

ВИЗУАЛЬНОЕ МОДЕЛИРОВАНИЕ СПРАВОЧНИКОВ В КАРКАСНОЙ ТЕХНОЛОГИИ ПОСТРОЕНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ

Продемонстрирована практика визуального моделирования номенклатуры информационной системы, занимающейся вопросами анализа, учета и контроля финансово-хозяйственной деятельности предприятия. Показаны некоторые недостатки традиционных методов моделирования, проиллюстрированы примерами способы приведения "классических" данных к унифицированному виду.

Проектирование информационной системы корпоративного масштаба по каркасной технологии [2] ставит разработчика перед необходимостью создания логической модели, несколько отличающейся от общепринятой. Главная причина такой специфики – наличие уже готовой логической модели, которая представляет собой собственно каркас информационной системы. Любое расширение и доработка этой модели делают неработоспособными основные механизмы, которые и определяют мобильность и гибкость каркасной ИС. Обоснование запрета на такие изменения, как и общие положения относительно технической реализации каркасной системы, приведены в работах [1,3].

Деятельность разработчика сводится, по сути, к формированию схемы справочников (которые будут реализованы в модели) и определению типа этих справочников (ресурсного, арматурного, служебного или неопределенного на данный момент). Оба этих действия лишь в малой степени могут быть реализованы базовыми средствами даже такого универсального средства моделирования, как язык UML [4], что объясняется спецификой каркасной технологии.

Рассмотрим в качестве примера простую информационную систему, задача которой – определение заработной платы преподавателя ВУЗа. Классическая логическая модель (нотация UML с минимальной демонстрацией атрибутов) показана на рис. 1.

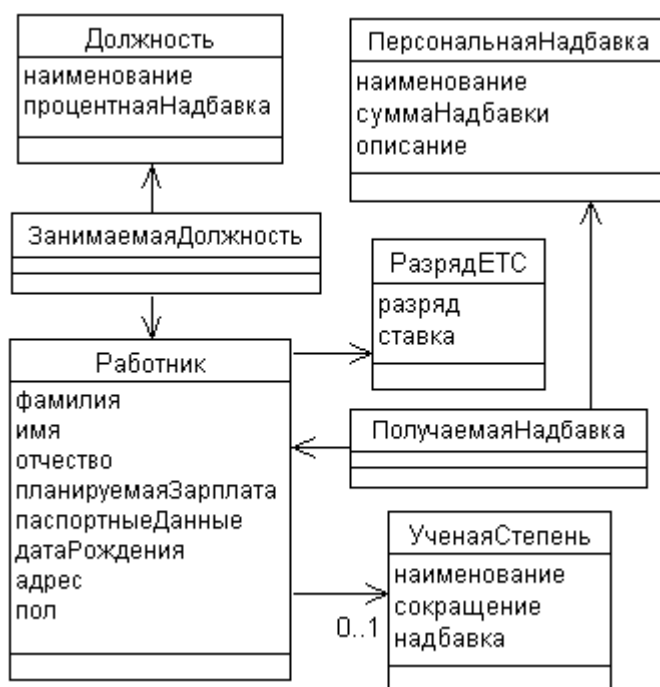


Рис. 1. Классическая логическая модель.

Согласно каркасной технологии разработки, все классы этой модели, рассматриваемые как справочники (т.е. не несущие оперативной информации, зависящей от времени), должны быть приведены к виду, показанному на рис.2. Для этого каждый из классов классической модели анализируется и классифицируется как ресурсный справочник, арматурный справочник, оперативные данные или норматив.

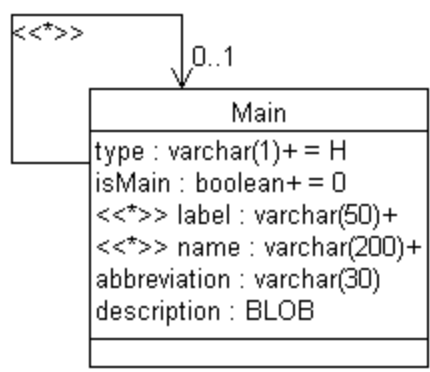


Рис. 2. Логическая модель справочников системы, выполненной по каркасной технологии.

Так как конечным вариантом модели остается логическая модель, выполненная по каркасной технологии, не может существовать каких бы то ни было компромиссов, определяющих реализацию тех или иных атрибутов классической модели. Каждый из справочников должен иметь фиксированный атрибутный набор, ограниченный наименованием, аббревиатурой и описанием в свободной форме; все прочие атрибуты класса Main являются служебными, а их заполнение и использование определяется базовыми положениями каркасной технологии.

