

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
**Федеральное государственное бюджетное
образовательное учреждение высшего образования**
**«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ»
(ТУСУР)**

Н.В. ЗАРИКОВСКАЯ

АНАЛИТИЧЕСКИЕ МЕТОДЫ ПРОЕКТИРОВАНИЯ
Учебно-методическое пособие по выполнению практических и
самостоятельных работ для студентов

2018

Зариковская Н.В. Аналитические методы проектирования: учебно-методическое пособие - Томск: Изд-во ТУСУР, 2018. - 35 с.

Методическое пособие для студентов ВУЗов содержит материал для выполнения практических работ и контрольные вопросы по изучению методов проектирования информационных систем.

© Зариковская Н.В. 2018

© Томский государственный университет систем управления и радиоэлектроники (ТУСУР)

СОДЕРЖАНИЕ

Краткое содержание тем и результатов их освоения	4
Введение	6
Практическая работа № 1	7
Практическая работа № 2	9
Практическая работа № 3	12
Практическая работа № 4	17
Практическая работа № 5	25
Указания к самостоятельной работе студентов (СРС) и контрольные вопросы для оценивания.....	33
Список используемой литературы.....	35

Краткое содержание тем и результатов их освоения

Темы практических занятий	Деятельность студента. Решая задачи, студент:
<p>Определение требований и спецификаций на создание информационной системы</p>	<ul style="list-style-type: none"> • <i>изучает</i> основные методы технического проектирования и конструирования; основные законы развития технических систем; • <i>учится</i> использовать современные технические средства в процессе технического проектирования; • <i>овладевает</i> основными методиками проектирования технических процессов и систем.
<p>Построение диаграмм работ информационной системы</p>	<ul style="list-style-type: none"> • <i>знакомится</i> с программным продуктом Enterprise Architect, методологиями функционального моделирования работ; • <i>учится</i> строить серию диаграмм работ для всей информационной системы в целом и для отдельных сценариев работ, отражающих логику и взаимоотношение подразделений (подсистем).
<p>Построение диаграмм потоков данных информационной системы</p>	<ul style="list-style-type: none"> • <i>знакомится</i> с методологией построения диаграмм потоков данных с использованием программного продукта VpWin; • <i>получает навыки</i> работы с программным продуктом Enterprise Architect в части средств работы с диаграммами потоков данных.
<p>Построение диаграмм прецедентов</p>	<ul style="list-style-type: none"> • <i>знакомится</i> с методологией моделирования прецедентов на основе языка UML; • <i>получает навыки</i> построения диаграмм прецедентов для своей предметной области.

Построение диаграмм деятельности	<ul style="list-style-type: none">• <i>знакомится</i> с методологией моделирования деятельности на основе языка UML;• <i>получает навыки</i> построения диаграммы деятельности для каждого прецедента присутствующего на диаграмме прецедентов.
----------------------------------	--

Введение

На начальных этапах создания информационной системы необходимо понять, как работает организация, которую собираются автоматизировать. Никто в организации не знает, как она работает в той мере подробности, которая необходима для создания ИС. Руководитель хорошо знает работу в целом, но не в состоянии вникнуть в детали работы каждого рядового сотрудника. Рядовой сотрудник хорошо знает, что творится на его рабочем месте, но плохо знает, как работают коллеги. Поэтому для описания работы предприятия (или его части) необходимо построить модель, которая бы была адекватна предметной области и содержала сведения обо всех участниках бизнес-процессов.

Виды моделей:

- Методология функционального моделирования работ SADT (Structured Analysis and Design Technique);
- Диаграммы потоков данных DFD (Data Flow Diagrams);
- Методология объектного проектирования на языке UML (UML-диаграммы);
- Модели «Сущность-связь» (ERD - Entity-Relationship Diagrams).

Задачей практических занятий является освоение системного подхода к проектированию информационных систем.

Проектирование информационной систем выполняется с использованием программного продукта Enterprise Architect.

Практическая работа № 1

ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ И СПЕЦИФИКАЦИЙ НА СОЗДАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ

Цель работы: для выбранного варианта информационной системы определить набор требований и спецификаций на создание информационной системы.

Содержание работы и методические указания к ее выполнению

Для успешной реализации базы данных объект проектирования должен быть прежде всего адекватно описан, должны быть построены полные и непротиворечивые функциональные модели базы данных. Опыт проектирования информационных систем показывает, что это логически сложная, трудоемкая и длительная по времени работа, требующая высокой квалификации участвующих в ней специалистов.

При проектировании информационной системы необходимо провести анализ целей этой системы и выявить требования к ней отдельных пользователей. Информация для построения модели информационной системы берется на основе проведения всестороннего обследования организации, для которой выполняется разработка информационной системы. Сбор данных начинается с изучения сущностей предметной области, процессов, использующих эти сущности, и связей между ними.

Последовательность выполнения практической работы:

1. Ознакомиться с предложенным вариантом описания предметной области. Проанализировать предметную область, уточнив и дополнив ее, руководствуясь собственным опытом, консультациями и другими источниками.

2. Выполнить структурное разбиение предметной области на отдельные подразделения (отделы, службы, подсистемы, группы и пр.) согласно выполняемым ими функциям.

3. Определить задачи и функции системы в целом и функции каждого подразделения (подсистемы).

4. Выполнить словесное описание работы каждого подразделения (подсистемы), алгоритмов и сценариев выполнения ими отдельных работ.

5. Оформить отчет со следующими разделами:

- исходное задание;
- расширенное описание предметной области с учетом сделанных дополнений;
- состав подразделений (подсистем) информационной системы;
- перечень функций и задач системы в целом и каждого подразделения (подсистемы) в отдельности, отношения их между собой;
- подробное описание работы каждого подразделения (подсистемы);
- описание отдельных сценариев работ подразделений (подсистем);
- входная и выходная информация для каждого подразделения (подсистемы).

