

*А.А. Саломатина, Ю.Н. Фомина*

**Методические рекомендации  
по выполнению СРС**

***РЕИНЖИНИРИНГ БИЗНЕС-ПРОЦЕССОВ  
ПРОЕКТИРОВАНИЯ И ПРОИЗВОДСТВА***

Под ред. к.т.н., доцента Е.И. Яблочникова

**ПРИЛОЖЕНИЕ II**

Санкт-Петербург 2008

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ



ПОБЕДИТЕЛЬ КОНКУРСА ИННОВАЦИОННЫХ ОБРАЗОВАТЕЛЬНЫХ ПРОГРАММ ВУЗОВ

**А.А. Саломатина, Ю.Н. Фомина**

## **Методические рекомендации по выполнению СРС**

***РЕИНЖИНИРИНГ БИЗНЕС-ПРОЦЕССОВ  
ПРОЕКТИРОВАНИЯ И ПРОИЗВОДСТВА***

**Под ред. к.т.н., доцента Е.И. Яблочникова**

**ПРИЛОЖЕНИЕ II**

*Методические рекомендации по выполнению СРС*



**Санкт-Петербург**

**2008**

А.А. Саломатина, Ю.Н. Фомина. Реинжиниринг бизнес-процессов проектирования и производства / Приложение II. Методические рекомендации по выполнению СРС. Под ред. к.т.н., доцента Е.И. Яблочникова – СПб: СПбГУИТМО, 2008. – 84 с.

Данные методические рекомендации служат для практического ознакомления студентов с методологией проведения реинжиниринга бизнес-процессов. Рассматриваются методика проведения предпроектного анализа предприятия с помощью системы визуального и имитационного моделирования бизнес-процессов Adonis. Приводится механизм выполнения сравнительного анализа различных сценариев деятельности отделов предприятия. Дается описание механизма проектирования диаграмм UML. Рассматривается методология технической подготовки производства на основе механизма WorkFlow PDM SmarTeam.

В результате выполнения самостоятельных работ у студентов должно сформироваться четкое представление о методологии проведения реинжиниринга бизнес-процессов.

Данные рекомендации предназначены для магистров, обучающихся по направлению подготовки 200100 – Приборостроение по магистерской программе «Управление жизненным циклом приборов и систем».

Рекомендовано УМО по образованию в области приборостроения и оптотехники в качестве учебного пособия для студентов высших учебных заведений, обучающихся по направлению подготовки 200100 – Приборостроение.



В 2007 году СПбГУ ИТМО стал победителем конкурса инновационных образовательных программ вузов России на 2007–2008 годы. Реализация инновационной образовательной программы «Инновационная система подготовки специалистов нового поколения в области информационных и оптических технологий» позволит выйти на качественно новый уровень подготовки выпускников и удовлетворить возрастающий спрос на специалистов в информационной, оптической и других высокотехнологичных отраслях экономики.

©Санкт-Петербургский государственный университет информационных технологий, механики и оптики, 2008

© А.А Саломатина, Ю.Н. Фомина, Е.И. Яблочников.

## Оглавление

Самостоятельная работа «Анализ бизнес-процессов конструкторского бюро приборостроительного предприятия».....	6
Предпроектный анализ бизнеса .....	6
Задание .....	8
Работа в системе Adonis.....	9
<i>Path analysis</i> .....	17
<i>Capacity Analysis</i> .....	19
<i>Workload Analysis</i> .....	20
Самостоятельная работа «Реинжиниринг бизнес-процессов конструкторского бюро приборостроительного предприятия» .....	22
Реинжиниринг бизнес-процессов.....	22
Задание .....	25
Работа в системе Adonis.....	26
<i>Flowmark Audit Trail Evaluation</i> .....	27
<i>Comparative representation of result</i> .....	27
<i>Evaluation queries</i> .....	30
Самостоятельная работа «Моделирование ТПП с использованием диаграмм последовательности и кооперации UML».....	34
Цель работы.....	34
Особенности построения диаграмм последовательности в нотации UML.....	34
<i>Общие положения</i> .....	34
<i>Объекты на диаграмме последовательности</i> .....	35
<i>Линии жизни на диаграмме последовательности</i> .....	36
<i>Фокус управления диаграммы деятельности</i> .....	37
<i>Сообщения на диаграммах деятельности</i> .....	38
<i>Ветвление потока управления на диаграммах деятельности</i> .....	40

<i>Временные ограничения на диаграммах последовательности</i> .....	41
<i>Рекомендации по построению диаграмм последовательности</i> .....	41
Особенности построения диаграмм кооперации в нотации UML.....	42
<i>Общие положения</i> .....	42
<i>Кооперация</i> .....	43
<i>Связи на диаграммах кооперации</i> .....	44
<i>Сообщения на диаграммах кооперации</i> .....	44
<i>Рекомендации по построению диаграмм кооперации</i> .....	45
Руководство к выполнению самостоятельной работы .....	47
Пример выполнения .....	47
<i>Разработка модели последовательности</i> .....	47
<i>Разработка модели кооперации</i> .....	48
Содержание отчёта .....	49
Самостоятельная работа «Моделирование ТПП с использованием диаграмм деятельности UML».....	50
Цель работы.....	50
Особенности построения диаграмм деятельности в нотации UML....	50
<i>Общие положения</i> .....	50
<i>Действия на диаграмме деятельности</i> .....	53
<i>Переходы на диаграмме деятельности</i> .....	54
<i>Дорожки диаграммы деятельности</i> .....	56
<i>Объекты на диаграммах деятельности</i> .....	56
<i>Рекомендации по построению диаграмм деятельности</i> .....	57
Руководство к выполнению самостоятельной работы .....	60
Пример выполнения .....	60
Содержание отчёта .....	61
Самостоятельная работа «Разработка комплекса графиков производственных процессов КБ предприятия на основе методологии Workflow».....	61

Управление бизнес-процессами на основе технологии Workflow .....	61
Задание .....	70
Разработка графиков Workflow .....	70
Приложение 1 .....	75
<i>Входные параметры для элементов Activity функциональной модели</i> .....	75
<i>Входные параметры для элемента начала процесса функциональной модели</i> .....	75
<i>Входные параметры для элементов Performer организационной модели</i> .....	76
Приложение 2. Варианты заданий для самостоятельного выполнения .....	81
Список литературы .....	82

# **Самостоятельная работа «Анализ бизнес-процессов конструкторского бюро приборостроительного предприятия»**

## **Предпроектный анализ бизнеса**

Предпроектный анализ предприятия заключается в исследовании деятельности компании и построении концептуальной модели предприятия, используя средства визуального и имитационного моделирования. Это позволяет выявить проблемные места и модифицировать структуру бизнес-процессов компании.

Комплекс из функционального, организационного и информационного анализа даёт возможность всестороннего описания предметной области.

Функциональное моделирование является важнейшим элементом концептуального анализа, который выполняется на начальном этапе проектирования любой автоматизированной информационной системы, в том числе и системы управления предприятием. Разработка и анализ функциональной модели деятельности предприятия позволяет достаточно глубоко погрузиться в предметную область, выявить бизнес-процессы, используемые на предприятии, определить информационные потоки, выявить «узкие» места в деятельности предприятия и т.д.

С помощью функционального анализа исследуются функциональные связи и последовательности выполнения функций, задач и процедур компании. Данный анализ основывается на многоуровневом рассмотрении предметной области: на основе декомпозиции каждая функция разделяется на задачи, а те в свою очередь разделяются на процедуры. Это означает, что на каждом уровне модель должна отвечать на вопросы пользователей с заданной степенью детализации. Функциональный анализ выполняется на основе соответствующей методической и инструментальной поддержки, а также с учётом информации о предприятии (конкретном и среднем).

Результатом функционального анализа является получение ряда функциональных моделей, отражающих деятельность конкретного предприятия. Их анализ позволит локализовать проблемные места, оценить преимущества использования новой системы.

Информационный анализ заключается в исследовании информационных потоков, циркулирующих между бизнес-процессами. Информационный анализ выполняется для определения взаимодействия между подразделениями предприятия и состава передаваемой документации. Результаты информационного анализа необходимы в дальнейшем при

классификации объектов. Они отражаются в PDM-системе как элементы описания, характеристики, атрибуты. Т.е. для создания классов объектов необходимо оперировать данными как информационной, так и функциональной моделей.

Организационный анализ позволяет представить состав и структуру подразделений компании.

Построение концептуальной модели предполагает установление взаимосвязей между функциональными, организационными и информационными моделями. Концептуальная модель очень важна для исследовательских задач, в ней должно содержаться представление о природе взаимосвязей моделей.

Следует отметить, что предпроектный анализ требует большого вложения времени и труда в изучение как самой предметной области (детальное представление деятельности компании, анализ, унификация), так и языков моделирования. В настоящее время на российском рынке представлено достаточно большое количество инструментальных средств (ARIS, AllFusion Modeling Suite, Rational Rose, Adonis и другие), которые позволяют, так или иначе, создавать описания (модели) бизнес-процессов. Поэтому перед непосредственным описанием предметной области необходимо выбрать систему визуального и имитационного моделирования.

Система Adonis выделяется среди аналогов своей наглядностью, непротиворечивостью и большой функциональностью. Adonis позволяет связать весь объем анализируемой информации в единое целое, то есть создать концептуальную модель предприятия. Это выгодно отличает ее от методологии SADT и UP. Сравнительный анализ систем визуального моделирования в рамках данного учебного пособия не проводится. Однако выбор этой системы для описания предметной области ТПП является результатом отдельного исследования, обоснован и далеко не случаен.

## Задание

Результатом выполненной работы студента должно являться комплексное описание деятельности конструкторского бюро с помощью системы визуального и имитационного моделирования Adonis. Кроме того, необходимо провести анализ построенных моделей, на основе которого выявить проблемные места в деятельности компании.

Таким образом, в данной самостоятельной работе необходимо:

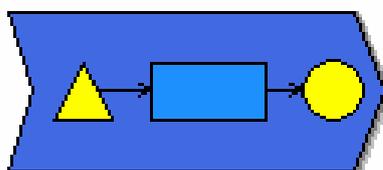
- Представить деятельность конструкторского бюро приборостроительного предприятия в виде функциональных, организационных и информационных моделей бизнес-процессов (т.е. построить модели бизнес-процессов «as is» – «как есть»). В качестве объекта исследования можно выбрать конструкторский отдел любого приборостроительного предприятия. При этом в качестве основных процессов отдела необходимо представить следующие:
  - Анализ заявки
  - Разработка ТЗ
  - Разработка КД
  - Проверка КД
  - Архивация
- Провести анализ построенных моделей, с помощью встроенных средств имитационного моделирования системы Adonis. Имитационное моделирование требуется провести по четырём алгоритмам (Path analysis, Capacity Analysis, Workload Analysis (steady state), Workload Analysis (fixed time period)). В качестве расчётного периода можно взять любой отрезок времени (от одного месяца). Для удобства в приложении 1 приведён список параметров, которые необходимо определить для выполнения имитационного моделирования.
- Выявить «узкие» места в управлении и проектировании.
- Разработать презентацию в Microsoft Office Power Point, в которой отразить:
  - Модели основных процессов
  - Результаты имитационного моделирования
  - Выявленные проблемы и «узкие» места в деятельности конструкторского отдела

## Работа в системе Adonis

Работа с системой Adonis начинается с создания новой модели: это может быть карта компании (Company map), модель бизнес-процессов (Business process model), модель документов (Document model), модель рабочей среды (Working environment model). По сути, это комплекс из функциональных, информационных и организационных моделей. Комплект функциональных моделей описывает все процедуры предприятия, информационных – потоки документов (состав и направления перемещения), организационных – структуру предприятия, отделов, субординацию служащих. Таким образом, Adonis позволяет всесторонне описать предмет исследования.

Моделирование можно начинать с создания диаграмм любого вида. Однако приоритетное положение занимают модели бизнес-процессов: в них задаются связи с документами и диаграммами рабочей среды. Получается следующая структура проекта:

- Верхним уровнем описания (первой моделью) является карта компании, она содержит наименование проекта и является ссылкой на модель бизнес-процессов следующего уровня детализации. На рис. 1 представлен пример карты компании.



Технологическая  
подготовка  
производства

*Рис. 1. Диаграмма верхнего уровня*

Структура представления атрибутов карты компании «записной книжки» элементов (notebook) Adonis представляет собой несколько закладок:

- Description – описание элемента;
- Input/output – входная и выходная документация (базовая и генерируемая документация проекта);
- Simulation result – результаты имитационного моделирования (все параметры доступны только для просмотра).
- Характеристиками процесса («process») карты компании являются:

- Name – название элемента;
  - Reference process – ссылка на присоединенную модель бизнес-процессов (данная ссылка позволяет перейти к детальной диаграмме процесса);
  - Description - описание;
  - Open questions – перечень вопросов, актуальных для данного процесса.
- Телом проекта служит набор функциональных моделей, описывающих выбранный бизнес-процесс. Функциональное моделирование очень похоже на составление блок-схемы программы: последовательно перечисляются необходимые действия, задаются условия переходов (параллельное выполнение, условия ЕСЛИ). Единственное отличие от привычной блок-схемы заключается в том, что все блоки обязательно должны иметь уникальные названия (действия отличаются датой, исполнителем или местом реализации).

Теоретически можно весь бизнес-процесс описать в одной диаграмме, но тогда она перестанет быть наглядной, поэтому создаётся несколько моделей, связанных между собой по принципу «программа-подпрограмма». На рис. 2 представлен пример функциональной диаграммы.

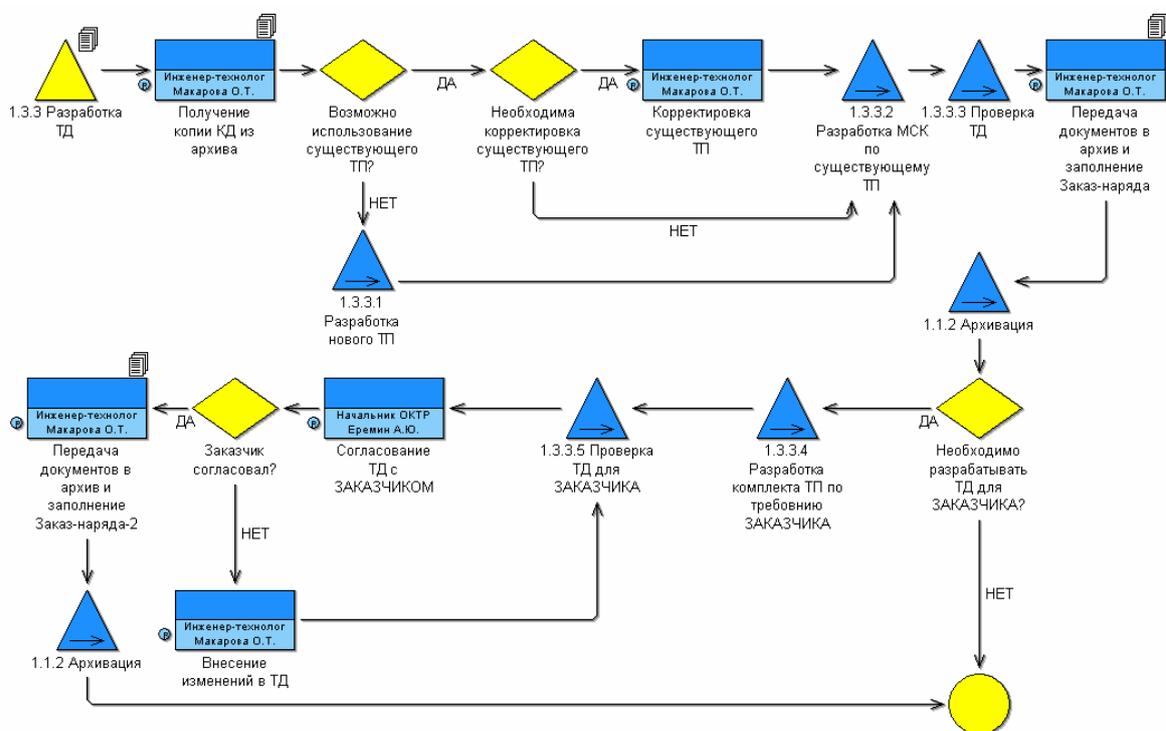
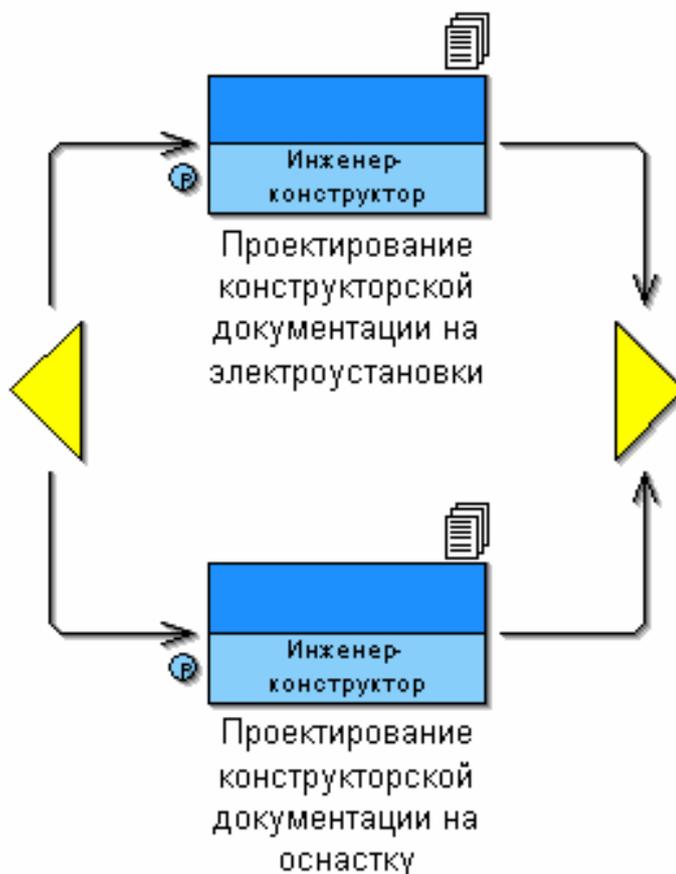


Рис. 2. Функциональная диаграмма «Разработка технологической документации»

На данной схеме представлено четыре условных перехода, которые обозначаются в виде прямоугольников.

Параллельные работы обозначаются на диаграмме с помощью элементов разветвления (Parallelism) и объединения (Merging). Пример работы с этими элементами представлен на рис.3.



*Рис. 3. Оформление параллельных действий в системе Adonis*

Атрибуты каждого элемента моделирования могут быть заданы в соответствии с требованиями проекта. Каждый элемент моделирования обладает «записной книжкой Adonis» (контекстное меню – Notebook). Она содержит одну или несколько глав, каждая из которых может содержать одну или более страниц. Признаки, описанные в записной книжке, могут обновляться (наименование ссылки на модель автоматически изменится при переименовании оригинала). Большинство полей сопровождаются описанием/подсказкой – значок «i» в верхнем правом углу над полем.

В большинстве случаев функциональные модели содержат условные переходы, которые для корректного выполнения имитационного моделирования требуется оформить в соответствии с правилами:

- в поле «Name» элемента «Variable» необходимо задать переменную, по которой будет производиться анализ направления перехода;
- в поле «Value» элемента «Random generator» задать вероятности перехода (например, при проверке технологического процесса это могут быть переходы «да/ТП утвержден» с вероятностью 0,8 % и «нет/ТП не утвержден» с вероятностью 0,2 %), оформив ее следующим образом Discrete (Да 0,8; Нет 0,2);
- в поле «Transition condition» соединительных элементов (исходящих соединителей для блока «Decision») необходимо задать значение переменной (например, a='Нет'), а в поле «Transition probability» вероятность данного перехода (например, 0,2);
- при задании переменной необходимо указать её тип – локальная или глобальная (элемент «Variable», закладка «Description», поле «Variable type»).

Следует обратить внимание на последнее правило оформления условного перехода. Если переменная задана как локальная, то её можно использовать только в пределах данного процесса, в то время как глобальную - во всех подпроцессах, аналогично с правилами объявления переменных в различных языках программирования (то же относится и к параллельным участкам процесса).

Вероятность перехода, указываемая в атрибутах соединительного элемента, означает частоту наступления последующего события (операции) за некоторый, задаваемый при анализе, промежуток времени. При выполнении расчётов в Adonis во входных данных требуется указать, проводится ли симуляция один раз или несколько (в течение месяца, года и пр.). Таким образом, в зависимости от заданных степеней вероятности при неоднократном выполнении процедуры некоторые пути реализации процесса будут выполняться чаще, чем другие.

- Модели рабочей среды и документов служат окружением диаграмм бизнес-процессов. Связи с ними задаются в Notebook какой-либо конкретной функции: для модели рабочей среды поле Responsible role (из каталога выбирается не только конкретная модель, но и роль исполнителя данной функции (технолог, начальник участка и пр.)), для модели документов поле Referenced document.

Модели рабочей среды содержат следующие элементы моделирования: исполнители (Performer), роли (Role) и подотделы (Organization unit). На рис. 4 изображен пример диаграммы рабочей среды.

