

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования «Казанский национальный исследовательский
технический университет им. А.Н. Туполева-КАИ»
(КНИТУ-КАИ)

Кафедра автоматизированных систем обработки информации и
управления

С.А. ЗАРАЙСКИЙ

**ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА
ИНФОРМАЦИОННЫХ СИСТЕМ**

**Методические указания к лабораторной
работе № 2**

Диаграммы действий (activity diagrams)

Казань – 2021

Цели лабораторной работы

1. Изучить принципы построения диаграмм действий (behavior diagrams).

Теоретические положения

В лабораторной работе №1 ставилась задача изучить основные сведения о диаграммах вариантов использования в языке UML и изучить основные этапы построения вариантов использования в Rational Rose для простого банкомата (АТМ). Будем считать, что построение диаграммы вариантов использования студентами освоено.

Для закрепления пройденного в лабораторной работе №1 материала, рассмотрим еще один пример построения более сложной информационной системы: проект регистрации учебных курсов [1].

Описание регистрации учебных курсов для университета Истерн (Eastern State University - ESU) будет использоваться в качестве основного примера лабораторных работ.

После того как преподаватели ESU решат, какие курсы они будут вести в течение семестра, служба регистрации курсов внесет информацию в компьютерную систему. Затем для преподавателей распечатают сводный отчет по курсам, которые они будут читать, а для студентов - каталог курсов.

На этом этапе студенты заполняют специальную регистрационную форму, где указывают выбранные курсы, и отдают ее в службу регистрации. Обычно студент подписывается на четыре курса, после чего информация заносится в компьютер. Далее запускается ночная пакетная программа, которая распределяет студентов по курсам. При возникновении конфликтной ситуации служба регистрации уточняет студенческие данные. После успешного распределения студенту высылается расписание для проверки. Обычно процесс регистрации на курсы занимает около недели, но в ряде

случаев может потребоваться до двух недель, чтобы уладить все вопросы. Затем преподаватели получают список студентов для каждого курса, который они будут читать.

Риски задачи регистрации курсов. Группа разработчиков определила, что главный риск системы связан с возможностью эффективно сохранять и получать информацию об учебных планах. С этой целью было создано несколько прототипов, чтобы оценить механизмы хранения и доступа к информации для каждой рассматриваемой системы управления базами данных. Результаты испытания прототипов показали, что риск неэффективной работы базы данных может быть уменьшен. Дополнительные прототипы были использованы для оценки аппаратных ресурсов, необходимых при создании онлайн-системы регистрации.

Постановка задачи регистрации курсов. В начале каждого семестра студенты могут запросить каталог курсов, в который включен список учебных предметов, предлагаемых в данном семестре. Информация о курсах должна содержать фамилию преподавателя, название факультета и краткое описание, помогающее студентам сделать выбор.

Новая система позволит студенту выбрать четыре курса из предложенных в наступающем семестре. Кроме того, каждому студенту нужно дополнительно указать еще два варианта, на случай, если курс будет переполнен или отменен. На курс не должно быть записано более десяти или менее трех студентов. Курс, на который запишутся менее трех студентов, будет отменен. По завершении регистрации система регистрации направляет информацию в систему оплаты для выставления счетов студентам. Преподаватели должны иметь возможность онлайн-доступа к системе для указания курсов, которые они будут читать, и для просмотра списка записавшихся студентов.

В каждом семестре выделяется определенное время, в течение которого студенты могут менять свое расписание и получать доступ к системе для добавления или удаления выбранных курсов.

Поведение системы. Поведение разрабатываемой системы (то есть функциональность, обеспечиваемая системой) описывается с помощью функциональной модели, которая отображает системные прецеденты (use cases), системное окружение (действующих лиц или актеров - actors) и связи между прецедентами и актерами (диаграммы прецедентов - use cases diagrams). Основная задача модели прецедентов - представлять собой единое средство, дающее возможность заказчику, конечному пользователю и разработчику совместно обсуждать функциональность и поведение системы.

Разработка модели прецедентов начинается на стадии задумки с выбора актеров и определения общих принципов функционирования системы. Затем на этапе проработки модель дополняется детальной информацией к существующим прецедентам, а при необходимости добавляются новые.

Актеры проектируемой системы.

Напомним, что **актеры** не являются частью системы - они представляют собой кого-то или что-то, что должно взаимодействовать с системой. **Актеры** могут:

- только снабжать информацией систему;
- только получать информацию из системы;
- снабжать информацией и получать информацию из системы.

Обычно актеры определяются из описания задачи или путем переговоров с заказчиками и экспертами. Для выявления актеров может быть использована следующая группа вопросов:

1. Кто заинтересован в определенном системном требовании?
2. Какую роль система будет выполнять в организации?
3. Кто получит преимущества от использования системы?
4. Кто будет снабжать систему информацией, использовать информацию и получать информацию от системы?
5. Кто будет осуществлять поддержку и обслуживание системы?
6. Использует ли система внешние ресурсы?

7. Выступает ли какой-либо участник системы в нескольких ролях?
8. Выступают ли различные участники в одной роли?
9. Будет ли новая система взаимодействовать со старой?

Актеры в системе регистрации курсов университета.

