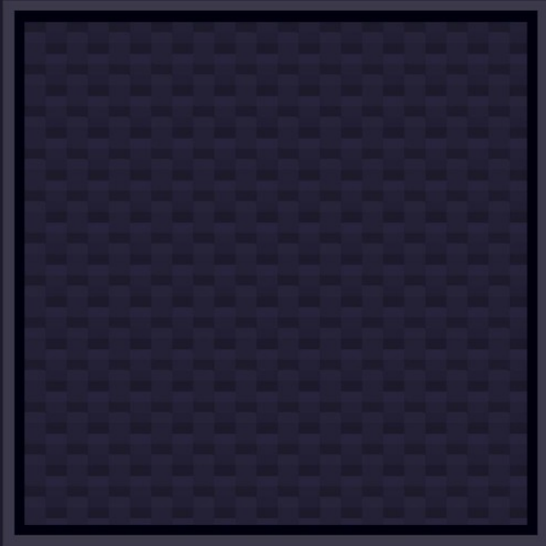


Системный Анализ. Предметная область

Михаил Кумсков



Модели на UML

Михаил Кумсков
Системный Анализ.
Предметная область.
Модели на UML

*http://www.litres.ru/pages/biblio_book/?art=49788965
ISBN 9785005093851*

Аннотация

Конспект лекции по определению модели предметной области на конкретном примере. Используется подход, существенно отличающийся от известного ER-моделирования. Модель имеет «визуальный характер» и изображается в нотации Unified Modeling Language (UML), которая «широко известна в узких кругах» аналитиков, архитекторов, разработчиков и программистов. Описаны паттерны, применяемые для преобразования диаграмм классов на UML и приведены примеры их практического использования.

Содержание

Введение	6
Благодарности	10
Раздел 1	12
Построение визуальной модели предметной области	12
Шаг №0. Определяем цели построения модели	13
Шаг №1. Определяем события, подлежащие учету	14
Шаг №2. Определяем справочники, подлежащие учету	16
Шаг №3. Для события определяем картотеки, связанные с ним (для каждого события)	17
Конец ознакомительного фрагмента.	18

Системный Анализ. Предметная область Модели на UML

Михаил Кумсков

© Михаил Кумсков, 2020

ISBN 978-5-0050-9385-1

Создано в интеллектуальной издательской системе Ridero
Книга представляет собой краткий конспект лекций по определению модели предметной области на конкретном примере. Используется объектно-ориентированный подход, существенно отличающийся от известного моделирования «сущность – связь», или ER-моделирования. Модель имеет визуальный характер и изображается в нотации Unified Modeling Language (UML), которая широко известна среди аналитиков, архитекторов, разработчиков и программистов. Описаны паттерны, применяемые для преобразования диаграмм классов на UML, и приведены примеры их практического использования. Изложение ведется согласно методологии IBM RUP.

Материал будет полезен студентам и аспирантам, участ-

никам проектов по разработке информационных систем, а также слушателям курсов по выявлению требований к ИС и по проектированию архитектуры ИС.

Введение

Модель предметной области служит разным целям:

- 1) помогает определить логическую структуру БД информационной системы;
- 2) является основой для составления «расширенного» словаря проекта;
- 3) помогает найти все сценарии (при выявлении функциональных требований в специальной форме – в виде сценариев использования (Use Cases));
- 4) позволяет не пропустить «вспомогательные» сценарии, которые могут быть не упомянуты в постановке задачи, полученной от заказчика ИС.

Важной особенностью модели предметной области является ее **независимость** от используемых ИС и баз данных. Это совокупность записей в организации, которые следует вести, чтобы руководство этой организации смогло определить, все ли в порядке. Например, такие записи велись много лет назад задолго до появления компьютеров. Чтобы не было разночтений, определим некоторые используемые далее термины. Их понимание важно, но они могут отличаться от общеизвестных своей специфической направленностью.

- *Модель* – это «упрощение реальности» в интересах заинтересованных лиц.
- *База данных* – набор картотек, взаимосвязанных друг

с другом и ведущихся на компьютере.

- *Сервер* – процесс, предоставляющий целостный доступ к общему ресурсу.

- *Сервер БД*, или СУБД (система управления БД), предоставляет целостный доступ к базе данных как общему ресурсу.

- *Картотека* – набор карточек с «одинаковой структурой», представляется в модели *классом*.

- *Карточка* – запись в картотеке, представляется в модели *объектом*. Все карточки одной и той же картотеки имеют одинаковую структуру: одинаковый набор полей карточки (или атрибутов) и одинаковый набор связей с карточками других картотек.

- *Класс* – это описание набора «одинаковых» объектов, т. е. объектов, имеющих одинаковый *набор атрибутов*, одинаковый набор *операций* и одинаковый набор указателей на другие объекты. Картотеки представимы классами.

