



**«Московский государственный технический университет  
имени Н.Э. Баумана»**

**(МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ Робототехника и комплексная автоматизация

КАФЕДРА Компьютерные системы автоматизации производства

## **РАСЧЁТНО - ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**к дипломному проекту на тему:**

Интеграция подходов дискретного имитационного

моделирования в РДО

Студент Луцан Д.Н. \_\_\_\_\_  
(Подпись, дата) (И.О.Фамилия)

Руководитель дипломного проекта \_\_\_\_\_  
(Подпись, дата) (И.О.Фамилия)

Консультант по  
исследовательской части \_\_\_\_\_  
(Подпись, дата) (И.О.Фамилия)

Консультант по проектной части \_\_\_\_\_  
(Подпись, дата) (И.О.Фамилия)

Консультант по  
организационно-экономической части \_\_\_\_\_  
(Подпись, дата) (И.О.Фамилия)

Консультант по охране труда и экологии \_\_\_\_\_  
(Подпись, дата) (И.О.Фамилия)

УТВЕРЖДАЮ

Заведующий кафедрой \_\_\_\_\_  
(Индекс)

\_\_\_\_\_  
(И.О.Фамилия)

« \_\_\_\_ » \_\_\_\_\_ 20 \_\_ г.

## З А Д А Н И Е на выполнение дипломного проекта

Студент \_\_\_\_\_ Луцан Дмитрий Николаевич  
(Фамилия, имя, отчество)

Интеграция подходов дискретного имитационного моделирования в РДО  
(Тема дипломного проекта)

Источник тематики (НИР кафедры, заказ организаций и т.п.) \_\_\_\_\_ НИР кафедры

Тема дипломного проекта утверждена распоряжением по факультету № \_\_\_\_\_  
от « 12 » марта \_\_\_\_\_ 2011 г.

### 1. Исходные данные

Постановка задачи на интеграцию подходов дискретного имитационного моделирования  
в РДО. Документация по РДО.

### 2. Техничко-экономическое обоснование

Высокая трудоемкость разработки имитационных моделей сложных дискретных систем,  
в которые входят процессы обслуживания.

(обзор и анализ альтернативных решений; выбор вариантов для сравнения;  
конкретные улучшаемые характеристики или параметры; возможный технико-экономический эффект и т.п.)

### 3. Научно-исследовательская часть

Анализ альтернативных вариантов создания ресурсов модели и передачи их в процесс  
обслуживания

Консультант \_\_\_\_\_  
(Подпись, дата) \_\_\_\_\_ (И.О.Фамилия)

#### **4. Техническое проектирование**

Разработать алгоритмы создания ресурсов модели, управления процессами из базы знаний модели и диаграммы, реализующие выбранную концепцию интеграции подходов имитационного моделирования в РДО.

---

---

---

---

---

Консультант \_\_\_\_\_  
(Подпись, дата) (И.О.Фамилия)

#### **5. Рабочее проектирование**

Реализовать интеграцию подходов дискретного имитационного моделирования в РДО.

Проверить работоспособность системы на модели.

---

---

---

---

---

Консультант \_\_\_\_\_  
(Подпись, дата) (И.О.Фамилия)

#### **6. Организационно-экономическая часть**

Расчет затрат на интеграцию подходов дискретного имитационного моделирования в РДО.

---

---

---

---

---

Консультант \_\_\_\_\_  
(Подпись, дата) (И.О.Фамилия)

#### **7. Охрана труда и экология**

Провести анализ опасных и вредных факторов, воздействующих на разработчика ИМ в РДО. Выполнить расчет системы заземления, выбор способа утилизации отработанного ПЭВМ.

---

---

---

---

---

Консультант \_\_\_\_\_  
(Подпись, дата) (И.О.Фамилия)

#### **8. Оформление дипломного проекта**

8.1. Расчетно-пояснительная записка на 121 листе формата А4.

8.2. Перечень графического материала (плакаты, схемы, чертежи и т.п.) всего 11 А1, из них 5 плакатов А1

---

---

---

---

---

Дата выдачи задания « \_\_\_ » \_\_\_\_\_ 20\_\_ г.

В соответствии с учебным планом дипломный проект выполнить в полном объеме в срок до « \_\_\_ » \_\_\_\_\_ 20\_\_ г.

**Руководитель дипломного проекта** \_\_\_\_\_  
(Подпись, дата) (И.О.Фамилия)

**Студент** \_\_\_\_\_  
(Подпись, дата) (И.О.Фамилия)

#### **Примечание:**

1. Задание оформляется в двух экземплярах; один выдаётся студенту, второй хранится на кафедре.

Государственное образовательное учреждение высшего профессионального образования  
«Московский государственный технический университет имени Н.Э. Баумана»  
(МГТУ им. Н.Э. Баумана)

УТВЕРЖДАЮ

Заведующий кафедрой \_\_\_\_\_  
(Индекс)

\_\_\_\_\_  
(И.О.Фамилия)

« \_\_\_\_ » \_\_\_\_\_ 20 \_\_ г.

## КАЛЕНДАРНЫЙ ПЛАН выполнения дипломного проекта

Студент \_\_\_\_\_ Лушан Дмитрий Николаевич \_\_\_\_\_  
(Фамилия, имя, отчество)

\_\_\_\_\_ Интеграция подходов дискретного имитационного моделирования в РДО \_\_\_\_\_  
(Тема дипломного проекта)

№ п/п	Наименование этапов дипломного проекта	Выполнение этапов		Примечание
		Срок	Объем, %	
1.	Предпроектное исследование		5%	
2.	Концептуальное проектирование		20%	
3.	Техническое задание		5%	
4.	Техническое проектирование		20%	
5.	Рабочее проектирование		20%	
6.	Научно-исследовательская часть		20%	
7.	Организационно-экономическая часть		5%	
8.	Охрана труда и экология		5%	

Руководитель дипломного проекта \_\_\_\_\_  
(Подпись, дата) (И.О.Фамилия)

Студент \_\_\_\_\_  
(Подпись, дата) (И.О.Фамилия)

**П р е с е д а т е л ю**  
**Государственной Аттестационной Комиссии № \_\_\_\_\_**

факультета \_\_\_\_\_ МГТУ им. Н.Э. Баумана

Направляется студент \_\_\_\_\_ на защиту дипломного проекта  
(фамилия, инициалы)

\_\_\_\_\_  
(наименование темы)

**С п р а в к а о б у с п е в а е м о с т и**

Студент \_\_\_\_\_ за время пребывания в МГТУ имени Н.Э. Баумана с 20\_\_ г.  
полностью выполнил учебный план специальности со следующими оценками:  
отлично – \_\_\_\_\_%, хорошо – \_\_\_\_\_%, удовлетворительно – \_\_\_\_\_%.

Секретарь факультета \_\_\_\_\_  
(фамилия, инициалы)

Декан факультета \_\_\_\_\_  
(фамилия, инициалы)

**З а к л ю ч е н и е р у к о в о д и т е л я д и п л о м н о г о п р о е к т а**

Студент Луцан Д.Н. выполнил дипломный проект на актуальную тему: «Интеграция подходов дискретного ИМ в РДО». В основе интеграции лежит использование общей БД модели всеми подходами моделирования в РДО. Помимо этой концепции, были добавлены специальные управляющие команды, позволяющие паттернам запускать процессы на исполнение. Работа является продолжением и обобщением проектов, выполненных студентом ранее. Результаты могут быть использованы в курсе Моделирование ТП и ПП. За время работы студент получил навыки проектирования и программирования на С++. Проект выполнен на высоком техническом уровне.

Руководитель « \_\_\_\_\_ » \_\_\_\_\_ 20\_\_ г.

**З а к л ю ч е н и е к а ф е д р ы о д и п л о м н о м п р о е к т е**

Дипломный проект просмотрен и студент \_\_\_\_\_ может быть допущен  
(фамилия, инициалы)  
к защите проекта в Государственной Аттестационной комиссии.

Заведующий кафедрой \_\_\_\_\_  
« \_\_\_\_\_ » \_\_\_\_\_ 20\_\_ г.

## РЕФЕРАТ

Отчет 121 с., 12 рис., 14 табл., 12 источников, 2 прил.

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ, РЕСУРС-ДЕЙСТВИЕ-ОПЕРАЦИЯ, МОДЕЛЬ, АКТИВНОСТЬ, СОБЫТИЕ, ПРОЦЕСС, ТИП РЕСУРСА, РЕСУРС.

Объектом разработки является мультипарадигменная система дискретного имитационного моделирования на основе РДО. Ресурс, действие, операция (РДО) – программный комплекс, предназначенный для имитационного моделирования сложных дискретных систем с целью проведения их анализа и синтеза.

Цель работы – интеграция в системе РДО всех классических подходов дискретного имитационного моделирования, которые поддерживаются в системе лишь по отдельности.

При создании системы проведены исследования, целью которых являлось установить оптимальный алгоритм создания ресурсов модели, учитывающий их дальнейшее использование во всех аспектах модели: и в событиях, и в процессах обслуживания, и в активностях базы знаний.

В результате работы произведена интеграция подходов дискретного имитационного моделирования в системе РДО, которая позволяет эффективно использовать и событийный подход, и подход сканирования активностей, и процессный подход имитационного моделирования в рамках одной модели.

Эффективность мультипарадигменного подхода при моделировании заключается в возможности использовать сильные стороны каждого классического подхода, тем самым увеличивая гибкость, наглядность, выразительность и надежность создаваемых моделей.

## ПЕРЕЧЕНЬ СОКРАЩЕНИЙ, СИМВОЛОВ И СПЕЦИАЛЬНЫХ ТЕРМИНОВ С ИХ ОПРЕДЕЛЕНИЕМ

ARIS	Architecture of Integrated Information Systems (методология и программный продукт для моделирования бизнес-процессов компании)
OMG	Object Management Group (Группа управления объектами)
UML	Universal Modeling Language (Универсальный язык моделирования)
БД	База данных
БЗ	База знаний
ИМ	Имитационная модель
ОС	Операционная система
ПП	Программный продукт
ПО	Программное обеспечение
ПЭВМ	Персональная электронно-вычислительная машина
РДО	Ресурс Действие Операция – система имитационного моделирования
СДС	Сложная дискретная система
ТЗ	Техническое задание
ЭП	Эскизный проект

## ВВЕДЕНИЕ

Благодаря естественному желанию людей найти и использовать некоторые общесистемные принципы и методы с целью обобщения накопленного опыта и результатов в различных сферах человеческой деятельности моделирование систем становится все более популярным. Именно этот общесистемный подход в перспективе должен стать той базой, которая позволит исследователю работать с любой сложной системой, независимо от ее физической сущности.

Широкое использование ИМ в задачах анализа и синтеза систем объясняется сложностью (а иногда и невозможностью) применения строгих методов оптимизации, которая обусловлена размерностью решаемых задач и неформализуемостью сложных систем.

Рынок программных продуктов предлагает множество инструментов для имитационного моделирования, например, Arena – разработка американской корпорации Rockwell Automation, или AnyLogic – российской компании XJ Technologies. Но подавляющее большинство подобных продуктов под дискретно-событийным имитационным моделированием подразумевают лишь реализацию процессноориентированного подхода, в то время как наиболее сложные и требующие тщательного анализа аспекты моделируемой системы находятся, как правило, в подсистеме управления, для моделирования которой процессный подход не подходит. Т.е. на практике моделированию в основном подлежат лишь процессы (бизнес-процессы, производственные процессы и пр.), протекающие в системе, без возможности варьировать правила поведения систем управления по причине сложности построения полных моделей.



С другой стороны, РДО изначально основывается на подходе сканирования активностей, который позволяет простым и явным образом моделировать системы управления. Позже в РДО появились также процессный и событийный подходы, поэтому возможность использования в одной модели всех возможных подходов делает его одновременно и привлекательным для пользователей, которые знакомы с процессным подходом, и уникально совмещающим в себе все сильные стороны различных подходов дискретного имитационного моделирования.

Таким образом, интеграция подходов дискретного имитационного моделирования в РДО является актуальной задачей развития РДО, которая логичным образом заканчивает внедрение в язык поддержки всех подходов дискретного имитационного моделирования, выполненных за последние несколько лет.

# 1 ПРЕДПРОЕКТНОЕ ИССЛЕДОВАНИЕ

## 1.1 Назначение программного комплекса РДО

Задачи системного анализа и синтеза объектов различной природы и назначения часто решаются с использованием имитационных моделей. Эти модели позволяют исследовать динамические аспекты поведения сложных дискретных систем и процессов. Имитация, в частности, позволяет выполнить анализ функционирования объекта, прогнозирование, организационное управление, поддержать принятие решений при проектировании и управлении.

RAO-studio является средством имитационного моделирования, позволяющим воспроизводить на ПЭВМ динамику объекта, принятие решений сложной системой управления, и даже моделировать деятельность человека при принятии решений. В основе имитатора лежит РДО-метод формализации знаний о дискретных системах и процессах. Знания представляются в форме модифицированных продукционных правил, событий и процессов. При этом сохраняются такие достоинства продукционных систем, как универсальность, гибкость и наличие формальных механизмов логического вывода. Традиционные продукционные правила являются частным случаем модифицированных, поэтому в имитационную модель легко могут быть включены, например, экспертные системы.

Язык описания объектов, алгоритмов управления и задач в RAO-studio – это по существу язык представления знаний. Он требует от пользователя лишь знаний в предметной области, а не в программировании. Пользователь описывает ресурсы, правила функционирования, требуемые показатели и анимационные кадры непосредственно в терминах предметной области, не прибегая при этом к

представлению своей системы в терминах какого-либо известного метода (системы очередей, сети Петри, автоматы) или языка типа SLAM-II, ARENA, SIMPLE++ и других. Это резко повышает гибкость, мощность и наглядность модели. РДО – язык высокого уровня, использующий символические имена, арифметические и логические выражения и функции, генераторы псевдослучайных чисел, модифицированные и простые продукции.

Основные элементы RAO-studio – это модифицированная продукционная система, аппарат событий, с помощью которых в системе реализованы событийный, процессный и подход сканирования активностей. События начала действий базы знаний и процессов инициируются механизмом логического вывода, а события, запланированные пользователем (в явном виде) или системой (в неявном виде – события окончания действий) – специальным событийным блоком. При имитации состояние системы изменяется в соответствии с описанием события, происходящим в данный момент модельного времени. После любого изменения состояния, т.е. при каждом событии, вызывается система вывода. Она просматривает в базе знаний и процессах продукционные правила и проверяет по предусловиям, могут ли начаться какие-либо действия. При нахождении таких действий инициируются события их начала.

Продукционная система, блок имитации событий совместно осуществляют построение имитационной модели системы. На основании анализа результатов имитации вычисляются требуемые показатели функционирования системы.

Система трассировки выводит подробную информацию о событиях в специальный файл, который затем обрабатывается для детального анализа

работы модели. Система анимации отображает на экране во время имитации поведение моделируемого объекта.

RAO-studio может быть применен для создания имитационных моделей, систем планирования, игр и тренажеров, экспертных систем реального времени и гибридных систем, включающих экспертные системы, имитационные модели и алгоритмы оптимизации.

Непрерывные процессы также могут быть описаны, поскольку формулы интегрирования переменных состояния можно записать в виде модифицированных продукции (1).

## **1.2 Функции программного комплекса**

При выполнении работ, связанных с созданием и использованием ИМ в среде РДО, пользователь оперирует следующими основными понятиями:

*Модель* – совокупность объектов языка РДО, описывающих какой-то реальный объект, собираемые в процессе имитации показатели, кадры анимации и графические элементы, используемые при анимации, результаты трассировки.

*Прогон* – это единая неделимая точка имитационного эксперимента. Он характеризуется совокупностью объектов, представляющих собой исходные данные и результаты, полученные при запуске имитатора с этими исходными данными.

*Проект* – один или более прогонов, объединенных какой-либо общей целью. Например, это может быть совокупность прогонов, которые направлены на исследование одного конкретного объекта или выполнение одного контракта на имитационные исследования по одному или нескольким объектам.

*Объект* – совокупность информации, предназначенной для определенных целей и имеющая смысл для имитационной программы.

Состав объектов обусловлен РДО-методом, определяющим парадигму представления СДС на языке РДО.

Объектами исходных данных являются:

- типы ресурсов (с расширением .rtp);
- ресурсы (с расширением .rss);
- событий (с расширением .evn);
- образцы активностей (с расширением .pat);
- точки принятия решений (с расширением .dpt);
- процессы (с расширением .prc);
- константы, функции и последовательности(с расширением .fun);
- кадры анимации (с расширением .frm, .bmp);
- требуемая статистика (с расширением .pmd);
- прогон (с расширением .smr);
- проект (с расширением .rdox).

Объекты, создаваемые РДО-имитатором при выполнении прогона:

- результаты (с расширением .pmv);
- трассировка (с расширением .trc) (2).

На данный момент РДО реализует следующие основные функции, которые представлены на 1 листе дипломного проекта (Функциональная структура РДО. Нотация ARIS):

1) Создание модели на языке РДО:

- создание основных объектов (\*.rtp, \*.rss, \*.evn, \*.pat, \*.dpt, \*.prc, \*.smr);
- создание объектов данных и функций (\*.fun);
- создание объектов вывода (\*.frm, \*.pmd, \*.bmp).

2) Проведение экспериментов:

- изменение параметров системы в процессе моделирования;

- генерация случайных чисел.

3) Вывод результатов моделирования:

- анимация объектов модели;
- вывод необходимых показателей;
- построение графиков;
- трассировка изменений объектов модели.

### **1.3 Предпосылки интеграции подходов дискретного имитационного моделирования в РДО**

А. Прицкер [1] предложил три основных подхода для описания сложных дискретных систем (СДС): процессный, событийный и сканирования активностей.

Каждый из этих подходов при моделировании дискретных систем проявляет себя по-разному: событийный — наиболее гибкий, но сложный; процессный — наоборот, наиболее простой, но не гибкий; сканирования активностей — средний по простоте и гибкости. Это привело к тому, что каждый из этих подходов наилучшим образом зарекомендовал себя при описании различных аспектов моделируемой системы:

1. процессный - процессов обслуживания;
2. событийный - внешних воздействий;
3. сканирования активностей - систем управления.

Система имитационного моделирования РДО поддерживает все три перечисленные подхода моделирования систем. Но только два из них: событийный и сканирования активностей - полностью интегрированы друг с другом. Это было достигнуто после реализации механизма управления планированием событий. Процессный подход интегрирован с остальными лишь частично — за счет использования общих ресурсов системы. Это дает возможность отслеживать в событиях и активностях изменения в системе, которые произошли в результате работы процесса обслуживания.

Но наиболее интересно обратное направление этой связи — управление процессом обслуживания из системы управления, которая наиболее часто моделируется с помощью сканирования активностей. Но, к сожалению, сейчас такой возможности нет. Это приводит к тому, что при моделировании СДС, включающих в себя СМО, нельзя использовать процессный подход со всеми его преимуществами. Это приводит к существенному увеличению кода модели, ухудшению ее читабельности, сложности в разработке и сопровождении.

