

УДК 004.78

В. Ю. АРЬКОВ, М. Н. МАМЧУР

## МОДЕЛЬНО-ОРИЕНТИРОВАННЫЙ ПОДХОД К ПРОЕКТИРОВАНИЮ ИНФОРМАЦИОННОЙ СИСТЕМЫ УПРАВЛЕНИЯ ФИЛИАЛОМ ВУЗА

Обсуждаются результаты применения модельно-ориентированного подхода Model Driven Architecture (MDA) к проектированию информационной системы управления филиалом вуза. Предлагается методика быстрой разработки и интеграции гетерогенных приложений на основе MDA. Разработан ряд собственных MDA-приложений, интегрированных в КИС управления филиалом вуза. *Корпоративные информационные системы; быстрая разработка приложений; базы данных*

### ВВЕДЕНИЕ

Инновации в управлении образовательным учреждением на базе информационных технологий являются ключевым механизмом, который позволит создать преимущества в конкурентной среде. В этой связи основными мероприятиями в развитии информатизации становятся создание ее надежной и эффективной инфраструктуры, внедрение унифицированных способов доступа к корпоративным данным, улучшение управляемости всего комплекса информационных ресурсов, а также обеспечение соответствия инфраструктуры стратегическим целям вуза. Комплексная реализация данных мероприятий может быть увязана с формированием корпоративной информационной среды (КИС), что обеспечит интеграцию информационных ресурсов и позволит создать информационную инфраструктуру в соответствии с действующей организационной структурой и принятыми бизнес-правилами. На современном этапе информационная среда из средства предоставления доступа к необходимой информации превратилась в обязательный компонент инфраструктуры управления вузом и совокупность интеллектуальных сервисов, без которых сегодня невозможно представить организацию управления и обучения в вузе.

Как правило, проекты по внедрению систем автоматизации управленческой деятельности вузов охватывают широкий спектр задач от дополнительной формализации процедур сбора и хранения управленческой информации до осуществления изменений в организационной структуре управления и перераспределения обязанностей. Ключевой особен-

ностью такого типа проектов является то, что от успеха или провала проекта может зависеть как эффективность функционирования подразделений, так и всего вуза в целом [1].

Опираясь на основные виды деятельности вуза (образовательная, научно-исследовательская, производственная), можно выделить два основных блока задач, на которые должна быть направлена автоматизация:

- управление учебным процессом и научно-исследовательскими работами;
- административно-хозяйственное управление и управление финансами.

Как правило, финансовые и бухгалтерские задачи во многих вузах решены на основе стандартных коммерческих приложений для бюджетных организаций и представляют интерес только на уровне интеграции в рамках КИС, а вот первый блок задач является объектом исследований как с точки зрения реализации, так и с точки зрения интеграции.

Одна из проблем в области автоматизации — сравнительно малый срок эксплуатации КИС по сравнению с длительностью ее проектирования. В частности, ведущие западные компании регулярно обновляют свои КИС раз в 5–10 лет, что соответствует затратам времени на проектирование и внедрение КИС на крупных российских предприятиях [2]. В этом состоит одно из отличий сферы автоматизации от материального производства. Поэтому RAD-технологии быстрой разработки, позволяющие оперативно и безболезненно вносить изменения в готовый или почти готовый проект на всех уровнях, включая и уровень требований, необходимы как никогда.

С одной стороны, когда речь заходит об автоматизации филиала вуза, ограниченные ресурсы не позволяют ставить «эксперименты по автоматизации». С другой стороны, автоматизация филиала существенно ускоряется из-за сравнительно малых масштабов и компактности подразделений. В данном случае можно использовать технологии быстрой разработки приложений, что сокращает сроки проектирования по сравнению с длительностью эксплуатации.

Учитывая вышесказанное, при разработке КИС управления вузом перед разработчиком встают две основные проблемы:

- интеграции уже готовых приложений, реализованных на основе различных технологий и архитектур;
- быстрого создания приложений, не связанных к конкретной платформе.

