

Моделирование предметной области как часть бизнес- и системной аналитики

Традиционно моделирование предметной области является частью процесса бизнес- или системной аналитики разработки программного проекта [26]. Унифицированный процесс [27] относит моделирование предметной области исключительно к бизнес-аналитике и включает в себя отдельный рабочий процесс — моделирование бизнес-процессов (или производственное моделирование).

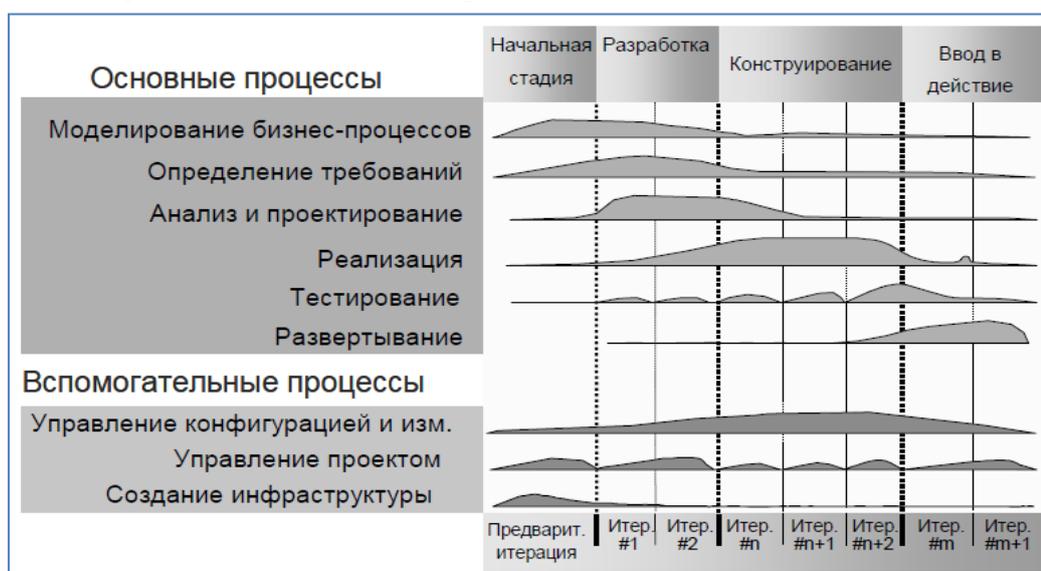


Рисунок 3. Фазы и процессы (дисциплины) единого процесса (УП)

Моделирование предметной области включает в себя два компонента:

- Моделирование процессов предметной области (бизнес-процессов);
- Моделирование структуры предметной области.

Бизнес-аналитика традиционно фокусируется на моделировании процессов и использует соответствующие модели (функциональные диаграммы, диаграммы деятельности UML , BPMN). Кроме того, бизнес-процессы частично эволюционируют в сценарии (юзкейсы) на более поздних стадиях проектирования, а сам бизнес-аналитик обычно не является членом команды разработчиков, и основная работа может быть передана на аутсорсинг .

Структура предметной области, таким образом, переходит в ведение системной аналитики и чаще востребована в процессе проектирования.

Системная аналитика отвечает за всю функциональность программного продукта, включая:

- адекватное отображение структуры предметной области в системе (диаграмма классов анализа);
- интерфейс взаимодействия предметной области и системы (варианты использования), графический интерфейс для приложений;

- бизнес-процессы в программной системе как продолжение бизнес-процессов предметной области (сценариев);
- базу знаний фактов и свойств системы, касающихся как отдельных сценариев, так и системы в целом (функциональные и другие требования).