Для примера, рассмотрим существующую имитационную модель системы, осуществляющей перевозку грузов по трем направлениям. База знаний этой модели:

```

$Decision_point формирование_составов_на_отправку: some
$Condition NoCheck
$Activities
    расчет_количества_вагонов :
Образец_расчета_количества_вагонов
    согласование_заявки_и_формсостава : Образец_списания_вагонов
    проверка_состояния_формсостава :
Образец_проверки_формируемого_состава
    проверка_состояния_заявки : Образец_проверки_заявки
    назначение_нового_номера_составу :
Образец_назначения_нового_номера_состава
    готовность_состава : Образец_готовности_состава
    создание_состава : Образец_создания_состава
    погрузка : Образец_погрузки
    проверка_аренды : Образец_проверки_аренды
$End

$Decision_point поездка: prior
$Condition NoCheck
$Activities
$End

$Decision_point маршрут1: prior
$Parent поездка
$Condition NoCheck

```

```

$Priority 0.1
$Activities
    путь1
    : Образец_следования_по_маршруту 1 Кульсары Аксарайская 550.0
прямое прибыл_на_таможню
    путь2
    : Образец_следования_по_маршруту 1 Аксарайская Кошта
2200.0 прямое прибыл_на_разгрузку
    путь3
    : Образец_следования_по_маршруту 1 Кошта Кульсары
2750.0 обратное прибыл_в_парк
$End

```

```

$Decision_point маршрут2: prior
$Parent поездка
$Condition NoCheck
$Priority 0.1
$Activities
    путь5
    : Образец_следования_по_маршруту 2 Кульсары Аксарайская
550.0 прямое прибыл_на_таможню
    путь6
    : Образец_следования_по_маршруту 2 Аксарайская Красное
1740.0 прямое прибыл_на_таможню
    путь7
    : Образец_следования_по_маршруту 2 Красное Брест
590.0 прямое прибыл_на_таможню
    путь8
    : Образец_следования_по_маршруту 2 Брест Малашевиче 14.0
прямое прибыл_на_разгрузку
    путь9
    : Образец_следования_по_маршруту 2 Малашевиче Брест 14.0
обратное необходимо_сменить_колею
    путь10
    : Образец_следования_по_маршруту 2 Брест Кульсары
2880.0 обратное прибыл_в_парк
$End

```

```

$Decision_point маршрут3: prior
$Parent поездка
$Condition NoCheck

```



```

$Priority 0.1
$Activities
    путь13
        : Образец_следования_по_маршруту 3 Кульсары Аксарайская
550.0 прямое прибыл_на_таможню
    путь14
        : Образец_следования_по_маршруту 3 Аксарайская Гуково
830.0 прямое прибыл_на_таможню
    путь15
        : Образец_следования_по_маршруту 3 Гуково Батеве
1650.0 прямое прибыл_на_таможню
    путь16
        : Образец_следования_по_маршруту 3 Батеве Тужер 20.0
прямое прибыл_на_разгрузку
    путь17
        : Образец_следования_по_маршруту 3 Тужер Кульсары
3050.0 обратное прибыл_в_парк
$End

```

```

$Decision_point общие_действия: prior
$Parent поездка
$Condition NoCheck
$Priority 0.1
$Activities
    таможня :
Образец_прохождения_таможни
    определение_действий_с_составом_после_таможни :
Образец_выбора_действий_после_таможни
    смена_колеи :
Образец_смены_колеи
    разгрузка :
Образец_разгрузки
    списывание_заявок :
Образец_списывания_обслуженных_заявок cf = 1
    отправка_на_мойку :
Образец_отправки_состава_на_мойку cf = 0
    мойка : Образец_мойки
    подготовка_состава_к_возврату :
Образец_подготовки_состава_к_возврату
    возврат_вагонов_в_парк :
Образец_возврата_вагонов_в_парк

```

```

        удаление_состава                                     :
Образец_удаления_состава
        удаление_заявки                                     :
Образец_удаления_заявки
        $End

```

10 из 22 активностей базы знаний этой модели описывают движение составов по заданному маршруту, т.е. заданную наперед, жесткую последовательность действий. Лучше всего для этих целей подходит процессный подход дискретного моделирования. Использование же сканирования активностей приводит к упомянутым выше проблемам.

Чтобы оценить эффект от внедрения в РДО поддержки процессного подхода одновременно с остальными, на основе имеющейся модели разработана модель-прототип (полный код модели-прототипа приведен ниже, см. ПРИЛОЖЕНИЕ 1. Код имитационной модели-прототипа грузоперевозок.). Этот прототип использует преимущества использования мультипарадигменного подхода моделирования в рамках одной модели при помощи новых управляющих конструкций внутри образца активности Образец\_создания\_состава, который создает в модели новый ресурс Состав:

```

        if (_состав.маршрут == 1)
            Движение_составов_1.ProcessStart(_состав);

```

Смысл этой инструкции в запуске процесса «Движение\_составов\_1», при выполнении соответствующего условия, и передачи ему в качестве транзакта вновь созданного ресурса «\_состав».

Вот несколько количественных сравнений: старая модель (оригинальная) имеет 26442 символа, а новая (модель-прототип) — 19522, т.е. размер кода модели сократился на 26,1%. Если проанализировать эти изменения более подробно, то описание ресурсов сократилось на 58,4% (4969 и 2066 символов соответственно), логика модели — на 29,2% (13775 и 9758 символов соответственно). Помимо количественной разницы стоит отметить, что код, описывающий непосредственно перемещение по

маршрутам стал более лаконичным и понятным, а ресурсу «состав» больше не требуется большая часть его состояний.

Таким образом, основная цель данного дипломного проекта — исправление этих недостатков путем полной интеграции всех основных подходов дискретного моделирования в системе имитационного моделирования РДО.

#### **1.4 Выводы по преддипломному этапу**

Рассмотренная выше задача моделирования демонстрирует прекрасный пример применения всех трех подходов имитационного моделирования в рамках одной модели СДС: событийного – для описания внешних воздействий на систему (поток заявок на обслуживание), сканирования активностей – для описания подсистемы управления (принятия решений) и процессного – для лаконичного и простого описания процессов обслуживания с постоянным порядком действий. Это четко поясняет необходимость интеграции разных подходов моделирования в РДО.

Взаимодействие разных частей модели потребует разработать новую концепцию передачи управления в процессы из событий, запланированных пользователем, или событий начала и окончания действий базы знаний. Также потребуется пересмотреть существующую концепцию «транзакта» – ключевого понятия процессного подхода для того, чтобы процессы могли работать со всеми ресурсами модели.

Так как система РДО имеет четко определенную иерархическую структуру, разработанную с применением современных подходов объектно-ориентированного и обобщенного проектирования, нацеленных на возможность удобного развития, масштабирования и изменения системы в будущем, есть все основания полагать, что задача интеграции дискретных подходов имитационного моделирования в РДО может быть

успешно решена. Остается определиться с выбором концепции, позволяющих достигнуть поставленных целей, затем разработать соответствующие алгоритмы функционирования системы на техническом этапе проектирования и приступить к рабочему этапу реализации системы.

## 2 КОНЦЕПТУАЛЬНОЕ ПРОЕКТИРОВАНИЕ

### 2.1 Цели разработки системы

Основной целью данного дипломного проекта является **повышение эффективности моделирования в среде РДО**, которая состоит из трех подцелей:

- сокращение времени обучения системе новых сотрудников;
- повышение производительности моделирования;
- упростить разработку и отладку моделей СДС.

В свою очередь, упрощение разработки и отладки моделей СДС также декомпозируется на 2 подцели:

- ❖ увеличение наглядности, модульности и выразительности моделей;
- ❖ возможность повторное использование разработанных ранее процессных моделей на языке GPSS.

Благодаря интеграции процессного подхода имитационного моделирования в РДО, его использование станет более привлекательным, в том числе и специалистам, знакомым с другими процессными подходами (например, GPSS), но не имеющие опыта разработки моделей на языке РДО.

Помимо этого, процессный подход привлекателен для системы и с точки зрения увеличения производительности при моделировании, потому что процессный подход менее ресурсоемок по сравнению с подходом сканирования активностей.

Из этого можно сделать вывод о том, что после интеграции процессного подхода имитационного моделирования все рассмотренные цели будут выполнены и это приведет к повышению эффективности моделирования в среде РДО.

## 2.2 Компоненты РДО

Система имитационного моделирования РДО безусловно является сложной и статически, и динамически. На это указывает сложная иерархическая структура системы со множеством различных связей между компонентами и ее сложное поведение во времени.

Ярко выраженная иерархическая структура и модульность системы определяют направление изучения системы сверху вниз. Т.е. принцип декомпозиции применяется до тех пор, пока не будет достигнут уровень абстракции, представление на котором нужных объектов не нуждается в дальнейшей детализации для решения данной задачи.

Для отображения зависимости между компонентами системы РДО и выделения среди них модернизируемых служит соответствующая диаграмма в нотации UML.

Базовый функционал представленных на диаграмме компонентов:

`rdo_kernel` реализует ядровые функции системы. Не изменяется при разработке системы.

`RAO-studio.exe` реализует графический интерфейс пользователя. Не изменяется при разработке системы.

`rdo_repository` реализует управление потоками данных внутри системы и отвечает за хранение и получение информации о модели. Не изменяется при разработке системы.

`rdo_mbuilder` реализует функционал, используемый для программного управления типами ресурсов и ресурсами модели. Не изменяется при разработке системы.

`rdo_converter` конвертирует модели созданные в старой версии РДО, производя резервное копирование файлов оригинальной модели. Благодаря ему обеспечивается обратная совместимость версий системы. Не изменяется при разработке системы.

rdo\_simulator управляет процессом моделирования на всех его этапах. Он осуществляет координацию и управление компонентами rdo\_runtime и rdo\_parser. Не изменяется при разработке системы.

rdo\_parser производит лексический и синтаксический разбор исходных текстов модели, написанной на языке РДО. Модернизируется при разработке системы.

rdo\_runtime отвечает за непосредственное выполнение модели, управление базой данных, базой знаний, планирование и выполнение событий, и работу процессов. Модернизируется при разработке системы.

Таким образом основные изменения должны затронуть модули rdo\_parser и rdo\_runtime.

## **2.3 Новые конструкции в языке РДО**

После решения поставленной задачи в распоряжении пользователя РДО должен появиться инструмент запуска процессов. Этим инструментом станет новая инструкции, доступная в теле событий и образцов активностей.

Таким образом лексический и синтаксический анализаторы компонента rdo\_parser должны начать правильно обрабатывать новые конструкции языка.

## **2.4 Ресурсы модели**

Сейчас в РДО существует три разновидности ресурсов:

1. ресурсы-транзакты – ресурсы, которые являются для процессов обслуживания транзактами;
2. процессные ресурсы – ресурсы с точки зрения процесса, они обслуживают транзакты в процессе;
3. обычные ресурсы – ресурсы для классического РДО-метода.

Ресурсы-транзакты и процессные ресурсы являются потомками обычных и просто добавляют к ним некоторые атрибуты и ограничения. Ресурсы-транзакты хранят в себе указатели на блок процесса, в котором они сейчас находятся, и указатель на процессный ресурс, который их сейчас обслуживает. Процессный ресурс не имеет дополнительных атрибутов. Однако и ресурсы-транзакты, и процессные ресурсы накладывают ограничения на свой список параметров, так ресурс-транзакт обязательно должен иметь вещественный параметр «время\_прихода», а процессный – параметр перечислимого типа «состояние».

Т.е., с точки зрения иерархии классов все эти ресурсы являются обычными ресурсами и, соответственно, могут стать релевантными любым событиям и активностям. Но только специальные ресурсы могут участвовать в процессе, и, поэтому единственное место создание ресурсов-транзактов сейчас – это блок GENERATE процесса.

Таким образом, главной проблемой, решаемой в рамках данного дипломного проекта, станет полная переработка концепции работы с ресурсами в системе и, в первую очередь, их создания.

После проведения соответствующих исследований (см. ИССЛЕДОВАТЕЛЬСКАЯ ЧАСТЬ) был сделан выбор в пользу следующей концепции:

## **2.5 Типы ресурсов модели**

После принятия решения о концепции создания ресурсов осталось предложить новую концепцию типов ресурсов.

Сейчас типы ресурсов модели можно описывать, во-первых, на закладке RTP и, во-вторых, на PRC. Второй вариант имеет крайне важное значение для упрощения процессных моделей. Таким образом в процессной модели нет описывать тип ресурса полностью, как того требует классический РДО-метод. Достаточно указать имя типа в месте его



использования в процессе и РДО сам создаст нужный тип с нужными параметрами (именно из-за этого на параметры ресурса-транзакта и процессного ресурса накладываются ограничения). Однако для увеличения гибкости модели типы ресурсов, используемые в процессе обслуживания можно описать полностью (например, чтобы расширить их список параметров) на закладке RTP. Такая гибкость приводит к тому, что компилятору моделей придется пересоздать тип ресурса, если позже выяснится, что он может участвовать в процессе.

Зато после создания всех типов ресурсов появляется возможность простым и наглядным способом, не зависящим от контекста, создавать ресурсы.

## **2.6 Разработка технического задания**

Техническое задание разработано в соответствии с ГОСТ 19.201-78 (Единая система программной документации. Техническое задание. Требования к содержанию и оформлению) (3).

### **2.6.1 Введение**

Интеграция подходов дискретного имитационного моделирования в РДО имеет большое значение для системы, которое позволит, с одной стороны, расширить инструментарий опытного разработчика моделей на РДО, с другой стороны, существенно упростить разработку простых моделей процессов обслуживания (использование для этой цели процессного подхода вместо сканирования активностей сокращает код модели, а, следовательно, и сложность, на порядок), которые со временем могут быть дополнены, например, системой управления.

### **2.6.2 Основания для разработки**

- 1) Разработка ведется на основании следующих документов:
  - Задание на выполнение дипломного проекта.

- Календарный план на выполнение дипломного проекта.

2) Документы утверждены « 09 » февраля 2011 года.

3) Тема дипломного проекта:

Интеграция подходов дискретного имитационного моделирования в РДО.

### 2.6.3 Назначение разработки

Функциональное и эксплуатационное назначение разработки интеграции подходов моделирования в РДО состоит в создании инструмента, предоставляющего пользователю возможность управлять процессной частью модели из базы знаний и событий модели.

### 2.6.4 Требования к программе или программному изделию

#### *2.6.4.1 Требования к функциональным характеристикам*

Требования к составу выполняемых функций реализуемой системы заключаются в возможности запуска и выполнения произвольного процесса обслуживания по выполнению условия, либо наступлению события.

С другой стороны, запущенный процесс так же изменяет общее состояние модели. Таким образом, каждое знание о реальной системе, отраженное в модели, вне зависимости от формы представления, способно реагировать на любые изменения состояния модели, которое характеризуется общей для модели базой данных ресурсов.

#### *2.6.4.2 Требования к надежности*

Действия пользователя не должны приводить к сбоям в работе программы.

Система должна обладать средствами самодиагностики и автоматической индикации типов ошибок.

Пользователю должен быть предоставлен интуитивно понятный интерфейс и система помощи.

#### **2.6.4.3 Условия эксплуатации**

Аппаратные средства должны эксплуатироваться в помещениях с выделенной розеточной электросетью 220В  $\pm 10\%$ , 50 Гц с защитным заземлением при следующих климатических условиях:

- температура окружающей среды – от 15 до 30 градусов С;
- относительная влажность воздуха - от 30% до 80%;
- атмосферное давление - от 630 мм. р.с. до 800 мм. р.с.

#### **2.6.4.4 Требования к составу и параметрам технических средств**

Программный продукт должен работать на компьютерах со следующими характеристиками:

- ✓ объем ОЗУ не менее 256 Мб;
- ✓ объем жесткого диска не менее 20 Гб;
- ✓ микропроцессор с тактовой частотой не менее 400 МГц;
- ✓ монитор не менее 15” с разрешением от 800\*600 и выше.

#### **2.6.4.5 Требования к информационной и программной совместимости**

Данная система должна работать под управлением операционных систем семейства Microsoft Windows для рабочих станций:

- Windows NT 4.0 Workstation,
- Windows 2000,
- Windows XP,
- Windows Vista,
- Windows 7;

а также для серверов:

- Windows NT 4.0 Server,
- Windows 2000 Server,
- Windows Server 2003,
- Windows Server 2008,
- Windows Server 2008 R2,

- Windows Home Server 2011.

#### **2.6.4.6 Требования к маркировке и упаковке**

Не предъявляются.

#### **2.6.4.7 Требования к транспортированию и хранению**

Не предъявляются.

#### **2.6.4.8 Требования к программной документации**

Не предъявляются.

### **2.6.5 Техничко-экономические показатели**

Расчет экономической эффективности разработанного приложения не является целью дипломного проектирования, однако возможный экономический эффект может быть достигнут за счет следующих преимуществ системы:

- 1) При разработке новых моделей появится возможность использования разработанных ранее процессных моделей на языке GPSS в качестве модулей разрабатываемой модели после незначительной модификации.
- 2) Использование в одной модели возможностей всех подходов ИМ позволит упростить разработку и отладку моделей сложных систем за счет увеличения модульности, наглядности, выразительности моделей.
- 3) Сокращение времени обучения системе РДО благодаря использованию наиболее распространенного и популярного процессного подхода.
- 4) Повысится производительность процесса моделирования (в связи с внедрением в имитационные модели процессных подмоделей, производительность которых выше).

## 2.6.6 Стадии и этапы разработки

Состав, содержание и сроки выполнения работ по созданию системы в соответствии с календарным планом на выполнение дипломного проекта приведены ниже (см. Таблица 2.1).

**Таблица 2.1. Стадии и этапы разработки системы.**

<b>Стадии</b>	<b>Этапы работ</b>	<b>Срок выполнения</b>
Предпроектное исследование	Предпроектное исследование РДО с точки зрения перспективы интеграции в нем всех подходов ИМ	25.02.11
Концептуальное проектирование	Разработка концепции реализации системы	15.03.11
	Разработка технического задания на систему	20.03.11
Техническое проектирование	Разработка алгоритмов функционирования системы, структуры программных средств	01.04.11
Рабочее проектирование	Программная реализация функционала системы	05.05.11
	Разработка тестового примера модели, использующей одновременно все три подхода имитационного моделирования, апробирование системы	10.05.11
Организационно-экономическая часть	Расчет затрат на разработку системы	15.05.11

Мероприятия по охране труда и технике безопасности	Требования безопасности при работе с системой	20.05.11
--	---	----------

### 2.6.7 Порядок контроля и приемки

Контроль и приемка распределенного приложения должны осуществляться в процессе проверки функциональности системы имитационного моделирования на тестовом примере распределенной модели в соответствии с требованиями к функциональным характеристикам системы.

### 2.6.8 Приложения

Документы, используемые при разработке приведены в списке использованных источников.

Используемое при разработке программное обеспечение:

- ❖ Операционная система Microsoft Windows 7 Professional;
- ❖ Среда разработки Microsoft Visual Studio 2005 Professional;
- ❖ Генератор лексических анализаторов Flex;
- ❖ Генератор синтаксических анализаторов Bison;
- ❖ Централизованная система управления версиями Subversion;
- ❖ Система документирования исходных текстов Doxygen;
- ❖ Исходные коды системы РДО.

## 3 ТЕХНИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

### 3.1 Проектирование новых синтаксических конструкций

На концептуальном этапе проектирования был сделан вывод о необходимости реализации новой инструкции языка РДО, функция которой будет заключаться в передаче релевантного ресурса активности или события в процесс.

В синтаксис этой инструкции должны быть включены имя процесса, который должен быть запущен, и имя релевантного ресурса, который должен передаться процессу в качестве транзакта:

```
<имя_процесса>.ProcessStart (<имя_релевантного_ресурса>);
```

Подробно новая инструкция показана на синтаксической диаграмме образцов активностей и событий системы.

### 3.2 Проектирование типов ресурсов

Как было предложено на концептуальном этапе проектирования, все ресурсы модели создаются исключительно с помощью их типов. Т.е. тип ресурса – это владелец всей информации о ресурсах и единственный их создатель в модели в не зависимости от этапа моделирования (инициализация модели или прогон) и контекста создания (создание временного ресурса в событии или активности, создание постоянных ресурсов, генерация ресурсов блоком GENERATE процесса).

Данная концепция представляет из себя известный паттерн объектно-ориентированного и обобщенного проектирования под названием фабричный метод. Кроме того, он уже активно используется в системе (модуль rdo\_parser) как низкоуровневая фабрика объектов. Новая же концепция типов ресурсов будет представлять из себя высокоуровневую фабрику ресурсов, которая будет пользоваться

существующей низкоуровневой, тем самым инкапсулируя в себе всю информацию о ресурсах.

Для детальной проработки решения была построена диаграмма классов ресурсов РДО. Основные идеи этой диаграммы: ресурс `rdoParse::RDORSSResource` хранит указатель на `rdoParse::RDORTPResType`, а тот в свою очередь на интерфейс `rdoRuntime::IResourceType`. Таким образом, `rdoParse::RDORSSResource` может запросить информацию об интерфейсе типа ресурса при создании `rdoRuntime::RDOCalcCreateResource`. В дальнейшем этот класс создает ресурсы во время прогона модели.

### 3.3 Проектирование процессов обслуживания

Новая концепция процессов обслуживания требует, чтобы процесс запоминал тип ресурсов-транзактов, которые могут в нем находиться (и в случае создания их блоком `GENERATE`, и в БЗ модели).

Если раньше владельцем информации о типе транзактов являлся блок `GENERATE`, то теперь этой информацией должен обладать процесс в целом, потому что блока `GENERATE` может и не быть в процессе.

Это приводит к переработки синтаксиса процессов:

```
$Process <имя_процесса> <тип_транзактов_процесса>
    <описание_блоков_процесса>
$End
```

Таким образом теперь после имени процесса необходимо указать тип ресурсов, которые станут для данного процесса транзактами.

Взаимосвязь между процессом, блоком-оператором процесса, ресурсом и типом ресурса показана на диаграмме классов процессов и ресурсов РДО.

Ресурсы модели создаются в фабриках процессов, они же типы ресурсов, реализующие интерфейс `rdoRuntime::IResourceType`.



Генерация ресурсов при работе процесса происходит благодаря связи: rdoRuntime::RDOPROCBlock – IPROCProcess – rdoRuntime::RDOPROCPProcess – rdoRuntime::IResourceType.