В рассматриваемой статье обсуждаются результаты применения модельно-ориентированного подхода к проектированию информационной системы управления филиалом вуза. При этом рассматриваются две группы вопросов:

- интеграция приложений, реализованных на основе различных технологий, архитектур и решений посредством модельно-ориентированного подхода,
- практическое применение модельно-ориентированного подхода к разработке одной из подсистем КИС управления филиалом вуза.

#### МОДЕЛЬНО-ОРИЕНТИРОВАННЫЙ ПОДХОД К РАЗРАБОТКЕ КИС

Эволюция программирования сопровождается повышением уровня абстракции написания программ — от машинных кодов к ассемблеру, от ассемблера к высокоуровневому языку. Поэтому потребность в интеграции и обеспечении взаимодействия систем, основанных на разных технологиях, а также в модернизации существующих программ и их переработке в соответствии с новой технологической основой возникает все чаще.

Существующие CASE-технологии (Computer Aided Software/Systems Engineering) позволяют автоматизировать процессы создания программного обеспечения и проектирования сложных систем. Можно отметить два основных направления в области CASE-технологий: структурный (IDEF) и объектно-ориентированный (UML) [3].

Дальнейшим развитием объектно-ориентированного подхода является разработка ин-

струментальных средств модельно-ориентированной технологии (MDA — Model Driven Architecture), предложенной консорциумом OMG в 2002 г. [4]. Во-первых, модельно-ориентированный подход облегчает интеграцию как разнородных, основанных на различных технологиях, распределенных систем, так и организацию взаимодействия между системами, основанными на одной технологии, но использующих разные интерфейсы и стандарты. Во-вторых, технология MDA дает возможность использовать модели для генерации кода, что существенно снижает время на разработку и уменьшает число ошибок за счет отказа от ручного кодирования.

Основу MDA составляет понятие платформенно-независимой модели (platform independent model, PIM), которая является детально исполняемой моделью на языке действий с пред- и постусловиями на языке UML. Такая модель не зависит от платформ и технологий и фактически является каркасом основной системы, описанным независимым от технологий реализацией образом (язык программирования, распределенные технологии, доступ к базам данных и т. д.). Эта модель воспроизводит логику, основные задачи и принципы функционирования приложения. PIM-модель — это модель тестирования и отладки, позволяющая затем вносить необходимые уточнения в интересах конкретной платформы. Построенная PIM-модель преобразуется в платформенно-зависимую модель (platform specific model, PSM). Это преобразование производится с помощью стандартных отображений, разных для каждой платформы промежуточного слоя — CORBA, Java, Net, Web-службы и другие. Благодаря тому, что преобразование от PIM к PSM стандартизировано, создаются анализаторы и генераторы описания моделей, существенно упрощающие это преобразование.

Реализуется технология MDA посредством программных продуктов: Bold for Delphi, который входит в состав Borland Delphi 7 Studio Architect и Borland ECO (Enterprise Core Objects), входящей в состав C# Builder Architect. Приложения, рассматриваемые в данной статье, разработаны в Borland Delphi 7 Studio Architect.

#### ИНТЕГРАЦИЯ ГЕТЕРОГЕННЫХ ПРИЛОЖЕНИЙ

Как было отмечено выше, при разработке КИС возникает необходимость в интеграции как уже готовых программных решений, так и приложений, разрабатываемых в рамках бу-

дущей системы. Причем необходимо учитывать, что в пределах корпоративной информационной системы обмен данными может осуществляться на двух уровнях: на уровне передачи готовой информации (отчеты, файлы, документы и т. д.) и на уровне внутрикорпоративного хранилища данных. Основу последнего составляют, как правило, базы данных, построенные на основе различных СУБД. При этом построение единой КИС требует решения двух проблем: интеграции приложений на уровне СУБД и интеграции данных на уровне передачи исходящей информации.