Структура предметной области описывается в процессе разработки в виде нескольких взаимосвязанных моделей:

- Диаграмма классов предметной области (собственно модель предметной области, либо модель анализа — с учетом изменений и уточнений бизнес-процессов);
- Описание структуры базы данных (модель базы данных).
- Обычно происходит более-менее формальное преобразование: хозяйственные сущности - в таблицы БД, атрибуты - в поля, 1: N ассоциации - в эталонные поля со значениями ключа в таблице для связанной сущности, N: N ассоциаций - в промежуточные таблицы пар ключей. Наследование и полиморфизм сложнее реализовать;
- Архитектурные классы бизнес-уровня являются классами бизнес-объектов. Бизнес-объект - представление в программной среде субъекта предметной области (бизнес-объекта) в виде объекта, системы объектов или структуры данных связанных бизнес-объектов.
- DAO (Data Доступ Object) для преобразования записи таблицы базы данных в изоморфный объект архитектурного класса и ORM (Object - Относительный Mapping) — «сборка» бизнес-объектов из объектов классов бизнес-уровня.

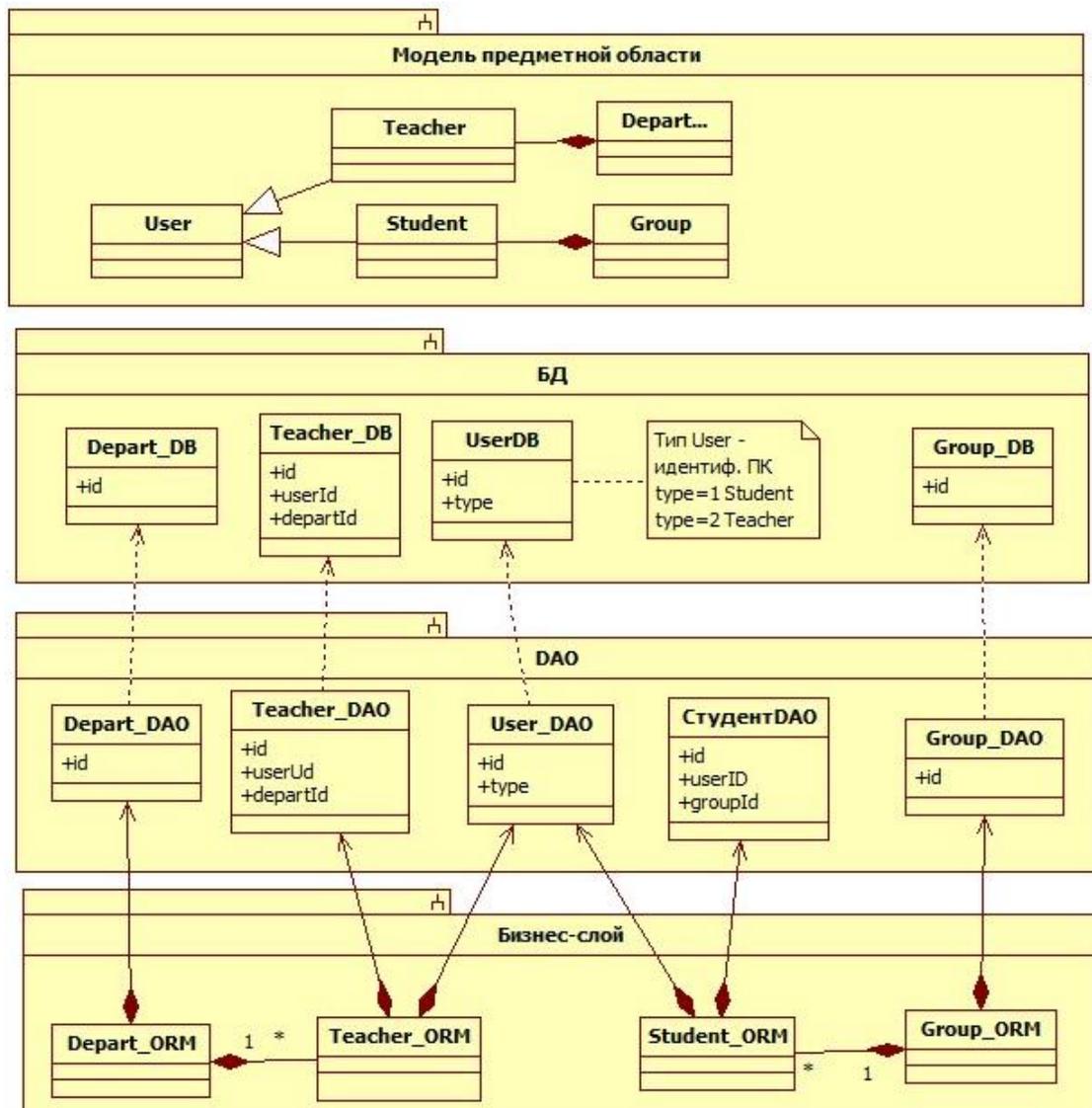


Рисунок 4. Эволюция структурной части модели предметной области

Ключевые моменты для разработки диаграммы классов предметной области:

- модель «отражает представление программной системы о том, с чем она работает»;
- любая бизнес-сущность предметной области становится классом диаграммы UML . В качестве атрибутов сущности могут оставаться только сущности примитивных типов или сущности, «принадлежащие» ей;
- наследование хозяйствующих субъектов осуществляется по естественному порядку отношения «общее-частное»;
- отношения между сущностями - *ассоциации* закрепляются как долгосрочные значимые отношения, т.е. структурный элемент. Поведенческие отношения-*зависимости* отражаются в диаграммах для описания динамики ;
- ассоциации устанавливаются на основе анализа формального определения сущностных отношений, например: билет - право просмотра в указанном месте и в

указанном сеансе, реализуемое одним из способов продажи (интернет, касса, бронирование);

- если есть разные варианты реализации ассоциации между бизнес-сущностями, она должна быть установлена между сущностью и *интерфейсом* или *абстрактный класс*;
- проверка модели осуществляется путем *ее тестирования* на конкретных сценариях (бизнес-процессах) - наличие необходимых связей, легкость перехода, а также с помощью логического анализа структуры системы;