Практическая работа № 2

ПОСТРОЕНИЕ ДИАГРАММ РАБОТ ИНФОРМАЦИОННОЙ СИСТЕМЫ

Цель работы: ознакомление с методологиями функционального моделирования работ.

Содержание работы и методические указания к ее выполнению

Разработка базы данных невозможна без ее тщательного проектирования: слишком велико влияние этого шага на последующие этапы жизненного цикла информационной системы, в основе которой лежит создаваемая база данных.

Для целей проектирования информационной системы могут быть использованы следующие виды моделей:

- методология функционального моделирования работ SADT (Structured Analysis and Design Technique);
- диаграммы потоков данных DFD (Data Flow Diagrams);
- методология объектного проектирования на языке UML (UML-диаграммы).

- Методология SADT (Structured Analysis and Design Technique - технология структурного анализа и проектирования) разработана Дугласом Т. Россом и является одной из самых известных и широко используемых методик проектирования. Новое название методики, принятое в качестве стандарта, - IDEF0 (Icam DEFinition) является частью программы ICAM (Integrated Computer -Aided Manufacturing - интегрированная компьютеризация производства).

Процесс моделирования в SADT включает сбор информации об исследуемой области, документирование полученной информации, представление ее в виде модели и уточнение модели. Кроме того, этот процесс подсказывает вполне определенный путь выполнения согласованной и достоверной структурной декомпозиции, что является ключевым моментом в квалифицированном анализе системы.

В IDEF0 система представляется как совокупность взаимодействующих работ (или функций). Связи между работами определяют технологический процесс или структуру взаимосвязи внутри организации. Модель SADT представляет собой серию диаграмм, разбивающих сложный объект на составные части.

Основными понятиями методологии функционального моделирования работ являются:

Работы (activity) - поименованные процессы, функции или задачи, которые происходят в течение определенного времени и имеют распознаваемые результаты. На диаграмме работы изображаются прямоугольниками.

Вход (Input) - материал или информация, которые используются работой для получения результата (стрелка, входящая в левую грань).

Управление (Control) - правила, стратегии, стандарты, которыми руководствуется работа (стрелка, входящая в верхнюю грань). В отличие от входной информации управление не подлежит изменению.

Выход (Output) - материал или информация, которые производятся работой (стрелка, исходящая из правой грани). Каждая работа должна иметь хотя бы одну стрелку выхода, так как работа без результата не имеет смысла и не должна моделироваться.

Механизм (Mechanism) - ресурсы, которые выполняют работу (персонал, станки, устройства - стрелка, входящая в нижнюю грань).

Вызов (Call) представляет собой взаимодействие одной модели работ с другой (стрелка, исходящая из нижней грани).

Различают в IDEF0 пять типов связей работ.

Связь по входу (input-output) имеет место, когда выход вышестоящей работы направляется на вход следующей работы.

Связь по управлению (output-control) обозначает ситуацию, когда выход вышестоящей работы направляется на управление следующей работы. Связь показывает доминирование вышестоящей работы.

Обратная связь по входу (output-input feedback) имеет место, когда выход нижестоящей работы направляется на вход вышестоящей. Используется для описания циклов.

Обратная связь по управлению (output-control feedback) обозначает ситуацию, когда выход нижестоящей работы направляется на управление вышестоящей. Является показателем эффективности бизнес-процесса.

Связь выход-механизм (output-mechanism) имеет место, когда выход одной работы направляется на механизм другой и показывает, что работа подготавливает ресурсы для проведения другой работы.

Из перечисленных блоков строятся диаграммы работ, описывающие принципы функционирования системы.

Подробнее методология диаграмм работ описана в работе Дэвид А.М, Клемент МакГоуэн «Методология структурного анализа и проектирования (SADT Structured Analysis & Design Technique)»

Выполнение практической работы производится с использованием программного продукта Enterprise Architect

Последовательность выполнения практической работы:

1. Ознакомиться с методологией структурного моделирования работ.
2. Ознакомиться с программным продуктом Enterprise Architect.
3. Построить серию диаграмм работ для всей информационной системы в целом и для отдельных сценариев работ, отражающих логику и взаимоотношение подразделений (подсистем).
4. Оформить отчет.

Практическая работа № 3

ПОСТРОЕНИЕ ДИАГРАММ ПОТОКОВ ДАННЫХ ИНФОРМАЦИОННОЙ СИСТЕМЫ

Цель работы: ознакомление с методологией построения диаграмм потоков данных с использованием программного продукта VpWin.

Содержание работы и методические указания к ее выполнению

Диаграммы потоков данных (Data Flow Diagrams - DFD) используются для описания движения документов и обработки информации как дополнение к IDEF0. В отличие от IDEF0, где система рассматривается как взаимосвязанные работы, стрелки в DFD показывают лишь то, как объекты (включая данные) движутся от одной работы к другой. DFD отражает функциональные зависимости значений, вычисляемых в системе, включая входные значения, выходные значения и внутренние хранилища данных. DFD – это граф, на котором показано движение значений данных от их источников через преобразующие их процессы к их потребителям в других объектах.

DFD содержит процессы, которые преобразуют данные, потоки данных, которые переносят данные, активные объекты, которые производят и потребляют данные, и хранилища данных, которые пассивно хранят данные.

Диаграмма потоков данных содержит:

- процессы, которые преобразуют данные;
- потоки данных, переносящие данные;
- активные объекты, которые производят и потребляют данные;
- хранилища данных, которые пассивно хранят данные.

Процесс DFD преобразует значения данных и изображается в виде эллипса, внутри которого помещается имя процесса.

