

## РАЗЛИЧНЫЕ ПОДХОДЫ В МОДЕЛИРОВАНИИ ПРЕДМЕТНОЙ ОБЛАСТИ НА ПРИМЕРЕ МОДУЛЯ РЕАЛИЗАЦИИ ЗАДАЧ СМК

С.А. Ермоченко, О.Г. Казанцева, Л.В. Маркова  
Витебск, УО «ВГУ им. П.М. Машерова»

При построении информационных систем первым этапом разработки является проектирование архитектуры проекта, которое начинается с анализа предметной области. В общем виде моделирование предметной области описать практически невозможно, так как существует несколько методологий моделирования [1]: структурная, объектно-ориентированная, функциональная и т.д. Для каждой из них существуют свои специализированные подходы. Целью работы является изучение подходов в моделировании предметной области для модуля системы «Электронный университет», предназначенного автоматизировать решение задач системы менеджмента качества университета. При реализации модуля было принято решение использовать объектно-ориентированную методологию, при этом для хранения объектов предметной области используется база данных под управлением реляционной СУБД. Рассмотрим различные способы описания предметной области в виде классов и таблиц базы данных.

**Результаты и их обсуждение.** Наиболее простым и часто используемым подходом является описание отдельного класса для каждого понятия предметной области. Далее, каждому классу ставится в соответствие таблица базы данных. Таким образом, каждой строке таблицы соответствует один объект описанного класса. Например, будем считать, что описаны классы «процесс», «целевой показатель», «ответственность за заполнение целевого показателя», «значение целевого показателя». Данные классы связаны со структурными подразделениями и между собой. Отображение этой связи можно представить в виде UML диаграммы классов (см. рис. 1).

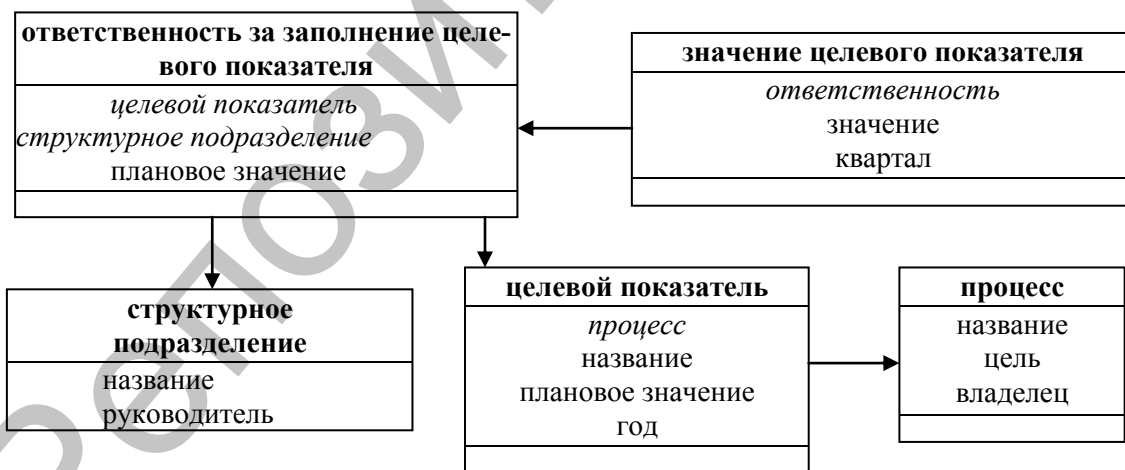


Рисунок 1 – Диаграмма классов–сущностей модуля реализации задач СМК

Проблемой такого подхода является необходимость хранения всех значений целевых показателей за предыдущие года. Решить проблему можно, добавив поле «год» в класс «целевой показатель». Но при изменении структуры университета, например при разделении структурного подразделения на два или объединении структурных подразделений, невозможно будет восстановить целевые показатели за предыдущий период.

Таким образом, если при моделировании предметной области классы и таблицы базы данных хранят информацию о каждом понятии предметной области, то в каждый момент времени можно получить доступ лишь к актуальной на текущее время информации.

Одним из вариантов хранения всех предыдущих состояний является полное дублирование данных с определенным интервалом времени. Но, во-первых, этот подход приводит к лавинообразному росту объемов хранимой информации, а во-вторых, нельзя гарантировать, что между двумя экземплярами база данных осталась неизменной.

Другим вариантом решения проблемы может быть создание дополнительных «архивных» таблиц, в которых сохраняются предыдущие значения изменяемых сущностей. Этот вариант позволяет создавать дубликаты записей лишь при необходимости, но при этом не гарантирует, что в базе данных не будет храниться избыточная информация. Продемонстрировать избыточность можно на примере класса «процесс». Так, при смене владельца процесса, можно в архивную таблицу заносить старые данные о процессе, но при этом сохраняться будет и название процесса и его цель, которая является строковым полем большого размера. Наряду с упомянутой избыточностью данных стоит помнить, что наличие нескольких таблиц с одинаковой структурой, соответствующих одному классу-сущности, значительно снижает понимание архитектуры приложения, что создает дополнительные сложности на этапе разработки и увеличивает вероятность возникновения ошибок в проекте.