Разработчику предлагается упрощенная нотация проектирования, выполненная на основе языка UML и позволяющая ему специфицировать наполнение каркаса реальными данными без излишней в данном случае проработки классической модели.

В первую очередь моделируются собственно справочники. Они представляются как обычные классы без демонстрации (с подавлением) атрибутов и процедур, при этом с помощью стереотипов ("ресурс", "арматура", "служебный", "неопределенный") указывается тип справочника.

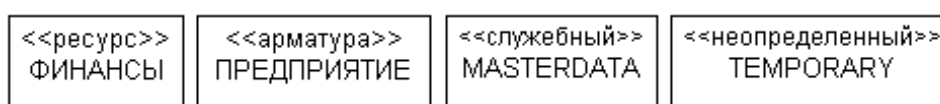


Рис. 3. Пример моделирования отдельных справочников.

Каждый из разрешенных стереотипов наследует от общего стереотипа "справочник каркасной ИС", который предполагает наличие трех смысловых атрибутов: наименования, аббревиатуры и описания. Таким образом, промоделированный на рис.3 справочник оставляет в стороне вопрос о том, какие именно атрибуты характеризуют "классический" справочник с точки зрения пользователя: этот вопрос касается только реального приложения системы и относится к обязанностям программиста, но не постановщика. В нормальном случае, учитывая изначальную направленность каркасной ИС на учет деятельности, а не ее непосредственное осуществление, оказывается достаточным формирование наименования и аббревиатуры на основе нескольких отдельных атрибутов по формуле и описания в виде регулярной структуры (см. рис.4.).

<i>Традиционное представление класса</i>	<i>Формула трансформации данных</i>	<i>Справочник каркасной технологии</i>
--	-------------------------------------	--

<i>Атрибут</i>	<i>Пример данных</i>	Наименование = фамилия + ' ' + 1(имя) + ' ' + 1(отчество) + ' ' + серия + ' ' + номер + ')'; Аббревиатура = фамилия + ' ' + 1(имя) + ' ' + 1(отчество) + ' ' + ' ' ; Описание = фамилия + имя + отчество + <перевод строки> + серия + ' ' + номер + <перевод строки> + пол + <перевод строки> + деньРожд	<i>Атрибут</i>	<i>Пример данных</i>
Фамилия	Иванов		Наименование	Иванов И.И. (36 00 111111)
Имя	Иван		Аббревиатура	Иванов И.И.
Отчество	Иванович		Описание	Иванов Иван Иванович 36 00 111111 М 01.01.1970
Серия	36 00			
Номер	111111			
Пол	М			
деньРожд	01.01.1970			

Рис. 4. Трансформация данных на примере класса "Персона".

Важно отметить, что подавляющее большинство атрибутов, которыми характеризуется класс традиционной логической модели, имеет отношение именно к осуществлению функциональной деятельности предприятия (организации), но не к ее контролю. Это положение легко иллюстрируется постановкой конкретных вопросов со стороны руководства организации. Так, в отношении класса "Персона" никогда не будут заданы, например, вопросы вида "какова эффективность деятельности работников с серией паспорта 36 00". Видно, что для учета и контроля финансово-хозяйственной деятельности предприятия представляют интерес только те характеристики любого класса, которые позволяют его идентифицировать (в примере с персоной – полное имя, расширенное паспортными данными только на случай совпадения имен у нескольких персон). Все прочие характеристики либо не представляют интереса вообще (например, не показанные на рис.4 атрибуты "имена детей", "тип прописки" и т.п.), либо представляют интерес опосредованно. Так, например, можно представить себе запрос руководителя вида "какова средняя заработная плата работников в возрасте старше 50 лет?". В этом случае атрибут "деньРождения" важен, но не сам по себе, а как стартовые данные для формирования принадлежности персоны к отдельному справочнику, называемому, например, "Возрастные группы".

Такая стандартизация всех справочников системы имеет то преимущество, что заставляет разработчика уже на стадии проектирования системы вплотную заниматься анализом всего комплекта отчетной документации, которая будет получена на основе данных системы. Классический подход ("есть данные – будут отчеты"), основанный на реализации как можно большего числа атрибутов, оказывается нереализуемым, а предлагаемая методология дисциплинирует как разработчика, так и пользователя, который, в свою очередь, вынужден обосновывать каждое предложение по расширению данных.