Перечислим ответы на ранее поставленные вопросы:

1. Студент хочет зарегистрироваться на курсы.
2. Преподаватель хочет выбрать курсы, которые он будет читать.
3. Регистратор должен создать учебный план и составить каталог на семестр.
4. Регистратор должен хранить информацию о курсах, преподавателях и студентах.
5. Система оплаты должна получать необходимую информацию из системы регистрации.

Основываясь на полученных ответах, можно выделить следующих **актеров**: студент (Student), преподаватель (Professor), регистратор (Register) и система оплаты (Billing system).

Создать в Rational Rose актеров (рис. 1).

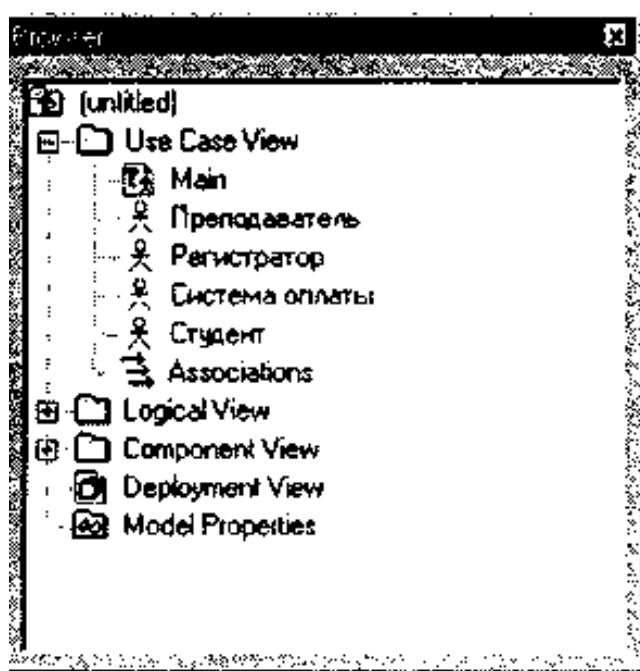


Рис. 1. Перечень актеров системы регистрации студентов на курсы.

В модель желательно включить краткое описание каждого актера, в котором нужно указать роль актера при взаимодействии с системой.

Для системы регистрации курсов описание актеров может быть следующим:

- студент - человек, который регистрируется для посещения занятий в университете;
- преподаватель - человек, который читает лекции в университете;
- регистратор - человек, управляющий системой регистрации курсов;
- система оплаты - внешняя система, выполняющая функции расчетов за курсы.

Описание актеров в программе Rational Rose осуществляется при выполнении следующих действий:

1. Если окна описания нет на экране, откройте его, выбрав команду меню **View => Documentation** (Вид => Описание).
2. Из списка браузера выберите актера, щелкнув по нему мышью.
3. Установите курсор в окне описания и введите текст описания актера.

С помощью **прецедентов** (use cases) моделируется диалог между актером и системой. Другими словами, они определяют возможности, обеспечиваемые системой для актера. Набор всех прецедентов системы определяет способы ее использования. Можно сказать, что прецедент - это последовательность транзакций, выполняемых системой, которая приводит к значимому результату для определенного актера.

Чтобы выделить прецеденты для системы, можно использовать следующую серию вопросов:

1. Каковы задачи каждого актера?
2. Будет ли актер создавать, хранить, изменять, удалять или получать информацию из системы?
3. Какой прецедент будет создавать, хранить, изменять, удалять или получать эту информацию?

Должен ли актер информировать систему о внезапных изменениях внешней среды?

1. Должен ли актер быть проинформирован об изменениях состояния системы?
2. Какие прецеденты будут поддерживать и обслуживать систему?
3. Могут ли все функциональные требования быть реализованы прецедентами?

Прецеденты в системе регистрации курсов университета.

В системе должны обеспечиваться следующие потребности:

- актер студент использует систему для регистрации на курсы;
- по завершении выбора курсов в систему оплаты должна поступить необходимая информация;
- актер преподаватель использует систему для выбора курсов, которые он будет читать в наступающем семестре, и должен получать от системы расписание занятий;
- регистратор отвечает за составление каталога курсов на семестр, за управление информацией об учебных курсах, а также о студентах и преподавателях, работающих с системой.

На основании перечисленных потребностей можно выделить следующие прецеденты:

- регистрация на курсы;
- выбор курсов для преподавания;
- запрос расписания курсов;
- управление информацией о курсах;
- управление информацией о преподавателях;
- управление информацией о студентах;
- создание каталога курсов.

Краткое описание прецедентов

В краткое описание прецедентов вносят информацию об их назначении. Такое описание обычно определяется на этапе задумки при выделении прецедентов для системы.

Для прецедента регистрация на курсах оно будет выглядеть так, как на рис. 3.

Этот прецедент инициируется студентом. Он обеспечивает возможность создавать, изменять, удалять и просматривать расписание студента в определенном семестре.

