

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ
Государственное образовательное учреждение высшего
профессионального образования
«Казанский национальный исследовательский технический
университет им. А.Н.Туполева»

Кафедра автоматизированных систем обработки информации и
управления

С.А. ЗАРАЙСКИЙ

Методические указания к лабораторным работам по дисциплине
«Инструментальные средства информационных систем» (Б1.В.16)

Казань – 2022

Лабораторная работа №1

Диаграммы вариантов использования (Use Case)

Цели работы:

1. Изучить основные сведения о диаграммах вариантов использования в языке UML.
2. Выполнить построение диаграммы вариантов использования в любом графическом редакторе для индивидуального задания.

Теоретические положения

Задача формирования требований к информационным системам (ИС) – одна из наиболее ответственных, трудно формализуемых и наиболее дорогих и тяжелых для исправления в случае ошибки. Современные инструментальные средства и программные продукты позволяют достаточно быстро создавать ИС по готовым требованиям. Но зачастую эти системы не удовлетворяют заказчиков, требуют многочисленных доработок, что приводит к резкому удорожанию фактической стоимости ИС. Основной причиной такого положения является неправильное, неточное или неполное определение требований к ИС на этапе анализа. Все эти обстоятельства привели к появлению специальной методологии, получившей название методологии объектно-ориентированного анализа и проектирования (ООАП).

Объектно-ориентированный анализ и проектирование. Язык UML.

Важным достижением методологии объектно – ориентированного программирования (ООП) явилось осознание того, что процесс написания программного кода может быть отделен от процесса проектирования структуры программы. Прежде, чем начать программирование классов, их свойств и методов, необходимо определить сами эти классы. Более того, нужно дать ответы на следующие вопросы: сколько и какие классы нужно определить для решения поставленной задачи, какие свойства и методы необходимы для придания классам требуемого поведения, а также установить взаимосвязи между классами. Эта совокупность задач не столько связана с написанием кода, сколько с общим анализом требований к будущей программе, а также с анализом конкретной предметной области, для которой разрабатывается программа. Все эти обстоятельства привели к появлению специальной методологии, получившей название методологии объектно-ориентированного анализа и проектирования (ООАП).

Объектно-ориентированный анализ и проектирование (Object-Oriented Analysis/ Design, OOA/D) – технология разработки программных систем, в основу которых положена объектно-ориентированная методология представления предметной области в виде объектов, являющихся экземплярами соответствующих классов. Методология ООАП тесно связана с концепцией автоматизированной разработки программного обеспечения (Computer Aided Software Engineering, CASE).

Унифицированный язык моделирования (Unified Modeling Language, UML) является графическим языком для визуализации, специфицирования, конструирования и документирования систем, в которых большая роль принадлежит, программному обеспечению. С помощью UML можно разработать детальный план создаваемой системы, содержащий не только ее концептуальные элементы (системные функции, бизнес - процессы), но и конкретные особенности (например классы, написанные на специальных языках программирования, схемы баз данных, программные компоненты многократного использования). Полной описание языка можно найти на сайтах www.omg.org и www.rational.com.

Создатели UML представляют его как язык для определения, представления, проектирования и документирования программных систем, организационно-экономических систем, технических систем и других систем различной природы. UML содержит стандартный набор диаграмм (всего 13 типов диаграмм) и нотаций самых разнообразных видов, например:

- **диаграммы вариантов использования (use case diagrams)** – для моделирования бизнес-процессов организации и требований к создаваемой системе);
- **диаграммы классов (class diagrams)** – для моделирования статической структуры классов системы и связей между ними;
- **диаграммы поведения системы (behavior diagrams)**, включающие в себя следующие виды диаграмм – (диаграммы взаимодействия (interaction diagrams),
- **диаграммы последовательности (sequence diagrams)** и **кооперативные диаграммы (collaboration diagrams)** – для моделирования процесса обмена сообщениями между объектами;
- **диаграммы состояний (statechart diagrams)** – для моделирования поведения объектов системы при переходе из одного состояния в другое;
- **диаграммы деятельности (activity diagrams)** – для моделирования поведения системы в рамках различных вариантов использования, или моделирования деятельности);
- **диаграммы реализации (implementation diagrams)**, включающий набор диаграмм (**диаграммы компонентов (component diagrams)** – для моделирования иерархии компонентов (подсистем) системы;
- **диаграммы размещения (deployment diagrams)** – для моделирования физической архитектуры системы).

