (рис. 3.11).

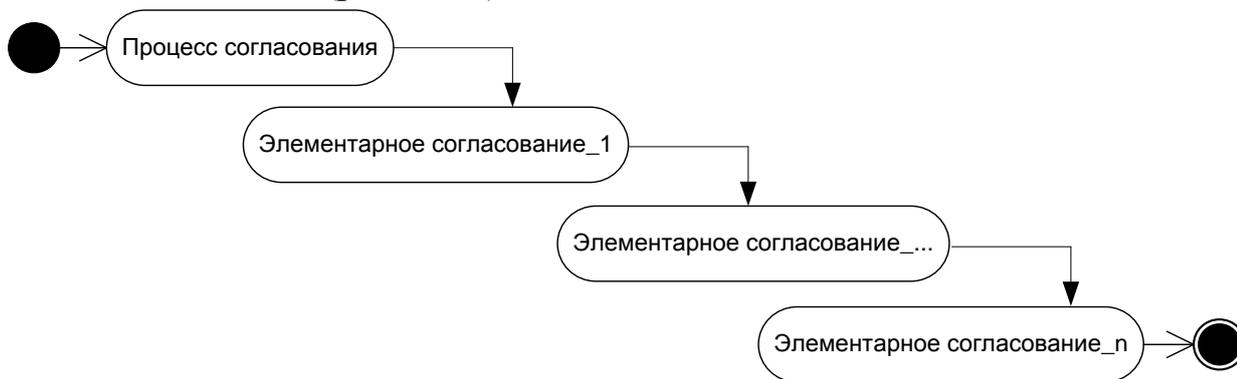


Рис.3.11. Последовательное согласование документа

Анализ рисунка 3.11 позволяет сделать вывод, что формирование элементарных заданий на согласование следующему исполнителю рационально только в случае согласованного результата предыдущим. Ес-

ли на каком-либо шаге исполнитель не согласовывает документ, то весь процесс согласования останавливается. Предлагаемый алгоритм последовательного согласования, описывающий порядок формирования элементарных заданий, показан на рисунке 3.12.

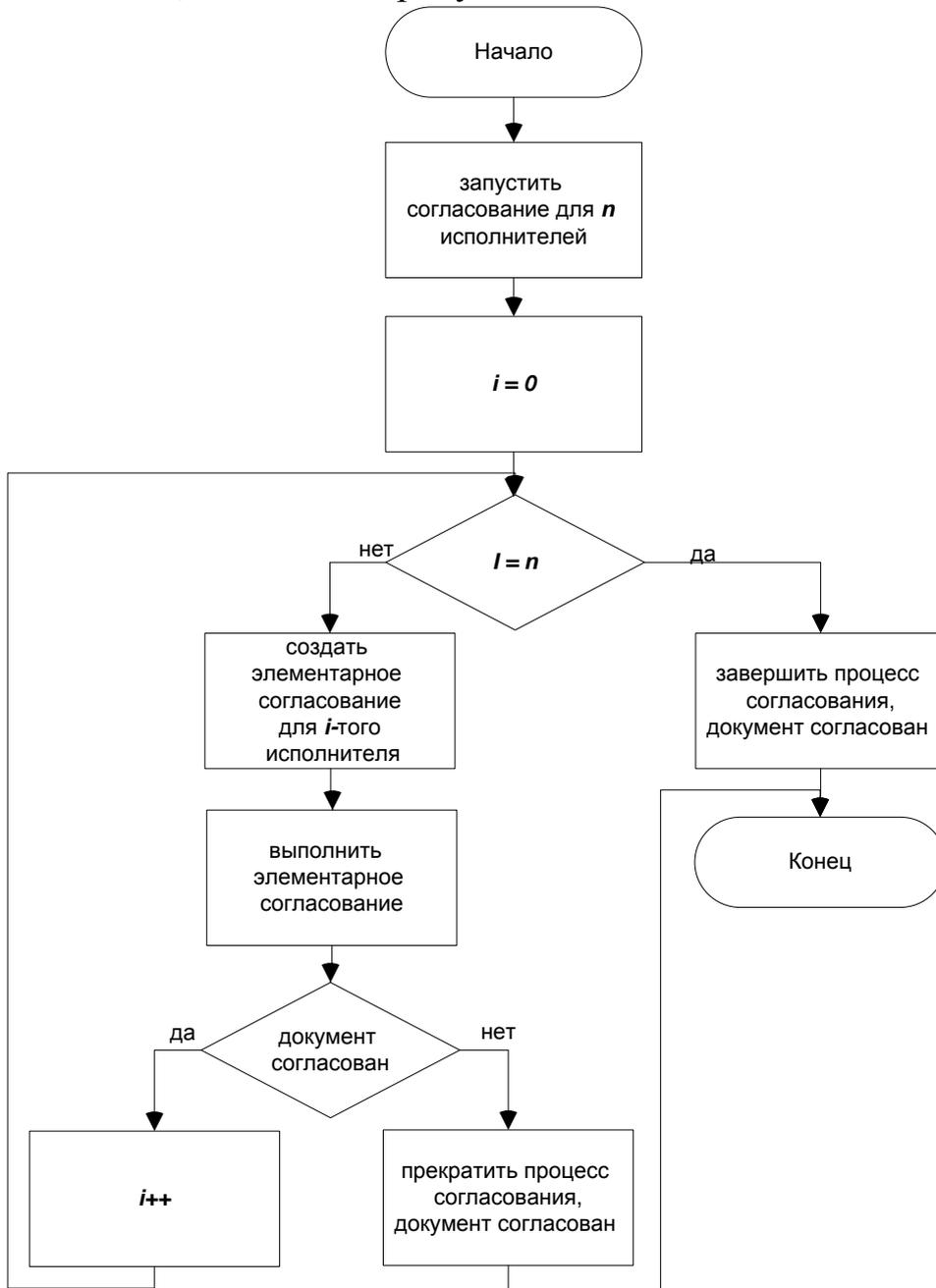


Рисунок 3.12 – Последовательное согласование документа

Согласно предложенному алгоритму, формирование элементарных заданий для единственного исполнителя, происходит только в случае успешного согласования предыдущим участником процесса. Иначе весь процесс прекращается. Это позволяет экономить временные затраты на выполнение неактуальных действий и повысить эффективность работы участников процесса согласования.

### 3.2.3 Параллельное согласование

В случае параллельного согласования работы выполняются несколькими исполнителями одновременно, поэтому процесс детализируется на множество дочерних элементарных заданий по числу исполнителей (рис. 3.13).

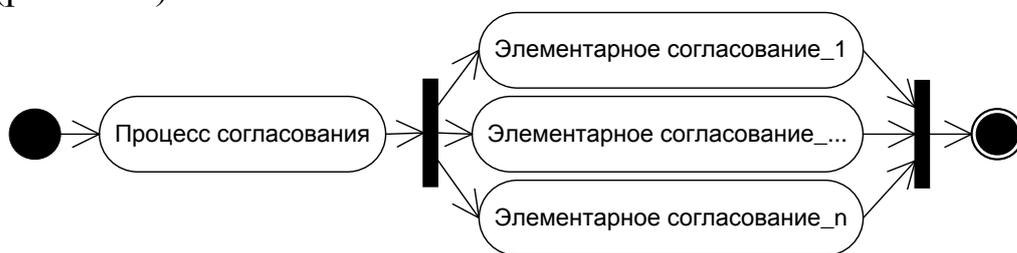


Рисунок 3.13 – Параллельное согласование документа

В отличие от последовательного согласования, при распараллеливании работ, каждая из них не зависит друг от друга. Пусть  $C$  – результат процесса согласования документа, состоящего из  $n$  элементарных заданий  $c$ . Результатом выполнения каждого из элементарных заданий является отметка о согласовании документа данным исполнителем. Данная отметка может быть реализована в качестве логического значения 1 (согласовано) или 0 (не согласовано) в свойстве объекта элементарного задания по согласованию, таким образом,  $C$  и  $c$  имеют одинаковую область значений:  $C, c = \{0, 1\}$ .

После того, как все исполнители элементарных заданий их выполнили, необходимо синхронизировать результаты, т.е. проверить, что все участники процесса согласования поставили отметки «согласовано». Только в этом случае документ считается согласованным и меняется его стадия ЖЦ, а также связанных с ним объектов проектирования. Если документ не согласован кем-либо из исполнителей, то разработчик его корректирует и публикует в СУД новую версию, которая доступна всем согласующим лицам для просмотра и внесения замечаний. Все эти итерации совершаются в рамках единого процесса согласования документа. Когда, наконец, все замечания согласующих лиц устранены и документ согласован каждым из них, документ считается согласованным. Таким образом, результат параллельного согласования документа представляет собой логическое произведение множества элементарных заданий:

$$C = \prod_{i=1}^n c_i \quad (11)$$

Современные информационные системы электронного документооборота, модули которых могут входить в СУД, как правило, поддерживают автоматизацию процессов согласования документов, включая логически сложные маршруты, определяемые пользователем. Функциональные возможности гибкой настройки ЖЦ документа присутствуют

практически в каждой современной системе электронного документооборота. Однако, проведенный анализ наиболее популярных СУД не выявил готовых программно-методических решений по автоматизированному изменению состояния ЖЦ объекта типа «Изделие» или «элемент ЭСИ» в зависимости от результата согласования связанных с ним документов. Поэтому предложенная модель распараллеливания и синхронизации операций процесса параллельного согласования документов была положена в основу разработанного алгоритмического и программного обеспечения автоматизированного управления жизненным циклом разнотипных взаимосвязанных объектов проектирования.

На рисунке 3.14 представлена UML-диаграмма последовательностей, которая иллюстрирует взаимодействие объектов класса «Работа» и пользователей ИИС при параллельном согласовании документов. В качестве акторов (независимых сущностей) показаны разработчик документа и согласующее лицо. Акторы являются пользователями СУД в рамках ИИС. Поскольку процесс проверки и элементарного согласования документа может выполняться несколько раз, он показан как множественное циклическое действие. После выполнения элементарного задания на согласование, объект этого класса становится недоступным для изменения.

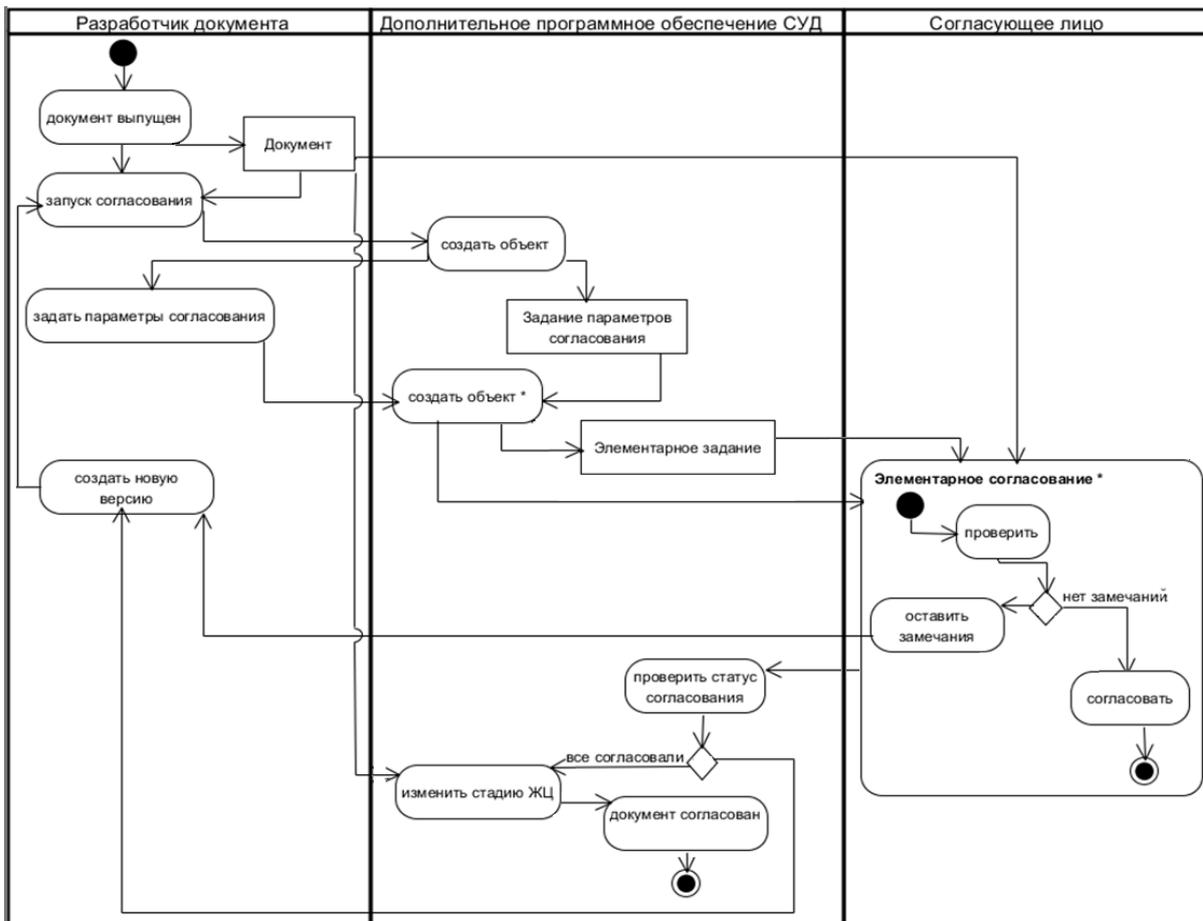


Рисунок 3.14 – UML-диаграмма деятельности при параллельном согласовании

Таким образом, при параллельном согласовании используется идея распараллеливания задач и синхронизации их результатов, проверки на предмет равнозначности результата выполнения. Задача синхронизации выполнения взаимосвязанных работ различными исполнителями во времени относится к области сетевого планирования в рамках дисциплины управления проектами. Методы сетевого планирования ориентированы, в первую очередь, на нахождение оптимальной последовательности выполнения работ в условиях временных и материальных ограничений. Совокупность предложенных моделей и алгоритмов последовательного и параллельного согласования документов в ИИС позволяет оптимизировать деятельность по проектированию высокотехнологичной продукции с точки зрения сокращения времени на выполнение операций согласования и повышение эффективности многопользовательского взаимодействия.

Предложенный подход к организации процессов командного сквозного проектирования на основе структурированного хранения множества версий объектов, автоматического формирования КД на основе согласованных информационных моделей позволит сократить количество документов, создаваемых в процессе разработки. В частности, пропадает необходимость выпуска извещений об изменениях. Таким образом, повышается эффективность процессов проектирования путем сокращения сроков их выполнения и повышения качества за счет динамического распределения потоков работ по согласованию документов между пользователями интегрированной информационной среды.

### **3.3 Использование шаблонов проектирования**

Полученные диаграммы состояний объектов проектирования (рис. 3.1-3.4) и процессов согласования документов (рис. 3.7-3.14) являются алгоритмической основой программного обеспечения управления ЖЦ разнотипных взаимозависимых объектов в ИИС. Для практической реализации предложенных моделей и алгоритмов целесообразно использовать существующие методы и приемы разработки, которые зарекомендовали себя в качестве лучших практик [79-94]. Одной из таких технологий является применение шаблонов проектирования (паттернов, от англ. design pattern), которые представляют собой формализованную модель архитектуры ПО для решения конкретной типовой задачи.

Сопоставление шаблонов, описанных в работах М. Фаулера, М. Гранда, Дж. Кериевски [127-131] и полученных UML-диаграмм (рис. 3.1-3.14) позволяет сделать вывод, что предлагаемые классы объектов проектирования и взаимозависимости между ними соответствуют архитектуре следующих поведенческих шаблонов:

- «Наблюдатель» (Observer), иначе называемый «Издатель-подписчик» (Publisher-Subscriber), который определяет зависимость типа «один ко многим» между объектами таким образом, что при изменении состояния одного объекта все зависящие от него оповещаются об этом событии;
- «Состояние» (State), используемый в тех случаях, когда во время выполнения программы объект должен менять свое поведение в зависимости от своего состояния.

Согласно терминологии [127-131], рассматриваемые классы являются следующими участниками в данном паттерне (рис. 3.15):

- класс «Context» («Контекст») определяет интерфейс работы с объектом, хранит экземпляр подкласса, которым определяется конкретное состояние и делегирует все полученные запросы объекту «current» («Текущее состояние»), который может использовать полученный дополнительный параметр для доступа к экземпляру класса «Context»;
- класс «State» («Состояние») определяет интерфейс для инкапсуляции поведения, ассоциированного с конкретным состоянием контекста с помощью указателя на экземпляр класса «Context»;
- производные от «State» (классы «StateOne», «StateTwo» и т.д.) определяют поведение, специфичное для конкретного состояния.

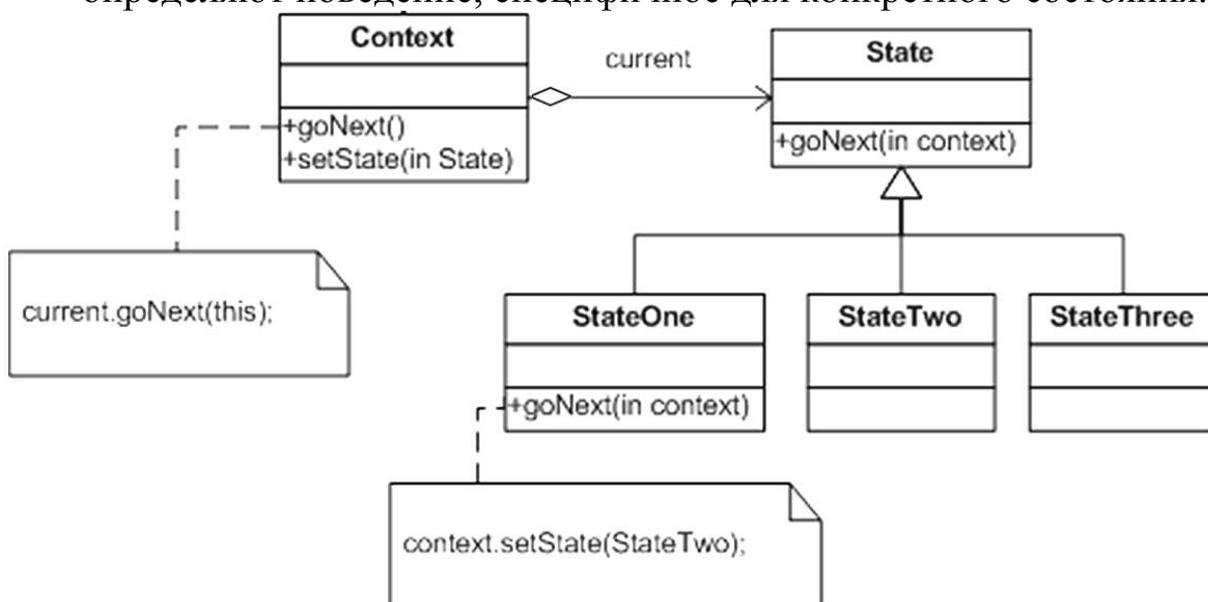


Рисунок 3.15 - UML-диаграмма классов шаблона проектирования «Состояние»

На рисунках 3.16 и 3.17 показано представление стадий ЖЦ объектов класса «Документ» и «элемент ЭСИ» в виде UML-диаграммы клас-

сов шаблона проектирования «Состояние». Согласно терминологии [127-131], рассматриваемые классы являются следующими участниками в данном паттерне:

- класс «Документ»/«элемент ЭСИ» представляет собой контекст;
- класс «Состояние Документа»/«Состояние элемента ЭСИ» определяет интерфейс для инкапсуляции поведения, ассоциированного с конкретным состоянием контекста (класс «Документ»/«элемент ЭСИ»);
- подклассы «Новый», «Опубликован», «Новая версия», «На редактировании», «Выпущен», «Согласован», «Подписан ЭП», «Заблокирован», «Устаревший», «Деактивирован» реализует конкретное поведение, ассоциированное с некоторым состоянием контекста (класс «Документ»/«элемент ЭСИ»).

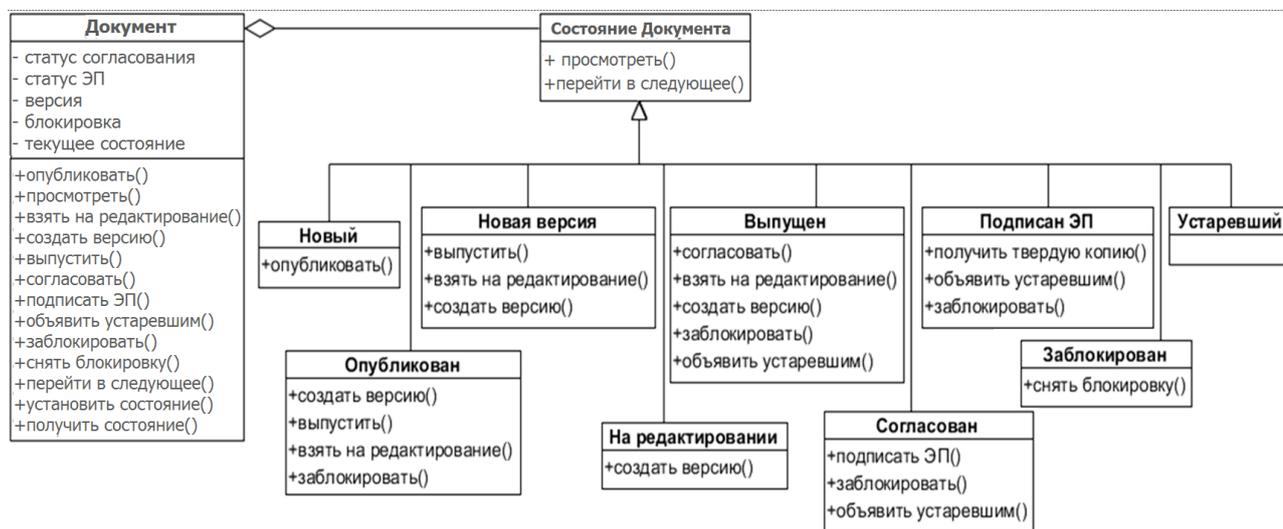


Рисунок 3.16 – Представление стадий ЖЦ объектов класса «Документ» в виде UML-диаграммы классов шаблона проектирования «Состояние»

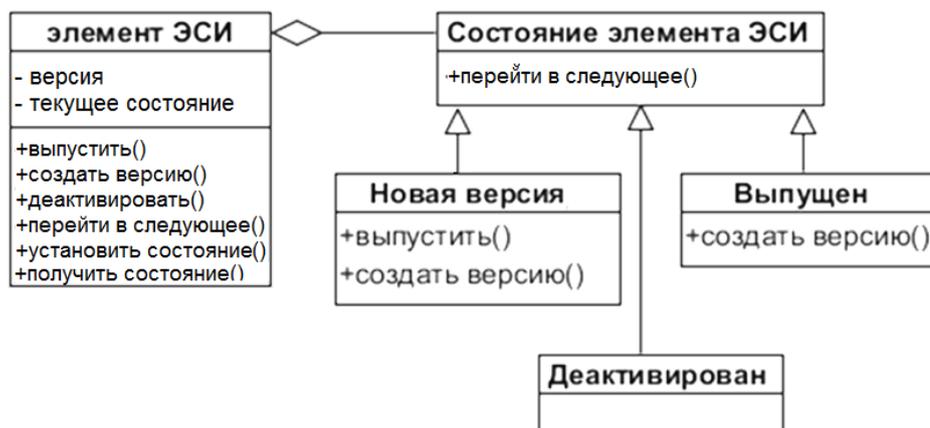


Рисунок 3.17 – Представление стадий ЖЦ объектов класса «элемент ЭСИ» в виде UML-диаграммы классов шаблона проектирования «Состояние»

Представленные на рисунках 3.15 и 3.16 UML-диаграммы классов показывают какие операции реализуется в том или ином состоянии, которые, в свою очередь, выделены как отдельный класс. В связи с тем, что операция просмотра документа должна быть доступна в любом его состоянии, она определена в классе «Состояние Документа». Использование шаблона проектирования «Состояние» позволяет локализовать зависящее от состояния поведение и делит его на части, соответствующие стадиям ЖЦ. За счет такого разделения по классам процесс добавления новых стадий ЖЦ к существующему набору состояний гораздо облегчается.