- степень детализации модели должна соответствовать функциональности системы. Например, для системы мониторинга дорожного движения должна быть разработана модель топологии улично-дорожной сети в виде графа с соответствующей координатной привязкой, а не просто создание объекта «карта».

Типичные ошибки при разработке модели предметной области:

- модель должна описывать только структуру системы, а не ее поведение. Поэтому такие сущности, как «заказ», «покупка билета» здесь не применяются;
- модель не включает физические объекты инфраструктуры, в которой работает программная система, такие как «сервер», «клиентское приложение» и т. д.;
- модель предметной области не включает временные сущности и отношения, например: *данные отчета*, если он только формируется и отображается, а не хранится в архиве отчета в виде файла;
- модель предметной области не включает отношения между сущностями, если они не хранятся в системе, не используются при описании ее поведения. Например, зритель, покупающий билет в кассе, взаимодействует с системой опосредованно через кассира, данные о нем в этом случае не сохраняются, поэтому *связь клиент-кассир (касса, смена)* в модели отсутствует;
- семантика субъекта полностью не конкретизирована. Например, спектакль – это постановка произведения или его исполнение в определенное время в определенном зале. Или: смена – это интервал времени работы всей организации или конкретного работника;
- в указанном выше случае бизнес-сущность обычно требуется моделировать как одновременно несколько различных. Например, смена организации и смена работника. Аналогичный пример — блюдо как пункт меню и блюдо как тот же пункт в определенном заказе: второе ссылается на первое и содержит ряд

атрибутов, таких как количество порций, время включения в заказ. Блюдо в заказе следовало бы поименовать как отдельную бизнес-сущность *позиция заказа*;

- иногда имеет место смешивание понятий бизнес-сущности как класса и его экземпляра. Например, бизнес-сущность *меню* означает не все меню как экземпляр, а отдельную *позицию, расписание* – это не расписание в целом, а *строка с данными конкретного рейса*;
- множественность ассоциаций (1:1, 1:N, N:N) определяется только одним участником. Например, билет-место - 1:1 действует только на один сеанс, должно быть N:1, так как билеты разных сеансов могут относиться к одному и тому же месту.

Во всех случаях рекомендуется подробный глоссарий, включающий бизнес-сущности с описанием их семантики, взаимосвязей и особенностей бизнес-процессов с их участием.