Диаграммы вариантов использования (Use Case)

Понятие варианта использования или прецедента (Use Case) является основным элементом разработки и планирования проекта. Вариант использования представляет собой последовательность действий (транзакций), выполняемых системой в ответ на событие, инициируемое некоторым внешним объектом (действующим лицом).

Назначение диаграммы вариантов использования:

- определить общие границы функциональности проектируемой системы в контексте моделируемой предметной области;
- специфицировать требования к функциональному поведению проектируемой системы в форме вариантов использования;
- определить исходную концептуальную модель системы для ее последующей детализации в форме логических и физических моделей;
- подготовить исходную документацию для взаимодействия разработчиков системы с ее заказчиками и пользователями.

Основные обозначения на диаграмме вариантов использования представлены на рис. 1.

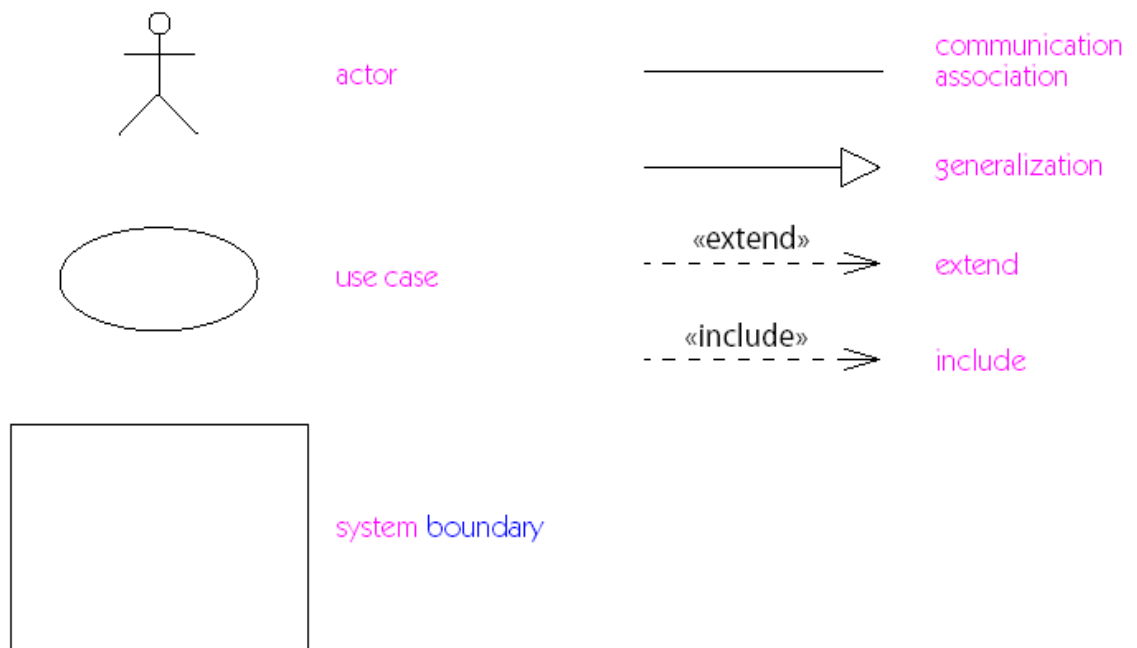


Рис. 1. Основные обозначения на диаграмме вариантов использования

Вариант использования (use case) – представляет собой общую спецификацию совокупности выполняемых системой действий с целью предоставления некоторого наблюдаемого результата, который имеет значение для одного или нескольких актеров

Отвечает на вопрос «Что должна выполнять система?», не отвечая на вопрос «Как она должна выполнять это?»

Имена – отглагольное существительное или глагол в неопределенной форме (рис. 2).

Проверка состояния текущего счета клиента

Рис. 2 Пример названия варианта использования.

Актор (actor) – любая внешняя по отношению к проектируемой системе сущность, которая взаимодействует с системой и использует ее функциональные возможности для достижения определенных целей или решения частных задач. Это может быть внешняя система, которая может произвести запрос к системе. Инициатором выполнения какого-либо действия всегда является актор.

