

Томский государственный университет  
Кемеровский государственный университет  
Кемеровский научный центр СО РАН  
Институт вычислительных технологий СО РАН  
Филиал Кемеровского государственного университета  
в г. Анжеро-Судженске

**ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ  
И МАТЕМАТИЧЕСКОЕ  
МОДЕЛИРОВАНИЕ  
(ИТММ-2011)**

**Материалы X Всероссийской  
научно-практической конференции  
с международным участием  
25–26 ноября 2011 г.**

**Часть 1**

Издательство Томского университета

2011

**УДК 519**  
**ББК 22.17**  
**И74**

**Редакционная коллегия:**

*Р.Т. Якупов*, д-р физ.-мат. наук, профессор;  
*А.А. Назаров*, д-р техн. наук, профессор;  
*И.Р. Гарайшина*, канд. физ.-мат. наук, доцент

**И74 Информационные технологии и математическое моделирование** (ИТММ-2011): Матер. X Всерос. науч.-практ. конфер. с междунар. участием (25–26 ноября 2011 г.). – Томск: Изд-во Том. ун-та, 2011. Ч. 1. 192 с.

ISBN 978-5-7511-2031-3

В часть 1 вошли материалы секций «Информационные технологии» и «Вероятностные методы и модели».

Для специалистов в области информационных технологий и математического моделирования.

**УДК 519**  
**ББК 22.17**

ISBN 978-5-7511-2031-3 © Томский государственный университет, 2011  
© Кемеровский государственный университет, 2011  
© Кемеровский научный центр СО РАН, 2011  
© Институт вычислительных технологий СО РАН, 2011  
© Фил-л Кемеровского государственного университета  
в г. Анжеро-Судженске, 2011

*Конференция проводится при поддержке Российского фонда фундаментальных исследований (проект № 11-07-06076-з)*

5. Антонов А.С. Параллельное программирование с использованием технологии OpenMP. М.: Изд-во МГУ, 2009. 78 с.

6. Антонов А.С. Параллельное программирование с использованием технологии MPI. М.: Изд-во МГУ, 2004. 72 с.

7. *Интуит*. Лекция: Параллельные методы решения дифференциальных уравнений в частных производных. [Электронный ресурс]. URL: <http://www.intuit.ru/department/calculate/paralltp/11/> (дата обращения: 08.03.2011).

## СЛОЙ ДАННЫХ ДЛЯ МОДЕЛИ А. ТЕНЦЕРА

*К.А. Герасимов*

*Томский государственный университет*

Всем нам каждый день, так или иначе, приходится работать с большими системами как в Internet, так с другими системами, через банкоматы, терминалы оплаты услуг и т.д. И сложно себе представить такую систему, которая бы не работала с базой данных.

Исторически сложилось так, что наибольшую популярность приобрели именно реляционные базы данных. В этой области сделаны большие наработки, отличная производительность, кроссплатформенность, математический и логический аппараты. И все бы хорошо, пока бизнес-логика приложений остается относительно не сложной, и темпы эволюционирования приложений были не высоки.

В процессе технического развития требования к приложениям стали больше, а бизнес-логика сложнее. К функциональному и процедурному проектированию добавилось объектно-ориентированное, где в центре внимания уже объект. Это позволило изрядно упростить приложения и легко их изменять вслед за усложнением бизнес-логики. А часто встречающиеся сложности решались с помощью типовых решений, так называемых паттернов. Но базы данных по-прежнему оставались и остаются на данный момент реляционными.

Объекты в реляционной базе данных представляются в виде карточка, состоящего из значений атрибутов отношения соответствующего множества сущностей конкретной предметной области. А это означает, что для конкретной предметной области нужна своя схема базы данных, однако основная сложность заключается в том, что внутреннее представление объектов в приложении и базе данных различны, то есть необходимо преобразование. В данном направлении проделано множество исследований и предложено несколько решений, такие как активная запись (active record), преобразователь данных (data mapper) и другие.

Одно из решений предложил Анатолий Тенцер. Он представил совершенно иной способ хранения объектов в базе данных. Хранить классы, поля, принадлежащие классу, объекты и значения полей объектов. Как и большинство универсальных решений, это решение имеет не только плюсы, но и минусы.

В данной работе был спроектирован слой данных (Data layer), состоящий из служб, которые будут реализованы в виде классов, фасад

(Facade), отделяющий внутреннюю структуру слоя и предоставляющий необходимый интерфейс для работы с данным слоем. Класс редактор (Editor) отвечает за классы и их структуру. Работник (Worker) обслуживает экземпляры классов. Класс интерфейс к базе данных (DBInterface) и класс "супертип слоя" (SLType), на который возлагается процесс инициализации и настройка интерфейса к базе данных.