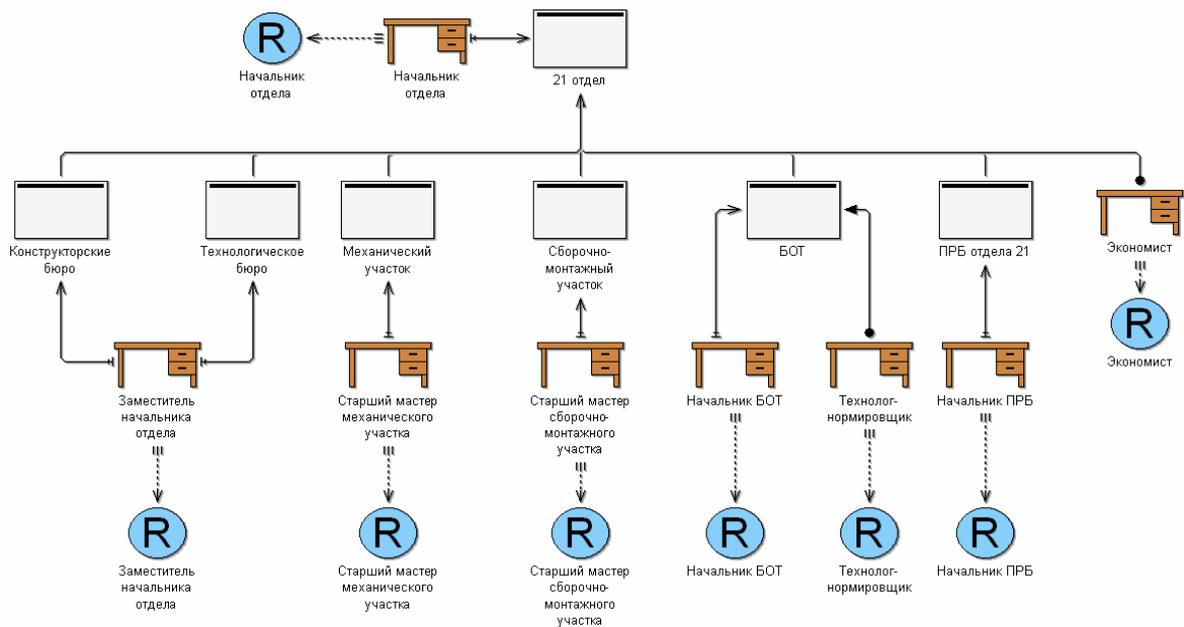


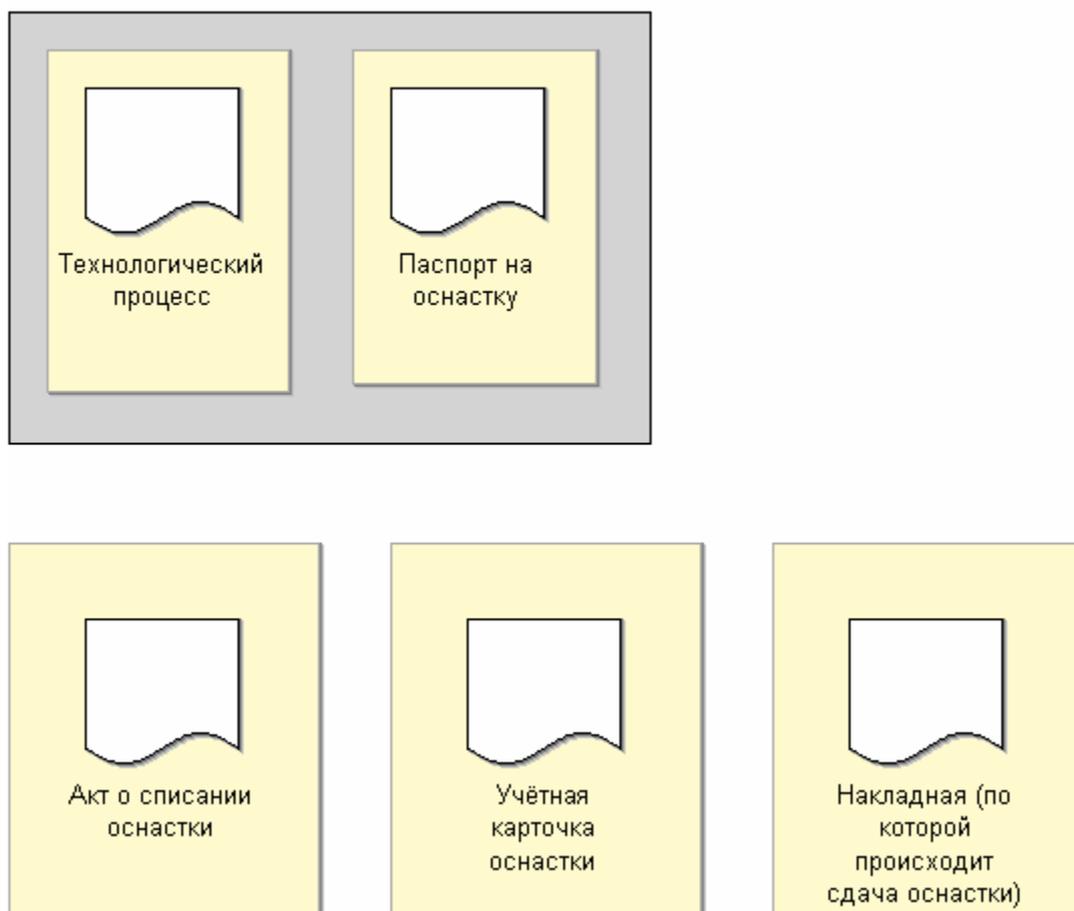
Рис. 4. Пример диаграммы рабочей среды

Модели документов включают документы и их группы. На рис. 5 представлен пример диаграммы документов.

После построения моделей бизнес-процессов можно выполнить их временную и стоимостную оценку. Для этого в Adonis имеется встроенный механизм имитационного моделирования, который позволяет:

- определить оптимальную последовательность протекания процесса на основе выходной информации, ранжированной по выбранному критерию;
- получить стоимостную и временную оценку как всего бизнес-процесса в целом, так и для каждой задачи и конкретного исполнителя;
- анимировано представить загрузку исполнителей на организационной модели (в виде перемещения заданий);
- выполнить статический и динамический расчёт рабочей загрузки процесса.

В результате можно провести ускоренный анализ модели компании, на основании которого ещё на ранних стадиях реинжиниринга деятельности предприятия можно выполнить оценку и модернизацию бизнес-процессов в соответствии с новыми требованиями.



*Рис. 5 Пример диаграммы документов*

Adonis предлагает четыре различных варианта имитационного моделирования:

- Path analysis – определение оптимального пути выполнения задания в моделях бизнес-процессов;
- Capacity Analysis – определение оптимального состава и структуры отделов за счёт оценки времени загрузки каждого отдельного исполнителя;
- Workload Analysis (steady state) – определение (динамическое) рабочей загрузки процесса (во время анализа процесс выполняется predetermined число раз независимо от расчетного периода);
- Workload Analysis (fixed time period) – определение (динамическое) рабочей загрузки процесса (процесс моделируется в течение predetermined периода времени не зависимо от того, сколько раз он будет выполнен).

Поскольку в Adonis представлено четыре алгоритма имитационного моделирования, очевидно, что набор входных параметров для них дол-

жен быть различен. Ниже приведена таблица 1, в которой сведены все параметры, оказывающие влияние на результаты того или иного вида имитационного моделирования. Т.е. знаком «+» отмечены параметры, участвующие в данном алгоритме моделирования, и, соответственно, знаком «—» – не участвующие.

*Таблица 1. Соотношение параметров и видов имитационного моделирования*

	Path analysis	Capacity Analysis	Workload Analysis (steady state)	Workload Analysis (fixed time period)
Execution time	+	+	+	+
Waiting time	+	+	+	+
Resting time	+	+	+	+
Transport time	+	+	+	+
Costs	+	+	+	+
EDP transaction costs	+	+	+	+
EDP batch costs	+	+	+	+
Print costs	+	+	+	+
Postal costs	+	+	+	+
Performer	—	+	+	+
Continuous execution	—	—	+	+
Execution interruptable	—	—	+	+
Task stack	—	—	+	+
Done by	—	+	+	+
Cooperative	—	+	+	+

	Path analysis	Capacity Analysis	Workload Analysis (steady state)	Workload Analysis (fixed time period)
Cooperation mode	—	+	+	+
Average number of participants	+	—	—	—
Min. quota of presence	—	—	+	+
Max. start period	—	—	+	+
Priority	—	+	+	+
Max. resource waiting time	—			
Quantity	—	+	—	—
Time period	—	+	—	—
Tolerance waiting time	—	—	+	+
Abandon after tolerance waiting time	—	—	+	+
Process calendar	—	—	+	+
Cost drive	+	+	+	+
Cost driver quantity	+	+	+	+
Hourly wages	—	+	+	+
Availability	—	+	—	—
Calendar	—	—	+	+
Time dependent cost	—	+	+	+

Имитационное моделирование с помощью Adonis позволяет провести ускоренный анализ модели компании, на основании которого еще на ранних этапах реинжиниринга деятельности предприятия можно выполнить оценку и модернизацию бизнес-процессов в соответствии с новыми требованиями.

Методология Adonis дает возможность комплексного подхода к анализу за счёт всестороннего описания (функционального, организационного и информационного) предметной области, реализованного в единой карте компании. Система позволяет проводить имитационное моделирование и оценку, как отдельных процессов предприятия, так и комплексной модели, включающей полное описание предметной области.

### ***Path analysis***

«Path analysis» позволяет определить оптимальную последовательность выполнения операций (рис. 6) по следующим критериям:

- «probability» (вероятность пути),
- «execution time» (время выполнения задания),
- «waiting time» (время ожидания задачи в очереди на исполнение),
- «transport time» (время транспортировки),
- «resting time» (время ожидания транспортировки),
- «cycle time» (общее время процесса),
- «cost» (стоимость процесса).

Соответственно, для выполнения данного вида анализа необходимо задать для элементов диаграммы, как все вышеперечисленные параметры, так и ряд других. В противном случае система выдаст сообщение – предупреждение об ошибке, либо в результирующих данных эти переменные будут иметь нулевые значения.

В отчёте «Path analysis» (помимо указания оптимального пути процесса) приводятся значения приведённых выше параметров (для всего процесса в целом – рис. 7).

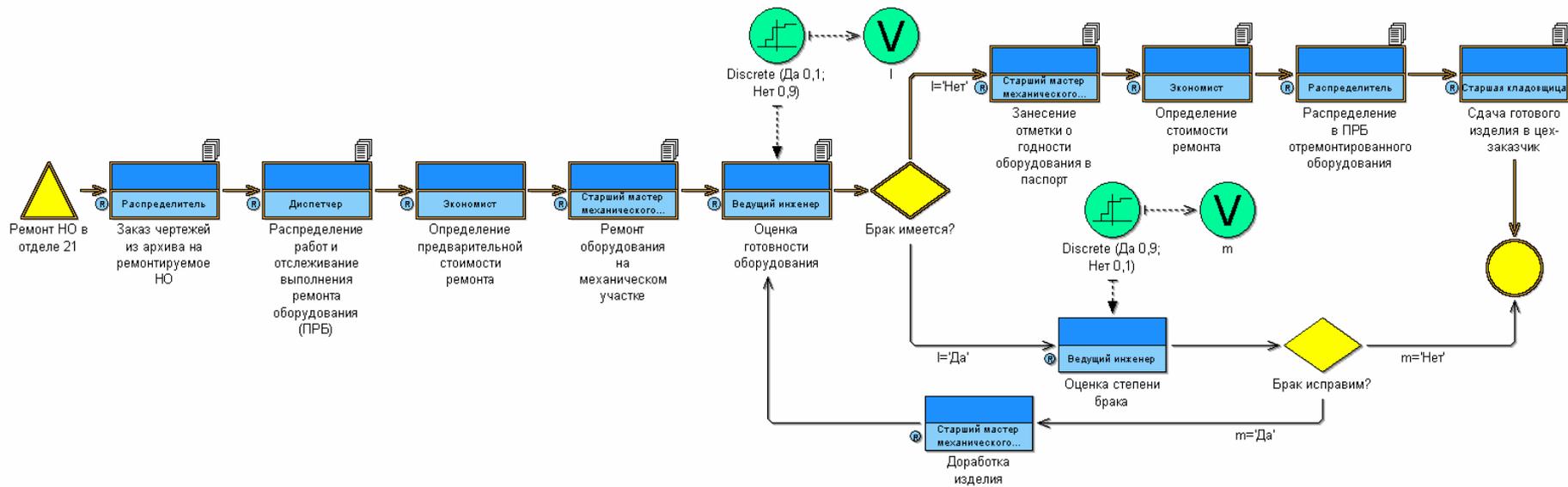


Рис. 6. Определение наиболее вероятного пути протекания процесса

Path analysis - Business process model: Ремонт НО в ...

	Expected value
Execution time	00:001:04:36:50
Waiting time	00:000:01:15:00
Resting time	00:000:00:19:48
Transport time	00:000:00:34:44
Cycle time	00:001:06:46:23
Costs	302,480000

Save... Print... Search... Close Help

Рис. 7. Системный отчет по анализу «Time and cost»

## *Capacity Analysis*

«Capacity Analysis» используется для управления персоналом, в том числе для определения оптимальной структуры рабочих групп, отделов. В результате моделирования возможно получить данные о затратах времени и финансов для каждого сотрудника.

Получаемые данные при «Capacity Analysis» оказываются более полными, чем при определении оптимального пути выполнения процесса («Path analysis»). Во-первых, все расчёты выполняются не только для процесса в целом, но и для элементов модели бизнес-процессов, и для конкретных исполнителей. Во-вторых, параметры времени и стоимости исполнения процесса указываются для каждой операции и каждого служащего отдельно. В-третьих, дополнительно производится расчёт затрат на выплату заработной платы сотрудникам и приводятся значения персональных ставок.

При расчете используются следующие данные (окно задания режимов моделирования): число запусков исполнения процесса, точность вычисления «steady state calculation» (с ее увеличением повышается точность расчетов и время симуляции), дата запуска процесса «simulation start». Возможно установить следующие режимы моделирования: «activity analysis» (дополнительный расчет рабочей загрузки), «computation» (если признак не активирован, то выполнение действий в модели рабочей среды визуализируется в течение анализа, но при этом система не генерирует таблицы с результатами расчетов), «protocol» (позволяет создать протокол моделирования в отдельном файле).

Содержание таблицы результатов можно настраивать. При указании «Process related (per year, month or process)» структура представления данных будет выглядеть следующим образом:

- модели бизнес-процессов;
- операции;
- исполнители.

При выборе «Person related (per year, month or process)» структура данных будет ориентироваться на исполнителей:

- исполнители;
- модели бизнес-процессов;
- операции, выполняемые данным исполнителем.

Третий режим «Working environment related (per year, month or process)» позволяет систематизировать информацию относительно элементов модели рабочей среды:

- отделы или роли;
- модели бизнес-процессов;

- операции, выполняемые данным исполнителем;
- исполнители.

Результаты имитационного моделирования могут быть представлены в табличной и графической форме.

### ***Workload Analysis***

Следующие два вида моделирования предназначены для определения рабочей загрузки процесса («Workload Analysis»). Он предназначен для динамического расчёта параметров времени и стоимости процесса. Признак «динамический процесс» означает, что имитационное моделирование будет производиться в течение некоторого промежутка времени (неоднократно). Так как «Workload Analysis» производится для анализа процесса, в результирующей таблице данных добавляется строка «total», содержащая сумму значений для каждого атрибута.

В первом случае «Workload Analysis (steady state)» число запусков исполнения процесса фиксируется пользователем. То есть если в поле «Number of simulation» (число запусков) стоит значение 1000, то во время моделирования указанный процесс будет инициирован 1000 раз. Каждый раз при имитации процесса значения атрибутов будут рассчитываться. Не обязательно, что эти значения будут совпадать для различных запусков этого процесса. На основании всех полученных данных (последовательность из 1000 исполнений) выводятся средние значения параметров. Далее пользователю предлагается выбрать структуру представления данных таблицы, после чего она генерируется.

Принципиальное отличие «Workload Analysis» заключается в том, что процесс в этом случае моделируется в течение некоторого промежутка времени. При каждом запуске процесса он будет выполняться поразному, то есть случайным образом будет инициироваться один из вариантов пути реализации процесса (см. «Path analysis»), а значит, будут отличаться и данные, получаемые при расчетах. Динамическое моделирование процесса позволяет приблизить экспертную оценку к реальным данным функционирования компании.

Используя «Workload Analysis» можно получить средние и суммарные данные о затратах времени и денежных средств на процесс, в том числе общие затраты на выполнение бизнес-процессов за плановый период, за месяц и за год. Например, система производит расчёт суммы затрат на оплату работы сотрудников для каждой операции в течение всего расчётного времени (за месяц, за год и др.), результат записывается в столбец «Personal cost (sum)» таблицы.

Для модуля «Workload Analysis» наглядно выполнена визуализация процесса перемещения заданий между исполнителями на модели работы

среды. Каждое задание отображается в виде бумажного документа, появляющегося на столе сотрудника. Количество этих документов (визуально высота стопки листов) меняется в течение моделирования, наглядно демонстрируется загрузка каждого исполнителя. Например, если в отделе имеется больше сотрудников, чем требуется для выполнения поставленной задачи, то при визуализации процесса «Workload Analysis» это будет наглядно отражено (малое количество документов на столах исполнителей). Численное подтверждение этому будет представлено в результирующей таблице, в колонке «Workload».

При «Workload Analysis (fixed time period)» зафиксированным оказывается период времени моделирования процесса. При указании входных параметров в данном случае необходимо выбрать дату начала и завершения процесса симуляции. При этом имитация процесса будет выполняться до тех пор, пока не завершится плановый промежуток времени. Система на основании собранных данных также произведёт расчёт средних значений параметров. Для одного и того же процесса значения атрибутов, указанные в таблице будут отличаться для «Workload Analysis (steady state)» и «Workload Analysis (fixed time period)». Это связано в первую очередь с тем, что моделирование для них будет выполнено разное число раз. Еще одной причиной является то, что «Workload Analysis (fixed time period)» привязывается к конкретным датам, то есть при расчётах учитывается количество рабочих, выходных и сокращённых дней за расчётный период.

# **Самостоятельная работа «Реинжиниринг бизнес-процессов конструкторского бюро приборостроительного предприятия»**

## **Реинжиниринг бизнес-процессов**

М. Хаммер и Дж. Чампи определяют реинжиниринг как «фундаментальное переосмысление и радикальное перепроектирование бизнес-процессов компании для достижения коренных улучшений в основных актуальных показателях их деятельности – стоимость, качество, услуги и темпы».

Необходимость реинжиниринга обосновывается высокой динамичностью современного делового мира. Непрерывные и довольно существенные изменения в технологиях, рынках сбыта и потребностях клиентов стали обычным явлением.

Решением проблемы является смена базовых принципов организации компаний и переход к ориентации не на функции, а на процессы. Предлагается использовать подход, когда менеджер компании принимает непосредственное участие в разработке модели компании. Суть этого подхода, названного интегрированным, состоит в том, что для разработки модели компании (существующей и будущей) предлагается использовать не CASE-средства, ориентированные на программиста, а интегрированные инструментальные средства, объединяющие на базе технологии динамических экспертных систем объектно-ориентированный подход, CASE-средства, средства имитационного моделирования и «активную» графику. Именно их использование дает возможность:

- разрабатывать модель компании непосредственно менеджером;
- создавать динамические модели, адекватно описывающие деятельность компании;
- автоматизировать основные бизнес-процессы – разработку образа будущей компании и спецификацию целей, разработку модели существующего бизнеса, разработку модели нового бизнеса, создание информационной системы поддержки.

Реинжиниринг предусматривает новый способ мышления – взгляд на построение компании как на инженерную деятельность.

Во всех случаях положительный результат получен не благодаря ориентации на узкую задачу, решаемую в predetermined организационных границах, а благодаря рассмотрению всего процесса в целом – ориентация на процесс. Средствами, позволяющими отказаться от уста-

ревших правил и создать модели новых процессов, скачкообразно улучшив основные показатели, явились информационные технологии.

Необходимо проводить реинжиниринг основных процессов компании. Проведение реинжиниринга как малых, так и больших процессов является трудной задачей. Кроме того, перепроектирование большого количества подчиненных процессов не приносит существенного результата, так как при этом неизбежно затрагиваются функции подразделений и исполнителей, что может привести к хаосу.

Реинжиниринг подразумевает, что выполнен исчерпывающий анализ существующего бизнеса. Его задача состоит в реконструировании существующих процессов при использовании новых технических достижений для лучшего обслуживания клиентов и достижения совершенно новых деловых целей компании.

Проект по реинжинирингу бизнеса обычно включает следующие четыре этапа:

- разработка образа будущей компании;
- анализ существующего бизнеса;
- разработка нового бизнеса;
- внедрение нового бизнеса.

Необходимо подчеркнуть, что перечисленные этапы выполняются не последовательно, а, по крайней мере, частично параллельно, причем некоторые этапы повторяются

Можно выделить четыре основных последствия проведения реинжиниринга:

- Переход от функциональных подразделений к командам процессов (сотрудники предприятий нового поколения в отличие от традиционно организованных фирм объединяются не по отделам, лабораториям, группам и т.п., а по рабочим командам, совместно выполняющим законченную часть работы – процесс).
- Работа исполнителя становится многоплановой и ориентированной на принятие самостоятельных решений (сотрудники компании уже как члены рабочих команд несут ответственность за процесс в целом, кроме этого устраняются излишние проверки и согласования, что делает их работу более содержательной).
- Изменяется оценка эффективности работы и модель оплаты труда (новая компания ориентируется на получение продукта, поэтому поощрение сотрудников производится по конечному результату и его качеству, поощряется личная инициатива).

- Изменяется организационная структура компании: уменьшается количество уровней управления (изменяются функции менеджеров от контролирующих к «тренерским»), повышается личная ответственность сотрудников за результаты коллективной работы над процессами компании).

После проведения реинжиниринга новая компания ориентируется на процессы. В результате появляется необходимость управления ими. При этом исполнители делятся не по отделам, а по группам, ориентированным на выполнение конкретной задачи. Инженеры на местах должны чётко представлять суть задания и свою роль в его исполнении, а руководители иметь возможность контролировать и по необходимости изменять процесс исполнения. Для этого в обновленной компании сотрудники подразделяются не по отделам, а по небольшим рабочим коллективам, выполняющим совместно некоторый законченный процесс. Выявление таких групп, а также всех изменений в структуре фирмы является целью организационного анализа.