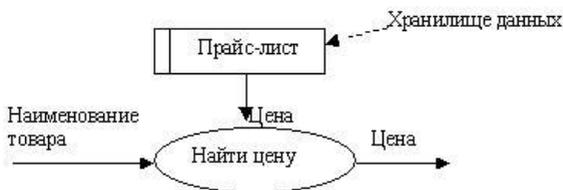


Поток данных соединяет выход объекта (или процесса) с входом другого объекта (или процесса) и представляет собой промежуточные данные вычислений. Поток данных изображается в виде стрелки между производителем и потребителем данных, помеченной именами соответствующих данных. Дуги могут разветвляться или сливаться, что означает соответственно разделение потока данных на части либо слияние объектов.

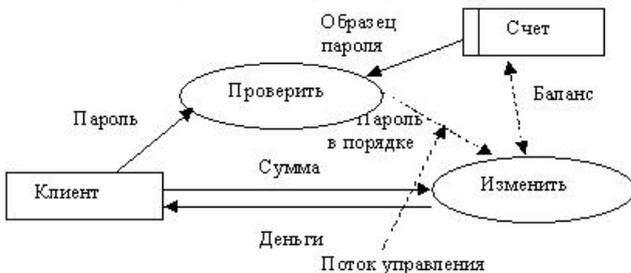
Активным объектом является объект, который обеспечивает движение данных, поставляя или потребляя их. Хранилище данных – это пассивный объект в составе DFD, в котором данные сохраняются для последующего доступа.



Хранилища данных. Хранилище данных – это пассивный объект в составе DFD, в котором данные сохраняются для последующего доступа. Хранилище данных допускает доступ к хранимым в нем данным в порядке, отличном от того, в котором они были туда помещены. Агрегатные хранилища данных, как, например, списки и таблицы, обеспечивают доступ к данным в порядке их поступления, либо по ключам.



Потоки управления. DFD показывает все пути вычисления значений, но не показывает в каком порядке значения вычисляются. Решения о порядке вычислений связаны с управлением программой, которое отражается в динамической модели. Эти решения, вырабатываемые специальными функциями, или предикатами, определяют, будет ли выполнен тот или иной процесс, но при этом не передают процессу никаких данных, так что их включение в функциональную модель необязательно. Тем не менее, иногда бывает полезно включать указанные предикаты в функциональную модель, чтобы в ней были отражены условия выполнения соответствующего процесса. Функция, принимающая решение о запуске процесса, будучи включенной в DFD, порождает в диаграмме поток управления и изображается пунктирной стрелкой.



Первым шагом при построении иерархии DFD является построение контекстных диаграмм. Обычно при проектировании относительно простых информационных систем строится единственная контекстная диаграмма со звездообразной топологией, в центре которой находится так называемый главный процесс, соединенный с приемниками и источниками

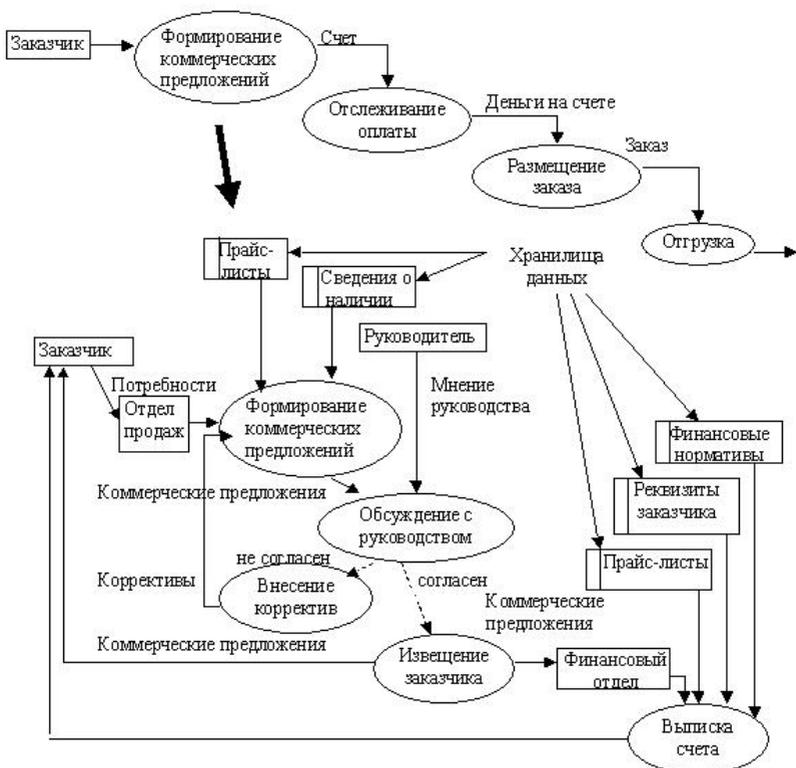
информации, посредством которых с системой взаимодействуют пользователи и другие внешние системы.

Если же для сложной системы ограничиться единственной контекстной диаграммой, то она будет содержать слишком большое количество источников и приемников информации, которые трудно расположить на листе бумаги нормального формата, и, кроме того, главный единственный процесс не раскрывает структуры распределенной системы.

Для сложных информационных систем строится иерархия контекстных диаграмм. При этом контекстная диаграмма верхнего уровня содержит не главный единственный процесс, а набор подсистем, соединенных потоками данных. Контекстные диаграммы следующего уровня детализируют контекст и структуру подсистем.

При построении иерархии DFD переходить к детализации процессов следует только после определения содержания всех потоков и накопителей данных, которое описывается при помощи структур данных. Структуры данных конструируются из элементов данных и могут содержать альтернативы, условные вхождения и итерации. Условное вхождение означает, что данный компонент может отсутствовать в структуре. Альтернатива означает, что в структуру может входить один из перечисленных элементов. Итерация означает вхождение любого числа элементов в указанном диапазоне. Для каждого элемента данных может указываться его тип (непрерывные или дискретные данные). Для непрерывных данных может указываться единица измерения (кг, см и т.п.), диапазон значений, точность представления и форма физического кодирования. Для дискретных данных может указываться таблица допустимых значений.

