Лабораторная работа 1 Моделирование ИС в нотации UML

Цель работы:

1. Изучить методологию структурного моделирования ИС с помощью UML [1, 2].
2. Получить навыки работы с программными пакетами, поддерживающими данную методологию (MS Visio) [3].

Теоретические сведения

Определение

**UML (**Unified Modeling Language– унифицированный язык моделирования**)** – это графический язык моделирования общего назначения, предназначенный для спецификации, визуализации, проектирования и документирования всех артефактов, создаваемых при разработке программных систем[].

Артефакт (от латинского *artefactum)* — искусственно созданное, сделанное, любой результат труда людей[]. Термин дает нам недвусмысленный намек на то, что программный код – целевой, но далеко не единственный результат разработки.

Визуализация – наглядное представление, ориентированное на максимально широкий круг людей, в большей или меньшей степени отвечающих за процесс разработки (аналитик, архитектор, разработчик, тестер, заказчик, пользователь). Специфицирование – уточнение начала разработки – согласование постановки задачи в более строго формализованной (по возможности и необходимости) форме.

**Модель UML** ‒ это совокупность конечного множества конструкций языка, главные из которых ‒ это сущности и отношения между ними.

Синтаксис UML

* актор(actor) – (действующее лицо) объект, который взаимодействует с системой, но не является ее частью;
* диаграммы сценариев (use-case);
* диаграммы классов (class);
* диаграммы состояния (statechart);
* диаграммы активности (activity);
* диаграммы последовательности (sequence);
* диаграммы взаимодействия (collaboration);
* диаграммы компонентов (component);
* диаграммы топологий (Deployment).

### Семантика UML-диаграмм

*Диаграмма вариантов использования* (use case diagram)(диаграмма прецедентов, диаграмма сценариев) показывает набор сценариев, актеров и их отношений. *Диаграмма вариантов использования* представляет собой наиболее общую концептуальную модель системы, которая является исходной для построения всех остальных диаграмм. Эти диаграммы особенно важны при орга­низации и моделировании поведения системы, задании требований заказчика к системе.

*Диаграмма классов* (class diagram) показывает набор классов, интерфейсов, коопераций и ихотношений. При моделировании объектно-ориентированных систем диаграммы классов используются наиболее часто. Диаграммы классов можно отнести к логической модели, отражающей статические аспекты структурного построения системы. Диаграммы классов, включающие активные классы, обеспечивают статическое представление процессов системы.

*Диаграмма объектов* (object diagram)показывает набор объектов и их отношения. Диаграмма объек­тов представляет статический «моментальный снимок» с экземпляров предметов, которые находятся на диаграммах классов. Как и диаграммы классов, эти диаграммы обеспечивают статическое проектное представление или статическое представ­ление процессов системы (с точки зрения реальных случаев).

*Диаграмма последовательности и диаграмма сотрудничества (кооперации)* – это разновиднос­ти *диаграмм взаимодействия* (interaction diagrams). Диаграмма взаимодействияпоказывает взаимодействие, включающее набор объек­тов и их отношений, а также пересылаемые между объектами сообщения. Диа­граммы взаимодействия обеспечивают динамическое представление системы.

*Диаграмма последовательности* (sequence diagram) – это диаграмма взаимодействия, которая выде­ляет упорядочение сообщений по времени.

*Диаграмма сотрудничества* (диаграмма кооперации) (collaboration diagram) – это диаграмма взаимодей­ствия, которая выделяет структурную организацию объектов, посылающих и при­нимающих сообщения.

Диаграммы последовательности и диаграммы кооперации изоморфны. Это означает, что одну диаграмму можно трансформировать в другую диаграмму.

*Диаграмма схем состояний* (statechart diagram)показывает конечный автомат, представляет состоя­ния, переходы, события и действия. Диаграммы схем состояний обеспечивают ди­намическое представление системы. Они особенно важны при моделировании по­ведения интерфейса, класса или кооперации. Эти диаграммы выделяют такое поведение объекта, которое управляется событиями.

*Диаграмма деятельности* (activity diagram) – специальная разновидность диаграммы схем состоя­ний, которая показывает поток от действия к действию внутри системы. Диаграм­мы деятельности обеспечивают динамическое представление системы. Они осо­бенно важны при моделировании функциональности системы и выделяют поток управления между объектами.

Диаграммы взаимодействия, диаграммы состояний и диаграммы деятельности относятся к *диаграммам поведения* (behavior diagrams). Диаграммы поведения представляют логическую модель системы и отражают динамические аспекты ее функционирования.

Диаграмма компонентов и диаграмма размещения – это разновидности *диаграмм реализации* (implementation diagrams). Эти диаграммы служат для представления физических компонентов системы и относятся к ее физической модели.

*Компонентная диаграмма* (component diagram) показывает организацию набора компонентов и зависи­мости между компонентами. Компонентные диаграммы обеспечивают статичес­кое представление реализации системы. Они связаны с диаграммами классов в том смысле, что в компонент обычно отображается один или несколько классов, ин­терфейсов или коопераций.

*Диаграмма размещения* (диаграмма развертывания, диаграмма топологии) (deployment diagram) показывает конфигурацию физических элементов системы, а также компоненты, размещенные в них. Диаграммы развертывания обеспечивают статическое представление размещения системы. Они связаны с компонентными диаграммами в том смысле, что узел обыч­но включает один или несколько компонентов

### Отношения диаграмм.

Отношения между различными моделями UML - это отображение итеративной природы объектно-ориентированного моделирования.