Вторым этапом проектирования образа новой компании является анализ полученной модели области исследования. Этот процесс начинается с создания различных сценариев. Для каждого из них формируется общее описание процесса, включающее исполнителей, заказчиков и сам бизнес-процесс. Далее проводится имитационное моделирование различных вариантов процессов. На основе всех собранных данных выбирается обновленная модель ТПП. Критерием оценки служит оптимальное сочетание основных показателей деятельности компании (времени процессов, затрат, загрузки исполнителей и пр.).

## Задание

В данной самостоятельной работе необходимо определить способы решения проблем, выявленных в моделях бизнес-процессов самостоятельной работы «Анализ бизнес-процессов конструкторского бюро приборостроительного предприятия». Т.е. необходимо построить модели процессов «as to be». Кроме того, требуется выполнить сравнительный анализ результатов имитационного моделирования моделей «as is» и «as to be», в котором обосновывается выбор нового сценария протекания бизнес-процессов.

Таким образом, в самостоятельной работе необходимо:

- Разработать различные сценарии деятельности конструкторского бюро приборостроительного предприятия.
- Выполнить имитационное моделирование различных вариантов процессов по четырём алгоритмам. В качестве расчётного периода взять тот же отрезок времени, что и в самостоятельной работе «Анализ бизнес-процессов конструкторского бюро приборостроительного предприятия».
- На основе всех собранных данных выбрать новый образ компании и обосновать принятое решение (критерием оценки служит оптимальное сочетание основных показателей деятельности компании (времени процессов, затрат, загрузки исполнителей и пр.)).
- Разработать презентацию в Microsoft Office Power Point, в которой отразить:
  - Модели «as is».
  - Выявленные проблемы и «узкие» места в деятельности конструкторского отдела и способы их решения.
  - Модели «as to be».
  - Сравнительный анализ результатов имитационного моделирования (по моделям «as is» и «as to be»)

## Работа в системе Adonis

Перед реорганизацией бизнес-процессов необходимо провести их тестирование, т.е. оценку и сравнительный анализ существующего бизнеса и будущего. Такое тестирование позволяет проводить система визуального и имитационного моделирования Adonis, а именно, встроенный в неё блок Evaluation. Этот блок предназначен для оценки результатов имитационного моделирования. В отличие от анализа (Analysis) результатом любого вида расчётов в Adonis является числовой массив значений различных параметров моделей. Входными данными для получения аналитической оценки служат результаты расчётов, проводимых в рамках моделирования (Simulation) Adonis. Полученные значения параметров, такие как время выполнения задания, время транспортировки, стоимость процесса, загрузка исполнителей и другие, должны быть сохранены заранее. При этом для «Comparative representation of result» (вид расчётов; сравнительная оценка результатов анализа нескольких процессов) результаты расчётов должны быть сохранены в отдельных файлах с расширением asc (comparing representation), а для «Evaluation queries» (вид расчётов; оценка процессов по стандартным запросам) – в атрибутах модели, на закладке «simulation data» записной книжки элементов.

Adonis предоставляет широкий выбор средств для проведения аналитической оценки полученных расчётных данных. Возможно проводить сравнительный анализ многих моделей, сопоставлять значения атрибутов. На основании получаемой информации становится возможным делать выводы о приемлемости тех или иных изменений процессов предприятия, выявлять проблемные участки процессов, характеризующиеся значительными потерями времени и/или денежных средств. Возможности Adonis – это средство для проведения реинжиниринга процессов ещё на этапе анализа деятельности компании, данный механизм позволяет быстро получать необходимые результаты.

При выполнении расчётов система генерирует таблицу значений параметров, используемых при анализе. Эти данные помимо табличной формы могут быть представлены в виде диаграммы или графика. Для создания диаграмм в Adonis интерфейс таблиц (сгенерированных в результате того или иного вида анализа) содержит клавишу «diagram». При её активации система запросит указать объекты, атрибуты и основные параметры визуализации (окно «diagram setting»), после чего выведет в отдельном окне сгенерированную диаграмму. В Adonis существует возможность сохранения диаграмм и графиков в виде графического объекта (файлы с расширениями bmp, psx, jpg, png).

При создании диаграмм существует возможность указывать лишь часть данных таблицы, что даёт возможность получать различные рапорты на основании одних и тех же данных.

В Adonis существует три вида аналитических расчётов:

- Flowmark Audit Trail Evaluation – позволяет рассчитать время исполнения и ожидания, время цикла для отдельных операций или процессов в диаграммах управления потоками производственных заданий «Flowmark» (например, модуль WorkFlow в PDM-системе SmarTeam);
- Comparative representation of result – выполняет сравнительную оценку результатов анализа нескольких процессов, полученные данные представляются в форме таблицы, диаграммы или графика;
- Evaluation queries – оценка процессов по стандартным запросам, хранимым в библиотеке системы, прикладная библиотека автоматически устанавливается вместе с системой.

### ***Flowmark Audit Trail Evaluation***

Для выполнения первого вида расчётов необходимо иметь сохранённые графики управления потоками заданий в виде файлов с расширением del (созданные в других системах). Для данного вида расчётов существует два режима работы. В первом случае «Process related» производится расчёт времени выполнения, ожидания и цикла для процесса в целом, а также частоты исполнения операции тем или иным исполнителем. Во втором режиме «Person related» выполняется расчёт только времени исполнения и ожидания, но для каждой операции. Кроме этого указывается конкретные исполнители для всех заданий.

«Flowmark Audit Trail Evaluation» позволяет выполнять анализ и сравнение процессов, созданных в системах типа «Flowmark». Данный модуль расширяет возможности использования Adonis, а также облегчает процесс внедрения данной системы в работу аналитических отделов компаний.

### ***Comparative representation of result***

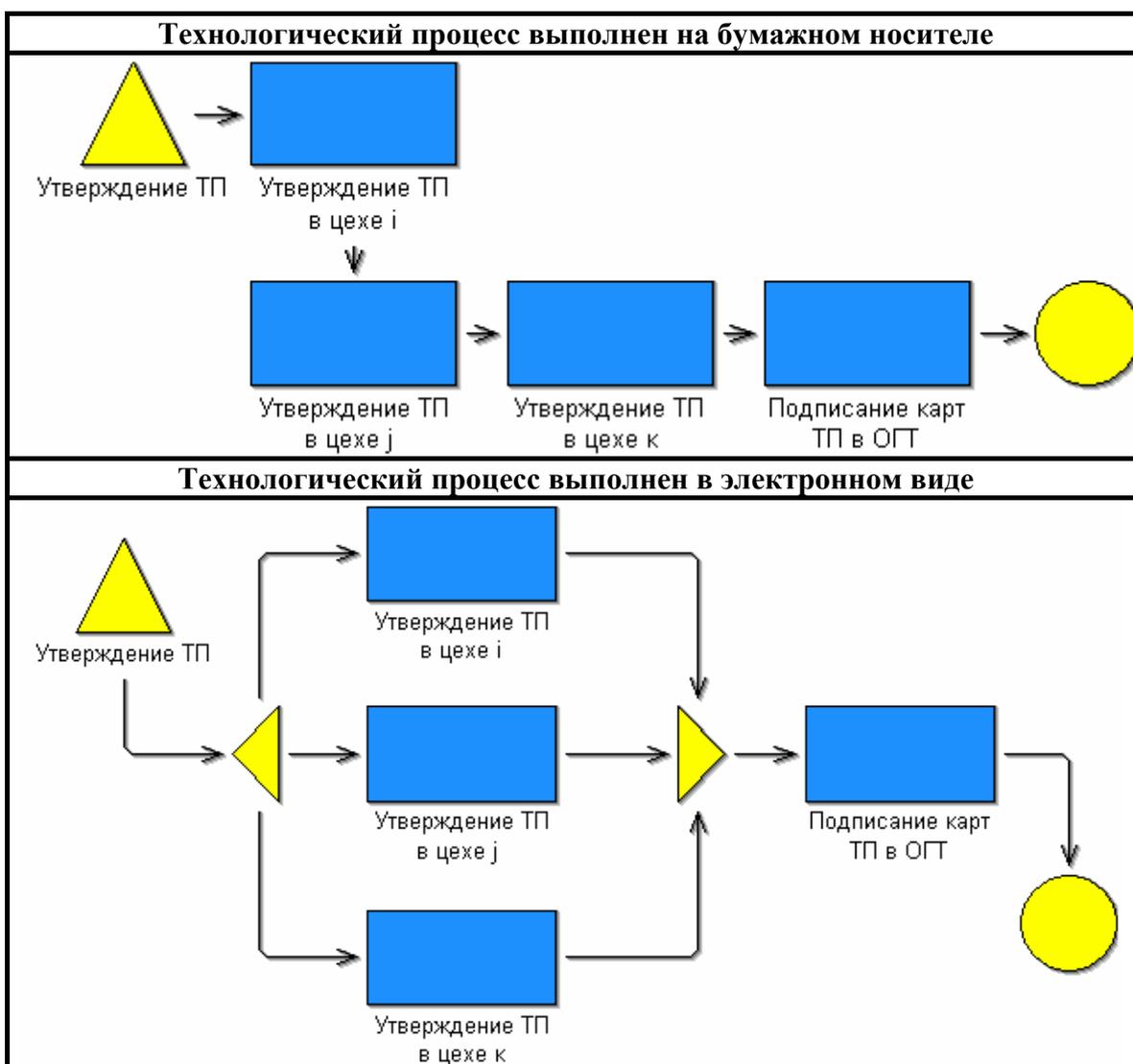
Второй вид расчётов «Comparative representation of result» представляет собой средство для сравнения результатов оценки нескольких процессов. Для работы с данным модулем необходимо предварительно сохранить те расчётные данные, для которых требуется провести аналитическую оценку. Таким образом, исходной информацией для анализа в этом случае являются таблицы параметров, полученные в процессе ими-

тационного моделирования (simulation). Как говорилось ранее, расчётные данные должны быть сохранены в виде файлов с расширением asr.

В результате данной оценки пользователь получит аудиторский отчёт о затратах времени и денежных средств на выполнение бизнес-процессов компании. Этот отчёт может быть представлен в виде таблицы, диаграммы или графика. Например, процесс ТПП «Утверждение технологического процесса изготовления детали» может быть представлен в двух вариантах, см. табл. 2:

- процесс проверки технологической документации, выполненной на бумажном носителе;
- процесс проверки технологической документации, выполненной в электронном виде.

Табл. 2. Исполнения процесса «Утверждение ТП»

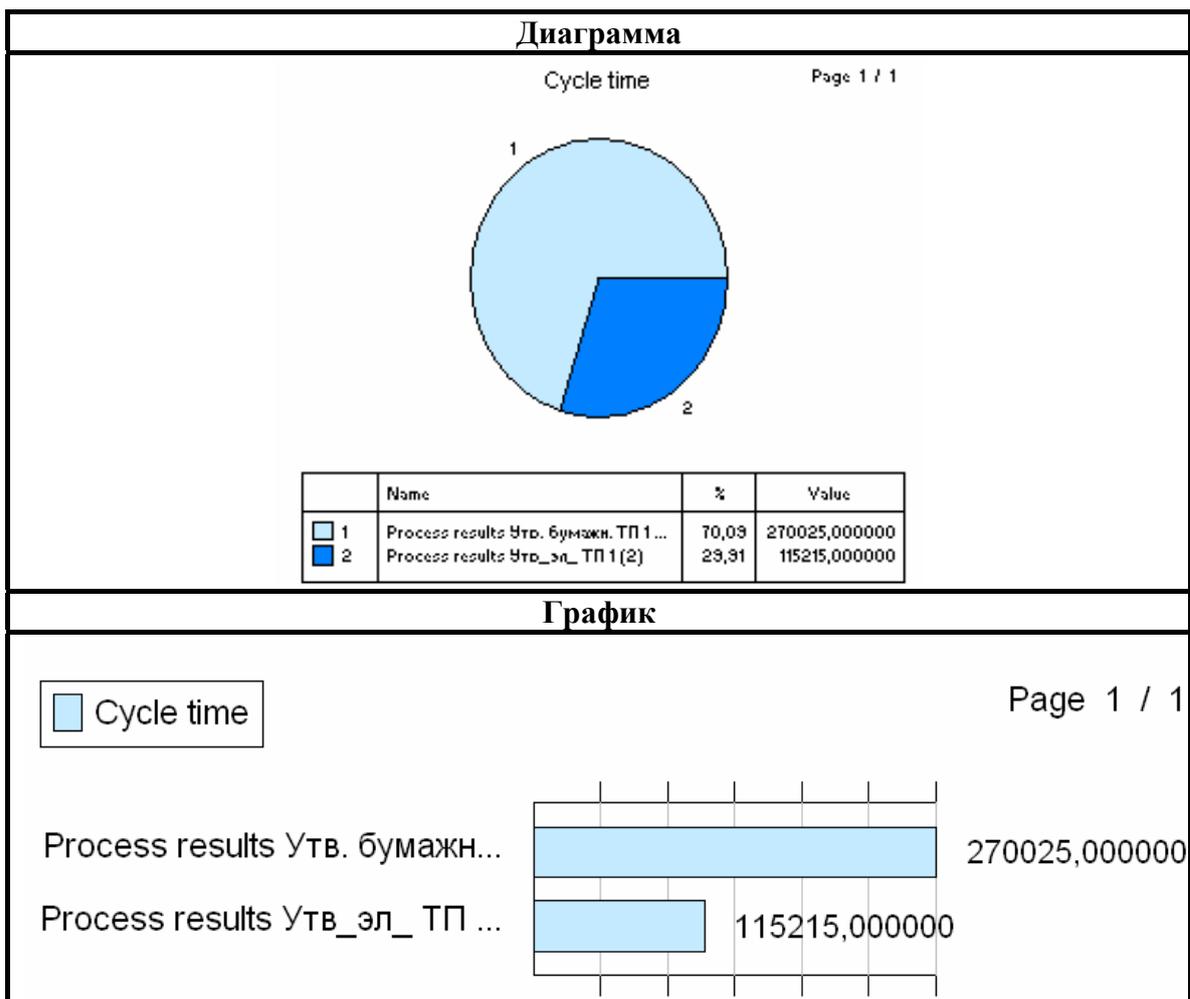


Как видно из графиков, представленных в таблице 3, при использовании автоматизированных систем проектирования при разработке ТП

становится возможным одновременно/параллельно выполнять некоторые операции ТПП.

Для этих двух процессов в Adonis можно провести оценку времени и стоимости их выполнения. Наиболее показательный результат будет для расчёта общего времени выполнения процесса «Cycle time». В следующей таблице представлены результаты сравнения времени цикла этих процессов, сгенерированные в Adonis.

Табл. 3. Исполнения процесса «Утверждение ТП»



Данные были получены в результате сравнения расчётов времени и стоимости («Time and cost», модель бизнес-процессов) для указанных выше процессов.

Для проведения сравнительного анализа необходимо:

- зайти в режим Evaluation (на горизонтальной панели инструментов

нажать на кнопку Evaluation ) и выбрать вид оценки Compari-

son of results (на горизонтальной панели инструментов нажать на



кнопку Comparison of results );

- с помощью панели Comparison of results – Result selection выбрать файлы, для которых требуется провести сравнительный анализ (на панели нажать на кнопку Add), и указать параметры, по которым будет проводиться сравнительный анализ.

## ***Evaluation queries***

«Evaluation queries» третий вид расчётов, позволяет провести оценку моделей по стандартным запросам, хранимым в библиотеке системы. Прикладная библиотека автоматически устанавливается вместе с системой и может быть дополнена администратором.

Получение данных в этом модуле возможно лишь в том случае, если результаты имитационного моделирования сохранены в атрибутах диаграммы, на закладке «simulation result». Для этого необходимо воспользоваться клавишей «Evaluation». При проведении любого вида «Simulation» перед формированием таблицы результатов система выводит на экран окно параметров, в котором указываются режимы расчётов (например, данные таблицы будут соответствовать однократному выполнению процесса («per process»), выполнению в течение месяца или года). Клавиша «Evaluation» располагается в этом окне, наряду с «Model info», «Help» и др. При её активации (указать «Evaluation» и в открывшемся окне выбрать «Including activity result») система заполнит поля «simulation result» элементов диаграммы, после этого предложит выполнить дальнейшие расчёты.

*Внимание!* Клавиша «Evaluation» будет активна только в том случае, если в текущей сессии работы программы открыты все модели, участвующие в расчётах.

«Evaluation queries» содержит несколько режимов расчётов, их описание приведено в таблице 4.

Таблица 4. Evaluation queries - оценка процессов по стандартным запросам

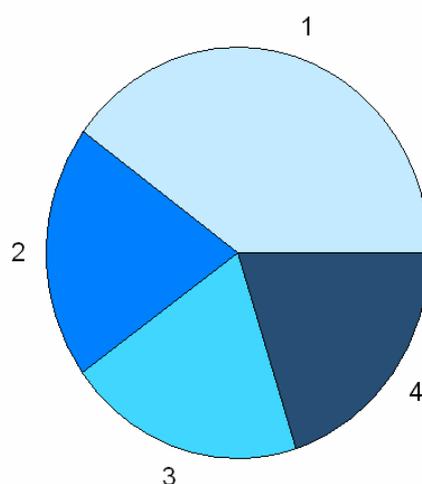
Evaluation queries		
<i>Вид анализа</i>		
<i>Режим расчета</i>	<i>Запросы</i>	<i>Описание</i>
Evaluation queries → <b>Evaluation queries on Business process model</b>		
Process-oriented queries on Business process model		Процессно-ориентированные запросы
	All business process with an execution time greater then ...	Найти все процессы, для которых время исполнения больше ...
	All business process with a cycle time greater then ...	Найти все процессы, для которых время цикла (общее время исполнения процесса) больше ...
	All business process with cost greater then ...	Найти все процессы, стоимость которых больше ...
Activity-oriented queries on Business process model		Объектно-ориентированные запросы
	All activity called more often then ...	Найти все операции, исполняемые в течение расчётного времени больше чем ... раз(а)
	All activity with an aggregated resting time greater then ...	Найти все операции, для которых время ожидания транспортировки (например, письмо ожидает времени отправки) больше ...

	All activity with an aggregated execution time greater than ...	Найти все операции, для которых время исполнения больше ...
	All activity with an aggregated waiting time greater than ...	Найти все операции, для которых время ожидания начала выполнения задания больше ...
	All activity with aggregated personnel cost greater than ...	Найти все операции, для которых заработная плата персонала больше ...
	All activity with aggregated cost greater than ...	Найти все операции, для которых затраты на выполнение задания больше ...
<b>Evaluation queries → Evaluation queries on Working environment model</b>		
Queries on Working environment model		Запросы по моделям рабочей среды
	All performers with a capacity greater than ...	Найти всех исполнителей, для которых коэффициент успеваемости (отношение фактического времени выполнения операций к запланированному) больше ...
	All performers with a workload greater than ...	Найти всех исполнителей, для которых коэффициент загрузки (отношение времени, затраченного на выполнение операций, к общему рабочему времени) больше ...
	All performers with personnel cost greater than ...	Найти всех исполнителей, у которых заработная плата больше ...
	All performers with the role ... and who have a capacity greater than ...	Найти всех исполнителей профессии (роли) ..., для которых коэффициент успеваемости больше ...

	All performers belonging to the organizational unit ... and who have a lower workload then ...	Найти всех сотрудников отдела ..., для которых коэффициент загрузки меньше ...
--	--	--

Для расчётов «Evaluation queries» полученные данные также могут быть представлены в виде таблиц, диаграмм и графиков. Например, для модели «Утверждение ТП, представленного в электронном виде» оценка по запросу «All activity with an aggregated execution time greater then ...» может выглядеть, как указано на рис. 8.

Aggregated execution time



	Name	%	Value
1	Подписание карт ТП в ОГТ	40,00	00:000:02:46:40
2	Утверждение ТП в цехе i	20,00	00:000:01:23:20
3	Утверждение ТП в цехе j	20,00	00:000:01:23:20
4	Утверждение ТП в цехе к	20,00	00:000:01:23:20

Рис. 8. Evaluation queries - диаграмма распределения времени выполнения процесса по операциям

# **Самостоятельная работа «Моделирование ТПП с использованием диаграмм последовательности и кооперации UML»**

## **Цель работы**

Цель данной практической работы состоит в том, чтобы на примере создания диаграмм последовательности и кооперации на языке UML:

- ознакомиться с особенностями представления бизнес-процессов ТПП в нотации UML;
- освоить динамический функциональный анализ системы в нотации UML;
- провести сравнительный анализ возможностей статического и динамического функционального анализа, реализованных в UML.

В результате выполнения самостоятельной работы студенты приобретут навыки по работе с системой, проведут анализ функциональных возможностей системы для описания предметной области ТПП.