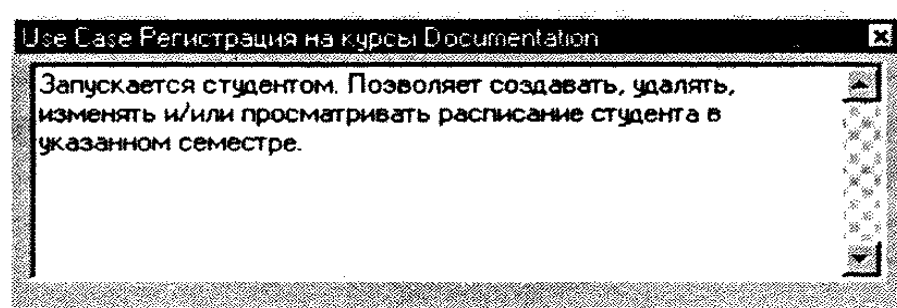


Рис. 3. Краткое описание прецедента Регистрация на курсы

Поток событий (flow of events) для прецедента - это последовательность событий, необходимых для обеспечения требуемого поведения. Поток событий описывается в терминах того, «что» система должна делать, а не «как» она должна это делать. То есть он описывается **на языке предметной области**, а не терминами реализации. Поток событий должен определять:

- когда и как прецедент начинается и заканчивается;
- как он взаимодействует с актером;
- какие данные ему нужны;
- нормальную последовательность событий для прецедента;
- описание потоков в альтернативных и исключительных ситуациях.

Документация на потоки событий обычно составляется в момент

проработки итеративным способом. Сначала дается только краткое описание необходимых шагов для нормального выполнения прецедента. В ходе анализа шаги уточняются. На завершающем этапе в прецедент добавляют потоки для исключительных ситуаций.

В каждом проекте должен использоваться стандартный шаблон для создания документа, описывающего поток событий. Самыми полезными я считаю следующие **шаблоны**:

Х. Поток событий для прецедента <имя>.

Х.1. Предусловия.

Х.2. Главный поток.

Х.3. Подпотоки (если применимы).

Х.4. Альтернативные потоки.

Здесь **Х** - число от единицы до количества прецедентов.

Рассмотрим пример полного документа с описанием потока событий для прецедента выбор курсов для преподавания (Select Courses to Teach).

Поток событий для прецедента «выбор курсов для преподавания»

1.1. Предусловия.

Под-поток создание учебных курсов (Create Course Offerings) прецедента управление информацией о курсах (Maintain Course Information) должен быть выполнен перед его началом.

1.2. Главный поток.

Прецедент начинает выполняться, когда преподаватель подключается к системе регистрации и вводит свой пароль. Система проверяет правильность пароля (E-1) и просит преподавателя выбрать текущий или будущий семестр (E-2). Преподаватель вводит нужный семестр. Система предлагает выбрать требуемую операцию: добавить (Add), удалить (Delete), просмотреть (Review), напечатать (Print) или выйти (Quit).

Если выбрана операция добавить (Add), S-1: выполняется поток добавить учебный курс (Add a Course Offering).

Если выбрана операция удалить (Delete), S-2: выполняется поток

удалить учебный курс (Delete a Course Offering).

Если выбрана операция просмотреть (Review), S-3: выполняется поток просмотреть расписание (Review Schedule).

Если выбрана операция напечатать (Print), S-4: выполняется поток напечатать расписание (Print Schedule).

Если выбрана операция выйти (Quit): прецедент завершается.

1.3. Подпотoki.

S-1: добавить учебный курс (Add a Course Offering)

Система отображает диалоговое окно, содержащее поле для ввода названия и номера предмета. Преподаватель вводит название и номер предмета (E-3). Система отображает список учебных курсов для указанного предмета (E-4). Преподаватель выбирает учебный курс. Система закрепляет за преподавателем выбранный учебный курс (E-5). Затем прецедент начинается сначала.

S-2: удалить учебный курс (Delete a Course Offering).

Система отображает диалоговое окно, содержащее поле для ввода названия и номера учебного курса. Преподаватель выбирает название и номер учебного курса (E-6). Система удаляет взаимосвязь курса с преподавателем (E-7). Затем прецедент начинается сначала.

S-3: просмотреть расписание (Review Schedule)

Система получает (E-8) и отображает следующую информацию для всех учебных курсов, за которыми закреплен данный преподаватель: название предмета, номер предмета, номер учебного курса, день недели, время и место проведения занятий. Когда преподаватель отмечает, что просматривает список, прецедент начинается сначала.

S-4: напечатать расписание (Print Schedule)

Система распечатывает расписание преподавателя (E-9). Прецедент начинается сначала.

1.4. Альтернативные потоки

E-1: введен неверный идентификационный номер преподавателя.

Пользователь должен повторить ввод идентификационного номера или завершить прецедент.

Е-2: введен неверный семестр. Пользователь должен повторить ввод семестра или завершить прецедент.

Е-3: введено неверное название или номер предмета. Пользователь должен повторить ввод названия и номера предмета или завершить прецедент.

Е-4: список учебных курсов не может быть отображен. Пользователю сообщается, что данная команда в настоящий момент недоступна. Прецедент начинается сначала.

Е-5: преподаватель не может быть прикреплен к выбранному учебному курсу. Информация сохраняется, система осуществит прикрепление позже. Выполнение прецедента продолжается.

Е-6: введено неверное название или номер учебного курса. Пользователь должен повторить ввод названия и номера учебного курса или завершить прецедент.