## 4 РАБОЧЕЕ ПРОЕКТИРОВАНИЕ

### 4.1 Реализация нового синтаксиса

Для внедрения новой синтаксической конструкции, которая передает процессу релевантный ресурс в качестве транзакта необходимо ввести в систему новое ключевое слово `ProcessStart`. Это реализуется добавлением в файл `rdo_lexer.l` такой конструкции:

```
ProcessStart          return(RDO_ProcessStart);
```

Далее необходимо обеспечить правильную обработку новой инструкции в `rdopat.y` и `rdoevn.y`:

```
process_input_statement
    : RDO_IDENTIF '.' RDO_ProcessStart '(' RDO_IDENTIF_RELRES ')'
    ;

    {
        tstring          processName      = RDOVALUE($1)-
>getIdentificator();
        LPRDOPROCPProcess  pProcess      =  PARSE-
>findPROCProcess(processName);
        if (!pProcess)
        {
            PARSE->error().error(@1, rdo::format(_T("Попытка
запустить неизвестный процесс: %s"), processName.c_str()));
        }

        LPIPROCBlock      pBlock          =      (* (pProcess-
>getBlockList().begin()))->getRuntimeBlock();
        ASSERT(pBlock);

        tstring relResName = RDOVALUE($5)->getIdentificator();

        LPRDOPATPattern  pPattern = PARSE->getLastPATPattern();
        ASSERT(pPattern);
        /*из-за использования RDO_IDENTIF_RELRES
findRelevantResource() всегда находит ресурс*/
        LPRDORelvantResource  pRelRes      =      pPattern-
>findRelevantResource(relResName);
```

```

tstring relResTypeName = pRelRes->getType()->name();

if (!pProcess->checkTransactType(relResTypeName))
{
    PARSER->error().error(@1, rdo::format(_T("Процесс
%s ожидает в качестве транзактов ресурсы типа %s, а не %s"),
processName.c_str(), _T("true_resTypeName"), relResTypeName.c_str()));
}

    rdoRuntime::LPRDOCalcProcessControl pCalc =
rdo::Factory<rdoRuntime::RDOCalcProcessControl>::create(pBlock, pRelRes-
>m_relResID);

    ASSERT(pCalc);

    $$ = PARSER->stack().push(pCalc);
}
| RDO_IDENTIF '.' RDO_ProcessStart '(' error ')' '; '
{
    PARSER->error().error(@5, _T("В качестве транзакта
процессу можно передавать только релеватный ресурс"));
}
;

```

В этом фрагменте кода происходит проверка корректности имени процесса, который должен быть запущен. Затем проверяется имеет ли данный релевантный ресурс тип, который ожидает процесс. При невыполнении этих условий пользователь получит соответствующее сообщение об ошибке. Если проверка всех условий закончилась положительно, то создается объект `rdoRuntime::RDOCalcProcessControl`

## 4.2 Изменения в компоненте имен `rdoParse`

Позднее связывание реализуется таким образом:

```

// -----
// ----- RDOParserPATPost
// -----
void RDOParserPATPost::parse()
{
    //! Позднее связывание для планирования событий
    STL_FOR_ALL_CONST(RDOParser::EventList, m_parser->getEvents(), eventIt)
    {
        LPRDOEvent pEvent = *eventIt;
    }
}

```

```

CPTR(RDOPATPattern) pPattern = m_parser->findPATPattern(pEvent->name());
if (!pPattern)
{
    STL_FOR_ALL_CONST(RDOEvent::CalcList, pEvent->getCalcList(), calcIt)
    {
        m_parser->error().push_only((*calcIt)->src_info(),
rdo::format(_T("Попытка запланировать неизвестное событие: %s"), pEvent->name().c_str()));
    }
    m_parser->error().push_done();
}
if (pPattern->getType() != RDOPATPattern::PT_Event)
{
    STL_FOR_ALL_CONST(RDOEvent::CalcList, pEvent->getCalcList(), calcIt)
    {
        m_parser->error().push_only((*calcIt)->src_info(),
rdo::format(_T("Паттерн %s не является событием: %s"), pEvent->name().c_str()));
    }
    m_parser->error().push_done();
}

LPIBaseOperation pRuntimeEvent =
static_cast<PTR(rdoRuntime::RDOPatternEvent)>(pPattern->getPatRuntime())-
>createActivity(m_parser->runtime()->m_metaLogic, m_parser->runtime(), pEvent->name());
ASSERT(pRuntimeEvent);
pEvent->setRuntimeEvent(pRuntimeEvent);

STL_FOR_ALL_CONST(RDOEvent::CalcList, pEvent->getCalcList(), calcIt)
{
    (*calcIt)->setEvent(pRuntimeEvent);
}
}
}

```

Здесь сначала, создается событие в пространстве имен `rdoRuntime`, а затем ссылка на вновь созданное событие сохраняется во всех объектах-вычислителях, которые хранятся в объекте `rdoParse::RDOEvent`.

## 5 ИССЛЕДОВАТЕЛЬСКАЯ ЧАСТЬ

### 5.1 Альтернативные решения

Основной задачей дипломного проекта является интеграция процессного подхода имитационного моделирования с классическим РДО-методом.

Данная интеграция заключается в возможности передачи в процесс ресурса, созданного во время выполнения активностей базы знаний или событий, в качестве транзакта.

Как было установлено ранее (см. Ресурсы модели) ресурс-транзакты наследуются от обычных ресурсов и имеют дополнительные атрибуты. Это значит, что либо во время передачи в процесс ресурса, созданного вне него, ресурс должен «оборачиваться» в некоторую обертку (представляющую из себя дельту между этими ресурсами) и становиться ресурсом-транзактом, либо он должен изначально быть ресурсом-транзактом (а это не будет мешать пользоваться им в базе знаний и событиях).

Назовем первую предложенное решение «пересоздание ресурсов», а второе – «подготовка правильных типов».

Таким образом данная задача имеет два альтернативных решения, которые необходимо исследовать и сделать аргументированный выбор в пользу обного из них.

### 5.2 Сравнение альтернативных решений

Основным плюсом первого решения является небольшой масштаб изменений в существующей концепции создания ресурсов в РДО. К его недостаткам следует отнести разрозненный подход к работе с ресурсами и, как следствие, проблемы масштабируемости в дальнейшем.

Реализация второго решения потребует фундаментальных изменений в системе: создание объектов компонента `rdo_runtime` с помощью фабрики объектов, пересоздание типа ресурсов, реализации концепции «тип ресурса = фабрика ресурса». С другой стороны все это приводит к улучшению дизайна системы, увеличению модульности и абстракции данных.

### **5.3 Основной критерий сравнения**

Основной критерий сравнения предложенных альтернатив должен учитывать основную цель, преследуемую при создании имитационных моделей.

ИМ применяется в основном для решения задач анализа и синтеза сложных систем. Как правило, в имитационных моделях масштабный коэффициент времени много больше единицы. Причина этого заключается в требованиях, предъявляемых к моделям. Это не только адекватность, но и быстродействие.

Таким образом, основной критерий сравнения альтернативных решений – это быстродействие во время прогона имитационной модели.

Решение «пересоздание ресурса» по своей архитектуре связано с дополнительными временными затратами во время прогона на выделение памяти для нового ресурса.

Решение «подготовка правильных типов» не страдает подобным недостатком. С другой стороны оно приводит, во-первых, к большему времени создания типов ресурсов, и, во-вторых, к выделению лишней памяти (в том случае, если ресурс-транзакт во время прогона не передается в процесс). Однако перерасход памяти не представляет проблемы из-за малой стоимости хранения указателя, а время компиляции модели – это незначительная составляющая всего прогона.

## **5.4 Выводы по результатам исследований**

Решение задачи связанное с переработкой концепции создания ресурсов не только является предпочтительной с точки зрения общей архитектуры, но еще и не приводит к дополнительным временным затратам на этапе выполнения прогона имитационной модели.

Поэтому именно это решение было выбрано как основное решение поставленной задачи, несмотря на более глубокие, значительные и долговременные изменения в архитектуре РДО.

## 6 ОРГАНИЗАЦИОННО-ЭКОНОМИЧЕСКАЯ ЧАСТЬ

Современная инженерная деятельность предполагает не только разработку современных конструкций и технологий, но также и концентрацию усилий специалиста, позволяющую заранее определить возможный рынок реализации разработки, оценить ожидаемую прибыль. Поэтому важной составляющей любого инженерного проекта является раздел, посвященный анализу экономических характеристик и определению экономических параметров, позволяющих сделать вывод о возможности реализации инженерной мысли.

В организационно-экономической части дипломного проекта происходит расчет стоимости интеграции подходов дискретного имитационного моделирования в РДО.

### 6.1 Организация и планирование процесса разработки программного продукта

Интеграция подходов дискретного имитационного моделирования в РДО – сложный и длительный процесс, требующий выполнения большого числа разнообразных операций.

Организация и планирование процесса разработки программного продукта или программного комплекса при традиционном методе планирования предусматривает выполнение следующих работ:

1. формирование состава выполняемых работ и группировка их по стадиям разработки;
2. расчет трудоемкости выполнения работ;
3. установление профессионального состава и расчет количества исполнителей;
4. определение продолжительности выполнения отдельных этапов разработки;



5. построение календарного графика выполнения разработки;
6. контроль выполнения календарного графика.

Ниже приведен перечень стадий и состава работ (Таблица 6.1) при создании программного продукта (3).

**Таблица 6.1. Перечень стадий и состава работ.**

Стадия разработки программного продукта	Состав выполняемых работ
Техническое задание	Постановка задач, выбор критериев эффективности. Разработка технико-экономического обоснования разработки. Определение состава пакета прикладных программ, состава и структуры информационной базы. Предварительный выбор методов выполнения работы. Разработка календарного плана выполнения работ.
Эскизный проект	Предварительная разработка структуры входных и выходных данных. Разработка общего описания алгоритмов реализации решения задач. Разработка пояснительной записки. Консультации разработчиков постановки задач. Согласование и утверждение эскизного проекта.
Технический проект	Разработка алгоритмов решения задач. Разработка пояснительной записки. Согласование и утверждение технического проекта. Разработка структуры программы. Разработка программной документации и передача ее для включения в технический проект. Уточнение структуры, анализ и определение формы представления входных и выходных данных. Выбор конфигурации технических средств.
Рабочий проект	Комплексная отладка задач и сдача в опытную эксплуатацию. Разработка проектной документации. Программирование и отладка программ. Описание контрольного примера. Разработка программной документации. Разработка, согласование программы и методики испытаний. Предварительное проведение всех видов испытаний.
Внедрение	Подготовка и передача программной документации для сопровождения с оформлением соответствующего Акта приема-сдачи работ. Проверка алгоритмов и программ решения задач, корректировка документации после опытной эксплуатации программного продукта.

Трудоемкость разработки программной продукции зависит от ряда факторов, основными из которых являются следующие: степень новизны

разрабатываемого программного комплекса, сложность алгоритма его функционирования, объем используемой информации, вид ее представления и способ обработки, а также уровень используемого алгоритмического языка программирования.

По степени новизны разрабатываемый программный продукт может быть отнесен к группе новизны «Б» (разработка ПП, не имеющей аналогов, в том числе разработка ППП).

По степени сложности алгоритма функционирования – к 1 группе сложности (реализующие оптимизационные и моделирующие алгоритмы).

По виду представления исходной информации ПП относится к группе 11 – исходная информация представлена в форме документов, имеющих различный формат и структуру. Требуется учитывать взаимовлияние показателей в различных документах.

По структуре выходной информации относим программный продукт к группе 22 – требуется вывод на печать одинаковых документов, вывод информационных носителей на машинные носители.

$t_{пп}$  - трудоемкость разработки программного продукта определяется как сумма величин трудоемкости выполнения отдельных стадий разработки. Формула для определения трудоемкости разработки ПП имеет вид:

$$t_{пп} = t_{тз} + t_{эп} + t_{тп} + t_{рп} + t_{в},$$

где  $t_{тз}$  - трудоемкость разработки технического задания на ПП;

$t_{эп}$  - трудоемкость разработки эскизного проекта ПП;

$t_{тп}$  - трудоемкость разработки технического проекта ПП;

$t_{рп}$  - трудоемкость разработки рабочего проекта ПП;

$t_{в}$  - трудоемкость внедрения разработанного ПП.

### 6.1.1 Расчет трудоемкости разработки технического задания

Трудоемкость разработки технического задания рассчитывается по формуле:  $t_{ТЗ} = t_{РЗ} + t_{РП}$ ,

где  $t_{РЗ}$  - затраты времени разработчика постановки задачи на разработку ТЗ, чел.-дн.;

$t_{РП}$  - затраты времени разработчика ПП на разработку ТЗ, чел.-дн.

$$t_{РЗ} = t_3 \cdot K_{РЗ},$$

$$t_{РП} = t_3 \cdot K_{РП},$$

где  $t_3$  - норма времени на разработку ТЗ на ПП, чел.-дн.

Исходя из рекомендаций (3) в данном случае  $t_3 = 12$  чел.-дн. (Группа новизны ПП - Б; функциональное назначение ПП – реализующие оптимизационные и моделирующие алгоритмы);

$K_{РЗ}$  - коэффициент, учитывающий удельный вес трудоёмкости работ, выполняемых разработчиком постановки задачи на стадии ТЗ. В случае совместной с разработчиком ПО разработки:  $K_{РЗ} = 0,65$ ;

$K_{РП}$  - коэффициент, учитывающий удельный вес трудоемкости работ, выполняемых разработчиком ПП на стадии ТЗ. В случае совместной с разработчиком ПО разработки:  $K_{РП} = 0,35$ .

$$t_{ТЗ} = 12 \cdot 0,65 + 12 \cdot 0,35 = 12 \text{ чел.-дн.}$$

### 6.1.2 Расчет трудоемкости выполнения эскизного проекта

Трудоемкость разработки эскизного проекта  $t_{ЭП}$  рассчитывают по формуле:  $t_{ЭП} = t_{РЗ} + t_{РП}$ ,

где  $t_{РЗ}$  - затраты времени разработчика постановки задач на разработку ЭП, чел.-дн.;

$t_{\text{ПП}}$  - затраты времени разработчика ПП на разработку ЭП, чел.-дн.;

$$t_{\text{РЗ}} = t_{\text{Э}} \cdot K_{\text{РЗ}},$$

$$t_{\text{ПП}} = t_{\text{Э}} \cdot K_{\text{ПП}},$$

где  $t_{\text{Э}}$  - норма времени на разработку ЭП программного продукта, чел.-дн.

$$t_{\text{Э}} = 26 \text{ чел.-дн.};$$

$K_{\text{РЗ}}$  - коэффициент, учитывающий удельный вес трудоемкости работ, выполняемых разработчиком постановки задачи на стадии ЭП. В случае совместной с разработчиком ПО разработки ЭП:  $K_{\text{РЗ}} = 0,7$ ;

$K_{\text{ПП}}$  - коэффициент, учитывающий удельный вес трудоемкости работ, выполняемых разработчиком ПП на стадии ЭП. В случае совместной с разработчиком постановки задач разработки ЭП:  $K_{\text{ПП}} = 0,3$ ;

$$t_{\text{ЭП}} = 26 \cdot 0,7 + 26 \cdot 0,3 = 26 \text{ чел.-дн.}$$

### 6.1.3 Расчет трудоемкости выполнения технического проекта

Трудоемкость разработки технического проекта зависит от функционального назначения ПП, количества разновидностей форм входной и выходной информации и определяется как сумма времени, затраченного разработчиком постановки задач и разработчиком ПП, и определяется по формуле:  $t_{\text{ТП}} = (t_{\text{РЗ}} + t_{\text{ПП}}) \cdot K_{\text{В}} \cdot K_{\text{Р}}$ ,

где  $t_{\text{РЗ}}$ ,  $t_{\text{ПП}}$  – норма времени, затрачиваемого на разработку ТП разработчиком постановки задач и разработчиком ПО:

$$t_{\text{РЗ}} = 19 \text{ ч.},$$

$$t_{\text{ПП}} = 5 \text{ ч.}$$

$K_{\text{Р}}$  - коэффициент учёта режима обработки информации:

$$K_{\text{Р}} = 1.45$$

$K_B$  - коэффициент учёта вида используемой информации, определяется из выражения:

$$K_B = \frac{K_{II} \cdot N_{II} + K_{HC} \cdot N_{HC} + K_B \cdot N_B}{N_{II} + N_{HC} + N_B},$$

где  $K_{II}$ ,  $K_{HC}$ ,  $K_B$  - значения коэффициентов учёта вида используемой информации для переменной, нормативно-справочной информации и баз данных соответственно;

$N_{II}$ ,  $N_{HC}$ ,  $N_B$  - количество наборов данных для переменной, нормативно-справочной информации и баз данных соответственно.

Для группы новизны Б:

$$K_{II} = 1,2 \quad N_{II} = 7$$

$$K_{HC} = 1,08 \quad N_{HC} = 1$$

$$K_B = 3,12 \quad N_B = 2$$

$$\text{Таким образом, } K_B = \frac{1,2 \cdot 2 + 1,08 \cdot 1 + 3,12 \cdot 2}{2 + 1 + 2} = 1,944$$

$$t_{TII} = (19 + 5) \cdot 1,944 \cdot 1,45 = 67,65 \text{ чел.-дн.}$$

#### 6.1.4 Расчет трудоемкости разработки рабочего проекта

Трудоёмкость разработки рабочего проекта зависит от функционального назначения ПП, количества разновидностей форм входной и выходной информации, сложности алгоритма функционирования, сложности контроля информации, степени использования готовых программных модулей, уровня алгоритмического языка программирования и определяется по формуле:

$$t_{ПП} = K_K \cdot K_P \cdot K_{Я} \cdot K_3 \cdot K_{ИА} \cdot (t_{P3} + t_{ПП}),$$

где  $K_P$  - коэффициент учёта режима обработки информации = 1,45;

$K_K$  - коэффициент учёта сложности контроля информации = 1,16;

$K_{я}$  - коэффициент учёта уровня алгоритмического языка программирования = 1.0 интерпретаторы);

$K_з$  - коэффициент учёта степени использования готовых программных модулей – 0.8;

(на 20 - 25% использования готовых прогр. Модулей);

$K_{ИА}$  - коэффициент учёта вида используемой информации и сложности алгоритма ПП.

Значение коэффициента  $K_{ИА}$  определяют из выражения:

$$K_{ИА} = \frac{K'_П \cdot N_{П} + K'_{НС} \cdot N_{НС} + K'_Б \cdot N_{Б}}{N_{П} + N_{НС} + N_{Б}},$$

где  $K'_П, K'_{НС}, K'_Б$  - значения коэффициентов учёта сложности алгоритма ПП и вида используемой информации для переменной, нормативно-справочной информации и баз данных соответственно.

В нашем случае присутствует только один вид используемой информации – переменной; соответствующее группе новизны Б значение коэффициента  $K'_П = 1.62$   $K'_{НС} = 0,97$   $K'_Б = 0.81$

(группа новизны - Б)

$$K_{ИА} = \frac{1,62 \cdot 2 + 0,97 \cdot 1 + 0,81 \cdot 2}{2 + 1 + 2} = 1,166$$

$t_{РЗ}$ ,  $t_{РП}$  - норма времени, затраченного на разработку РП на алгоритмическом языке высокого уровня разработчиком постановки задач и разработчиком ПП.

$$t_{РЗ} = 6$$

$$t_{РП} = 35$$

$$t_{РП} = 1,16 \cdot 1,45 \cdot 1,0 \cdot 0,8 \cdot 1,166 \cdot (6 + 35) = 80,41 \text{ чел.-дн.}$$

### 6.1.5 Расчет трудоемкости выполнения внедрения

Трудоёмкость выполнения стадии внедрения может быть рассчитана по формуле:

$$t_B = (t_{PЗ} + t_{PП}) \cdot K_K \cdot K_P \cdot K_З,$$

где  $t_{PЗ}$ ,  $t_{PП}$  - норма времени, затраченного разработчиком постановки задач и разработчиком ПО на выполнение процедур внедрения ПП.

$$t_{PЗ} = 6$$

$$t_{PП} = 7$$

$$t_B = (6 + 7) \cdot 1,16 \cdot 1,45 \cdot 0,8 = 17,49 \text{ чел.-дн.}$$

### 6.1.6 Расчет суммарной трудоемкости

Таким образом, суммарная трудоемкость разработки программной продукции:

$$t_{III} = 12 + 26 + 67,65 + 80,41 + 17,49 = 203,55 \text{ чел.-дн.}$$

Трудоемкость всех этапов разработки программного продукта представлена наже (см. Таблица 6.2).

**Таблица 6.2. Трудоемкость этапов разработки ПП.**

Стадия разработки	Трудоемкость чел.-дн.
Техническое задание	12
Эскизный проект	26
Технический проект	67,65
Рабочий проект	80,41
Внедрение	17,49

Для целей контроля и планирования выполнения работ в данном случае используем ленточный график, потому что разработку осуществляет небольшой, стабильный по составу коллектив исполнителей. Для построения ленточного графика необходимо знать срок начала работ,

срок окончания работ и количество работников, участвующих на каждом этапе разработки.

Продолжительность выполнения всех работ по этапам разработки программного продукта определяется из выражения

$$T_i = \frac{\tau_i + Q}{n_i},$$

где  $\tau_i$  - трудоемкость  $i$ -й работы, чел.-дн.;

$Q$  - трудоемкость дополнительных работ, выполняемых исполнителем, чел.-дн.;

$n_i$  - количество исполнителей, выполняющих  $i$ -ю работу, чел.

