

баз данных, которые обслуживают автоматизированную систему управления университетом; файловые серверы, которые обеспечивают коллективный доступ к драйверам, обновлениям антивирусных баз, программного обеспечения, служат своеобразным буфером для временного хранения информации или обмена ею; сервер-шлюз, который обеспечивает подключение к глобальной сети передачи данных Internet; веб-сервер, который обеспечивает функционирование системы веб-сайтов университета; почтовый сервер, обеспечивающий функционирование электронной почты университета.

После анализа существующих методов решения подобных задач, был предложен свой подход, предусматривающий возможность последующей интеграции локальной компьютерной сети учебного Машиностроительного корпуса в единую корпоративную сеть СумГУ. При этом учитывались следующие особенности: специфика организации корпоративной сети СумГУ (информационные носители, коммутирующее оборудование), географическое расположение учебных корпусов университета, количество и размещение компьютерных рабочих мест в рассматриваемом учебном корпусе. В дальнейшем было разработано черновое техническое задание, проработаны основные технические характеристики технического проекта. В результате был составлен технический проект локальной компьютерной сети учебного Машиностроительного корпуса Сумского государственного университета.

Разработанный проект удовлетворяет требованиям к единой информационной системе Сумского государственного университета. А также, соответствует стандартам IEEE 802.3, EIA/TIA 569 регламентирующим требования к системам на витой паре и на оптико-волоконном кабеле.

ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ ПРИЛОЖЕНИЙ БАЗ ДАННЫХ НА ОСНОВЕ КОНЦЕПЦИИ MDA

Овдиенко Я.А.

Данная работа относится к области автоматизации разработки информационных систем.

В настоящее время задачи автоматизации решаются в рамках концепции CASE (Computer Aided Software Engineering).

Ниже приведены средства, автоматизирующие ту или иную совокупность процессов жизненного цикла ПО:

1. Построение и анализ моделей предметной области (Upper CASE); Design/IDEF (Meta Software);
2. Анализ и проектирование (Middle CASE); Vantage Team Builder (Cayenne), Designer/2000;
3. Проектирование баз данных; Erwin (Logic Works), S-Designor (SDP), DataBase Designer;
4. Разработка приложений (4GL); Uniface (Compuware), JAM (JYAAC), PowerBuilder (Sybase);
5. Анализ программных кодов и схем баз данных; Rational Rose (Rational Software), Object Team;
6. Планирование и управление проектом; SE Companion, Microsoft Project;
7. Конфигурационное управление; PVCS (Intersolv);
8. Тестирование; Quality Works (Segue Software);
9. Документирование; SoDA (Rational Software).

Как видим все выше перечисленные средства на основании концепции CASE не поддерживают полный ЖЦ ПО.

Цель данной работы выбрать средство для проектирования и реализации приложений баз данных, удовлетворяющее критериям:

- **репозиторий**, являющийся основой CASE-средства (хранение версий проекта и его отдельных компонентов, синхронизация поступления информации от различных разработчиков при групповой разработке, контроль метаданных на полноту и непротиворечивость);

- **графические средства анализа и проектирования** (создание и редактирование иерархически связанных диаграмм, образующих модели ИС);
- средства **разработки приложений** (включая языки 4GL и генераторы кодов);
- средства **конфигурационного управления**;
- средства **документирования**;
- средства **тестирования**;
- средства **управления проектом**;
- средства **реинжиниринга**.

Выбирая модель решения задачи, выбор пал на концепцию MDA (Model Driven Architecture, дословно переводится как “архитектура, управляемая моделью”), которая заключается в следующем:

- ✓ **самостоятельный** и обязательный этап разработки логики функционирования приложения (**бизнес-логики**). Разработка приложения должна **начинаться** с этапа **создания модели приложения**, которая определяет состав, структуру и поведение будущего программного продукта. Такая модель создается не на языке программирования, а посредством языка унифицированного моделирования (Unified Modelling Language, UML) и является **платформенно-независимой** (Platform Independent Model, **PIM**), то есть при ее создании разработчик полностью абстрагируется от особенностей конкретных программных и аппаратных средств реализации приложения;
- ✓ создаются одна или несколько **платформенно-зависимых моделей PSM** (Platform Specific Model), которые являются своеобразными адаптерами, обеспечивающими интеграцию PIM с одной или несколькими технологиями разработки программных продуктов. Т.е. выбираем одну или несколько платформ разработки и создаем соответствующий набор PSM-моделей;

- ✓ на основании PIM и PSM, на этапе **генерации кода** получаем приложение или базу данных, функционирующее на этих платформах. Генерация кода и баз данных осуществляется автоматически, посредством специальных инструментальных программных средств.

Как можно увидеть из выше сказанного, концепция MDA является интегрированным CASE-средством, которое поддерживает **полный ЖЦ ПО**.

Для проектирования была выбрана реализация MDA фирмой Bold.

Bold for Delphi, интегрируясь в Borland Delphi, предоставляет разработчику полный набор визуальных и невизуальных компонентов, достаточный для реализации объектного пространства (Object Space) приложения. Разработчик получает возможность работать не на уровне кода и таблиц БД, а на уровне объектов внутри этого объектного пространства.

Трехуровневая схема создания приложения:

- ✓ **бизнес-уровень** (на нем ведется разработка, графический интерфейс пользователя);
- ✓ **объектно-реляционное отображение и взаимодействие** (автоматическая, прозрачная для разработчика трансляция OCL в операторы SQL, выполнение операций с таблицами БД);
- ✓ **работает не только на этапе** разработки приложения, но и на этапе его **исполнения** (именно это качество позволяет называть Bold инструментом реализации MDA).

Итак, из этого следует, что Bold for Delphi является тем средством, который полностью поддерживает концепцию MDA, а значит и поддерживает полный ЖЦ ПО, что необходимо в автоматизации разработки информационных систем. Выводы:

1. CASE-средство предназначено для реализации только этапов проектирования и моделирования; Концепция MDA охватывает

- весь ЖЦ ПО (сохраняет модель приложения в исполняемом файле, на этапе выполнения приложения использует эту модель для управления бизнес-уровнем, для контроля целостности объектного пространства, для управления взаимодействием бизнес-уровня с уровнем данных и графическим интерфейсом).
2. Концепцию MDA реализует средство разработки ПО ИС Bold for Delphi.
 3. Использование Bold for Delphi является целесообразным и оптимальным вариантом для автоматизации разработки приложений и информационных систем.

РАЗРАБОТКА ОБЪЕКТНО-РЕЛЯЦИОННОГО ПРЕОБРАЗОВАТЕЛЯ ДЛЯ ЯЗЫКА ПРОГРАММИРОВАНИЯ PERL

Зубань С.А.

Одной из главных задач, стоящих перед разработчиком сложных интернет-систем является проектирование и работа с базой данных. В настоящее время наиболее распространёнными СУБД являются *реляционные базы данных*. Отличительной особенностью реляционной модели является минимальная избыточность при представлении данных (при правильном проектировании базы данных). Но реляционная модель имеет и ряд отрицательных характеристик. Одной из них является абсолютно разная структура представления данных в такой базе в сравнении с объектами объектно-ориентированного языка программирования, что делает задачу работы с базой данных довольно сложной.

Для решения этой проблемы существуют *объектно-реляционные преобразователи*, которые позволяют максимально абстрагироваться от того факта, что данные хранятся в реляционной базе данных и оперировать уже исключительно объектами объектно-ориентированного языка программирования.

Представляет интерес использование ОРП в многопользовательских интернет-приложениях, например, в системах дистанционного