Примеры актеров: кассир, клиент банка, банковский служащий, президент, продавец магазина, менеджер отдела продаж, пассажир авиарейса, водитель автомобиля, администратор гостиницы, сотовый телефон, банкомат, внешняя по отношению к проектируемой системе система управления и т.д.



Рис.3. Примеры актеров.

Вопросы для идентификации актеров в системе:

- какие организации или лица будут использовать систему;
- кто будет получать пользу от использования системы;
- кто будет использовать информацию от системы;
- будет ли использовать система внешние ресурсы;
- может ли один пользователь играть несколько ролей при взаимодействии с системой;
- могут ли различные пользователи играть одну роль при взаимодействии с системой;
- будет ли система взаимодействовать с законодательными, исполнительными, налоговыми или другими органами.

В языке UML на диаграммах вариантов использования поддерживается несколько типов связей между элементами диаграммы. Это связи **коммуникации (communication)**, **включения (include)**, **расширения (extend)** и **обобщения (generalization)**.

Связь коммуникации (communication) – это связь между вариантом использования и действующим лицом. На языке UML связи коммуникации показывают с помощью однонаправленной ассоциации (сплошной линии со стрелкой). Направление стрелки позволяет понять, кто инициирует коммуникацию.

Связь расширения (extend) применяется при описании изменений в нормальном поведении системы. Она позволяет варианту использования только при необходимости использовать функциональные возможности другого варианта использования. Этот вариант использования может использоваться при определенных условиях (например, предоставление скидки особа только важным клиентам).

Связь включения (include) позволяет варианту использования включать обязательные функциональные возможности другого варианта использования (этот вариант ранее был описан и в данном случае не требуется повторное описание). Например, это может быть операция аутентификации.

На языке UML связи **включения (include)** и **расширения (extend)** показывают в виде зависимостей с соответствующими стереотипами, как показано на рис. 4. Для примера на рис. 4, вариант использования «Аутентифицировать клиента» должен быть описан выше в документации и он обязательно выполняется при выполнении варианта использования (функции проектируемой системы) «Снять деньги со счета».

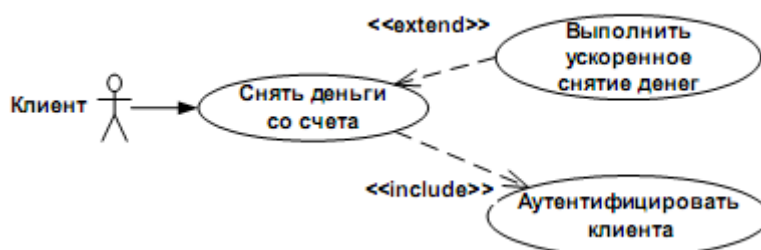


Рис. 4. Связи использования (include) и расширения (extend).

С помощью связи **обобщения (generalization)** показывают, что у нескольких действующих лиц имеются общие черты. Например, клиенты могут быть двух типов: корпоративные и индивидуальные. Эту связь можно моделировать с помощью нотации, показанной на рис. 5.

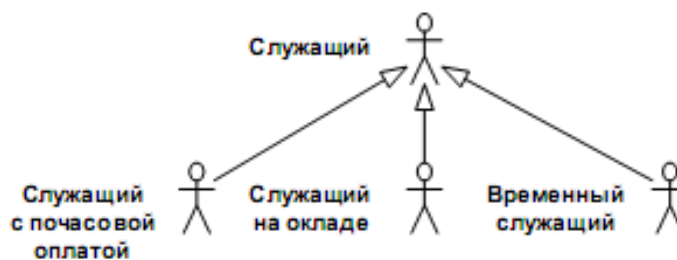


Рис. 5. Обобщение действующего лица

Таким образом, диаграмма вариантов использования иллюстрирует требования к проектируемой системе. Варианты использования всегда нужно анализировать вместе с действующими лицами системы, определяя при этом реальные задачи пользователей и рассматривая альтернативные способы решения этих задач. **Конкретная цель диаграммы вариантов использования – документирование вариантов использования (всё, входящее в сферу применения системы), действующих лиц (всё вне этой сферы) и связей между ними.**

Пример формирования варианта использования для системы «Банкомат» и документирование бизнес-процессов проектируемой системы