Однако, согласно [128], одним из недостатков паттерна «Состояние» является то, что в нем не содержатся сведения о том, какой именно из классов-участников данного шаблона проектирования определяет критерий перехода между состояниями. Эта задача определения моментов изменения состояний решается с помощью подробного описания взаимодействий объектов проектирования разных типов, ранее показанных в настоящей работе с помощью UML-диаграмм последовательностей. Согласно выявленным зависимостям состояний объектов проектирования разных типов друг от друга, изменение объекта одного типа запускает изменение стадий ЖЦ связанных с ним объектов проектирования других типов (например, при деактивации элемента ЭСИ, остановке процесса разработки изделия и т.д.). Таким образом, при модификации одного объекта требуется изменить другие, число которых может варьироваться и связь между объектами может быть задана не жестко. Следовательно, один объект должен оповещать других, не делая предположений об уведомляемых сущностях. На основании анализа работ М. Фаулера, М. Гранда, Дж. Кериевски [127-131] можно сделать вывод, что в данном случае целесообразно применение паттерна «Наблюдатель».

Согласно терминологии [127-131], в данном шаблоне проектирования участвуют следующие классы (рис. 3.18):

- класс «Субъект» предоставляет интерфейс для присоединения и отделения наблюдателей располагает информацией о своих наблюдателях, за субъектом может «следить» любое число наблюдателей;
- класс «Наблюдатель» определяет интерфейс обновления для объектов, которые должны быть уведомлены об изменении субъекта;
- класс «Конкретный субъект» сохраняет состояние, представляющее интерес для конкретного наблюдателя (класс «Конкретный наблюдатель»), посылает информацию своим наблюдателям, когда происходит изменение;

– класс «Конкретный наблюдатель» хранит ссылку на объект класса «Конкретный субъект», сохраняет данные, которые должны быть согласованы с данными субъекта, реализует интерфейс обновления, определенный в классе «Наблюдатель», чтобы поддерживать согласованность с субъектом.



Рис. 3.18. UML-диаграмма классов шаблона проектирования «Наблюдатель»

На рисунке 3.19 показано представление взаимосвязей классов «Изделие», «элемент ЭСИ» и «Документ» в виде UML-диаграммы классов шаблона проектирования «Наблюдатель». Согласно терминологии [127-131], рассматриваемые классы являются следующими участниками в данном паттерне:

- класс «Управление наблюдателем» представляет собой субъект;
- классы «Элемент ЭСИ» и «Изделие» являются конкретными субъектами;
- класс «Наблюдатель» определяет интерфейс обновления для объектов, которые должны быть уведомлены об изменении субъекта, в частности, обновляет значение атрибута «блокировка» у объекта класса «Документ» или объявляет его устаревшим;
- класс «Документ» хранит ссылку на связанные с ним объекты классов «Изделие» и «элемент ЭСИ», реализует операции изменения состояния документа (например, блокирует его, если разработка изделия остановлена или объявляет устаревшим, если элемент ЭСИ деактивирован).



Рис.3.19. Взаимосвязи классов «Изделие», «элемент ЭСИ» и «Документ» в виде UML-диаграммы классов шаблона проектирования «Наблюдатель»

Паттерн «Наблюдатель» также может быть использован при реализации процесса согласования документов. В этом случае классы, показанные на рисунке 3.7, являются следующими участниками в данном паттерне:

- класс «Задание параметров согласования» представляет собой субъект (в нем реализуется интерфейс добавления и оповещения наблюдателей);
- классы «Элементарное задание на согласование» является конкретным субъектом, сохраняет результат согласования документа конкретным пользователем, при этом посылает информацию наблюдателю (класс «Запуск на согласование»);
- класс «Запуск на согласование» является наблюдателем, определяет интерфейс обновления для объектов, которые должны быть уведомлены об изменении конкретного субъекта: операция «изменить статус согласования» у класса «Запуск на согласования» запускает операцию изменения атрибута «статус согласования» у объекта класса «Документ»;
- объекты класса «Документ» являются конкретными наблюдателями, поскольку состояние меняется в зависимости от значения атрибута «статус элементарного согласования» объектов класса «Элементарное согласование», связанных с согласуемым документом.

Совокупное использование типовых архитектурных приемов в виде шаблонов проектирования «Состояние» и «Наблюдатель» позволит сократить время разработки ПО на основе предложенных моделей и алгоритмов и повысить уровень абстракции программного кода.

Анализируя разработанные модели и алгоритмы управления ЖЦ разнотипных взаимозависимых объектов (рис. 3.1-3.19), можно сделать вывод, что они относятся к задаче расширения функциональных возможностей СУД. Однако, ИИС для проектирования высокотехнологичной продукции предполагает, как минимум, взаимодействие СУД с несколькими САПР. Таким образом, возникает задача разработки метода построения ИИС на базе интеграций САПР и СУД.

### **3.4 Метод построения интегрированной информационной среды**

На основании выполненного анализа источников литературы по проблемам PLM-технологий [7, 45-63, 69-75] сделан вывод, что ключевыми программными компонентами ИИС проектирования высокотехнологичной продукции являются САПР и СУД. Поэтому далее в настоящей работе под ИИС понимается взаимодействие информационных систем этих категорий.

В рамках разработки метода построения ИИС на базе САПР и СУД необходимо решить следующие задачи:

- выявление требований к интеграции САПР и СУД;
- определение структуры хранения и вариантов использования справочных данных, которые создаются и применяются при проектировании;
- разработка моделей и алгоритмов представления данных об изделии в СУД;
- анализ существующих методов и технологий программной интеграции ИС.

Разрабатываемый метод построения ИИС на базе интеграций САПР и СУД с целью управления ЖЦ объектов проектирования должен носить общий характер и быть адаптируемым к особенностям предметных областей внедрения и специфике конкретных предприятий.

#### **3.4.1 Разработка требований к интеграции САПР и СУД в рамках ИИС**

На основании взаимовлияния разнотипных объектов проектирования, описанного в предыдущих разделах настоящей работы, можно выделить следующие категории требований:

- 1. Требования к целостности, полноте и аутентичности информации:**
  - СУД является централизованным хранилищем всех характеристик изделия, проектируемых в различных САПР;
  - однозначное непротиворечивое представление в СУД объекта, описанного с различных точек зрения в разных САПР;
  - исключение дублирования данных;
  - автоматическая синхронизация данных в различных ИС;
- 2. Требования к автоматизации процесса проектирования:**
  - автоматическое формирование ЭСИ в СУД;
  - автоматическое формирование полного комплекта КД в СУД;
- 3. Требования к безопасности:**
  - аутентификация и авторизация пользователей;
  - распределение прав пользователям на доступ к информации согласно их ролям в процессе проектирования;
  - организация хранилищ данных на распределенных серверах;
  - резервирование хранилищ данных;
- 4. Требования к управлению версиями данных:**
  - возможность существования нескольких версий объектов;
  - любое изменение опубликованного в СУД объекта должно отображаться в его новой версии, генерируемой автоматически;
  - накопление данных и опыта, возможность использования информации из предшествующих проектов для текущих разработок.

Рассматривая вышеуказанные требования к ИИС с точки зрения программного сопряжения САПР и СУД, можно выделить следующие аспекты интеграции данных программных компонент ИИС:

- возможность просмотра в СУД содержимого файлов ИМИ, разработанных в различных САПР;
- возможность открытия для редактирования ИМИ-файлов САПР из интерфейса СУД, с учетом версии и стадии ЖЦ документа;
- экспорт ИМИ-файлов САПР в СУД (сборка/проект и их составные части) с выбором проекта, изделия, элемента ЭСИ, а также с учетом версии и стадии ЖЦ документа;
- автоматическая генерация ЭСИ на основе данных из множества ИМИ-файлов, импортированных в СУД из различных САПР;
- автоматическая генерация комплекта КД на изделие, с учетом версии и стадии ЖЦ документа, по заданному шаблону на основе

ЭСИ и данных из множества ИМИ, импортированных в СУД из различных САПР;

– непротиворечивость и полнота представления в СУД информации об объекте, описанного в различных САПР.

Перечисленные задачи сводятся к определению правил хранения и обработки информации, включая как вновь созданные в процессе проектирования данные (ИМИ, КД, ЭСИ), так и сведения об уже существующих и успешно применяемых решениях. Также следует принять во внимание, что высокотехнологичные изделия, например, в отрасли машиностроения, космической промышленности и т.д., состоят из множества типовых компонентов, применение которых четко регламентировано государственными и отраслевыми стандартами. Информация об этом может быть названа типовыми (справочными) данными, которые, согласно положению PLM-методологии о центральной роли СУД в ИИС, должны храниться в СУД с возможностью дальнейшего использования.

### **3.4.2 Разработка структуры хранения типовых справочных данных**

В соответствии с классификацией составных частей изделия по [22] и [120], многие его элементы являются типовыми, т.е. описываются с помощью унифицированных характеристик, 3D-моделей и условных графических обозначений (УГО). Поэтому целесообразно организовать хранение типовых объектов в СУД в виде справочных данных, на которые будут ссылаться сущности из ЭСИ.

Типовой объект характеризуется меньшим набором атрибутов, чем реальная сущность, например, атрибут расположения объекта относительно других элементов изделия (x-, y- координаты) присущ только реальному элементу изделия, и потому отсутствует в наборе атрибутов связанного с ним справочных данных (типового объекта). Кроме того, справочными данными описывается лишь диапазоны значений атрибутов элементов, а не конкретные значения. Например, атрибут «Ширина выводов» у типовой микросхемы (объект класса «Прочие изделия») может варьироваться от 0.33 до 0.51 [132], однако в каждом конкретном случае элемента реального изделия этот атрибут принимает единственное дискретное значение. Аналогичные ситуации возникают и в случае механических изделий, например, болтов и т.д.

Следовательно, экземпляры классов суперкласса «ЭСИ» должны ссылаться на типовые объекты, которые содержат справочные данные о рассматриваемой сущности. Каждый элемент реального изделия связан с соответствующим ему типовым элементов отношением кратности «один к одному», поскольку реальная сущность, к примеру, транзистор

или болт, не может быть основанным на нескольких типовых элементах одного класса одновременно.

Таким образом, каждый реальный элемент ЭСИ, который используется при проектировании изделия, описывается следующей совокупностью данных:

- значения атрибутов экземпляра соответствующего класса «ЭСИ»;
- справочная информация о типовом элементе рассматриваемого класса.

Поскольку в основе любого элемента изделия может лежать типовой объект, классификация справочных данных аналогична реальным и может быть описана выражениями, полученными в главе 2 настоящей работы. Поэтому целесообразно создать специальный проект «Справочники», с которым будут связаны нормативно-справочные документы (например, государственные и отраслевые стандарты), а также элементы ЭСИ.

Значения атрибутов объекта класса «ЭСИ» (например, «Ширина выводов») характеризуют только некоторые количественные и качественные параметры данной сущности. Полное описание элемента изделия содержится в его информационных моделях, которые представляют собой файлы САПР. Таким образом, с типовыми (справочными) объектами ЭСИ должны быть связаны описывающие их информационные модели. На рисунке 3.20 приведена схема предлагаемой структуры хранения справочных данных в СУД в виде UML-диаграммы объектов.



Рис. 3.20 – UML-диаграммы объектов, которая иллюстрирует предложенную структуру хранения справочных данных (типовых объектов) в СУД

Схема на рисунке 3.20 показывает, что в рамках дерева проектов создан специальный проект «Справочники», с которым связано множество отдельное дерево документов. В дереве проектов организована иерархическая структура типов справочных данных, например, ЭРИ. При этом множество элементов (ЭРИ), которые относятся к рассматриваемому типу, представлены в отдельном дереве ЭСИ. Документация на каждый элемент ЭСИ, включая его информационные модели, содержится в отдельном, связанном с ним дереве документов.

В свою очередь, файлы, импортированные из САПР в СУД, хранятся в экземплярах класса «ИМИ» и могут быть использованы в качестве основы для разработки новых объектов, т.е. вновь экспортированы в САПР, доработаны и сохранены в СУД под новым именем. При этом необходимо сохранить связь между всеми этими объектами. Это условие должно быть учтено при разработке алгоритма экспорта ИМИ из САПР в СУД.

Использование типовых объектов класса «ЭСИ», описывающих элементы изделия, позволит сократить объем информации, хранящейся в СУД, и избежать дублирования данных. Поэтому необходимо определить процедуру наполнения СУД справочной информацией. В связи с большим объемом подобных данных, их создание в СУД должно выполняться автоматически, при импорте ИМИ из САПР. Для обеспечения непротиворечивости и полноты представления информации об объекте, описанного в различных ИС, разработанная в САПР сущность должна быть без потерь экспортирована в СУД. Это обеспечивается наличием в СУД соответствующих контейнеров (классов с необходимым набором атрибутов).

Анализируя разработанную UML-диаграмму классов (рис. 2.7), а также требования к однозначности и полноте представления в СУД информации об объекте, описанного в различных САПР, можно отметить следующее:

- экземпляры классов суперкласса «ЭСИ» в СУД являются хранилищами информации о составной части изделия (детали, сборочной единицы, комплекта, комплекса и т.д. по [22]);
- различные характеристики составной части изделия описываются с помощью экземпляров классов суперкласса «ИМИ», файлы которых разработаны в различных САПР и импортированы в СУД.

Учитывая данные замечания, требования к непротиворечивости и полноте представления в СУД информации об объекте, описанного в различных САПР, а также положения объектно-ориентированного подхода, можно сделать вывод, что класс «ЭСИ» является контейнером для информации из файлов ИМИ, импортированных в СУД из различных

САПР. Поэтому определение набора атрибутов класса «ЭСИ» является отдельной задачей, решение которой влияет на организацию программного сопряжения САПР и СУД в рамках ИИС.

Чтобы избежать ситуации с дублированием или потерей данных, необходимо знать контрольное значение атрибута, которое будет одинаково для одного элемента ЭСИ, описанного в разных ИМИ, разработанных в различных САПР. Это позволит реализовать однозначное представление элемента изделия в СУД. С целью выделения данного атрибута из набора характеристик класса, обозначим его термином «ключевой атрибут». Например, для ЭРИ таким атрибутом может являться составное выражение: «Обозначение корпуса элемента и его расположение в ячейке схемы». Значение ключевого атрибута должно проверяться при автоматическом формировании ЭСИ в СУД на основе ИМИ, разработанных в разных САПР.

Таким образом, далее целесообразно определить алгоритм экспорта ИМИ из САПР в СУД, с учетом обозначенных выше особенностей структуры хранения и использования справочных данных.

### **3.4.3 Разработка алгоритма экспорта ИМИ из САПР в СУД**

Согласно выражениям (1)-(6), файлы ИМИ и формируемая на их основе КД являются экземплярами соответствующих классов суперкласса «Документ». В свою очередь, элемент изделия, описываемый с помощью ИМИ, должен быть интегрирован в дерево элементов ЭСИ. При этом следует также учитывать, является ли объект, импортируемый в СУД, типовым. В этом случае его следует считать справочным данным и требуется настроить ссылку на уже существующую в СУД сущность. Таким образом, при экспорте данных из САПР в СУД необходимо генерировать объекты соответствующих классов в рамках одного проекта и одного изделия. Кроме того, поскольку СУД является многопользовательской ИС, необходимо предусмотреть процедуры аутентификации и авторизации при импорте данных из САПР в СУД.

Учитывая вышесказанное, целесообразно определить алгоритм действий при экспорте ИМИ из САПР в СУД в виде следующей логической последовательности:

1. Аутентификация и авторизация пользователя СУД, включая выбор базы данных СУД и определение набора операций, доступных пользователю в рамках выбранной БД.
2. Учитывая выражение (2), смысл которого в том, что разработка изделия выполняется в рамках конкретного проекта, необходимо выбрать один экземпляр класса «Проект» из существующих в выбранной базе данных СУД. При этом

пользователь должен иметь право работать с выбранным проектом.

3. Далее аналогичным образом, следует произвести выбор изделия и его составной части (сборочной единицы), т.к. в рамках одного проекта может выполняться разработка нескольких изделий и сборочных единиц.
4. Поскольку экспортируемые из САПР файлы ИМИ в СУД являются экземплярами соответствующих классов суперкласса «Документ», должно происходить автоматическое создание этих объектов в дереве документов, включая связь с проектируемым изделием и его составной частью. При этом следует учитывать комплексный характер САПР-файлов, например, сборка из конечного множества деталей. В таком случае в СУД должна сохраниться иерархичная совокупность всех составных частей объекта.
5. После того, как экспорт файлов ИМИ из САПР в СУД завершен, становится возможным формирование ЭСИ.

Графическая схема разработанного алгоритма представлена на рисунке 3.21.

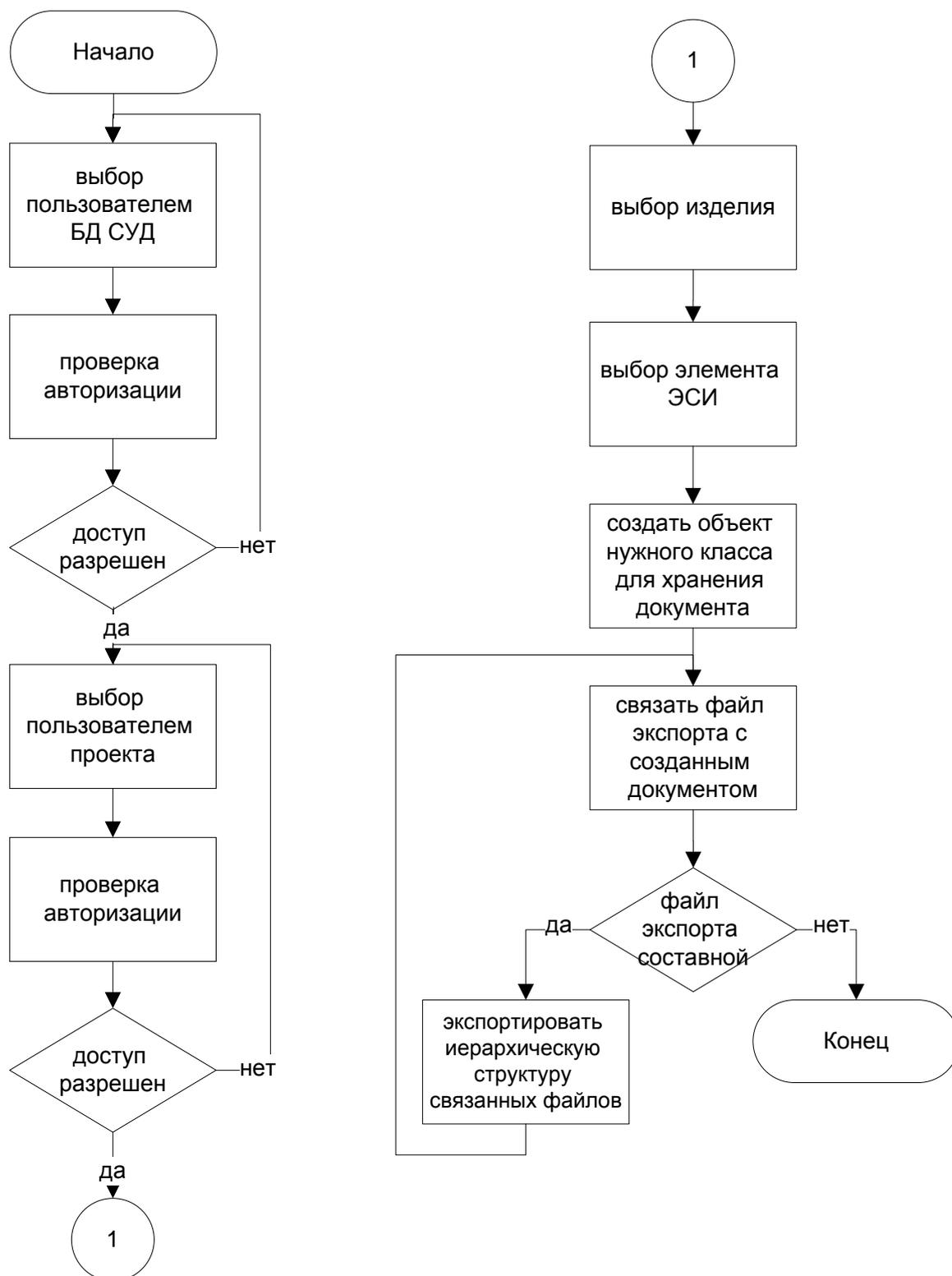


Рисунок 3.21 – Алгоритм экспорта ИМИ из САПР в СУД

### 3.4.4 Разработка алгоритма формирования ЭСИ в СУД

Согласно выражению (3), первоисточником данных для формирования ЭСИ является содержимое множество файлов ИМИ, в которых описаны количественные и качественные характеристики элементов ЭСИ. Таким образом, для создания элементов изделия в структурированном порядке необходимо, прежде всего, определить,

какие сущности содержатся в файлах ИМИ. Это возможно сделать, если данные о внутреннем содержимом ИМИ представлены в открытом структурированном формате, например, XML, JSON, STEP [144-146]. В этом файле, согласно семантике языка должны быть описаны элементы ЭСИ и их характеристики, которые моделируются в САПР. Наиболее простой и доступный способ формирования этих данных состоит в использовании функций сохранения результатов в САПР, например «Сохранить как» или «Экспортировать в ...». Затем необходимо присоединить эти данные к пакету файлов ИМИ, которые должны быть переданы в СУД.

Чтобы определить алгоритм формирования ЭСИ, рассмотрим логическую последовательность функций данного процесса:

1. Начальным шагом формирование электронной структуры изделия или его сборочной единицы в СУД является ее выделение и вызов пользователем соответствующей команды.
2. Согласно выражению (3), первоисточником данных для ЭСИ является множество ИМИ. Таким образом, следует выполнить поиск документа ИМИ, связанного с изделием (сборочной единицей), электронную структуру которого необходимо построить.
3. Далее следует прочитать информацию об элементе ЭСИ из файла, в котором описано содержимое ИМИ (файл формата JSON, XML, STEP и т.д.). Значение ключевого атрибута найденного объекта определяет, к какому классу «ЭСИ» он относится.
4. При наличии в дереве элементов ЭСИ, связанных с изделием (сборочной единицей), структура которого формируется, объекта с равным значением ключевого атрибута, следует дополнить информацию об объекте (которая представлена в виде атрибутов) данными, описанными в рассматриваемом файле ИМИ;
5. Иначе (при отсутствии объекта с равным значением ключевого атрибута), следует создать новый объект необходимого класса и заполнить его атрибутами значениями на основе информации из файла, который описывает рассматриваемую ИМИ;
6. Найти в структуре проекта «Справочники» объект, который соответствует вновь созданному (найденному) элементу ЭСИ;
7. Иначе следует создать в структуре проекта «Справочники» новый типовой объект соответствующего класса и заполнить его значения атрибутов;
8. Связать типовой (в структуре проекта «Справочники») и реальный объекты ЭСИ;

9. Повторение шагов 3-8 выполняется до тех пор, пока не обработаны все элементы ЭСИ, описанные в рассматриваемом файле с описанием ИМИ.
10. Повторение шагов 2-9 выполняется до тех пор, пока не обработаны все документы ИМИ, связанные с изделием, электронную структуру которого необходимо сформировать. Сформированное по завершению данного цикла структурированное множество объектов представляет собой ЭСИ в стадии ЖЦ «Новая версия».

На рисунке 3.22 приведена графическая иллюстрация разработанного алгоритма по формированию ЭСИ в СУД.