Ниже приведена диаграмма потоков данных верхнего уровня с ее последующим уточнением:



Выполнение практической работы производится с использованием программного продукта Enterprise Architect

Последовательность выполнения практической работы:

1. Ознакомиться с методологией диаграмм потоков данных.
2. Ознакомиться с программным продуктом Enterprise Architect в части средств работы с диаграммами потоков данных.
3. Построить серию диаграмм потоков данных для отдельных сценариев работ, отражающих логику и взаимоотношение подразделений (подсистем).
4. Оформить отчет.

Практическая работа № 4

ПОСТРОЕНИЕ ДИАГРАММЫ ПРЕЦЕДЕНТОВ

Цель работы: ознакомление с методологией моделирования прецедентов на основе языка UML

Содержание работы и методические указания к ее выполнению

UML (Universal Modeling Language) - универсальный язык моделирования, который был разработан компанией Rational Software с целью создания наиболее оптимального и универсального языка для описания как предметной области, так и конкретной задачи в программировании. Визуальное моделирование в UML можно представить как некоторый процесс поуровневого спуска от наиболее общей и абстрактной концептуальной модели системы к логической, а затем и к физической модели соответствующей системы. Любая задача, таким образом, моделируется при помощи некоторого набора иерархических диаграмм, каждая из которых представляет собой некоторую проекцию системы.

Диаграмма (Diagram) – это графическое представление множества элементов. Чаще всего она изображается в виде связного графа с вершинами (сущностями) и ребрами (отношениями).

В UML определено восемь видов диаграмм:

- диаграмма прецедентов (Use case diagram) - диаграмма поведения, на которой показано множество прецедентов и актеров, а также отношения между ними;
- диаграмма деятельности (Activity diagram) - диаграмма поведения, на которой показан автомат и подчеркнуты переходы потока управления от одной деятельности к другой;
- диаграмма классов (Class diagram) – структурная диаграмма, на которой показано множество классов, интерфейсов, коопераций и отношения между ними;
- диаграмма состояний (Statechart diagram) – диаграмма поведения, на которой показан автомат и подчеркнуто поведение объектов с точки зрения порядка получения событий;

- диаграмма последовательностей (Sequence diagram) – диаграмма поведения, на которой показано взаимодействие и подчеркнута временная последовательность событий;
- диаграмма кооперации (Collaboration diagram) – диаграмма поведения, на которой показано взаимодействие и подчеркнута структурная организация объектов, посылающих и принимающих сообщения;
- диаграмма компонентов (Component diagram) – диаграмма поведения, на которой показан автомат и подчеркнута поведение объектов с точки зрения порядка получения событий
- диаграмма развертывания (Deployment diagram) – структурная диаграмма, на которой показаны узлы и отношения между ними.

Диаграмма прецедентов

Диаграммы прецедентов применяются для моделирования вида системы с точки зрения внешнего наблюдателя. На диаграмме прецедентов графически показана совокупность прецедентов и Субъектов, а также отношения между ними.

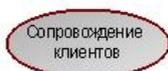
Рассмотрим основные элементы диаграммы прецедентов.

Субъект (actor) - любая сущность, взаимодействующая с системой извне или множество логически связанных ролей, исполняемых при взаимодействии с прецедентами. Стандартным графическим обозначением субъекта на диаграммах является фигурка "человечка", под которой записывается конкретное имя субъекта, однако субъектом может быть не только человек, но и техническое устройство, программа или любая другая система, которая может служить источником воздействия на моделируемую систему так, как определит сам разработчик.



Прецеденты (use case) – это описание множества последовательностей действий (включая их варианты), которые выполняются системой для того, чтобы актер получил результат, имеющий для него определенное значение. При этом ничего не

говорится о том, каким образом будет реализовано взаимодействие субъектов с системой, это одна из важнейших особенностей разработки прецедентов. Стандартным графическим обозначением прецедента на диаграммах является эллипс, внутри которого содержится краткое название прецедента или имя в форме глагола с пояснительными словами.



Сущность концепции прецедентов подразумевает несколько важных пунктов.

- Прецедент представляет собой **завершенный** фрагмент функциональных возможностей (включая основной поток логики управления, его любые вариации (подпотoki) и исключительные условия (альтернативные потоки)).

- Фрагмент **внешне наблюдаемых функций** (отличных от внутренних функций).

- **Ортогональный** фрагмент функциональных возможностей (прецеденты могут при выполнении совместно использовать объекты, но выполнение каждого прецедента независимо от других прецедентов).

- Фрагмент функциональных возможностей, **иницируемый субъектом**. Будучи инициирован, прецедент может взаимодействовать с другим субъектом. При этом возможно, что субъект окажется только на принимающем конце прецедента, опосредованно инициированного другим субъектом.

- Фрагмент функциональных возможностей, который предоставляет субъекту **ощутимый полезный результат** (и этот результат достигается в пределах одного прецедента).

Между субъектами и прецедентами – основными компонентами диаграммы прецедентов – могут существовать различные отношения, которые описывают взаимодействие экземпляров одних субъектов и прецедентов с экземплярами других субъектов и прецедентов. В языке UML имеется несколько стандартных видов отношений между субъектами и прецедентами:

Отношение ассоциации (association) – определяет наличие канала связи между экземплярами субъекта и прецедента (или между экземплярами двух субъектов). Обозначается сплошной линией, возможно наличие стрелки и указание мощности связи.

Отношение расширения (extend) – определяет взаимосвязь экземпляров отдельного прецедента с более общим прецедентом, свойства которого определяются на основе способа совместного объединения данных экземпляров. Обозначается пунктирной линией со стрелкой, направленной от того прецедента, который является расширением для исходного прецедента, и помечается ключевым словом «extend» («расширяет»).