Е-7: система не может удалить связь курса с преподавателем. Информация сохраняется, система удалит связь позже. Выполнение прецедента продолжается.

Е-8: система не может получить информацию о расписании. Прецедент начинается сначала.

Е-9: расписание не может быть распечатано. Пользователю сообщается, что данная опция в данный момент недоступна. Прецедент начинается сначала.

Связанный документ с описанием потока событий показан на рис. 4.

Отношения прецедентов

Между актером и прецедентом может существовать ассоциативное отношение. Такой тип связи часто называют *коммуникативной ассоциацией*

(communicate association), потому что она отражает связь между актером и прецедентом.

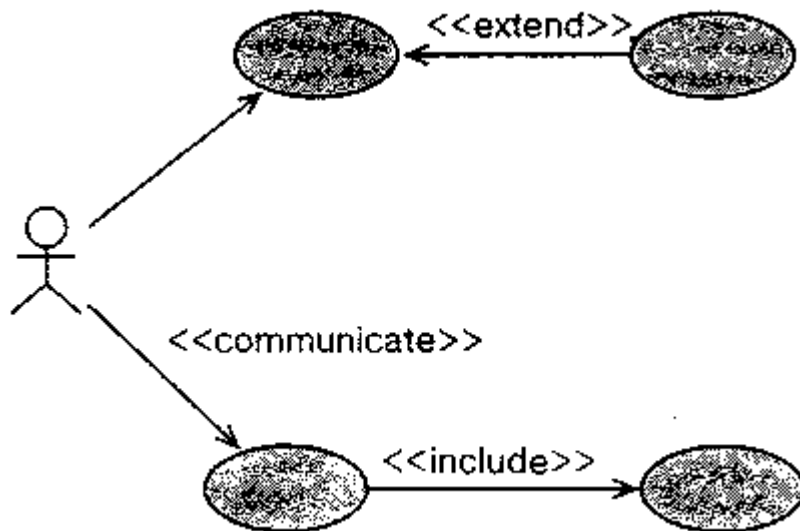


Рис. 4. Пример Use case диаграмм

Диаграммы прецедентов

Диаграмма прецедентов (use case diagram) - это графическое представление всех или части актеров, прецедентов и их взаимодействий в системе. В каждой системе обычно есть главная диаграмма прецедентов, которая отображает границы системы (актеров) и основное функциональное поведение системы (прецеденты). Другие диаграммы прецедентов могут создаваться при необходимости.