Пусть разработка системы ведется одним специалистом, не привлекаемым к дополнительным работам, то продолжительность разработки программного продукта:

$$T = 203,55/1 = 203,55 \text{ раб.дня.}$$

Для построения календарного плана необходимо перевести рабочие дни в календарные. Для этого длительность каждого этапа нужно разделить на поправочный коэффициент  $K=0.7$ . В результате получим:

$$T = \frac{203,55}{0.7} = 291 \text{ кал.дн}$$

$$T_{ТЗ} = \frac{12}{0.7} = 17 \text{ кал.дн}$$

$$T_{Эп} = \frac{26}{0,7} = 37 \text{ кал.дн}$$

$$T_{ПП} = \frac{67,65}{0,7} = 97 \text{ кал.дн}$$

$$T_{РП} = \frac{80,41}{0,7} = 115 \text{ кал.дн}$$

$$T_{В} = \frac{17,49}{0,7} = 25 \text{ кал.дн}$$



## 6.2 Определение стоимости разработки ПП

Для определения стоимости работ, необходимо на основании плановых сроков выполнения работ и численности исполнителей, рассчитать общую сумму затрат на разработку программного продукта.

Себестоимость ПП представляет собой стоимостную оценку используемых в процессе производства продукции работ, услуг, природных ресурсов, сырья материалов, топлива, энергии, основных фондов, трудовых ресурсов, а также других затрат на ее производство и реализацию.

Затраты, образующие себестоимость ПП, группируются в соответствии с их экономическим содержанием по следующим элементам:

- Расчёт основной заработной платы;
- Расчёт дополнительной заработной платы;
- Отчисления в социальный фонд;
- Накладные расходы.

### 6.2.1 Расчёт основной заработной платы

В статью включается основная заработная плата всех исполнителей, непосредственно занятых разработкой данного ПП, с учётом их должностного оклада и времени участия в разработке. Расчёт ведётся по

формуле: 
$$C_{zo} = \sum_i \frac{Z_i}{d} \cdot \tau_i,$$

где  $Z_i$  - среднемесячный оклад  $i$ -го исполнителя, руб.;

$d$  – среднее количество рабочих дней в месяце;

$\tau_i$  - трудоемкость работ, выполняемых  $i$ -м исполнителем, чел.-дн.

(определяется из календарного плана-графика).

$Z_i = 25000$  руб.;

$d = 21$ ;

$$\tau_i = 203,55;$$

$$C_{зо} = \frac{25000}{21} \cdot 203,55 = 241650 \text{ руб.}$$

### 6.2.2 Расчёт дополнительной заработной платы

В статье учитываются все выплаты непосредственным исполнителям за время (установленное законодательством), непроработанное на производстве, в том числе: оплата очередных отпусков, компенсация за неиспользованный отпуск, оплата льготных часов подросткам и др. Расчет ведётся по формуле:

$$C_{зд} = C_{зо} \cdot \alpha_{\partial},$$

где  $\alpha_{\partial}$  - коэффициент на дополнительную заработную плату.

$$\alpha_{\partial} = 0,2$$

$$C_{зд} = 241650 \cdot 0,2 = 48330 \text{ руб.}$$

### 6.2.3 Отчисления на социальное страхование

В статье учитываются отчисления в бюджет социального страхования по установленному законодательному тарифу от суммы основной и дополнительной заработной платы, т.е.

$$C_{сс} = \alpha_{сс} \cdot (C_{зо} + C_{зд});$$

$$\alpha_{\partial} = 0,37;$$

$$C_{зд} = (241650 + 48330) \cdot 0,37 = 107292,2.$$

### 6.2.4 Накладные расходы

В статье учитываются затраты на общехозяйственные расходы, непроизводительные расходы и расходы на управление. Накладные расходы определяют в процентном отношении к основной заработной плате, т.е.

$$C_{н} = C_{зо} \cdot \alpha_{н};$$

$$\alpha_n = 1,8;$$

$$C_{30} = 241650 \cdot 1,8 = 434970 \text{ руб.}$$

### 6.2.5 Расходы на амортизацию оборудования

Затраты, связанные с использованием вычислительной техники, определяются по формуле:  $C_{ЭВМ} = t^{ЭВМ} \cdot K_{И}^{ЭВМ} \cdot Ц^{ЭВМ} \cdot K_{БД}^{ЭВМ} \cdot K_{Э}^{ЭВМ}$ ,

где  $t^{ЭВМ}$  - время использования ЭВМ для разработки данного ПП, ч;

$K_{И}^{ЭВМ}$  - поправочный коэффициент учета времени использования ЭВМ;

$Ц^{ЭВМ}$  - цена 1-го часа работы ЭВМ;

$K_{БД}^{ЭВМ}$  - коэффициент учета степени использования СУБД;

$K_{Э}^{ЭВМ}$  - коэффициент учета быстродействия ЭВМ.

$$t^{ЭВМ} = 77;$$

$$K_{И}^{ЭВМ} = 1,26;$$

$$Ц^{ЭВМ} = 50;$$

$$K_{БД}^{ЭВМ} = 1,0;$$

$$K_{Э}^{ЭВМ} = 1,0;$$

$$C_{ЭВМ} = 77 \cdot 1,26 \cdot 50 \cdot 1,0 \cdot 1,0 = 4950 \text{ руб.}$$

### 6.2.6 Результаты расчетов затрат на разработку программного продукта.

Результаты расчетов приведены ниже (см. Таблица 6.3).

Таблица 6.3. Стоимость работ на каждом этапе разработки ПП.

Наименование статьи	Сметная стоимость, руб
Основная заработная плата	241650

Дополнительная заработная плата	48330
Отчисления на социальное страхование	107292,2
Накладные расходы	434970
Расходы на амортизацию оборудования	4950
Итого	837192,2

### **6.3 Определение стоимости разработки системы**

Произведенный расчет показал:

- Суммарная трудоемкость продукта составляет 203 чел.дней. Поэтому на каждом этапе необходимо привлечение нескольких специалистов.
- Длительность разработки программного продукта составляет 291 календарный день.
- Себестоимость программного продукта составляет 837192,2 руб.

## 7 МЕРОПРИЯТИЯ ПО ОХРАНЕ ТРУДА И ТЕХНИКЕ БЕЗОПАСНОСТИ

### 7.1 Опасные и вредные факторы

Опасные факторы:

- пожарная опасность, обусловленная наличием на рабочем месте мощного источника энергии;
- повышенное значение напряжения в электрической цепи, замыкание которой может произойти через тело человека.

Вредные факторы:

#### 7.1.1 Физические

повышенные уровни электромагнитного излучения;

повышенные уровни рентгеновского излучения;

повышенные уровни ультрафиолетового излучения;

повышенный уровень инфракрасного излучения;

повышенный уровень статического электричества;

повышенные уровни запыленности воздуха рабочей зоны;

повышенное содержание положительных аэроионов в воздухе рабочей зоны;

пониженное содержание отрицательных аэроионов в воздухе рабочей зоны;

пониженная или повышенная влажность воздуха рабочей зоны;

пониженная или повышенная подвижность воздуха рабочей зоны;

повышенный уровень шума;

повышенный или пониженный уровень освещенности;

повышенный уровень прямой блескости;

повышенный уровень отраженной блескости;

повышенный уровень ослепленности;  
неравномерность распределения яркости в поле зрения;  
повышенная яркость светового изображения;  
повышенный уровень пульсации светового потока;

### 7.1.2 Химические

повышенное содержание в воздухе рабочей зоны двуокиси углерода, озона, аммиака, фенола, формальдегида и полихлорированных бифенилов;

### 7.1.3 Психофизиологические

напряжение зрения;  
напряжение внимания;  
интеллектуальные нагрузки;  
эмоциональные нагрузки;  
длительные статические нагрузки;  
монотонность труда;  
большой объем информации обрабатываемой в единицу времени;  
нерациональная организация рабочего места.

### 7.1.4 Биологические

повышенное содержание в воздухе рабочей зоны микроорганизмов.

## 7.2 Требования к помещениям для работы с ПЭВМ

Помещения для эксплуатации ПЭВМ должны иметь естественное и искусственное освещение. Эксплуатация ПЭВМ в помещениях без естественного освещения допускается только при соответствующем обосновании и наличии положительного санитарно-эпидемиологического заключения, выданного в установленном порядке.

Естественное и искусственное освещение должно соответствовать требованиям действующей нормативной документации. Окна в

помещениях, где эксплуатируется вычислительная техника, преимущественно должны быть ориентированы на север и северо-восток. Оконные проемы должны быть оборудованы регулируемыми устройствами типа: жалюзи, занавесей, внешних козырьков и др.

Площадь на одно рабочее место пользователей ПЭВМ с ВДТ на базе электроннолучевой трубки (ЭЛТ) должна составлять не менее 6 м<sup>2</sup>, в помещениях культурно-развлекательных учреждений и с ВДТ на базе плоских дискретных экранов (жидкокристаллические, плазменные) - 4,5 м<sup>2</sup>. При использовании ПЭВМ с ВДТ на базе ЭЛТ (без вспомогательных устройств - принтер, сканер и др.), отвечающих требованиям международных стандартов безопасности компьютеров, с продолжительностью работы менее 4-х часов в день допускается минимальная площадь 4,5 м<sup>2</sup> на одно рабочее место пользователя (взрослого и учащегося высшего профессионального образования).

Для внутренней отделки интерьера помещений, где расположены ПЭВМ, должны использоваться диффузно-отражающие материалы с коэффициентом отражения для потолка - 0,7 - 0,8; для стен - 0,5 - 0,6; для пола - 0,3 - 0,5.

Полимерные материалы используются для внутренней отделки интерьера помещений с ПЭВМ при наличии санитарно-эпидемиологического заключения.

Помещения, где размещаются рабочие места с ПЭВМ, должны быть оборудованы защитным заземлением (занулением) в соответствии с техническими требованиями по эксплуатации.

Не следует размещать рабочие места с ПЭВМ вблизи силовых кабелей и вводов, высоковольтных трансформаторов, технологического оборудования, создающего помехи в работе ПЭВМ.

### 7.2.1 Требования к микроклимату, содержанию аэроионов и вредных химических веществ в воздухе на рабочих местах, оборудованных ПЭВМ

В производственных помещениях, в которых работа с использованием ПЭВМ является основной и связана с нервно-эмоциональным напряжением, должны обеспечиваться оптимальные параметры микроклимата для категории работ 1а в соответствии с действующими санитарно-эпидемиологическими нормативами микроклимата производственных помещений. На других рабочих местах следует поддерживать параметры микроклимата на допустимом уровне, соответствующем требованиям указанных выше нормативов.

В производственных помещениях, в которых работа с использованием ПЭВМ является основной, величины показателей микроклимата на рабочих местах должны соответствовать действующим санитарным нормам (СанПиН 2.2.4.548-96 Гигиенические требования к микроклимату производственных помещений). При использовании данной подсистемы характер труда оператора ПЭВМ относится к категории 1а, т.к. работы производятся сидя и не требуют физического напряжения, расход энергии составляет до 120 ккал/ч, следовательно, величины показателей микроклимата на рабочих местах в ПЭО должны соответствовать оптимальным величинам показателей для категории работ 1а (см. Таблица 7.1).

**Таблица 7.1. Оптимальные величины показателей микроклимата.**

Период года	Категория работ	Температура воздуха, °С не более	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
Холодный	Легкая – 1а	22-24	40-60	0,1
Теплый	Легкая – 1а	23-25	40-60	0,1

Уровни положительных и отрицательных аэроионов в воздухе помещений, где расположены ПЭВМ, должны соответствовать



действующим санитарно-эпидемиологическим нормативам (см. Таблица 7.2).

**Таблица 7.2. Уровни положительных и отрицательных аэроионов.**

Уровни	Число ионов в 1 см. куб. воздуха	
	n+	n-
Минимально необходимые	400	600
Оптимальные	1500-3000	3000-5000
Максимально допустимые	50000	50000

Содержание вредных химических веществ в воздухе производственных помещений, в которых работа с использованием ПЭВМ является основной, не должно превышать предельно допустимых концентраций вредных веществ в воздухе рабочей зоны в соответствии с действующими гигиеническими нормативами (Гигиенические нормативы ГН 2.1.6.13 1338-03 Предельно допустимые концентрации (ПДК) загрязняющих веществ в атмосферном воздухе населенных мест). ПДК среднесуточная: окись углерода –  $3,0 \text{ мг/м}^3$ , окислы азота –  $0,04 \text{ мг/м}^3$ .

Данный микроклимат может обеспечиваться системой общеобменной приточно-вытяжной вентиляции. В приточной ветви нужно предусмотреть использование фильтров очистки воздуха, для удаления из воздуха вредных веществ перед его подачей на рабочие места. Для повышения влажности воздуха в помещениях с ВДТ и ПЭВМ следует применять увлажнители воздуха, заправляемые ежедневно дистиллированной или прокипяченной питьевой водой. Также в помещениях, оборудованных ПЭВМ, должна проводиться ежедневная влажная уборка и систематическое проветривание после каждого часа работы на ПЭВМ.

Для поддержания необходимого уровня ионизации необходимо применять ионизаторы воздуха (например, ионизатор «Люстра Чижевского»).

## 7.2.2 Требования к уровням шума и вибрации на рабочих местах, оборудованных ПЭВМ

В производственных помещениях при выполнении основных или вспомогательных работ с использованием ПЭВМ уровни шума на рабочих местах не должны превышать предельно допустимых значений, установленных для данных видов работ в соответствии с действующими санитарно-эпидемиологическими нормативами (*СанПиН 2.2.2/2.4.1340-03 Гигиенические требования к персональным электронно-вычислительным машинам и организации работы*) (см. Таблица 7.3).

Таблица 7.3. Допустимые значения уровней шума.

Уровни звукового давления, дБ, в октавных полосах со среднегеометрическими частотами, Гц									Уровни звука в дБА
31,5	63	125	250	500	1000	2000	4000	8000	
86	71	61	54	49	45	42	40	38	50

Снизить уровень шума в помещениях с ВДТ и ПЭВМ можно использованием подвесных акустических звукопоглощающих панелей с максимальными коэффициентами звукопоглощения в области частот 250-1000 Гц (источниками шума являются операторы ПЭВМ). Дополнительным звукопоглощением служат однотонные занавеси из плотной ткани, гармонирующие с окраской стен и подвешенные в складку на расстоянии 15 - 20 см от ограждения. Ширина занавеси должна быть в 2 раза больше ширины окна.

Шумящее оборудование (печатающие устройства, серверы и т.п.), уровни шума которого превышают нормативные, должно размещаться вне помещений с ПЭВМ.

В помещениях, в которых работа с ПЭВМ является основной, уровень вибрации на рабочих местах не должен превышать допустимых значений для жилых и общественных зданий в соответствии с действующими санитарно-эпидемиологическими нормативами

(Санитарные нормы СН 2.2.4/2.1.8.566-96 Производственная вибрация, вибрация в помещениях жилых и общественных зданий) (см. Таблица 7.4).

**Таблица 7.4. Допустимые значения уровней вибрации.**

Среднегеометрические частоты октавных полос, Гц	Допустимые значения оси X, Y			
	по виброускорению		по виброскорости	
	$\text{м/с}^2 \times 10^{-3}$	дБ	$\text{м/с} \times 10^{-3}$	дБ
2	10	80	0,79	84
4	11	81	0,45	79
8	14	83	0,28	75
16	28	89	0,28	75
31,5	56	95	0,28	75
63	110	101	0,28	75
Корректированные значения и их уровни	10	80	0,28	75

Для снижения вибрации в помещениях оборудование, аппараты и приборы, являющиеся источниками вибрации, необходимо устанавливать на амортизирующие прокладки. Также могут быть использованы средства индивидуальной защиты.

### 7.2.3 Требования к освещению на рабочих местах, оборудованных ПЭВМ

Рабочие столы следует размещать таким образом, чтобы видеодисплейные терминалы были ориентированы боковой стороной к световым проемам, и естественный свет падал преимущественно слева.

Искусственное освещение в помещениях для эксплуатации ПЭВМ должно осуществляться системой общего равномерного освещения. В помещениях, в случаях преимущественной работы с документами, следует применять системы комбинированного освещения (к общему освещению дополнительно устанавливаются светильники местного освещения, предназначенные для освещения зоны расположения документов).

Освещенность на поверхности стола в зоне размещения рабочего документа должна быть 300 - 500 лк. Освещение не должно создавать

бликов на поверхности экрана. Освещенность поверхности экрана не должна быть более 300 лк.

Следует ограничивать прямую блескость от источников освещения, при этом яркость светящихся поверхностей (окна, светильники и др.), находящихся в поле зрения, должна быть не более 200 кд/м<sup>2</sup>.

Следует ограничивать отраженную блескость на рабочих поверхностях (экран, стол, клавиатура и др.) за счет правильного выбора типов светильников и расположения рабочих мест по отношению к источникам естественного и искусственного освещения, при этом яркость бликов на экране ПЭВМ не должна превышать 40 кд/м<sup>2</sup> и яркость потолка не должна превышать 200 кд/м<sup>2</sup>.

Показатель дискомфорта в помещениях – не более 15.

Яркость светильников общего освещения в зоне углов излучения от 50 до 90 градусов с вертикалью в продольной и поперечной плоскостях должна составлять не более 200 кд/м<sup>2</sup>, защитный угол светильников должен быть не менее 40 градусов.

Светильники местного освещения должны иметь не просвечивающий отражатель с защитным углом не менее 40 градусов.

Следует ограничивать неравномерность распределения яркости в поле зрения пользователя ПЭВМ, при этом соотношение яркости между рабочими поверхностями не должно превышать 3:1 - 5:1, а между рабочими поверхностями и поверхностями стен и оборудования – 10:1.

В качестве источников света при искусственном освещении следует применять преимущественно люминесцентные лампы типа ЛБ и компактные люминесцентные лампы (КЛЛ). В светильниках местного освещения допускается применение ламп накаливания, в том числе галогенных.

Для освещения помещений с ПЭВМ следует применять светильники с зеркальными параболическими решетками, укомплектованными электронными пускорегулирующими аппаратами (ЭПРА). Допускается использование многоламповых светильников с электромагнитными пускорегулирующими аппаратами (ЭПРА), состоящими из равного числа опережающих и отстающих ветвей.

Применение светильников без рассеивателей и экранирующих решеток не допускается.

При отсутствии светильников с ЭПРА лампы многоламповых светильников или рядом расположенные светильники общего освещения следует включать на разные фазы трехфазной сети.

Общее освещение при использовании люминесцентных светильников следует выполнять в виде сплошных или прерывистых линий светильников, расположенных сбоку от рабочих мест, параллельно линии зрения пользователя при рядом расположении видеодисплейных терминалов. При периметральном расположении компьютеров линии светильников должны располагаться локализовано над рабочим столом ближе к его переднему краю, обращенному к оператору.

Коэффициент запаса ( $K_z$ ) для осветительных установок общего освещения должен приниматься равным 1,4.

Коэффициент пульсации не должен превышать 5%.

Для обеспечения нормируемых значений освещенности в помещениях для использования ПЭВМ следует проводить чистку стекол оконных рам и светильников не реже двух раз в год и проводить своевременную замену перегоревших ламп.

## 7.2.4 Требования к уровням электромагнитных полей на рабочих местах, оборудованных ПЭВМ

Временные допустимые уровни ЭМП, создаваемых ПЭВМ на рабочих местах пользователей, представлены ниже (см. Таблица 7.5).

**Таблица 7.5.**

### **Временные допустимые уровни ЭМП, создаваемых ПЭВМ на рабочих местах.**

Наименование параметров		ВДУ
Напряженность электрического поля	в диапазоне частот 5 Гц - 2 кГц	25 В/м
	в диапазоне частот 2 кГц - 400 кГц	2,5 В/м
Плотность магнитного потока	в диапазоне частот 5 Гц - 2 кГц	250 нТл
	в диапазоне частот 2 кГц - 400 кГц	25 нТл
Напряженность электростатического поля		15 кВ/м

Основным источником электромагнитного излучения на рабочем месте оператора компьютера является монитор. Величина излучения монитора во-многом зависит от его модели, среди наиболее безопасных необходимо выделить мониторы с маркировкой “Low Radiation”, компьютеры с жидко-кристаллическим экраном и самые безопасные – мониторы с установленной защитой по методу замкнутого металлического экрана. Рекомендуется использовать подобные видеотерминалы.

В случае использования видеотерминалов с ЭЛТ для уменьшения воздействия электромагнитных полей на человека устанавливаются поглощающие или отражающие экраны перед видеомонитором. Современный качественный защитный экран позволяет уменьшить уровень электромагнитного излучения и электростатического поля в 10-15 раз и более. Кроме того, существуют специальные очки для защиты глаз от электромагнитного излучения.

