

Лабораторная работа №2

Моделирование предметной области на основе UML

1 Цель работы. Сформировать практические навыки моделирования предметной области путем построения диаграмм вариантов использования с использованием MS Visual Studio 2010.

2 Краткая теория

UML - унифицированный язык моделирования.

Диаграмма вариантов использования = диаграмма прецедентов.

Диаграммы вариантов использования описывают функциональное назначение системы или то, что система должна делать. Разработка диаграммы преследует следующие цели:

- определить общие границы и контекст моделируемой предметной области;
- сформулировать общие требования к функциональному поведению проектируемой системы;
- разработать 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.

Суть диаграммы вариантов использования состоит в следующем. Проектируемая система представляется в виде множества сущностей или актеров, взаимодействующих с системой с помощью вариантов использования. При этом актером (actor) или действующим лицом называется любая сущность, взаимодействующая с системой извне. Это может быть человек, техническое устройство, программа или любая другая система, которая может служить источником воздействия на моделируемую систему так, как определит сам разработчик. Вариант использования служит для описания сервисов, которые система предоставляет актеру. Диаграмма вариантов использования может дополняться пояснительным текстом, который раскрывает смысл или семантику составляющих ее компонентов.

Отдельный **вариант использования** обозначается на диаграмме **эллипсом**, внутри которого содержится его краткое название или имя в форме глагола с пояснительными словами.

Цель варианта использования заключается в том, чтобы определить законченный аспект или фрагмент поведения некоторой сущности без раскрытия её внутренней структуры. В качестве такой сущности может выступать система или любой элемент модели, который обладает собственным поведением.

Варианты использования могут применяться как для спецификации внешних требований к проектируемой системе, так и для спецификации функционального поведения уже существующей системы.

Примерами вариантов использования могут являться следующие действия: проверка состояния текущего счета клиента, оформление заказа на

покупку товара, отображение графической формы на экране монитора и другие действия.

Актёр представляет собой любую внешнюю по отношению к моделируемой системе сущность, которая взаимодействует с системой и использует ее функциональные возможности для достижения определенных целей. При этом актёры служат для обозначения согласованного множества ролей, которые могут играть пользователи в процессе взаимодействия с проектируемой системой. Каждый актёр может рассматриваться как некая отдельная роль относительно конкретного варианта использования. Стандартным графическим обозначением актёра на диаграммах является **фигурка человечка**, под которой записывается имя актёра.

Примерами актёров могут быть: клиент банка, банковский служащий, продавец магазина, менеджер отдела продаж, пассажир авиарейса, водитель автомобиля, администратор гостиницы, сотовый телефон и другие сущности, имеющие отношение к концептуальной модели соответствующей предметной области.

Примечания (notes) в языке UML предназначены для включения в модель произвольной текстовой информации, имеющей непосредственное отношение к контексту разрабатываемого проекта. В качестве такой информации могут быть комментарии разработчика (например, дата и версия разработки диаграммы или ее отдельных компонентов), ограничения (например, на значения отдельных связей или экземпляры сущностей) и помеченные значения. Графически примечания обозначаются **прямоугольником с загнутым верхним правым углом**. Внутри прямоугольника содержится текст примечания.

Если в примечании указывается ключевое слово «constraint», то оно является ограничением, налагаемым на соответствующий элемент модели.

Между элементами диаграммы вариантов использования могут существовать различные **отношения**, которые описывают взаимодействие экземпляров актёров и вариантов использования.

В языке UML существует несколько стандартных видов отношений между актёрами и вариантами использования:

- **ассоциации** (association relationship);
- **расширения** (extend relationship);
- **общения** (generalization relationship);
- **включения** (include relationship).

Отношение ассоциации

Применительно к диаграммам вариантов использования **ассоциация** специфицирует семантические особенности взаимодействия актёров и вариантов использования в графической модели системы, то есть, это отношение устанавливает, какую конкретную роль играет актёр при взаимодействии с экземпляром варианта использования. На диаграмме вариантов использования отношение ассоциации обозначается **сплошной линией** между актёром и вариантом использования. Эта линия может иметь условные обозначения, такие как имя и кратность.

Отношение расширения

Отношение **расширения** определяет взаимосвязь экземпляров отдельного варианта использования с более общим вариантом, свойства которого определяются на основе способа совместного объединения данных экземпляров. В метамодели отношение расширения является направленным и указывает, что применительно к отдельным примерам некоторого варианта использования должны быть выполнены конкретные условия, определенные для расширения данного варианта использования. Так, если имеет место отношение расширения от варианта использования **A** к варианту использования **B**, то это означает, что свойства экземпляра варианта использования **B** могут быть дополнены благодаря наличию свойств у расширенного варианта использования **A**.