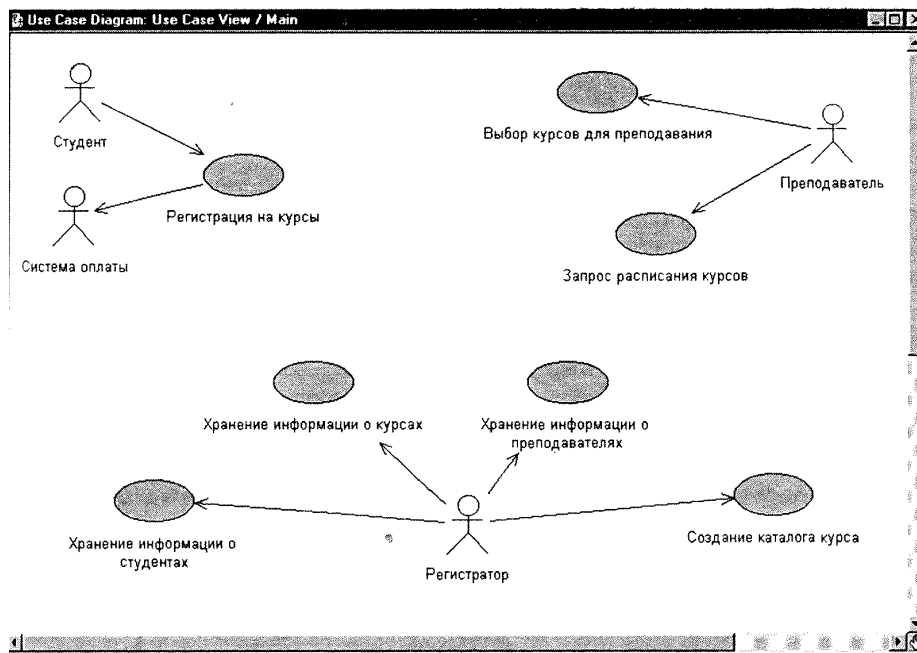


Рис. 5. Главная диаграмма прецедентов системы регистрации курсов университета

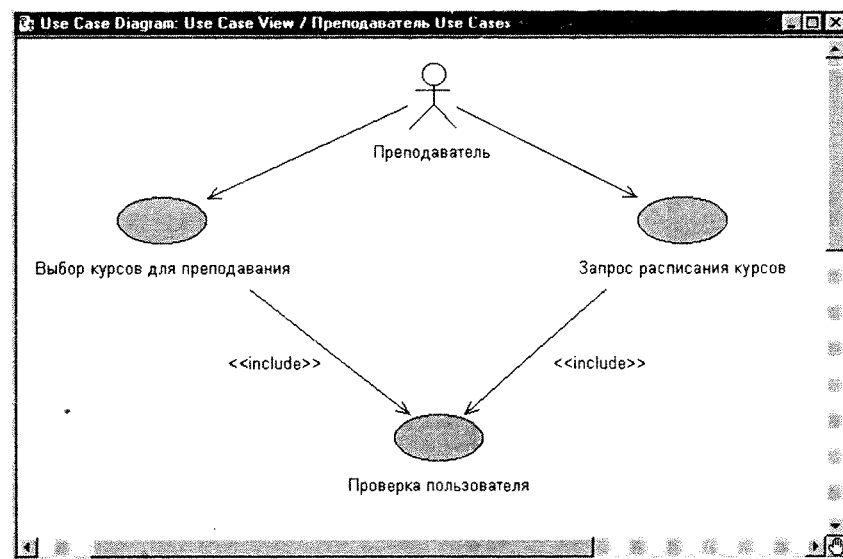


Рис. 6. Дополнительная диаграмма прецедентов системы регистрации курсов университета

Диаграммы действий

После согласования с заказчиком диаграмм Use case производится детальное описание каждого Use case (претендента) в виде последовательности отдельных действий (операций). На данном этапе жизненного цикла проектирования ИС также могут быть построены **диаграммы действий** (activity diagrams). Они отражают динамику проекта и представляют собой схемы потоков управления в системе от действия к действию, а также параллельные действия и альтернативные потоки.

В конкретной точке жизненного цикла диаграммы действий могут представлять потоки между функциями или внутри отдельной функции. На разных этапах жизненного цикла они создаются для отражения последовательности выполнения операции.

Диаграммы действий иллюстрируют действия, переходы между ними, элементы выбора и линии синхронизации. В языке UML действие изображается в виде прямоугольника с закругленными углами, переходы - в виде направленных стрелок, элементы выбора - в виде ромбов, линии синхронизации - в виде толстых горизонтальных или вертикальных линий (см. рис. 7).

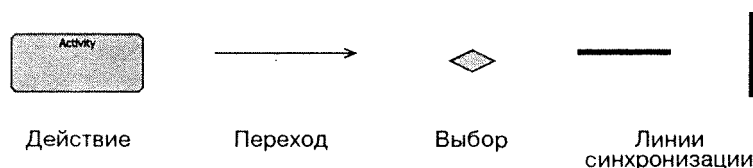


Рис. 7. Нотация языка UML для элементов диаграммы действий

Действия

Действием называется исполнение определенного поведения в потоке управления системы.

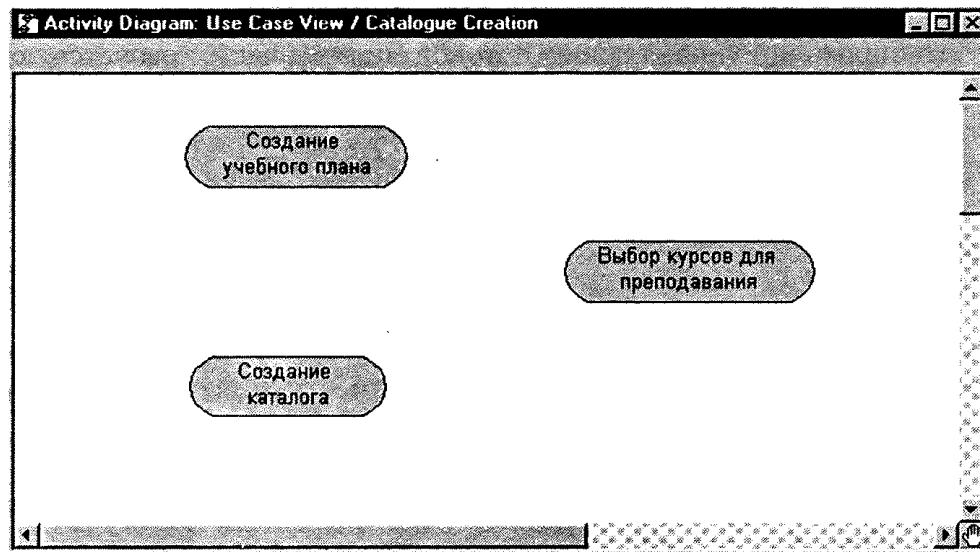


Рис. 9. Действия Use Case Создание каталога

Переходы

Переходы используются для изображения пути потока управления от действия к действию (см. рис. 10). Они обычно осуществляются по завершении очередного действия.

Элементы выбора

При моделировании управляющих потоков системы часто требуется показать места их разделения на основе условного выбора. Переходы из элемента выбора содержат ограничительные условия, определяющие, какое направление перехода будет выбрано. Элементы выбора и условия позволяют задавать альтернативные пути потока управления.

Элемент выбора показан на рис. 10.

Указатель перехода с условием изображен на рис. 12.