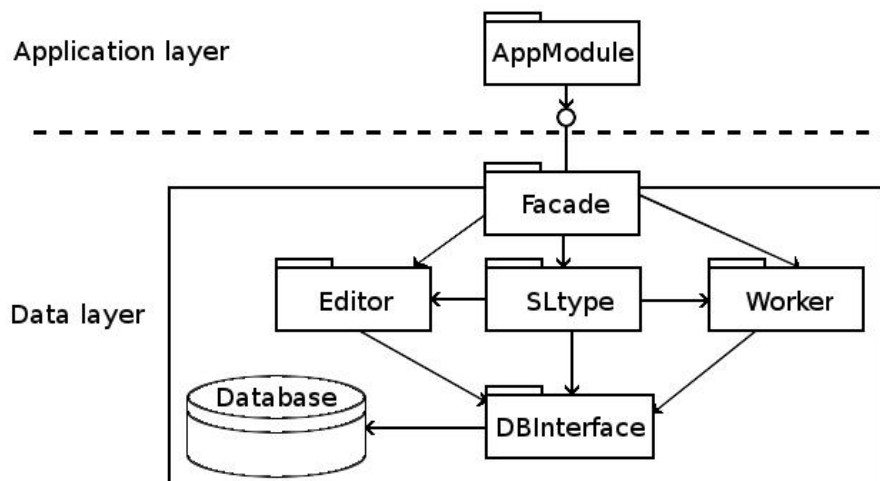


Рис. 1. Структура слоя данных

Ниже представлены основные проблемы и решения данной модели:

1. Ослабленный контроль за целостностью базы данных. Этот контроль осуществлен за счет бизнес-логики на уровне приложения.

2. Пониженное быстродействие при выборке данных. Частично решено при использовании типового решения "загрузка по требованию". При выборке большого множества объектов, объекты будут содержать минимальную информацию, а при просмотре конкретного или внесения изменений не загруженных полей, объект будет полностью инициализирован.

3. Сложность в понимании структуры и выборки данных. Решение полностью возлагается на класс Worker. При получении объекта типа Запрос, формируют запрос для конкретного интерфейса базы данных.

**Заключение.** Спроектирован слой данных, предоставляющий единый интерфейс для работы с объектами и поддерживающий различные интерфейсы.

Следует заметить, что, используя данное решение, разработчику необходимо ознакомиться с классом Запрос и, в случае использования редких структур хранения данных (баз данных), создать наследника от класса DBInterface, и реализовать необходимые операции.

#### Литература

1. Гамма Э. Библиотека программиста: примеры объектно-ориентированного проектирования: паттерны проектирования. М.: Санкт-Петербург, 2001. 344 с.
2. Объектно-ориентированный анализ и проектирование [OOA&П/ООA&D]. [Электронный ресурс]. URL: <http://ooad.asf.ru/Pattern.aspx?IdKat=7&IdPat=43> (дата обращения 05.09.2011).

3. Первый в России ежемесячный компьютерный журнал КомпьютерПресс. [Электронный ресурс]. URL: <http://www.compress.ru/article.aspx?id=11515&iid=452> (дата обращения 06.09.2011).

## **ИНФОРМАЦИОННАЯ МОДЕЛЬ ДЕЯТЕЛЬНОСТИ КАФЕДРЫ**

*А.М. Гудов, С.Ю. Завозкин, И.В. Огородникова, Е.Д. Пфайф*

*Кемеровский государственный университет*

В работе предлагается рассмотреть решение задачи автоматизации деятельности кафедры. Отличительными чертами предлагаемой информационной системы (ИС) являются:

- использование в системе всех основных принципов интеграции;
- в основе ИС лежит понятие электронного документа;
- универсальность для использования в любой среде.

Проведенный анализ существующих систем показал, что в данный момент на рынке существует большое число систем, автоматизирующих работу кафедры ВУЗа, обладающих, как правило, схожими базовыми функциями. Главной проблемой таких систем является недостаточная развитость имеющихся средств интеграции с другими системами. Так же немаловажной проблемой является отсутствие возможности автоматического формирования документации связанной с деятельностью кафедры. Требования к документации, кафедры часто меняются, поэтому должен быть разработан универсальный подход к формированию документооборота подразделения. Предлагается для создания ИС «Кафедра» разработать модель, которая должна учесть описанные проблемы.

Создаваемая модель должна поддерживать три принципа интеграции [1]:

- информационно-ориентированный, основанный на использовании одной и той же информации двумя и более системами. Для обеспечения работы со своей информацией у каждой системы имеется набор открытых сервисов. Данный принцип реализуется на уровнях интеграции приложений.

- сервисно-ориентированный, основанный на использовании стандартизованного описания формата передачи данных, которые хранятся в единой базе данных (БД) ПО промежуточного слоя, и имеется набор сервисов для работы с ними. Данный принцип реализуется на уровнях интеграции данных, платформ и на уровне использования стандартов интеграции.

- процессно-ориентированный, основанный на возможности присоединения к внутренним прикладным процессам каждой ИС таким образом, чтобы не просто использовать их функции, а создать новый бизнес-процесс (БП), который бы связал эти ИС. Данный принцип реализуется на уровнях интеграции БП, платформ и на уровне использования стандартов обмена данными.