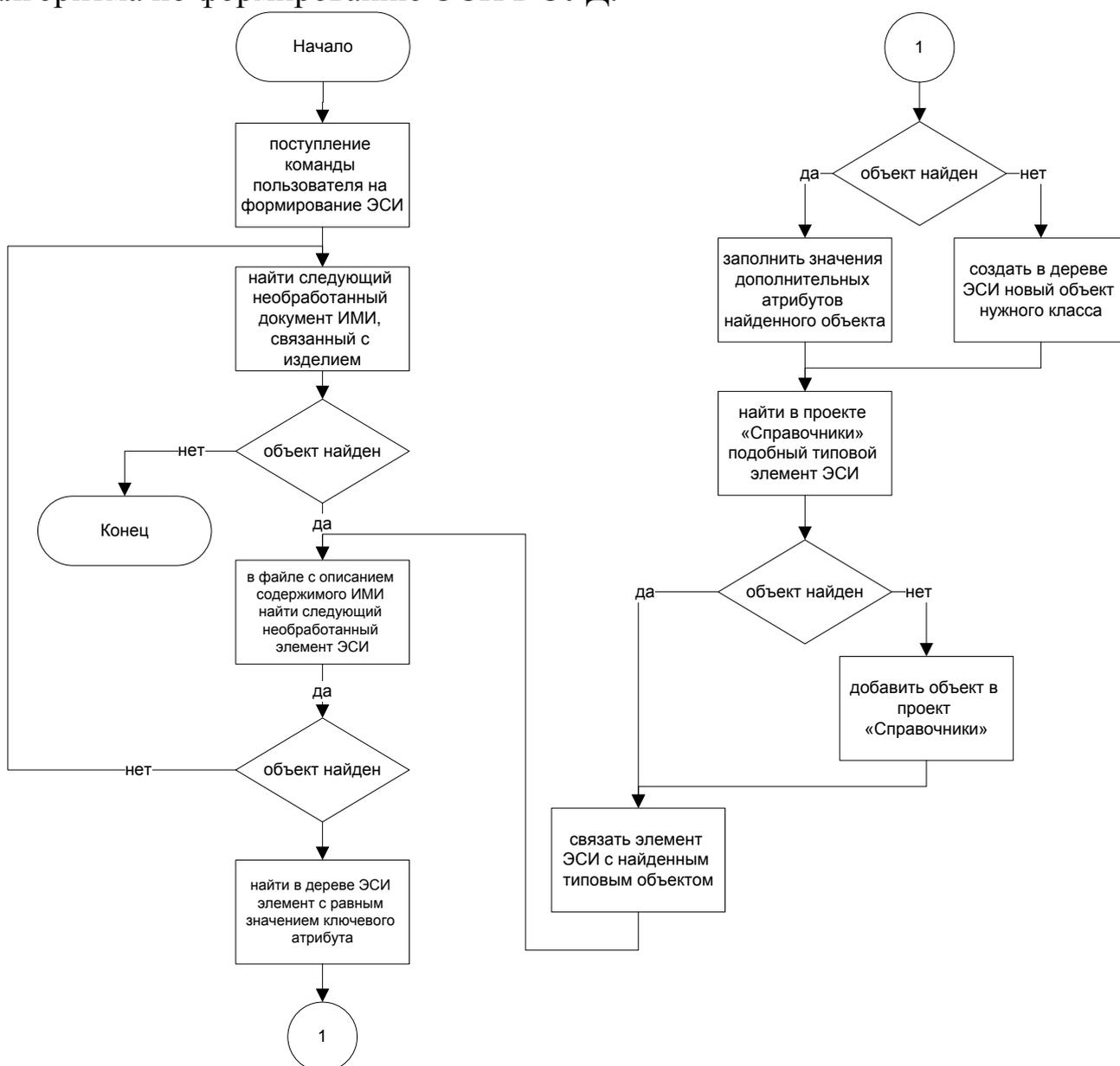


Рисунок 3.22 – Алгоритм формирования ЭСИ в СУД

Разработанный алгоритм формирования ЭСИ в СУД связан с другим, ранее рассмотренным компонентом предложенного метода построения ИИС на базе СУД и ее интеграций с САПР – структурой хранения типовых справочных данных в СУД. Таким образом, предложенный метод построения ИИС включает следующие взаимосвязанные составляющие:

- требования к взаимодействию САПР и СУД;
- структура хранения и варианты использования типовых справочных данных;
- алгоритм экспорта ИМИ из САПР в СУД;
- алгоритм формирования ЭСИ в СУД.

На рисунке 3.23 показана схема взаимосвязи перечисленных компонент в рамках единого метода с целью его комплексного представления.

Показанная на рисунке 3.23 схема взаимосвязи компонентов предложенного метода построения ИИС наглядно иллюстрирует развитие выявленных требований к взаимодействию САПР и СУД в виде формальных алгоритмов и моделей. Например, авторизация пользователей, относящаяся к категории «требования к безопасности», является неотъемлемой функцией алгоритма экспорта ИМИ из САПР в СУД. Аналогичным образом, требования к автоматизации процесса проектирования, такие как автоматическое формирование ЭСИ и КД в СУД, находят свое отражения в алгоритмах формирования ЭСИ в СУД и экспорта ИМИ из САПР. Выполняемое при этом автоматическое формирование новых версий объектов относится к категории требований к управлению версиями данных. При формировании ЭСИ в СУД реализуется требование к целостности, полноте и аутентичности данных, выражающееся в необходимости однозначного представления элемента ЭСИ, на основе множества ИМИ из разных САПР. Это влияет на структуру хранения типовых справочных данных в СУД, которая, в свою очередь, учитывается при формировании ЭСИ.

Таким образом, все ранее рассмотренные компоненты представленного метода построения ИИС на базе САПР и СУД, взаимосвязаны между собой в рамках единой концептуальной основы. Однако, для практической реализации разработанных моделей и алгоритмов в рамках апробации предложенного метода необходимо выполнить анализ существующих методов и технологий интеграции приложений. Затем, в зависимости от конкретной прикладной специфики, необходимо выбрать наиболее подходящий вариант.

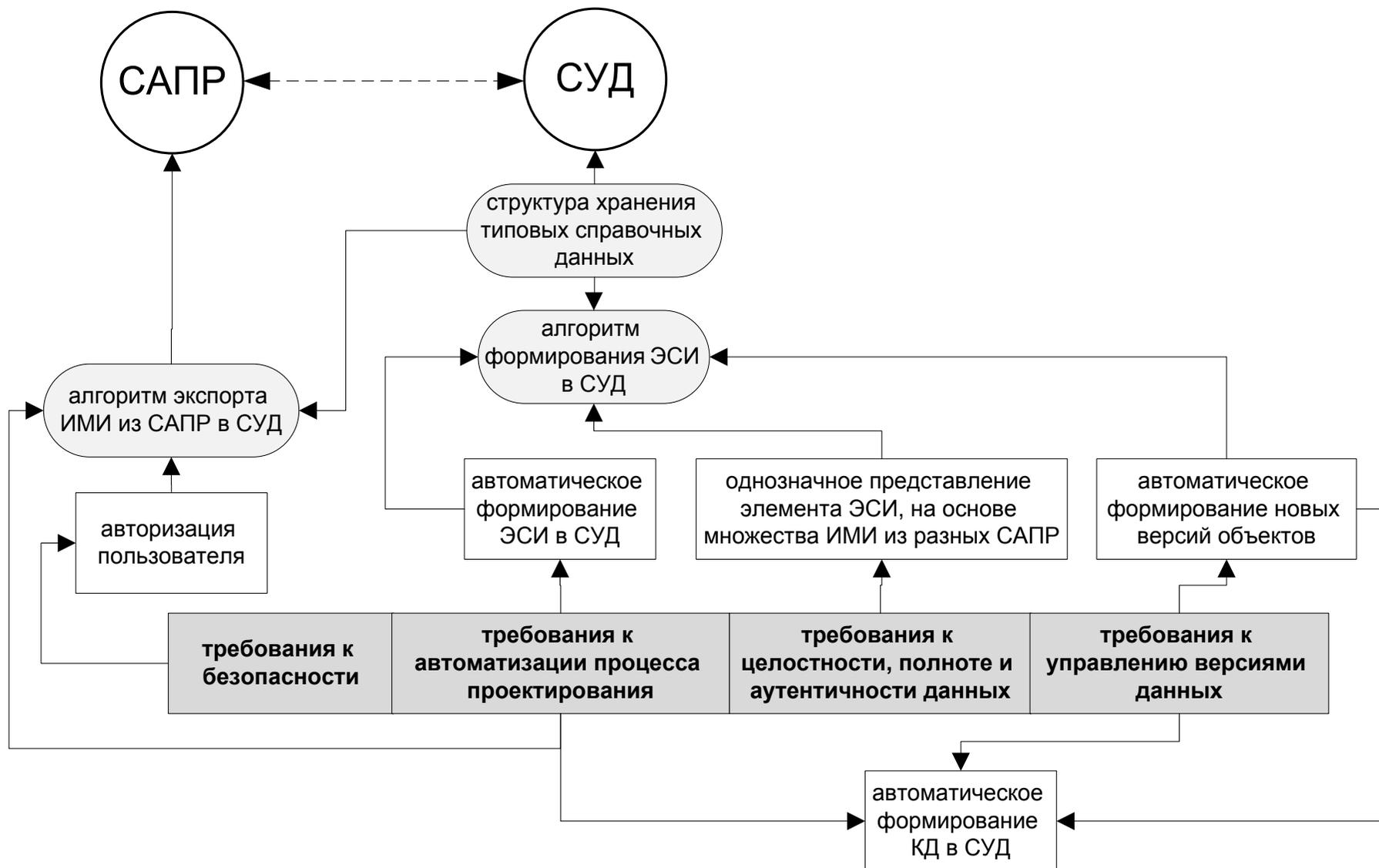


Рисунок 3.23 – Концептуальная схема взаимосвязи компонентов разработанного метода построения ИИС на базе СУД и ее интеграций с САПР

### **3.5 Анализ и поиск методов программной интеграции информационных систем**

#### **3.5.1 Общий обзор существующих методов интеграции приложений**

Согласно [133], ключевые различия информационных систем определяются следующими их архитектурными компонентами:

- схема или модель данных, которая лежит в основе интегрируемых приложений;
- технологический стек, на котором реализовано приложение (базовое программное обеспечение: СУБД, сервер приложений и т.д.);
- различие в моделях бизнес-процессов и в механизмах их реализации является главным препятствием для того, чтобы приложения стали частью единой области деятельности.

Кроме концептуальной стороны вопроса об интеграции ИС, следует также определить технические средства для решения данной задачи. На основании анализа работ В. Боркуса, Г. Хопа, Б. Вульфа, Г. Ладыженского [133-137], можно выделить несколько наиболее популярных на сегодня методов интеграции приложений:

- обмен файлами, в которые помещаются общие данные;
- общая база данных (БД), в которой сохраняется общая информация;
- удаленный вызов процедур в рамках систем обмена сообщениями для выполнения действий или обмена данными.

Выбор определенного метода и технологии из данного списка зависит от специфики связываемых ИС. Рассматривая возможность интеграции САПР и СУД с помощью метода общей базы данных, следует отметить, что большинство СУБД, обладает открытым программным интерфейсом. Поэтому возможно отслеживать значимые события в БД через механизм триггеров, перехватывать их, извлекать данные, ассоциированные с этими событиями, упаковывать их в стандартный формат, например, язык разметки XML, и передавать в транспортную среду предприятия, построенную, к примеру, на основе IBM MQSeries или Oracle Advanced Queuing. Такой функциональностью обладает технологический адаптер к базе данных, который является основным средством интеграции «закрытых» ИС. На основе адаптера можно создавать приложения, перебрасывающие блоки данных из БД одной ИС в БД другой [133]. Однако, подобное решение, подобно обмену файлами между приложениями, не является полноценной интеграцией двух разных ИС, поскольку предлагает лишь фактический перенос данных из одной ИС в другой и не поддерживает процессной связи между функциями разных приложений. Таким образом, можно

сделать вывод, что обмен файлами и общая БД также не уместны из-за различных схем данных и моделей бизнес-процессов САПР и СУД, интегрируемых в рамках ИИС.

На основе анализа разработанных моделей и алгоритмов управления ЖЦ объектов проектирования в рамках ИИС, включая алгоритмы экспорта ИМИ из САПР в СУД, можно сделать вывод, что взаимодействие САПР и СУД фактически представляет собой удаленный вызов процедур одного приложения в контексте другого. Поэтому следует подробнее рассмотреть данный метод в рамках задачи интеграции САПР и СУД.

### **3.5.2 Обзор технологий интеграции приложений методом удаленного вызова процедур**

Под термином «удаленный вызов процедур» понимают класс технологий, которые позволяют компьютерным программам вызывать функции или процедуры в другом адресном пространстве. Поэтому данный метод целесообразно использовать для интеграции разных ИС. Согласно [136], наибольшая эффективность использования метода удаленного вызова процедур достигается в тех приложениях, в которых существует интерактивная связь между удаленными компонентами с небольшим временем ответов и относительно малым количеством передаваемых данных. Характерными чертами метода вызова удаленных процедур являются:

- асимметричность (одна из взаимодействующих сторон является инициатором);
- синхронность (выполнение вызывающей процедуры приостанавливается с момента выдачи запроса и возобновляется только после возврата из вызываемой процедуры).

Распределенные системы доступа к объектам и системы удаленного вызова процедур называют системами обмена сообщениями. Сообщение реализуется путём посылки пакетов информации получателю (актору, процессу, потоку, сокету и т. д.). Передаваемой информацией могут являться сигналы, данные, а также удаленные вызовы процедур.

Интегрируемые ИС представляют собой независимые сущности, способные к взаимной координации по принципу слабой связи. Обмен сообщениями позволяет вызывающему приложению отправить информацию и вернуться к выполнению текущей задачи, переложив всю ответственность за доставку данных на систему обмена сообщениями. При необходимости вызывающее приложение может быть уведомлено о результате доставки информации с помощью функции обратного вызова [137].

В настоящее время выделяют следующие наиболее распространенные технологии реализации метода удаленного вызова процедур в рамках систем обмена сообщениями:

- компонентная объектная модель Component Object Model (COM), разработанная корпорацией Microsoft;
- общая архитектура брокеров объектных запросов Common Object Request Broker Architecture (CORBA);
- веб-сервисы, например, ESB (Enterprise Service Bus, сервисная шина предприятия) и SOA (Service Oriented Architecture, сервис-ориентированная архитектура).

Все вышеуказанные технологии включают сетевой протокол для обмена данными в режиме клиент-сервер и язык сериализации объектов (перевода данных в последовательность битов для передачи их по сети и сохранения в файлы). В основе перечисленных технологий лежит парадигма компонентно-ориентированного программирования, которая возникла в 90-е гг. XX в. с целью развития объектно-ориентированного подхода в сторону повышения надежности больших программных комплексов путем устранения проблемы хрупких базовых классов. Выделяют следующие ключевые характеристики понятия «компонент»:

- компонент – это независимый модуль программного кода, предназначенный для повторного использования и развертывания;
- компонент может содержать «множественные классы»;
- компонент, как правило, не зависит от конкретного языка программирования.

Идея Н.Вирта, в том, что компонент компилируется отдельно от других, а на стадии выполнения необходимые компоненты подключаются динамически, получила развитие в работе Б.Мейера, который предложил единое взаимодействие между вызываемым и вызывающим компонентами. Эти исследования воплотились в виде архитектурных решений, таких как, например, CORBA, COM, SOAP и протокола JAVA [138].

Выбор конкретной архитектуры зависит от специфики интегрируемых ИС, например, в случае веб-платформенных приложений наиболее эффективным будет применение протоколов SOA, JAVA и т.д. Однако, если сопрягаемые ИС функционируют как настольные приложения (что часто распространено среди САПР и СУД), наиболее уместным будет их интеграция на основе взаимодействующих компонент, каждая из которых может использоваться во многих ИС одновременно. Эту возможность позволяют реализовать компонентно-объектные технологии CORBA и COM. Несмотря на сходные базовые

принципы, они имеют множество различий в деталях реализации. Поэтому для выбора конкретной технологии интеграции САПР и СУД в рамках метода удаленного вызова процедур, целесообразно рассмотреть объектно-компонентные технологии CORBA и COM более подробно.

### **3.5.3 Выбор объектно-компонентной технологии интеграции приложений методом удаленного вызова процедур**

На основании изученных источников литературы [138-141], которые описывают объектно-компонентные технологии CORBA и COM, выделены следующие их сходства и различия:

- и CORBA, и COM позиционируются как универсальные клиент-серверные технологии создания распределенных систем;
- CORBA предназначена для распределенных систем в гетерогенных средах, тогда как COM создавалась фирмой Microsoft как средство взаимодействия приложений ОС Windows и ее составных частей, функционирующих на одном компьютере, с последующим развитием для использования в пределах локальной сети;
- и CORBA, и COM поддерживают множество языков программирования: Visual Basic, C++, Delphi, Java, C# и многие другие, перечень которых постоянно расширяется;
- и CORBA, и COM реализуют примерно одинаковый и достаточно высокий уровень абстракций, т.е. все вопросы низкого уровня, например, взаимодействие с операционной системой или сетевыми средствами, остаются за рамками прикладного программирования;
- CORBA является развивающейся концепцией с различными вариантами реализаций, тогда как COM предоставляет собой сформированную методологию и конкретные средства для интеграции разных приложений в рамках MS Windows.

Резюмируя сходства и различия COM и CORBA, можно сделать вывод, что выбор конкретной технологии определяется гетерогенностью среды. Согласно [69, 71, 133-137], в связи с широким распространением операционной системы (ОС) Microsoft Windows в качестве основной программной платформы рабочих станций на российских предприятиях, большинство САПР и СУД функционируют под управлением этой ОС. Поэтому далее рассмотрены средства COM-технологии для примера построения ИИС на базе САПР и СУД, функционирующих под управлением ОС Microsoft Windows.

### 3.5.4 Средства реализации COM-технологии

В качестве универсального способа взаимодействия приложений COM-технология начала использоваться в 1993 г., с появлением OLE (от англ. Object Linking and Embedding – связывание и встраивание объектов) версии 2.0. На сегодняшний момент можно выделить следующие средства реализации COM-технологий:

- базы данных серверных объектов (библиотеки типов), которые могут быть импортированы для анализа структуры серверов COM;
- универсальная технология доступа к базам данных - OLE DB/ADO;
- универсальный протокол обмена между клиентами и серверами;
- спецификации составных документов ActiveDoc;
- объектный монитор транзакций Microsoft Transaction Server (MTS);
- компонентная модель: OLE, ActiveX, COM.

Основным понятием, которым оперирует стандарт COM, является COM-компонент. Теоретически для создания COM-компонентов может использоваться любой язык, и сами COM-компоненты могут использоваться большим числом языков и инструментов, например, C#, Java, Microsoft Visual C++, Microsoft Visual Basic, Delphi, PowerBuilder и Micro Focus COBOL. Такая независимость от языка позволяет также выполнять быстрое прототипирование: вначале компоненты могут быть разработаны на языке высокого уровня, таком как Microsoft Visual Basic, а позже - на другом языке, таком как C#, C++ или Java, в котором, например, на высоком уровне реализована поддержка многопоточности.

Таким образом, технология COM не зависит от языка программирования и исполняющей среды. Программы, построенные на стандарте COM, фактически не являются автономными программами, а представляют собой набор взаимодействующих между собой COM-компонентов. Каждый компонент имеет уникальный идентификатор (GUID, от англ. Globally Unique Identifier) и может одновременно использоваться многими программами. Компонент взаимодействует с другими программами через COM-интерфейсы, которые представляют собой наборы абстрактных функций и свойств. При этом каждый COM-компонент должен, как минимум, поддерживать стандартный интерфейс IUnknown, предоставляющий базовые средства для работы с компонентом.

COM-объекты помещаются в файлы следующих видов:

- динамически подключаемая библиотека DLL (от англ. dynamic-link library);

- исполняемый файл EXE;
- OCX-модули, которые содержат элементы управления ActiveX.

Файлы, в которые помещаются СОМ-объекты, представляют собой двоичный код функций программного приложения, сгруппированных по контексту применения. Стандарт СОМ определяет взаимодействие компонентов с их клиентами. Это осуществляется таким образом, чтобы клиент и компонент могли соединяться без использования промежуточных элементов и клиент мог вызывать методы компонента.

### **3.5.5 Интеграция информационных систем средствами СОМ-технологии**

В современных ОС процессы изолированы друг от друга, поэтому клиент, которому нужно связаться с компонентом другого объекта, не может вызвать компонент напрямую, а должен использовать некоторую форму связи между процессами, предусмотренную ОС. СОМ организует подобное соединение в полностью прозрачной манере: он перехватывает вызовы со стороны клиента и адресует их компоненту другого процесса. На основании анализа работ Чистякова В.Ю., Цимбала А., Оберга Р.Дж. [138-141], можно сделать вывод, что объект СОМ – это зарегистрированная в ОС сущность, имеющая состояние и методы доступа, позволяющие изменять это состояние.

Таким образом, СОМ-объекты – это набор готовых классов, процедур, функций, структур и констант, предоставляемых приложением для использования во внешних программных продуктах. Такое определение соответствует термину интерфейса программирования приложений, называемого API (от англ. Application Programming Interface) [142]. Представить API приложения в виде набора СОМ-объектов позволяют специальные инструменты: встроенные внутрь приложения редакторы программного кода или внешние среды разработки, IDE (от англ. Integrated Development Environment).

Итак, кроме того, что СОМ-объект должен быть помещен в файл одного из вышеуказанных форматов (DLL, EXE или OCX), он должен быть особым образом зарегистрирован на компьютере, где его предполагается использовать. Для ОС Windows регистрация СОМ-объекта заключается в его занесении в системный реестр, иерархическую древовидную базу данных параметров и настроек [143]. API операционной системы Windows предоставляет базовые функции, позволяющие использовать СОМ-объекты. Под регистрацией СОМ-объекта понимается создание в системном реестре ОС Windows ветки с записями, которые описывают компонент. Каждый класс в компоненте

имеет уникальный идентификатор, под которым он регистрируется в системном реестре (рис. 3.24).

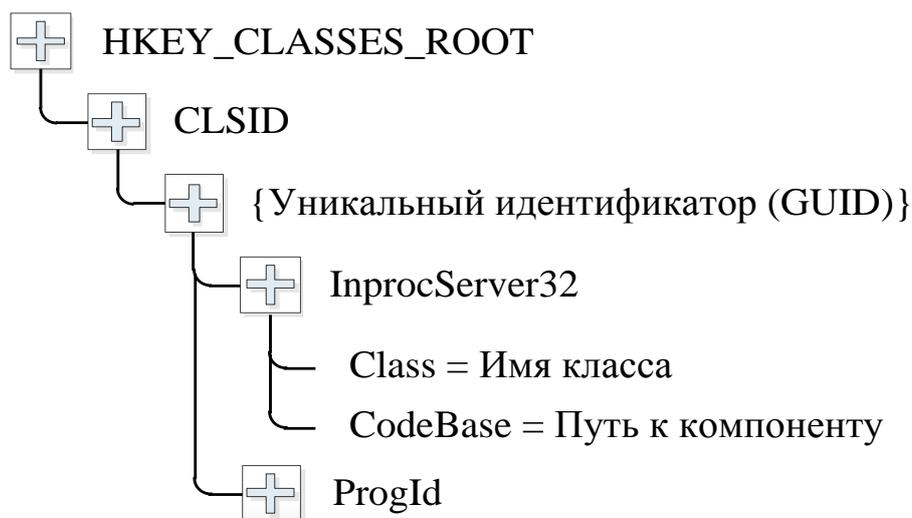


Рисунок 3.24 – Шаблон записи в системном реестре ОС

На рисунке 3.23 представлены только базовые ключи, которые описывают компонент Class – содержит в себе название компонента и наименование класса, а также CodeBase – который содержит путь к компоненту на жестком диске компьютера.

Некоторые приложения автоматически регистрируют свои API в виде COM-объектов в системном реестре. При отсутствии автоматической регистрации COM-объектов приложения ее можно выполнить самостоятельно с помощью внутренних или внешних сред разработки ПО. По умолчанию файл, в который помещен COM-объект, регистрируется как внутрипроцессный сервер. Это приводит к тому, что файл будет загружаться внутрь каждого процесса, который попытается создать находящийся в нем объект. Чтобы с объектами из этого файла можно было работать по сети, необходимо зарегистрировать его в мониторе транзакций MTS или использовать распределенную версию стандарта, Distributed COM (DCOM). DCOM развивает многокомпонентную модель COM до уровня связи между объектами различных компьютеров по локальной сети или через Интернет, включая способ создания удаленного объекта, протокол вызова его методов и механизмы обеспечения безопасного доступа к нему.

Подводя итог описанию COM-технологии, можно сделать вывод, что ее использование при построении ИИС на базе интеграций САПР и СУД является целесообразным, если интегрируемые ИС являются настольными приложениями, функционирующими под управлением ОС Microsoft Windows.

По результатам выполненного обзора COM-технологии, целесообразно предположение, что описания интерфейсных функций

при интеграции СУД и САПР в рамках ИИС, могут быть объединены в DLL-библиотеку. Полученная DLL-библиотека должна быть зарегистрирована в системном реестре ОС Windows, с целью дальнейшего использования в качестве СОМ-объекта для интеграции приложений.