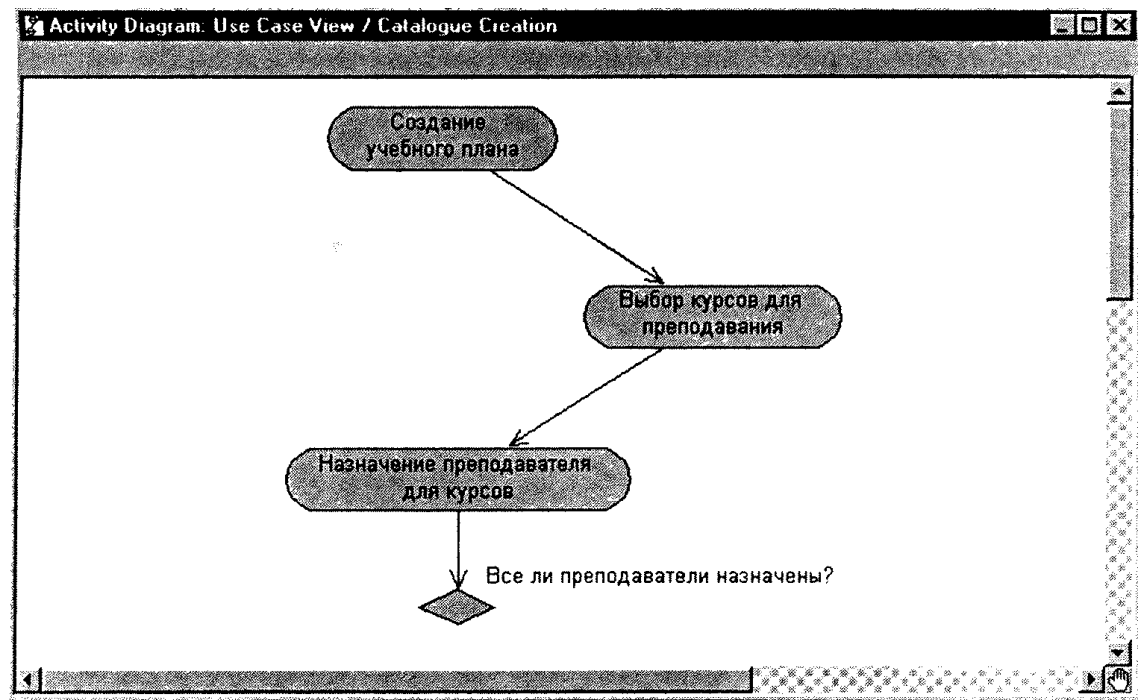


Рис. 10. Элемент выбора действий для Use Case Создание каталога

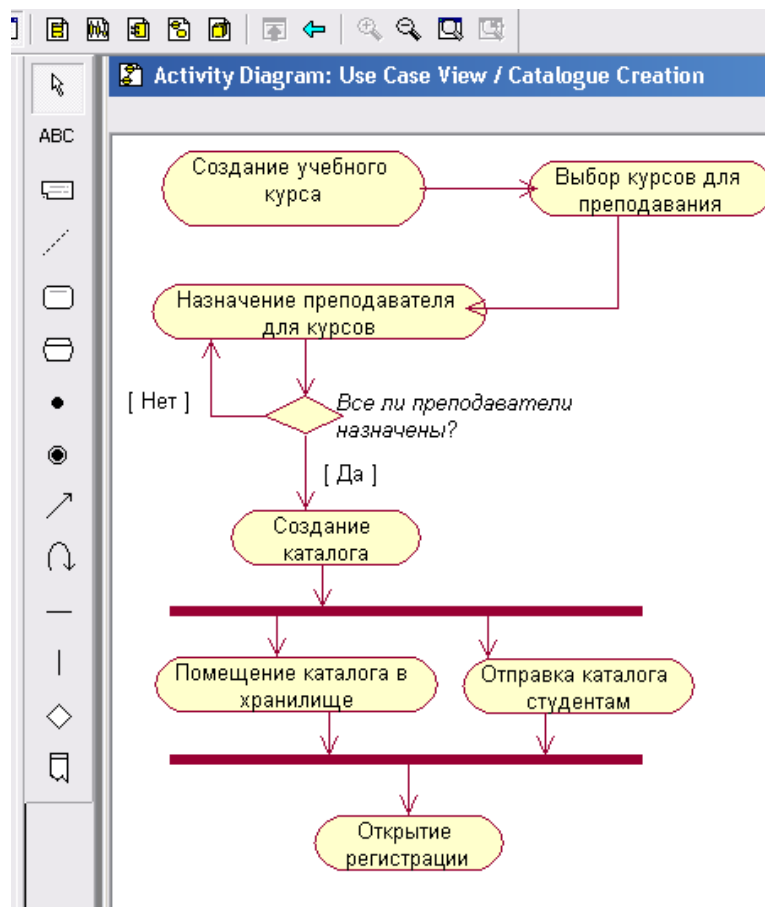


Рис. 12. Прямолинейные линии на диаграмме действий.

Линии синхронизации

В потоке обычно существуют действия, выполняемые **параллельно**. **Линия синхронизации** (synchronization bar) позволяет указать на необходимость их одновременного выполнения, а также обеспечивает единое выполнение действий в потоке (то есть указывает на необходимость завершения определенных действий для перехода к следующему) - см. рис. 13. Таким образом, линии перехода могут иметь несколько входящих линий переходов и одну исходящую либо одну входящую и несколько исходящих.