Отношение включения (include) – указывает, что некоторое заданное поведение для одного прецедента включает в качестве составного компонента поведение другого прецедента. Данное отношение является направленным бинарным отношением в том смысле, что пара экземпляров прецедентов всегда упорядочена в отношении включения. Обозначается пунктирной линией со стрелкой, направленной от базового прецедента к включаемому, и помечается ключевым словом «include» («включает»).

Отношение обобщения (generalization) – служит для указания того факта, что некоторый прецедент А может быть обобщен до прецедента В. В этом случае прецедент А будет являться специализацией прецедента В. При этом В называется предком или родителем по отношению к А, а прецедент А - потомком по отношению к прецеденту В. Следует подчеркнуть, что потомок наследует все свойства и поведение своего родителя, а также может быть дополнен новыми свойствами и особенностями поведения. Графически данное отношение обозначается сплошной линией со стрелкой в форме незакрашенного треугольника, которая указывает на родительский прецедент.

Пример. Магазин видеопродукции.

Магазин продает видеокассеты, DVD-диски, аудио-кассеты, CD-диски и т.д., а также предлагает широкой публике прокат видеокассет и DVD-дисков.

Товары поставляются несколькими поставщиками. Каждая партия товара предварительно заказывается магазином у некоторого поставщика и доставляется после оплаты счета. Вновь поступивший товар маркируется, заносится в базу данных и затем распределяется в торговый зал или прокат.

Видеоносители выдаются в прокат на срок от 1 до 7 дней. При прокате с клиента взимается залоговая стоимость видеоносителя. При возврате видеоносителя возвращается залоговая стоимость минус сумма за прокат. Если возврат задержан менее чем на 2 дня, взимается штраф в размере суммы за прокат за 1 день* кол-во дней задержки. При задержке возврата более чем на 2 дня - залоговая сумма не возвращается. Клиент может взять одновременно до 4 видеоносителей (прокат-заказ). На каждый видеоноситель оформляется квитанция.

Клиенты могут стать членами видео-клуба и получить пластиковые карточки. С членов клуба не берется залог (за исключением случая, описанного ниже), устанавливается скидка на ставку проката и покупку товаров. Члены клуба могут делать предварительные заказы на подбор видеоматериалов для проката или покупки.

Каждый член клуба имеет некоторый статус. Первоначально – «новичок». При возврате в срок 5 прокат-заказов, статус меняется на «надежный». При задержке хотя бы одного видеоносителя более чем на 2 дня, статус «новичок» или «надежный» меняется на «ненадежный» и клиенту высылается предупреждение. При повторном нарушении правил статус меняется на «нарушитель». Члены клуба со статусом «надежный» могут брать до 8 видеоносителей одновременно, все остальные - 4. С членов клуба со статусом «нарушитель» берется залоговая сумма.

Клиенты при покупке товара или получении видеоносителя в прокат могут расплачиваться наличными или кредитной картой.

Прокатные видеоносители через определенное количество дней проката списываются и утилизируются по акту. Списываются также товары и прокатные видеоносители, у которых обнаружился брак.

На рисунке ниже приведена диаграмма прецедентов для рассматриваемого примера. В этом примере можно выделить следующие субъекты и соответствующие им прецеденты:

- Менеджер – изучает рынок видеопродукции, анализирует продажи (прецедент «Запрос сведений»), работает с поставщиками: составляет заявки на поставки товара (прецедент «Оформление заказа»), оплачивает и принимает товар (прецедент «Прием товара»), списывает товар (прецедент «Списание товара»).

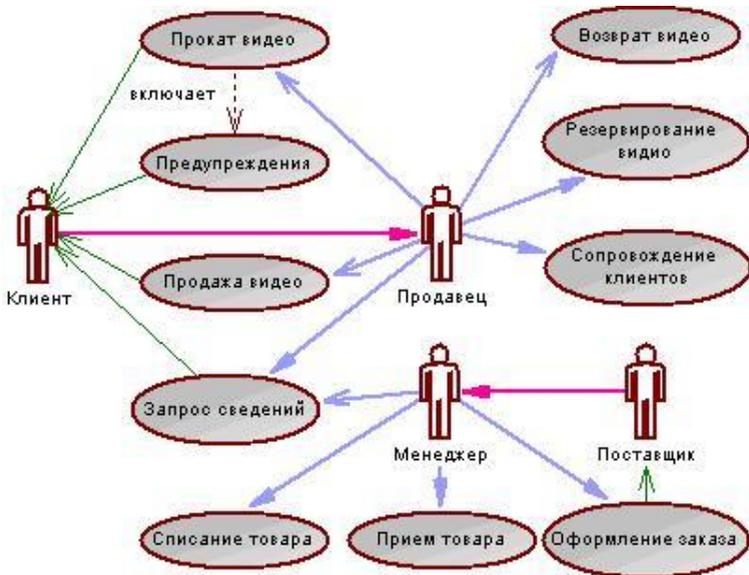
- Продавец – работает с клиентами: продает товар (прецедент «Продажа видео»), оформляет членство в клубе (прецедент «Сопровождение клиентов»), резервирует (прецедент «Резервирование видео»), выдает в прокат (прецедент «Прокат видео») и принимает назад видеоносители (прецедент «Возврат видео»), отвечает на вопросы клиента (прецедент «Запрос сведений»).

- Поставщик – оформляет документы для оплаты товара (прецедент «Оформление заказа»), поставяет товар (прецедент «Прием товара»).

- Клиент – покупает (прецедент «Продажа видео»), берет на прокат и возвращает видеоносители (прецеденты «Прокат видео» и «Возврат видео»), вступает в клуб (прецедент «Сопровождение клиентов»), задает вопросы (прецедент «Запрос сведений»).

Последние два субъекта **Поставщик и Клиент** не будут иметь непосредственного доступа к разрабатываемой системе (второстепенные субъекты), однако именно они являются основным источником событий, инициализирующих прецеденты, и получателями результата работы прецедентов

От прецедента «Прокат видео» к прецеденту «Предупреждения» установлено отношение включения на том основании, что каждый выданный видеоноситель должен быть проверен на своевременный возврат и, в случае необходимости, выдано предупреждение клиенту.