В рамках проделанной работы были исследованы механизмы, которые предоставляет технология MDA для решения намеченных проблем.

### Работа с несколькими СУБД

Интеграцию приложений, реализованных на основе различных СУБД, согласно технологии MDA, осуществляют два механизма: дескрипторы и адаптеры.

Дескрипторы объектного пространства играют центральную роль при разработке MDA-приложений. Они представляют собой специальные невидимые компоненты, реализующие бизнес-уровень приложения и обеспечивающие взаимодействие с уровнем данных и графическим интерфейсом пользователя.

Для обеспечения работы с конкретными типами СУБД в состав Bold for Delphi встроены специальные компоненты-адаптеры баз данных.

Таким образом, для преобразования ранее созданного приложения для работы с другой СУБД необходимо выполнить следующие действия:

- построить «копию» набора компонентов для работы с новой СУБД,
- добавить в модуль данных еще один системный дескриптор, а также дескриптор и адаптер новой СУБД,
- создать новую пустую базу данных, и настроить соответствующие свойства для соединения с нею,
- подключить новый дескриптор уровня данных к ранее созданной модели и сгенерировать схему новой базы.

### Использование XML-документов

Кроме всего прочего MDA обеспечивает сохранение состояния объектного пространства

в файле формата XML без ограничений на вид информации. Таким образом, имеется возможность сохранять не только текстовые данные, но и фотографии, документы Word, PDF-документы и т. д., что существенно расширяет возможности обмена информацией между подсистемами, поддерживающими открытые форматы.

### КОРПОРАТИВНАЯ ИНФОРМАЦИОННАЯ СРЕДА ФИЛИАЛА УГАТУ В НЕФТЕКАМСКЕ

В Нефтекамском филиале УГАТУ на основе модельно-ориентированного подхода разрабатывается КИС, включающая приложения, приведенные в табл. 1 (в первом столбце кодами обозначено: «к» — коммерческое решение, «с» — собственная разработка).

Безусловно, в табл. 1 приведена лишь малая часть приложений, свойственных полномасштабной КИС управления вузом, но, учитывая трехлетнюю историю филиала и трудности, связанные с ограниченностью ресурсов, это дает надежду на реализацию всех поставленных задач в ближайшее время.

Как видно из табл. 1, КИС филиала объединяет различные технологии, архитектуры и решения. Кроме того, разработка КИС началась параллельно с внедрением приложений, не учитывающих требования самой КИС. Поэтому проблема интеграции уже функционирующих локальных автоматизированных систем стоит гораздо острее, нежели проблема интеграции систем, находящихся на стадии разработки.

### РАЗРАБОТКА ПОДСИСТЕМЫ «УЧЕБНАЯ НАГРУЗКА»

Продемонстрируем вышеописанный подход на конкретном примере разработки модуля «Расчет учебной нагрузки».

Рассмотрим функциональную схему задачи с точки зрения действующих в ней информационных потоков (рис. 1).

Как видно из схемы, учебный план является одной из основополагающих задач при создании модуля «Расчет учебной нагрузки».

Опишем неформально модель «Учебный план» набором бизнес-правил:

- описываемая предметная область состоит из множества дисциплин, циклов и семестров;
- цикл однозначно идентифицируется названием;

- дисциплина идентифицируется названием и может читаться (изучаться) на протяжении нескольких семестров;
- одна дисциплина не может входить в несколько циклов;
- учебный план составляется для конкретной специальности, которая однозначно идентифицируется кодом специальности;
- специальность может состоять из специализаций;
- учебные планы одной и той же специальности, но разных специализаций различа-

ются только содержанием одного цикла «Дисциплины по выбору»;

- каждая дисциплина закреплена за конкретной кафедрой.