- *Объект* – это экземпляр класса, т. е. запись, или «карточка», в соответствующей картотеке.

- *Атрибут* – поименованное свойство объекта.

- *Операция* – сервис, который может быть запрошен у объекта. Метод или функция, инкапсулированная в объект.

Для проведения визуального моделирования будем использовать специальные программные инструменты, называемые CASE-средствами (Computer Assist Software

Engineering)¹. Тогда будет возможно проведение «генерации кода» по модели («прямое проектирование», или forward engineering) и обратное проектирование (reverse engineering) – восстановление модели по программному коду или по существующей БД.

Книга состоит из двух разделов. В первом описан пошаговый процесс выявления элементов модели и построения набора диаграмм классов UML как модели предметной области. Второй раздел вводит понятие процесса выявления требований к ИС в специальной форме «сценариев использования» (UC – Use Cases) и разъясняет, как использовать модель предметной области для выявления сценариев использования.

Модель предметной области отличается от моделирования «сущность – связь», или ER-моделирования (Entity-Relationship), методически и содержательно. ER-модель служит для описания логической структуры БД и имеет собственную визуальную нотацию. С технической точки зрения в сущностях визуальной ER-модели следует явно «прописывать» так называемые первичные и внешние ключи реляционной модели данных. Эти отличия кратко излагаются в приложении.

Ранее представленная методика была описана в методи-

¹ Если пользоваться графическими редакторами, например, MS Visio, то возникнут проблемы синхронизации визуальных элементов при «разрастании» (масштабировании) модели.

ческом пособии Кумсков М. И. Базы данных и процессы их создания. Введение. М.: Мехмат МГУ, 2004.

Благодарности

Появлению этой книги способствовали общение и совместная работа с многими профессионалами своего дела. Пользуюсь случаем и выражаю свою признательность:

Бахвалову Николаю Сергеевичу,
Позину Борису Ароновичу,
Сорокину Александру Викторовичу,
Ивановой Елене Владимировне,
Авдошину Сергею Михайловичу.

Представленные ниже методики и материалы прошли апробацию на курсах и тренингах, читаемых на мехмате МГУ, в департаменте программной инженерии факультета компьютерных наук ВШЭ, в учебном центре «Люксофт», при проведении мастер-классов и выступлений на конференциях. Видеозапись некоторых из них можно найти по следующим ссылкам:

1. Lviv IT Arena 2015: Mikhail Kumskov, Best Practices of Project Execution According to IBM Rational. URL: https://www.youtube.com/watch?time_continue=2&v=p5NzDKDzLOY
2. Value Management and Business Analysis (VM&BA). Mikhail Kumskov. Workshop. URL: <https://www.youtube.com/watch?v=6FEUlwXrfgQ>
3. Analyst Day 2014. Синергия UML: Модель предмет-

ной области, Бизнес-системы, Информационные системы: переход шаг за шагом. URL: https://www.youtube.com/watch?time_continue=5&v=Vl6SFx0rzqw

4. Летний аналитический фестиваль 2013 (ЛАФ-2013) «Системный анализ ИС и бизнес-системы – связь, сходства и различия». URL: https://www.youtube.com/watch?time_continue=3&v=4LtIQVj3juw

5. Конференции REQ Labs 2011. Процессы и люди. URL: <https://www.youtube.com/watch?v=cz5IBkf5E20>

Раздел 1

Построение визуальной модели предметной области

Модель – это «упрощение реальности» в интересах заинтересованных лиц. Такое определение относится и к нашему моделированию. Здесь главным заинтересованным лицом является инвестор или топ-менеджер организации. Есть и другие заинтересованные лица – аналитики, архитекторы, разработчики информационной системы (ИС), и поэтому одной модели, как правило, недостаточно. Нужны разные «упрощения» для разных читателей модели².

Первым шагом процесса моделирования является определение *целей* моделирования. Будем содержательно разбирать процесс построения на примере ИС, учитывающей расход продуктов в кафе и ресторанах организации, которую назовем «Комбинат питания». Текст с описанием задачи, полученный от владельца комбината, приведен в начале приложения 1.

Общий взгляд на процесс, состоящий из семи шагов, мож-

² «Сложность – это простота, изложенная подробно». Такое определение перекликается с понятием моделирования как «упрощения „сложной“ реальности».