## **Особенности построения диаграмм последовательности в нотации UML**

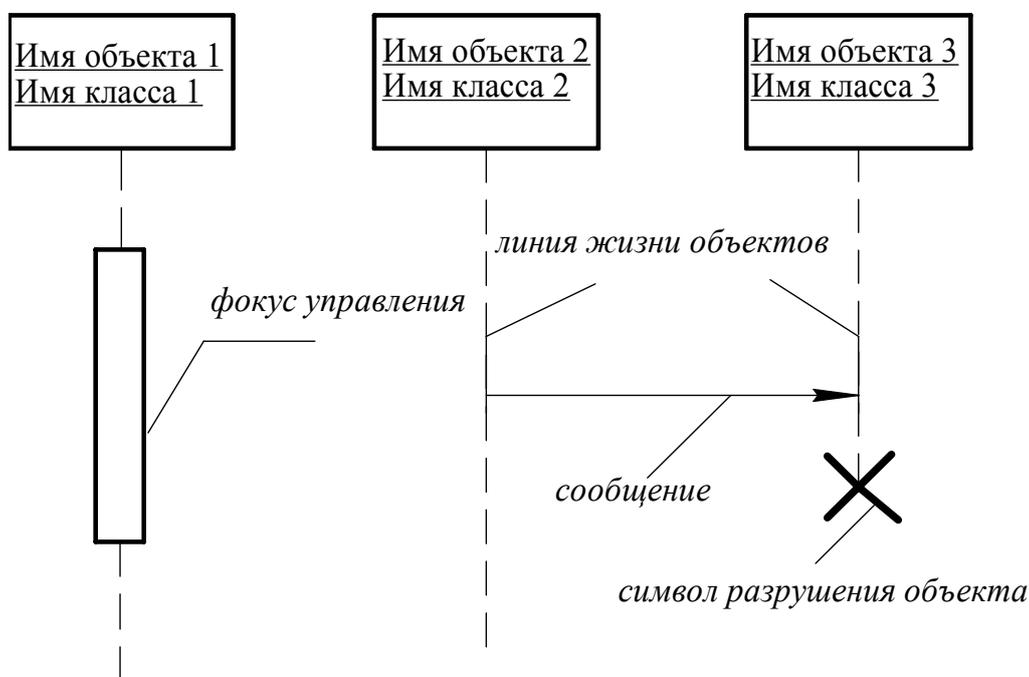
### ***Общие положения***

Одной из характерных особенностей систем различной природы и назначения является взаимодействие между собой отдельных элементов, из которых образованы эти системы. В языке UML взаимодействие элементов рассматривается в информационном аспекте их коммуникации, т. е. взаимодействующие объекты обмениваются между собой некоторой информацией. При этом информация принимает форму законченных сообщений. Другими словами, хотя сообщение и имеет информационное содержание, оно приобретает дополнительное свойство оказывать направленное влияние на своего получателя.

Для моделирования взаимодействия объектов в языке UML используются соответствующие диаграммы взаимодействия. Говоря об этих диаграммах, имеют в виду два аспекта взаимодействия. Во-первых, взаимодействия объектов можно рассматривать во времени, и тогда для представления временных особенностей передачи и приема сообщений между объектами используется диаграмма последовательности.

## ***Объекты на диаграмме последовательности***

На диаграмме последовательности изображаются исключительно те объекты, которые непосредственно участвуют во взаимодействии и не показываются возможные статические ассоциации с другими объектами. Для диаграммы последовательности ключевым моментом является именно динамика взаимодействия объектов во времени. При этом диаграмма последовательности имеет как бы два измерения. Одно - слева направо в виде вертикальных линий, каждая из которых изображает линию жизни отдельного объекта, участвующего во взаимодействии. Графически каждый объект изображается прямоугольником и располагается в верхней части своей линии жизни (рис. 9). Внутри прямоугольника записываются имя объекта и имя класса, разделенные двоеточием.



*Рис. 9. Графическое изображение примитивов диаграммы последовательности*

Крайним слева на диаграмме изображается объект, который является инициатором взаимодействия. Правее изображается другой объект, который непосредственно взаимодействует с первым. Таким образом, все объекты на диаграмме последовательности образуют некоторый порядок, определяемый степенью активности этих объектов при взаимодействии друг с другом.

Второе измерение диаграммы последовательности - вертикальная временная ось, направленная сверху вниз. Начальному моменту времени

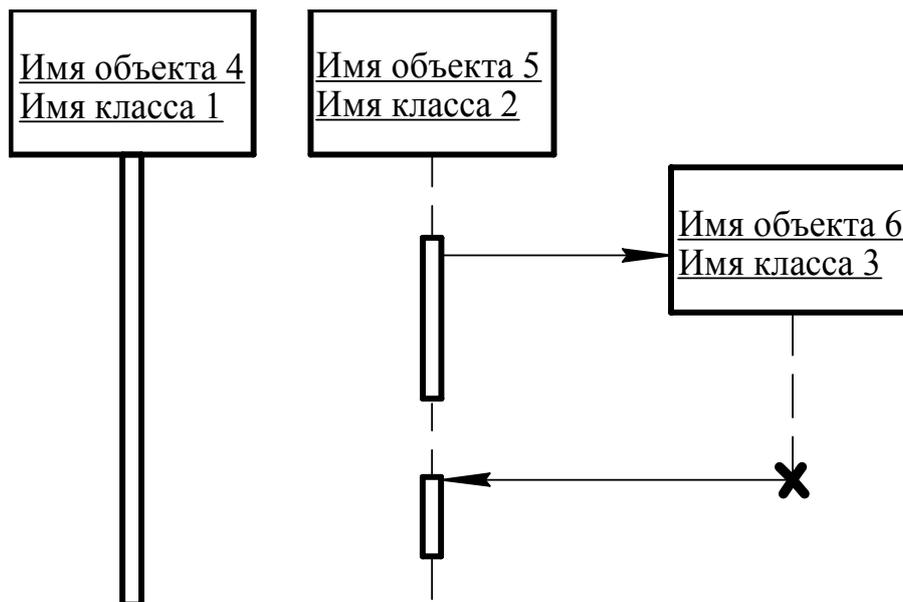
соответствует самая верхняя часть диаграммы. При этом взаимодействия объектов реализуются посредством сообщений, которые посылаются одними объектами другим. Сообщения изображаются в виде горизонтальных стрелок с именем сообщения и также образуют порядок по времени своего возникновения. Другими словами, сообщения, расположенные на диаграмме последовательности выше, иницируются раньше тех, которые расположены ниже. При этом масштаб на оси времени не указывается, поскольку диаграмма последовательности моделирует лишь временную упорядоченность взаимодействий типа "раньше-позже".

### *Линии жизни на диаграмме последовательности*

Линия жизни объекта (object lifeline) изображается пунктирной вертикальной линией, ассоциированной с единственным объектом на диаграмме последовательности. Линия жизни служит для обозначения периода времени, в течение которого объект существует в системе и, следовательно, может потенциально участвовать во всех ее взаимодействиях. Если объект существует в системе постоянно, то и его линия жизни должна продолжаться по всей плоскости диаграммы последовательности от самой верхней ее части до самой нижней.

Отдельные объекты, выполнив свою роль в системе, могут быть уничтожены (разрушены), чтобы освободить занимаемые ими ресурсы. Для таких объектов линия жизни обрывается в момент его уничтожения. Для обозначения момента уничтожения объекта в языке UML используется специальный символ в форме латинской буквы "X" (рис. 10). Ниже этого символа пунктирная линия не изображается, поскольку соответствующего объекта в системе уже нет, и этот объект должен быть исключен из всех последующих взаимодействий.

Вовсе не обязательно создавать все объекты в начальный момент времени. Отдельные объекты в системе могут создаваться по мере необходимости, существенно экономя ресурсы системы и повышая ее производительность. В этом случае прямоугольник такого объекта изображается не в верхней части диаграммы последовательности, а в той ее части, которая соответствует моменту создания объекта (рис. 10). При этом прямоугольник объекта вертикально располагается в том месте диаграммы, которое по оси времени совпадает с моментом его возникновения в системе. Очевидно, объект обязательно создается со своей линией жизни и, возможно, с фокусом управления.



*Рис. 10. Графическое изображение различных вариантов линий жизни и фокусов управления*

### ***Фокус управления диаграммы деятельности***

В процессе функционирования объектно-ориентированных систем одни объекты могут находиться в активном состоянии, непосредственно выполняя определенные действия или в состоянии пассивного ожидания сообщений от других объектов. Чтобы явно выделить подобную активность объектов, в языке UML применяется специальное понятие, получившее название фокуса управления (focus of control). Фокус управления изображается в форме вытянутого узкого прямоугольника (см. объект 1 на рис. 9), верхняя сторона которого обозначает начало получения фокуса управления объектом (начало активности), а ее нижняя сторона - окончание фокуса управления (окончание активности). Этот прямоугольник располагается ниже обозначения соответствующего объекта и может заменять его линию жизни (объект 4 на рис. 10), если на всем ее протяжении он является активным.

С другой стороны, периоды активности объекта могут чередоваться с периодами его пассивности или ожидания. В этом случае у такого объекта имеются несколько фокусов управления (объект 5 на рис. 10). Важно понимать, что получить фокус управления может только существующий объект, у которого в этот момент имеется линия жизни. Если же некоторый объект был уничтожен, то вновь возникнуть в системе он уже не может. Вместо него лишь может быть создан другой экземпляр этого же класса, который, строго говоря, будет являться другим объектом.

В отдельных случаях инициатором взаимодействия в системе может быть актер или внешний пользователь. В этом случае актер изобра-

жается на диаграмме последовательности самым первым объектом слева со своим фокусом управления (рис. 11). Чаще всего актер и его фокус управления будут существовать в системе постоянно, отмечая характерную для пользователя активность в инициировании взаимодействий с системой. При этом сам актер может иметь собственное имя либо оставаться анонимным.

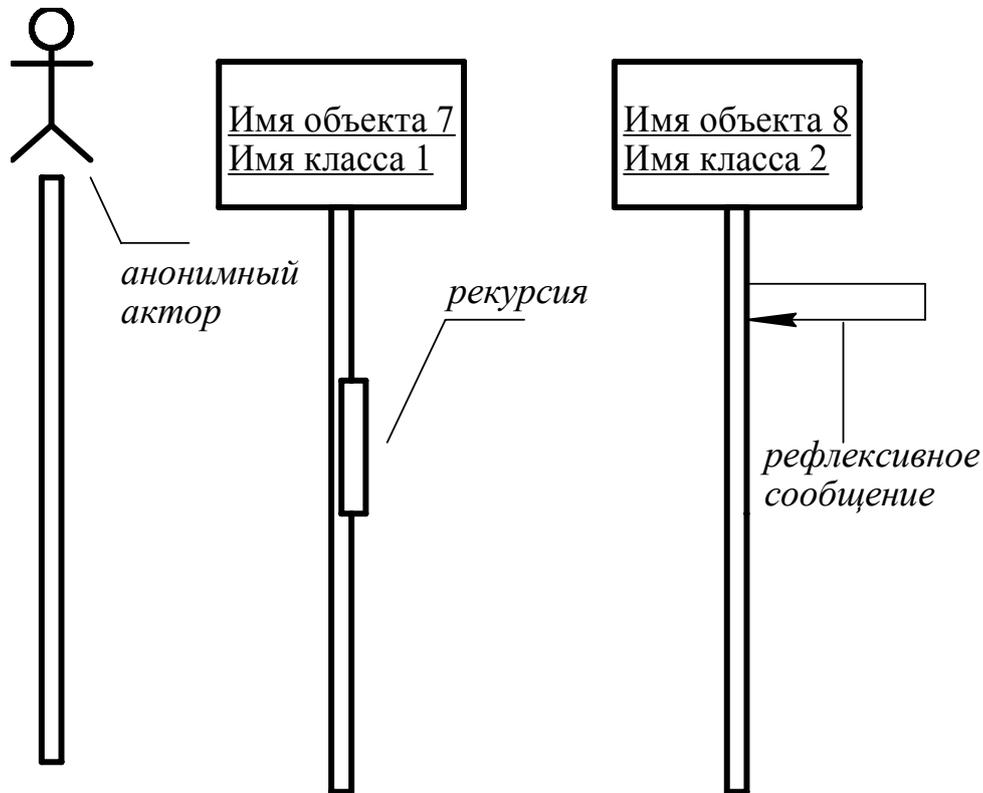


Рис. 11. Графическое изображение актера, рекурсии и рефлексивного сообщения на диаграмме последовательности

### **Сообщения на диаграммах деятельности**

Как было отмечено выше, цель взаимодействия в контексте языка UML заключается в том, чтобы специфицировать коммуникацию между множеством взаимодействующих объектов. Каждое взаимодействие описывается совокупностью сообщений, которыми участвующие в нем объекты обмениваются между собой. В этом смысле сообщение (message) представляет собой законченный фрагмент информации, который отправляется одним объектом другому. При этом прием сообщения инициирует выполнение определенных действий, направленных на решение отдельной задачи тем объектом, которому это сообщение отправлено.

Таким образом, сообщения не только передают некоторую информацию, но и требуют или предполагают от принимающего объекта вы-

полнения ожидаемых действий. Сообщения могут инициировать выполнение операций объектом соответствующего класса, а параметры этих операций передаются вместе с сообщением. На диаграмме последовательности все сообщения упорядочены по времени своего возникновения в моделируемой системе.

В таком контексте каждое сообщение имеет направление от объекта, который инициирует и отправляет сообщение, к объекту, который его получает. Иногда отправителя сообщения называют клиентом, а получателя - сервером. При этом сообщение от клиента имеет форму запроса некоторого сервиса, а реакция сервера на запрос после получения сообщения может быть связана с выполнением определенных действий или передачи клиенту необходимой информации тоже в форме сообщения.

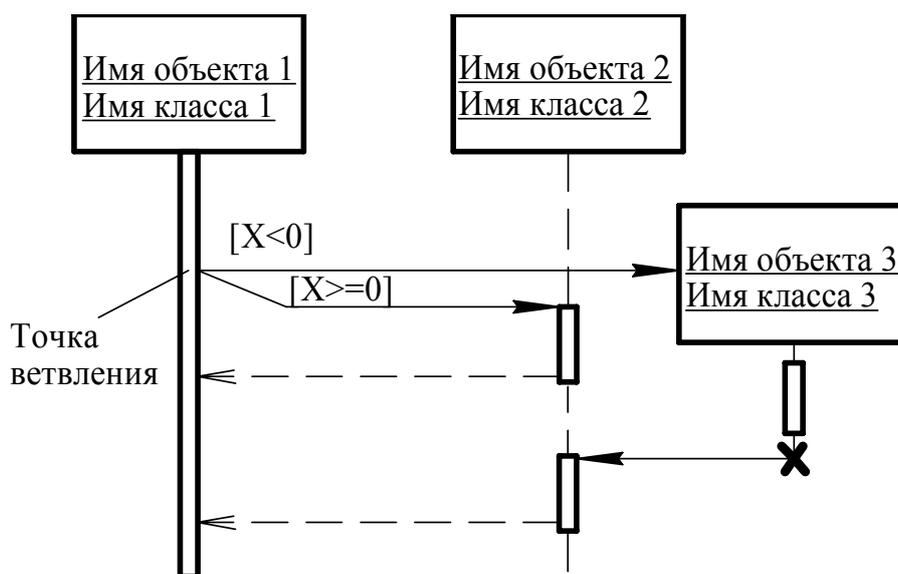
Обычно сообщения изображаются горизонтальными стрелками, соединяющими линии жизни или фокусы управления двух объектов на диаграмме последовательности. При этом неявно предполагается, что время передачи сообщения достаточно мало по сравнению с процессами выполнения действий объектами. Считается также, что за время передачи сообщения с соответствующими объектами не может произойти никаких событий. Другими словами, состояния объектов остаются без изменения. Если же это предположение не может быть признано справедливым, то стрелка сообщения изображается под некоторым наклоном, так чтобы конец стрелки располагался ниже ее начала.

В отдельных случаях объект может посылать сообщения самому себе, инициируя так называемые рефлексивные сообщения. Такие сообщения изображаются прямоугольником со стрелкой, начало и конец которой совпадают. Подобные ситуации возникают, например, при обработке нажатий на клавиши клавиатуры при вводе текста в редактируемый документ, при наборе цифр номера телефона абонента.

Таким образом, в языке UML каждое сообщение ассоциируется с некоторым действием, которое должно быть выполнено принявшим его объектом. При этом действие может иметь некоторые аргументы или параметры, в зависимости от конкретных значений которых может быть получен различный результат. Соответствующие параметры будет иметь и вызывающее это действие сообщение. Более того, значения параметров отдельных сообщений могут содержать условные выражения, образуя ветвление или альтернативные пути основного потока управления.

## ***Ветвление потока управления на диаграммах деятельности***

Для изображения ветвления рисуются две или более стрелки, выходящие из одной точки фокуса управления объекта (фокус управления объекта 1 на рис. 12). При этом соответствующие условия должны быть явно указаны рядом с каждой из стрелок в форме сторожевого условия. Как нетрудно представить, если условие записано в форме булевского выражения, то ветвление будет содержать только две ветви. В любом случае условия должны взаимно исключать одновременную передачу альтернативных сообщений.



*Рис. 12. Графическое изображение бинарного ветвления потока управления на диаграмме последовательности*

С помощью ветвления можно изобразить и более сложную логику взаимодействия объектов между собой. Если условий более двух, то для каждого из них необходимо предусмотреть ситуацию единственного выполнения. Рассматриваемый пример относится к моделированию взаимодействия программной системы обслуживания клиентов в банке. На этом примере диаграммы последовательности объект 1 передает управление одному из трех других объектов.

## ***Временные ограничения на диаграммах последовательности***

В отдельных случаях выполнение тех или иных действий на диаграмме последовательности может потребовать явной спецификации временных ограничений, накладываемых на сам интервал выполнения операций или передачу сообщений. В языке UML для записи временных ограничений используются фигурные скобки. Временные ограничения могут относиться как к выполнению определенных действий объектами, так и к самим сообщениям, явно специфицируя условия их передачи или приема. Важно понимать, что в отличие от условий ветвления, которые должны выполняться альтернативно, временные ограничения имеют обязательный или директивный характер для ассоциированных с ними объектов.

Временные ограничения могут записываться рядом с началом стрелки соответствующего сообщения. Но наиболее часто они записываются слева от этой стрелки на одном уровне с ней. Если временная характеристика относится к конкретному объекту, то имя этого объекта записывается перед именем характеристики и отделяется от нее точкой.

## ***Рекомендации по построению диаграмм последовательности***

Как уже отмечалось, построение диаграммы последовательности целесообразно начинать с выделения из всей совокупности тех и только тех классов, объекты которых участвуют в моделируемом взаимодействии. После этого все объекты наносятся на диаграмму с соблюдением некоторого порядка инициализации сообщений. Здесь необходимо установить, какие объекты будут существовать постоянно, а какие временно - только на период выполнения ими требуемых действий.

Когда объекты визуализированы, можно приступать к спецификации сообщений. При этом следует учитывать те роли, которые играют сообщения в системе. При необходимости уточнения этих ролей надо использовать их разновидности и стереотипы. Для уничтожения объектов, которые создаются на время выполнения своих действий, нужно предусмотреть явное сообщение.

Наиболее простые случаи ветвления процесса взаимодействия можно изобразить на одной диаграмме с использованием соответствующих графических примитивов. Однако следует помнить, что каждый альтернативный поток управления может существенно затруднить понимание построенной модели. Поэтому общим правилом является визуализация каждого потока управления на отдельной диаграмме после-

довательности. В этой ситуации такие отдельные диаграммы должны рассматриваться совместно как одна модель взаимодействия.

Дальнейшая детализация диаграммы последовательности связана с введением временных ограничений на выполнение отдельных действий в системе. Для простых асинхронных сообщений временные ограничения могут отсутствовать. Однако необходимость синхронизировать сложные потоки управления, как правило, требуют введение в модель таких ограничений. Общая их запись должна следовать семантике языка объектных ограничений.

## **Особенности построения диаграмм кооперации в нотации UML**

### ***Общие положения***

Особенности взаимодействия элементов моделируемой системы могут быть представлены на диаграммах последовательности и кооперации. Если первая служит для визуализации временных аспектов взаимодействия, то диаграмма кооперации предназначена для спецификации структурных аспектов взаимодействия. Главная особенность диаграммы кооперации заключается в возможности графически представить не только последовательность взаимодействия, но и все структурные отношения между объектами, участвующими в этом взаимодействии.

Прежде всего, на диаграмме кооперации в виде прямоугольников изображаются участвующие во взаимодействии объекты, содержащие имя объекта, его класс и, возможно, значения атрибутов. Далее, как и на диаграмме классов, указываются ассоциации между объектами в виде различных соединительных линий. При этом можно явно указать имена ассоциации и ролей, которые играют объекты в данной ассоциации. Дополнительно могут быть изображены динамические связи - потоки сообщений. Они представляются также в виде соединительных линий между объектами, над которыми располагается стрелка с указанием направления, имени сообщения и порядкового номера в общей последовательности инициализации сообщений.

В отличие от диаграммы последовательности, на диаграмме кооперации изображаются только отношения между объектами, играющими определенные роли во взаимодействии. С другой стороны, на этой диаграмме не указывается время в виде отдельного измерения. Поэтому последовательность взаимодействий и параллельных потоков может быть определена с помощью порядковых номеров. Следовательно, если необходимо явно специфицировать взаимосвязи между объектами в реальном времени, лучше это делать на диаграмме последовательности.

Поведение системы может описываться на уровне отдельных объектов, которые обмениваются между собой сообщениями, чтобы достичь нужной цели или реализовать некоторый сервис. С точки зрения аналитика или конструктора важно представить в проекте системы структурные связи отдельных объектов между собой. Такое статическое представление структуры системы как совокупности взаимодействующих объектов и обеспечивает диаграмма кооперации.

Таким образом, с помощью диаграммы кооперации можно описать полный контекст взаимодействий как своеобразный временной "среза" совокупности объектов, взаимодействующих между собой для выполнения определенной задачи или бизнес-цели программной системы.

## ***Кооперация***

Понятие кооперации (collaboration) является одним из фундаментальных понятий в языке UML. Оно служит для обозначения множества взаимодействующих с определенной целью объектов в общем контексте моделируемой системы. Цель самой кооперации состоит в том, чтобы специфицировать особенности реализации отдельных наиболее значимых операций в системе. Кооперация определяет структуру поведения системы в терминах взаимодействия участников этой кооперации.

Кооперация может быть представлена на двух уровнях:

- На уровне спецификации - показывает роли классификаторов и роли ассоциаций в рассматриваемом взаимодействии.
- На уровне примеров - указывает экземпляры и связи, образующие отдельные роли в кооперации.