На первом этапе разработки MDA-приложения создается бизнес-уровень. Для этого необходимы три компонента:

- **BoldModel1** (компонент, обеспечивающий хранение модели);
- **BoldSystemHandle1** (основной компонент — дескриптор объектного пространства);
- **BoldSystemTypeInfoHandle1** (основной компонент — дескриптор типов модели).

Таблица 1

## Информационные системы и сервисы КИС филиала УГАТУ в г. Нефтекамске

Код	Контур информатизации	Используемые решения
<i>I. Административное управление и управление финансами</i>		
К	Управление персоналом и организационной структурой	«1С: Предприятие», 1С
К	Учет труда и материальный учет	«1С: Бухгалтерия», 1С
С	Управление договорной деятельностью и начисление стипендии	Delphi (MDA), Interbase
<i>II. Управление учебным процессом</i>		
С	Ведение приемной компании	Delphi (MDA), Interbase
С	Управление студенческим составом	Delphi (MDA), Interbase
С	Расчет учебной нагрузки	Delphi (MDA), Interbase
<i>III. Учебный процесс</i>		
К	Библиотечная система, включающая электронный каталог литературы и средства контроля выдачи литературы	RUSLAN
<i>IV. Управление информационными ресурсами КИС</i>		
К	Система управления файловой службой	Active Directory
С	Корпоративный портал филиала	.Net C#, Interbase

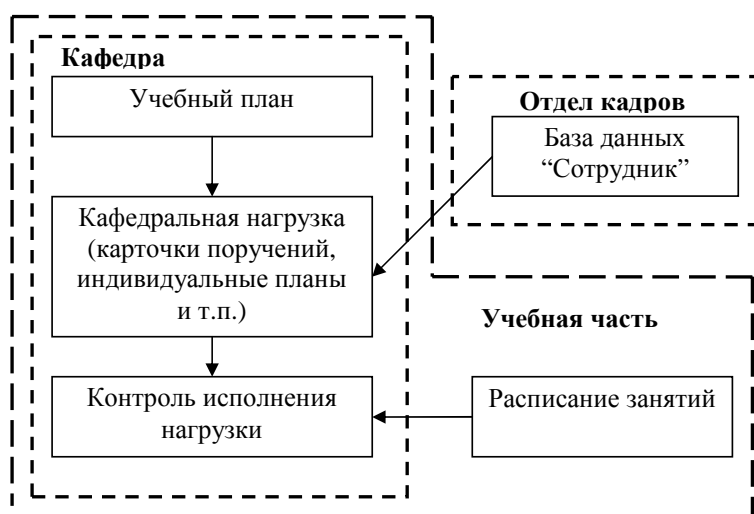


Рис. 1. Функциональная схема задачи «Расчет учебной нагрузки кафедры»