но представить следующим списком задач, выполняемых в ходе моделирования:

- Шаг №0. Определяем цели построения модели.
- Шаг №1. Определяем события-картотеки, подлежащие учету на предприятии.
- Шаг №2. Определяем справочники-картотеки, подлежащие учету.
- Шаг №3. Для события определяем картотеки, связанные с ним (для каждого события).
- Шаг №4. Для справочника определяем картотеки, связанные с ним (для каждого справочника).
- Шаг №5. Отображаем (визуально) картотеки, связанные с ней на диаграмме классов UML.
- Шаг №6. Применяем паттерны преобразования отношений на диаграммах классов UML.

Шаг №0. Определяем цели построения модели

Цель построения модели в задаче «Комбинат питания» была определена в постановке задачи.

Это учет заказов гостей, движения продуктов и денег за них в пунктах питания – кафе и ресторанах. Теперь мы не будем учитывать и вводить в модель те детали, которые не относятся к заявленной цели. Например, не учитываем события «бронирование столика в кафе».

Далее следует определить те *события* («бизнес-транзакции»), которые подлежат учету, согласно заданным целям. Для таких событий на предприятии будут вестись учетные записи, или – в нашей терминологии – *картотеки* (гроссбухи, если учет бумажный).

Шаг №1. Определяем события, подлежащие учету

Для нашего примера мы выявляем бизнес-события по «движению продуктов питания и денег за них». Очевидно, что такими событиями будут:

1. **«Заказ»** гостя.
2. **«Оплата заказа»**.
3. **«Покупка продуктов»** в кафе.

Для каждого события определяется картотека – при возникновении события в этой картотеке должна быть создана новая учетная запись (карточка).

Для выявления других событий будем использовать *паттерны*³. Первым паттерном является введение «расхода» учетных сущностей – продуктов и блюд – через «*брак*» или «некачественную сущность», подлежащую списанию. По этому паттерну («Списание брака») вводим два новых со-

³ *Паттерн* — стандартный шаблон решения стандартной задачи, который хорошо себя зарекомендовал в прошлых проектах.

бытия:

4. «**Списание бракованных продуктов**» (по паттерну).
5. «**Списание бракованных блюд**» (по паттерну).

Следующий паттерн называется «**Инвентаризация**»: если на предприятии есть учетная система (информационная или на бумаге), то периодически следует выявлять реальный состав предметов учета на складе. Полученные в результате инвентаризации списки предметов (в нашем случае – продуктов питания) следует сравнить со списками, полученными (как отчеты) из учетной системы. Вводим в наш список событие:

6. «**Инвентаризация**» склада кафе (по паттерну).

Итак, пометка «по паттерну» означает, что данное событие может быть не указано в постановке задачи, но мы применяем паттерн (стандартное решение стандартной часто встречающейся задачи). Паттерн «**Списание брака**» означает учет (некачественных изделий) – в нашем случае – продуктов и блюд. Паттерн «**Инвентаризация**» используется, если имеется некоторая учетная система и требуется периодически сравнивать фактический состав продуктов на складе со «списочным составом», полученным из учетной системы. Событие «инвентаризация» – это проведение учета фактического состава продуктов на складе кафе ресторана на данную календарную дату. Проведение сравнения двух

списков («Инвентаризация» – «Отчет системы») позволяет выявить «недостачу».

Третьим паттерном является паттерн «**Прайс-лист**», который в комбинате питания называется «меню». Цены блюд в каждом кафе могут различаться. Для этого ведется меню как прайс-лист блюд на данную дату работы кафе. Поскольку для цен определена дата (или интервал дат) их «действия», то эту сущность мы отнесем к бизнес-событиям.

7. «**Меню**» кафе (по паттерну).

В результате мы выявили семь событий, подлежащих учету, – три непосредственно из постановки задачи и еще четыре на основе применения паттернов.

Шаг №2. Определяем справочники, подлежащие учету

Кроме картотек по событиям, на предприятии ведутся «учетные списки», которые можно назвать *справочниками*. Справочники более «стабильны» – изменения в них вносятся значительно реже и они «не зависят» от даты. Например, в комбинате питания следует вести список всех «пунктов питания» (ресторанов, кафе, закусовых). Это же относится и к списку сотрудников комбината. Такие справочники также являются картотеками, и они также подлежат учету в нашей модели как классы:

1. «**Пункт питания**» (ПП).
2. «**Сотрудник**».
3. «**Блюдо**» (рецепты).
4. «**Продукты**»⁴ (включая напитки).

Следует понимать, что на первых этапах могут быть выявлены не все события и справочники. Новые картотеки будут появляться далее в ходе итерационного построения модели, при применении паттернов и при получения обратной связи от заинтересованных лиц.

Шаг №3. Для события определяем картотеки, связанные с ним (для каждого события)

⁴ Включая напитки. Названия справочников приведены в единственном числе, поскольку название картотеки – это название класса.

Конец ознакомительного фрагмента.

Текст предоставлен ООО «ЛитРес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на ЛитРес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.