Таким образом, интеграция САПР и СУД методом удаленного вызова процедур с применением технологии СОМ возможна при выполнении следующих условий:

Наличие возможностей вызова функций одного приложения непосредственно из другого (как правило, такое программное сопряжение реализуется разработчиками информационных систем в рамках единой программной платформы и не охватывает всех приложений, которые необходимо связать между собой согласно специфике отрасли или предприятия).

Интегрируемые ИС должны поддерживать технологии расширения их функциональных возможностей, поэтому необходимо проверить наличие внутреннего редактора программного кода и/или возможность обращения к СОМ-объектам рассматриваемого приложения из внешней IDE.

При выполнении вышеуказанных условий, интеграция САПР и СУД методом удаленного вызова процедур с применением СОМ-технологии представляет собой целый комплекс мероприятий, методику выполнения которого предлагается изложить в виде следующей логической последовательности этапов:

1. Проверить наличие СОМ-объектов в системном реестре, которые описывают функции связываемых приложений.
2. При отсутствии СОМ-объектов интегрируемых приложений в системном реестре, следует в среде встроенного редактора программного кода или внешней IDE описать необходимые функции API исследуемого приложения в виде DLL-библиотеки.
3. Далее, с применением системных утилит, полученные на предыдущем шаге DLL-библиотеки следует зарегистрировать в качестве СОМ-объектов в системном реестре.
4. С использованием программного обеспечения интеграции САПР и СУД в рамках ИИС выполняется обращение к зарегистрированным в системном реестре СОМ-объектам интегрируемых приложений с целью вызова их конкретных функций.

### **3.6 Резюме по разработке алгоритмических решений**

В данной главе выполнена разработка алгоритмического обеспечения для управления ЖЦ объектов проектирования в рамках ИИС. В результате проведенных исследований:

- выполнено формальное описание разработанной модели динамики изменения стадий ЖЦ объектов проектирования в виде UML-диаграмм состояний объектов и UML-диаграмм деятельности, которые показывают структуру действий пользователей ИИС;
- работоспособность ранее разработанной методики получения формальных моделей, описывающих ЖЦ разнотипных взаимозависимых объектов, проверена на процессе согласования документов в ИИС, предусматривающем изменение состояний разных сущностей при взаимодействии нескольких пользователей;
- разработан набор оригинальных алгоритмов параллельного и последовательного согласования документов, отличающийся использованием положений теории распараллеливания и синхронизации операций для динамического формирования заданий на согласование;
- обосновано использование типовых архитектурных приемов в виде паттернов «Состояние» и «Наблюдатель» для разработки программного обеспечения управления ЖЦ объектов проектирования в ИИС;
- предложена структура хранения типовых справочных данных и варианты их использования в ИИС;
- предложен метод построения интегрированной информационной среды на базе системы управления данными и ее интеграций с системами автоматизированного проектирования, представленный правилами взаимодействия программных средств и оригинальными алгоритмами их реализации, включая автоматическое заполнение справочных каталогов при построении электронной структуры изделия.
- на основании проведенного анализа существующих методов программной интеграции ИС, обоснован выбор СОМ-технологии для организации взаимодействия САПР и СУД, функционирующих в качестве настольных приложений под управлением ОС Microsoft Windows.

Все вышеперечисленные результаты носят общий характер и могут быть использованы для построения ИИС управления ЖЦ объектов

проектирования при разработке высокотехнологичной продукции с длительным сроком активного существования в различных прикладных отраслях промышленности. Полученные модели и алгоритмы положены в основу прикладного программного обеспечения ИИС космического приборостроения на базе СУД Enovia SmarTeam и ее интеграций с ECAD-САПР Altium Designer и MCAD-САПР SolidWorks. Практическое внедрение разработанных положений настоящего исследования используется для информационного сопровождения процесса проектирования бортовой РЭА КА с помощью внедрения PLM-технологий в отделении проектирования и испытаний бортовой РЭА ОАО «Информационные спутниковые системы» имени академика М.Ф. Решётнева.

#### **4 РЕАЛИЗАЦИЯ РАЗРАБОТАННЫХ МОДЕЛЕЙ И АЛГОРИТМОВ В ВИДЕ КОМПЛЕКСА ПРОГРАММНЫХ РЕШЕНИЙ**

Задача практической реализации предложенных моделей и алгоритмов в виде комплекса программных решений носит прикладной характер, поэтому ее целесообразно решать на примере конкретного набора информационных систем, которые используются в рассматриваемой предметной области. Для этого необходимо произвести выбор предприятия, которое занимается проектированием высокотехнологичной продукции с длительным сроком активного существования. Одним из таких предприятий является ОАО «Информационные спутниковые системы» имени академика М.Ф. Решетнева (далее ОАО «ИСС»), расположенное в г. Железногорске Красноярского края.

Данное предприятие входит в состав госкорпорации «Роскосмос» и является одним из ведущих предприятий российской космической промышленности. Поскольку деятельность ОАО «ИСС» охватывает полный цикл создания космических комплексов, в рамках данного предприятия функционирует отделение проектирования и испытаний бортовой РЭА. Основными задачами данного подразделения являются следующие работы по проектированию и испытанию бортовой РЭА КА:

- разработка схем и конструкций бортовой РЭА;
- моделирование схем и разработка программно-методической документации испытаний КА;
- обеспечение технологической подготовки производства печатных плат (ПП) РЭА, выпуск технологических носителей информации для изготовления и контроля ПП;
- сопровождение производства РЭА, бортовых и наземных кабельных систем;
- проведение НЭО разрабатываемых приборов и их составных частей;
- проведение унификации схемно-технических решений и конструкции РЭА, выпуск КД;
- выбор и отработка новых схемно-конструкторских решений, комплектующих элементов, материалов, технологий РЭА и кабелей;
- автоматизация проектирования разрабатываемой продукции, создание сквозных систем проектирования, изготовление и испытание РЭА и кабелей.

В данной главе описана программная реализация всех разработанных автором настоящей работы моделей и алгоритмов автоматизированного управления жизненным циклом разнотипных взаимозависимых

объектов проектирования в рамках ИИС на примере комплекса программных решений для процессов разработки бортовой РЭА КА в ОАО «ИСС». Работа выполнена в порядке реализации постановления № 218 Правительства РФ от 9.04.2010 г. «О мерах государственной поддержки развития кооперации российских высших учебных заведений и организаций, реализующих комплексные проекты по созданию высокотехнологичного производства», и договора № 13.G25.31.0017 от 07.09.2010 между ОАО «ИСС» и Минобрнауки РФ.

Апробация комплекса программных решений, полученного на основе разработанных моделей и алгоритмов, а также оценка практической значимости настоящей работы были проведены в рамках отделения проектирования и испытаний бортовой РЭА ОАО «ИСС».

В качестве основных результатов проведенной апробации можно выделить комплекс программных решений, разработанный на основе предложенных моделей и алгоритмов интеграции информационных систем, а также сценарии взаимодействия пользователей ИИС согласно их ролям в процессах проектирования бортовой РЭА. Фактически, данные результаты являются основой для разработки документов, регламентирующих работу сотрудников предприятия с точки зрения использования PLM-технологий.

#### **4.1 Постановка прикладных задач предметной области на примере процесса проектирования бортовой РЭА**

##### **4.1.1 Специфика процессов проектирования бортовой РЭА**

По технологии проектирования бортовой РЭА этап электрического проектирования, на котором решаются задачи схемотехнического и радиотехнического характера, предшествует этапу механического моделирования, на котором рассматриваются конструктивные и теплотехнические вопросы.

Электрическое проектирование представляет собой разработку электрических схем и печатных плат РЭС второго и последующего уровней по [21]. Результатом электрического проектирования является один или несколько файлов информационной модели разрабатываемого объекта, которая описывает его электрическую принципиальную схему или топологию ПП. В настоящее время одним из наиболее часто используемых программных средств для электрического проектирования изделий является ECAD-САПР Altium Designer.

Трехмерное представление (3D-модели) элементов изделия используются для разработки конструкции – механической информационной модели проектируемого объекта. Сегодня существует множество программных продуктов для трехмерного моделирования, наиболее широко распространенными среди которых являются САТИА,

Компас-3D, Pro/ENGINEER, AutoCAD, SolidWorks. В результате проектирования в MCAD-САПР получается 3D-модель изделия. При этом возникает необходимость формирования ЭСИ с учетом двойственного представления ЭРИ: с точки зрения электрического проектирования ЭРИ представляет собой единый объект, тогда как его 3D-модель – это сборка из нескольких компонентов (корпус, выводы и т.д.). Однако, согласно специфике предметной области, ЭРИ должно являться объектом типа «Прочее изделие», а не сборочной единицей. В космическом приборостроении не все составные части изделия подлежат электрическому моделированию, некоторые детали создаются только на этапе механического проектирования, как, например, элементы крепления конструкции. Моделирование таких объектов выполняется только в MCAD-САПР.

Таким образом, возникает задача целостного описания элемента ЭРИ в СУД, которая решается средствами интеграции СУД с ECAD- и MCAD-САПР, используемыми при проектировании бортовой РЭА. В свою очередь, совокупность ЭРИ представляет собой структуру изделия, которая является основой конструкторской документации. Интеграция САПР и СУД в рамках ИИС позволит реализовать создание и изменение объектов проектирования, а также управление их жизненным циклом в зависимости друг от друга.

Применяя основные положения концепции PLM-технологий к технологии проектирования бортовой РЭА, предлагается рассмотреть использование САПР и СУД в рамках ИИС в виде следующей совокупности процессов:

- разработка электрических схем и печатных плат РЭС второго и последующего уровней по [21] осуществляется с помощью ECAD-САПР. При этом используется БД СУД, в которой хранятся электрические характеристики ЭРИ, включая условное графическое обозначение (УГО) на схемах. Также идет получение данных из БД ECAD-САПР, в которой хранится информация о параметрах проектируемых объектов с точки зрения электрического моделирования. Результатом электрического проектирования в ECAD-САПР является один или несколько файлов информационной модели разрабатываемого объекта, которая описывает его электрическую принципиальную схему или топологию ПП;
- с точки зрения концепции PLM-технологий, СУД является центральным ядром ИИС, в котором аккумулируется полная целостная информация об объекте. Поэтому результаты разработки изделия из ECAD-САПР, включая информацию о применяемых ЭРИ, загружаются в БД СУД. В свою очередь, все

ЭРИ, помимо своих электрических характеристик, моделирование которых осуществляется с помощью ECAD-САПР, имеют трехмерное представление, пространственные параметры которого являются объектом моделирования в MCAD-САПР. Таким образом, в БД СУД должны храниться также 3D-модели элементов изделия, необходимые на этапе проектирования конструкции;

- трехмерные представления (3D-модели) элементов изделия используются для разработки механической информационной модели проектируемого объекта. В результате проектирования в MCAD-САПР получается 3D-модель изделия.

Таким образом, использование СУД в качестве центрального ядра ИИС также позволяет решить проблему структурированного хранения проектных данных настоящих и предшествующих работ. Для этого возможно применение предложенной структуры хранения типовых объектов (справочных данных) в СУД. Подобное использование и модификация их в текущих проектах позволит сократить объем информации, хранящейся в СУД и предотвратить дублирование данных в рамках ИИС.

#### **4.1.2 Программные компоненты ИИС проектирования бортовой РЭА**

Задача составления перечня программных компонент ИИС и определения их взаимосвязей носит прикладной характер, поэтому ее целесообразно решать на примере конкретного набора информационных систем, которые используются в рассматриваемой предметной области. Для выявления особенностей использования САПР и СУД в процессах проектирования в рамках ИИС на примере ОАО «ИСС» следует более детально рассмотреть технологию разработки изделий с применением указанных программных компонент ИИС.

Проведенный анализ деятельности отделения проектирования и испытаний бортовой РЭА ОАО «ИСС» позволяет сделать вывод, что основными программными продуктами, которые используются при проектировании изделий, являются следующие:

- ECAD-САПР Altium Designer R9 для электрического проектирования изделия (разработка принципиальной схемы, топологии ПП);
- MCAD-САПР SolidWorks 2009 SP9 для механического проектирования изделия (разработка конструкции);
- СУД Enovia SmarTeam V5R19 для структурированного хранения и управления информацией об изделии, а также распределения потоков работ.

На основании выполненного анализа существующих широко распространенных PLM-решений (таких, как продукты компаний АСКОН, Dassault Systems, Лоция Софт, РТС и т.д. [69]) сделан вывод, что вышеперечисленные программные продукты широко применяются для проектирования бортовой РЭА также и на многих других предприятиях космической и авиационной отраслей промышленности (например, ФГУП «ММПП «САЛЮТ», ОАО «ОКБ Сухого», ОАО «КНААПО им. Ю.А. Гагарина», ОАО «НПП «Геофизика-Космос», ФГУП МОКБ «Марс», ОАО «УФ КБ Туполев»). Поэтому рассмотрение ECAD-САПР Altium Designer, MCAD-САПР SolidWorks и СУД Enovia SmartTeam в качестве программных компонент ИИС проектирования бортовой РЭА соответствует общему характеру положений работы, результаты которой могут быть адаптированы к специфике любого конкретного предприятия.

В соответствии с предложенной выше технологией проектирования РЭА в рамках ИИС (рис. 4.1., пункт 4.1.1), были развернуты автоматизированные рабочие места (АРМ) для следующих участников этого процесса:

- схемотехника, результатом работы которого является электрическая принципиальная схема изделия, разработанная в ECAD-САПР Altium Designer;
- конструктора, результатом работы которого является 3D-модель изделия, разработанная в MCAD-САПР SolidWorks;
- тополога, результатом работы которого является топология изделия – ПП, разработанная в ECAD-САПР Altium Designer.

Компонентный состав вышеуказанных АРМов в рамках клиент-серверной ИИС, построенной по предложенным в настоящей работе моделям и алгоритмам в отделении проектирования и испытаний бортовой РЭА ОАО «ИСС», представлен на рисунке 4.2 с помощью графических инструментов UML-диаграммы развертывания.

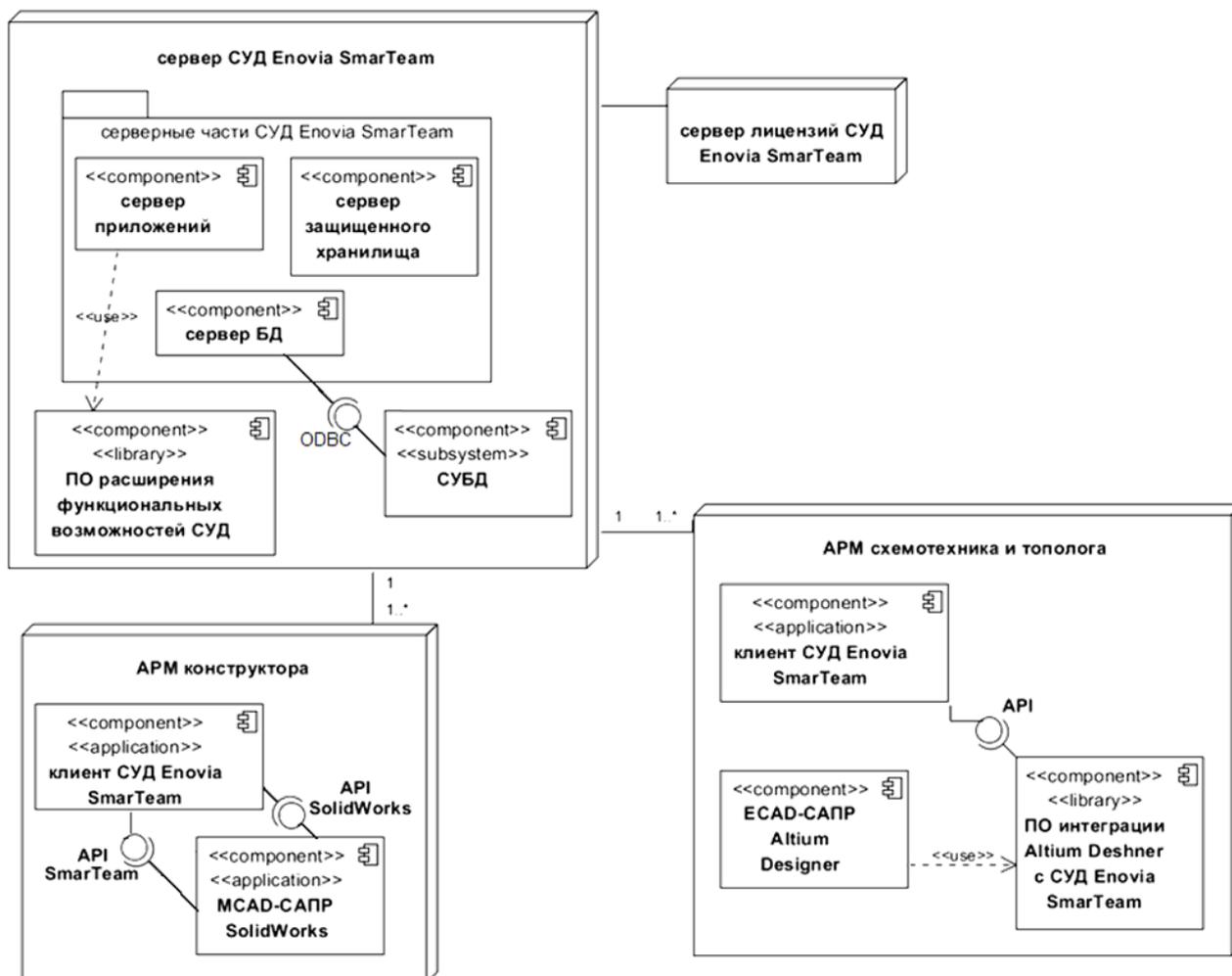


Рисунок 4.1 – UML-диаграмма развертывания и компонентов, иллюстрирующая программно-аппаратную архитектуру разработанного прототипа ИИС

Ядро СУД Enovia SmarTeam V5 функционирует в рамках клиент-серверной архитектуры, обеспечивающей работу с данными в распределенном сетевом режиме. Серверная архитектура ядра СУД Enovia SmarTeam состоит из 4-х компонентов, которые могут работать и взаимодействовать между собой как на одном физическом/виртуальном компьютере, так и быть физически распределенными:

- сервер приложений обеспечивает доступ и формирование всех данных для клиента, а также настройку СУД Enovia SmarTeam;
- сервер БД хранит в табличной форме и обеспечивает доступ ко всем данным системы;
- сервер защищенного хранилища выполняет роль электронного архива и хранит все файлы, зарегистрированные в системе, а также реализует контроль прав доступа к файлам и версию документов;
- сервер лицензий обеспечивает сетевое распределение лицензий между пользователями.

Представленная на рисунке 4.1 UML-диаграмма развертывания иллюстрирует следующие отношения между программными компонентами ИИС и аппаратными узлами их развертывания:

- серверные части (сервер приложений, БД и хранилища) СУД Enovia SmarTeam V5R19 развернуты на специально выделенном компьютере серверного типа под управлением MS Windows Server 2008 R2 Enterprise;
- серверная часть СУД Enovia SmarTeam V5R19 обеспечивает хранение данных с помощью СУБД Oracle 11g R2, дополнительные функциональные возможности СУД Enovia SmarTeam V5R19 (распараллеливание потоков работ, формирование ЭСИ, изменение стадий ЖЦ объектов проектирования), обеспечиваются разработанными программными скриптами;
- соединение с сервером лицензий ОАО «ИСС», физически расположенном на другом компьютере, выполняется по локальной сети с использованием сетевых протоколов;
- АРМ схемотехника и тополога под управлением ОС MS Windows включает клиентскую часть СУД Enovia SmarTeam V5R19, которая, с использованием сетевых протоколов локальной сети, обращается к серверу данной ИС с целью получения или отправки информации. Кроме того, АРМ схемотехника и тополога включает ECAD-САПР Altium Designer R9 и DLL-библиотеки, содержащие COM-объекты, которые обеспечивают интеграцию с СУД Enovia SmarTeam V5R19;
- аналогичным образом организовано АРМ конструктора под управлением ОС MS Windows, включая клиентскую часть СУД Enovia SmarTeam V5R19, которая, с использованием сетевых протоколов локальной сети, отправляет серверу данной ИС запросы с целью получения или отправки информации. Кроме того, АРМ конструктора включает MCAD-САПР SolidWorks 2009, которая связана с СУД Enovia SmarTeam V5R19 с помощью программных API-средств существующей интеграции между этими системами.

Взаимодействие клиентской и серверной частей организуется с помощью сетевых протоколов локальной сети и определяется физическими характеристиками каналов и средств передачи информации. Взаимодействие САПР и СУД, экземпляры которых присутствуют на АРМах пользователей ИИС, обеспечивается с помощью программной интеграции, особенности которой рассмотрены далее.

#### **4.1.3 Реализация ИИС проектирования бортовой РЭА КА на базе СУД Enovia SmarTeam и ее интеграций с САПР Altium Designer и SolidWorks**

Существует интеграция MCAD-САПР SolidWorks 2009 SP9 и СУД Enovia SmarTeam V5R19, предоставляемая компанией Dassault Systems, которой принадлежат права авторства на указанные ИС. На основе проведенного анализа существующей интеграции сделаны следующие выводы:

- клиентские части указанных систем функционируют на одном компьютере пользователя под управление ОС Microsoft Windows;
- взаимодействие между MCAD-САПР SolidWorks и СУД Enovia SmarTeam организовано с помощью технологии взаимосвязи их программных компонент, а также вспомогательных объектов промежуточного программного приложения SolidWorks To SmarTeam;
- наличие регистрации программных компонент MCAD-САПР SolidWorks и СУД Enovia SmarTeam, а также вспомогательных объектов промежуточного программного приложения SolidWorks To SmarTeam в реестре ОС позволяет осуществлять вызов необходимых функций указанных информационных систем.

Анализ сопоставления полученных выводов и описания СОМ-технологии, позволяет сказать, что существующая интеграция MCAD-САПР SolidWorks и СУД Enovia SmarTeam реализована с использованием методов и средств СОМ-технологии. Наличие системных атрибутов в модели данных СУД Enovia SmarTeam, (например, у классов «Документы» и «ЭСИ») позволяет связать объекты данных классов и создать элементы ЭСИ в СУД Enovia SmarTeam на основе 3D-моделей, импортированных из MCAD-САПР SolidWorks. Также наличие существующей интеграции СУД Enovia SmarTeam с MCAD-САПР SolidWorks включает следующие функциональные возможности:

- открытие в MCAD-САПР SolidWorks сборки (включая все входящие файлы деталей) из СУД Enovia SmarTeam;
- экспорт сборки (включая все входящие файлы деталей) из MCAD-САПР SolidWorks в СУД Enovia SmarTeam с прикреплением к конкретному проекту и изделию;
- просмотр файлов SolidWorks в СУД Enovia SmarTeam с помощью встроенного просмотрщика файлов eDrawing (без использования дополнительного ПО);

- формирование дерева элементов ЭСИ с помощью инструмента «Мастер копирования дерева». Для создания в дереве ЭСИ объектов разных классов по видам изделий согласно [22] (например, «Детали», «Стандартные изделия», «Прочие изделия», «Изделия из материала») выполняется проверка условия, которое устанавливается в специальной утилите Enovia SmarTeam по настройке интеграции с SolidWorks.

С использованием предложенного метода построения ИИС на базе САПР и СУД, а также методов и средств СОМ-технологии, в рамках проекта реализации ЕИП в отделении проектирования и испытаний бортовой РЭА ОАО «ИСС» было разработано ПО интеграции ЕСАД-САПР Altium Designer с СУД Enovia SmarTeam.

ЕСАД-САПР Altium Designer содержит различные редакторы (модули) для реализации функций сквозного проектирования электронных приборов на базе ПП (например, модуль разработки электрических схем, топологии ПП, аналитических расчетов и т.д.). Все редакторы обладают собственным API: редактор топологии ПП включает PCB API, редактор схем – Schematic API, редактор управления проектом Project Manager – Workspace Manager API. Функции программной платформы Design Explorer (DXP) описаны в DXP API. Все перечисленные API объединены в динамически подключаемую во время выполнения библиотеку Altium Designer Run Time Library, которая также включает специализированные классы и систему вызова процедур.