Дальнейшее развитие модели поведения системы предполагает спецификацию прецедентов. Для этого традиционно используют два способа. Первый - описание с помощью текстового документа. Такой документ описывает, что должна делать система, когда субъект инициировал прецедент. Типичное описание содержит следующие разделы:

- Краткое описание
- Участвующие субъекты
- Предусловия, необходимые для инициирования прецедента
 - Поток событий (основной и, возможно, подпотоки, альтернативный)
 - Постусловия, определяющие состояние системы, по достижении которого прецедент завершается.

Выполнение практической работы производится с использованием программного продукта Enterprise Architect.

Последовательность выполнения практической работы:

1. Ознакомиться с методологией моделирования прецедентов на основе языка UML.

2. Ознакомиться с программным продуктом Enterprise Architect.
3. Построить диаграмму прецедентов для своей предметной области.
4. Описать несколько (2-3) прецедентов.
5. Оформить отчет.

Практическая работа № 5

ПОСТРОЕНИЕ ДИАГРАММ ДЕЯТЕЛЬНОСТИ

Цель работы: ознакомление с методологией моделирования деятельности на основе языка UML

Содержание работы и методические указания к ее выполнению

Диаграммы деятельности (Activity diagram), называемые также диаграммами активности или диаграммами видов деятельности, были введены в язык UML сравнительно недавно. Диаграмма деятельности — это, по существу, блок-схема, которая показывает, как поток управления переходит от одной деятельности к другой, при этом внимание фиксируется на результате деятельности. Результат может привести к изменению состояния системы или возвращению некоторого значения.

Диаграмма деятельности отличается от традиционной блок-схемы

- более высоким уровнем абстракции;
- возможностью представления с помощью диаграмм деятельности управления параллельными потоками наряду с последовательным управлением.

Основными направлениями использования диаграмм деятельности являются

- визуализация особенностей реализации операций классов;
- отображение внутрисистемной точки зрения на прецедент.

В последнем случае диаграммы деятельности применяют для описания шагов, которые должна предпринять система после того, как инициирован прецедент

Разработка диаграммы деятельности преследует цели:

- детализировать особенности алгоритмической и логической реализации выполняемых системой операций и прецедентов;
- выделить последовательные и параллельные потоки управления;

- подготовить детальную документацию для взаимодействия разработчиков системы с ее заказчиками и проектировщиками.

Графически диаграмма деятельности представляется в форме графа деятельности, вершинами которого являются состояния действия или состояния деятельности, а дугами - переходы от одного состояния действия/деятельности к другому. Каждая диаграмма деятельности должна иметь единственное начальное и единственное конечное состояния (на практике иногда можно видеть несколько конечных состояний на одной диаграмме, но это одно и то же состояние, изображенное несколько раз для лучшей читабельности диаграммы). Саму диаграмму деятельности принято располагать таким образом, чтобы действия следовали сверху вниз. В этом случае начальное состояние будет изображаться в верхней части диаграммы, а конечное - в ее нижней части.

Рассмотрим основные элементы диаграммы деятельности.

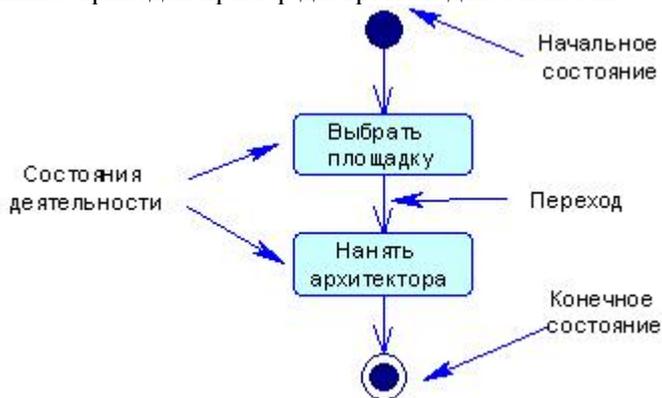
Состояние деятельности (Activity, Process) – это продолжающийся во времени неатомарный шаг вычислений в автомате. Состояния деятельности могут быть подвергнуты дальнейшей декомпозиции, вследствие чего выполняемую деятельность можно представить с помощью других диаграмм деятельности. Состояния деятельности не являются атомарными, то есть могут быть прерваны. Предполагается, что для их завершения требуется заметное время.

Состояния действия (action state) – состояние, которое представляет вычисление атомарного действия, как правило - вызов операции. Состояния действия не могут быть подвергнуты декомпозиции. Они атомарны, то есть внутри них могут происходить различные события, но выполняемая в состоянии действия работа не может быть прервана. И наконец, обычно предполагается, что длительность одного состояния действия занимает неощутимо малое время. Действие может заключаться в вызове другой операции, послышке сигнала, создании или уничтожении объекта либо в простом вычислении - скажем, значения выражения.

Состояния деятельности и состояния действия имеют одинаковое стандартное графическое обозначение – прямоугольник с закругленными краями. Внутри такого символа записывают произвольное выражение (action-expression), которое должно быть уникальным в пределах одной диаграммы деятельности.

Начальное и конечное состояния на диаграммах деятельности изображаются как закрашенный кружок и закрашенный кружок внутри окружности, соответственно.

Ниже приведен пример диаграммы деятельности:



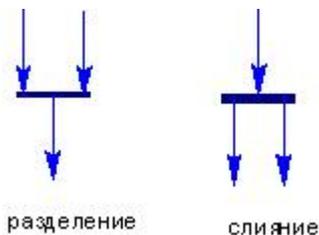
Переход (Transitions) – отношение между двумя состояниями, показывающее, что объект, находящийся в первом состоянии, должен выполнить некоторые действия и перейти во второе состояние. Когда действие или деятельность в некотором состоянии завершается, поток управления сразу переходит в следующее состояние действия или деятельности. Для описания этого потока и используются переходы, показывающие путь из одного состояния действия или деятельности в другое. В UML переход представляется простой линией со стрелкой.