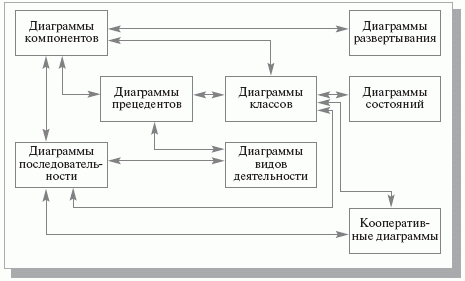


Рисунок 1.1 Отношения диаграмм

Представление процесса разработки - последовательное. На рисунке 3.1 показано, что диаграммы состояний используются для документирования диаграмм классов. Точнее диаграмма состояний будет использоваться для документирования единственного класса на диаграмме классов.

**Диаграммы сценариев**

Диаграммы использования описывают функциональность ИС, которая будет видна пользователям системы. "Каждая функциональность" изображается в виде "прецедентов использования" (use case) или просто прецедентов. Прецедент — это типичное взаимодействие пользователя с системой, которое при этом:

·                    описывает видимую пользователем функцию,

·                    может представлять различные уровни детализации,

·                    обеспечивает достижение конкретной цели, важной для пользователя.

Прецедент обозначается на диаграмме овалом, связанным с пользователями, которых принято называть действующими лицами (актеры, actors). Действующие лица используют систему (или используются системой) в данном прецеденте. Действующее лицо выполняет некоторую роль в данном прецеденте. На диаграмме изображается только одно действующее лицо, однако реальных пользователей, выступающих в данной роли по отношению к ИС, может быть много. Список всех прецедентов фактически определяет функциональные требования к ИС, которые лежат в основе разработки технического задания на создание системы.

На диаграммах прецедентов, кроме связей между действующими лицами и прецедентами, возможно использование еще двух видов связей между прецедентами: "использование" и "расширение" ( [рис. 1.2](http://www.intuit.ru/studies/courses/2195/55/lecture/846?page=3#image.11.4)). Связь типа "расширение" применяется, когда один прецедент подобен другому, но несет несколько большую функциональную нагрузку. Ее следует применять при описании изменений в нормальном поведении системы. Связь типа "использование" позволяет выделить некий фрагмент поведения системы и включать его в различные прецеденты без повторного описания.

На [рис. 1.2](http://www.intuit.ru/studies/courses/2195/55/lecture/846?page=3#image.11.4) показано, что при исполнении прецедента " формирование заказа " возможно использование информации из предыдущего заказа, что позволит не вводить все необходимые данные. А при исполнении прецедентов " оценить риск сделки " и " согласовать цену " необходимо выполнить одно и то же действие — рассчитать стоимость заказа.

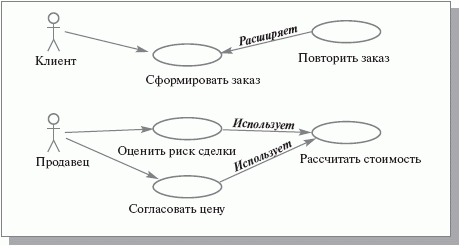


Рисунок 1.2 - Связи на диаграммах прецедентов

Правила разработки диаграмм вариантов использования

1. Не моделируйте связи между действующими лицами. По определению акторы находятся вне сферы действия системы. Это означает, что связи между ними также не относятся к ее компетенции.
2. Не соединяйте стрелкой непосредственно два варианта использования, кроме случаев отношений между ними (обобщение, включение и расширение). Диаграммы данного типа описывают только, какие варианты использования доступны системе, а не порядок их выполнения.
3. Каждый *конкретный* вариант использования должен быть инициирован действующим лицом. Это означает, что всегда должна быть стрелка, начинающаяся на действующем лице и заканчивающаяся на варианте использования. Исключением являются отношения между вариантами использования (обобщение, включение и расширение).
4. База данных – это слой, находящийся под диаграммой. С помощью одного варианта использования можно вводить данные в базу, а получать их с помощью другого. Для изображения потока информации не нужно рисовать стрелки от одного варианта использования к другому.

Порядок выполнения лабораторной работы

1 Внимательно изучить УЧЕБНЫЙ ПРОЕКТ [3].

2 Для заданной предметной области построить две диаграммы использования (UseCase) : общей функциональности и раскрытие одного из сценариев.

3 Оформить отчёт, который должен содержать:

1) титульный лист;

2) название лабораторной работы, цель;

3) построенные UML-диаграммы.

Рекомендуемая литература

1 Буч, Г. Язык UML. Руководство пользователя [Электронный ресурс] / Г. Буч, Д. Рамбо, И. Якобсон. - 2-е изд.: Пер. с англ. Н. Мухин. - Москва : ДМК Пресс, 2008. - 496 с.

2 Введение в UML / Текст: электронный//Портал Национальный Открытый Университет «ИНТУИТ [сайт]. URL: <http://www.intuit.ru/studies/courses/1007/229/lecture/5950>

3 Проектирование информационных систем UML / Текст: электронный//Портал Национальный Открытый Университет «ИНТУИТ [сайт]. URL: <https://intuit.ru/studies/courses/2195/55/lecture/1642>