Диаграмма кооперации уровня спецификации показывает роли, которые играют участвующие во взаимодействии элементы. Элементами кооперации на этом уровне являются классы и ассоциации, которые обозначают отдельные роли классификаторов и ассоциации между участниками кооперации.

Диаграмма кооперации уровня примеров представляется совокупностью объектов (экземпляры классов) и связей (экземпляры ассоциаций). При этом связи дополняются стрелками сообщений. На данном уровне показываются только релевантные объекты, т. е. имеющие непосредственное отношение к реализации операции или классификатора.

В кооперации уровня примеров определяются свойства, которые должны иметь экземпляры для того, чтобы участвовать в кооперации. Кроме свойств объектов на диаграмме кооперации также указываются ассоциации, которые должны иметь место между объектами кооперации. При этом вовсе не обязательно изображать все свойства или все ассоциации, поскольку на диаграмме кооперации присутствуют только ро-

ли классификаторов, но не сами классификаторы. Таким образом, в то время как классификатор требует полного описания всех своих экземпляров, роль классификатора требует описания только тех свойств и ассоциаций, которые необходимы для участия в отдельной кооперации.

Отсюда вытекает важное следствие. Одна и та же совокупность объектов может участвовать в различных кооперациях. При этом, в зависимости от рассматриваемой кооперации, могут изменяться как свойства отдельных объектов, так и связи между ними. Именно это отличает диаграмму кооперации от диаграммы классов, на которой должны быть указаны все свойства и ассоциации между элементами диаграммы.

### ***Связи на диаграммах кооперации***

Связь (link) является экземпляром или примером произвольной ассоциации. Связь как элемент языка UML может иметь место между двумя и более объектами. Бинарная связь на диаграмме кооперации изображается отрезком прямой линии, соединяющей два прямоугольника объектов. На каждом из концов этой линии могут быть явно указаны имена ролей данной ассоциации. Рядом с линией в ее средней части может записываться имя соответствующей ассоциации.

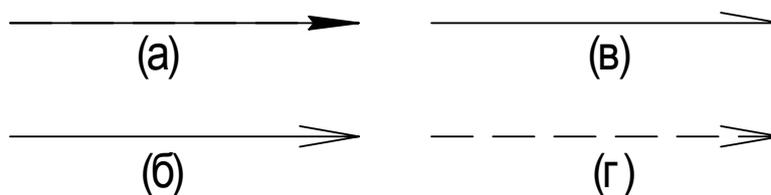
Связи не имеют собственных имен, поскольку полностью идентичны как экземпляры ассоциации. Другими словами, все связи на диаграмме кооперации могут быть только анонимными и записываются без двоеточия перед-именем ассоциации. Для связей не указывается также и кратность. Однако другие обозначения специальных случаев ассоциации (агрегация, композиция) могут присутствовать на отдельных концах связей.

### ***Сообщения на диаграммах кооперации***

При построении диаграммы кооперации они имеют некоторые дополнительные семантические особенности. Сообщение на диаграмме кооперации специфицирует коммуникацию между двумя объектами, один из которых передает другому некоторую информацию. При этом первый объект ожидает, что после получения сообщения вторым объектом последует выполнение некоторого действия. Таким образом, именно сообщение является причиной или стимулом для начала выполнения операций, отправки сигналов, создания и уничтожения отдельных объектов. Связь обеспечивает канал для направленной передачи сообщений между объектами от объекта-источника к объекту-получателю.

Сообщения в языке UML также специфицируют роли, которые играют объекты - отправитель и получатель сообщения. Сообщения на диаграмме кооперации изображаются помеченными стрелками рядом

(выше или ниже) с соответствующей связью или ролью ассоциации. Направление стрелки указывает на получателя сообщения. Внешний вид стрелки сообщения имеет определенный смысл. На диаграммах кооперации может использоваться один из четырех типов стрелок для обозначения сообщений (рис. 13):



*Рис. 13. Графическое изображение различных типов сообщений на диаграмме кооперации*

- Сплошная линия с треугольной стрелкой (рис. 13, а) обозначает вызов процедуры или другого вложенного потока управления. Может быть также использована совместно с параллельно активными объектами, когда один из них передает сигнал и ожидает, пока не закончится некоторая вложенная последовательность действий.
- Сплошная линия с V-образной стрелкой (рис. 13, б) обозначает простой поток управления. Каждая такая стрелка изображает один этап в последовательности потока управления. Обычно все такие сообщения являются асинхронными.
- Сплошная линия с полустрелкой (рис. 13, в) используется для обозначения асинхронного потока управления. Обычно сообщения этого типа являются начальными в последовательности потока управления и чаще всего инициируются актерами.
- Пунктирная линия с V-образной стрелкой (рис. 13, г) обозначает возврат из вызова процедуры. Стрелки этого типа зачастую отсутствуют на диаграммах кооперации, поскольку неявно предполагается их существование после окончания процесса активизации некоторой деятельности.

### ***Рекомендации по построению диаграмм кооперации***

Построение диаграммы кооперации можно начинать сразу после построения диаграммы вариантов использования. В этом случае каждый из вариантов использования может быть специфицирован в виде отдельной диаграммы кооперации уровня спецификации. Эта диаграмма спо-

способствует более полному пониманию особенностей реализации функций системой, хотя и не может содержать всю информацию, необходимую для их реализации.

Следует помнить, что на диаграмме кооперации изображаются только те объекты, которые непосредственно в ней участвуют. При этом объекты могут выступать в различных ролях, которые должны быть явно указаны на соответствующих концах связей диаграммы. Применение стереотипов унифицирует кооперацию, обеспечивая ее адекватную интерпретацию как со стороны заказчиков, так и со стороны разработчиков. Тем не менее, целесообразно различать терминологию, используемую на диаграммах кооперации уровня спецификации и уровня примеров.

Так, при построении диаграмм кооперации уровня спецификации желательно применять наиболее понятную заказчику терминологию, избегая технических фраз и словосочетаний. Например, "оформить заказ", "отгрузить товар", "представить отчет", "разработать план" и т. д. Такие известные разработчикам слова как "сервер", "защищенный протокол", "закрытая операция класса", а также стереотипы и помеченные значения на этом уровне применять не рекомендуется. На уровне спецификации нужно стремиться достичь по возможности полного взаимопонимания между заказчиком и командой разработчиков всех вариантов использования проектируемой системы в контексте их кооперации.

При построении диаграмм кооперации уровня примеров терминология должна наиболее точно отражать все аспекты реализации соответствующих объектов и связей. Поскольку диаграмма этого уровня является документацией для разработчиков системы, здесь допустимо использовать весь арсенал стереотипов, ограничений и помеченных значений, который имеется в языке UML. Если типовых обозначений недостаточно, разработчики могут дополнить диаграмму собственными элементами, используя механизм расширений языка UML.

## **Руководство к выполнению самостоятельной работы**

Поскольку самостоятельная работа проводится с использованием инструментального средства Rational Rose, то для выполнения работы необходима предварительная установка данного программного обеспечения. Реализация работы осуществляется на базе компьютерного класса кафедры ТПС.

На практическом занятии требуется разработать модели последовательности и кооперации ТПП в нотации UP.

### **Пример выполнения**

#### ***Разработка модели последовательности***

В качестве примера рассматривается построение диаграммы кооперации для моделирования процесса телефонного разговора с использованием обычной телефонной сети. Объектами в этом примере являются два абонента *a* и *b*, два телефонных аппарата, коммутатор и сам разговор как объект моделирования.

Абоненты в данном случае рассматриваются как акторы, первый из них играет активную роль, а второй – пассивную. Поэтому первый получает фокус управления сразу после своего появления в системе, а второй имеет только линию жизни. Разговор как объект появляется только после установки соединения и уничтожается с его прекращением. Поэтому он изображается позже на диаграмме последовательности.

Окончательный вариант диаграммы последовательности может содержать некоторые временные ограничения и комментарии (рис. 14).

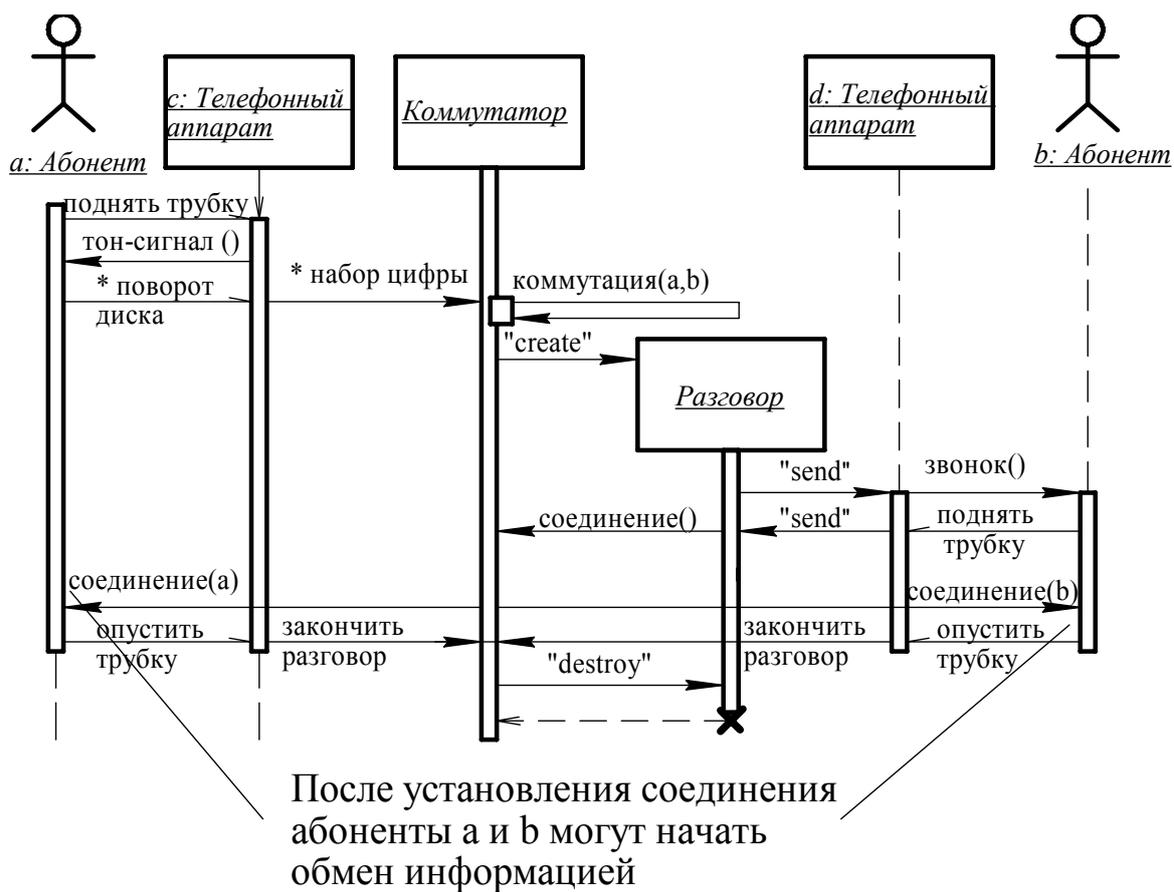


Рис. 14. Окончательный вариант диаграммы последовательности для моделирования телефонного разговора

### Разработка модели кооперации

В качестве примера также рассматривается модель телефонного разговора.

В последующем необходимо специфицировать все связи на этой диаграмме, указав на их концах необходимую информацию в форме ролей связей. Наконец, на диаграмму кооперации необходимо нанести все сообщения, указав их порядок и семантические особенности. Окончательный фрагмент диаграммы кооперации изображен на рис. 15.

На диаграмме кооперации изображаются только те объекты, которые непосредственно в ней участвуют. При этом объекты могут выступать в различных ролях, которые должны быть явно указаны на соответствующих концах связей диаграммы.

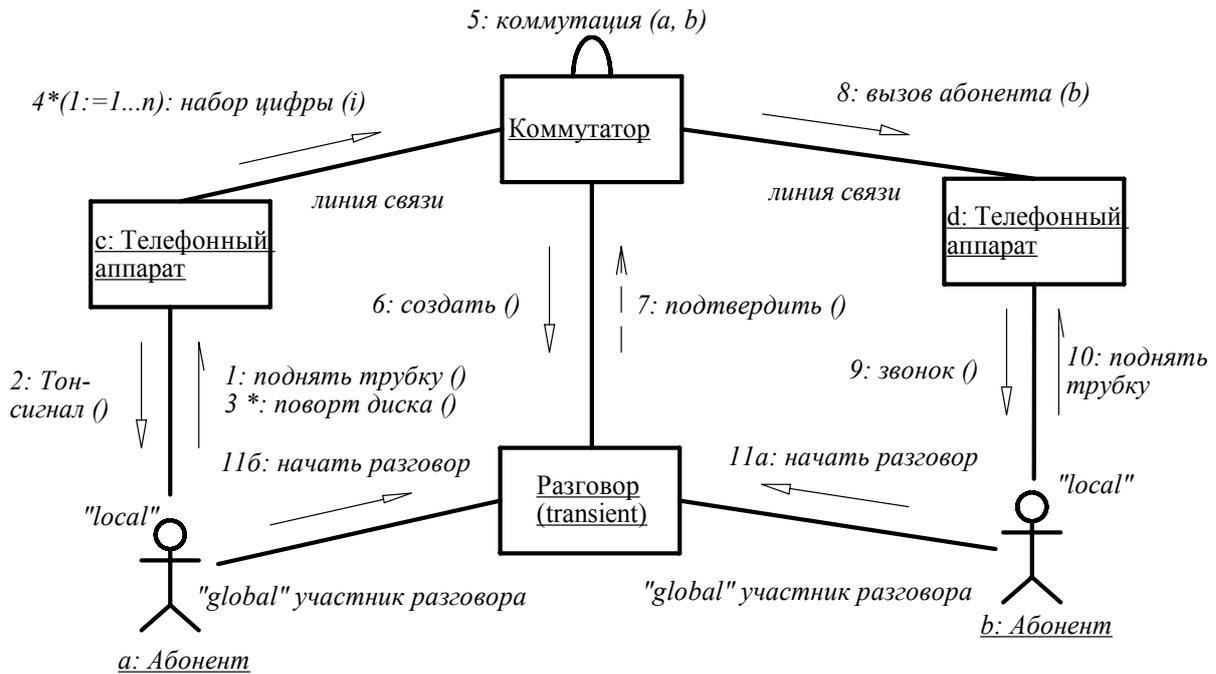


Рис. 15. Окончательный вариант диаграммы кооперации для моделирования телефонного разговора

## Содержание отчёта

Отчёт по самостоятельной работе должен содержать:

- Исходные данные (описание бизнес-процессов ТПП, задействованных в построении).
- Постановку задачи.
- Результат построения: диаграммы деятельности UML.
- Сравнительный анализ возможностей по описанию моделируемой предметной области диаграмм последовательности и кооперации UML.
- Таблицу результатов функционального анализа диаграмм последовательности и кооперации UML.
- Сравнительный анализ функциональных возможностей статических и динамических моделей UML по описанию предметной области ТПП.
- Таблицу результатов функционального анализа статических и динамических моделей UML.
- Выводы по работе.

Задание выполняется индивидуально.

## **Самостоятельная работа «Моделирование ТПП с использованием диаграмм деятельности UML»**

### **Цель работы**

Цель данной практической работы состоит в том, чтобы на примере создания диаграмм деятельности на языке UML:

- ознакомиться с особенностями представления бизнес-процессов предприятия в нотации UML;
- определить достоинства и недостатки языка UML применительно к описанию ТПП;
- провести сравнительный анализ нотаций UML и ADONIS для описания процессов ТПП (выделить критерии сравнения).

В результате выполнения самостоятельной работы студенты приобретут навыки по работе с системой, проведут анализ функциональных возможностей системы для описания предметной области ТПП.

### **Особенности построения диаграмм деятельности в нотации UML.**

#### ***Общие положения***

Язык UML представляет собой общецелевой язык визуального моделирования, который разработан для спецификации, визуализации, проектирования и документирования компонентов программного обеспечения, бизнес-процессов и других систем. Язык UML одновременно является простым и мощным средством моделирования, который может быть эффективно использован для построения концептуальных, логических и графических моделей сложных систем самого различного целевого назначения. Этот язык вобрал в себя наилучшие качества методов программной инженерии, которые с успехом использовались на протяжении последних лет при моделировании больших и сложных систем.

Конструктивное использование языка UML основывается на понимании общих принципов моделирования сложных систем и особенностей процесса объектно-ориентированного анализа и проектирования в частности. Выбор выразительных средств для построения моделей сложных систем предопределяет те задачи, которые могут быть решены с использованием данных моделей. При этом одним из основных принципов построения моделей сложных систем является принцип абстрагирования, который предписывает включать в модель только те аспекты проектируемой системы, которые имеют непосредственное отношение к

выполнению системой своих функций или своего целевого предназначения. При этом все второстепенные детали опускаются, чтобы чрезмерно не усложнять процесс анализа и исследования полученной модели.

Другим принципом построения моделей сложных систем является принцип многомодельности. Этот принцип представляет собой утверждение о том, что никакая единственная модель не может с достаточной степенью адекватности описывать различные аспекты сложной системы.

В рамках языка UML все представления о модели сложной системы фиксируются в виде специальных графических конструкций, получивших название диаграмм. В терминах языка UML определены следующие виды диаграмм:

- Диаграмма вариантов использования (use case diagram).
- Диаграмма классов (class diagram).
- Диаграммы поведения (behavior diagrams).
  - Диаграмма состояний (statechart diagram).
  - Диаграмма деятельности (activity diagram).
  - Диаграммы взаимодействия (interaction diagrams).
    - Диаграмма последовательности (sequence diagram).
    - Диаграмма кооперации (collaboration diagram).
- Диаграммы реализации (implementation diagrams).
  - Диаграмма компонентов (component diagram).
  - Диаграмма развертывания (deployment diagram).

Из перечисленных выше диаграмм некоторые служат для обозначения двух и более других подвидов диаграмм. При этом в качестве самостоятельных представлений в языке UML используются следующие диаграммы:

- Диаграмма вариантов использования.
- Диаграмма классов.
- Диаграмма состояний.
- Диаграмма деятельности.
- Диаграмма последовательности.
- Диаграмма кооперации.
- Диаграмма компонентов.
- Диаграмма развертывания.

Перечень этих диаграмм и их названия являются каноническими в том смысле, что представляют собой неотъемлемую часть графической нотации языка UML.

При этом совокупность построенных таким образом диаграмм (рис. 16) является самодостаточной в том смысле, что в них содержится вся информация, которая необходима для реализации проекта сложной системы. Каждая из этих диаграмм детализирует и конкретизирует различные представления о модели сложной системы в терминах языка UML.

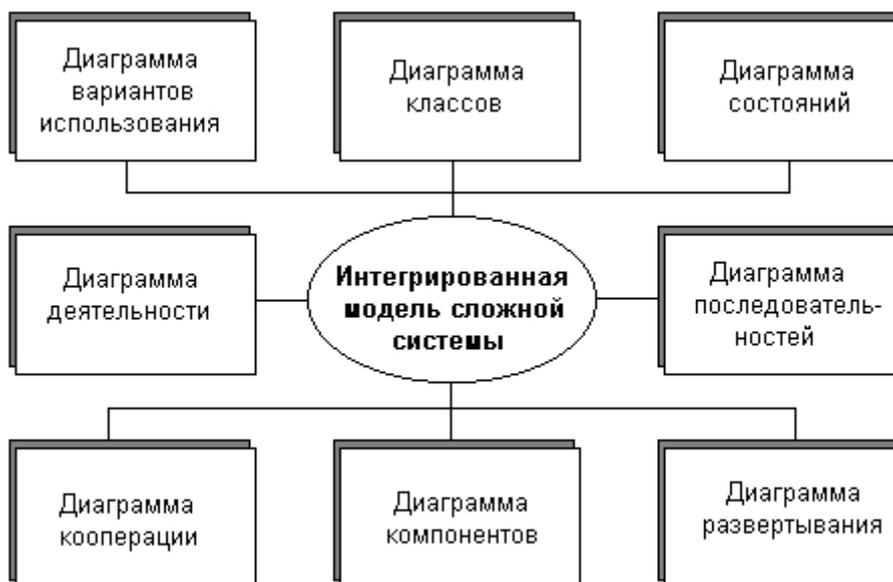


Рис. 16. Интегрированная модель сложной системы в нотации UML

Построение диаграмм деятельности можно отнести к статическому функциональному анализу. Это связано с тем, что данный вид диаграмм хотя и используются для описания динамики поведения систем, но параметр времени в явном виде в них не присутствует.

При моделировании поведения проектируемой или анализируемой системы возникает необходимость не только представить процесс изменения ее состояний, но и детализировать особенности алгоритмической и логической реализации выполняемых системой операций. Традиционно для этой цели использовались блок-схемы или структурные схемы алгоритмов. Каждая такая схема акцентирует внимание на последовательности выполнения определенных действий или элементарных операций, которые в совокупности приводят к получению желаемого результата. Увеличением сложности системы строгое соблюдение последовательности выполняемых операций приобретает все более важное значение.

Для моделирования процесса выполнения операций в языке UML используются диаграммы деятельности. Применяемая в них графическая нотация во многом похожа на нотацию диаграммы состояний: также присутствуют обозначения состояний и переходов. Отличие заключается в семантике состояний, которые используются для представления операций, а не состояний (деятельностей) и в отсутствии на переходах

сигнатуры событий. Каждое состояние на диаграмме деятельности соответствует выполнению некоторой элементарной операции, а переход в следующее состояние срабатывает только при завершении текущей операции.

Графически диаграмма деятельности представляется в форме графа деятельности, вершинами которого являются состояния действия, а дугами - переходы от одного состояния действия к другому.