Отношение расширения между вариантами использования обозначается **пунктирной линией со стрелкой** (вариант отношения зависимости), направленной от того варианта использования, который является расширением для исходного варианта использования. Данная линия со стрелкой помечается ключевым словом «extend» (расширяет). Отношение расширения отмечает тот факт, что один из вариантов использования может присоединять к своему поведению некоторое дополнительное поведение, определенное для другого варианта использования.

Отношение обобщения

Отношение **обобщения** служит для указания того факта, что некоторый вариант использования **A** может быть обобщен до варианта использования **B**. В этом случае вариант **A** будет являться специализацией варианта **B**. При этом, **B** называется предком или родителем по отношению **A**, а вариант **A** - потомком по отношению к варианту использования **B**. 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 вариант использования.

Отношение обобщения между вариантами использования применяется в том случае, когда необходимо отметить, что дочерние варианты использования обладают всеми атрибутами и особенностями поведения родительских вариантов.

Между отдельными актерами также может существовать отношение обобщения. Данное отношение является направленным и указывает на факт специализации одних актеров относительно других. Например, отношение обобщения от актера **A** к актеру **B** отмечает тот факт, что каждый экземпляр актера **A** является одновременно экземпляром актера **B** и обладает всеми его свойствами.

Отношение включения

Отношение **включения** между двумя вариантами использования указывает, что некоторое заданное поведение для одного варианта использования включается в качестве составного компонента в последовательность поведения другого варианта использования.

Семантика этого отношения определяется следующим образом. Когда экземпляр первого варианта использования в процессе своего выполнения достигает точки включения в последовательность поведения экземпляра второго варианта использования, экземпляр первого варианта использования выполняет последовательность действий, определяющую поведение экземпляра второго варианта использования, после чего продолжает выполнение действий своего поведения. Один вариант использования может быть включен в несколько других вариантов, а также включать в себя другие варианты.

Отношение включения, направленное от варианта использования **A** к варианту использования **B**, указывает, что каждый экземпляр варианта **A** включает в себя функциональные свойства, заданные для варианта **B**. Эти свойства специализируют поведение соответствующего варианта **A** на данной диаграмме. Графически данное отношение обозначается *пунктирной линией со стрелкой*, которая помечается ключевым словом «include» (включает).

3 Порядок выполнения работы

Этапы:

1. Моделирование контекста системы.
2. Моделирование требований.
3. Сдача отчета преподавателю.

Этап 1. Моделирование контекста системы

Моделирование контекста подразумевает выявление актеров, которые находятся вне системы и взаимодействуют с ней. Диаграммы прецедентов нужны на этом этапе для идентификации актеров и семантики их ролей.

Любая система содержит внутри себя какие-либо сущности, в то время как другие сущности остаются за ее пределами. Например, в системе проверки кредитных карточек имеются счета, транзакции и механизмы проверки подлинности. В то же время обладатели кредитных карточек и торговые предприятия находятся вне системы. Сущности внутри системы отвечают за реализацию поведения, которого ожидают сущности, находящиеся снаружи. Сущности, находящиеся вне системы и взаимодействующие с ней, составляют ее контекст. Таким образом, контекстом называется окружение системы.

UML позволяет моделировать контекст с помощью диаграмм прецедентов, в которых внимание акцентируется на окружающих систему актерах. Важно правильно определить актеров, поскольку это позволяет описать класс сущностей, взаимодействующих с системой. Еще важнее определить, что не является актером, так как при этом ограничивается окружение системы: в нем остаются только те элементы, которые участвуют в ее работе.

Моделирование контекста системы состоит из следующих шагов:

1. Идентифицируйте окружающих систему актеров. Для этого нужно найти группы, которым участие системы требуется для выполнения их задач; группы, которые необходимы для осуществления системой своих функций; группы, взаимодействующие с внешними программными и аппаратными средствами, а также группы, выполняющие вспомогательные функции администрирования и поддержки.
2. Организуйте похожих актеров с помощью отношений обобщения/специализации.

Например, на рис. 2.1 показан контекст системы, работающей с кредитными карточками, где основное внимание уделяется окружающим ее актерам. В первую очередь это Клиенты двух типов (Индивидуальный клиент и Корпоративный клиент), соответствующие ролям, которые играют люди при взаимодействии с системой. В этом контексте показаны и актеры, представляющие другие организации, такие как Торговые предприятия (с ними покупатели совершают карточные транзакции, приобретая вещи или услуги) и Субсидирующие финансовые институты (выполняющие роль клиринговой палаты для карточных счетов). В реальном мире последние два актера, скорее всего, сами будут программными системами.