Каждый API содержит свою объектную модель, которая является иерархической системой интерфейсов, описывающих объект. Открытая архитектура DXP-платформы позволяет использовать преимущества объектно-ориентированных и клиент-серверных технологий. При работе со скриптами идет обращение пользователя к интерфейсам объекта с помощью DLL-библиотек. Функциональность каждого модуля может быть расширена через внутренний API этой ИС с использованием скриптовых языков, таких как, DelphiScript, EnableBasic, VisualBasic Script, JavaScript и др. Особенностью всех API системы Altium Designer является их «закрытость», т.е. возможность вызова функций только внутри данной ИС.

Поэтому было разработано приложение Altium To SmarTeam, которое является связующим звеном между ЕСАД-САПР Altium Designer и СУД Enovia SmarTeam. Вызов СОМ-объектов указанных ИС осуществляется в приложении Altium To SmarTeam, при этом также используется информация о внутреннем содержании ИМИ-файлов, разработанных в ЕСАД-САПР Altium Designer. Эта информация представляет собой данные об элементах ЭСИ и описание их электрических характеристик.

Сведения о проекте, схемах и ПП сохраняются в текстовый документ в JSON-формате.

ECAD-САПР Altium Designer позволяет создавать различные виды проектов, одним из них является скрипт-проект, который представляет собой набор инструкций, которые интерпретируются каждый раз при запуске системы. В проекте могут быть использованы различные прикладные программные интерфейсы для модификации проектируемых объектов в документах проекта.

В качестве языка программирования для обработки проектируемых объектов используется язык Visual Basic, на котором во внутренней среде разработки Altium Designer написан скрипт формирования JSON-файла с описанием содержимого ИМИ. После формирования JSON-файл преобразуется в объектную модель для повышения удобства работы с полученными данными. При этом каждая запись в файле становится объектом определенного класса, которые объединяются в коллекции и могут использоваться как вспомогательными объектами, которые используются при интеграции ECAD-САПР Altium Designer с СУД Enovia SmarTeam, так и непосредственно объектами СУД. Например, таким образом, согласно предложенному алгоритму формирования ЭСИ (рис. 3.17) в СУД Enovia SmarTeam выполняется заполнение значений атрибутов у объектов класса «Прочие изделия».

#### **4.1.4 Формирование ЭСИ в СУД СУД Enovia SmarTeam на основе ИМИ-файлов, разработанных в САПР Altium Designer и SolidWorks**

Согласно технологии проектирования РЭА [23, 111, **Ошибка! Источник ссылки не найден.-123**], описания таких составных частей проектируемого изделия, как его сборочные единицы, детали, стандартные и прочие изделия создаются на этапе конструкторского проектирования в MCAD-САПР SolidWorks. Поэтому для корректного определения типа объекта в дереве ЭСИ СУД Enovia SmarTeam была выполнена настройка соответствия компонентов 3D-модели изделия (объектов класса «Документ») и экземпляров классов «ЭСИ». Для этого была использована специальная утилита Enovia SmarTeam «Copy Mapping Tool». В таблице 4.1 показано настроенное с ее помощью соответствие классов «MCAD» в суперклассе «Документ» классам суперкласса «ЭСИ».

### Соответствие классов MCAD в суперклассе «Документ» классам суперкласса ЭСИ

Подкласс класса MCAD в суперклассе «Документ»	Расширения файла САПР	Соответствующий класс в суперклассе «ЭСИ»
SolidWorksAssembly	*.sldasm	Сборочная единица
SolidWorksStandartPart	*.sldpart	Стандартное изделие
SolidWorksPart	*.sldpart	Деталь
SolidWorksPart	*.sldpart	Прочее изделие
SolidWorksPart	*.sldpart	Изделие из материала

Анализ таблицы 4.1 показывает необходимость задания условия для создания объектов классов «Стандартное изделие», «Деталь» «Прочее изделие» и «Изделие из материала». Согласно движению потоков данных в процессах проектирования бортовой РЭА, 3D-модели некоторых элементов изделия, которые уже существуют в СУД Enovia SmarTeam, могут быть использованы для разработки нового изделия через операцию импорта из СУД Enovia SmarTeam в MCAD-САПР SolidWorks. Такие элементы хранятся в СУД Enovia SmarTeam в качестве типовых в проекте «Справочники» в виде объектов класса «SolidWorksStandartPart», включая их САПР-файлы с расширением \*.sldpart. При экспорте файлов с расширением \*.sldpart из MCAD-САПР SolidWorks в СУД Enovia SmarTeam они прикрепляются к объектам класса «SolidWorksPart». Данные настройки позволяют получить в дереве документов, связанных с изделием, структурированный набор объектов классов «SolidWorksStandartPart», «SolidWorksPart», «SolidWorksAssembly» (рис. 4.2).

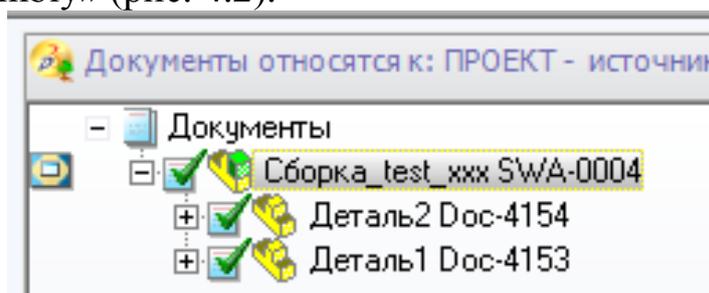


Рисунок 4.2 – Дерево документов опубликованной сборочной единицы SolidWorks

Таким образом, при использовании команды пользовательского интерфейса СУД Enovia SmarTeam «Мастер копирования дерева» из контекстного меню экземпляра класса SolidWorksAssembly в суперклассе «Документы» на основе данных из 3D-модели в дереве элементов ЭСИ создаются экземпляры классов «Сборочная единица», «Деталь» «Стандартное изделие», «Прочее изделие» (рис. 4.3).

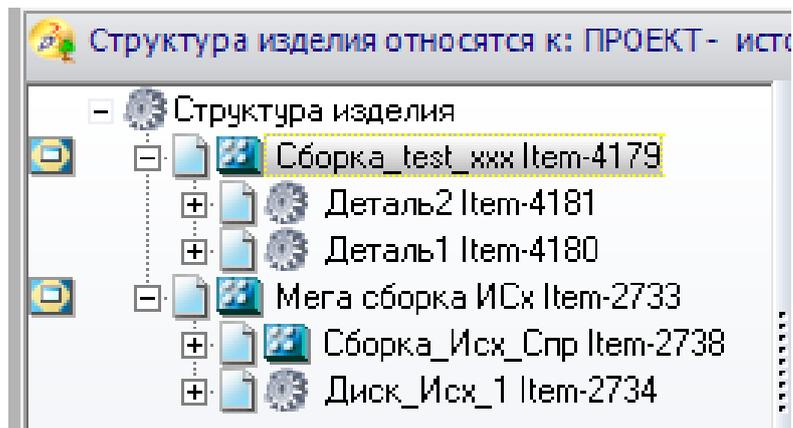


Рисунок 4.3 – Дерево элементов ЭСИ

При генерации в дереве элементов ЭСИ объектов, данные сущности связываются с файлами САПР, на основе которых они сгенерированы, связью спецификации «Specification Link». Таким образом, сохраняется зависимость элементов ЭСИ от соответствующих им файлов ИМИ (рис. 4.4).

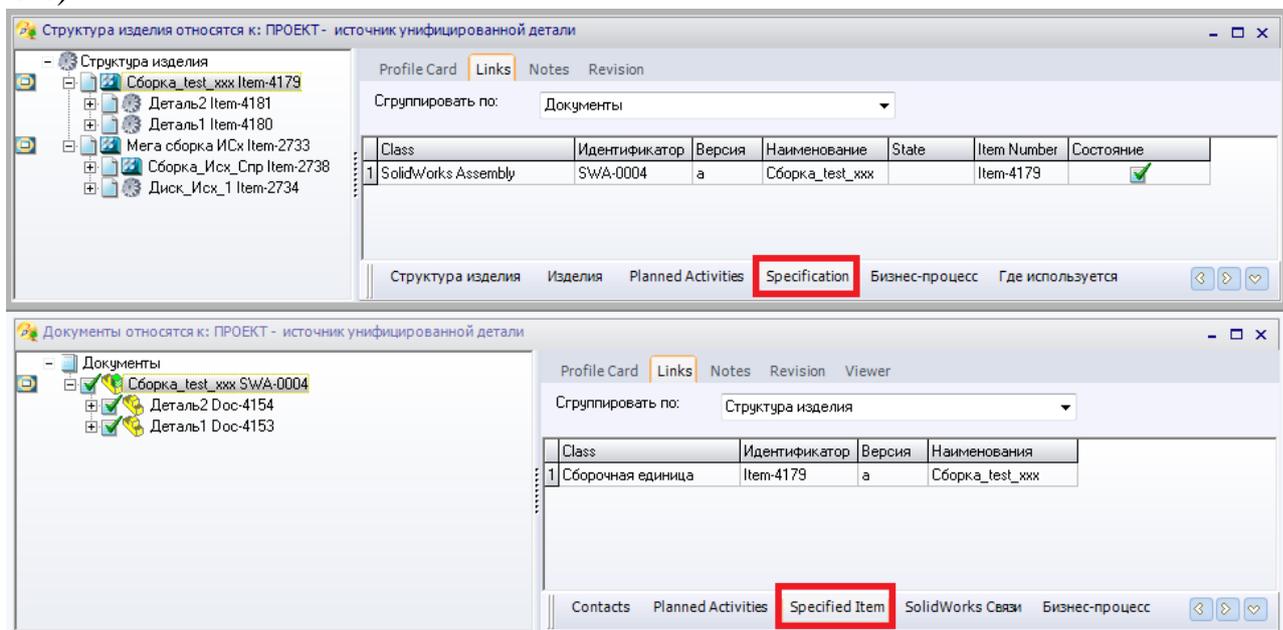


Рисунок 4.4 – Связь спецификации между объектами типов «ИМИ» и «ЭСИ»

#### 4.1.5 Обоснование необходимости разработки дополнительного программного обеспечения для управления ЖЦ объектов проектирования в СУД Enovia SmarTeam

СУД Enovia SmarTeam предоставляет достаточно широкий набор инструментов администратора в виде автономных приложений для настройки и конфигурирования данной системы в части разработки модели данных, определения хранилищ файлов, правил изменения ЖЦ элемента ЭСИ, а также компонентов пользовательского интерфейса. Однако, существующие инструменты не позволяют обеспечить решение

задач, специфичных для процессов управления ЖЦ объектов проектирования. Данные задачи и обоснование разработки инструмента их решения приведены в таблице 4.2.

Таблица 4.2

**Обоснование разработки программных решений для специфических задач управления ЖЦ объектов проектирования в СУД Enovia SmarTeam**

№	Задача управления ЖЦ объектов проектирования	Обоснование разработки программного решения
1	ЖЦ документа отличается от типового набора стадий ЖЦ и маршрута их изменения, предлагаемого в СУД Enovia SmarTeam	Существующие средства настройки СУД Enovia SmarTeam не позволяют задать условия автоматического изменения состояний жизненного цикла ИМИ и КД в зависимости от связанных с ним объектов проектирования других типов (ЭСИ, Изделие), поэтому для решения этой задачи необходимо разработать дополнительное программное обеспечение
2	Инструменты настройки ЖЦ объектов в СУД Enovia SmarTeam позволяют определять набор состояний ЖЦ и маршрут их изменения у элементов ЭСИ, но не предполагают наличия ЖЦ у объектов класса «Изделия»	Состояние ЖЦ изделия определяется состоянием ЖЦ его составных частей, элементов ЭСИ. Существующие средства администрирования СУД Enovia SmarTeam не позволяют задавать подобные правила управления ЖЦ объектов проектирования. Поэтому для решения этой задачи необходимо определить атрибут, значение которого характеризует состояние ЖЦ изделия и разработать дополнительное программное обеспечение для его автоматического заполнения
3	Технология разработки и согласования документов предполагает декомпозицию работы на набор одинаковых элементарных заданий для нескольких исполнителей, количество которых заранее неизвестно и может измениться в процессе работы	Согласно разработанным моделям и алгоритмам управления ЖЦ исследуемых объектов проектирования, результаты выполнения элементарных заданий по разработке и согласованию документов определяют состояние ЖЦ объектов проектирования других типов (ЭСИ, Изделие). Поэтому требуется динамическое формирование множества элементарных заданий и одновременное их распределение между исполнителями, а также синхронизация результата. Существующие средства администрирования СУД Enovia SmarTeam не позволяют задавать подобные правила управления потоками работ, поэтому для решения этой задачи необходимо разработать дополнительное программное обеспечение

Анализ таблицы 4.2 позволяет сделать вывод, что для реализации предлагаемых моделей и алгоритмов управления жизненным циклом объектов проектирования в рамках ИИС, необходимо разработать комплекс программных решений по расширению функциональных возможностей СУД Enovia SmarTeam. Рассматривая СУД Enovia

SmarTeam с точки зрения разработчика, необходимо обратить внимание на правила хранения и манипулирования информацией, а также инструменты настройки и администрирования данной ИС.

Начальным шагом практической реализации разработанных моделей и алгоритмов управления ЖЦ разнотипных взаимозависимых объектов на примере отрасли космического приборостроения была выполнена настройки модели данных СУД Enovia SmarTeam. С использованием специального инструмента администратора рассматриваемой СУД в объектную модель данных системы были внесены новые классы, а также изменены существующие, в частности, добавлена совокупность атрибутов, описывающих состояние ЖЦ исследуемых разнотипных объектов проектирования (табл. 2.7). В качестве исходной информации для модификации модели данных были использованы разработанные теоретико-множественные и графические модели, описывающие состав и взаимосвязи сущностей исследуемой предметной области (выражения (1)-(6), рис. 2.7).

## **4.2 Программная реализация функций управления жизненным циклом объектов проектирования в ИИС на базе интеграций СУД Enovia SmarTeam с САПР Altium Designer и SolidWorks**

### **4.2.1 Средства расширения функциональных возможностей СУД Enovia SmarTeam**

СУД Enovia SmarTeam предоставляет возможности создания собственных программ с помощью встроенного редактора, который поддерживает скриптовый язык SmartBasic, но не обладает полной типовой функциональностью современных интегрированных сред разработки ПО. Благодаря открытости API СУД Enovia SmarTeam для программной настройки этой ИС возможно использовать внешние среды разработки. Поэтому целесообразно выбрать специализированную среду разработки ПО, которая предоставляет широкие возможности разработки и отладки программных средств. Добавление дополнительной функциональности в СУД Enovia SmarTeam возможно следующими способами: назначение скриптов для пользовательских команд и для событий, возникающих при работе с объектами, файлами, авторизациями пользователей, а также импорте/экспорте данных. Назначение скриптов для событий в СУД Enovia SmarTeam осуществляется с помощью специальных приложений Enovia SmarTeam. Скрипты могут выполняться до события, после события или вместо него.

В состав СУД Enovia SmarTeam входит средство расширения стандартных возможностей в виде редактора программного кода для написания скриптов на языке Visual Basic Script. Однако,

функциональные возможности специализированных сред разработки шире, чем у редактора СУД Enovia SmarTeam. В связи с наличием открытого API СУД Enovia SmarTeam возможно использование внешних сред разработки для создания собственных DLL-библиотек на любом языке программирования. Поэтому встроенный редактор программного кода СУД Enovia SmarTeam целесообразно использовать только для вызовов функций DLL-библиотек. На рисунке 4.5 приведена UML-диаграмма компонентов, которая иллюстрирует связь между СУД Enovia SmarTeam и файлом DLL-библиотеки программных функций этой системы.

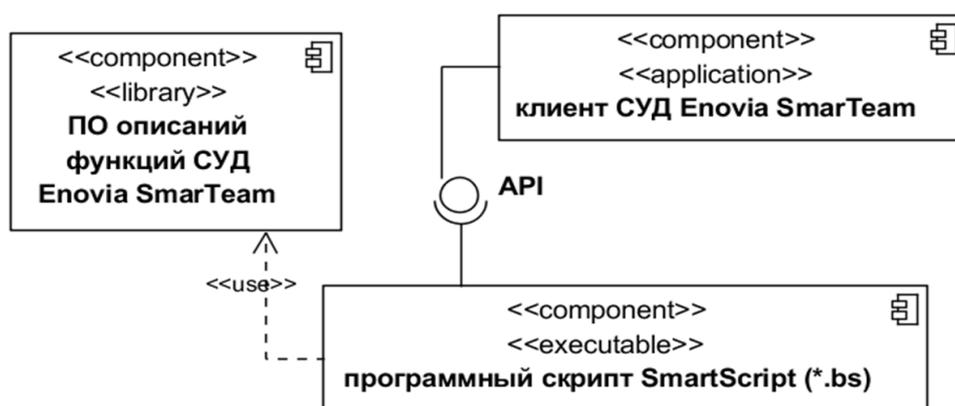


Рисунок 4.5 – Взаимосвязь программных компонент по расширению функциональных возможностей СУД Enovia SmarTeam

Через элементы графического интерфейса пользователя, такие, как, например, кнопки, пункты меню и т.д., происходит вызов программных скриптов, написанных на языке Visual Basic Script во встроенном редакторе программного кода СУД Enovia SmarTeam. В свою очередь, вызванный программный скрипт через API СУД обращается к файлу DLL-библиотеки, который находится на компьютере пользователя и содержит описание необходимых функций СУД. Выполнение вызванной функции возвращает результат в СУД Enovia SmarTeam. Таким образом, СУД Enovia SmarTeam имеет открытый API, что позволяет использовать уже существующие и зарегистрированные в реестре операционной системы COM-объекты, а также создавать новые.

СУБД системы Enovia SmarTeam (MS SQL или Oracle) реализует реляционную модель, однако, в приложении для настройки модели данных этой ИС, отображение структур хранения информации организовано в соответствии с объектно-ориентированным подходом. Характерными особенностями реализации этой методологии представления данных в СУД Enovia SmarTeam можно отметить следующие:

- типизация объектов осуществляется с помощью классов, с сохранением порядка иерархии (суперклассы, базовые, абстрактные и т.д.);
- класс описывается набором характеристик – атрибутов;
- объекты как одного, так и разных классов могут быть связаны между собой с помощью связей;
- связи также являются объектами класса «Связи» и описываются набором атрибутов;
- внутри одного класса объекты могут быть связаны связью типа «Иерархическая ссылка»;
- объекты разных классов могут быть связаны связью типа «Общая ссылка»;
- объекты классов «Документы» и «ЭСИ» могут быть связаны связью типа «Ссылка спецификации», которая показывает, что экземпляр класса «ЭСИ» был создан на основе экземпляра класса «Документы».

Вышеперечисленные положения объектно-ориентированного подхода, реализованы в СУД Enovia SmartTeam для повышения уровня абстракции с целью упрощения процедур настройки данной ИС. В конечном итоге манипулирование данными сводится к реляционным операциям над записями таблиц БД. Таким образом, в случае программной настройки СУД Enovia SmartTeam можно сочетать преимущества двух технологий разработки ПО: объектно-ориентированного программирования и SQL-запросов.

#### **4.2.2 Выбор среды разработки и языка программирования**

Как уже было отмечено ранее, СУД Enovia SmartTeam предоставляет возможности создания своих программ с помощью встроенного редактора, который поддерживает скриптовый язык SmartBasic. Однако наиболее рационально использовать полнофункциональную специализированную среду разработки ПО (IDE, от англ. Integrated Development Environment), которая предоставляет широкие возможности по проектированию, компиляции и отладке программ. Ввиду большого количества программных продуктов данной категории, выбор среды разработки целесообразно производить на основе соответствия следующим критериям:

- поддержка нескольких языков программирования;
- автоматическая оптимизация управления памятью (сборка «мусора», проверка границ массивов, списков и т.д.);
- возможность контроля версий исходного кода;

- автоматическая сборка различных составных частей проекта (модулей, динамических библиотек и т.д.);
- возможность многопользовательской командной работы над единым проектом;
- поддержка, развитие и продвижение программного продукта со стороны компании-разработчика.

Анализ работ [133-147] позволяет сделать вывод, что данным критериям соответствует несколько наиболее распространенных на сегодняшний день сред разработки ПО, такие как, например, SharpDeveloper, MonoDeveloper, NetBeans, Eclipse, Microsoft Visual Studio.

Итак, целью применения среды разработки ПО является создание комплекса программных решений для расширения программной функциональности СУД Enovia SmarTeam. Поскольку указанная ИС функционируют на платформе Microsoft Windows, целесообразно выбрать продукт из линейки продуктов компании Microsoft. Одним из таких программных продуктов, соответствующим всем вышеперечисленным критериям, является Microsoft Visual Studio.

В настоящее время Microsoft Visual Studio – это один из наиболее развитых и широко распространенных современных инструментов разработки ПО, который поддерживает множество языков программирования и программную платформу .NET Framework. Основой платформы .NET Framework является общезыковая исполняющая среда CLR (от англ. Common Language Runtime) способная выполнять автономные программы и серверные веб-приложения, написанные на разных языках. CLR осуществляет автоматическое управление памятью и автоматическую проверку индексов массивов, что исключает возможность возникновения ошибок «Переполнение буфера» и «Выход за пределы массива» [149].

Поскольку IDE Microsoft Visual Studio поддерживает несколько языков программирования, возникает задача выбора конкретного языка. При этом необходимо принимать во внимание следующие условия:

- поддержка объектно-ориентированного подхода и SQL-запросов;
- наличие встроенных средств взаимодействия с неуправляемым кодом (DLL, COM-объекты).

Этим условиям отвечает язык программирования C#, разработанный в 1998—2001 гг. в компании Microsoft как основной язык разработки приложений для платформы Microsoft .NET. Программы, написанные на языке программирования C#, выполняются в среде CLR, компилятор этого языка входит в стандартную установку самой .NET, поэтому программы на нём можно создавать и компилировать даже при

отсутствии инструментальных средств среды разработки ПО (например, Microsoft Visual Studio).

C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет строгую статическую типизацию, поддерживает полиморфизм, перегрузку операторов, указатели на функции-члены классов, атрибуты, события, свойства, исключения, комментарии в формате XML. Переняв многое от своих предшественников (C++, Java, Delphi, Модула и Smalltalk), C# исключает некоторые модели, зарекомендовавшие себя на практике как проблематичные при разработке программных систем (например, в отличие от C++, C# не поддерживает множественное наследование классов) [150-155].

#### 4.2.3 Разработка программных решений

Учитывая существующие средства разработки СУД Enovia SmarTeam, последовательность действий по расширению функциональных возможностей этой ИС, с применением COM-технологии можно представить в виде следующей совокупности этапов:

1. в среде разработки Microsoft Visual Studio пишется скрипт-программа на языке C#, которая включает набор COM-объектов (описание функций СУД Enovia SmarTeam) в виде DLL-библиотеки;
2. для вызова созданной на предыдущем шаге скрипт-программы ее следует связать с элементами графического интерфейса пользователя (кнопки, пункты меню и т.д.);
3. с помощью системной утилиты regasm.exe, которая является частью программной платформы Microsoft .Net, в реестре ОС Microsoft Windows следует зарегистрировать COM-объекты, содержащихся в DLL-библиотеке.

Вышеприведенные действия относятся к общему способу расширения функциональных возможностей информационной системы с открытым API и применимы к практически любым частным случаям. В частности, данная технология была использована при организации ИИС на базе СУД Enovia SmarTeam и ее интеграции с САПР Altium Designer.

Согласно разработанному алгоритму экспорта ИМИ из САПР в СУД (рис. 3.21), начальным шагом является выбор пользователем базы данных СУД. При этом производится проверка полномочий на доступ к выбранной базе. Таким образом, из интерфейса Altium Designer производится удаленный вызов функции авторизации СУД Enovia SmarTeam. При этом выполняется процедура проверки подлинности пользователя, т.е. его аутентификация. В СУД Enovia SmarTeam это

осуществляется путем создания новой сессии – сеанса связи одного пользователя с одной базой данных.

В объектной модели SmarTeam контейнером информации о сессии является класс SmEngine. Объект класса SmEngine хранит перечень всех открытых сеансов в виде списка, доступ к которому можно получить через свойство объекта «Sessions». Класс SmSession представляет пользовательский сеанс и предоставляет следующую функциональность:

- доступ к соединению с базой данных;
- доступ к информации о пользователях;
- доступ к метаданным модели данных;
- доступ к хранилищу объектов;
- доступ к коллекции устанавливаемых дополнительных служб;
- доступ к общей зоне для обмена информацией;
- доступ к параметрам конфигурации.