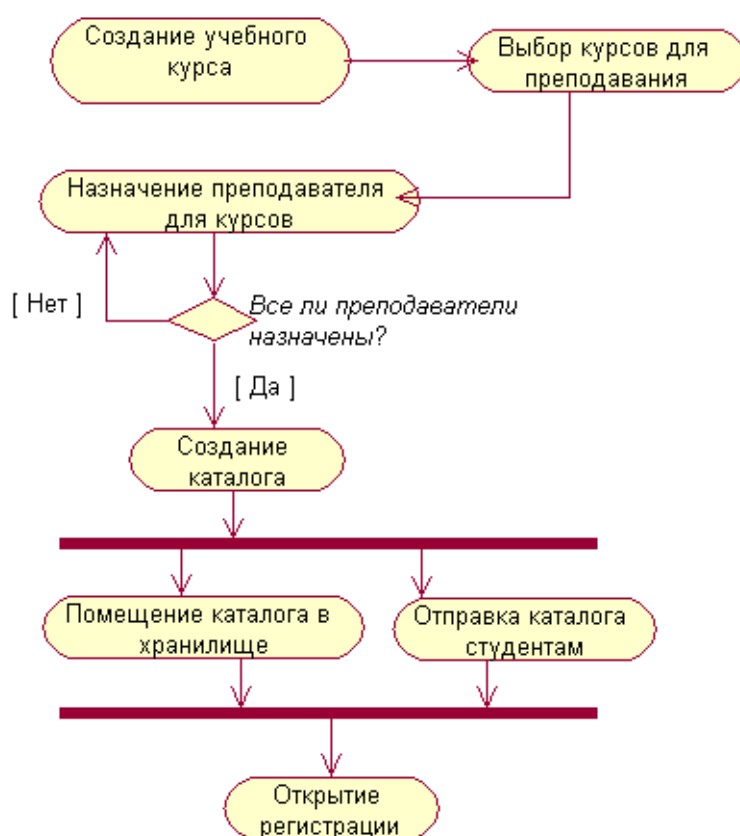


Рис. 13. Линии синхронизации на диаграмме действий.

Секции

Секции (swimlanes) делят диаграммы действий на несколько участков. Это нужно для того, чтобы показать, кто отвечает за выполнение действий на каждом участке.

Диаграмма действий с разделительными линиями показана на рис. 14.

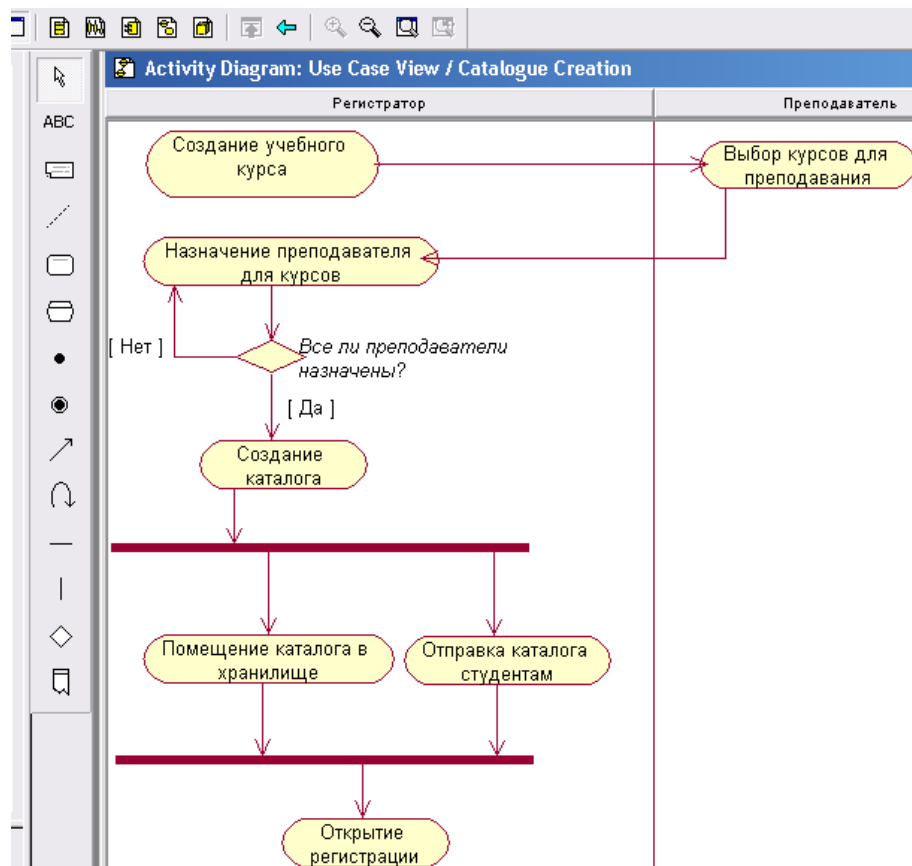


Рис. 14. Секции на диаграмме действий для Use Case Создание каталога

Начальное и конечное состояния

Для обозначения начального и конечного состояний в потоке управления системы используются специальные символы. Начальное состояние изображается в виде закрашенного круга, а конечное - в виде закрашенного круга, обведенного дополнительной окружностью. Обычно в потоке существуют одно начальное и несколько конечных состояний - для

каждого альтернативного направления.

Окончательная диаграмма действий для Use case с начальным и конечным состояниями показана на рис. 15.