Рис. 2.1.

Аналогично можно моделировать контекст **подсистемы**. Элемент, который на одном уровне абстракции выглядит как система, часто становится подсистемой на другом, более высоком уровне абстракции. Моделирование контекста подсистемы может пригодиться при построении системы из нескольких взаимосвязанных частей.

Задание. Для выявленных заинтересованных лиц определить, являются ли они актерами, провести обобщение актеров (если это возможно и необходимо), построить диаграмму.

Этап 2. Моделирование требований к системе

Моделирование требований к системе предполагает указание на то, что система должна делать (с точки зрения внешнего наблюдателя), независимо от того, как она должна это делать. Диаграммы прецедентов нужны здесь для специфицирования желаемого поведения системы.

Требование (Requirement) - это особенность проекта, свойство или поведение системы. Приступая к сбору требований, вы как бы описываете условия контракта, заключаемого между системой и сущностями вне нее, в котором декларируется, что система должна делать. При этом, как правило, вас заботит не то, как именно система будет выполнять поставленные задачи, а только то, что она будет делать. Хорошо спроектированная система должна полностью выполнять все требования, причем делать это предсказуемо и надежно. Ее создание начинается с соглашения о том, каково ее назначение, хотя в ходе разработки понимание требований будет постепенно изменяться.

Требования можно выразить по-разному, от неструктурированного текста до выражений на формальном языке (или, например, с помощью примечаний, user story). Большая часть функциональных требований к системе - или даже все они - может быть выражена в виде вариантов использования, в чем помогают диаграммы прецедентов UML.

Моделирование требований к системе производится следующим образом:

1. Установите контекст системы, идентифицировав окружающих ее актеров.
2. Для каждого актера рассмотрите поведение, которого он ожидает или требует от системы (используя истории пользователей из л.р.1).
3. Поименуйте эти общие варианты поведения как прецеденты.
4. Выделите общее поведение в новые прецеденты, которые будут использоваться другими; выделите вариации поведения в новые прецеденты, расширяющие основные потоки событий.
5. Смоделируйте эти прецеденты, актеры и отношения между ними на диаграмме прецедентов.
6. Дополните прецеденты примечаниями, описывающими требования; некоторые из таких примечаний можно присоединить к системе в целом.

На рис. 2.2. приведен пример диаграммы использования для выявления требований. Для повышения наглядности в диаграмму можно добавлять примечания. Такая же методика применима и при моделировании требований к подсистеме.



Рис. 2.2.

Задание. Построить ДВИ разрабатываемой системы.

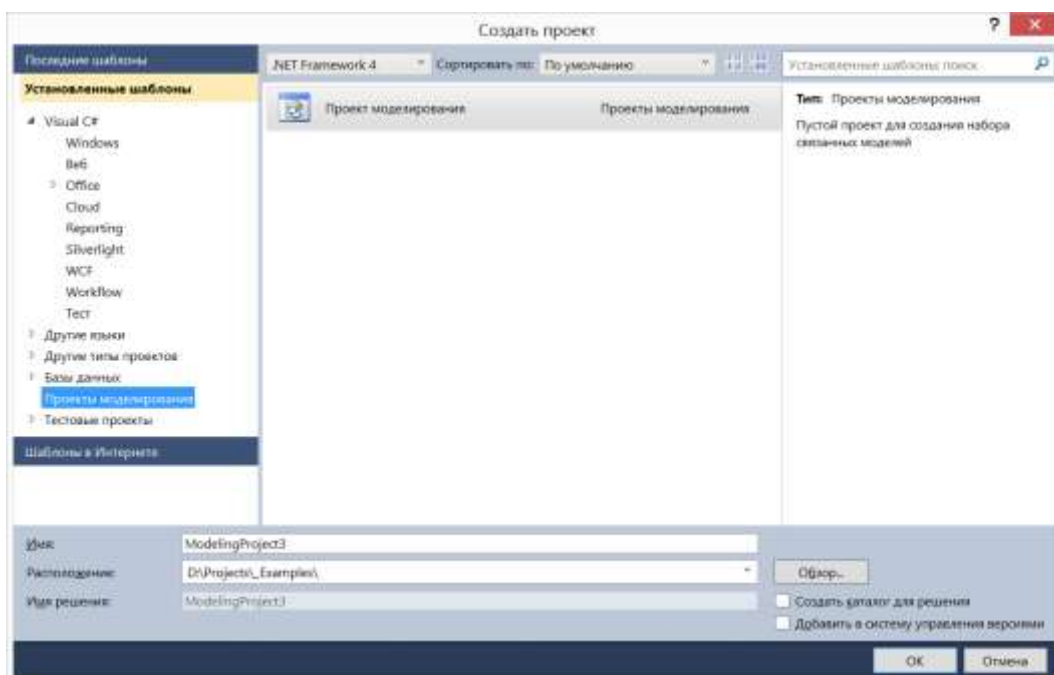
Этап 3. Сдача отчета преподавателю.