## 7.2.5 Требования к визуальным параметрам ВДТ, контролируемым на рабочих местах

Предельно допустимые значения визуальных параметров ВДТ, контролируемые на рабочих местах, представлены в .Таблица 7.6

Таблица 7.6.

N	Параметры	Допустимые значения
1	Яркость белого поля	Не менее 35 кд/кв.м
2	Неравномерность яркости рабочего поля	Не более $\pm 20\%$
3	Контрастность (для монохромного режима)	Не менее 3:1
4	Временная нестабильность изображения (мелькания)	Не должна фиксироваться
5	Пространственная нестабильность изображения (дрожание)	Не более $2 \times 10(-4L)$ , где L - проектное расстояние наблюдения, мм

Для дисплеев на ЭЛТ частота обновления изображения должна быть не менее 75 Гц при всех режимах разрешения экрана, гарантируемых нормативной документацией на конкретный тип дисплея, и не менее 60 Гц для дисплеев на плоских дискретных экранах (жидкокристаллических, плазменных и т.п.).

Обеспечить выполнение этих требований можно соответствующим образом выбирая компьютерную технику, а также используя различные защитные экраны.

### 7.2.6 Общие требования к организации рабочих мест пользователей ПЭВМ

При размещении рабочих мест с ПЭВМ расстояние между рабочими столами с видеомониторами (в направлении тыла поверхности одного видеомонитора и экрана другого видеомонитора) должно быть не менее 2,0 м, а расстояние между боковыми поверхностями видеомониторов - не менее 1,2 м.

Рабочие места с ПЭВМ при выполнении творческой работы, требующей значительного умственного напряжения или высокой концентрации внимания, рекомендуется изолировать друг от друга перегородками высотой 1,5 - 2,0 м.

Экран видеомонитора должен находиться от глаз пользователя на расстоянии 600 - 700 мм, но не ближе 500 мм с учетом размеров алфавитно-цифровых знаков и символов.

Конструкция рабочего стола должна обеспечивать оптимальное размещение на рабочей поверхности используемого оборудования с учетом его количества и конструктивных особенностей, характера выполняемой работы. При этом допускается использование рабочих столов различных конструкций, отвечающих современным требованиям эргономики. Поверхность рабочего стола должна иметь коэффициент отражения 0,5 - 0,7.

Конструкция рабочего стула (кресла) должна обеспечивать поддержание рациональной рабочей позы при работе на ПЭВМ, позволять изменять позу с целью снижения статического напряжения мышц шейно-плечевой области и спины для предупреждения развития утомления. Тип рабочего стула (кресла) следует выбирать с учетом роста пользователя, характера и продолжительности работы с ПЭВМ.

Рабочий стул (кресло) должен быть подъемно-поворотным, регулируемым по высоте и углам наклона сиденья и спинки, а также расстоянию спинки от переднего края сиденья, при этом регулировка каждого параметра должна быть независимой, легко осуществляемой и иметь надежную фиксацию.

Поверхность сиденья, спинки и других элементов стула (кресла) должна быть полумягкой, с нескользящим, слабо электризующимся и воздухопроницаемым покрытием, обеспечивающим легкую очистку от загрязнений.

### **7.2.7 Требования к организации и оборудованию рабочих мест с ПЭВМ для взрослых пользователей**

Высота рабочей поверхности стола для взрослых пользователей должна регулироваться в пределах 680 - 800 мм; при отсутствии такой возможности высота рабочей поверхности стола должна составлять 725 мм.



Модульными размерами рабочей поверхности стола для ПЭВМ, на основании которых должны рассчитываться конструктивные размеры, следует считать: ширину 800, 1000, 1200 и 1400 мм, глубину 800 и 1000 мм при нерегулируемой его высоте, равной 725 мм.

Рабочий стол должен иметь пространство для ног высотой не менее 600 мм, шириной - не менее 500 мм, глубиной на уровне колен - не менее 450 мм и на уровне вытянутых ног - не менее 650 мм.

Конструкция рабочего стула должна обеспечивать:

- ширину и глубину поверхности сиденья не менее 400 мм;
- поверхность сиденья с закругленным передним краем;
- регулировку высоты поверхности сиденья в пределах 400 - 550 мм и углам наклона вперед до 15 град, и назад до 5 град.;
- высоту опорной поверхности спинки 300  $\pm$  20 мм, ширину - не менее 380 мм и радиус кривизны горизонтальной плоскости - 400 мм;
- угол наклона спинки в вертикальной плоскости в пределах  $\pm$  30 градусов;
- регулировку расстояния спинки от переднего края сиденья в пределах 260 - 400 мм;
- стационарные или съемные подлокотники длиной не менее 250 мм и шириной - 50 - 70 мм;
- регулировку подлокотников по высоте над сиденьем в пределах 230  $\pm$  30 мм и внутреннего расстояния между подлокотниками в пределах 350 - 500 мм.

Рабочее место пользователя ПЭВМ следует оборудовать подставкой для ног, имеющей ширину не менее 300 мм, глубину не менее 400 мм, регулировку по высоте в пределах до 150 мм и по углу наклона опорной

поверхности подставки до 20°. Поверхность подставки должна быть рифленой и иметь по переднему краю бортик высотой 10 мм.

Клавиатуру следует располагать на поверхности стола на расстоянии 100 - 300 мм от края, обращенного к пользователю или на специальной, регулируемой по высоте рабочей поверхности, отделенной от основной столешницы.

### 7.2.8 Электробезопасность рабочего помещения

Работа с разработанным ПО должна проводиться в офисных помещениях, относящихся, согласно ПУЭ (правилам устройства электроустановок), к классу помещений «без повышенной опасности», т.к. в помещении нет условий повышенной и особой опасности

Питание устройств ПЭВМ осуществляется от однофазной сети переменного тока с частотой 50 Гц, и напряжением 220 В.

В целях обеспечения необходимой электробезопасности при проведении работ в помещениях с ПЭВМ, необходимо выполнять следующие требования:

- для обеспечения работы операторов ПЭВМ необходимо исключить возможность случайного соприкосновения людей с токонесущими частями оборудования. Это достигается путем изоляции токоведущих частей ЭВМ и приборов и размещения их в недоступных зонах;
- не оставлять ЭВМ и другое оборудование под напряжением без наблюдения;
- не подключать разъёмы кабелей ЭВМ при включении напряжения в сети;
- сеть электропитания технических средств должна обеспечивать защитное заземление технических средств, предусмотренное их технической документацией.

### 7.2.9 Требования пожарной безопасности

Помещение, где функционирует разработанная подсистема, относится к категории В — пожароопасные помещения, в котором имеются твердые сгораемые вещества, способные только гореть, но не взрываться при контакте с кислородом воздуха. Наиболее вероятной причиной пожара является неисправность электрооборудования и электросетей. При эксплуатации ЭВМ возможны возникновения следующих аварийных ситуаций: короткие замыкания, перегрузки, повышение переходных сопротивлений в электрических контактах, перенапряжение, возникновение токов утечки. При возникновении аварийных ситуаций происходит резкое выделение тепловой энергии, которая может явиться причиной возникновения пожара. Требования к пожаробезопасности зданий и сооружений определяются согласно СНиП 21.01-97.

Для снижения вероятности возникновения пожара необходимо проводить различные профилактические мероприятия:

- Организационные — правильная эксплуатация электрооборудования, правильное содержание зданий и помещений;
- Технические — соблюдение противопожарных правил и норм, норм при проектировании зданий, при устройстве отопления, вентиляции освещения, правильное размещение оборудования;
- Мероприятия режимного характера — запрещение курения в неустановленных местах и т.д.;
- Эксплуатационные — своевременные профилактические осмотры и ремонт неисправного электрооборудования.

Для снижения вероятности возникновения и распространения пожара на ранней стадии необходимо:

- установить пожарную сигнализацию с системой оповещения работников, дежурного по объекту и, желательно, автоматическое оповещение противопожарных служб;
- иметь в наличии несколько ручных углекислотных огнетушителей (например, огнетушитель марки ОУ-3);
- помещение должно быть планом эвакуации при пожаре и аптечкой первой помощи. В помещении должен быть ответственный за пожарную безопасность.

#### 7.2.10 Требования к организации медицинского обслуживания пользователей ПЭВМ

Лица, работающие с ПЭВМ более 50% рабочего времени (профессионально связанные с эксплуатацией ПЭВМ), должны проходить обязательные предварительные при поступлении на работу и периодические медицинские осмотры в установленном порядке.

Женщины со времени установления беременности переводятся на работы, не связанные с использованием ПЭВМ, или для них ограничивается время работы с ПЭВМ (не более 3-х часов за рабочую смену) при условии соблюдения гигиенических требований, установленных *СанПиН 2.2.2/2.4.1340-03 Гигиенические требования к персональным электронно-вычислительным машинам и организации работы*. Трудоустройство беременных женщин следует осуществлять в соответствии с законодательством Российской Федерации.

#### 7.2.11 Требования к проведению государственного санитарно-эпидемиологического надзора и производственного контроля

Государственный санитарно-эпидемиологический надзор за производством и эксплуатацией ПЭВМ осуществляется в соответствии с

### **СанПиН 2.2.2/2.4.1340-03 Гигиенические требования к персональным электронно-вычислительным машинам и организации работы.**

Не допускается реализация и эксплуатация на территории Российской Федерации типов ПЭВМ, не имеющих санитарно-эпидемиологического заключения.

Инструментальный контроль за соблюдением требований СанПиН 2.2.2/2.4.1340-03 осуществляется в соответствии с действующей нормативной документацией.

Производственный контроль за соблюдением санитарных правил осуществляется производителем и поставщиком ПЭВМ, а также предприятиями и организациями, эксплуатирующими ПЭВМ в установленном порядке, в соответствии с действующими санитарными правилами и другими нормативными документами.

#### **7.2.12 Расчет системы защитного заземления компьютера**

Данное помещение является помещением с повышенной электроопасностью, т.к. имеется возможность касания токоприемников (ВДТ и ПЭВМ) с проводящими системами, связанными с землей (система централизованного отопления), поэтому заземление необходимо.

Определяем расчетное удельное сопротивление грунта  $\rho_{расч}$  с учетом климатического коэффициента  $\Psi$  (табл. 5 и 6):  $\rho_{расч} = \rho_{изм} \cdot \Psi$ , где  $\rho_{изм}$  - удельное сопротивление грунта, измеренное или полученное из справочной литературы.

**Таблица 7.7. Приближенные значения удельных сопротивлений грунтов.**

Грунт	Удельное сопротивление, $10^2$ Ом·м	
	Возможные пределы колебаний	При влажности 10-12% к массе грунта
Песок	4-7	7

Супесок	1,5-4	3
Суглинок	0,4-1,5	1
Глина	0,08-0,7	0,4
Чернозем	0,09-5,3	2
Речная вода	0,5	-
Морская вода	0,002-0,01	-

Выбираем суглинок, как наиболее близкий тип почвы. При нормальной влажности почвы (10-12%), получаем  $\rho_{изм} = 100 \text{ Ом}\cdot\text{м}$ .

**Таблица 7.8. Значения расчетных климатических коэффициентов сопротивления грунта.**

Грунт	Глубина заложения, м.	$\Psi_1$	$\Psi_2$	$\Psi_3$
Суглинок	0,8—3,8	2,0	1,5	1,4

Примечание. Расчетное сопротивление грунта определяется по значениям:

$\Psi_1$  – при большой влажности грунта;  $\Psi_2$  – при средней влажности грунта;  $\Psi_3$  – при сухом грунте.

Выбираем средне влажный грунт:  $\Psi = 1,5$ . Тогда  $\rho_{расч} = 100 \cdot 1,5 = 150 \text{ Ом}\cdot\text{м}$ .

Теперь выбираем тип заземлителя. Выбираем протяженную полосу в грунте.

Тип заземлителя	Формула	Дополнительные указания
Протяженная – полоса в грунте	$R_3 = \frac{\rho}{2\pi l} \cdot \ln \frac{2l^2}{bH}$	$\frac{l}{H} \geq 5$

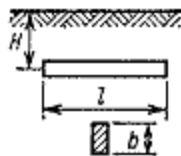


Рис.1. Схема заземлителя.

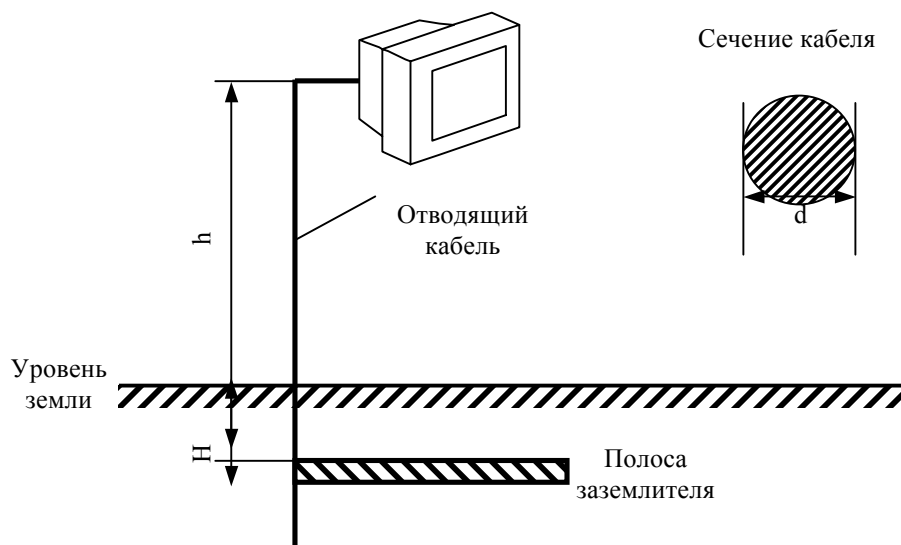


Рис.7.2. Схема заземления ВДТ

$H = 3$  м – глубина залегания полосы заземления ;

$h = 6$  м – высота от уровня земли до ВДТ;

$d = 10$  мм – диаметр отводящего кабеля.

Материал кабеля: медь.

$\rho_{Cu} = 0,0175$  Ом·мм<sup>2</sup>/м – удельное сопротивление меди.

Сопротивление отводящего кабеля:  $R_{\hat{E}\hat{A}\hat{A}} = \rho_{Cu} \frac{h + H}{\pi \cdot d^2 / 4}$

$$R_{\hat{E}\hat{A}\hat{A}} = 0,002\hat{i}$$

Общее сопротивление системы заземления:  $R_{ОБЩ} = R_3 + R_{КАБ}$

$$\rho = 150 \text{ Ом}\cdot\text{м};$$

$$b = 0,1 \text{ м}.$$

В целях проведения проверочного расчета принимаем предельный случай  $l/H = 5 \Rightarrow l = 15i$  .

(Уравнение рассчитано с использованием пакета MathCAD 2002)

$$R_{\zeta} = \frac{\rho}{2\pi l} \cdot \ln \frac{2l^2}{bH} \Rightarrow R_{\zeta} = 6.51\hat{l}$$

$$R_{i\dot{u}} = R_{\zeta} + R_{\dot{e}\dot{a}\dot{a}} \Rightarrow R_{i\dot{u}} = 6.512\hat{l} > 4\hat{l}$$

Чтобы напряжение на заземленном корпусе было минимальным, ограничивают сопротивление заземления. Для данного случая с напряжением 380/220В оно должно быть не более 4Ом. Т.е. имеющаяся система защитного заземления компьютера не удовлетворяет требованиям электробезопасности. Необходимо пересмотреть параметры системы

защитного заземления, например, варьировать соотношение  $\frac{l}{H} \geq 5$

### 7.3 Утилизация ПЭВМ

Извлечение драгоценных металлов из вторичного сырья является частью проблемы использования возвратных ресурсов, которая включает в себя следующие аспекты: нормативно-правовой, организационный, сертификационный, технологический, экологический, экономико-финансовый. Проблема использования вторичного сырья, содержащего драгоценные материалы из компьютеров, периферийного оборудования и иных средств вычислительной техники (СВТ) актуальна в связи с техническим перевооружением отраслей промышленности.

К драгоценным металлам относятся: золото, серебро, платина, палладий, родий, иридий, рутений, осмий, а также любые химические соединения и сплавы каждого из этих металлов. Статья 2 п. 4 "Федерального закона о драгоценных металлах и драгоценных камнях" от 26 марта 1998 года №1463 гласит: "Лом и отходы драгоценных металлов подлежат сбору во всех организациях, в которых образуются указанные лом и отходы. Собранные лом и отходы подлежат обязательному учёту и могут перерабатываться собирающими их организациями для вторичного использования или реализовываться организациям, имеющим лицензии на



данный вид деятельности, для дальнейшего производства и аффинажа драгоценных металлов".

Порядок учёта, хранения, транспортировки, инвентаризации, сбор и сдача отходов драгоценных металлов из СВТ, деталей и узлов, содержащих в своём составе драгоценные металлы для предприятия, учреждения и организации (далее - предприятие), независимо от форм собственности, установлен инструкцией Министерства финансов Российской Федерации от 4 августа 1992 года №67. Все виды работ с драгоценными металлами строго регламентированы нормативно-правовыми документами, перечень которых представлен в Приложении 1.

### **7.3.1 Разборка изделий**

Последовательность разборки определяется типом изделия СВТ, его конструктивными особенностями и комплектацией.

Как правило, процесс разборки должен выполняться в последовательности, обратной процессу сборки изделия. Основные направления деятельности на этапе «Разборка»

#### ***7.3.1.1 Разборка персональных компьютеров (ПЭВМ), рабочих станций и серверов***

Технологии разборки ПЭВМ, рабочих станций, серверов и информационно-вычислительных систем едины поскольку состав их модулей стандартный. Он содержит системный блок и комплект периферийных устройств.

Разборку ПЭВМ и составных модулей целесообразно осуществлять по технологической схеме представленной на рисунке (см. Рисунок 7.1).

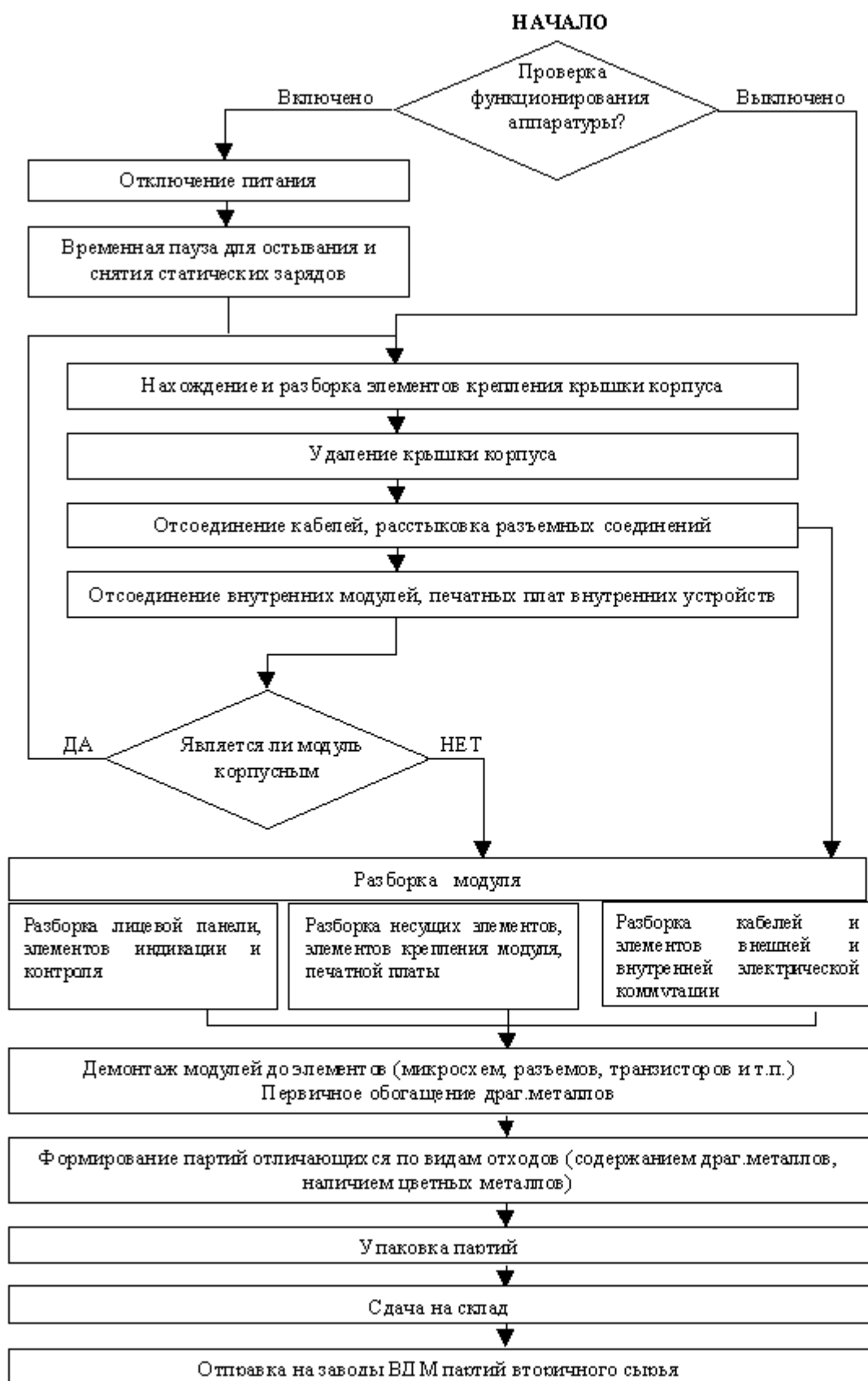


Рис. 7. Технологическая схема разборки ПЭВМ

Порядок разборки системного блока:

Выключить компьютер и отсоединить шнур питания от розетки и системного блока. Отсоединить переходной шнур питания от системного блока к монитору.