Таким образом, диаграммы деятельности можно считать частным случаем диаграмм состояний. Именно они позволяют реализовать в языке UML особенности процедурного и синхронного управления.

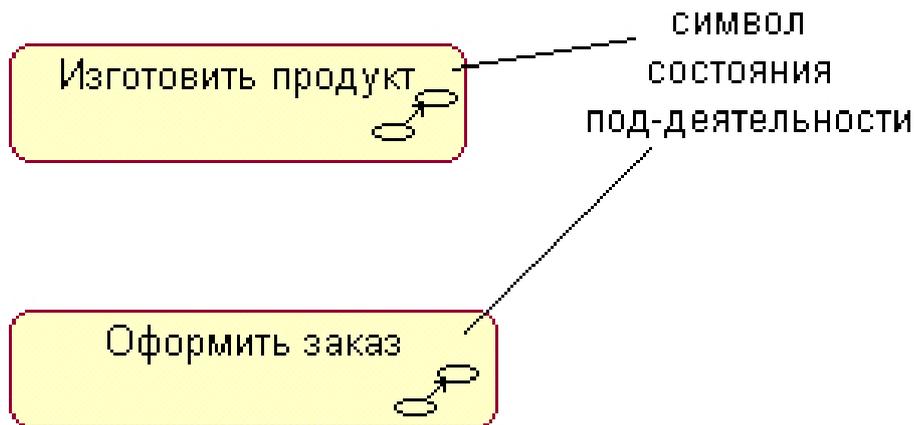
На диаграмме деятельности отображается логика или последовательность перехода от одной деятельности к другой, при этом внимание фиксируется на результате деятельности.

### ***Действия на диаграмме деятельности***

Графически состояние действия изображается фигурой, напоминающей прямоугольник, боковые стороны которого заменены выпуклыми дугами. Внутри этой фигуры записывается выражение действия (action-expression), которое должно быть уникальным в пределах одной диаграммы деятельности. Действие не может иметь внутренних переходов, поскольку оно является элементарным.

Рекомендуется в качестве имени простого действия использовать глагол с пояснительными словами. Иногда возникает необходимость представить на диаграмме деятельности некоторое сложное действие, которое, в свою очередь, состоит из нескольких более простых действий. В этом случае можно использовать специальное обозначение так называемого состояния под-деятельности (subactivity state). Такое состояние также является графом деятельности и обозначается специальной пиктограммой в правом нижнем углу символа (рис. 17).

Каждая диаграмма деятельности должна иметь единственное начальное и конечное состояние. Саму диаграмму деятельности принято располагать таким образом, чтобы действия следовали сверху вниз.



*Рис. 17. Графическое изображение состояния под-деятельности*

### ***Переходы на диаграмме деятельности***

При построении диаграммы деятельности используются только не-триггерные переходы, т. е. такие, которые срабатывают сразу после завершения деятельности или выполнения соответствующего действия. Этот переход переводит деятельность в последующее состояние сразу, как только закончится действие в предыдущем состоянии. На диаграмме такой переход изображается сплошной линией со стрелкой. Если из состояния действия выходит единственный переход, то он может быть никак не помечен.

Если же таких переходов несколько, то «сработать» может только один из них. Именно в этом случае для каждого из таких переходов должно быть явно записано сторожевое условие в прямых скобках.

Графически ветвление на диаграмме деятельности обозначается небольшим ромбом, внутри которого нет никакого текста. В этот ромб может входить только одна стрелка от того состояния действия, после выполнения которого поток управления должен быть продолжен по одной из взаимно исключающих ветвей.

Пример изображения перехода представлен на рис. 18.

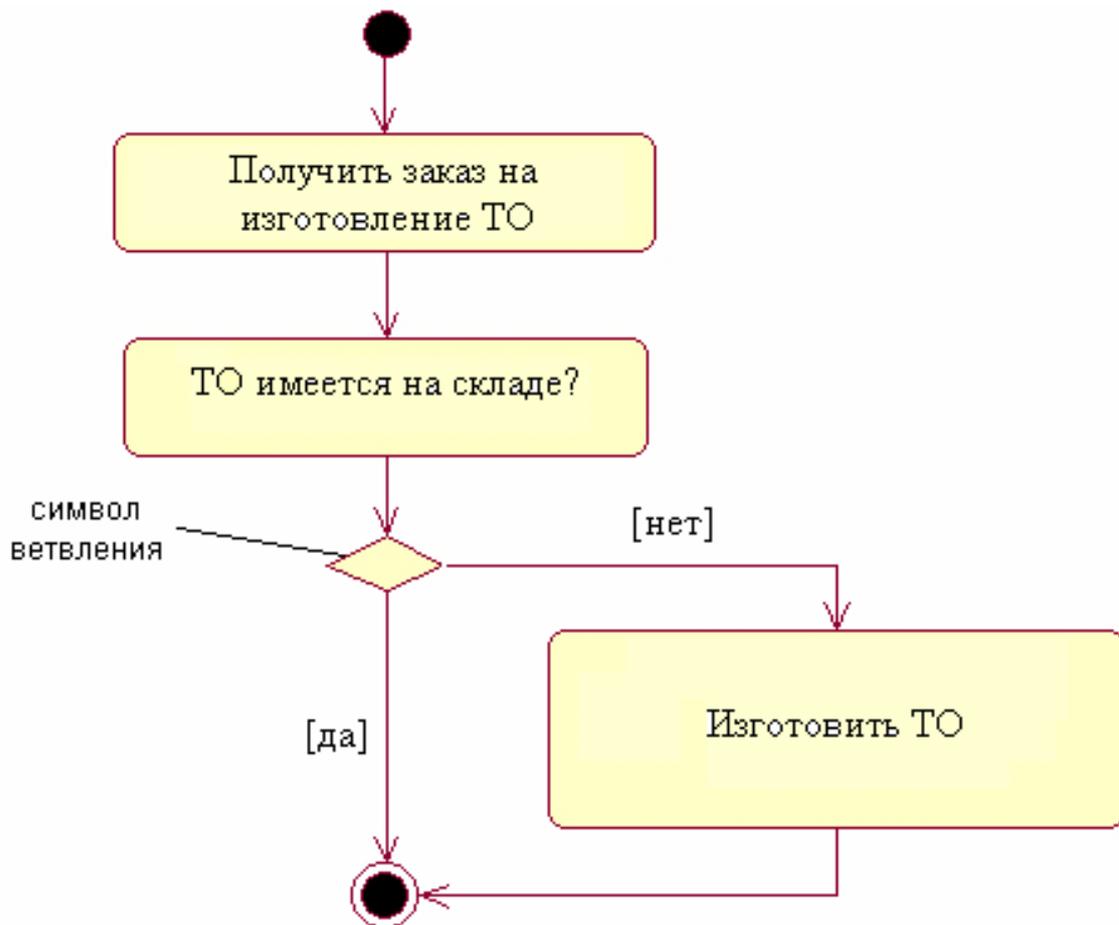


Рис. 18. Графическое изображение условного перехода

Один из наиболее значимых недостатков обычных блок-схем или структурных схем алгоритмов связан с проблемой изображения параллельных ветвей отдельных вычислений. Поскольку распараллеливание вычислений существенно повышает общее быстродействие программных систем, необходимы графические примитивы для представления параллельных процессов. В языке UML для этой цели используется специальный символ для разделения и слияния параллельных вычислений или потоков управления (рис. 19).

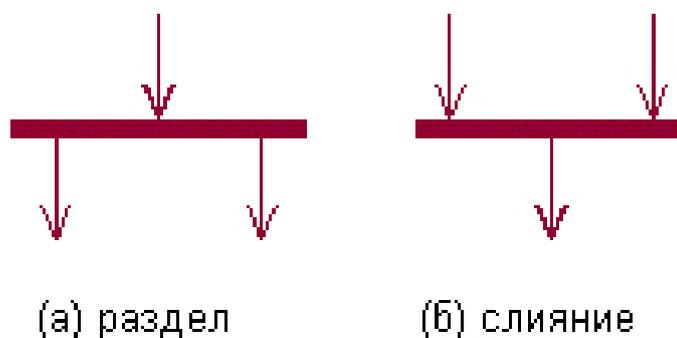


Рис. 19. Графическое изображение начального и конечного состояний на диаграмме состояний

## ***Дорожки диаграммы деятельности***

Для отображения отделов предприятия, задействованных в моделируемом процессе, в языке UML используется специальная конструкция, получившее название дорожки (swimlanes). При этом все состояния действия на диаграмме деятельности делятся на отдельные группы, которые отделяются друг от друга вертикальными линиями. Две соседние линии и образуют дорожку, а группа состояний между этими линиями выполняется отдельным подразделением (отделом, группой, отделением, филиалом) компании.

Названия подразделений явно указываются в верхней части дорожки. Пересекать линию дорожки могут только переходы, которые в этом случае обозначают выход или вход потока управления в соответствующее подразделение компании. Порядок следования дорожек не несет какой-либо семантической информации и определяется соображениями удобства.

## ***Объекты на диаграммах деятельности***

В общем случае действия на диаграмме деятельности выполняются над теми или иными объектами. Эти объекты либо иницируют выполнение действий, либо определяют некоторый результат этих действий. При этом действия специфицируют вызовы, которые передаются от одного объекта графа деятельности к другому. Поскольку в таком ракурсе объекты играют определенную роль в понимании процесса деятельности, иногда возникает необходимость явно указать их на диаграмме деятельности.

Для графического представления объектов, как уже упоминалось в главе 5, используются прямоугольник класса, с тем отличием, что имя объекта подчеркивается. Далее после имени может указываться характеристика состояния объекта в прямых скобках. Такие прямоугольники объектов присоединяются к состояниям действия отношением зависимости пунктирной линией со стрелкой. Соответствующая зависимость определяет состояние конкретного объекта после выполнения предшествующего действия.

На диаграмме деятельности с дорожками расположение объекта может иметь некоторый дополнительный смысл. А именно, если объект расположен на границе двух дорожек, то это может означать, что переход к следующему состоянию действия в соседней дорожке ассоциирован с готовностью некоторого документа (объект в некотором состоянии). Если же объект целиком расположен внутри дорожки, то и состояние этого объекта целиком определяется действиями данной дорожки.

На диаграмме деятельности один и тот же объект может быть изображен несколько раз, что не должно вносить неопределенность в спецификацию состояний действия. Пример отображения объектов на диаграмме деятельности приведен на рис. 20.

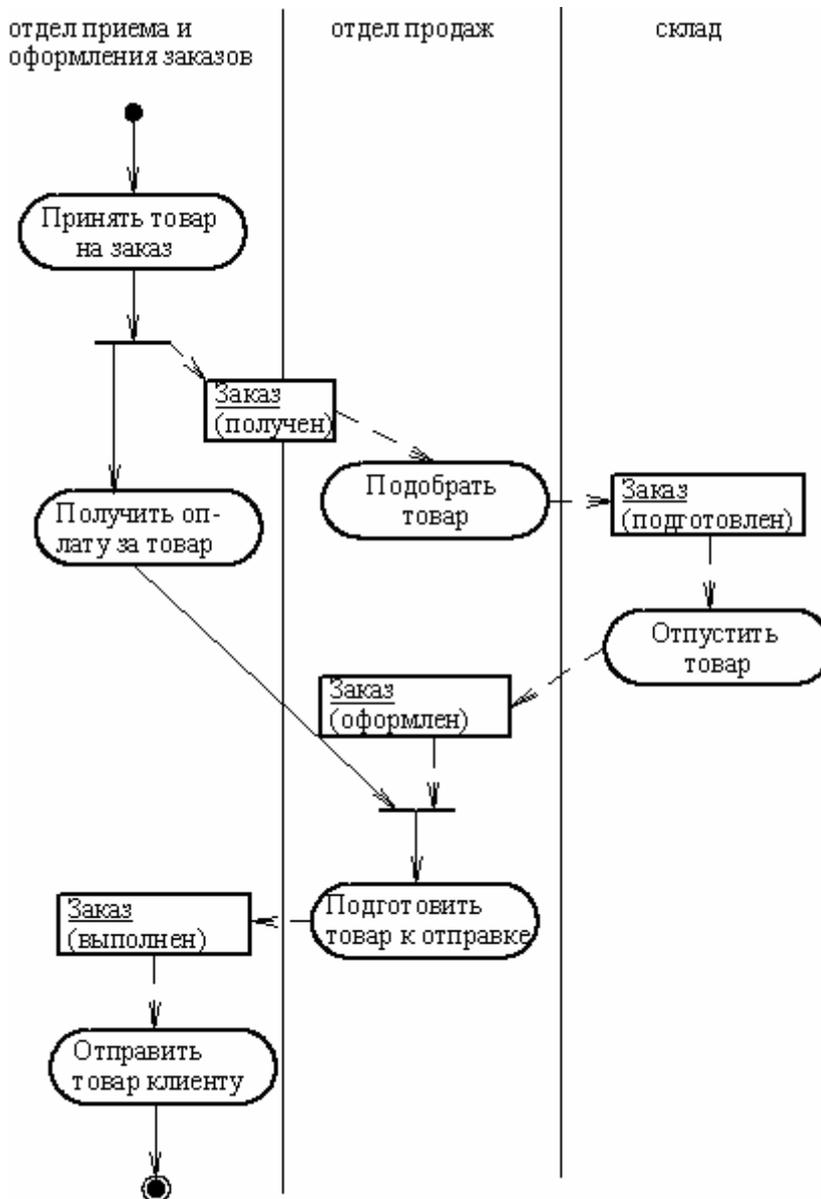


Рис. 20. Графическое изображение объектов на диаграмме деятельности

### Рекомендации по построению диаграмм деятельности

Диаграммы деятельности играют важную роль в понимании процессов реализации алгоритмов выполнения операций классов и потоков управления в моделируемой системе. Используемые для этой цели традиционные блок-схемы алгоритмов обладают серьезными ограничениями в представлении параллельных процессов и их синхронизации. При-

менение дорожек и объектов открывает дополнительные возможности для наглядного представления бизнес-процессов, позволяя специфицировать деятельность подразделений компаний и фирм.

Содержание диаграммы деятельности во многом напоминает диаграмму состояний, хотя и не тождественно ей.

Деятельностей в проектируемой системе неизвестны, построение диаграммы деятельности начинают с выделения поддеятельностей, которые в совокупности образуют деятельность подсистем. В последующем, по мере разработки диаграмм классов и состояний, эти поддеятельности уточняются в виде отдельных вложенных диаграмм деятельности компонентов подсистем, какими выступают классы и объекты.

С другой стороны, отдельные участки рабочего процесса в существующей системе могут быть хорошо отлаженными, и у разработчиков может возникнуть желание сохранить этот механизм выполнения действий в проектируемой системе. Тогда строится диаграмма деятельности для этих участков, отражающая конкретные особенности выполнения действий с использованием дорожек и объектов. В последующем такая диаграмма вкладывается в более общие диаграммы деятельности для подсистемы и системы в целом, сохраняя свой уровень детализации.

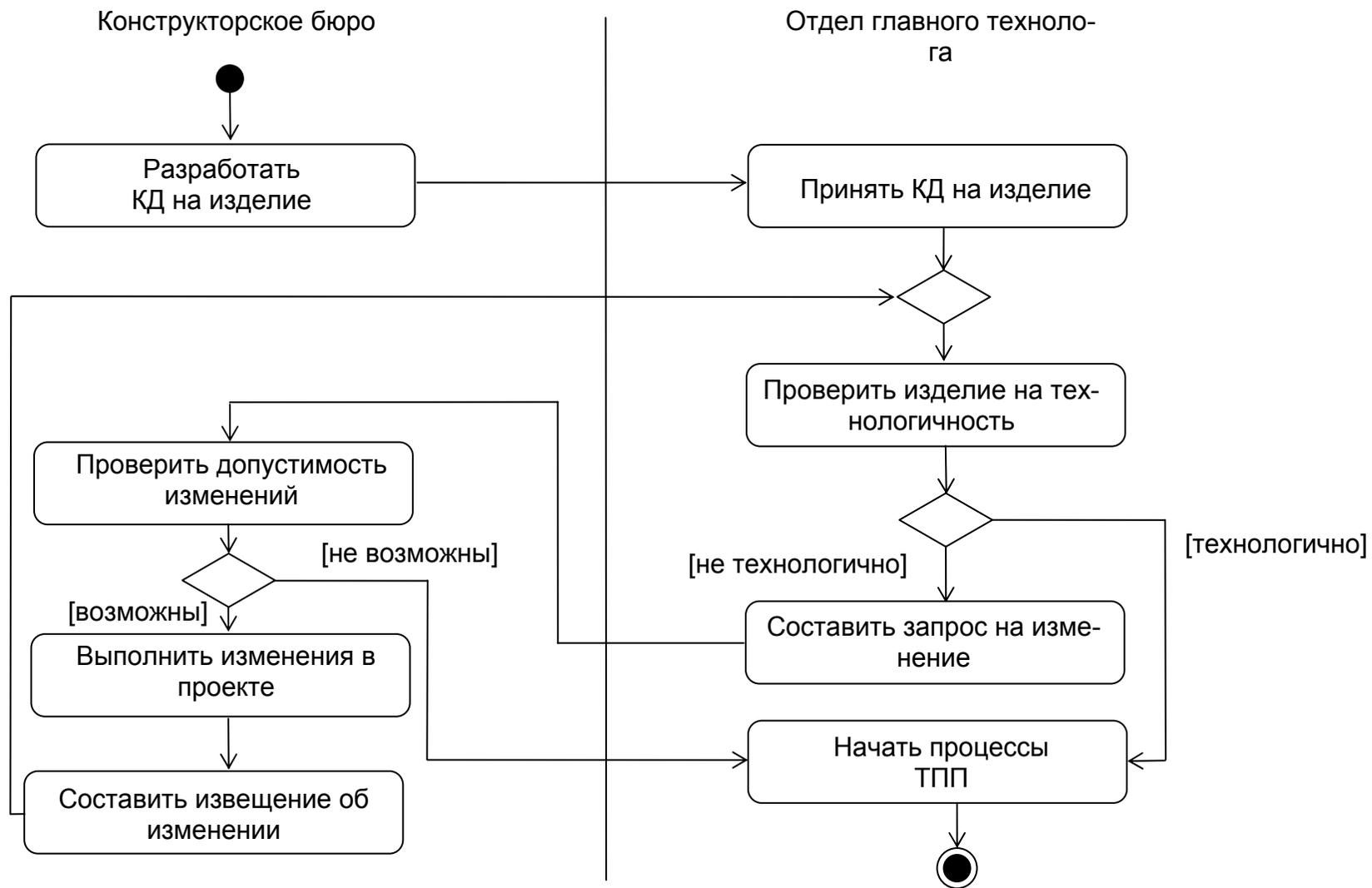
Таким образом, процесс объектно-ориентированного анализа и проектирования сложных систем представляется как последовательность итераций нисходящей и восходящей разработки отдельных диаграмм, включая и диаграмму деятельности. Доминирование того или иного из направлений разработки определяется особенностями конкретного проекта и его новизной.

На начальных этапах проектирования, когда детали реализации

Необходимо синхронизировать отдельные действия на диаграмме деятельности. Такая необходимость возникает всякий раз, когда параллельно выполняемые действия оказывают влияние на друг на друга.

В заключение следует заметить, что диаграмма деятельности, так же как и другие виды диаграмм UML, не содержит средств выбора оптимальных решений. При разработке сложных проектов проблема выбора оптимальных решений становится весьма актуальной.

На рисунке 21 представлен пример диаграммы деятельности технологического назначения – модель «Проверки КД изделия на технологичность».



*Рис. 21. Диаграмма деятельности «Проверка КД изделия на технологичность».*

## Руководство к выполнению самостоятельной работы

Поскольку самостоятельная работа проводится с использованием инструментального средства Rational Rose, то для выполнения работы необходима предварительная установка данного программного обеспечения. Реализация работы осуществляется на базе компьютерного класса кафедры ТПС.

На практическом занятии требуется разработать модели деятельности ТПП в нотации UP.

### Пример выполнения

Диаграммы деятельности разрабатываются в несколько уровней в зависимости от требуемой степени детализации системы. При этом каждый следующий уровень детализации содержится на диаграмме верхнего уровня в виде поддеятельности.

Целесообразно моделирование системы начинать с наиболее концептуального (общего) уровня, постепенно переходя на более детальные модели. Каждая диаграмма обязательно должна быть логически завершенной. Не следует перегружать каждую отдельную модель информацией, это усложнит понимание модели и затруднит ее освоение конечным пользователем.

Проектируемая диаграмма должна содержать операции управления: «принять заказ», «передать техническое задание» и пр. Это необходимо для регламентирования взаимодействия между отделами.

В качестве примера рассмотрим модель, представленную на рис. 21.

В процессе проверки конструкторской документации на технологичность в самом общем случае задействованы два отдела:

- отдел главного технолога;
- конструкторское бюро.

Таким образом, для большей наглядности целесообразно разделить диаграмму на два участка (дорожки). Проектируемый бизнес-процесс имеет несколько условных переходов.

Процесс создания диаграммы деятельности включает последовательное описание операций, распределение их по дорожкам (исполнителям), проверку соответствия числа разделений и слияний переходов и описание переходов.

Необходимо проследить, чтобы полученная модель не приводила к заикливанию процесса. Неформальный характер языка UML не позволяет провести проверку автоматически.

## Содержание отчёта

Отчёт по самостоятельной работе должен содержать:

- Исходные данные (описание бизнес-процессов ТПП, задействованных в построении).
- Постановку задачи.
- Результат построения: диаграммы деятельность UML.
- Сравнительный анализ нотаций UML и ADONIS для описания процессов ТПП: определить достоинства и недостатки методов, выделить критерии сравнения, сравнить функциональные возможности систем.
- Таблицу результатов функционального анализа нотаций UML и ADONIS.
- Выводы по работе.