Автор данной статьи проводил аналогичные работы по моделированию "выжимки" информационной системы на основе получаемых из нее отчетных форм. Результаты показывают, что в действующей системе содержится до 50% данных, никогда не используемых вне системы и служащих только для внутренних целей либо сформировавшихся исторически и давно утерявших какую бы то ни было семантику.

Помимо чисто технологических преимуществ проектирования, предлагаемая методика позволяет существенно сокращать не только атрибутивный состав традиционных классов системы, но и само количество классов. Действительно, если любой класс может быть описан только тремя атрибутами (наименование, аббревиатура, описание) и если справочник, выполненный по каркасной технологии, имеет иерархическую структуру, то несколько каскадных классов, связанных отношениями типа "один-ко-многим", могут быть легко объединены в один справочник.

Рассмотрим в качестве примера информационную систему, каталогизирующую учебные подразделения ВУЗов страны с точки зрения (для наглядности будем считать, что один ВУЗ располагается только в одном населенном пункте). Традиционная

логическая модель может иметь вид, показанный на рис.5 (для атрибутов опущена демонстрация всех деталей, кроме собственно имени).



Рис. 5. Традиционная логическая модель.

При переходе к предлагаемой нотации отметим, что каждый из классов легко приводится к необходимому формату. Зная специфику системы, решающую именно задачу каталогизации, можно просто отказаться от демонстрации различной природы таких понятий, как "населенный пункт", "ВУЗ", "факультет" и "кафедра". Тогда все показанные на рис.5 классы реализуются единственным справочником "Структура" (рис. 6).

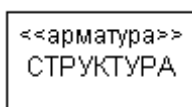


Рис. 6. Логическая модель каркасной информационной системы.

В этом случае полученная иерархическая структура данных может иметь, например, следующий вид:

1. Российская Федерация (*государство*)
 - 1.1. Самарская область (*регион*)
 - 1.1.1. Самара (*населенный пункт*)
 - 1.1.1.1. Технический университет (*вуз*)
 - 1.1.1.1.1. Механический факультет (*факультет*)
 - 1.1.1.1.1.1. Кафедра машиностроения (*кафедра*)
 - 1.1.1.2. Ставропольский район (*регион*)
 - 1.1.2.1. Тольятти (*населенный пункт*)
 - 1.1.2.1.1. Институт управления (*вуз*)
 - 1.1.2.1.1.1. Юридический факультет (*факультет*)
 - 1.1.2.1.1.1.1. Кафедра гражданского права (*кафедра*)

Видно, что предложенная структура более гибка (например, на третьем уровне вложенности и глубже реализованы параллельно объекты разной природы (например, 1.1.1 и 1.1.2)). Если такая свобода недопустима, всегда может быть введено жесткое ограничение, запрещающее использование свободных структур. В крайнем же варианте каркасная ИС всегда может включать количество справочников, повторяющее количество классов традиционной модели.

Моделирование систем по каркасной технологии было практически апробировано как в работах уровня дипломного проектирования, так и при выполнении реальных контрактов по информатизации предприятий и учреждений (например, при

автоматизации Дорожной Клинической Больницы станции Самара Куйбышевской железной дороги). Результаты внедрения показывают, что при сохранении тех же сроков проектирования, что и при традиционном подходе, результирующая система оказывается освобожденной от большого количества ненужной информации, а типовые (т.е. заранее сформированные и готовые к формированию еще до проектирования системы) отчетные формы покрывают все потребности заказчика по обобщению информации. Помимо моделирования собственно справочников (классов), использовалось моделирование всех типов связей между ними, что должно стать темой отдельных публикаций.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Караваев М.А., Лыноградский Ю.Л.* Об одном аспекте задачи администрирования данных. – "Вестник Самарского Государственного Технического Университета", серия "Технические науки" (вып. 15), -Самара, 2002.
2. *Лыноградский Л.А.* Горизонты системного анализа.- Самара, ИЭКА "Поволжье",- 2000.
3. *Лыноградский Ю.Л.* Об одном подходе к организации эволюционирующих справочников. "Вестник Самарского Государственного Технического Университета", серия "Технические науки" (вып. 15), -Самара, 2002.
4. *Ivar Jacobson, Grady Booch, James Rumbaugh.* Unified Modeling Language User Guide. - Addison-Wesley Longman, 1999 [англ.]