Для наглядного представления процесса разработки **диаграммы вариантов использования**, рассмотрим пример разработки программного обеспечения для банкомата (Automated Teller Machine, ATM). На рис. 6 представлен итоговая **диаграмма вариантов использования**. На данной диаграмме человеческие фигурки обозначают действующих лиц, овалы – варианты использования, а линии и стрелки – различные связи между действующими лицами и вариантами использования. На этой диаграмме показаны два действующих лица: клиент и кредитная система. Существует шесть основных действий, выполняемых моделируемой (проектируемой программной системой) системой:

- перевести деньги;
- сделать вклад;
- снять деньги со счета;
- показать баланс;
- изменить идентификационный код;
- осуществить оплату.

Таким образом, **варианты использования – это функции, выполняемые системой**, а действующие лица – это заинтересованные лица (stakeholders) по отношению к создаваемой системе. Такие диаграммы показывают, какие действующие лица инициируют варианты использования. Из них также видно, когда действующее лицо получает информацию от

варианта использования. Данная диаграмма, например, отражает взаимодействие между вариантами использования и действующими лицами системы АТМ (банкома).

В сущности, диаграмма вариантов использования иллюстрирует требования к проектируемой системе со стороны заказчика. Действующими лицами могут быть внешние системы, и потому в данном случае Банковская система показана именно как действующее лицо – она внешняя для системы АТМ. Направленная от варианта использования к действующему лицу стрелка показывает, что вариант использования предоставляет некоторую информацию, используемую действующим лицом. В данном случае вариант использования «Осуществить оплату» предоставляет Банковской системе информацию об оплате по кредитной карточке.



Рис. 6. Диаграмма вариантов использования для системы «Банкомат»

При разработке диаграмм вариантов использования желательно придерживаться следующих правил:

1. Действующие лица находятся вне сферы действия проектируемой системы, а значит и связи между ними моделировать не стоит.
2. Не желательно соединять коммуникационной связью два варианта использования, так как диаграмма должна описывать доступные системе варианты использования, а не порядок их следования.
3. Вариант использования (use case) должен быть инициирован действующим лицом (должна быть сплошная стрелка, идущая от актёра к прецеденту).

Начинать строить диаграмму лучше всего с перечисления всех событий внешнего мира, на которые система должна реагировать.

Более конкретные детали вариантов использования описываются в специальном документе, называемом поток событий (Flow of Events). Его целью является документирование процесса обработки данных в рамках варианта использования.

Поток событий обычно включает (все, что должно быть отражено в проектной документации на все use case):

- краткое описание;
- предусловия (pre-conditions);
- основной поток событий;
- альтернативный поток событий (один или несколько);
- постусловия (post-conditions).

Краткое описание. Каждый прецедент должен иметь связанное с ним короткое описание того, что он будет делать. Например, вариант использования «Перевести деньги» позволяет клиенту или служащему банка переводить с одного счёта на другой.

Предусловия. Предусловия прецедента – это такие условия, которые должны быть выполнены, прежде чем прецедент начнёт выполняться сам. Например, выполнение другого варианта использования, наличие у пользователя прав доступа и т.д.

Основной и альтернативный потоки событий. Поток событий поэтапно описывает, что должно выполняться в варианте использования:

- способ запуска варианта использования;
- различные пути выполнения варианта использования;
- нормальный поток событий;
- отклонения от нормального потока (альтернативные потоки);
- потоки ошибок;
- способ завершения варианта использования.

Пример: поток событий для прецедента «Снять деньги со счёта».

Основной поток событий для банкомата «Снять деньги со счёта»:

1. Вариант использования начинается, когда клиент вставляет карточку в банкомат.

2. Банкомат выводит приветствие и предлагает пользователю ввести пин-код.

3. Клиент вводит пин-код.

4. Банкомат подтверждает введенный пин-код. Если код не подтверждается, выполняется альтернативный поток A1.

5. Банкомат выводит список доступных действий: положить деньги, снять деньги, перевести деньги.

6. Клиент выбирает пункт «Снять деньги».

7. Банкомат запрашивает сумму.

8. Клиент вводит сумму.

9. Банкомат проверяет наличие введенной суммы на счете. Если денег на счете недостаточно, выполняется альтернативный поток A2. При возникновении ошибки выполняется поток ошибки E1.