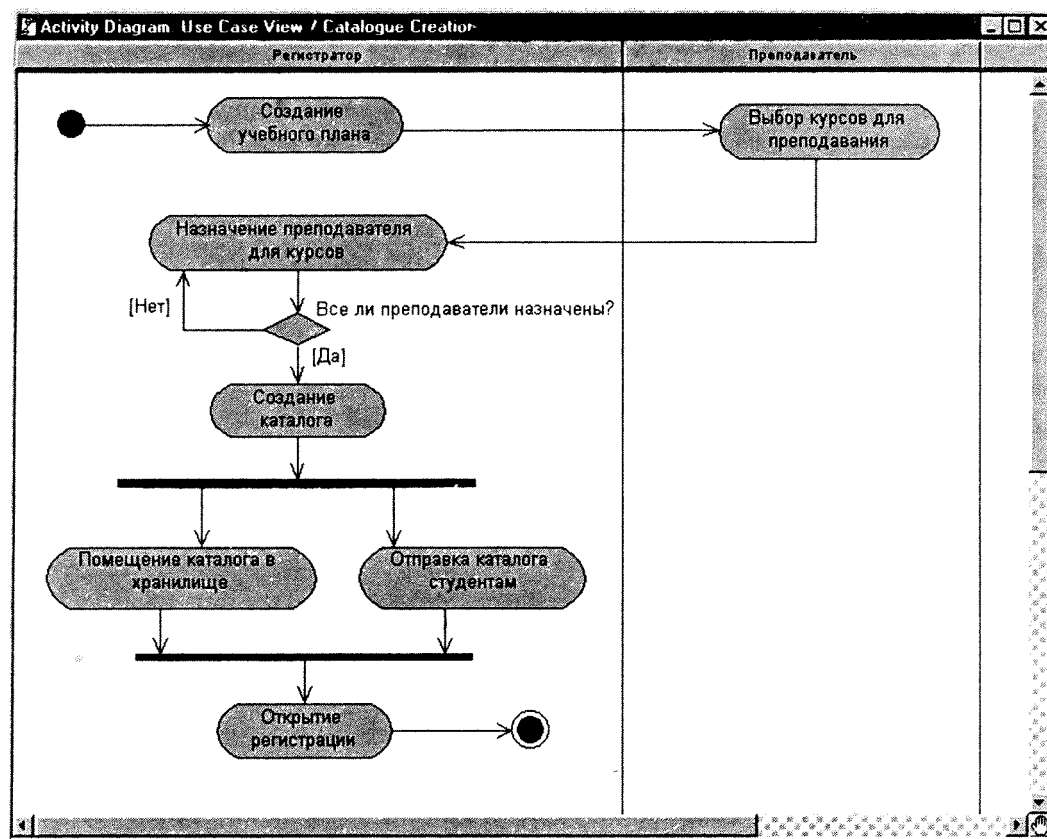


Рис. 15. Секции на диаграмме действий

Заключение

Поведение системы описывается с помощью модели прецедентов, которая содержит системные прецеденты, системное окружение - актеров и связи между прецедентами и актерами - диаграммы прецедентов. Модель прецедентов должна представлять собой единое средство для обсуждения функциональности и поведения системы с заказчиком и конечным пользователем.

Разработка модели прецедентов начинается на стадии задумки с выбора актеров и определения общих принципов прецедентов системы.

Затем модель дополняется на этапе проработки.

Актеры не являются частью системы - это кто-то или что-то, что должно взаимодействовать с системой. Прецеденты определяют функциональность системы. Они служат для моделирования диалога между актером и системой.

Для каждого прецедента указывается поток событий, описывающий те из них, которые необходимы для обеспечения требуемого поведения. Поток событий описывается в терминах того, «что» система должна делать, а не «как» она должна это делать. Диаграмма прецедентов - это графическое представление всех или части актеров, прецедентов и их взаимодействий в системе.

Диаграммы действий отражают динамику системы. Они представляют собой схемы потоков управления в системе. В конкретной точке жизненного цикла диаграммы действий могут представлять потоки между прецедентами или внутри отдельного прецедента. На последующих этапах жизненного цикла диаграммы действий могут создаваться для отражения последовательности выполнения операции.

Порядок выполнения лабораторной работы

1. Изучить методические материалы по лабораторной работе.
2. Анализ вариантов использования.

3. В данной работе Вам предлагается создать представление вариантов использования. При этом должны быть уточнены детали моделей и 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);

4. Составить диаграмму действия рис. 15.

5. Составить диаграмму действия для своего варианта задания.

6. 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.

7. 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?
20. Назовите основные структурные компоненты Rational Rose и их функции.

Литература:

1. Кватрани Т. Rational Rose 2000 и UML. Визуальное моделирование: Пер. с англ. - М.: ДМК Пресс, 2001. - 176 с.: ил. (Серия «Объектно-ориентированные технологии в программировании»).
2. Вендров А.М. Проектирование программного обеспечения экономических информационных систем. Учебник. М.: Финансы и статистика, 2003, 352 с.
3. Вендров А.М. Практикум по проектированию программного обеспечения экономических информационных систем. М.: Финансы и статистика, 2002, 192 с.