Задание выполняется индивидуально.

## **Самостоятельная работа «Разработка комплекса графиков производственных процессов КБ предприятия на основе методологии Workflow»**

### **Управление бизнес-процессами на основе технологии Workflow**

Как известно, подготовка к модернизации предприятия включает три этапа:

- описание существующих процессов компании;
- анализ и оптимизацию этих процессов;
- создание рабочих моделей бизнес-процессов для их реализации на производстве.

Первые два из них реализуются с помощью систем визуального и имитационного моделирования. Они были рассмотрены в самостоятельных работах «Анализ бизнес-процессов конструкторского бюро приборостроительного предприятия» и «Реинжиниринг бизнес-процессов конструкторского бюро приборостроительного предприятия». Последний этап осуществляется уже средствами специальных модулей PDM-систем, предназначенных для управления потоками производственных заданий. В данном случае для создания рабочих моделей бизнес-процессов конструкторского бюро приборостроительного предприятия используется модуль WorkFlow Smarteam. Необходимо выяснить каким

образом осуществляется переход от концептуальной (теоретической) модели компании к бизнес-процессам реального производства.

По определению, технологии WorkFlow – это методы и средства автоматизации управления потоками производственных заданий на предприятии. Более широко технологии WorkFlow определяют как автоматизацию бизнес-процессов, протекающих на предприятии и составляющих суть его деятельности.

Сегодня любой менеджер должен физически передавать задание от пользователя к пользователю, от цеха к цеху, доводя его до исполнителей в течение всего процесса работы. Но без использования специальных средств управления практически нет возможности контролировать процесс после того, как он был запущен. Это в свою очередь позволяет следить за важнейшим фактором производства – временем процесса.

Таким образом, без использования специальных средств управления практически нет возможности контролировать задание после того, как оно было запущено. На базе PDM-системы для каждого этапа ТПП можно запустить диаграмму управления бизнес-процессами – электронный вариант маршрута/последовательности этапов задания. При этом исполнители будут последовательно получать уведомление о необходимости выполнения данной работы, а так же все сопутствующие документы. Такой подход к организации производства позволяет чётко наладить работу сотрудников, повысить производительность их труда, сделать процесс более прозрачным, то есть лучше контролируемым. Весь предварительный сбор информации, построение моделей и их анализ служат основой для реализации данного механизма на практике – на заводе, в компании, в проектном бюро и пр. Это стало возможным благодаря использованию объектно-ориентированного подхода при описании проектов в PDM.

Основой для построения диаграммы управления служит модель бизнес-процесса. То есть входной информацией для создания рабочих моделей является набор диаграмм, описывающих предметную область и выполненных, например, в среде Adonis. С помощью этой системы осуществляется этап предварительного анализа и проектирования. Таким образом, системы визуального моделирования являются ресурсом для реализации в PDM механизма управления заданиями и процессами.

Существует два способа перехода от предварительных моделей среды к рабочим процессам. Первый метод – это воспроизведение подготовленных моделей оператором PDM, с помощью специальных встроенных в систему средств создания графиков управления потоками заданий. По сути, данный подход приводит к повторному конструированию каждой диаграммы. Однако для некоторых систем существует возмож-

ность прямой передачи данных. Это относится к рассматриваемым программным продуктам Adonis и Smarteam (механизм FDL Export).

Графическое отображение функциональных моделей в этих системах практически идентично. Основные отличия появляются при задании параметров процесса: временных ограничений, исполнителей, задач и других. Организационная модель создается средствами Smarteam.

Информационная модель есть ничто иное, как соединительные элементы графика WorkFlow совместно с привязанными к ним документами и другой информацией.

Для примера в системе Adonis построена модель “Разработка Технологического процесса”. Она представляет собой процедуру, содержащую соответствующую модель бизнес-процесса. Описание всей задачи представлено с помощью нескольких вложенных моделей, каждая из них соответствует конкретной подзадаче, см. рис. 22.

Графики заданий в WorkFlow представляют собой совокупность блоков и соединителей, по которым информация перемещается от одного узла или состояния к другому. Блок определяет производственное задание и его характеристики. После того, как график создан, он может быть использован для запуска процесса выполнения конкретного задания. При этом происходит “привязка” графика к конкретной ситуации.

Возвращаясь к моделям WorkFlow Smarteam, следует отметить, что блоки диаграмм (функциональные компоненты) соответствуют узлам, а отношения между ними (связи) – соединителям графика Workflow. Узел графика характеризуется также исполнителями данного действия и сроками сдачи работы; на диаграммах Adonis эти данные отражаются в свойствах функциональной компоненты.

Процесс WorkFlow – это последовательность выполняемых действий, построенная по рабочим операциям предприятия (например, разработка КД, внесение изменений в документацию и прочее). График WorkFlow подробно описывает путь, по которому протекают процессы в организации, включая задания, которые должны быть выполнены пользователями на каждом этапе. После того как график создан, он закрепляется за одним или несколькими процессами. У каждого процесса должен быть установленный по умолчанию график WorkFlow; некоторые процессы могут иметь дополнительные графики, назначаемые через утилиту FlowChart Designer.

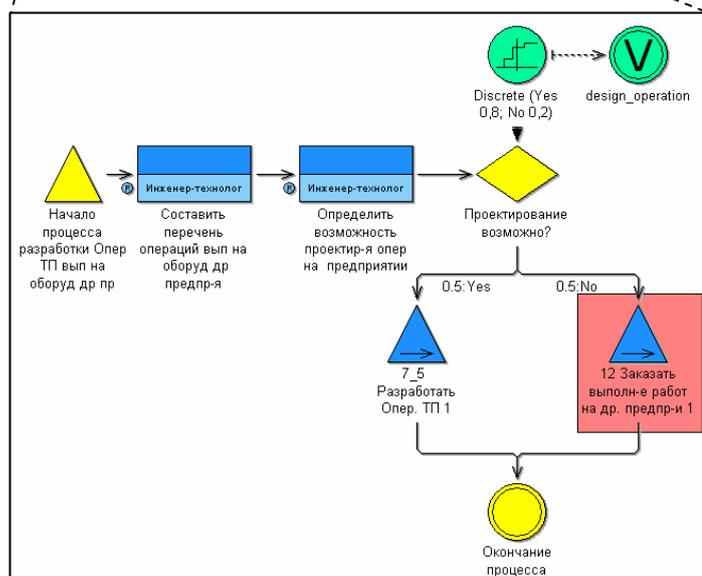
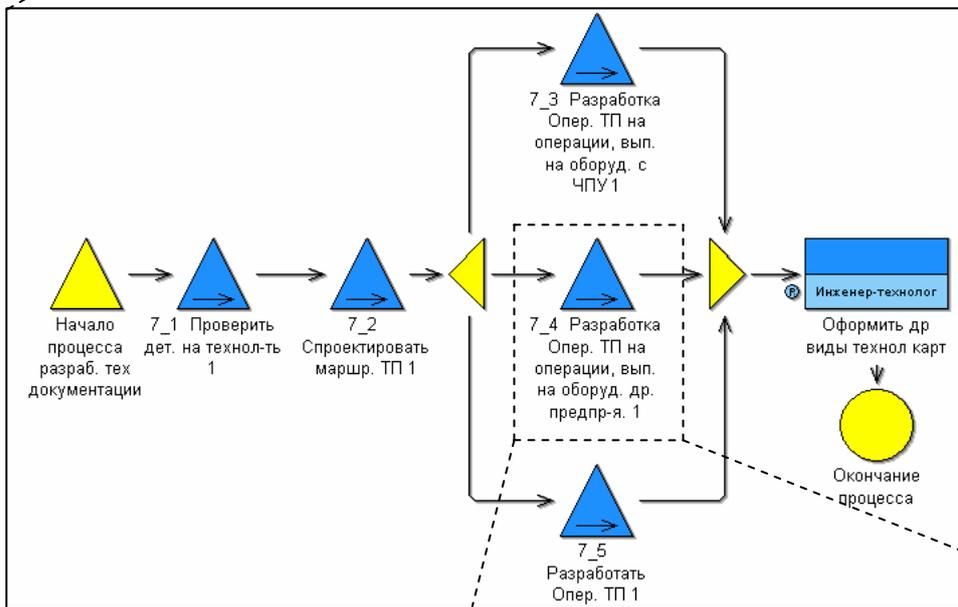
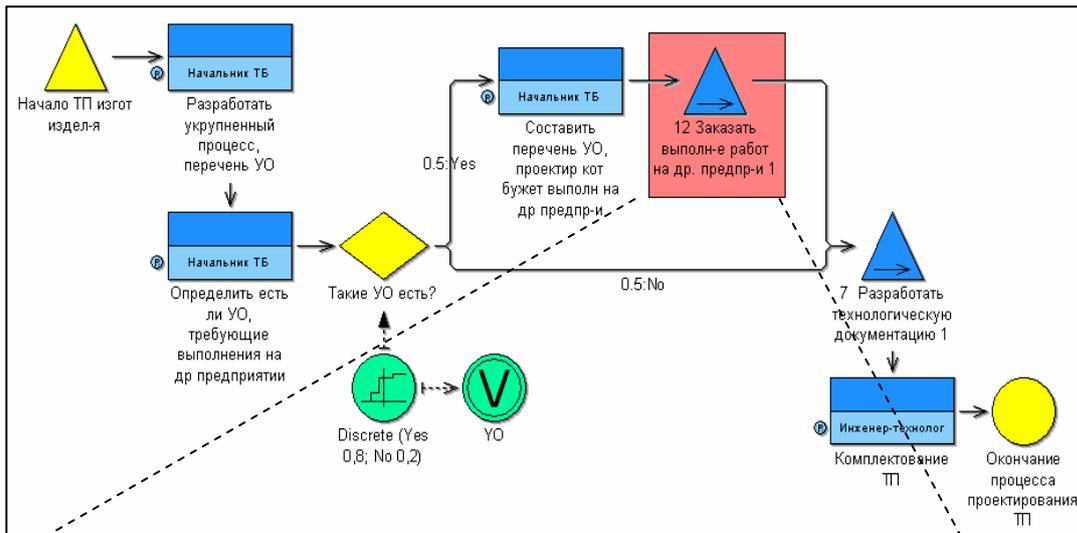


Рис. 22. Модель «Разработка технологического процесса», выполненная в системе Adonis

Этап – это основной элемент графика. Он определяет, какие действия должны быть произведены и кем, для того чтобы продвинуться по процессу на следующий шаг. Поэтому каждый этап обычно связан с одним или несколькими пользователями и заданиями, которые должны быть выполнены. При движении в процессе объекты (назначаемые при запуске графика Workflow) переходят от одного этапа к другому. Для каждого из них необходимо задать значения различных характеристик, таких как ограничение по времени выполнения, уровень важности и др.

При составлении производственного задания для каждого узла указываются такие свойства, как пользователь, действия которого в рабочем процессе соответствует этому узлу графика заданий, и задание, которое он должен выполнить, а также сроки или другие условия выполнения задания. В принципе можно создавать такие узлы, задания в которых будут выполняться не пользователем, а самой системой SmartTeam (например, передача данных или выдача сообщений). Задания бывают трёх типов:

- Ручное (пользователь просто выполняет то, что ему предписано на данном этапе, и отправляет результаты дальше согласно графику).
- Операция (выполнение каких-либо стандартных действий системой, например, “Взять на изменение”, “Утвердить” и пр.).
- Скрипт (в этом случае запускается составленная ранее программа).

В виде скрипта задаётся событие, происходящее автоматически после того, как пользователь получит процесс на определенном этапе или перешлёт его на следующий.

Каждый этап представлен иконкой. При необходимости можно задавать визуальные свойства иконки (например, назначить фотографию исполнителя): определять её форму, растровое отображение, цвет и текст, отображаемый в ней.

Соединительный элемент – это путь, определенный в графике, который направляет документы в процессе от одного этапа к другому. Проектируя график, необходимо создавать связи. Каждый соединительный элемент отображается на графике в виде стрелки в соответствии с заданным стилем. Так как пользователь может “Принять” или “Отклонить” (вернуть на изменение) задание во время работы, то необходимо предусмотреть и задать соединительные элементы обратной связи.

На рис. 23 представлены графики Workflow, соответствующие модели “Разработка технологического процесса”.

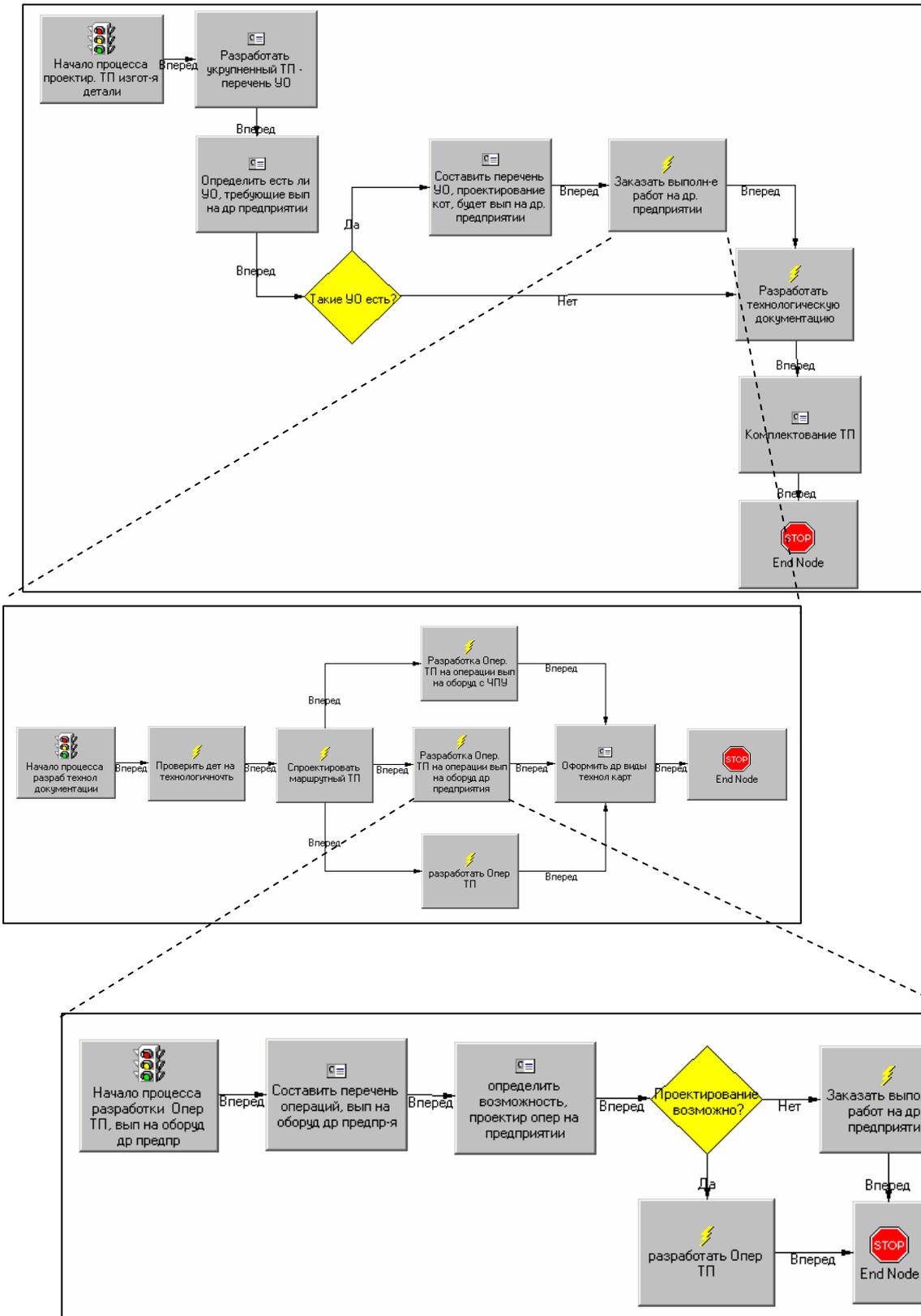


Рис. 23. Рабочая модель процесса «Разработка технологического процесса», выполненная с использованием WorkFlow – диаграмм

WorkFlow позволяет рационализировать бизнес-процессы посредством объединения людей, информации и процессов. Пакет WorkFlow является новым решением проблемы управления, и спроектирован так, чтобы дать пользователям возможность отслеживать и наблюдать процессы, а также управлять ими.

При внедрении рабочей модели накапливается статистика реальных затрат времени на исполнение всего спектра производственных задач. На основе этих данных можно проводить корректировку нормативов времени. Кроме того, в зависимости от успеха выполнения задания супервизор или инициатор процесса могут корректировать график WorkFlow в режиме реального времени.

Причина получаемой оптимизации заключается в ужесточении временных нормативов и распараллеливании работ. Деятельность сотрудников становится более организованной, исключаются затраты времени на ожидание и транспортировку.

Некоторые специалисты придерживаются мнения, что специализированные модули управления потоками заказов наподобие WorkFlow Smarteam могут выполнять функции систем визуального и имитационного моделирования. Действительно такие модули PDM-систем способны предоставить информацию о времени прохождения заданий, их частей или работы отдельных сотрудников компании. Однако подобные данные возможно будет получить лишь после опробования рабочего процесса в реальных условиях предприятия. Значит, результаты исследования будут получены только после завершения задачи, на это может уйти несколько месяцев. Сравнение различных вариантов реализации одного и того же процесса еще более проблематично. Результаты, полученные подобным образом, являются наиболее достоверными, однако для предварительного анализа и моделирования столь длительные временные задержки недопустимы. Такой вариант возможен лишь для компаний, структура бизнес-процессов которых не требует радикальных изменений.

С помощью WorkFlow возможно:

- легко и наглядно строить графики бизнес-процессов предприятия;
- назначать различные задания любому пользователю в процессе;
- распределять отдельных пользователей или группы по этапам бизнес-процесса, более того возможно динамическое перераспределение заданий;

- запускать процесс, выбрав его из списка ранее разработанных (шаблонов), указав заново необходимые документы, которые будут участвовать в данном процессе;
- автоматически отслеживать выполнение (задержку) заданий, направлять документы на следующий этап в процессе;
- определить точное местоположение (статус) документов внутри процесса, текущее состояние процесса;
- оперативно вносить изменения в структуру процесса, перенаправлять документооборот, назначать новых исполнителей, тем самым предотвращая возможность невыполнения задания и устраняя "узкие места";
- встраивать в процесс исполнения задания макросы, активация которых происходит при выполнении какого-либо заранее назначенного действия (динамический процесс).

Workflow состоит из трех элементов:

- FlowChart Designer: позволяет администратору (супервизору) проектировать графики Workflow;
- FlowBox: позволяет пользователям получать, запускать и просматривать подготовленные для них бизнес-процессы, выполнять задания и пересылать их на следующий этап графика Workflow;
- Workflow Manager: позволяет руководителю наблюдать за уже начатыми процессами, давая возможность изменять те, что вызывают проблемы и задержки.

Последовательность этапов создания графика Workflow показана на рис. 24.



*Рис. 24. Схема основных этапов создания графика WorkFlow*

После того, как график создан, он может быть использован для запуска процесса выполнения конкретного задания. Реализуется это следующим образом.

После инициирования процесса, запускаются на выполнение те этапы графика, которые непосредственно связаны соединителями со стартовым узлом. Пользователи, назначенные на процесс, средствами внутренней электронной почты получают уведомления о необходимости отработать указанные задания и всю необходимую информацию. Если задание выполнено, то это инициирует следующие этапы и т.д., до завершения процесса в конечном узле графика.

Графики заданий предусматривают возможность создания последовательных, параллельных, а также комбинированных путей (соединений). Задания в графиках могут иметь так называемые триггеры (переключатели) событий. С их помощью можно инициировать другой график заданий или программу-скрипт.

Инструкции, файлы и данные, которые могут быть задействованы, присоединяются к посылаемым сообщениям. Большие наборы данных, например, САД-модели, в действительности не посылаются, но пользователю сообщается их расположение, чтобы он мог легко получить дос-

туп к ним. Это обеспечивает небольшой размер присоединительного пакета. Отправленные данные задач и инструкции появляются в почтовом ящике получателя. Уведомления о событиях (например, об утверждении) могут быть также посланы другим пользователям, которым не нужно выполнять никаких действий.

Внедрение методов и средств WorkFlow затрагивает не только технических специалистов (конструкторов, технологов), уже знакомых с компьютерными технологиями. Оно требует определённого уровня компьютерной грамотности от большого числа других специалистов и руководителей подразделений, значительного повышения общей исполнительской дисциплины. Поэтому руководство предприятия должно быть готово к проведению соответствующих подготовительных работ по внедрению технологий WorkFlow.

## **Задание**

Результатом выполненной работы студента должен являться комплекс графиков процессов конструкторского бюро, построенных на основе методологии Workflow. В качестве основы можно взять модели бизнес-процессов, разработанных в самостоятельной работе «Реинжиниринг бизнес-процессов конструкторского бюро приборостроительного предприятия».

Таким образом, в данной самостоятельной работе необходимо выполнить следующее:

- Построить графики производственных процессов конструкторского бюро приборостроительного предприятия на основе технологии Workflow.
- Назначить исполнителей и задачи для каждого этапа процесса. При этом обратите внимание, что предварительно необходимо заполнить БД персонала SmartTeam с помощью утилиты User Maintenance.
- Задать временные характеристики для каждого этапа процесса.