10. Банкомат вычитает сумму из счета.

11. Банкомат выдает требуемую сумму.

12. Банкомат возвращает клиенту карточку.

13. Банкомат печатает чек.

14. Вариант использования завершается.

Подумайте:

Попробуйте аналогично самостоятельно описать альтернативные потоки A1 и A2, а также поток ошибки E1.

Постусловия. Условия, которые должны быть выполнены после завершения прецедента (посметка флажком переключателя, выполнение другого варианта использования).

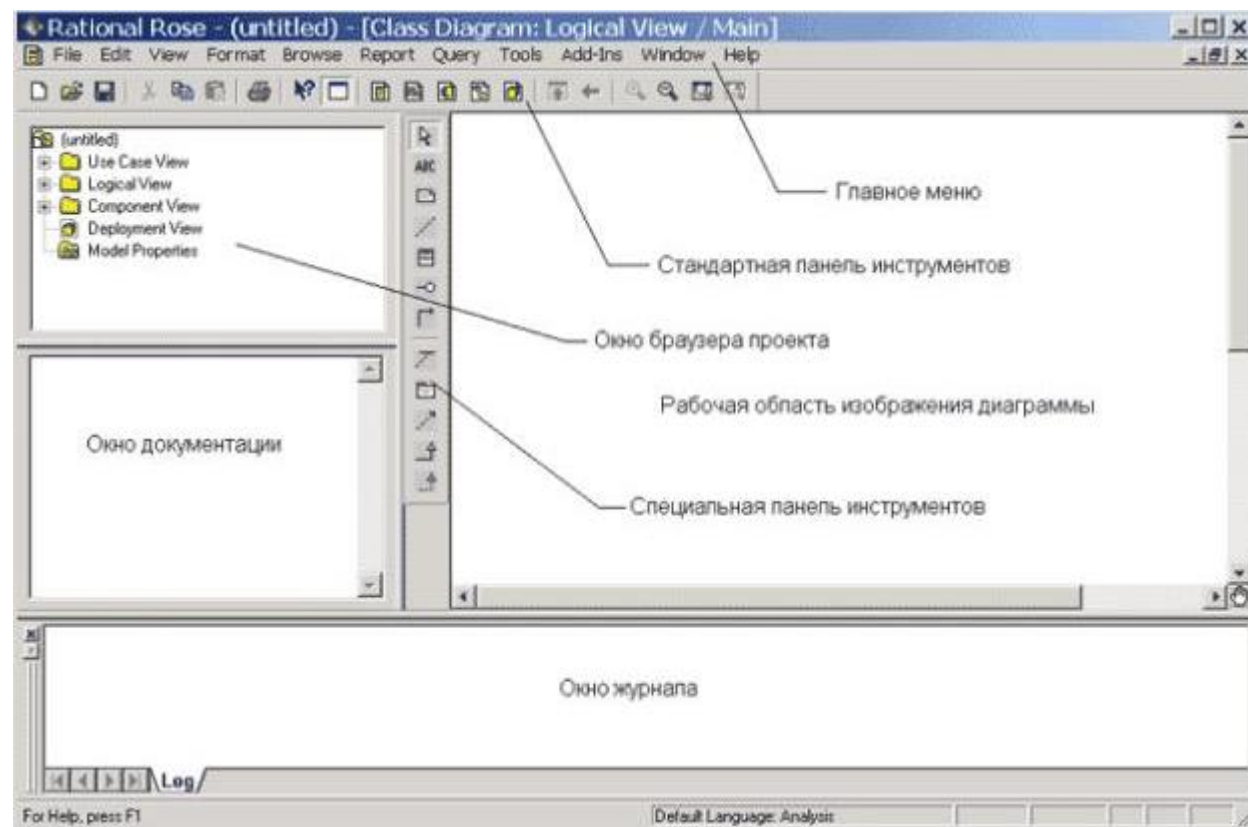


Рис. 7. Панели инструментов Rational Rose

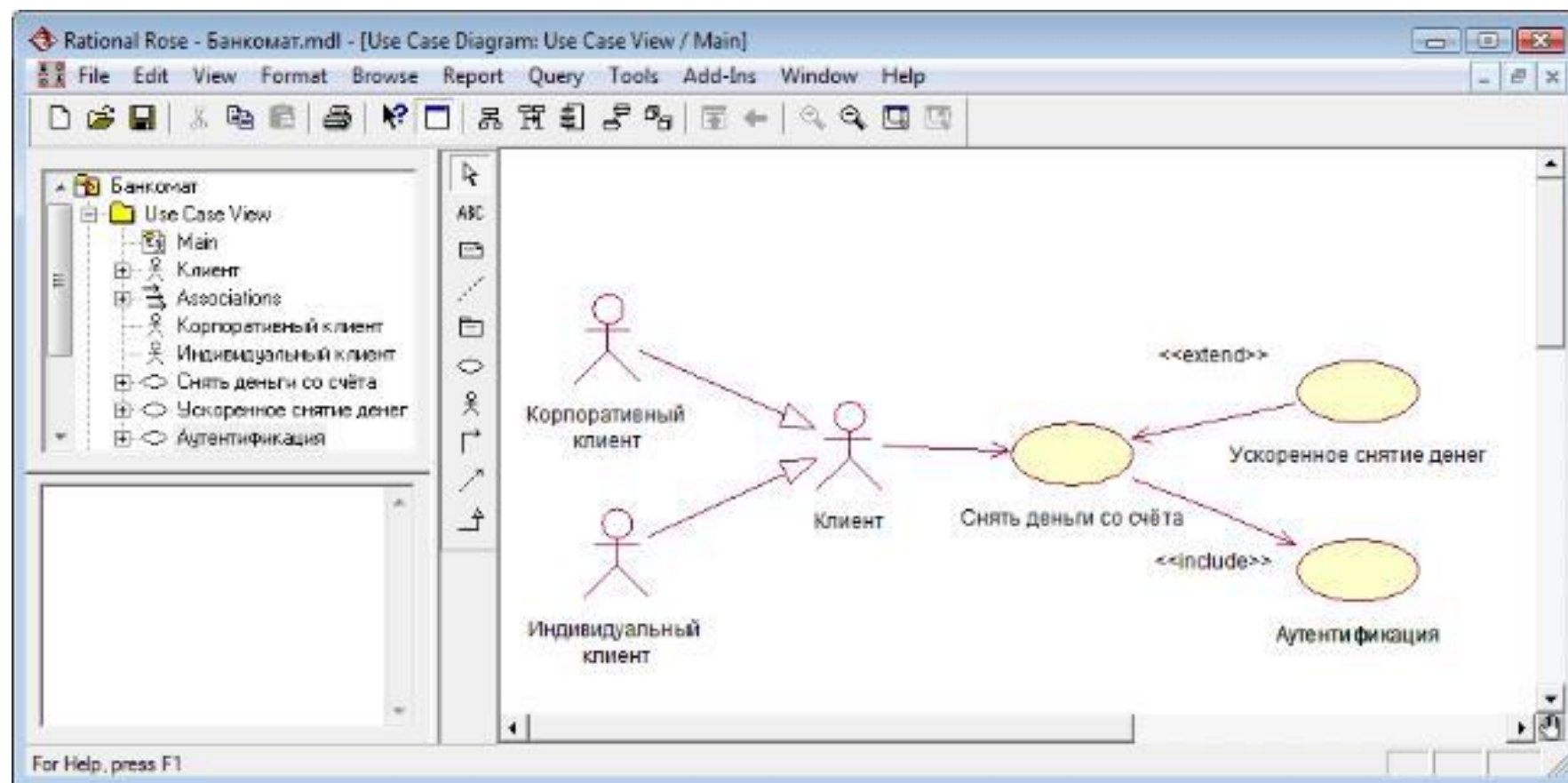


Рис. 8. Пример диаграммы вариантов использования, представленной в Rational Rose.