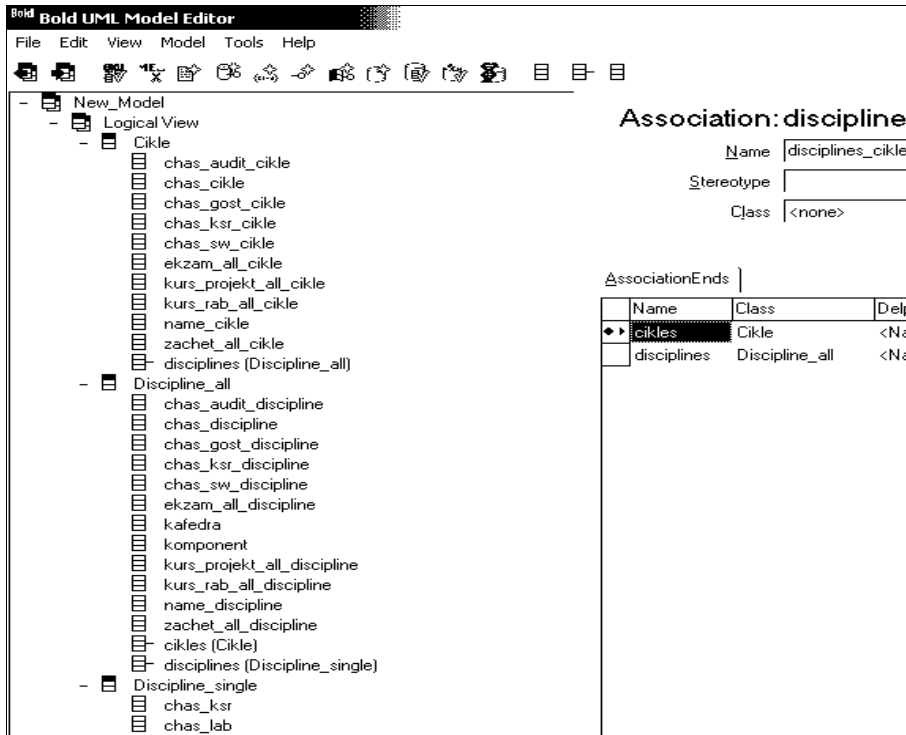


Рис. 2. Вид модели «Учебный план»

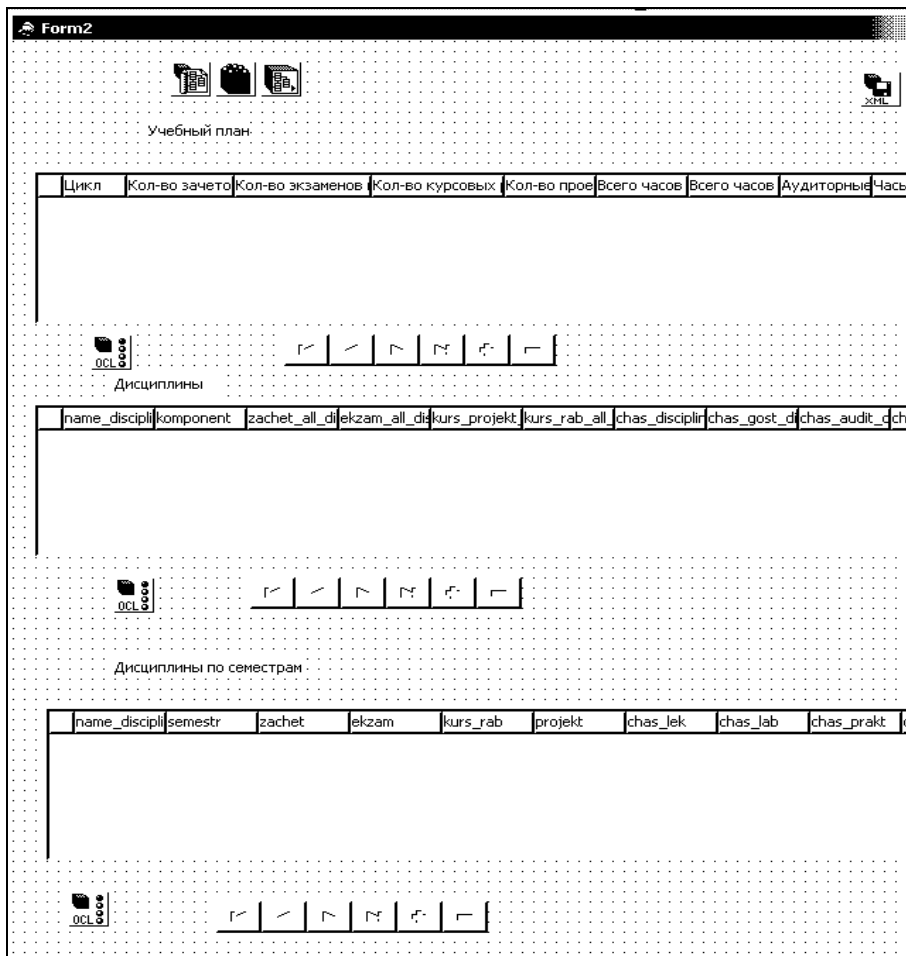


Рис. 3. Графический интерфейс приложения «Учебный план»