Отсоединить от компьютера клавиатуру, монитор, манипулятор "мышь", принтер, сканер и иные внешние устройства.

Найти элементы крепления крышки корпуса (винты, шурупы, пружинные защелки и т.д.). Освободить крышку от элемента крепления.

Снять крышку.

Отсоединить внутренние кабели и плоские шлейфы.

Найти элементы крепления дисководов (НМД, НГМД) в отсеке для дисководов (винты, шурупы, саморезные винты, пружинные защелки и др.). Освободить дисководы и извлечь их из дискового отсека.

Освободить от крепёжных элементов периферийные платы. Извлечь из разъёмов непосредственного контактирования все периферийные платы.

Найти элементы крепления системной платы к корпусу (винты, шурупы). Освободить элементы крепления и извлечь системную плату из корпуса.

Извлечь модули памяти из разъёмов системной платы.

Найти элементы крепления блока питания к корпусу (винты, шурупы, саморезные винты, пружинные защелки и пр.). Освободить элементы крепления и извлечь блок питания.

Разобрать блок питания и извлечь высоковольтные конденсаторы содержащие тантал.

Разобрать ПП и модули памяти до компонентов (микросхем, транзисторов, разъёмов и т.п.).

Произвести сортировку компонентов и сформировать партии электронного лома.

Упаковать партии, составить опись, произвести расчёт (анализ) драгметаллов и передать их на склад.

Провести сортировку цветных и чёрных металлов, пластмасс, сформировать партии и передать их на склад или на переработку.

При оценке содержания драгоценных металлов в партии электронного лома отечественных ПЭВМ необходимо руководствоваться паспортными данными. При оценке ПЭВМ импортного производства необходимо провести ориентировочные расчёты по отечественным аналогам.

### **7.3.1.2 Обеспечение комплексности технологии разборки**

При разборке изделий СВТ образуются материалы и изделия, которые имеют материальную ценность и подлежат реализации.

Примерный перечень материалов представлен ниже см. Таблица 7.9.

**Таблица 7.9. Перечень материалов, подлежащих утилизации.**

<b>Вид материалов или изделий</b>	<b>Характеристика</b>
Печатные платы, разъемы и соединители, микросхемы	вторичные драгоценные металлы
Электрические провода и кабели, соединители	вторичная медь и её сплавы
Свинец и олово из печатных плат	вторичные припойные пасты (олово и свинец)
Танталовые конденсаторы К-53-1	вторичный танталл
Некоторые корпуса компьютеров, дисковод и т.д.	алюминиевые сплавы
Корпуса стоек, ячеек, шкафов, компьютеров	сталь
Крепежные изделия	болты, гайки, винты
Вентиляторы и электромоторы	по паспорту СВТ
Пластиковая "фракция"	стеклотекстолит, пластмасса разъемов и соединителей

Вид материалов или изделий	Характеристика
Экраны компьютеров	стеклофаза, содержащая Pb, Cd, CdS, редкоземельные металлы

Таким образом, можно сделать вывод о целесообразности извлечения вторичных чёрных и цветных металлов, пластмасс, стекла, крепёжных изделий, вентиляторов и электромоторов.

### ***7.3.1.3 Извлечение вторичных чёрных металлов***

Отечественная практика показывает, что на 1 г извлекаемого золота приходится около 1 кг лома чёрных металлов. В связи с высокой стоимостью транспортно-погрузочных работ рекомендуется производить отгрузку предприятиям-покупателям партий лома чёрных металлов весом не менее 10 тонн. Блоки, панели, съёмные кожухи, рамы, каркасы шкафов и стоек стационарных ЭВМ, изготовленные из стального нормализованного профиля или листа, подвергаются сортировке, набираются в партии и реализуются.

Предпочтительно заключение договоров при условии, когда предприятие-покупатель своим транспортом вывозит вторичные металлы с территории предприятия-продавца.

Крепёжные изделия, заготовки стального профиля, листов, вентиляторы, электропускатели, кнопки, электрический кабель направляются на реализацию непосредственно в торговую сеть.

Опыт показывает, что денежные средства от реализации этих изделий не превышают 0,6 % от общей суммы.

### ***7.3.1.4 7.3.1.2.3. Извлечение вторичных цветных металлов***

В процессе разборки изделий СВТ образуется лом (содержащий медь) классификация которого должна проводиться по ГОСТ 1639.

В соответствии с ГОСТ 1639 медные шины целесообразно относить к классу А, группам I и II; латунь - к группам IV-VIII; бронзу - к группам

XI-XII; отходы кабеля и проводов ПП следует относить к классу Г, группа XIII.

Все виды ломов необходимо сортировать по классам и группам, формировать в партии и реализовывать.

В процессе разборки изделий СВТ алюминий и его сплавы обычно содержатся в типовых конструкциях изделий. По ГОСТ 1639 их следует относить к классам А3 и Б5.

Все виды отходов необходимо сортировать, формировать в партии и реализовывать.

Свинцово-оловянные припой содержатся в печатных платах и их количество превышает количество золота в десятки раз.

Припой регенерируются при переработке печатных плат.

При разборке СВТ танталовые конденсаторы необходимо складировать отдельно для последующей реализации.

Переработка изделий из пластмасс

Пластмассы следует сортировать по видам.

Переработке подлежат термопласты: поливинилхлорид, полиэтилен, полистирол и т.п.

Стёкла люминесцентных экранов электронно-лучевых трубок следует использовать в производстве керамики и в качестве сырья при производстве новых люминесцентных трубок.

### 7.3.2 Реализация партий

На рисунке ниже (см. Рисунок 8) представлены основные направления деятельности на этапе "Реализация партий".

Основные действия на этапе "Реализация партий" представляют собой последовательность действий, создающих основу для успешного выполнения процедур завершающего этапа утилизации СВТ.

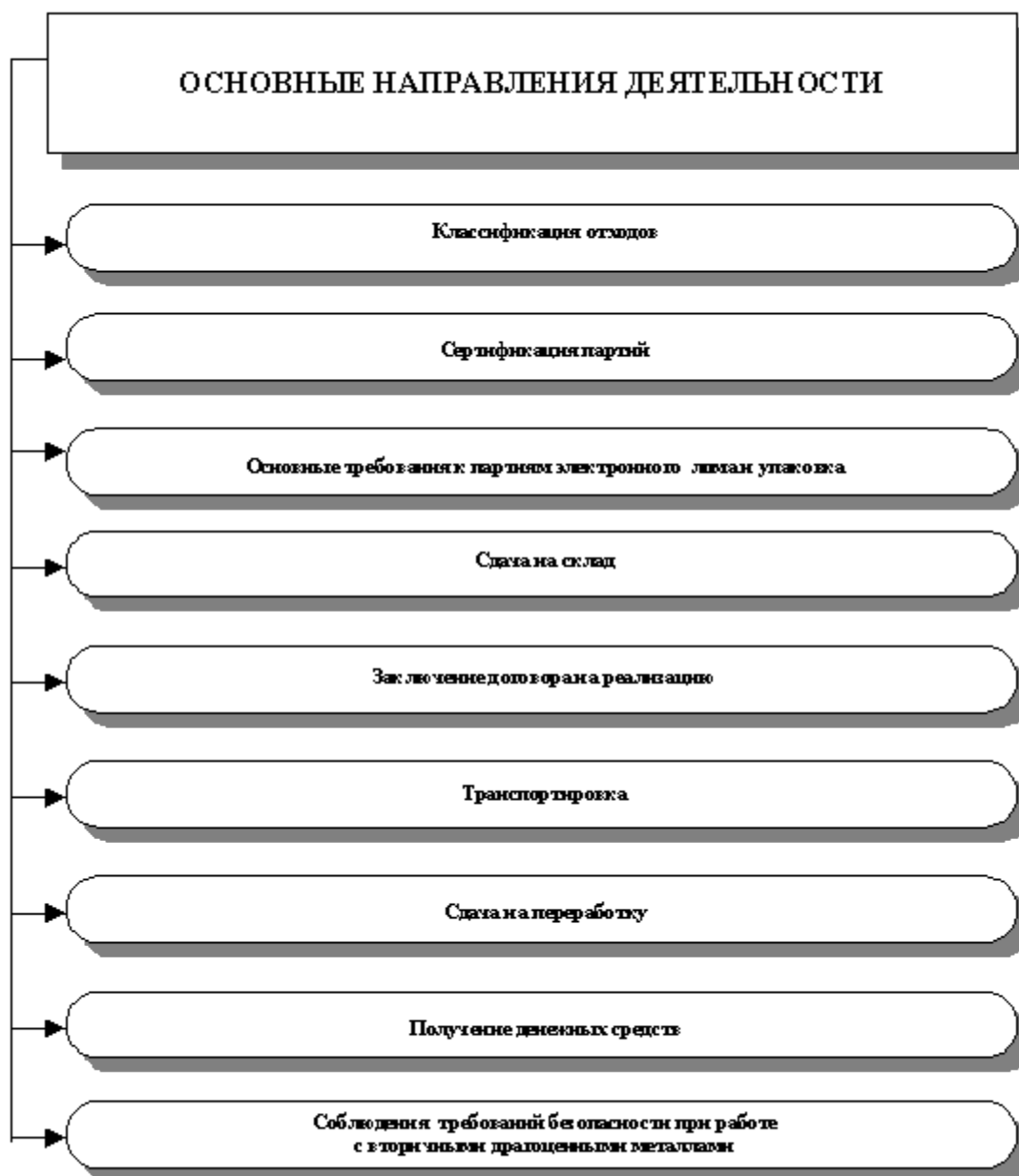


Рис. 8. Основные направления деятельности на этапе «Реализации партий»

### 7.3.2.1 Классификация отходов

В настоящее время в России и за рубежом не существует единой классификации вторичного сырья, содержащего драгоценные металлы. Поэтому, возможно разделение вторичного сырья по следующим признакам.

По содержанию драгоценных металлов:

- бедное (менее 1 % золота, 5 % серебра и 1 % металлов платиновой группы);
- богатое (более 1 % золота, 5 % серебра и 1 % металлов платиновой группы).

По составу материала основания:

- на металлической основе;
- на органической (пластиковой) основе;
- на керамической основе;
- на комбинированной основе.

По физическим признакам:

- твёрдые компактные отходы;
- сыпучие (порошки);
- жидкие.

Возможна классификация вторичного сырья в зависимости от сферы производства в:

- ювелирной промышленности;
- химической промышленности;
- электронной, электрохимической, оборонной, радиопромышленности (радиолампы, разъёмы, контакты, контактные устройства, платы на органической основе, микросхемы, радиодетали, кабели и провода, лента, высечка, вырубка, аккумуляторы, элементы питания, прочие отходы);
- бытовых отходах (лом бытовой радиоэлектронной аппаратуры, бытовой стеклянный и фарфоровый бой, лом ювелирных украшений и т.д.).

Отходы классифицируются по элементному составу.

При этом электронный лом отличается особым многообразием состава. Например, современный компьютерный лом содержит несколько



десятков видов деталей, содержащих благородные, цветные и чёрные металлы.

### *7.3.2.1.6. Классификация сырья вторичных драгоценных металлов*

Согласно методике опробования электронного лома и отходов, содержащих драгоценные металлы, разработанной ИАСЦ ГНЦ "ГИРЕДМЕТ", сырьё вторичных драгоценных металлов следует рассортировывать на классы, приведённые в табл.7.

Таблица 7

#### **Классификация электронного лома и отходов, содержащих драгоценные металлы (по видам)**

<b>Номер класса</b>	<b>Вид сырья</b>
1.	Микросхемы, транзисторы, диоды россыпью
2.	Конденсаторы россыпью
3.	Ножки разъёмов позолоченные и посеребренные
4.	Контакты разделанные
5.	Платы, содержащие элементы классов 1 и 2
6.	Разъёмы с позолоченными и посеребренными ножками
7.	Реле, переключатели, тумблеры
8.	Радиолампы
9.	Платы, содержащие элементы классов 1, 2, 6, 7 и 8
10.	Крупногабаритные детали (волноводы и т.п.)с покрытием из
11.	драгметаллами
12.	Элементы питания, аккумуляторы, ампульные батареи
	Сыпучие материалы (шихта катализаторов, зола фотоматериалов, шлам фиксажный и т.п.)

### *7.3.2.2. Сертификация партий*

#### *7.3.2.2.1.*

В целях обеспечения строгого учёта, сохранности, сокращения потерь и эффективности использования драгоценных металлов, содержащихся в электронном ломе и отходах, а также для обеспечения единства и требуемой точности измерений при опробовании и проведении анализов химического состава, необходимо руководствоваться нормативными документами утверждёнными Комитетом Российской Федерации по драгоценным металлам и драгоценным камням, Комитетом Российской Федерации по стандартизации, метрологии и сертификации: "Порядком выдачи сертификатов химического состава на партии электронного лома и отходов, содержащих драгоценные металлы", "Временной методикой опробования электронного лома и отходов, содержащих драгоценные металлы".

Указанные документы определяют порядок проведения работ и требования по опробованию и сертификации химического состава партий электронного лома и отходов.

#### *7.3.2.2.2. Сертификация химического состава электронного лома и отходов, содержащих драгоценные металлы, включает следующие работы:*

- оформление и представление заявки в соответствующий орган по сертификации;
- создание комиссии по опробованию;
- опробование, оформление документов по результатам опробования, передача пробы на анализ;
- собственно анализ пробы и оформление количественного химического анализа;

- оформление сертификата.

Выполнение измерений химического состава проб (анализ) электронного лома и отходов, содержащих драгоценные металлы, следует производить методами количественного химического анализа по аттестованным методикам, Приложение 2.

Анализ проб осуществляется аналитическими лабораториями, которые аккредитованы Комитетом Российской Федерации по стандартизации, метрологии и сертификации, а также рекомендованными организациями и органами по сертификации, Приложение 3.

По результатам анализа аккредитованная лаборатория оформляет протокол измерений химического состава пробы электронного лома или отходов по установленной форме.

По результатам процедур опробования и анализа химического состава электронного лома или отходов, орган по сертификации оформляет и выдает заявителю Сертификат химического состава на содержание драгоценных металлов установленной формы.

Сертификат химического состава и комплекс документов по опробованию сертифицируемой партии электронного лома включается в состав сопроводительной документации при передаче партии вторичного сырья от сдатчика заготовителю или переработчику, а также при вывозе за границу для переработки.

При возникновении разногласий, в процессе передачи сертифицированных партий электронного лома и отходов от сдатчика заготовителю или переработчику, производится арбитраж. В этом случае партия не может быть передана переработчику до получения заключения арбитражной лаборатории, аккредитованной Комитетом Российской Федерации по стандартизации, метрологии и сертификации.

*7.3.2.3. Основные требования к партиям электронного лома и упаковка*

Утверждённые технические требования к партиям электронного лома, в частности его упаковке, до настоящего времени отсутствуют.

В связи с этим рекомендуется придерживаться следующих правил.

Не допускается смешивание различных классов лома.

В ломе и отходах драгоценных металлов не допускаются посторонние предметы, не относящиеся к естественной засорённости.

Лом и отходы драгоценных металлов должны храниться в специально предназначенной для этого таре: в пакетах из полиэтиленовой плёнки по ГОСТ 10354, в фанерных ящиках по ГОСТ 9396, в металлических ящиках с замками собственного изготовления или мешках из мешочной бумаги по ГОСТ 2226.

Ящики внутри должны быть выложены упаковочной бумагой по ГОСТ 515 или каким-либо плёночным материалом. Пакеты и мешки, предназначенные для хранения и отправки лома и отходов, должны изготавливаться с вывернутыми внутрь двумя боковыми швами без нижнего шва.

Мешки или пакеты, уложенные в деревянные или металлические ящики, допускается клеивать какой-либо клеевой лентой по ГОСТ 18351.

Опечатывание отходов в таре производится после прошивания мешков и пакетов шпагатом по ГОСТ 17308 или заваривания открытых краев полиэтиленовых пакетов с последующим складыванием краёв "гармошкой" и прошиванием её двумя концами шпагата.

Партия лома и отходов должна состоять из одного или нескольких мест в каждом из которых вложена одна или несколько позиций различающихся по составу компонентов, конфигураций, габаритным размерам и другими признаками, не меняющих принципиально сущности последующего опробования на перерабатывающих заводах.

Взвешивание и упаковка отходов производится с участием материально ответственных лиц подразделений предприятия сдатчика.

После контрольного взвешивания каждое место должно быть опломбировано или опечатано сургучной печатью сдатчика.

В сопроводительных документах указывается описание пломбы или печати.

#### *7.3.2.4. Сдача на склад*

Передача партии электронного лома из производственного подразделения на склад драгоценных металлов предприятия осуществляется на основе приёмно-передаточного акта, акта изъятия узлов и изделий техники и других нормативных документов.

#### *7.3.2.5. Заключение договора на реализацию*

Между продавцом и покупателем заключается типовой договор, предметом которого является полученная партия электронного лома.

В договоре указываются обязанности сторон, порядок подготовки, отправки лома и отходов драгоценных металлов, порядок приёмки, опробования лома и отходов, порядок расчётов и ответственность сторон.

#### *7.3.2.6. Транспортировка*

Транспортирование лома и отходов драгоценных металлов с содержанием золота и металлов платиновой группы более 5 % должно производиться через специальную связь в соответствии с инструкцией Министерства связи Российской Федерации о перевозке ценностей.

Лом и отходы с содержанием золота и металлов платиновой группы менее 5 %, а также отходы серебра отправляются на перерабатывающие заводы почтовыми посылками, багажом по железной дороге или другим видом транспорта с оценочной стоимостью отгружаемого груза.

Сдача на переработку.

Порядок сдачи партии на переработку определяется условиями договора. Типичные условия договора для завода ВДМ следующие.

Вскрытие посылок (мест) производится на заводе приёмной комиссией, которая взвешивает и сверяет фактическое наличие сырья и его качественный состав по каждому виду сырья с данными продавца. По результатам приёмки сырья покупатель высылает продавцу приёмный акт в течение 15 дней от даты поступления сырья.

При доставке сырья транспортом продавца, приёмный акт на количество мест выдаётся на руки уполномоченному представителю продавца в день сдачи сырья.

Представителю продавца необходимо иметь копию описи сдаваемого сырья.

В случае нарушения упаковки или целостности печати материально ответственный работник покупателя в акте на приём отходов отмечает все нарушения.

При расхождении фактически установленных данных при приёмке сырья с данными, значившимися в сопроводительных документах продавца, а также при отсутствии сопроводительных документов, окончательными результатами приёмки является масса брутто, нетто сырья, установленные приёмной комиссией покупателя.

Взвешивание, опробование, пробоотбор и химический анализ проб каждой партии производится в соответствии с нормативно-технической документацией и по технологии, применяемой покупателем.

Однотипные позиции партии подлежат объединению и опробыванию по единой технологической схеме.

По результатам опробования сырья на содержание драгоценных металлов, которое производится в течение 60 дней со дня его поступления, покупатель высылает продавцу паспорт с указанием количества

драгоценного металла с учетом процента выхода чистого металла в готовую продукцию.

Паспорт подписывается руководителем предприятия-покупателя, главным бухгалтером или их заместителями и скрепляется печатью покупателя.

Порядок получения денежных средств зависит от условия договора. Типичные условия завода ВДМ следующие.

Стоимость поставленного продавцом сырья определяется паспортом покупателя, составленным на основании прейскуранта завода, по мировым ценам на продукцию, получаемую из сырья на день, предшествующий выписке паспорта и пересчитанным в рубли по курсу, установленному Центральным Банком Российской Федерации на день выписки паспорта.

Денежные средства перечисляются на расчётный счет продавца в течение трёх банковских дней со дня получения подтверждающего документа о поступлении денежных средств на расчётный счёт покупателя.

Продавец производит оплату с каждой поставленной партии за опробование.

Все платежи по договору должны производиться в валюте Российской Федерации в безналичной форме.

#### *7.3.2.7. Соблюдение требований безопасности при работе с вторичными драгоценными металлами*

Выполнение работ по разборке списанных СВТ предполагает соблюдение общих правил, изложенных в инструкциях по охране труда для слесаря механо-сборочных работ и лиц, работающих с ручным электроинструментом.