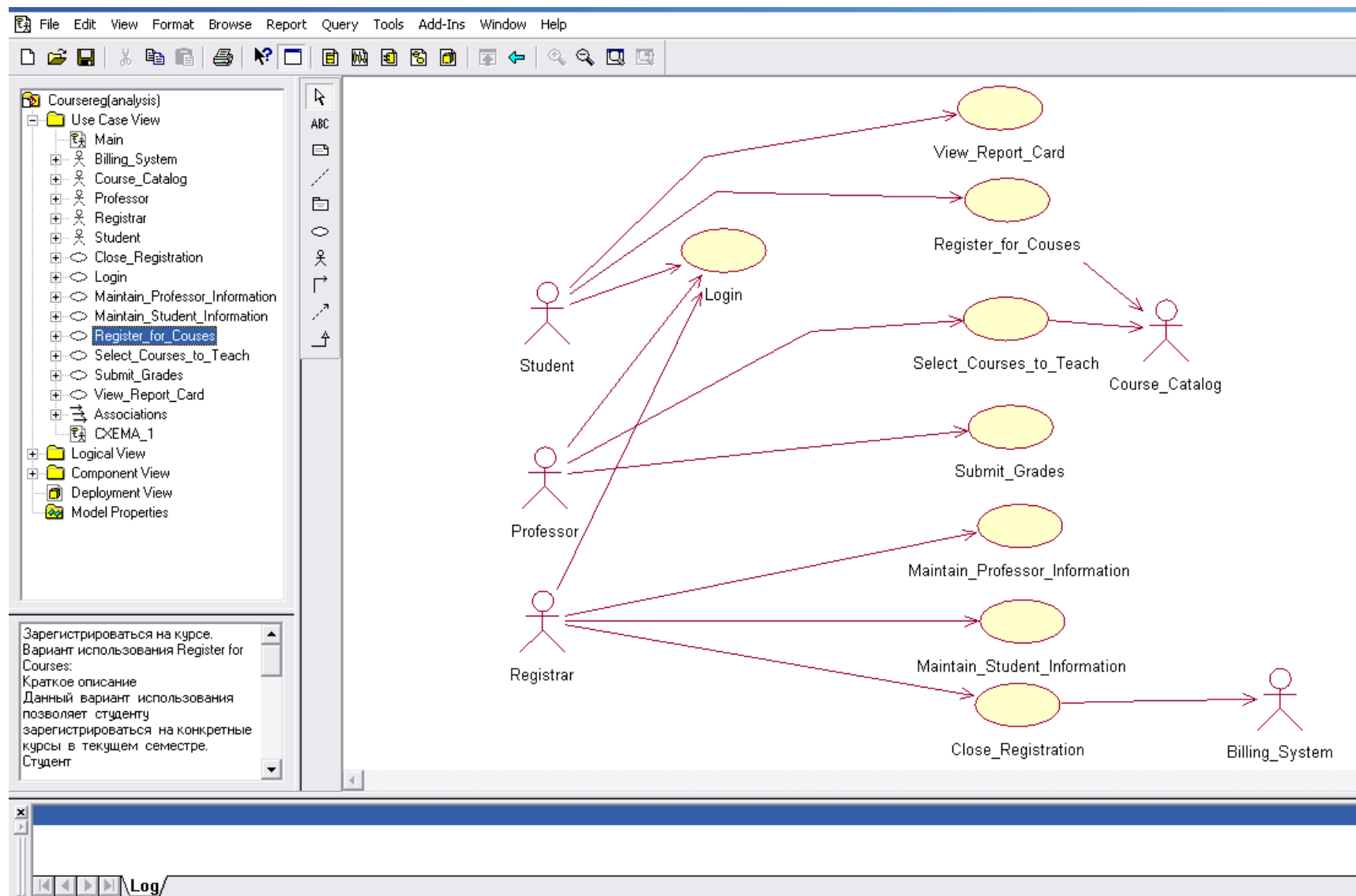


Рис. 9. Пример диаграммы вариантов использования для подсистемы «Регистрация студента на курсы», представленной в Rational Rose.

2. Порядок выполнения лабораторной работы

1. Изучить методические материалы по лабораторной работе.
2. Построить каждому студенту индивидуально диаграмму вариантов использования по своему варианту (вариант задания определяется номером студента в списке преподавателя) в MS Word (MS Visio). Ввести описание бизнес – проекта в соответствующую модель использования.

3. Процесс создания модели должен включать:

1. Составление глоссария проекта и дополнительных спецификаций (т.е. описание всех терминов, например, индивидуальный клиент банка – гражданин РФ, открывший счет в любом филиале банка и имеющий не просроченную, без повреждений свою банковскую карту, помнящий свой пин – код и т.д.).

2. Создание модели вариантов использования.

3. Анализ вариантов использования.