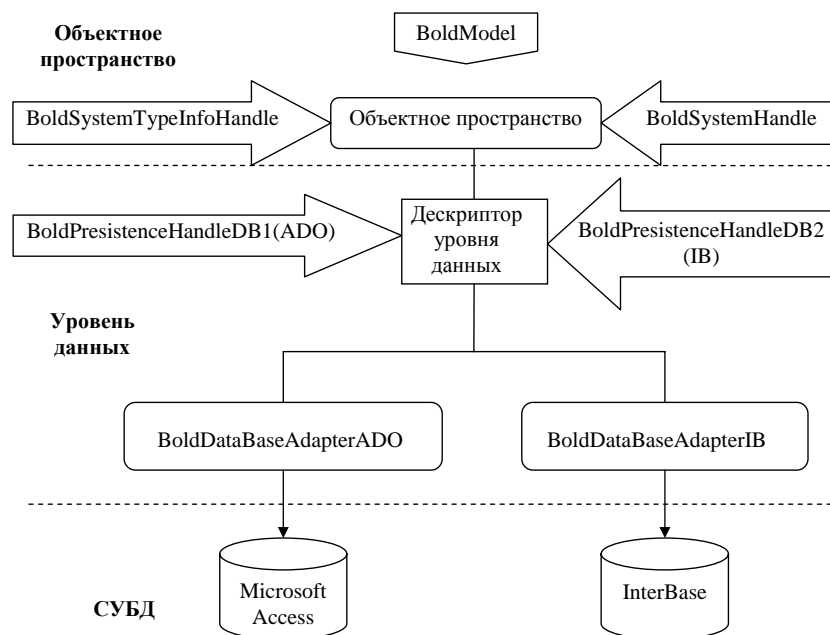


Рис. 4. Состав уровня данных для двух СУБД приложения «Учебный план»

Эти компоненты реализуют основу «объектного пространства» (Object Space). Однако бизнес-уровень не наполнен функциональным содержанием до тех пор, пока не создана модель приложения, в соответствии с которой оно будет работать.

На следующем этапе создается модель приложения (PIM-модель). Для этого в Bold for Delphi имеется встроенный редактор моделей, который позволяет создавать UML-модели (в текстовом режиме), а также обеспечивает реализацию основных функциональных возможностей Bold на этапе разработки приложения, таких как экспорт и импорт моделей из сторонних UML-редакторов, модификация, верификация и тонкая настройка моделей, генерация баз данных и кода.

Созданная модель приложения «Учебный план» представлена на рис. 2. Слева расположено дерево модели, в котором находятся классы, атрибуты и ассоциации, реализующие описанные выше бизнес-правила, в правой части представлены свойства активизированного в текущий момент элемента модели.

Следующий этап разработки приложения подразумевает создание графического интерфейса (рис. 3). Для этого в Bold имеется множество визуальных компонентов, аналогичных используемым при создании приложений баз данных при объектно-ориентированном подходе.

Заключительный этап разработки Bold-приложения предполагает создание уровня

данных (Persistence Layer), который обеспечивает выполнение следующих функций:

- сохранение элементов объектного пространства в долговременной памяти;
- загрузка элементов объектного пространства из долговременной памяти;
- поддержка механизма объектно-реляционного отображения, т. е. преобразование объектной UML-модели в структуру реляционной базы данных;
- генерация схемы реляционной базы данных по имеющейся объектной модели;
- преобразование OCL-выражений в операторы SQL.

Рассмотрим здесь же возможность получить доступ к базе данных «Учебный план» посредством приложения, реализованного на СУБД Microsoft Access.