Специальные требования техники безопасности при работе с вторичными драгоценными металлами следующие.

Не допускается сбор, заготовка и переработка радиоактивного лома и отходов драгоценных металлов.

Степень воздействия на организм человека вредных веществ, выделяющихся в процессе заготовки и переработки лома и отходов драгоценных металлов, класс опасности и их предельно-допустимая концентрация (ПДК) в воздухе рабочей зоны установлены по ГОСТ 12.1.005 и ГОСТ 12.1.007, табл.8.

Таблица 8

**Степень воздействия драгоценных металлов на организм человека**

Наименование металла	Характер действия на организм человека	Пути проникновения	Класс опасности	ПДК вредных веществ в воздухе рабочей зоны, мг/м <sup>3</sup>
Серебро и его соединения	Отходы могут оказывать раздражающее действие на слизистую оболочку носа и дыхательных путей.	Органы дыхания	11	0,5-1
Золото, платина и его соединения	При длительном контакте могут вызывать аллергические дерматиты и экземы.	Открытые участки кожи	-	-



Рутений		Органы дыхания	11	-
Родий	Оказывает раздражающее действие на слизистую оболочку носа и дыхательных путей. У рабочих, занятых очисткой родия, иногда развивается сверхчувствительность	-	-	-

Контроль за содержанием вредных веществ в воздухе рабочей зоны проводится в соответствии с требованиями ГОСТ 12.1.005 и ГОСТ 12.1.007. Анализ проб воздуха проводится в соответствии с нормативно-технической документацией, утверждённой Минздравом Российской Федерации.

Производственные помещения в местах образования вредных веществ и пыли должны быть оборудованы вентиляцией согласно ГОСТ 12.4.021 с обеспечением санитарно-гигиенических требований к воздуху рабочей зоны.

Для снятия статического электричества пылеприёмники, воздуховоды вентиляционных установок должны иметь заземление, выполненное и обозначенное в соответствии с ГОСТ 12.2.007.0, ГОСТ 12.2.007.14 и ГОСТ 21130.

Для предотвращения попадания пыли, твёрдых веществ на слизистую оболочку глаз необходимо пользоваться защитными очками типа ПО-2, ПО-3 согласно ГОСТ 12.4.013.

При работе с пылящими отходами необходимо пользоваться фильтрующими респираторами РУ-60 и РУ-60му по ГОСТ 17269 и респираторами "Лепесток" по ГОСТ 12.4.028.

При этом респираторы должны периодически подвергаться промывке.

Средства индивидуальной защиты работающих с ломом и отходами драгоценных металлов и сплавов должны соответствовать типовым отраслевым нормам бесплатной выдачи рабочим и служащим металлургических производств. Спецодежда должна соответствовать ГОСТ 29057 и ГОСТ 29058.

Помещения в местах выгрузки и загрузки лома и отходов, оказывающих вредное воздействие на организм человека, должны быть оборудованы местными отсосами согласно ГОСТ 12.4.021.

Производственные помещения должны соответствовать требованиям "Санитарных норм проектирования промышленных предприятий СН 245-71".

Метеорологические условия производственных помещений должны соответствовать санитарным нормам проектирования промышленных предприятий по ГОСТ 12.1.005.

Требования безопасности при погрузочно-разгрузочных работах лома и отходов драгоценных металлов и сплавов должны соответствовать ГОСТ 12.3.009.

Требования по обеспечению взрывобезопасности.

Предприятия и организации, заготавливающие и перерабатывающие лом и отходы драгоценных металлов сплавов, должны проверять весь лом и отходы драгоценных металлов на взрывобезопасность.

Из лома необходимо отобрать и удалить взрывоопасные предметы, материалы, в том числе электронно-вакуумные трубки дисплеев.

Замкнутые сосуды, резервуары и другие полые предметы (баллоны, цилиндры, сосуды, электровакуумные изделия и т.д.) разгерметизируются и освобождаются от содержимого (газов или жидкостей).

Разгерметизацию должны производить рабочие, прошедшие специальное обучение, которые перед началом работы инструктируются в установленном порядке о мерах предосторожности.

## ЗАКЛЮЧЕНИЕ

В рамках данного дипломного проекта были получены следующие результаты:

Проведено предпроектное исследование системы имитационного моделирования РДО и сформулированы предпосылки создания в системе инструмента передачи в процесс обслуживания ресурсов-транзактов, созданных в событиях и базе знаний модели. Разработана соответствующая модель-прототип.

На этапе концептуального проектирования системы с помощью диаграммы компонентов нотации UML укрупненно показано внутреннее устройство РДО и выделены те компоненты, которые потребуют внесения изменений в ходе этой работы.

На этапе технического проектирования разработан новый синтаксис образцов активностей и событий, который представлен на синтаксической диаграмме. С помощью диаграммы классов разработана архитектура новой системы. С помощью блок-схемы разработаны алгоритмы, реализующие в системе новую концепцию создания ресурсов.

На этапе рабочего проектирования написан программный код для реализации спроектированных ранее алгоритмов работы и архитектуры компонентов `rdo_parser` и `rdo_runtime` системы РДО. Проведены отладка и тестирование новой системы, в ходе которых исправлялись найденные ошибки. Работоспособность системы была проверена на разработанной ранее модели-прототипе.

Таким образом, поставленная цель дипломного проекта достигнута в полном объеме.

## СПИСОК ЛИТЕРАТУРЫ

1. Материалы по системе имитационного моделирования РДО. [В Интернете] <http://rdo.rk9.bmstu.ru/forum/viewforum.php?f=4>.
2. **Емельянов В.В., Ясиновский С.И.** *Введение в интеллектуальное имитационное моделирование сложных дискретных систем и процессов. Язык РДО.* Москва : Анвик, 1998.
3. *Единая система программной документации. Техническое задание.*
4. **Арсеньев В.В., Сажин Ю.Б.** *Методические указания к выполнению.* Москва : МГТУ, 1994. Т. 52.
5. **Прицкер, А.** *Введение в имитационное моделирование и язык СЛАМ II.* Москва : Мир, 1987.
6. *Нормы пожарной безопасности «Определение категорий».*
7. *Система стандартов безопасности труда. Электробезопасность.*
8. *Сборник типовых расчетов по курсу «Охрана труда» для студентов.* МВТУ : б.н., 1984 г.
9. *Изделия электротехнические.*
10. *Производственные вибрации в помещениях жилых и общественных.*
11. *Шум на рабочих местах, в помещениях жилых, общественных.*
12. *Предельно допустимые концентрации (ПДК) вредных веществ в.*

## ПРИЛОЖЕНИЕ 1. Код имитационной модели-прототипа грузоперевозок.

Файл model.rtp (типы ресурсов):

\$Resource\_type системы: permanent

\$Parameters

число: integer = 0

\$End

\$Resource\_type счетчики: permanent

\$Parameters

количество\_обслуженных\_заявок: integer = 0

счет\_заявок : integer = 0

количество\_удаленных\_заявок : integer = 0

количество\_удаленных\_составов: integer = 0

время\_обслуживания : real = 0

время : real = 0

среднее\_время\_обслуживания : real = 0

\$End

\$Resource\_type заявки: temporary

\$Parameters

номер\_заявки : integer = 0

вес\_груза : real = 0.0

маршрут : integer [1..3]

количество\_вагонов\_в\_заявке: integer = 0

остаток\_в\_заявке : integer = 0

состояние : (пришла, распределение\_вагонов,

списание\_начато, списание\_закончено, распределена, обслужена, delete ) =

пришла

номер\_последнего\_состава : integer = 0

буфер : integer = 0

время\_прихода : real = 0

время\_обслуживания : real = 0

\$End

\$Resource\_type формируемые\_составы: permanent

\$Parameters

номер\_формируемого\_состава : integer = 1

```

оставшееся_число_вагонов : integer = 60
состояние                  : (распределение, списание,
заполнен, следующий) = распределение
маршрут                    : such_as заявки.маршрут
$End

```

```

$Resource_type составы : temporary
$Parameters
    Время_создания      : real
    состояние            : (сформирован, создается, прибыл_в_парк,
расформирование, расформирован) = сформирован
    маршрут              : such_as заявки.маршрут = 1
    номер_состава       : integer = 0
$End

```

```

$Resource_type парки : permanent
$Parameters
    количество_вагонов_в_парке: integer [0..2000] = 240
    дефицит                    : integer = 0
    заказ_аренды               : bool = false
$End

```

### Файл model.rss (ресурсы):

```

$Resources
    система                : системы                *
    счетчик                : счетчики                trace * * * * * * *
    формируемый_состав1   : формируемые_составы trace * * * 1
    формируемый_состав2   : формируемые_составы trace * * * 2
    формируемый_состав3   : формируемые_составы trace * * * 3
    парк                   : парки                trace * * *
$End

```

```

Файл model.evn (события):
$Pattern Образец_прихода_заявок: event
$Relevant_resources
    _счетчик: счетчик Keep

```

```

        _заявка : заявки Create
$Body
    _счетчик
        Convert_event
            if (счетчик.счет_заявок < 100)
Образец_прихода_заявок.planning(time_now + ф_приход_заявок (34, 38));
                счет_заявок += 1;
    _заявка
        Convert_event trace
            номер_заявки = _счетчик.счет_заявок;
            вес_груза = ф_вес_груза (900.0, 1800.0);
            маршрут = ф_тип_маршрута;
            время_прихода = Time_now;
$End

```

### Файл model.pat (образцы активностей):

```

$Pattern Образец_расчета_количества_вагонов: rule
$Relevant_resources
    _заявка: заявки Keep
$Body
    _заявка
        Choice from _заявка.состояние == пришла
        Convert_rule
            if (Int(_заявка.вес_груза/тоннаж) <
_заявка.вес_груза/тоннаж)
                количество_вагонов_в_заявке =
Int(_заявка.вес_груза/тоннаж) + 1;
            else
                количество_вагонов_в_заявке =
Int(_заявка.вес_груза/тоннаж);
                состояние = распределение_вагонов;
                остаток_в_заявке =
_заявка.количество_вагонов_в_заявке;
$End

$Pattern Образец_списания_вагонов: rule
$Relevant_resources
    _заявка : заявки Keep
    _формируемый_состав: формируемые_составы Keep
    first

```



```

$Body
_заявка
    Choice from _заявка.состояние == распределение_вагонов
    Convert_rule
        if (_заявка.остаток_в_заявке >
_формируемый_состав.оставшееся_число_вагонов)
            буфер =
_формируемый_состав.оставшееся_число_вагонов;
        else
            буфер = _заявка.остаток_в_заявке;
            остаток_в_заявке -= _заявка.буфер;
            состояние = списание_закончено;
            номер_последнего_состава =
_формируемый_состав.номер_формируемого_состава;
            _формируемый_состав
                Choice from _формируемый_состав.состояние == распределение
                    and _заявка.маршрут ==
_формируемый_состав.маршрут
                Convert_rule
                    оставшееся_число_вагонов -= _заявка.буфер;
                    состояние = списание;
$End

$Pattern Образец_проверки_формируемого_состава: rule
$Relevant_resources
    _формируемый_состав: формируемые_составы Keep
$Body
_формируемый_состав
    Choice from _формируемый_состав.состояние == списание
    Convert_rule
        if (_формируемый_состав.оставшееся_число_вагонов == 0)
            состояние = заполнен;
        else
            состояние = распределение;
$End

$Pattern Образец_проверки_аренды: operation trace
$Relevant_resources
    _парк: парк Keep Keep
$Time = 10
$Body

```

```

_парк
    Choice from _парк.количество_вагонов_в_парке < вагоны_в_составе
        and _парк.заказ_аренды == false
    Convert_begin
        заказ_аренды = true;
    Convert_end
        количество_вагонов_в_парке += вагоны_в_составе;
        заказ_аренды = false;
$End

$Pattern Образец_проверки_заявки: rule
$Relevant_resources
    _заявка: заявки          Keep
$Body
_заявка
    Choice from _заявка.состояние == списание_закончено
    Convert_rule
        if (_заявка.остаток_в_заявке == 0)
            состояние = распределена;
        else
            состояние = распределение_вагонов;
$End

$Pattern Образец_готовности_состава: rule
$Relevant_resources
    _формируемый_состав: формируемые_составы Keep
    _парк                : парк                Keep
    _состав              : составы             Create
    first
$Body
_формируемый_состав
    Choice from _формируемый_состав.состояние == заполнен
    Convert_rule
        состояние = следующий;
_парк
    Choice from _парк.количество_вагонов_в_парке >=
вагоны_в_составе
    Convert_rule
        количество_вагонов_в_парке -= вагоны_в_составе;
_состав
    Convert_rule trace

```

```

        маршрут          = _формируемый_состав.маршрут;
        номер_состава          =
_формируемый_состав.номер_формируемого_состава;
        Время_создания      = Time_now;
$End

$Pattern Образец_назначения_нового_номера_состава: rule
$Relevant_resources
    _формируемый_состав: формируемые_составы Keep
$Body
    _формируемый_состав
        Choice from _формируемый_состав.состояние == следующий
        Convert_rule
            номер_формируемого_состава += 1;
            оставшееся_число_вагонов      = 60;
            состояние                      = распределение;
$End

$Pattern Образец_создания_состава: operation
$Relevant_resources
    _состав          : составы          Keep Keep
    first
$Time = ф_создания_состава (35.0, 45.0)
$Body
    _состав
        Choice from _состав.состояние == сформирован
        Convert_begin
            состояние = создается;
        Convert_end
            if (_состав.маршрут == 1)
                Движение_составов_1.ProcessStart(_состав);
            if (_состав.маршрут == 2)
                Движение_составов_2.ProcessStart(_состав);
            if (_состав.маршрут == 3)
                Движение_составов_3.ProcessStart(_состав);
$End

$Pattern Образец_списывания_обслуженных_заявок: rule
$Relevant_resources
    _состав : составы Keep
    _заявка : заявки Keep

```

```

        _счетчик: счетчик Keep
        first
$Body
    _состав
        Choice from _состав.состояние == прибыл_в_парк
        Convert_rule
    _заявка
        Choice      from      _заявка.номер_последнего_состава      ==
_состав.номер_состава
                        and _заявка.маршрут                        ==
_состав.маршрут
                        and _заявка.состояние                        ==
распределена
        Convert_rule
            состояние = обслужена;
    _счетчик
        Convert_rule
            количество_обслуженных_заявок +=1;
            время                          = Time_now;
            время_обслуживания              += _счетчик.время -
_заявка.время_прихода;
            среднее_время_обслуживания    =
_счетчик.время_обслуживания/_счетчик.количество_обслуженных_заявок;
$End

$Pattern Образец_возврата_вагонов_в_парк: operation
$Relevant_resources
    _состав: составы Keep      Keep
    _парк  : парк      NoChange Keep
$Time = ф_возврата_вагонов_в_парк(35.0, 50.0)
$Body
    _состав
        Choice from _состав.состояние == прибыл_в_парк
        Convert_begin
            состояние = расформирование;
        Convert_end
            состояние = расформирован;
    _парк
        Convert_end
            количество_вагонов_в_парке += вагоны_в_составе;
$End

```

```

$Pattern Образец_удаления_состава: rule
$Relevant_resources
    _состав : составы Erase
    _счетчик: счетчик Keep
first
$Body
_состав
    Choice from _состав.состояние == расформирован
_счетчик
    Convert_rule
        количество_удаленных_составов += 1;
$End

```

```

$Pattern Образец_удаления_заявки: operation
$Relevant_resources
    _заявка : заявки Keep Erase
    _счетчик: счетчик Keep NoChange
first
$Time = 0
$Body
_заявка
    Choice from _заявка.состояние == обслужена
    Convert_begin
        состояние = delete;
        время_обслуживания = time_now - _заявка.время_прихода;
_счетчик
    Convert_begin
        количество_удаленных_заявок += 1;
$End

```

### Файл model.dpt (точки принятия решений):

```

$Decision_point формирование_составов_на_отправку: some
$Activities
    расчет_количества_вагонов :
Образец_расчета_количества_вагонов
    согласование_заявки_и_формсостава : Образец_списания_вагонов
    проверка_состояния_формсостава :
Образец_проверки_формируемого_состава
    проверка_состояния_заявки : Образец_проверки_заявки

```

```

назначение_нового_номера_составу :
Образец_назначения_нового_номера_состава
    готовность_состава : Образец_готовности_состава
    создание_состава : Образец_создания_состава
    проверка_аренды : Образец_проверки_аренды
$End

$Decision_point общие_действия: prior
$Activities
    списывание_заявок :
Образец_списывания_обслуженных_заявок cf = 1
    возврат_вагонов_в_парк :
Образец_возврата_вагонов_в_парк
    удаление_состава :
Образец_удаления_состава
    удаление_заявки :
Образец_удаления_заявки
$End

```

### Файл model.prc (процессы обслуживания):

```

$Process Движение_составов_1 составы
    SEIZE    ПогрузкаКульсары
    ADVANCE  ф_погрузки (18.0, 22.0)
    RELEASE  ПогрузкаКульсары

    ADVANCE  ф_время_в_пути(550.0) //Кульсары    Аксарайская

    SEIZE    ТаможняАксарайская
    ADVANCE  ВремяПрохожденияТаможни(1, 5) //Аксарайская
    RELEASE  ТаможняАксарайская

    ADVANCE  ф_время_в_пути(2200.0) //Аксарайская Кошта

    SEIZE    РазгрузкаКошта
    ADVANCE  ВремяРазгрузки(3, 8) //Кошта
    RELEASE  РазгрузкаКошта

    SEIZE    МойкаКошта
    ADVANCE  ВремяМойки(2, 3) //Кошта
    RELEASE  МойкаКошта

```

```
ADVANCE    ф_время_в_пути(2750.0) //Кошта Кульсары

$End

$Process Движение_составов_2 составы
SEIZE      ПогрузкаКульсары
ADVANCE    ф_погрузки (18.0, 22.0)
RELEASE    ПогрузкаКульсары

ADVANCE    ф_время_в_пути(550.0) //Кульсары    Аксарайская

SEIZE      ТаможняАксарайская
ADVANCE    ВремяПрохожденияТаможни(1,5)//Аксарайская
RELEASE    ТаможняАксарайская

ADVANCE    ф_время_в_пути(830.0) //Аксарайская    Гуково

SEIZE      ТаможняГуково
ADVANCE    ВремяПрохожденияТаможни(1,5)//Гуково
RELEASE    ТаможняГуково

ADVANCE    ф_время_в_пути(1650.0) //Гуково    Батеве

SEIZE      ТаможняБатеве
ADVANCE    ВремяПрохожденияТаможни(1,5)//Батеве
RELEASE    ТаможняБатеве

ADVANCE    ф_время_в_пути(20.0) //Батеве    Тужер

SEIZE      РазгрузкаТужер
ADVANCE    ВремяРазгрузки(1,5)//Тужер
RELEASE    РазгрузкаТужер

SEIZE      МойкаТужер
ADVANCE    ВремяМойки(1,5)//Тужер
RELEASE    МойкаТужер

ADVANCE    ф_время_в_пути(3050.0) //Тужер    Кульсары

TERMINATE
```

\$End

\$Process Движение\_составов\_3 составы

```

SEIZE   ПогрузкаКульсары
ADVANCE ф_погрузки (18.0, 22.0)
RELEASE ПогрузкаКульсары

ADVANCE ф_время_в_пути(550.0) //Кульсары Аксарайская

SEIZE   ТаможняАксарайская
ADVANCE ВремяПрохожденияТаможни(1,5)//Аксарайская
RELEASE ТаможняАксарайская

ADVANCE ф_время_в_пути(1740.0) //Аксарайская Красное

SEIZE   ТаможняКрасное
ADVANCE ВремяПрохожденияТаможни(1,5)//Красное
RELEASE ТаможняКрасное

ADVANCE ф_время_в_пути(590.0) //Красное Брест

SEIZE   ТаможняБрест
ADVANCE ВремяПрохожденияТаможни(1,5)//Брест
RELEASE ТаможняБрест

SEIZE   СменаКолесныхПарБрест
ADVANCE ВремяСменыКолесныхПар(1,5)//Брест
RELEASE СменаКолесныхПарБрест

ADVANCE ф_время_в_пути(14.0) //Брест Малашевиче

SEIZE   РазгрузкаМалашевиче
ADVANCE ВремяРазгрузки(1,5)//Малашевиче
RELEASE РазгрузкаМалашевиче

SEIZE   МойкаМалашевиче
ADVANCE ВремяМойки(1,5)//Малашевиче
RELEASE МойкаМалашевиче

ADVANCE ф_время_в_пути(14.0) //Малашевиче Брест

```



```

SEIZE    СменаКолесныхПарБрест
ADVANCE  ВремяСменыКолесныхПар(1,5)//Брест
RELEASE  СменаКолесныхПарБрест

ADVANCE  ф_время_в_пути(2880.0) //Брест Кульсары