В данной работе Вам предлагается создать представление вариантов использования. При этом должны быть выполнены следующие требования:

- Глоссарий проекта должен иметь форму таблицы и храниться в отдельном файле;
- дополнительные спецификации также хранятся в отдельном файле;
- на диаграммах прецедентов каждый актёр и прецедент должны сопровождаться описанием на русском языке;
- описание актёра должно коротко (1-2строки) сообщать о роли данного лица в системе;
- описание прецедента должно включать пояснения, предусловия, потоки событий, постусловия (см. п.3 кратких теоретических сведений);
- описать основной и альтернативный потоки событий, предусловия и постусловия.

Каждому студенту получить задание на разработку диаграммы вариантов использования (номер задания в приложении А, номер задания соответствует номеру студента в журнале преподавателя). Составить словесное описание бизнес – процесса, составить глоссарий, составить основной и альтернативный потоки событий, предусловия и постусловия. Согласовать диаграммы с преподавателем. Оформить диаграмму в Rational Rose.

5. Оформить лабораторную работу. Отчёт составляется в печатной (на бумаге) форме, каждому студенту. Отчет включает описание всей проделанной работы (интерпретацию постановки задачи, глоссарий проекта, дополнительные спецификации, описание элементов диаграммы вариантов использования) и скриншот полученной диаграммы.

6. Защитить лабораторную работу. Защита работы сопровождается необходимыми пояснениями и подробными и точными ответами на контрольные вопросы из нижеприведённого списка.

Контрольные вопросы.

1. Что такое ООАП (OOA/D)?
2. Что такое UML, для чего он нужен?
3. Какие диаграммы UML отражают статическую структуру проектируемой системы? Какие отражают динамическую?
4. Каковы основные элементы диаграммы взаимодействия?
5. Назначение диаграммы вариантов использования.
6. Что такое вариант использования на UML.
7. Назначение типа связи коммуникация на диаграмме вариантов использования.
8. Назначение типа связи коммуникация на диаграмме вариантов использования. Примеры.
9. Назначение типа связи расширение на диаграмме вариантов использования. Примеры.
10. Назначение типа связи включение на диаграмме вариантов использования. Примеры.
11. Назначение типа связи обобщения на диаграмме вариантов использования. Примеры.
12. Что такое глоссарий?
13. Что такое Краткое описание? Пример.
14. Что такое Предусловие? Пример.
15. Что такое основной потоки событий. Примеры.
16. Что такое альтернативный потоки событий? Примеры.
17. Что такое постусловие? Примеры.
18. Что такое поток событий, из каких элементов он состоит?
19. Почему на диаграмме прецедентов не стоит моделировать связи между актёрами, в каких случаях это всё же будет уместно?

Литература:

1. Вендров А.М. Проектирование программного обеспечения экономических информационных систем. Учебник. М.: Финансы и статистика, 2003, 352 с.
2. Вендров А.М. Практикум по проектированию программного обеспечения экономических информационных систем. М.: Финансы и статистика, 2002, 192 с.

Варианты заданий

1. Цифровой диктофон.
2. Торговый автомат по продаже напитков.
3. Терминал для оплаты за сотовую связь.
4. Диспетчерская служба отдела информационных технологий (ОИТ).
5. ИС Мастерская по ремонту вычислительной техники.
6. Мини-АТС.
7. Проводной телефон.
8. Стиральная машина.
9. Запись в поликлинику через Интернет.
10. Холодильник (2 морозильника).
11. Сейфовый электронный кодовый замок.
12. Обработка заказов от клиентов на перевозки через Интернет (бронирование ТС).
13. Библиотечная система.
14. Интернет-магазин одежды.
15. WWW-конференция (объявление конференции, регистрация участников, обработка материалов от участников, контроль материалов от участников, включение участника в конференцию).
16. Заказ продуктов в магазине через Интернет.
17. Будильник (часы, выбор мелодий, будильник на каждый день недели отдельно, до 10 будильников в день).
18. Телевизор.
19. Домофон.
20. Ячейка камеры автоматической камеры хранения.
20. Лифт.
21. Заказ билетов на междугородние автобусы через Интернет.
22. Заказ книг в Интернете.
23. Заказ лекарств через Интернет.
24. Запись на ТО в Мастерская по ремонту автомашин.