## **Разработка графиков Workflow**

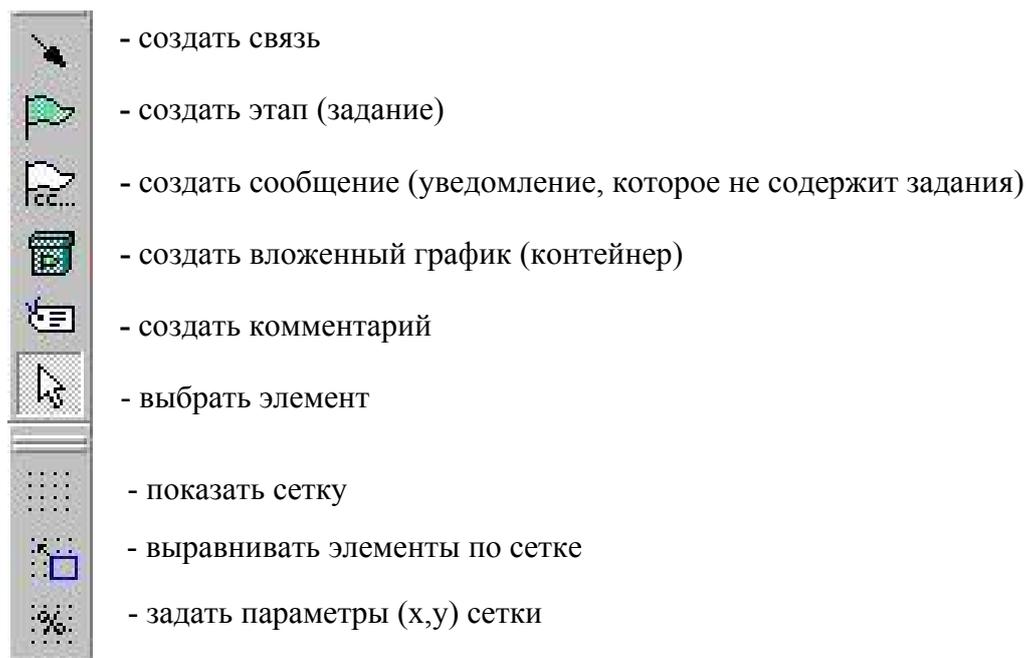
Графики WorkFlow визуально представляют собой совокупность этапов и их соединений, по которым информация перемещается из одного состояния к другому. Каждый этап описывает функцию и содержит сведения, необходимые для её выполнения. То есть перед созданием графика WorkFlow необходимо создать функциональную модель заданного бизнес-процесса.

Перед тем, как приступать к разработке модели, необходимо собрать информацию о составе и последовательности этапов процесса, возможностях проведения параллельных работ, исполнителях и примерных сроках выполнения заданий, то есть провести анализ бизнес-процесса. Но перед Вами такая задача не ставится, поскольку у Вас выполнена уже эта работа (в самостоятельной работе «Реинжиниринг бизнес-процессов конструкторского бюро приборостроительного предприятия»).

График WorkFlow соответствует определённому бизнес-процессу предприятия. Этот процесс должен быть представлен в виде схемы заранее. Вы должны чётко представлять, какие в нем задействованы исполнители и какова политика их назначения (назначаются заранее, назначаются предыдущим пользователем после завершения своей части работы), какие ограничения накладываются на время исполнения каждого этапа, параллельно или последовательно выполняются этапы графика.

Создание нового графика WorkFlow производится с помощью утилиты FlowChart Designer. Когда Вы первый раз открываете FlowChart Designer, то новый график появляется автоматически. Он содержит этапы начала и конца процесса. С помощью FlowChart Designer Вы можете создать новый график WorkFlow или изменить уже существующий (на панели инструментов Файл – Открыть). Если Вы изменяете график WorkFlow, который уже назначен на Процесс, то эти изменения не действуют на уже инициализированные бизнес-процессы.

Перед началом работы Вы можете задать свойства для всего графика целиком (на панели инструментов Вид → Свойства). При создании или редактировании графика используется панель, представленная на рис. 25.



*Рис. 25. Панель инструментов FlowChart Designer*

Для каждого этапа необходимо определить его характеристики, они задаются в окне свойств (контекстное меню этапа – Свойства (Properties) или F4), см. рис.26.

На закладке Движение (Flow) задаются основные параметры, такие как исполнители, задания (одно или несколько), ограничения по времени и прочее; на закладке События (Event) назначаются макросы, которые исполняются автоматически (перед/после/отправки/получения/открытия событий); на закладке Отображение (Visual) – вид узла (цвет, размер и др.). Параметр Задание (Tasks) обязателен для заполнения. Он может быть задан несколькими способами: в виде текста, операции или макроса (скрипта). Вторым способом удобен тем, что позволяет исключить из работы инженера выполнение стандартных действий в SMARTTEAM, таких как «Сдать руководителю», «Взять на изменение» и пр.

Для заданий, добавляемых вручную, можно установить опции: Требуемое задание, Выполнить задание по объекту, Заменить версию после операции, Задание автоматом.

Если задание обязательно, устанавливается опция Требуемое задание (Required Task). Кроме того задание может выполняться над каждым объектом – опция Выполнить задание по объекту (Perform task per object) или быть автоматическим – Задание автоматом (Automatic Task).

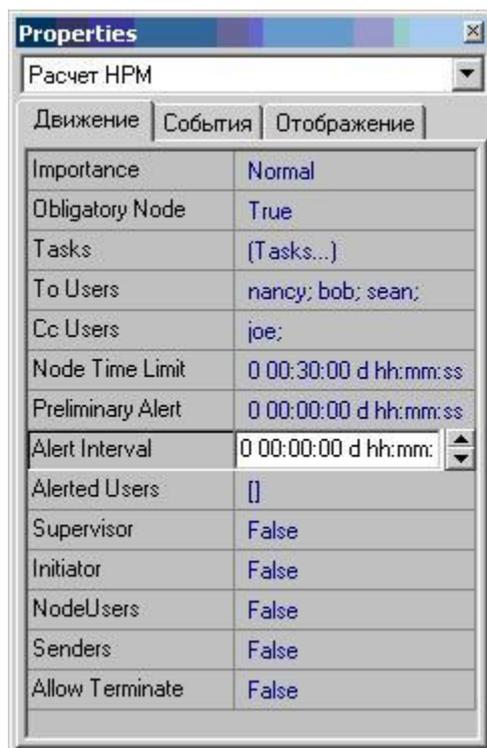


Рис. 26. Окно свойств для этапа графика WorkFlow

Существует возможность того, чтобы специалист, выполняющий задание на текущем этапе, назначал следующего исполнителя. Задается это с помощью опции Выбрать пользователей при запуске в окне назначения пользователей (вызвать его можно, нажав на клавишу обзора в поле назначения пользователей (To Users, Cc Users)).

Здесь же задается политика назначения исполнителей на задание: И – все должны выполнить задание, ИЛИ – задание выполняет первый, кто его возьмет на исполнение.

Чтобы задать исполнителей, необходимо предварительно заполнить БД персонала SMARTTEAM с помощью утилиты User Maintenance. Этот модуль позволяет определять новых пользователей, создавать группы пользователей, а так же назначать пароли и права пользователей на доступ к данным на выполнение определенных действий. С помощью этой утилиты может быть, например, задан режим «Только просмотр» для определенных пользователей или групп пользователей. Подобные настройки позволяют обеспечить оптимальные меры безопасности. Описание работы с данной утилитой представлено в самостоятельной работе «Назначение прав пользователей для сотрудников отдела конструирования технологической оснастки в системе PDM SmarTeam».

Существует возможность сохранять набор этапов в Библиотеку, чтобы многократно их использовать в других графиках. Вы имеете возможность определить различные директории в Библиотеке и с их помо-

щью организовать Этапы. Например, Вы можете создать директорию для каждого отдела в Вашей организации. Это значительно ускоряет процесс создания графика.

Помимо свойств узлов графика необходимо задать параметры соединительных элементов.

После того, как график WorkFlow создан, его необходимо сохранить и проверить на наличие ошибок. Некоторые из них система отслеживает автоматически (проверка на заикливание). Далее график необходимо “привязать” к процессу. Для этого необходимо: на панели инструментов выбрать Вид – свойства (View – properties), в поле Назначить

Процесс (Assign to Process) нажать кнопку выбора , чтобы появилось окно *Назначение Процесса (Process Assignment)*.

В нём отображен список процессов (предварительно определённых установками). Отметьте те процессы, на которые необходимо назначить текущий график. И тогда при инициировании процесса пользователь сможет выбрать (контекстное меню – изменить сетевой график) среди предложенных графиков данный. Кроме того, установив флажок **Set as default for all checked processes**, вы делаете текущий график используемым по умолчанию для данного бизнес-процесса.

## **Приложение 1**

### ***Входные параметры для элементов Activity функциональной модели***

- Execution time
- Waiting time
- Resting time
- Transport time
- Costs
- EDP transaction costs
- EDP batch costs
- Print costs
- Postal costs
- Continuous execution
- Execution interruptable
- Task stack
- Done by
- Cooperative
- Cooperation mode
- Min. quota of presence
- Average number of participants
- Max. start period
- Priority
- Max. resource waiting time

### ***Входные параметры для элемента начала процесса функциональной модели***

- Quantity
- Time period
- Tolerance waiting time
- Abandon after tolerance waiting time
- Process calendar
- Cost driver
- Cost driver quantity

## ***Входные параметры для элементов Performer организационной модели***

- Hourly wages
- Availability
- Calendar
- Time dependent cost

Для выполнения имитационного моделирования задавать все эти параметры необязательно. Однако при этом пострадает точность выводимых результатов. В таблице 5 приведён минимальный перечень входных данных, необходимый для имитации.

*Таблица 5. Минимальный набор параметров, необходимый для имитационного моделирования*

Параметр элемента	Элемент	Path analysis	Capacity Analysis	Workload Analysis
Quantity	Process start	—	+	—
Execution time	Activity	+	+	+
Waiting time*	Activity	+	+	+
Resting time*	Activity	+	+	+
Transport time*	Activity	+	+	+
Cost *	Activity	+	+	+
Performer	Activity	—	+	+
Hourly wages	Performer	—	+	+
availability	Performer	—	+	—
Calendar	Performer	—	—	+

\* Параметры, помеченные звездочкой, необходимы для получения ненулевых значений при расчете. Однако, если эти поля останутся незаполненными, Adonis все равно выполнит моделирование процесса.

В функциональной модели необходимо задать параметры для стартового элемента (Process start) и каждой функции (Activity). Для элементов Activity эти характеристики задаются в полях закладок Time/cost, Working environment и Other simulation data записной книжки (Notebook). Для элемента Process start заполняются поля закладки Simulation data. В таблице 6 приведено описание всех параметров стартового элемента, а в таблице 7 – для функции.

Таблица 6. Описание параметров элементов *Process start*

Наименование закладки	Simulation data
Наименование параметра	Описание параметра
Quantity	частота возникновения процесса в пределах указанного интервала времени «Time period»
Time period	период времени, в течение которого происходит имитационное моделирование процесса; параметр необходим для выполнения «Capacity Analysis» - определение оптимального состава и структуры отделов
Calendar	позволяет составить календарь, дает возможность назначить профили дня (рабочий, выходной, укороченный) и привязать их к конкретной дате; данный параметр необходим для «Workload Analysis» – определения рабочей загрузки процессов с учетом частоты возникновения процесса и реальной доступности исполнителей (календарное планирование)
Tolerance waiting time	время ожидания между началом процесса и выполнением первой операции после завершения которого, процесс будет отменен
Abandon after tolerance waiting time	если данный признак активирован, то после завершения времени ожидания (Tolerance waiting time) процесс будет прерван; параметр учитывается при выполнении «Workload Analysis»
Cost driver	инфляция
Cost driver quantity	рост инфляции

Таблица 7. Описание параметров элементов Activity

<b>Наименование закладки</b>	<b>Time/cost</b>
<i>Наименование параметра</i>	<i>Описание параметра</i>
Execution time	время фактического выполнения задания
Waiting time	время ожидания перед началом выполнения задания
Resting time	время ожидания транспортировки
Transport time	время транспортировки
Cost	затраты на выполнение данной деятельности
EDP transaction cost	затраты на электронную обработку данных
EDP batch cost	затраты на электронную обработку партии данных (этот признак будет использован при оценке стоимости процесса)
Print cost	затраты на печать
Postal cost	почтовые расходы
<b>Наименование закладки</b>	<b>Working environment</b>
<i>Наименование параметра</i>	<i>Описание параметра</i>
Performer	<p>С помощью этого признака назначают исполнителя на данную операцию (иначе не удастся осуществить анализ). Параметр можно указать в диалоговом режиме с помощью специальной клавиши, находящейся справа вверху над полем. При её активации откроется окно, в котором необходимо выбрать модель рабочей среды, исполнителя (classes), а также механизм его назначения (relation), после чего нажать клавишу Ok. В списке classes можно выбрать как конкретного исполнителя, так и группу специалистов, объединенных одной ролью.</p> <p><i>Примечание: в данной лабораторной работе исполнитель уже изначально задан, поэтому Вам уже не нужно определять данный параметр.</i></p>
Continuous execution	Если признак активирован, то данная деятельность будет выполнена в следующий раз тем же исполнителями, что и в предыдущий. Это задание будет размещено первым в списке на выполнение (во избежание нереалистично большого времени ожидания).

Exceptional interruptible	Если этот признак активирован, выполнение операции прерывается сразу после окончания зарезервированного времени ее исполнения (Execution time). Данное задание будет выполняться данным исполнителем в свободное время.
Task stack (personal/global)	Необходимо выбрать «personal» в том случае, если требуется указать конкретного исполнителя. Задание будет отправлено ему, даже если в данный момент он не доступен. Выбор «global» позволит отправить задание на исполнение только свободному на данный момент исполнителю.
Done by	В этом поле задается имя переменной, в которой при выполнении Workload analysis и Capacity analysis будет сохранено имя исполнителя данной деятельности
<b>Наименование закладки</b>	<b>Other simulation data</b>
<i>Наименование параметра</i>	<i>Описание параметра</i>
Cooperative	Если данный признак активирован, то для выполнения операции может быть выделено несколько исполнителей (например, задание направлено всем свободным исполнителям). При этом задание не будет подано к исполнению до тех пор, пока в системе не будет достаточного количества свободных исполнителей («Average number of participants») или не истечёт установленное время ожидания («Max. start period»).
Cooperation mode	При установке «synchron» задание будет одновременно выполняться всеми назначенными исполнителями. При установке «asynchron» задание будет выполняться исполнителями последовательно (порядок зависит от их занятости).

Average number of participants	<p>В этом поле задается коэффициент поправки времени исполнения задания. Данный атрибут участвует в расчётах только в том случае, если данная операция выполняется несколькими исполнителями (активирован признак «cooperative»). Этот параметр необходим для расчёта оптимального пути исполнения процесса (Path analysis). При этом если для кооперативной операции задан параметр «asynchron», то данный коэффициент будет равен среднему числу исполнителей данной задачи.</p> <p>В случае если задан параметр «synchron», то теоретически выполнение задачи будет завершено только после окончания работы над ней последнего исполнителя, как при параллельных процессах. При расчёте «Path analysis» конкретные исполнители не заданы, поэтому необходимо время реализации данной задачи скорректировать в зависимости от возможных задержек. Это позволит провести более корректный расчёт.</p>
Min. quota of presence	<p>В этом поле указывается минимальное число свободных исполнителей, необходимых для активации задания. В случае если в системе нет достаточного числа свободных специалистов, задание откладывается. Время отсрочки, однако, не должно превышать указанного в поле «Max. start period» минимума.</p>
Max. start period	<p>Минимальное время отсрочки выполнения задания при кооперативной работе, см. « Min. quota of presence».</p>
Priority	<p>Приоритет задания. Все операции выполняются в порядке уменьшения их приоритета. Задания с одинаковым приоритетом исполняются в порядке поступления.</p>
Max. resource waiting time	<p>Максимальное время ожидания, в течение которого исполнитель ждёт освобождения ресурсов, необходимых для реализации задачи. Если это время будет превышено, автоматически запускается выполнение последующей задачи. После чего происходит возврат к пропущенной операции и, снова запускается механизм ожидания ресурсов и так далее.</p>

В организационной модели необходимо определить характеристики только для исполнителей (Performer). Эти параметры задаются в полях записной книжки закладки Simulation data. В таблице 8 приведено описание этих характеристик.

Таблица 8. Описание параметров элементов Performer

Наименование закладки	Simulation data
Наименование параметра	Описание параметра
Hourly wages	почасовая заработная плата исполнителя
Availability	доступность исполнителя (рассчитывается по формуле $\frac{\text{Число} \cdot \text{раб.} \cdot \text{дн.} \cdot \text{в} \cdot \text{нед} \times \text{Число} \cdot \text{раб.} \cdot \text{часов} \cdot \text{в} \cdot \text{дн.}}{40} \times 100$ ); принимает значения от 1 до 100 (при работе 5 дней в неделю по 8 часов этот параметр будет равен 100); возможно выставить тип «значение по умолчанию», тогда система высчитает значение самостоятельно на основании других введенных данных.
Calendar	позволяет составить календарь, даёт возможность назначить профили дня (рабочий, выходной, укороченный) и привязать их к конкретной дате.
Time dependent cost	затраты времени исполнителем на операцию (рассчитываются при стоимостном анализе).

## Приложение 2. Варианты заданий для самостоятельного выполнения

№ варианта	Имя файла
1	БП_ТПП_001.adl
2	БП_ТПП_002.adl
3	БП_ТПП_003.adl
4	БП_ТПП_004.adl
5	БП_ТПП_005.adl

## Список литературы

1. Буч Г., Рамбо Д., Джекобсон А. Язык UML. Руководство пользователя // М.: ДМК, 2000. 432 с.
2. Зильбербург Л.И., Молочник В.И., Яблочников Е.И. Реинжиниринг и автоматизация технологической подготовки производства в машиностроении. – СПб: Политехника, 2004. – 152 с.
3. Леоненков А.В. UML 2 Самоучитель. // СПб: БХВ-Петербург, 2007.-311 с.
4. Ойхман Е.Г., Попов Э.В. Реинжиниринг бизнеса: реинжиниринг организации и информационные технологии. – М.: Финансы и статистика, 1997.
5. Репин В.В., Елиферов В.Г. Процессный подход к управлению. Моделирование бизнес-процессов. – М.: РИА "Стандарты и качество", 2004.
6. Фомина Ю.Н. Средства визуального моделирования технической подготовки производства / Научно-технический вестник СПбГУ ИТМО. Выпуск 28. I Сессия научной школы «Задачи механики и проблемы точности в приборостроении». – СПб: СПбГУ ИТМО, 2006.
7. Яблочников Е.И. Методологические основы построения АСТПП / СПб: СПбГУ ИТМО, 2005. 84 с.
8. Яблочников Е.И., Фомина Ю.Н., Дмитриев С.А.. Реализация бизнес-процессов в сфере ТПП с использованием WF-диаграмм / Научно-технический вестник СПбГУ ИТМО. Выпуск 33. Технологии управления. – СПб: СПбГУ ИТМО, 2006.
9. Якобсон А., Буч Г., Рамбо Дж. Унифицированный процесс разработки программного обеспечения. // СПб: Питер, 2002.- 496 с.
- 10.Справочная документация SMARTTEAM Editor Administrator Guide
- 11.Справочная документация по системе Adonis.
- 12.<http://khpi-iip.mipk.kharkiv.edu/library/case/leon/>
- 13.[www.boc-eu.com](http://www.boc-eu.com)

В 2007 году СПбГУ ИТМО стал победителем конкурса инновационных образовательных программ вузов России на 2007–2008 годы. Реализация инновационной образовательной программы «Инновационная система подготовки специалистов нового поколения в области информационных и оптических технологий» позволит выйти на качественно новый уровень подготовки выпускников и удовлетворить возрастающий спрос на специалистов в информационной, оптической и других высокотехнологичных отраслях экономики.

---

## **КАФЕДРА ТЕХНОЛОГИИ ПРИБОРОСТРОЕНИЯ**

Кафедра технологии приборостроения относится к числу ведущих кафедр института со дня его основания в 1931 году. Тогда она называлась кафедрой механической технологии и возглавлялась известным ученым в области разработки инструмента профессором А.П. Знаменским. Позже она была переименована в кафедру технологии приборостроения.

За время своего существования кафедра выпустила из стен института более тысячи квалифицированных инженеров, более сотни кандидатов и докторов наук. В разные годы ее возглавляли известные ученые и педагоги профессора Николай Павлович Соболев и Сергей Петрович Митрофанов.

Кафедра имеет выдающиеся научные достижения. Заслуженным деятелем науки и техники РСФСР, профессором С.П. Митрофановым были разработаны научные основы группового производства, за что он был удостоен Ленинской премии СССР. Методы группового производства с успехом применяются в промышленности и постоянно развиваются его учениками. Заслуженным изобретателем Российской Федерации Юрием Григорьевичем Шнейдером разработаны метод и инструментарий нанесения регулярного микрорельефа на функциональной поверхности.

В настоящее время кафедра осуществляет выпуск специалистов по специальностям "Технология приборостроения" (инженер-технолог, инженер-технолог-менеджер, инженер-технолог по искусственному интеллекту в приборостроении) и "Системы автоматизированного проектирования" (инженер-системотехник). На кафедре ведется подготовка бакалавров, магистров, инженеров и аспирантов по названным специализациям силами семи профессоров и девяти доцентов.

Анна Алексеевна Саломатина  
Юлия Николаевна Фомина  
Под ред. к.т.н., доцента Евгения Ивановича Яблочникова

## **Реинжиниринг бизнес-процессов проектирования и производства**

### Приложение II Методические рекомендации по выполнению СРС

В авторской редакции

Зав. редакционно-издательским отделом

Н.Ф. Гусарова

Лицензия ИД № 00408 от 05.11.99

Подписано к печати 08.10.2008

Отпечатано на ризографе

Тираж 100

Заказ № 1259

**Редакционно-издательский отдел**  
Санкт-Петербургского государственного  
университета информационных  
технологий, механики и оптики  
197101, Санкт-Петербург, Кронверкский пр., 49