```

```
$End
```

### Файл model.fun (константы, последовательности и функции):

```

$Constant
    тоннаж : real = 60.0
    вагоны_в_составе: integer = 60
$End

$Sequence ф_приход_заявок: real [0..3000]
$Type = uniform 3487478
$End

$Sequence ф_вес_груза : real
$Type = uniform 34556
$End

$Sequence ф_тип_маршрута: such_as заявки.маршрут
$Type = by_hist 1111
$Body
    1 2 5.0
    2 3 3.0
    3 3 3.0
$End

$Sequence ф_создания_состава : real[0..100000]
$Type = uniform 235
$End

$Sequence ф_погрузки : real[0..100000]
$Type = uniform 768434
$End

$Function ф_время_в_пути : real
$Type = algorithmic

```

```
$Parameters
```

```
  x : real
```

```
$Body
```

```
  return x*24/500;
```

```
$End
```

```
$Sequence ВремяПрхожденияТаможни : real[0..100000]
```

```
$Type = uniform 545478
```

```
$End
```

```
$Sequence ВремяСменыКолесныхПар : real[0..100000]
```

```
$Type = uniform 54747848
```

```
$End
```

```
$Sequence ВремяРазгрузки : real[0..100000]
```

```
$Type = uniform 321258
```

```
$End
```

```
$Sequence ВремяМойки : real[0..100000]
```

```
$Type = uniform 45789211
```

```
$End
```

```
$Sequence  $\phi$ _возврата_вагонов_в_парк : real[0..100000]
```

```
$Type = uniform 80504030890
```

```
$End
```

### Файл model.smr (прогон):

```
Show_mode      = NoShow
```

```
Show_rate      = 3600.0
```

```
Образец_прихода_заявок.planning(time_now +  $\phi$ _приход_заявок (34, 38))
```

```
Terminate_if счетчик.количество_обслуженных_заявок >= 20
```

## ПРИЛОЖЕНИЕ 2. Листинг файла описания ресурсов и типов ресурсов в РДО

```

#include "rdo_lib/rdo_runtime/pch.h"
#include "rdo_lib/rdo_runtime/rdo_resource.h"
#include "rdo_lib/rdo_runtime/rdo_runtime.h"
#include "rdo_lib/rdo_runtime/rdoprocess.h"
#include "rdo_common/smart_ptr/intrusive_ptr.h"

OPEN_RDO_RUNTIME_NAMESPACE

// -----
// ----- RDOResourceType
// -----
-----

RDOResourceType::RDOResourceType(rsint    number,    RDORuntimeParent*
parent)
    : RDORuntimeObject    (parent)
    , RDOTraceableObject(false )
    , m_id                (number)
{}

RDOResourceType::~RDOResourceType ()
{}

LPRDOResource    RDOResourceType::createRes (PTR (RDORuntime)    runtime,
CREF (std::vector<RDOValue>)    paramsCalcs,    rbool    traceFlag,    rbool
permanentFlag)
{
    LPRDOResourceType    pResType (this);
    ASSERT (pResType);
    LPIResourceType                pIResType                =
pResType.interface_cast<IResourceType> ();
    ASSERT (pIResType);
    return    rdo::Factory<RDOResource>::create (runtime,    paramsCalcs,
pIResType,    runtime->getResourceId(),    this->getTraceID(),    traceFlag,
permanentFlag);
}

```

```

// -----
-----
// ----- RDOResourceTypeTransact
// -----
-----

RDOResourceTypeTransact::RDOResourceTypeTransact(rsint      number,
RDORuntimeParent* parent)
    : RDORuntimeObject (parent)
    , RDOTraceableObject(false )
    , m_id              (number)
{}

RDOResourceTypeTransact::~RDOResourceTypeTransact ()
{}

LPRDOResource      RDOResourceTypeTransact::createRes (PTR (RDORuntime)
runtime, CREF(std::vector<RDOValue>) paramsCalcs, rbool traceFlag, rbool
permanentFlag)
{
    LPRDOResourceTypeTransact pResType(this);
    ASSERT(pResType);
    LPIResourceType           pIResType           =
pResType.interface_cast<IResourceType>();
    ASSERT(pIResType);
    return                    rdo::Factory<RDOPROCTransact>::create(runtime,
paramsCalcs, pIResType, runtime->getResourceId(), this->getTraceID(),
traceFlag, permanentFlag);
}

// -----
-----
// ----- RDOResourceTypeProcess
// -----
-----

RDOResourceTypeProcess::RDOResourceTypeProcess(rsint      number,
RDORuntimeParent* parent)
    : RDORuntimeObject (parent)
    , RDOTraceableObject(false )
    , m_id              (number)
{}

```

```

RDOResourceTypeProcess::~~RDOResourceTypeProcess()
{}

LPRDOResource RDOResourceTypeProcess::createRes(PTR(RDORuntime)
runtime, CREF(std::vector<RDOValue>) paramsCalcs, rbool traceFlag, rbool
permanentFlag)
{
    LPRDOResourceTypeProcess pResType(this);
    ASSERT(pResType);
    LPIResourceType pIResType =
pResType.interface_cast<IResourceType>();
    ASSERT(pIResType);
    return rdo::Factory<RDOPROCResource>::create(runtime,
paramsCalcs, pIResType, runtime->getResourceId(), this->getTraceID(),
traceFlag, permanentFlag);
}

CLOSE_RDO_RUNTIME_NAMESPACE

/**
@file rdo_res_type.h
@authors Урусов Андрей, Лушан Дмитрий
@date 07.06.2011
@brief Типы ресурсов в runtime
@indent 4T
*/

#ifndef RDO_RES_TYPE_H
#define RDO_RES_TYPE_H

#include "rdo_lib/rdo_runtime/rdotrace.h"
#include "rdo_lib/rdo_runtime/rdo_res_type_i.h"

OPEN_RDO_RUNTIME_NAMESPACE

class RDORuntime;
class RDOResource;
class RDOPROCTransact;

```

```

// -----
-----
// ----- RDOResourceType
// -----
-----

OBJECT (RDOResourceType)
    IS IMPLEMENTATION_OF (IResourceType )
    AND INSTANCE_OF      (RDORuntimeObject )
    AND INSTANCE_OF      (RDOTraceableObject)

{
DECLARE_FACTORY (RDOResourceType);
friend class RDOCalcCreateResource;
friend class RDOPROCGenerate;
private:
    RDOResourceType (rsint number, PTR (RDORuntimeParent) pParent =
NULL);
    virtual ~RDOResourceType ();

    rsint m_id;

    DECLARE_IResourceType;
};

// -----
-----
// ----- RDOResourceTypeTransact
// -----
-----

OBJECT (RDOResourceTypeTransact)
    IS IMPLEMENTATION_OF (IResourceType )
    AND INSTANCE_OF      (RDORuntimeObject )
    AND INSTANCE_OF      (RDOTraceableObject)

{
DECLARE_FACTORY (RDOResourceTypeTransact);
private:
    RDOResourceTypeTransact (rsint number, PTR (RDORuntimeParent)
pParent = NULL);
    virtual ~RDOResourceTypeTransact ();

    rsint m_id;

```

```

        DECLARE_IResourceType;
};

// -----
-----
// ----- RDOResourceTypeProcess
// -----
-----

OBJECT (RDOResourceTypeProcess)
    IS IMPLEMENTATION_OF (IResourceType )
    AND INSTANCE_OF      (RDORuntimeObject )
    AND INSTANCE_OF      (RDOTraceableObject)
{
    DECLARE_FACTORY (RDOResourceTypeProcess);
private:
    RDOResourceTypeProcess (rsint number, PTR (RDORuntimeParent)
pParent = NULL);
    virtual ~RDOResourceTypeProcess ();

    rsint m_id;

    DECLARE_IResourceType;
};

CLOSE_RDO_RUNTIME_NAMESPACE

#endif // RDO_RES_TYPE_H

/**
 * @file rdo_resource.cpp
 * @authors Урусов Андрей, Луцан Дмитрий
 * @date unknown
 * @brief RDOResource implementation
 * @indent 4T
 */

//
=====
INCLUDES

```

```

//
=====
SYNOPSIS
#include "rdo_lib/rdo_runtime/pch.h"
#include "rdo_lib/rdo_runtime/rdo_resource.h"
#include "rdo_lib/rdo_runtime/rdo_runtime.h"
//
=====
=====

OPEN_RDO_RUNTIME_NAMESPACE

// -----
-----
// ----- RDOResource
// -----
-----

RDOResource::RDOResource (PTR (RDORuntime) runtime,
CREF(std::vector<RDOValue>) paramsCalcs, LPIResourceType pResType, ruInt
resID, ruInt typeID, rbool trace, rbool temporary)
    : RDORuntimeObject (NULL )
    , RDOTraceableObject (trace, resID, rdo::toString(resID + 1))
    , RDORuntimeContainer(runtime )
    , m_state (RDOResource::CS_None )
    , m_type (typeID )
    , m_referenceCount (0 )
    , m_resType (pResType )
    , m_temporary (temporary )
{
    appendParams(paramsCalcs.begin(), paramsCalcs.end());
    runtime->insertNewResource(this);
}

RDOResource::RDOResource (PTR (RDORuntime) runtime, CREF (RDOResource)
copy)
    : RDORuntimeObject (NULL )
    , RDOTraceableObject (copy.traceable(), copy.getTraceID(),
copy.traceId())
    , RDORuntimeContainer(runtime )
    , m_type (copy.m_type )
    , m_state (copy.m_state )

```



```

    , m_typeId          (copy.m_typeId   )
    , m_params          (copy.m_params   )
    , m_referenceCount  (0               )
    , m_resType         (copy.m_resType  )
    , m_temporary       (copy.m_temporary)
{
    appendParams(copy.m_params.begin(), copy.m_params.end());
    runtime->insertNewResource(this);
    //! @TODO посмотреть history и принять решение и комментарии
    //  getRuntime()->incrementResourceIdReference( getTraceID() );
}

RDOResource::~RDOResource()
{
    //! TODO: Дима, поставь тут breakpoint и посмотри на this
    //getRuntime()->fireMessage(RDORuntime::RO_BEFOREDELETE,
(void*)getTraceID());
    //getRuntime()->onResourceErase(this);
}

bool RDOResource::operator!=(RDOResource &other)
{
    if ( m_type != other.m_type ) return true;
    if ( m_params.size() != other.m_params.size() ) return true;

    int size = m_params.size();
    for ( int i = 0; i < size; i++ ) {
        if ( m_params.at(i) != other.m_params.at(i) ) return
true;
    }
    return false;
}

LPRDOResource RDOResource::clone(PTR(RDORuntime) runtime) const
{
    return rdo::Factory<RDOResource>::create(runtime, m_params,
m_resType, getTraceID(), m_type, traceable(), m_temporary);
}

std::string RDOResource::getTypeId()
{

```

```

        std::ostringstream str;
        str << m_type;
        return str.str();
    }

    std::string RDOResource::traceParametersValue()
    {
        std::ostringstream str;
        if(m_params.size() > 0)
        {
            std::vector<RDOValue>::iterator end = m_params.end();
            for(std::vector<RDOValue>::iterator it =
m_params.begin();;)
            {
#ifdef RDOSIM_COMPATIBLE
                std::ostringstream _str;
                _str << *it;
                std::string::size_type pos = _str.str().find( "e"
);

                if ( pos != std::string::npos ) {
                    std::string __str = _str.str();
                    __str.erase( pos + 2, 1 );
                    str << __str.c_str();
                } else {
                    str << _str.str().c_str();
                }
            }
#else
                str << *it;
#endif
            if(++it == end)
                break;
            str << " ";
        }
        return str.str();
    }

    std::string RDOResource::traceResourceState( char prefix,
RDOSimulatorTrace* sim )
    {
        std::ostringstream res;

```

```

        if ( traceable() || (prefix != '\0') ) {
            if ( m_state == RDOResource::CS_NoChange || m_state ==
RDOResource::CS_NonExist ) return "";
            if ( prefix != '\0' ) res << prefix;
            switch ( m_state ) {
                case RDOResource::CS_Create: res << "RC "; break;
                case RDOResource::CS_Erase : res << "RE "
#ifdef RDOSIM_COMPATIBLE
                    << sim->getCurrentTime() << " " <<
traceTypeId() << " " << traceId() << std::endl; return res.str();
                #else
                    ;
                #endif
                break;
                default : res << "RK ";
break;
            }
            res << sim->getCurrentTime() << " " << traceTypeId() << "
" << traceId() << " " << traceParametersValue() << std::endl;
        }
        return res.str();
    }

CLOSE_RDO_RUNTIME_NAMESPACE

/**
 * @file rdo_resource.h
 * @authors Урусов Андрей, Лушан Дмитрий
 * @date 03.06.2011
 * @brief Ресурсы в runtime
 * @indent 4T
 */

#ifdef RDO_RESOURCE_H
#define RDO_RESOURCE_H

//
=====
INCLUDES

```

```

//
=====
SYNOPSIS
#include "rdo_lib/rdo_runtime/rdotrace.h"
#include "rdo_lib/rdo_runtime/rdo_object.h"
#include "rdo_lib/rdo_runtime/rdo_value.h"
//
=====
=====

OPEN_RDO_RUNTIME_NAMESPACE

class RDORuntime;
PREDECLARE_OBJECT_INTERFACE(IResourceType);

// -----
-----
// ----- RDOResource
// -----
-----

OBJECT(RDOResource) IS INSTANCE_OF(RDORuntimeObject) AND
INSTANCE_OF(RDOTraceableObject) AND INSTANCE_OF(RDORuntimeContainer)
{
friend class RDOResourceType;
public:
enum ConvertStatus {
    CS_None = 0,
    CS_Keep,
    CS_Create,
    CS_Erase,
    CS_NonExist,
    CS_NoChange
};

    RDOResource(PTR(RDORuntime) runtime,
    CREF(std::vector<RDOValue>) paramsCalcs, LPIResourceType pResType, ruInt
resID, ruInt typeID, rbool trace, rbool temporary);
    RDOResource(PTR(RDORuntime) runtime, CREF(RDOResource) copy);
    virtual ~RDOResource();

    void setRuntime(RDORuntime* runtime);

```

```

ConvertStatus      getState      (          ) const;
CREF(RDOValue)    getParam      (ruint index) const;
rbool             checkType     (ruint type ) const;
rbool             canFree       (          ) const;
CREF(LPIResourceType) getResType (          ) const;
ruint             getType       (          ) const;
virtual ruint     paramsCount   (          ) const;
LPRDOResource     clone         (PTR(RDORuntime) runtime)
const;

CREF(std::vector<RDOValue>) getParams(          ) const;

void              makeTemporary          (rbool value
);

void              setState                (ConvertStatus value
);

tstring          traceResourceState      (char prefix,
PTR(RDOSimulatorTrace) sim);
REF(RDOValue)    getParamRaw            (ruint index
);

void              setParam                (ruint index,
CREF(RDOValue) value          );
virtual void     appendParams            (const
std::vector<RDOValue>::const_iterator& from_begin,      const
std::vector<RDOValue>::const_iterator& from_end);
tstring          getTypeId               ();
tstring          traceParametersValue    ();
virtual tstring  whoAreYou                ();
void             incRef                   ();
void             decRef                   ();
bool operator!= (REF(RDOResource) other);

protected:
std::vector<RDOValue> m_params;
bool                 m_temporary;
ConvertStatus        m_state;

private:
ruint                m_type;
ruint                m_referenceCount;
LPIResourceType      m_resType;

```

```
tstring      m_typeId;

tstring traceTypeId();
};

CLOSE_RDO_RUNTIME_NAMESPACE

#include "rdo_lib/rdo_runtime/rdo_resource.inl"

#endif // RDO_RESOURCE_H
```

## ОГЛАВЛЕНИЕ

РЕФЕРАТ-----	6
ПЕРЕЧЕНЬ СОКРАЩЕНИЙ, СИМВОЛОВ И СПЕЦИАЛЬНЫХ ТЕРМИНОВ С ИХ ОПРЕДЕЛЕНИЕМ	7
ВВЕДЕНИЕ -----	8
1 ПРЕДПРОЕКТНОЕ ИССЛЕДОВАНИЕ -----	10
1.1 Назначение программного комплекса РДО -----	10
1.2 Функции программного комплекса -----	12
1.3 Предпосылки интеграции подходов дискретного имитационного моделирования в РДО	14
1.4 Выводы по преддипломному этапу -----	19
2 КОНЦЕПТУАЛЬНОЕ ПРОЕКТИРОВАНИЕ-----	21
2.1 Цели разработки системы-----	21
2.2 Компоненты РДО-----	22
2.3 Новые конструкции в языке РДО-----	23
2.4 Ресурсы модели -----	23
2.5 Типы ресурсов модели -----	24
2.6 Разработка технического задания -----	25
2.6.1 Введение -----	25
2.6.2 Основания для разработки -----	25
2.6.3 Назначение разработки-----	26
2.6.4 Требования к программе или программному изделию-----	26
2.6.5 Техничко-экономические показатели -----	28
2.6.6 Стадии и этапы разработки -----	29
2.6.7 Порядок контроля и приемки -----	30
2.6.8 Приложения -----	30
3 ТЕХНИЧЕСКОЕ ПРОЕКТИРОВАНИЕ -----	31
3.1 Проектирование новых синтаксических конструкций-----	31
3.2 Проектирование типов ресурсов-----	31
3.3 Проектирование процессов обслуживания-----	32
4 РАБОЧЕЕ ПРОЕКТИРОВАНИЕ -----	34
4.1 Реализация нового синтаксиса -----	34
4.2 Изменения в компоненте имен rdoParse -----	35
5 ИССЛЕДОВАТЕЛЬСКАЯ ЧАСТЬ-----	37
5.1 Альтернативные решения -----	37
5.2 Сравнение альтернативных решений -----	37

5.3	Основной критерий сравнения -----	38
5.4	Выводы по результатам исследований -----	39
6	ОРГАНИЗАЦИОННО-ЭКОНОМИЧЕСКАЯ ЧАСТЬ-----	40
6.1	Организация и планирование процесса разработки программного продукта -----	40
6.1.1	Расчет трудоемкости разработки технического задания-----	43
6.1.2	Расчет трудоемкости выполнения эскизного проекта -----	43
6.1.3	Расчет трудоемкости выполнения технического проекта -----	44
6.1.4	Расчет трудоемкости разработки рабочего проекта-----	45
6.1.5	Расчет трудоемкости выполнения внедрения -----	47
6.1.6	Расчет суммарной трудоемкости-----	47
6.2	Определение стоимости разработки ПП -----	49
6.2.1	Расчёт основной заработной платы-----	49
6.2.2	Расчёт дополнительной заработной платы-----	50
6.2.3	Отчисления на социальное страхование-----	50
6.2.4	Накладные расходы -----	50
6.2.5	Расходы на амортизацию оборудования -----	51
6.2.6	Результаты расчетов затрат на разработку программного продукта. -----	51
6.3	Определение стоимости разработки системы-----	52
7	МЕРОПРИЯТИЯ ПО ОХРАНЕ ТРУДА И ТЕХНИКЕ БЕЗОПАСНОСТИ-----	53
7.1	Опасные и вредные факторы-----	53
7.1.1	Физические -----	53
7.1.2	Химические-----	54
7.1.3	Психофизиологические -----	54
7.1.4	Биологические -----	54
7.2	Требования к помещениям для работы с ПЭВМ -----	54
7.2.1	Требования к микроклимату, содержанию аэроионов и вредных химических веществ в воздухе на рабочих местах, оборудованных ПЭВМ-----	56
7.2.2	Требования к уровням шума и вибрации на рабочих местах, оборудованных ПЭВМ -----	58
7.2.3	Требования к освещению на рабочих местах, оборудованных ПЭВМ -----	59
7.2.4	Требования к уровням электромагнитных полей на рабочих местах, оборудованных ПЭВМ 62	
7.2.5	Требования к визуальным параметрам ВДТ, контролируемым на рабочих местах-----	62
7.2.6	Общие требования к организации рабочих мест пользователей ПЭВМ-----	63
7.2.7	Требования к организации и оборудованию рабочих мест с ПЭВМ для взрослых пользователей-----	64
7.2.8	Электробезопасность рабочего помещения -----	66
7.2.9	Требования пожарной безопасности -----	67
7.2.10	Требования к организации медицинского обслуживания пользователей ПЭВМ -----	68
7.2.11	Требования к проведению государственного санитарно-эпидемиологического надзора и производственного контроля-----	68
7.2.12	Расчет системы защитного заземления компьютера -----	69



7.3	Утилизация ПЭВМ	72
7.3.1	Разборка изделий	73
7.3.2	Реализация партий	78
ЗАКЛЮЧЕНИЕ		92
СПИСОК ЛИТЕРАТУРЫ		93
ПРИЛОЖЕНИЕ 1. Код имитационной модели-прототипа грузоперевозок.		94
ПРИЛОЖЕНИЕ 2. Листинг файла описания ресурсов и типов ресурсов в РДО		107
ОГЛАВЛЕНИЕ		119