Для использования вышеперечисленных функций СУД Enovia SmarTeam из интерфейса САПР Altium Designer был разработан соответствующий скрипт авторизации с применением технологии COM. Данный скрипт включает следующие разделы:

- преобразование параметров функции в COM-объекты;
- получение текущей сессии в виде объекта класса SmSession – пользовательского сеанса, связанного с одним пользователем и одной БД;
- выполнение основных операций (например, импорт/экспорт файлов) с использованием API;
- обратное преобразование COM-объектов в процедурные параметры;
- возвращение кода ошибки в вызывающую программу.

Далее представлен участок программного кода функции авторизации пользователя СУД Enovia SmarTeam, разработанный в IDE Microsoft Visual Studio 2010 на языке программирования C#.

```
Engine = new SmEngine();  
if (Engine == null) return false;  
Engine.Init("SmTeam32");  
Session = Engine.CreateSession("MySession", "SmTeam32");
```

Однако, создание пустой сессии не достаточно для проверки данных пользователя, поскольку при аутентификации необходимо подключиться к базе данных, в которой хранятся персональные данные пользователя. Для этого используется строка подключения Session.OpenDatabaseConnection(DB.Alias, DB.Password, true). При этом все конфигурационная информация о выбранной базе данных (уникальный

идентификатор, наименования классов объектной модели данных СУД Enovia SmarTeam и др.) хранятся в отдельном конфигурационном файле. На основании информации об уникальном идентификаторе базы данных осуществляется получение объекта класса SmDatabase, который содержит в себе данные для подключения к БД. После чего происходит соединение с базой данных.

На рисунке 4.6 показано разработанное средствами IDE Microsoft Visual Studio 2010 диалоговое окно для аутентификации пользователя при экспорте файлов из САПР Altium Designer в СУД Enovia SmarTeam.

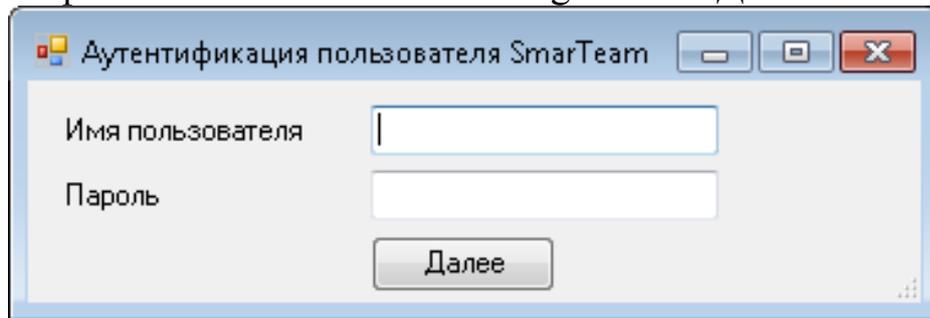


Рисунок 4.6 – Окно аутентификации

Для проверки прав пользователя на доступ к данным, например, при выборе изделия, элемента ЭСИ в случае экспорте файлов из СУД в САПР, согласно разработанному алгоритму, который показан на рисунке 3.21, используется функция SmarTeam API:

*ShowLifeCycleView\_CheckIn(myObject, Session),*

где *myObject* – имя объекта, *Session* – сессия пользовательского сеанса.

Также данная функция SmarTeam API была использована при расширении программных возможностей СУД Enovia SmarTeam с целью управления жизненным циклом объектов проектирования в ИИС.

Существующие в СУД Enovia SmarTeam возможности управления ЖЦ документов и элементов ЭСИ предполагают изменение значений атрибута «состояние ЖЦ» у объектов классов «ЭСИ» и «Документы». Этот атрибут (в случае документа) имеет ограниченное множество значений, хранящихся в системной таблице «State», модификация которой запрещена внутренними правилами платформы Enovia SmarTeam. При этом проверка состояния ЖЦ объектов осуществляется по атрибуту FIELD\_STATE, который, например, у объекта типа «Документ» может принимать следующие значения: 0 – «новый», 1 – «опубликован», 2 – «выписан», 3 – «выпущен».

Аналогичным образом выполняется проверка состояния ЖЦ у объектов класса «ЭСИ». С помощью привязки скриптов к событиям, которые возникают при, например, изменении значения атрибута

объекта, было организовано автоматическое изменение значений атрибутов «Стадия ЖЦ» у объектов проектирования (ИМИ, ЭСИ, КД, Изделие). При этом были использованы архитектурные решения шаблонов проектирования программного обеспечения «Состояние» и «Наблюдатель» согласно моделям, представленным на рис. 3.16, 3.17 и 3.19.

Итак, с использованием IDE MS Visual Studio для создания COM-объектов в виде DLL-библиотек на языке C# было разработано следующее программное обеспечение для расширения функциональных возможностей СУД Enovia SmarTeam:

- набор функций для создания объектов проектирования (элементов ЭСИ, документов);
- набор функций для изменения состояния жизненного цикла разнотипных взаимосвязанных объектов проектирования;
- набор функций для автоматического заполнения справочных каталогов элементов ЭСИ и описывающих их документов;
- набор функций для формирования текстовых конструкторских документов (спецификация, перечень элементов, ведомость покупных изделий) на основе дерева элементов ЭСИ, полученного на основе выпущенных документов ИМИ.

Согласно выражениям (3), (6), (9), показывающим зависимости элементов ЭСИ от файлов ИМИ и в соответствии с разработанными моделями и алгоритмами управления ЖЦ разнотипных объектов проектирования, формирование элементов ЭСИ возможно после публикации и выпуска документов ИМИ. Поскольку в дереве элементов ЭСИ объекты класса «Прочее изделие», согласно разработанной UML-диаграмме классов (рис. 2.7) являются дочерними по отношению к объекту класса «Сборочная единица», предложены следующие положения:

- команда «Сформировать структуру сборочной единицы», которая вызывает команду формирования объектов в дереве элементов ЭСИ;
- поскольку, в соответствии с [120], основные конструкторские документы (например, ведомость покупных изделий, спецификация, перечень элементов) разрабатываются на сборочную единицу, целесообразно также добавить команды вызова их формирования в пункт контекстного меню объекта класса «Сборочная единица» в дереве элементов ЭСИ.

На рисунке 4.7 показано контекстное меню, которое содержит команды вызова разработанного программного обеспечения для формирования ЭСИ и КД. Данное контекстное меню запускается от объекта класса «Сборочная единица» в дереве ЭСИ.

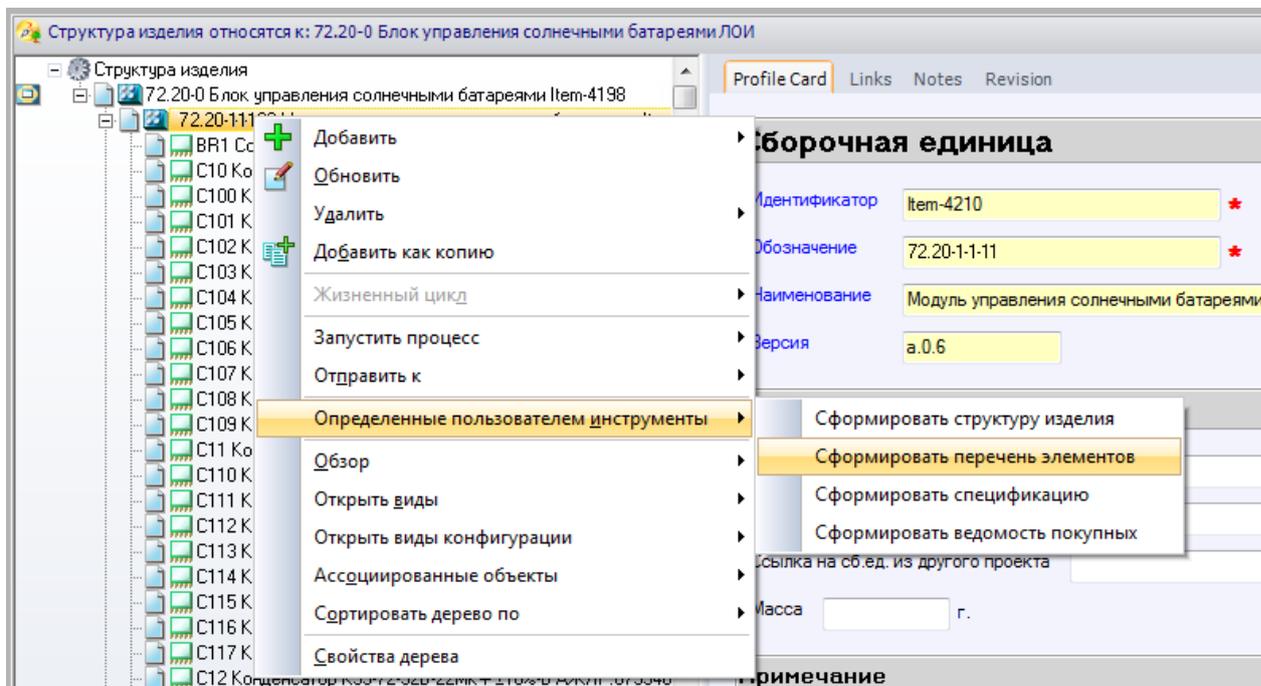


Рисунок 4.7 – Контекстное меню для запуска команд формирования ЭСИ и КД

Согласно разработанным моделям взаимосвязей объектов проектирования и алгоритмам управления их жизненным циклом, КД представляет собой итоговое статичное текстовое отображение элементов структуры изделия, которые формируются на основе ИМИ-файлов, разработанных в САПР. Поскольку при изменении САПР-файлов происходит автоматическое обновление дерева элементов структуры изделия, целесообразно формировать КД в виде документов текстового формата, который не должен подлежать изменению в ручном режиме. Одним из таких форматов является \*.pdf. В комплексе программных решений, разработанных для ИИС проектирования бортовой РЭА КА в ОАО «ИСС», на основании методики, подхода и алгоритмов, предложенных автором настоящей работы, созданы функции программного формирования следующих конструкторских документов: ведомость покупных изделий, спецификация и перечень элементов. Таким образом, реализовано положение о взаимозависимости документов КД от элементов ЭСИ и файлов ИМИ.

#### 4.3 Структура и взаимосвязь проектных данных в СУД Enovia SmarTeam

В качестве исходных данных для практической апробации разработанного алгоритмического и программного обеспечения был выбран проект по разработке следующего изделия: «Блок электронный устройства поворота батареи солнечной». Разработка этого изделия, место которого в иерархической структуре КА показано на рисунке 4.8,

выполняется в отделении проектирования и испытаний бортовой РЭА ОАО «ИСС».



Рисунок 4.8 – Место рассматриваемого изделия в иерархической структуре КА

Разработка изделия «Блок электронный устройства поворота батареи солнечной», фактически, является одной из задач в составе комплекса работ по проектированию КА. Экземпляр класса «Проект» является ключевым объектом, который обеспечивает семантическую связь между разнотипными сущностями (изделием, его элементами, документами) с помощью ссылок. Таким образом, проект представляет собой портфель данных о предмете разработки. Применяя к процессу проектирования изделий принцип «железного треугольника проекта» [13], можно сделать вывод, что экземпляр класса «Проект» представляет собой контейнер для структурированного хранения информации о временных, стоимостных и качественных ограничениях (параметрах) работ. Под качественными характеристиками проекта по разработке изделия следует понимать соответствие разрабатываемого объекта требованиям технического задания (ТЗ).

Таким образом, целевым результатом проекта по разработке изделия является непосредственно само изделие. Поэтому, согласно UML-диаграмме классов, разработанной на этапе концептуального проектирования модели данных СУД (рис. 2.7), с экземпляром класса «Проект», который хранит сведения общего характера о разрабатываемом изделии, следует связать экземпляр класса «Изделие».

Экземпляр класса «Изделие» предназначен для структурированного хранения обобщенной технической информации о проектируемом изделии, такой как, например, наименование, массогабаритные характеристики, функциональное назначение и т.д. Все эти данные являются значением атрибутов класса «Изделие». Поскольку специфика космического приборостроения предполагает одновременное существование нескольких вариантов изделия – на каждую стадию испытаний, например, ЛОИ, КДИ и т.д., для отслеживания степени завершения проектирования на этих стадиях введен соответствующий атрибут.

Компонентный состав разных версий изделия для каждой стадии испытаний реализуется с использованием объектов класса «ЭСИ». Анализируя разработанную UML-диаграмму классов для хранения информации об объектах проектирования в СУД, можно сделать вывод, что на верхнем уровне абстракции компонентный состав изделия аккумулируется объектом класса «Сборочная единица». Таким образом, в классе «Сборочная единица» необходим соответствующий атрибут для отслеживания стадии испытаний, для которой выполняется разработка изделия.

Кроме того, согласно разработанным моделям и алгоритмам управления ЖЦ объектов проектирования, следует предусмотреть атрибуты для возможности существования разных версий элементов ЭСИ, а также для выражения степени завершения работ по их проектированию. На рисунке 4.9 показана иерархическая структура дерева проектов и взаимосвязь объектов проектирования различных типов в СУД Enovia SmartTeam: к экземпляру класса «Проект» присоединен экземпляр класса «Изделие», к которому, в свою очередь, относится несколько экземпляров класса «Сборочная единица» (ЭСИ).

В СУД Enovia SmartTeam существует несколько видов связей между объектами, поэтому в деревьях разных классов присутствует цветовая дифференциация связей:

- общие связи (между объектами разных классов) показаны зеленым цветом;
- иерархические связи (между объектами одного класса) – показаны черным цветом.

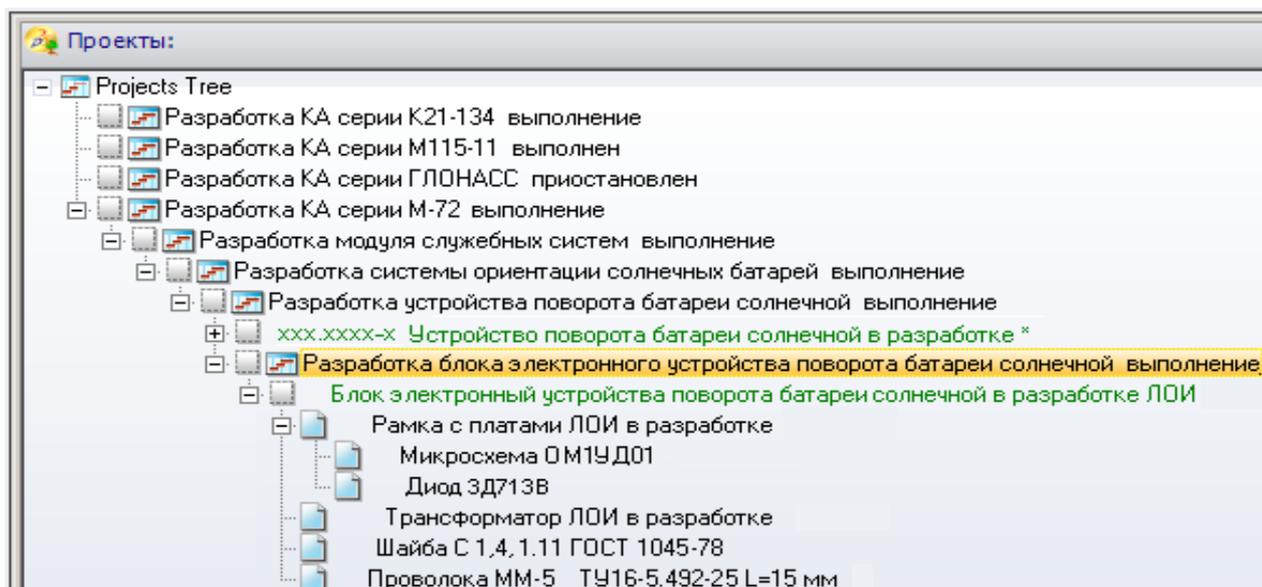


Рисунок 4.9 – Дерево проектов СУД Enovia SmarTeam со ссылками на экземпляры класса «Изделия» и «ЭСИ»

#### 4.4 Динамика автоматизированного управления жизненным циклом разнотипных объектов проектирования в СУД Enovia SmarTeam

Согласно технологии проектирования бортовой РЭА, первоначальным этапом разработки изделия является создание его электрической принципиальной схемы. Данный процесс выполняется в ОАО «ИСС» с использованием ECAD-САПР Altium Designer. После создания схемы с помощью разработанного программного обеспечения интеграции данной САПР с СУД Enovia SmarTeam выполняется экспорт файлов проекта Altium Designer в СУД. Практически, экспортируемые файлы проекта Altium Designer являются одной из информационных моделей проектируемого изделия. В соответствии с разработанными моделями и алгоритмами обмена данными между САПР и СУД, при экспорте осуществляются процедуры авторизации пользователя, а также выбора необходимого проекта, изделия и сборочной единицы.

Файлы проекта Altium Designer, загруженные впервые в СУД Enovia SmarTeam, автоматически прикрепляются к вновь созданным объектам класса «ИМИ» в дереве документов, связанных с выбранным объектом класса «Сборочная единица» из дерева элементов ЭСИ. В начале своего ЖЦ объект типа «ИМИ» характеризуется атрибутами, которые имеют значения, показанные в таблице 4.4.

**Начальные значения атрибутов, описывающих состояние ЖЦ  
объекта «ИМИ»**

Атрибут	Значение атрибута
версия	а.0
этап ЖЦ в СУД	новый
статус ЭП	не подписан ЭП
статус согласования	не просмотрен

После публикации документа возможна его доработка, при этом изменения будут сохранены в файле, который будет прикреплен к новой версии документа. Когда все необходимые изменения в документ внесены, его следует отправить на согласование, которое (в соответствии с разработанными моделями и алгоритмами управления жизненным циклом объектов проектирования) возможно только с выпущенными документами. Таким образом, некоторые атрибуты, описывающие состояние ЖЦ объекта типа «ИМИ» примут значения, показанные в таблице 4.5.

**Изменение значений атрибутов, описывающих состояние ЖЦ  
объекта «ИМИ»**

Атрибут	Значение атрибута
версия	а
этап ЖЦ в СУД	выпущен

Согласование документа представляет собой процесс, при запуске которого, в соответствии с предложенной моделью в виде UML-диаграммы классов (рис. 3.9), создается новый экземпляр класса «Задание параметров согласования». С вновь созданным объектом связан согласуемый документ общей связью. Задание параметров согласования документа выполняется с помощью заполнения значений графической формы объекта класса «Задание параметров согласования» (рис. 4.10). При этом производится выбор согласующих лиц и указание порядка их работы с документом (последовательно или параллельно).

**Согласование**

Идентификатор: Proc-15 \*

Название: Proc-15 - Согласование \*

Документ: xxx.xxx-x.Altium a

Инициатор: joe . . .

Итерация: 1

**Срок**

Дата запуска: 05/07/2012 11:16

Дата окончания (план):

Дата окончания (факт):

**Информация о задании**

При отсутствии замечаний согласовать документ .

**Исполнители**

Последовательно

Параллельно

Выбран	Логин	Фамилия	Имя	Отчество	Должность	Отдел
<input type="checkbox"/>	Ivan	Иванов	Иван	Иванович	конструктор	XXX
<input type="checkbox"/>	Peter	Петров	Петр	Петрович	технолог	XX

Рисунок 4.10 – Графическая форма задания параметров согласования

Согласно методологии Workflow, используемой во многих современных корпоративных информационных системах, в том числе СУД, в рамках рассматриваемой Enovia SmarTeam процесс представлен в виде последовательности узлов. Узел представляет собой действие пользователя или системы, характеризующееся различными параметрами: исполнитель, уведомляемый пользователь, срок исполнения, возможность возврата к предыдущему шагу и т.д. Как правило, разработка данной последовательности узлов выполняется в специальном графическом редакторе. Созданная диаграмма привязывается к соответствующему классу-контейнеру информации о рассматриваемом процессе.

На рисунке 4.11 показана Workflow-диаграмма процесса согласования документа в СУД Enovia SmarTeam.



Рисунок 4.11 - Workflow-диаграмма процесса согласования документа

После задания параметров согласования, согласно моделям, представленным на рисунках 3.9-3.14, в зависимости от выбранных значений начинается генерация элементарных задания для указанных ранее исполнителей. Workflow-диаграмма элементарного согласования фактически представляет собой единственный узел для конкретного исполнителя и узел отправки уведомления инициатору процесса о результатах согласования документа текущим исполнителем (рис. 4.12).

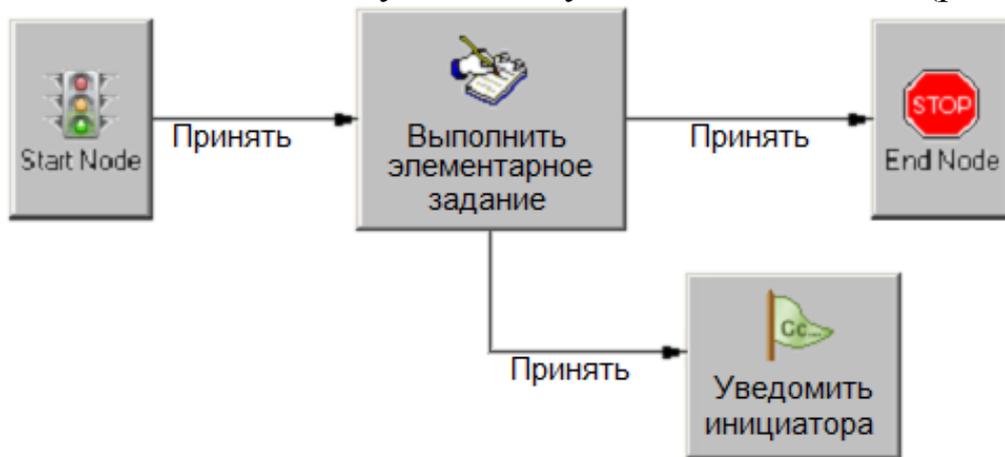


Рисунок 4.13 - Workflow-диаграмма элементарного согласования

Исполнитель оповещается о задании посредством письма, направленного ему в рамках встроенного модуля СУД Enovia SmartTeam (рис. 4.13).

Входящие						
	Тема/процесс	Тело/Узел	Создан	Получено	Время работы	
1	Рос-19	Согласовать	05/07/2012 12:46	05/07/2012	0 00:07	

Рисунок 4.13 – Входящее сообщение о новом задании

Просмотр сообщения запускает графическую форму созданного объекта класс «Элементарное согласование» (рис. 4.14), при этом значения некоторых атрибутов, например, сведения о текущем исполнителе, заполняются автоматически за счет срабатывания скриптов, привязанных к узлам Workflow-диаграммы.

The screenshot displays a web-based interface for task management. At the top, there are navigation buttons: 'Принять', 'Уровень вверх', 'Закрыть', and 'Помощь'. Below this is a breadcrumb trail: 'Profile Card > Links > Notes > Блок-схема'. The main content area is titled 'Согласование' and contains the following fields:

- Идентификатор:** Proc-19 \*
- Название:** Proc-19 - Согласование \*
- Документ:** xxx.xxxx-x.Altium a
- Инициатор:** joe . . .
- Итерация:** 1
- Срок:**
  - Дата запуска: 05/07/2012 11:16
  - Дата окончания (план):
  - Дата окончания (факт):
- Информация о задании:** При отсутствии замечаний согласовать документ .
- Исполнитель:** Иванов Иван Иванович, конструктор, отдел XXX, Ivan \*
- Статус выполнения задания:** Документ не просмотрен

Рисунок 4.14 – Графическая форма элементарного согласования конкретным исполнителем

Динамическое распределение потоков работ реализуется за счет программной настройки узлов линейных Workflow-диаграмм, к которым привязываются скрипты, вызывающие функции расширения функциональных возможностей СУД Enovia SmarTeam. При этом также используется технология COM для создания DLL-библиотек с описанием функций, как показано на рисунке 4.5. Для отслеживания статусов выполнения элементарных задания был разработан набор функций, включающих следующие параметры:

- сведения об активном процессе, выполнившего вызов функции;
- сведения о пользователе (workflow-сессия, действие);
- сведения о текущем процессе (объект, родитель, дочерние процессы, текущий узел);
- сведения о графической форме процесса (операции над графической формой, списки входных и выходных параметров).