Наше приложение «Учебный план» функционирует в СУБД InterBase, а Microsoft Access базируется на интерфейсе ADO. Ранее в данной статье была приведена последовательность действий по интеграции гетерогенных приложений на уровне СУБД. Состав и структура уровня данных, функционирующего на основе двух СУБД, показаны схематично на рис. 4.

Таким образом, в этой главе мы детально показали ход разработки MDA-приложения «Учебный план», работающего с двумя СУБД.

Процеируя рассмотренный пример на большие корпоративные информационные системы, содержащие сотни классов и тысячи

атрибутов, можно в полной мере представить насколько кардинально повышается эффективность разработки и сопровождения таких систем [4].

Ожидаемые результаты внедрения информационных технологий управления носят косвенный характер и не выражаются напрямую в терминах финансовой эффективности. В частности, оптимизируется учебный процесс, ускоряется процесс принятия решений руководством. Кроме того, повышается обоснованность управленческих решений, принимаемых на основе объективной информации, создаются предпосылки для контроля реализации принятых решений в соответствии методологией менеджмента качества.

### ВЫВОДЫ

Предложена методика быстрой разработки корпоративных информационных систем на основе модельно-ориентированного подхода MDA. Модельно-ориентированный подход представляет собой дальнейшее развитие CASE-технологий. В отличие от структурного и объектно-ориентированного подходов, MDA позволяет генерировать приложения, работающие с реляционными базами данных.

Предложена методика интеграции гетерогенных приложений при построении корпоративной информационной системы. Интеграция приложений различных СУБД производится с помощью дескрипторов и адаптеров MDA. Интеграция данных выполняется на уровне передачи исходящей информации за счет сохранения состояния объектного пространства в файлах XML.

На основе предложенных методик, разработан ряд собственных MDA-приложений, интегрированных в КИС управления филиалом вуза. В данном случае необходимо обеспечивать интеграцию функционирующих локальных автоматизированных систем в рамках единой КИС, что позволяет ускорить разработку системы.

На примере разработки подсистемы «Учебная нагрузка» продемонстрированы особенности применения MDA, включая планирование компонентов объектного пространства, создание PIM-модели приложения, проектирование графического интерфейса и реализацию уровня данных. При этом

MDA-приложение оперирует одновременно с СУБД InterBase и MS Access. Использование модельно-ориентированного подхода позволяет существенно сократить сроки проектирования КИС по сравнению с длительностью эксплуатации системы.

### СПИСОК ЛИТЕРАТУРЫ

1. **Shakhgelyan, С.** Integration of university information resources into the unified information environment / С. Shakhgelyan, V. Kryukov // Proc. of the 10th Int. Conf. of European University Information Systems (ENUS 2004). Slovenia, 2004. P. 321–327.
2. **Брукс, Ф. П.** Как проектируются и создаются программные комплексы: мифический человеко-месяц. Очерки по системному программированию / Ф. П. Брукс. М. : Наука, 1979. 151 с.
3. **Рамбо, Дж.** UML 2.0. Объектно-ориентированное моделирование и разработка / Дж. Рамбо. СПб. : Питер, 2007. 544 с.
4. **Грибачев, К. Г.** Delphi и Model Driven Architecture. Разработка приложений баз данных / К. Г. Грибачев. СПб. : Питер, 2004. 348 с.
5. **Мамчур, М. Н.** Применение технологии MDA к разработке АИС «Учебный план» / М. Н. Мамчур, Л. Р. Рустамханова // Матер. Всерос. науч.-практ. конф. Нефтекамск, 2005. С. 139–144.

### ОБ АВТОРАХ



**Арьков Валентин Юльевич**, проф. каф. АСУ. Дипл. инж. по пром. электронике (УАИ, 1986). Д-р. техн. наук по сист. анализу и упр-ю (УГАТУ, 2001). Иссл. в обл. идентификации и моделирования САУ ГТД.



**Мамчур Марина Николаевна**, асс. каф. информатики. Дипл. инж. по автоматиз. техн. процессов и производств (УГАТУ, 1997).