Ветвления. Простые последовательные переходы встречаются наиболее часто, но их одних недостаточно для моделирования любого потока управления. Как и в блок-схему, в диаграмму деятельности может быть включено *ветвление или множественный переход со сторожевыми условиями*. Ветвление

описывает различные пути выполнения в зависимости от значения некоторого булевского выражения. Графически точка ветвления представляется ромбом. В точку ветвления может входить ровно один переход, а выходить - два или более. Для каждого исходящего перехода задается булевское выражение, которое вычисляется только один раз при входе в точку ветвления. Ни для каких двух исходящих переходов сторожевые условия не должны одновременно принимать значение «истина», иначе поток управления окажется неоднозначным. Но эти условия должны покрывать все возможные варианты, иначе поток остановится.

Разделения и слияния. Простые и ветвящиеся последовательные переходы в диаграммах деятельности используются чаще всего. Однако часто возникает потребность изображения параллельных потоков, и это особенно характерно для моделирования бизнес-процессов. В UML для обозначения разделения и слияния таких параллельных потоков выполнения используется синхронизационная черта, которая рисуется в виде жирной вертикальной или горизонтальной линии. При этом *разделение (concurrent fork)* имеет один входящий переход и несколько выходящих, *слияние (concurrent join)*, наоборот, имеет несколько входящих переходов и один выходящий.

Пример разделения и слияния потоков приведен ниже:

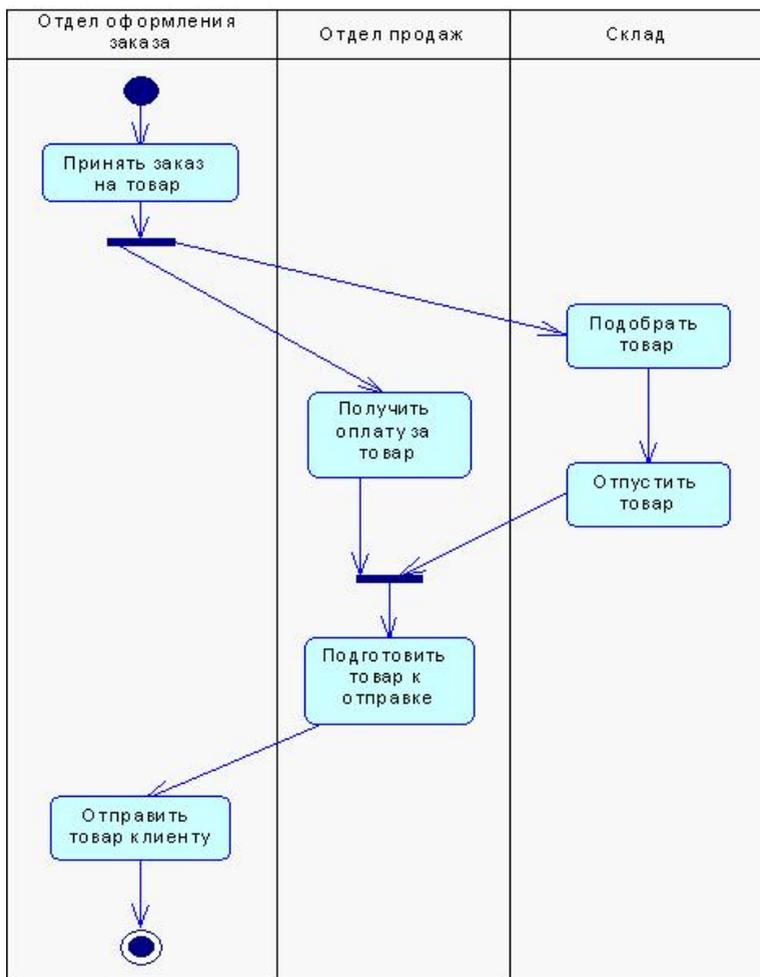


Дорожки. При моделировании течения бизнес-процессов иногда бывает полезно разбить состояния деятельности на диаграммах деятельности на группы, каждая из которых представляет отдел компании, отвечающий за ту или иную работу. В UML такие группы называются дорожками (Swimlanes), поскольку визуалью каждая группа отделяется от