Перечисленные параметры являются атрибутами суперкласса «Работа», предназначенного для хранения информации о процессах в СУД Enovia SmarTeam. Таким образом, разработанные функции позволяют отследить состояние связанных процессов, за счет чего реализуется синхронизация результатов элементарных заданий в случае параллельного согласования или генерация задания для нового исполнителя при последовательном течении процесса. При разработке программного обеспечения для решения этих задач были использованы

типовые архитектурные шаблоны проектирования «Наблюдатель» и «Состояние», согласно моделям, представленных на рисунках 3.12-3.15, 3.18.

Если документ не согласован хотя бы одним из согласующих лиц, необходима его доработка. В СУД Enovia SmartTeam это реализуется операцией «взять на редактирование», при которой объект становится недоступным для изменения другим пользователям и открывается в той системе, в которой он был разработан, например, ECAD-САПР Altium Designer. После доработки ИМИ при попытке ее экспорта обратно в СУД Enovia SmartTeam выполняется проверка его существования в дереве документов указанного проекта, изделия и сборочной единицы. При существовании объекта в СУД Enovia SmartTeam автоматически создается его новая опубликованная версия, к которой прикрепляются файлы проекта Altium Designer.

В случае согласования документа всеми согласующими лицами, в соответствии с разработанными моделями и алгоритмами управления жизненным циклом объектов проектирования, становится возможным формирование ЭСИ на основе данных из ИМИ. Аналогичным образом выполняется конструкторское проектирование изделия в MCAD-САПР SolidWorks и экспорт разработанной ИМИ в СУД Enovia SmartTeam. В левой части рисунка 4.13 показано дерево документов, относящихся к рассматриваемому примеру, разработке изделия «Блок электронный устройства поворота батареи солнечной».

Конструкторский документ «Спецификация» связан связью спецификации с элементами ЭСИ, входящими в состав специфицируемого изделия, и общими связями с документами, которые обозначены в нем. В правой части рисунка 4.15 представлен перечень связей спецификации сформированного конструкторского документа и элементов ЭСИ.

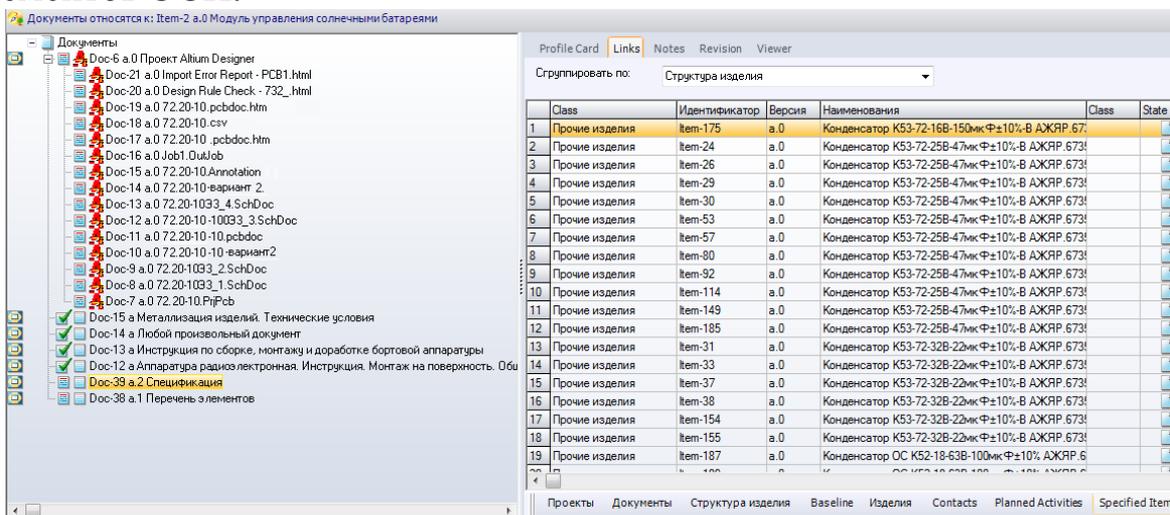


Рисунок 4.15 – Дерево документов сборочной единицы изделия и связь спецификации с элементами ЭСИ

В связи с выявленной спецификой представления ЭРИ в ECAD-САПР Altium Designer и MCAD-САПР SolidWorks, а также необходимостью существования единой сущности, которая описывает ЭРИ в СУД Enovia SmarTeam, формирование ЭСИ на основе данных из согласованных ИМИ выполняется следующим образом:

1. объекты классов «Сборочная единица», «Прочее изделие», «Деталь», «Стандартное изделие», автоматически формируются в дереве элементов ЭСИ на основе данных из файлов SolidWorks и связываются с соответствующим объектом класса «Документы» связью спецификации с использованием встроенного инструмента СУД Enovia SmarTeam «Мастер копирования дерева» и механизмов встроенной интеграции СУД с MCAD-САПР SolidWorks;
2. объекты класса «Прочее изделие» связываются с объектом класса «Документы», (ИМИ-файл, разработанный в Altium Designer) связью спецификации с использованием разработанного программного обеспечения, которое расширяет существующие функциональные возможности СУД Enovia SmarTeam;
3. при наличии элементов ЭСИ, информация о которых отсутствует в файлах ИМИ, соответствующие объекты следует добавить в дерево элементов ЭСИ вручную, при этом у вновь созданного объекта класса «ЭСИ» будет отсутствовать связь спецификации.
4. При этом создаваемые объекты классов «ЭСИ» будут иметь значение атрибута «стадия ЖЦ в СУД» - в разработке, а связь спецификации позволяет отследить источник появления объекта класса «ЭСИ». Далее, согласно разработанными моделями и алгоритмами управления жизненным циклом объектов проектирования, возможно автоматическое получение КД на выбранную сборочную единицу. Сформированные конструкторские документы автоматически связываются с элементами ЭСИ, на основе которых они созданы и публикуются в СУД Enovia SmarTeam.
5. Далее, при отсутствии ошибок при первичной проверке, созданные КД выпускаются. Затем, как и в случае с ИМИ, по выпущенным объектам класса «КД» запускаются процессы согласования. Если КД не согласован хотя бы одним из согласующих лиц, то необходима коррекция первоисточника – документа ИМИ, а затем создание новой версии ЭСИ. При этом связь спецификации объекта класса «ЭСИ» с объектом класса «Документ» позволяет быстро определить необходимый объект для внесения изменений.
6. В случае согласования конструкторских документов изменяются связанные с ними связью спецификации объекты типа «ЭСИ»:

значение атрибута «стадия ЖЦ в СУД» становится равным «спроектирован». В случае согласования КД на сборочную единицу верхнего уровня иерархии, связанную с объектом класса «Изделие», у изделия изменяется значение атрибута «статус готовности» на «разработано». При создании новой версии элементов, из которых состоит изделие, в дереве ЭСИ, у объекта класса «Изделие» значение атрибута «статус готовности» снова меняется становится равным «в разработке». Значение атрибута «стадия испытаний» становится таким же, как у вновь созданной версии дерева элементов ЭСИ в состоянии ЖЦ «в разработке». Когда значение атрибута «стадия ЖЦ в СУД» у всех элементов всех деревьев ЭСИ становится «разработано», атрибут «статус готовности» у объекта класса «Изделие» принимает значение «разработано». При этом также автоматически происходит изменение атрибута «статус выполнения» у связанного с изделием экземпляра класса «Проект» (становится равным «выполнен»).

Согласно концепции технологий информационной поддержки ЖЦ изделий, СУД в качестве центрального ядра ИИС предназначена не только для работы с данными непосредственно в процессе проектирования, но и, в том числе, для структурированного хранения информации из предшествующих работ с возможностью ее использования и модификации в текущих проектах. Поэтому, в соответствии с классификацией составных частей изделия по [22], было организовано хранение в СУД Enovia SmarTeam в виде справочных данных следующих объектов:

- унифицированные детали;
- унифицированные сборочные единицы;
- стандартные изделия;
- прочие изделия;
- материалы и сортаменты.

В связи с прикладной спецификой СУД Enovia SmarTeam, справочные данные размещаются в проекте «Справочник», который включает подпроекты по типам справочных объектов, экземпляров классов «ЭСИ». Детальное описание типового объекта ЭСИ, такое, как, например, его 3D-модель, реализуется экземплярами класса «Документы»: ИМИ и КД.

#### **4.5 Резюме практической реализации**

В данной главе приведены результаты практической реализации разработанных моделей и алгоритмов автоматизированного управления жизненным циклом разнотипных взаимозависимых объектов проектирования. Показана реализация предложенного метода

построения ИИС для процессов проектирования бортовой РЭА КА в ОАО «ИСС» на базе интеграций СУД Enovia SmarTeam с САПР Altium Designer и SolidWorks.

Прикладная апробация разработанного алгоритмического и программного обеспечения проведена на базе отделения проектирования и испытаний бортовой РЭА в ОАО «ИСС». Для этого было проведено исследование набора программных средств, используемых в этом отделении для задач проектирования бортовой РЭА и предложены варианты их интеграции в рамках ИИС.

При выполнении работы достигнуты следующие практические результаты:

- программная адаптация СУД Enovia SmarTeam к специфике задач проектирования бортовой РЭА в ОАО «ИСС» с помощью разработки скриптов для автоматического создания разнотипных взаимозависимых объектов проектирования и изменения состояний их жизненного цикла;
- развертывание прототипа ИИС на базе интеграций СУД Enovia SmarTeam с САПР Altium Designer и SolidWorks на реальных рабочих местах сотрудников отделения проектирования и испытаний бортовой РЭА.

Анализ результатов апробации, включая опросы участников тестовой эксплуатации прототипа ИИС в ОАО «ИСС», а также экспертные оценки, позволяет сделать вывод, что использование разработанного комплекса программных решений оптимизировало выполнение процессов проектирования бортовой РЭА КА за счет следующих аспектов:

- сокращение времени разработки КД за счет автоматического формирования на основе согласованных файлов информационных моделей;
- сокращение длительности процесса согласования документов за счет предварительного параллельного просмотра ИМИ и КД всеми согласующими;
- сокращение длительности процесса электрического проектирования и уменьшение ошибок при разработке бортовой РЭА за счет интеграции СУД Enovia SmarTeam с ECAD-САПР Altium Designer;
- сокращение количества изменений документов при согласовании (текстовые конструкторские документы формируются автоматически, на основе уже согласованных информационных моделей изделия);

- уменьшение конфликтов в процессе взаимодействия за счет возможности отслеживания актуальной версии разрабатываемых документов/данных (в СУД доступны для работы только актуальные и утвержденные данные);
- накопление интеллектуального капитала за счет структурированного хранения результатов предыдущих проектных работ и возможности их повторного использования, в том числе с внесением необходимых изменений.

Таким образом, прототип ИИС проектирования и испытаний бортовой РЭА КА, в основе которого лежит разработанное авторами настоящего исследования алгоритмическое и программное обеспечение, обладает широкими функциональными возможностями для решения большого количества задач, стоящих перед предприятиями космического приборостроения, и может быть рекомендован к внедрению в эксплуатацию.

## ЗАКЛЮЧЕНИЕ

В настоящей работе решена актуальная научно-техническая задача, заключающаяся в разработке моделей и алгоритмов автоматизированного управления жизненным циклом разнотипных взаимозависимых объектов с целью сокращения сроков процессов проектирования высокотехнологичной продукции с длительным сроком активного существования за счет применения современных информационных технологий.

Получены следующие основные научные и практические результаты:

1. Сформулирована и проанализирована проблема управления высокотехнологичной продукцией с длительным сроком активного существования на этапе ее проектирования. Показано взаимовлияние разнотипных объектов проектирования друг на друга и необходимость автоматизированного управления ими на протяжении всего цикла проектирования за счет автоматического изменения их состояний. Показана возможность применения информационных систем и технологий поддержки жизненного цикла изделий к проблемам проектирования высокотехнологичной продукции с длительным сроком активного существования при условии их адаптации к особенностям данной предметной области. Выполнен поиск математических методов для формализации сложной структуры взаимосвязей между разнотипными объектами проектирования и динамики изменения состояний их ЖЦ.

2. Выполнена разработка математических и алгоритмических положений для автоматизированного управления жизненным циклом разнотипных взаимозависимых объектов в интегрированной информационной среде, включая методы получения формальных информационных моделей жизненного цикла рассматриваемых объектов, концептуальное проектирование базы данных СУД, методику представления жизненного цикла объектов в виде раскрашенного орграфа и алгоритмы изменения их состояний.

3. Разработан метод построения интегрированной информационной среды на базе системы управления данными и ее интеграций с системами автоматизированного проектирования, представленный набором требований к взаимодействию программных средств и оригинальными алгоритмами их реализации, включая автоматическое заполнение справочных каталогов при построении электронной структуры изделия.

4. Выполнена программная адаптация СУД Enovia SmarTeam к специфике задач проектирования высокотехнологичной продукции с длительным сроком активного существования на примере бортовой радиоэлектронной аппаратуры.

5. Выполнена программная реализация разработанных моделей и алгоритмов на примере комплекса программных решений для интегрированной информационной среды космического приборостроения на базе СУД Enovia SmarTeam и ее интеграций с САПР Altium Designer и SolidWorks.

6. Апробация разработанного алгоритмического и программного обеспечения проведена в отделении проектирования и испытаний бортовой РЭА ОАО «ИСС». Было выполнено исследование набора программных средств, используемых в этом отделении для проектирования бортовой РЭА, разработаны варианты их интеграции в рамках ИИС и развернут прототип ИИС на реальных рабочих местах сотрудников.

7. Подтверждена работоспособность предложенных в настоящей работе теоретических и практических положений, в числе которых:

- формальная информационная модель взаимосвязей разнотипных объектов проектирования, позволяющая настроить модель данных СУД для структурированного хранения информации и автоматизированного управления жизненным циклом объектов в интегрированной информационной среде согласно особенностям процессов разработки высокотехнологичной продукции с длительным сроком активного существования;

- методика формализации жизненного цикла разнотипных взаимозависимых объектов с целью получения формальных информационных моделей, показывающих динамику изменений исследуемых сущностей и позволяющих определить правила управления ими в интегрированной информационной среде;

- алгоритмы согласования документов, отличающиеся автоматизированным распределением потоков работ между исполнителями процессов за счет применения положений теории сетевого планирования, распараллеливания и синхронизации операций;

- метод построения интегрированной информационной среды на базе системы управления данными и ее интеграций с системами автоматизированного проектирования, представленный набором требований к взаимодействию программных средств и оригинальными алгоритмами их реализации, включая автоматическое заполнение справочных каталогов при построении электронной структуры изделия;

- программное обеспечение расширения функциональных возможностей СУД Enovia SmarTeam (управление процессами, создание и изменение объектов);

- методика формирования электронной структуры изделия и текстовых конструкторских документов в СУД Enovia SmarTeam на основе файлов информационных моделей, разработанных в САПР Altium Designer и SolidWorks.

Полученные теоретические положения и практические результаты настоящего исследования носят общий прикладной характер и, при необходимой доработке, могут быть использованы не только в области космического приборостроения, но и в других сферах создания высокотехнологичной продукции с длительным сроком активного существования.

## ЛИТЕРАТУРА

1. Боргест Н.М. Онтология проектирования: теоретические основы. Часть 1. Понятия и принципы: учебное пособие. – Самара: Издательство Самарского государственного аэрокосмического университета, 2010. – 88 с.
2. Индикаторы инновационной деятельности: статистический сборник – М.: ГУ-ВШЭ, 2009. – 488 с.
3. Бендиков М.А., Фролов И.Э. Высокотехнологичный сектор промышленности России. Состояние, тенденции, механизмы инновационного развития. – М.: Наука, 2008. – 584 с.
4. Р 50-605-80-93 Система разработки и постановки продукции на производство. Термины и определения. – М.: Стандартинформ, 1993. – 44 с.
5. ГОСТ Р 34.10-2012 Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи. – М.: Стандартинформ, 2012. – 24 с.
6. ГОСТ Р 34.11-2012 Информационная технология. Криптографическая защита информации. Функция хэширования. – М.: Стандартинформ, 2012. – 24 с.
7. Левин А.И., Судов Е.В. CALS-сопровождение жизненного цикла // Открытые системы. - 2001. - № 3(59). - С.58-62
8. Лунева Е.Е., Куренков И.Н., Суханова Ю.А. Совмещение процессного управления и проектной деятельности на приборостроительном предприятии // Молодежь и современные информационные технологии: сборник трудов IX Всероссийской научно-практической конференции студентов, аспирантов и молодых ученых с международным участием, 11-13 мая 2011г. Труды в 2-х т. – Томск: Изд-во ТПУ, 2011. – С. 180-182.
9. Романов А. В. Теория комплексной оптимизации проектирования космических аппаратов с ядерными термоэмиссионными

энергетическими установками. – СПб: НПО «Профессионал», 2010. – 473 с.

10. Ярошенко Ф. А., Бушуев С. Д., Танака Х. Управление инновационными проектами и программами на основе системы знаний Р2М. – Киев: Саммит-книга, 2011. – 268с.

11. ГОСТ Р 15.201-2000 Система разработки и постановки продукции на производство. Продукция производственно-технического назначения. Порядок разработки и постановки продукции на производство. – М.: Стандартинформ, 2000. – 10 с.

12. Гущин В.Н. Основы устройства космических аппаратов: учебник для вузов. – М.: Машиностроение, 2003. – 272 с.

13. Мазур И.И., Шапиро В.Д. Управление проектами. – М.: Омега-Л, 2007. – 664с.

14. Товб А.С., Ципес Г.Л. Управление проектами: стандарты, метод, опыт. – М.: Олимп-Бизнес, 2003. – 240 с.

15. Гагарский В. Процессное и проектное управление сходство и различия [Электронный ресурс] // Портал «Профессионал управления проектами». – Режим доступа: <http://www.pmpofy.ru/content/rus/219/2197-article.asp>

16. Теоретические основы испытаний и экспериментальная отработка сложных технических систем: учебное пособие / Александровская Л.Н., Круглов В.И., Кузнецов А.Г. и др. – М.: Логос, 2003. – 736 с.

17. ГОСТ 16504-81 Система государственных испытаний продукции. Испытания и контроль качества продукции. Основные термины и определения. – М.: Стандартинформ, 2004. – 26 с.

18. Ключев Е.А. Принципы и средства отработки программного обеспечения блоков управления перспективных космических аппаратов //Электронные и электромеханические системы и устройства: тезисы докладов научно-технической конференции молодых специалистов. – Томск: ОАО «НПЦ Полюс», 2008. – с.142-144

19. ГОСТ Р 50782-95 Аппаратура бортовая системы передачи сигналов оповещения с космических аппаратов – М.: Стандартинформ, 2013. – 8 с.
20. Медведчиков Д.А.. Роль управления рисками космической деятельности на современном этапе реформирования отечественной ракетно-космической промышленности // Общероссийский научно-технический журнал «Полет», 2003 – №12. – С.45-49
21. ГОСТ Р 52003-2003 Уровни разукрупнения радиоэлектронных средств. Термины и определения. – М.: Стандартинформ, 2003. – 11 с.
22. ГОСТ 2.101-68 Единая система конструкторской документации. Виды изделий. – М.: Стандартинформ, 2013. – 4 с.
23. Каленкович Н.И., Боровиков С.М. Радиоэлектронная аппаратура и основы её конструкторского проектирования. – Минск: БГУИР, 2008. – 199 с.
24. Зубавичус В. О значении электроники: военный аспект // Современная электроника, 2006. – № 4. – С. 18.
25. Электроника. Аспекты развития // Под ред. Мартынова В. В. – М.: ЦНИИ «Электроника», 2004. – 459 с.
26. Засыпкин А.В. ГЛОНАСС [Электронный ресурс] // Российская экономика, 2008. – Режим доступа: <http://www.pravda.ru/science/technologies/telecommunications/07-02-2008/254947-ghonass-0/07.02.2008>
27. Общероссийский классификатор видов экономической деятельности. – М.: Стандартинформ, 2003.
28. Базров Б.М. Основы технологии машиностроения. – М.: Машиностроение, 2005. – 736 с.
29. ГОСТ 2.103-68 Единая система конструкторской документации. Стадии разработки. – М.: Стандартинформ, 2011. – 7 с.
30. ГОСТ 2.102-68 Единая система конструкторской документации. Виды и комплектность конструкторских документов. – М.: Стандартинформ, 2007. – 29 с.

31. ГОСТ 2.052-2006 Единая система конструкторской документации. Электронная модель изделия. Общие положения. – М.: Стандартиформ, 2006. – 15 с.
32. ГОСТ 2.109-73 Единая система конструкторской документации. Основные требования к чертежам. – М.: Стандартиформ, 2006. – 34 с.
33. ГОСТ 2.053-2006 Единая система конструкторской документации. Электронная структура изделия. Общие положения. – М.: Стандартиформ, 2006. – 13 с.
34. ГОСТ Р ИСО 9001-2001 Системы менеджмента качества. Требования. – М.: Стандартиформ, 2004. – 31 с.
35. ГОСТ Р ИСО 9004-2001 Системы менеджмента качества. Рекомендации по улучшению деятельности Требования. – М.: Стандартиформ, 2004. – 59 с.
36. ГОСТ Р 53791-2010 Ресурсосбережение. Стадии жизненного цикла изделий производственно-технического назначения. – М.: Стандартиформ, 2010. – 12 с.
37. ГОСТ Р ИСО/МЭК 15288-2005 Информационная технология. Системная инженерия. Процессы жизненного цикла систем. – М.: Стандартиформ, 2005. – 57 с.
38. Управление качеством продукции. Инструменты и методы менеджмента качества: учебное пособие / С.В. Пономарев, С.В. Мищенко, В.Я. Белобрагин, В.А. Самородов, Б.И. Герасимов, А.В. Трофимов и др. – М.: РИА «Стандарты и качество», 2005. – 240 с.
39. Краснухин А.И. Реализация методологии проектирования сложных изделий в среде систем SMARTTEAM и CATIA V5 // Открытые системы, 2003 – №6. – С. 32-36
40. Dunne A. Hertzian tales: electronic products, aesthetic experience and critical design. – London: Royal College of Art computer related design research studio, 1999. - pp. 117-123
41. Raby F. Design Noir: The Secret Life of Electronic Objects. – Basel: Birkhäuser, 2001. – pp. 54-60

42. McCormack J., Dorin A., Innocent T. Generative Design: a paradigm for design research // Design Research Society, Melbourne, 2004. – pp. 46-51
43. Хорошев А.Н. Основы системного проектирования [Электронный ресурс]. – Режим доступа: [http://www.cfin.ru/management/controlling/sys\\_project.shtml](http://www.cfin.ru/management/controlling/sys_project.shtml)
44. ГОСТ 2.051-2006 Единая система конструкторской документации. Электронные документы. Общие положения. – М.: Стандартинформ, 2007. – 16 с.
45. Зильбербург Л.И., Молочник В.И., Яблочников Е.И. Реинжиниринг и автоматизация технологической подготовки производства в машиностроении. – СПб: «Компьютербург», 2003. - 152 с.
46. Яблочников Е.И., Молочник В.И., Фомина Ю.Н. Реинжиниринг бизнес-процессов проектирования и производства: учебное пособие – СПб: СПбГУИТМО, 2008. – 152 с.
47. Пестрецов С.И. CALS-технологии в машиностроении: основы работы в CAD/CAE-системах: учебное пособие - Тамбов: Изд-во ГОУ ВПО ТГТУ, 2010. – 104 с.
48. Информационное обеспечение, поддержка и сопровождение жизненного цикла изделия / Под ред. В.В. Бакаева. – М.: Машиностроение-1, 2005. 624 с.
49. Яблочников Е.И., Молочник В.И., Миронов А.А. ИПИ-технологии в приборостроении: учебное пособие. – СПб.: СПбГУИТМО, 2008. – 128 с.
50. Яблочников Е.И. Автоматизация технологической подготовки производства в приборостроении: учебное пособие. – СПб., СПбГИТМО (ТУ), 2002 - 92 с.
51. Яблочников Е.И., Молочник В.И. и др. Методы управления жизненным циклом приборов и систем в расширенных предприятиях. – СПб: СПбГУ ИТМО, 2009. - 148 с.