В качестве альтернативы подходу, основанному на хранении данных о понятиях предметной области, можно предложить подход, основанный на хранении информации о событиях, происходящих с объектами из предметной области. Изначально такой подход использовался при разработке информационных систем архивов [2]. Так, для хранения семейного архива можно использовать структуру классов, показанную на рис. 2. На этой диаграмме представлены классы «тип события», который описывает все возможные в системе события, и «событие», который описывает конкретное произошедшее событие. Дополнительную информацию о произошедшем событии выносят в отдельные классы (и таблицы), при этом по аналогии с выделением классов «роль» и «участник» создаются классы, характеризующие те типы данных, которые должны характеризовать каждый тип события, и классы, хранящие данные о каждом конкретном событии.

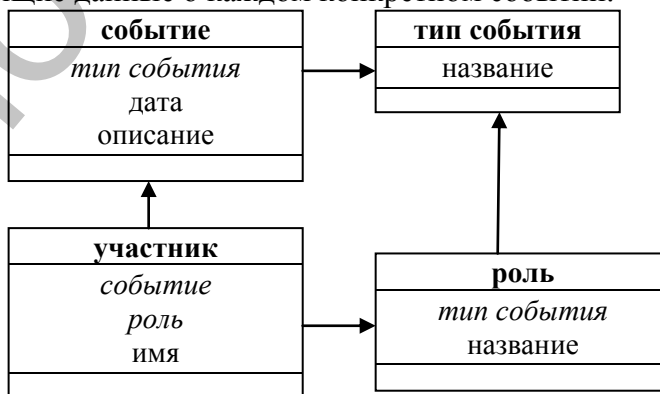


Рисунок 2 – Диаграмма классов для хранения событий и их участников

При этом данный подход позволяет получить данные за любой период путем просмотра событий нужного типа за требуемый период. Минусом предложенного подхода является большая вычислительная сложность при поиске необходимых данных. При этом если задержка при запросе данных за предыдущие го-

да приемлема, то работа с текущим годом должна происходить достаточно быстро. Еще описанный вариант содержит потенциальную возможность хранения несогласованных данных. Действительно, объект класса «участник» хранит ссылку на объекты класса «событие» и «роль», каждый из которых ссылается на объект класса «тип события», но механизмы реляционной СУБД не могут гарантировать, чтобы это был один и тот же объект. Проверку согласованности данных необходимо при этом дополнительно проводить программным способом, что увеличивает сложность разработки.

**Заключение.** В работе рассмотрены два подхода в моделировании предметной области, один основан на хранении информации о понятиях предметной области, другой – на информации о событиях, возникающих с объектами предметной области. В явном виде ни один из этих подходов не может быть применен, так как влечет за собой либо неоправданные вычисления, либо чрезмерный объем хранимых данных. Объединение этих двух подходов позволит увеличить скорость работы с данными за текущий год, если хранить и обрабатывать данные о самих объектах предметной области, а, с другой стороны, позволит сократить объем хранимых данных, если для архива использовать событийную модель.

#### Список литературы

1. Буч, Г. Объектно-ориентированный анализ и проектирование с примерами приложений на С++, 2-е изд. / Г. Буч. – СПб: Бинум, 1999.
2. Программа «Наследие» [Электронный ресурс] // Архивное дело. – 2012. – Режим доступа: <http://www.1archive-online.com/partners/nasledia.htm>. – Дата доступа: 4.02.2012

### СОЗДАНИЕ МОДУЛЯ РЕАЛИЗАЦИИ ЗАДАЧ СМК: РАЗРАБОТКА, ФУНКЦИОНИРОВАНИЕ, СОВЕРШЕНСТВОВАНИЕ

*О.Г. Казанцева, С.А. Ермоченко, Л.В. Маркова  
Витебск, УО «ВГУ им. П.М. Машерова»*

В 2011–2012 учебном году кафедра прикладной математики и механики принимает участие в программе «TEMPUS: внедрение инструментов и политики для улучшения качества образования на институциональном уровне». В рамках этой программы осуществляется разработка программного обеспечения (web-приложения), которое предназначено для мониторинга целевых показателей университета и всех структурных подразделений, входящих в его состав.

**Результаты и их обсуждение.** Создание сложного и качественного программного продукта предполагает командную форму работы по реализации программного обеспечения и документирования процесса разработки создаваемой системы на всех этапах – от бизнес-идеи до внедрения.

На первом этапе работы нами были подготовлены и согласованы с отделом качества основные функциональные требования к модулю СМК «Мониторинг целевых показателей в области качества УО «ВГУ им. П.М. Машерова», определены роли пользователей в системе и разработана архитектура приложения. Изначально было понятно, что разработка данного программного продукта будет представлять собой итерационную модель программного обеспечения, поскольку на любом этапе возможны изменения или уточнения требований к приложению. Итерационная модель предполагает прохождение полного цикла разработки программы на каждой итерации: определение функциональных требований, необходимых для реализации на текущем этапе; проектирование и разработка функцио-