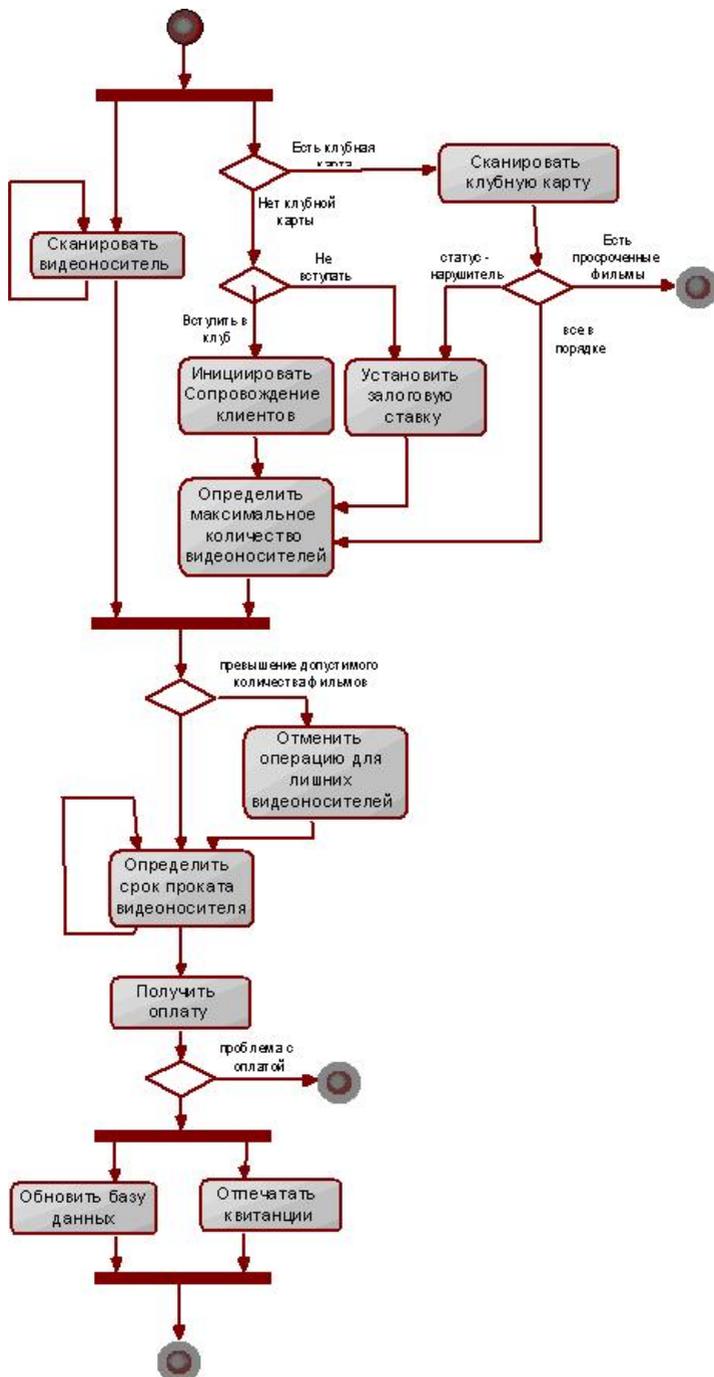
соседней вертикальной черты, как плавательные дорожки в бассейне. Дорожки – это разновидность пакетов, описывающие связанную совокупность работ.

Каждой присутствующей на диаграмме дорожке присваивается уникальное имя. Никакой глубокой семантики дорожка не несет, разве что может отражать некоторую сущность реального мира. Каждая дорожка представляет сферу ответственности за часть всей работы, изображенной на диаграмме. На диаграмме деятельности, разбитой на дорожки, каждая деятельность принадлежит ровно одной дорожке, но переходы могут пересекать границы дорожек.

Имеется некоторая связь между дорожками и параллельными потоками выполнения. Концептуально деятельность внутри каждой дорожки обычно - но не всегда - рассматривается отдельно от деятельности в соседних дорожках. Это разумно, поскольку в реальном мире подразделения организации, представленные дорожками, как правило, независимы и функционируют параллельно.



На следующем рисунке приведена диаграмма деятельности прецедента «Прокат видео» для рассмотренного в практической работе 4 примера «Магазин видеопродукции».



Выполнение практической работы производится с использованием программного продукта Enterprise Architect.

Последовательность выполнения практической работы:

1. Ознакомиться с методологией моделирования деятельности на основе языка UML.
2. Построить диаграммы деятельности для каждого прецедента присутствующего на диаграмме прецедентов, разработанной при выполнении практической работы 4.
3. Оформить отчет.

Указания к самостоятельной работе студентов (СРС) и контрольные вопросы для оценивания

Вид самостоятельной работы:

1. Выполнение индивидуальных заданий.
2. Текущая проработка теоретического материала учебников и лекций, в том числе тем, вынесенных для самостоятельного изучения.

Изучение программы курса:

На лекциях преподаватель рассматривает вопросы программы курса, составленной в соответствии с государственным образовательным стандартом. Из-за недостаточного количества аудиторных часов некоторые темы не удается осветить в полном объеме, поэтому преподаватель, по своему усмотрению, некоторые вопросы выносит на самостоятельную работу студентов, рекомендуя ту или иную литературу.

Кроме этого, для лучшего освоения материала и систематизации знаний по дисциплине, необходимо постоянно разбирать материалы лекций по конспектам и учебным пособиям. В случае необходимости обращаться к преподавателю за консультацией.

Контрольные вопросы

1. Информационные системы. Основные понятия. Классификация.
2. Жизненный цикл ИС. Процессы, стадии, модели.
3. Методы и технологии проектирования ИС.
4. Средства проектирования ИС.
5. Подходы к проектированию ИС (структурно-ориентированный и объектно-ориентированный)
6. Каноническое проектирование ИС. Стадии и этапы процесса канонического
7. Проектирования ИС.

8. Методологии проектирования и их автоматизация. Определение CASE-технологий.
9. CASE-технологии: определение, классификация и эволюция.
10. Сущность системного подхода к проектированию ИС.
11. Моделирование как основа проектирования ИС. Типология моделей.
12. Представление ИС как иерархической многоуровневой системы.
13. Иерархия моделей ИС и сетей.
14. Основные фазы проектирования ИС: системный анализ ИС, проектирование АИС в целом, проектирование БД, концептуальное проектирование БД, логическое и физическое проектирование банка данных, разработка лингвистического и математического обеспечения АИС.

Список использованной литературы

1. Гвоздева, Татьяна Вадимовна Проектирование информационных систем: учебное пособие для вузов по специальности «Прикладная информатика»: рек. УМО вузов РФ / Т. В. Гвоздева, Б. А. Баллод. - Москва: Феникс, 2009. - 508 с.: ил. - (Высшее образование). - Библиогр.: с. 497-503. - ISBN 978-5-222-14075-8.

2. Буч Г., Рамбо Д., Джекобсон А. Язык UML. Руководство пользователя: Пер. с англ. - М.: ДМК Пресс, 2001.

3. Леоненков А.В. Самоучитель UML. - СПб.: БХВ-Петербург, 2001.