52. Норенков И. П. Основы автоматизированного проектирования: учебник для вузов. – 4-е изд., перераб. и доп. – М.: Изд-во МГТУ им. Н. Э. Баумана, 2009. – 430 с.
53. Brown Jim. PLM and MES: Dassault Systemes says «Yes!» [Электронный ресурс] // Tech-Clarity Publication Archive. – Режим доступа: <http://tech-clarity.com/publications.htm>
54. Collaborative Innovations: Author Archives [Электронный ресурс]. – Режим доступа: <http://collaborativeinnovation.org/author/andrea/>
55. Кузнецов А.В. Интеграция систем: подходы и решения // Автоматика, связь, информатика, 2011. – №7. – С. 14-15
56. Шалумов А.С., Никишкин С.И., Носков В.Н. Введение в CALS-технологии: учебное пособие. – Ковров: КГТА, 2002. – 137 с.
57. Судов Е.В., Левин А.И., Петров А.В., Чубарова Е.В. Технологии интегрированной логистической поддержки изделий машиностроения. М.: Информбюро, 2006. - 232 с.
58. PLM by Oleg Shilovitsky [Электронный ресурс]. – Режим доступа: <http://plmtwine.com>
59. Левин Д.Я., Малюх В.Н., Ушаков Д.М. Энциклопедия PLM. – Новосибирск: Ледас, 2008 г. – 448 с.
60. Малюх В.Н. PLM и BIM: близницы или двойняшки [Электронный ресурс]. – Режим доступа: [http://isicad.ru/ru/articles.php?article\\_num=14344](http://isicad.ru/ru/articles.php?article_num=14344)
61. Райкин Л.И., Райкин И.Л., Сидорук Р.М., Кабанова Т.Ю. Создание цифровых прототипов с помощью технологий Autodesk. Часть 1. Этап концептуального дизайна // CAD/ CAM/CAE Observer, 2010. - №1 – С. 28-34
62. Технология цифровых прототипов. [Электронный ресурс]. – Режим доступа: <http://www.autodesk.ru/adsk/servlet/index?siteID=871736&id=12098379>
63. А. Н. Давидович, А. М. Вещагин. Использование гетерогенных САПР (Autodesk Inventor и Catia) на примере электронно-цифрового

- макета скоростного катера проекта // Рациональное Управление Предприятием, 2008. – №3 – С.48-51
64. Стандарты CALS [Электронный ресурс]. – Режим доступа:<http://www.itstan.ru/it-i-is/standarty-cals.html>
65. Горячев А., Нисковский А.. Модель OSI. [Электронный ресурс]. – Режим доступа: <http://www.compress.ru/article.aspx?id=10853&iid=437>
66. ГОСТ Р ИСО/МЭК 7498-1-99 Взаимосвязь открытых систем. Базовая эталонная модель. – М.: Стандартинформ, 2006. – 62 с.
67. Черняк Л. JT как основа PLM // Открытые системы, 2011. - №8 – С. 21-23
68. Либликин М. ЭП и ЭЦП: в чем разница? [Электронный ресурс]. – Режим доступа: <http://bankir.ru/tehnologii/s/ep-i-etsp-v-chem-raznitsa-10000236>
69. Все о САПР, PLM, ERP [Электронный ресурс]. – Режим доступа: <http://isicad.ru/ru/>
70. М. Головкин Технический документооборот: система управления документацией или прилогаток к приложениям обработки данных [Электронный ресурс]. – Режим доступа:[http://www.doc-online.ru/a\\_id/35/](http://www.doc-online.ru/a_id/35/)
71. Михайлов В.Г. Анализ и сравнение существующих PDM [Электронный ресурс]. – Режим доступа:<http://fsapr2000.ru/index.php?showtopic=17636>
72. Гаврилов В. Использование систем класса PDM при управлении проектными работами // САПР и графика, 2001. - №11. – С. 43-46
73. PDM-система вместо планово-диспетчерского отдела [Электронный ресурс]. – Режим доступа: <http://www.bytemag.ru/articles/detail.php?ID=9143>
74. Жуков И. Отслеживание жизненного цикла изделия в Lotsia PLM // САПР и графика, 2005. - № 6. – С. 22-25
75. Вичугова А. А., Вичугов В.Н., Дмитриева Е. А. Жизненный цикл документа в информационных системах управления данными//Вестник

науки Сибири. Серия: Информационные технологии и системы управления. - 2011 - №.1 - С. 328-334.

76. Вичугова А. А., Дмитриева Е. А., Цапко Г. П.. Разработка модели данных PDM-системы Enovia SmarTeam для управления спецификациями при создании радиоэлектронной аппаратуры// Прикладная информатика. - 2010. - № 5 (29). - С. 23-29

77. R. W. Butler What is Formal Methods? [Электронный ресурс]. – Режим доступа: <http://shemesh.larc.nasa.gov/fm/fm-what.html>

78. C. Michael Holloway. Why Engineers Should Consider Formal Methods. [Электронный ресурс]. – Режим доступа: <http://klabs.org/richcontent/verification/holloway/nasa-97-16dasc-cmh.pdf>

79. Орлов С. А. Технологии разработки программного обеспечения: учебное пособие. – 2-е изд. – СПб. : Питер, 2003. – 480 с.

80. Багиева М. Г. Автоматизация проектирования систем: учебное пособие. - Владикавказ: СКГМИ, 2008. – 86 с.

81. Брауде Э.Дж. Технология разработки программного обеспечения: Пер. с англ. – СПб: Питер, 2004. – 655 с.

82. Вендров А.М. Проектирование программного обеспечения экономических информационных систем: учебник. – М.: Финансы и кредит, 2006. – 352 с.

83. Дурнов П.А. Современные тенденции построения распределенных информационных систем //Сборник научных трудов:Т.1. – Вологда: ВоПИ, 1997.- С.4-9

84. Thomas E. Potok, Mladen Vouk, Andy Rindos. Productivity Analysis of Object-Oriented Software Developed in a Commercial Environment // Practice and Experience, Vol. 29, 1999. - № 10, pp 833-847

85. The strengths of the academic enterprise [Электронный ресурс]. – Режим доступа: <http://www.cs.utexas.edu/users/EWD/transcriptions/EWD11xx/EWD1175.html>,

86. Frederick P. Brooks, Jr. No Silver Bullet. Essence and Accidents of Software Engineering» // Computer Magazine, 1987 - №4. – pp. 27-30

87. Бадд Т. Объектно-ориентированное программирование в действии. – СПб.: «Питер», 1997. – 464 с.
88. Иванов А.Г., Карпова А.В., Семик В.П., Филинов Ю.Е. Объектно-ориентированная среда программирования // Системы и средства информатики, 1991. - №2 – С. 12-16
89. Иванов А.Г., Пятницкий А.А., Филинов Ю.Е. Объектно-ориентированный подход к технологии программирования [Электронный ресурс]. – Режим доступа: <http://www.math.rsu.ru/smalltalk/article91.ru.html>
90. Liskov B. CLU Reference Manual. – California: Springer-Verlag, 1984. – 190 pp.
91. Liskov B. Program development in Java: Abstraction, specification and object-oriented design. – California: Addison-Wesley, 2001. – 464 pp.
92. Дрождин В.В., Зинченко Р.Е. Системный подход к концептуальному моделированию предметной области в самоорганизующейся информационной системе // Программные продукты и системы, 2009. – № 4. – С. 73–79
93. Лоффредо Д. Архитектура информационной системы: как избежать моделирования степлеров [Электронный ресурс]. – Режим доступа: <http://msdn.microsoft.com/ru-ru/library/ee895180.aspx>
94. Гради Буч. Объектно–ориентированный анализ и проектирование с примерами приложений. 3–е издание. Издательство: Вильямс, 2010. – 720 стр.
95. Information Technology. Investment Management: A Framework for Assessing and Improving Process Maturity. [Электронный ресурс]. – Режим доступа: <http://www.gao.gov/new.items/d04394g.pdf>
96. ГОСТ Р ИСО/МЭК ТО 10032-2007 Эталонная модель управления данными. – М.: Стандартинформ, 2007. – 45 с.
97. Дейт К.Дж. Введение в системы баз данных: пер. с англ. – 8-е изд. – М.: «Вильямс», 2006. – 1328 с.
98. Когаловский М.Р. Перспективные технологии информационных систем – М.: ДМК Пресс, 2003. – 288 с.

99. Когаловский М.Р. Энциклопедия технологий баз данных – М.: Финансы и статистика, 2002. – 800 с.
100. Цикритзис Д., Лоховски Ф., Модели данных. – М.: Финансы и статистика, 1985, – 344 с.
101. Кузнецов С.Д. Основы баз данных – 2-е изд. – М.: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2007. – 484 с.
102. Сениченков Ю.Б. Визуальное моделирование сложных динамических систем. – Санкт-Петербург: Мир и Семья, 2001. – 242 с.
103. Pressman R.S. Software Engineering: A Practioner's Approach. 5th ed. California: McGraw-Hill, 2000. – 943 pp.
104. Колесов Ю.Б., Сениченков Ю.Б. Моделирование систем. Объектно-ориентированный подход: учебное пособие. – СПб: БВХ-Петербург, 2012. – 192 с.
105. Фаулер М., Скотт К. UML. Основы: пер. с англ. - СПб: Символ, 2002. - 192 с.
106. Леоненков А. Самоучитель UML: БХВ-Петербург – 2004. – 432 с.
107. Лойко М. Что такое workflow? // Делопроизводство и документооборот на предприятии. – 2011. - №6. – С. 43-54
108. Белайчук А., Вагнер Ю. BPM в действии // Директор информационной службы», 2007. - № 02. – С. 14-16
109. Хаммер М. Бизнес в XXI веке: повестка дня. – М: Хорошая книга, 2005. – 336 с.
110. Озерова И.Г., Дмитриева Е.А. Цапко Г.П., Вичугов В.Н. Методика автоматизированного построения схем в системах управления бизнес-процессами // Известия Томского политехнического университета, 2007 - т. 311. – № 5. – С. 51-55
111. Пичкалев А. В. Испытания радиоэлектронной аппаратуры на лабораторном отработочном комплексе // Решетневские чтения: материалы XII Междунар. науч. конф.; Сиб. гос. аэрокосмич. ун-т. Красноярск, 2008. – С. 158-159

112. Прудков В. В. Особенности построения программного обеспечения автономной отработки подсистем блока управления перспективных КА // Решетневские чтения XII. – Красноярск: СибГАУ, 2009. – С. 531–532
113. Андерсон Дж. Дискретная математика и комбинаторика: перевод с англ. – М.: «Вильямс», 2006. – С. 960
114. Бертуланфи Л. фон. Общая теория систем – обзор проблем и результатов // Системные исследования: Ежегодник. – М.: Наука, 1969. – С. 30-54
115. Хопкрофт Дж., Мотвани Р., Ульман Джеф. Введение в теорию автоматов, языков и вычислений. – М.: Вильямс, 2002. – 528 с.
116. Белоусов А. И., Ткачев С. Б. Дискретная математика. Серия: Математика в техническом университете. – М: МГТУ им. Н. Э. Баумана, 2001. – 744 с.
117. Мурох А.А. Элементы теории графов в анализе современных систем управления// Фундаментальные исследования. – 2006. – № 3 – С.91-92
118. Зыков А. А. Основы теории графов – М.: «Вузовская книга», 2004. – С. 664
119. Э. Таненбаум, М. ван Стеен. Распределенные системы. Принципы и парадигмы: перевод с англ.– СПб: Питер, 2003. – 877 с.
120. ГОСТ 2.106-96 Единая система конструкторской документации. Текстовые документы. – М.: Стандартинформ, 2007. – 37 с.
121. Ивченко В.Г. Конструирование и технология ЭВМ: Конспект лекций. Часть 1. - Таганрог: ТРТУ, 2001. - 52 с.
122. Конструкторско-технологическое проектирование электронной аппаратуры: Учебник для вузов. – М.: Изд. МГТУ им. Н.Э. Баумана, 2002. – 528 с.
123. Валетов В.А., Кузьмин Ю.П., Орлова А.А., Третьяков С.Д. Технология приборостроения: Учебное пособие. - СПб.: СПбГУ ИТМО, 2008. – 336 с.

124. Тупик В.А. Технология и организация производства радиоэлектронной аппаратуры. – СПб: Издательство: СПбГЭТУ "ЛЭТИ" – 2005. – 148 с.
125. Кормен Томас Х., Лейзерсон Чарльз И., Ривест Рональд Л. Алгоритмы: построение и анализ, 2-е издание: пер. с англ. — М.: Вильямс, 2010. — 1296 с.
126. Машков М. Поиск кратчайшего пути между двумя вершинами в ориентированном графе методом Дейкстры [Электронный ресурс]. – Режим доступа: <http://www.exponenta.ru/soft/mathcad/stud25/index.asp>
127. М. Фаулер. Шаблоны корпоративных приложений: перевод с англ. – М.: «Вильямс», 2009. – 544 с.
128. Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влссидес. Приемы объектно-ориентированного проектирования. Паттерны проектирования: перевод с англ. – СПб: «Питер», 2007. – 366 с.
129. Марк Гранд. Шаблоны проектирования в JAVA. Каталог популярных шаблонов проектирования, проиллюстрированных при помощи UML: перевод с англ.. – М.: «Новое знание», 2004. – 560 с.
130. Крэг Ларман. Применение UML 2.0 и шаблонов проектирования: перевод с англ. – М.: «Вильямс», 2006. – 736 с.
131. Джошуа Кериевски. Рефакторинг с использованием шаблонов (паттернов проектирования): перевод с англ. – М.: «Вильямс», 2006. – 400 с.
132. Типы корпусов микросхем. [Электронный ресурс]. – Режим доступа: <http://www.radiodetali.com/article/mikroshema/mikroshema.htm>
133. Storey, M.A.; Ryall, J.; Bull, I.; Myers, D.; Singer, Janice TODO or To Bug: Exploring How Task Annotations Play a Role in the Work Practices of Software Developers // Proceedings of the International Conference on Software Engineering, 2008. – Leipzig: ICSE. – pp. 46-50
134. «Interaktives Programmieren als Systems-Schlager» from Computerwoche [Электронный ресурс]. – Режим доступа: <http://www.computerwoche.de/heftarchiv/1975/47/1205421/>

135. Ладыженский Г.Б. Интеграция приложений такая, как она есть [Электронный ресурс]. – Режим доступа: <http://citforum.ru/gazeta/50/>
136. В. Боркус. Методы и инструменты интеграции корпоративных приложений. М.: RC Group, 2005. – 215 с.
137. Г. Хоп, Б. Вульф. Шаблоны интеграции корпоративных приложений: перевод с англ.. М: Издательство: «Вильямс», 2007.– 672 с.
138. Цимбал А. Сравнительный анализ CORBA и COM [Электронный ресурс]. – Режим доступа: <http://www.interface.ru/borland/corbacom.htm>
139. Дубова Н. COM или CORBA? Вот в чем вопрос [Электронный ресурс]. – Режим доступа: <http://vladdavkis.chat.ru/comorcorba.htm>
140. Роберт Дж. Технология COM+. Основы и программирование: перевод с англ. – М.: «Вильямс», 2000. – 480 с.
141. Чистяков В.Ю. COM [Электронный ресурс]. – Режим доступа: <http://www.rsdn.ru/article/com/COMvs.xml>
142. API [Электронный ресурс]. – Режим доступа: <http://ru.wikipedia.org/wiki/API>
143. Реестр [Электронный ресурс]. – Режим доступа: [http://ru.wikipedia.org/wiki/Реестр\\_Windows](http://ru.wikipedia.org/wiki/Реестр_Windows)
144. М. Murata, D. Kohn, and C. Lilley Internet Drafts: XML Media Types [Электронный ресурс]. – Режим доступа: <http://tools.ietf.org/html/draft-murata-kohn-lilley-xml-03>
145. David Megginson. Imperfect XML: Rants, Raves, Tips, and Tricks. [Электронный ресурс]. – Режим доступа: [http://ptgmedia.pearsoncmg.com/images/0131453491/samplechapter/megginson\\_ch08.pdf](http://ptgmedia.pearsoncmg.com/images/0131453491/samplechapter/megginson_ch08.pdf)
146. Douglas Crockford. JSON: The Fat-Free Alternative to XML [Электронный ресурс]. – Режим доступа: <http://www.json.org/fatfree.html>
147. Скотт В. Эмблер, Прамодкумар Дж. Садаладж. Рефакторинг баз данных: эволюционное проектирование: перевод с англ.. – М.: «Вильямс», 2007.-368 с.

148. Тема вечного холивара: выбор IDE [Электронный ресурс]. – Режим доступа: <http://javaprofi.blogspot.com/2011/04/ide.html>
149. CRL [Электронный ресурс]. – Режим доступа: [http://ru.wikipedia.org/wiki/Common\\_Language\\_Runtime](http://ru.wikipedia.org/wiki/Common_Language_Runtime)
150. Эндрю Троелсен. Язык программирования C# 2010 и платформа .NET 4.0. – 5-е изд. – М.: Вильямс, 2010. – 1392 с.
151. Герберт Шилдт C# 4.0: полное руководство. – М.: «Вильямс», 2010. – 1056 с.
152. Джон Скит C#: программирование для профессионалов, 2-е издание: перевод с англ.. – М.: «Вильямс», 2011. – 544 с.
153. Кристиан Нейгел, Карли Уотсон и др. Visual C# 2010: полный курс: перевод с англ..- М.: Диалектика, 2010
154. Трей Нэш C# 2010: ускоренный курс для профессионалов: перевод с англ.. – М.: Вильямс, 2010. – 592 с.
155. Кристиан Нейгел, Билл Ивсен, Джей Глинн, Карли Уотсон, Морган Скиннер C# 4: Платформа .NET 4 для профессионалов: перевод с англ. – М.: Диалектика, 2010. – 1440 с.
156. Вичугова А.А., Вичугов В.Н. Применение объектно-ориентированной методологии при создании аналитических таблиц на основе графических ARIS-моделей // Экономика, статистика и информатика. Вестник УМО, 2010, - № 6. - С. 227-230
157. А. А. Вичугова, Э.С. Аметова, В.Н. Вичугов, Ю.А. Суханова, С.Г. Цапко. Проект построения единого информационного пространства для процессов создания бортовой радиоэлектронной аппаратуры космических аппаратов в ОАО «ИСС» // Вестник СибГАУ: СибГАУ, Красноярск, 2012. - № 1 (41). - с. 4-10
158. Вичугова А.А. Модель изменения стадий жизненного цикла объектов проектирования космического приборостроения // Экономика, статистика и информатика. Вестник УМО. - 2012 - №. 4 - С. 181-185
159. Вичугова А.А., Вичугов В.Н., Дмитриева Е.А. Особенности работы с документами в информационных системах управления данными //

Программные продукты и системы. - 2012 - №. 3 - С. 184-189 Вичугова А.А., Яковлева Е.М. Использование САПР DipTrace для автоматизации проектирования печатных плат // Прикладная информатика, 2008. -№ 6(18) -С. 44-50

160. Вичугова А.А., Аметова Э.С., Суханова Ю.А., Фень А.М., Дмитриева Е.А. Средства автоматизации функционирования единого информационного пространства проектирования бортовой радиоэлектронной аппаратуры космических аппаратов // Системы управления и информационные технологии. - 2012 - Т. 4 - №. 50 – С. 70-74

161. А.А. Вичугова, В. Н. Вичугов, Г.П. Цапко. Формальная модель структуры взаимосвязей разнотипных объектов проектирования // Известия Томского политехнического университета / Томский политехнический университет (ТПУ) . – 2013. – Т. 322, № 5: Управление, вычислительная техника и информатика. – С. 164-169

162. Вичугова А.А. Разработка модели данных информационной системы поддержки жизненного цикла изделий для задач космического приборостроения // Программные продукты и системы. – 2013 – №3(103). – С. 209-215

163.

## СПИСОК ОБОЗНАЧЕНИЙ И СОКРАЩЕНИЙ

Термин	Значение термина
API	Интерфейс программирования приложений (Application Programming Interface)
CAD	Система автоматизированного проектирования (Computer-Aided Design)
CAE	Системы автоматизации инженерных расчетов (Computer-Aided Engineering)
CALS	Непрерывная информационная поддержка поставок и жизненного цикла (Continuous Acquisition And Lifecycle Support)
CAM	Автоматизированные системы технологической подготовки производства (Computer-Aided Manufacturing)
COM	Объектная модель компонентов (Component Object Model)
CORBA	Общая архитектура брокера объектных запросов (Common Object Request Broker Architecture)
DFD	Диаграммы потоков данных (Data Flow Diagrams)
DLL	Динамически подключаемая библиотека (Dynamic-Link Library)
ECAD	Информационная система электрического проектирования изделий (Electronic Computer-Aided Design)
ESB	Сервисная шина предприятия (Enterprise Service Bus)
GUID	Статистически уникальный 128-битный идентификатор (Globally Unique Identifier)
IDE	Интегрированная среда разработки (Integrated Development Environment)
IDEF	методологии семейства ICAM (Integrated Computer-Aided Manufacturing) для решения задач моделирования сложных систем (Icam DEfinition)
IGES	Цифровое представление для коммуникации данных определения продукта (Digital Representation for Communication of Product Definition Data)
JAVA	Объектно-ориентированный язык программирования
JSON	Текстовый формат обмена данными, основанный на JavaScript (JavaScript Object Notation)
MCAD	Информационная система механического проектирования изделий (Mechanical Computer-Aided Design)
PLM	Система управления жизненным циклом изделия (Product Lifecycle Management)
SADT	Методология структурного анализа и проектирования (Structured Analysis And Design Technique)
SOAP	Протокол обмена структурированными сообщениями в распределённой вычислительной среде (Simple Object Access Protocol)

STEP	Международный CALS-стандарт для компьютерного представления и обмена промышленными данными
UML	Унифицированный язык моделирования (Unified Modeling Language)
XML	Расширяемый язык разметки (eXtensible Markup Language)
ГОСТ	Государственный стандарт
ЕИП	Единое информационное пространство
ЕСКД	Единая система конструкторской документации
ЕСТД	Единая система технологической документации
ЕСПД	Единая система программной документации
ЖЦ	Жизненный цикл
ИИС	Интегрированная информационная среда
ИМИ	Информационная модель изделия
ИС	Информационная система
ИТ	Информационные технологии
КА	Космический аппарат
КД	Конструкторская документация
КДИ	Конструкторско-доводочные испытания
МПН	Модуль полезной нагрузки
МСС	Модуль служебных систем
ЛОИ	Лабораторно-отрабочные испытания
НИР	Научно-исследовательская работа
НСИ	Нормативно-справочная информация
НЭО	Наземно-экспериментальная отработка
ОС	Операционная система
ОАО «ИСС»	Открытое акционерное общество «Информационные спутниковые системы» имени академика М.Ф. Решетнева
ПО	Программное обеспечение
ПП	Печатная плата
При	Приемочные испытания
РЭА	Радиоэлектронная аппаратура
САПР	Система автоматизированного проектирования
СУБД	Система управления базами данных
СУД	Система управления данными
ТЗ	Техническое задание
УГО	Условные графические обозначения
ФКП	Федеральная космическая программа
ФЦП	Федеральная целевая программа
ЭМИ	Электронная модель изделия
ЭП	Электронная подпись
ЭРИ	Электрорадиоизделия
ЭСИ	Электронная структура изделия

*Научное издание*

*Анна Александровна Вичугова,  
Владимир Николаевич Вичугов,  
Елена Анатольевна Дмитриева,  
Геннадий Павлович. Цапко,  
Сергей Геннадьевич Цапко*

**МЕТОДОЛОГИЧЕСКИЕ ОСНОВЫ ПРОЕКТИРОВАНИЯ  
СЛОЖНЫХ НАУКОЕМКИХ ИЗДЕЛИЙ И  
ПРИНЦИПЫ ПОСТРОЕНИЯ  
ИНТЕГРИРОВАННОЙ ИНФОРМАЦИОННОЙ СРЕДЫ  
НА БАЗЕ CALS-ТЕХНОЛОГИЙ**

Монография

**Издано в авторской редакции**

Дизайн обложки *Фатеева Т.А.*

---

Издательство ТПУ. 634050, г. Томск, пр. Ленина, 30. [www.tpu.ru](http://www.tpu.ru)  
Подписано к печати 29.11.2103.  
Формат 60x84/16. Бумага «Снегурочка» . Печать XEROX.  
Усл. печ. л. 10,47. Усл.-изд.л. 9,46  
Тираж 500 экз. Заказ 1344-13

---

Тираж отпечатан в издательстве