Отчет сдается в электронном виде: каталог созданного в Visual Studio проекта моделирования архивируется и высылается по адресу typer2010@mail.ru. В теме письма указать ФИО, группа, номер лабораторной работы, в теле письма – название проектируемой системы.

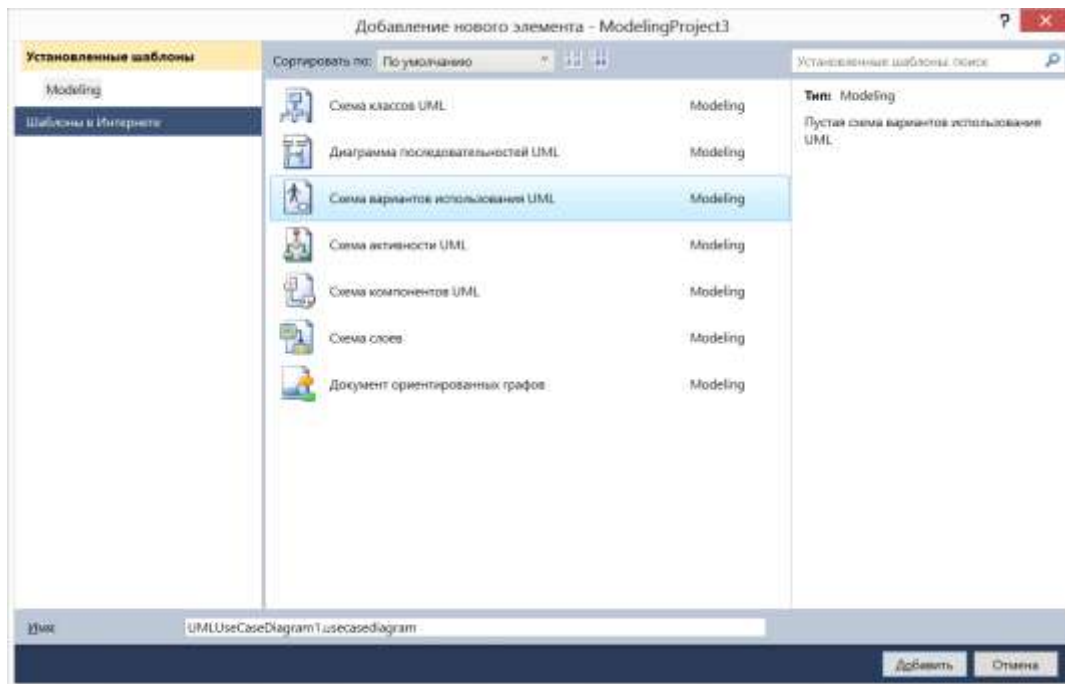
Прежде чем высылать отчет по лабораторной работе преподавателю, необходимо защитить ее!

4. Диаграммы вариантов использования в Visual Studio 2010

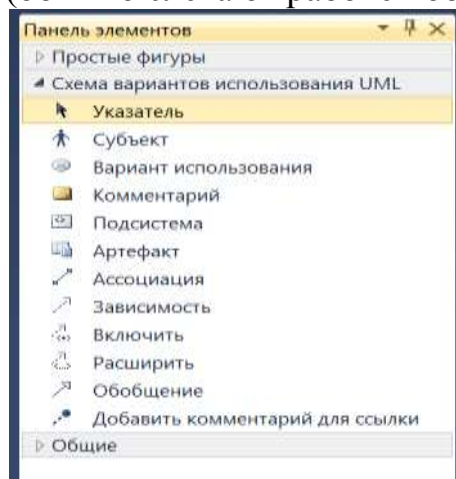
1. Создать новый проект, используя шаблон «Проекты моделирования»:



2. Добавить новую диаграмму в проект через *Проект - Добавить новый элемент*:



3. Панель инструментов для построения диаграммы использования (обычно слева от рабочей области):



Примечание. Проект моделирования может содержать не только диаграммы вариантов использования. Необходимо добавлять другие виды диаграмм к уже созданному вами проекту. Не надо для каждой новой диаграммы создавать отдельный проект.