

На правах рукописи

Вершинин Максим Михайлович

Разработка графического подхода к проектированию корпоративных приложений на основе технологий Java 2, Enterprise Edition

Специальность 05.13.11 –
Математическое и программное обеспечение
вычислительных машин, комплексов и компьютерных сетей

Автореферат

диссертации на соискание ученой степени кандидата технических наук

Санкт – Петербург 2006

Работа выполнена в Государственном образовательном учреждении высшего профессионального образования «Санкт-Петербургский государственный политехнический университет».

Научный руководитель – доктор технических наук, профессор
Черноруцкий Игорь Георгиевич

Официальные оппоненты – доктор технических наук, профессор
Гаврилова Татьяна Альбертовна
– кандидат технических наук, доцент
Афанасьев Александр Сергеевич

Ведущая организация – Федеральное государственное унитарное
предприятие научно-производственное
объединение “Импульс”

Защита состоится « 15 июня 2006г. » в 16 часов на заседании диссертационного совета Д 212.229.18 в ГОУ ВПО «Санкт-Петербургский государственный политехнический университет» по адресу: 195251, Санкт-Петербург, Политехническая ул., д.29, 9 уч. корп., ауд. 325.

С диссертацией можно ознакомиться в Фундаментальной библиотеке ГОУ ВПО «Санкт-Петербургский государственный политехнический университет».

Автореферат разослан « ___ » мая 2006 г.

Ученый секретарь
диссертационного совета

Шашихин В.Н.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность

Объектом исследования диссертационной работы является процесс проектирования так называемых корпоративных приложений, то есть программных продуктов, так или иначе связанных с функционированием предприятия и нацеленных на повышение его эффективности. Проектирование программных продуктов — сложный итеративный процесс, успех которого зависит от грамотной организации самого процесса, выбора архитектуры и технологии разработки, а также от качества проектной документации.

Java 2, Enterprise Edition (J2EE) объединяет современные технологии, предназначенные для разработки распределенных программных систем уровня предприятия. Однако преимущества компонентных технологий, которые должны были бы сделать процесс проектирования простым, четким и технологичным, на практике нивелируются сложностями, связанными с трудозатратами на их изучение и многочисленными проблемами, возникающими при их реальном использовании. Руководители проектов неизбежно сталкиваются с нехваткой специалистов достаточно высокой квалификации и необходимостью проектирования в условиях дефицита времени. Задачи облегчения использования технологий J2EE и автоматизации проектирования корпоративных приложений становятся особенно актуальными в связи с все возрастающим спросом на разработку программных систем уровня предприятия.

В связи с этим крайне важно создание инструментальных средств, которые бы позволили автоматизировать сам процесс создания приложений уровня предприятия и снизили бы требования к квалификации участников проекта. Платформа J2EE начала приобретать популярность в 1999 - 2000 годах, и на тот момент не было ни одного продукта, облегчающего жизнь разработчика J2EE. Тогда и было принято решение создать среду разработки, позволяющую быстро и удобно моделировать корпоративные приложения и на выходе иметь синхронизированный исходный код и дескриптор поставки.

Под моделью будем понимать упрощенное представление реальной системы. Модель должна полностью и исчерпывающе описывать предполагаемую функциональность на данном уровне абстракции, однако без излишней детализации. Общеизвестно, что наиболее удобным и легким для восприятия средством представления любой системы является ее графическое отображение (визуальное моделирование). Стремление к унификации графических элементов, используемых для проектирования, привело к тому, что международной рабочей группой по развитию стандартов объектного программирования (Object Management Group, OMG) в 1997 г. был принят стандарт унифицированного языка

моделирования (Unified Modeling Language, UML). Этот язык является основой автоматизированного проектирования и создания программ (Computer-Aided Software Engineering, CASE). Язык UML непрерывно развивается, и в настоящее время действующим является стандарт UML 2.0. Создание среды разработки, позволяющей представлять модель проектируемого корпоративного приложения в графическом виде, следовало вести в рамках общепринятого стандарта UML.

Из публикаций, посвященных общим вопросам объектно-ориентированного моделирования, проектированию с использованием образцов и описанию технологий J2EE, следует особо отметить работы Г.Буча, Д.Рамбо, И.Якобсона, П.Коада, Э.Йордана, М.Фаулера, К.Скотта, П.Перроуна, В.Кришны, И.Грэхэма, Э.Гаммы и Э.Романа. В них излагаются базовые понятия, архитектура J2EE и детали технологий, но проблемы практического освоения этих технологий, их реального применения и потенциальные возможности автоматизации проектных процедур рассматриваются недостаточно.

На сегодняшний день существует не так много продуктов, поддерживающих разработку J2EE-приложений, из которых только один (WSAD) поддерживает визуальное моделирование с помощью UML. Связано это в первую очередь с тем, что сама платформа J2EE, как и другие компонентные технологии, появилась сравнительно недавно. Кроме того, для создания соответствующих спецификаций требовалось исследовать и обобщить свойства наиболее распространенных промышленных серверов приложений и предложить унифицированный подход к созданию среды разработки. Существующие программные средства, как правило, страдают односторонностью и не в полном объеме удовлетворяют требованиям пользователей. Как видно из Таблицы 1, ни один из существующих продуктов не поддерживает моделирование корпоративных приложений, поэтому на предприятиях используется несколько продуктов: одни - для описания системы, другие - для ее разработки. Это усложняет процесс проектирования и требует дополнительных затрат для синхронизации при малейших изменениях в системе.

Таблица 1. Сравнение инструментальных средств разработки

Показатель	Idea	JBuilder	WSAD
Интегрированная среда разработки (Integrated Development Environment, IDE)	+	+	+
J2EE-разработка	+	+	+
Размещение на сервере	+	+	+
Визуальное моделирование	-	-	+
J2EE-моделирование	-	-	-
Возможность расширения	+	-	-

В настоящей работе предлагаются теоретическая база и методика визуального проектирования корпоративных приложений. Результатом применения предлагаемого подхода является исчерпывающий набор спецификаций, на основе которых создана инструментальная программная система автоматизированной разработки корпоративных приложений. Кроме того, предложен универсальный абстрактный интерфейс и абстрактная модель на его основе, позволяющая описывать практически любые реальные модели данных.

Актуальность проблемы автоматизации проектирования корпоративных приложений подтверждается тем, что ее решение позволит облегчить и формализовать процесс моделирования, а также ускорить разработку приложений и их размещение на серверах. Процесс проектирования при этом становится более качественным и технологичным, что существенно сокращает общие затраты на проектирование и кодирование программных продуктов рассматриваемого класса.

Цель и задачи работы

Целью диссертации является разработка методики проектирования корпоративных приложений с использованием их графического представления, создание новых типов визуальных элементов, адекватно отображающих свойства программных систем данного класса, разработка абстрактного подхода к проектированию с использованием компонентных технологий и, в частности, платформы J2EE. Предлагаемые методики позволят повысить производительность труда разработчиков и эффективность проектирования в целом.

Для достижения этих целей в работе ставились и решались следующие задачи:

разработка универсального набора абстрактных интерфейсов и реализация на его базе абстрактной модели с возможностью ее интеграции в другие программные продукты; создание новых визуальных элементов для представления приложений J2EE и их отдельных компонентов (в полном соответствии со стандартом UML) с целью автоматизации процесса проектирования;

разработка методики автоматической генерации исходного кода и дескрипторов поставки на основе созданных моделей, их синхронизация с моделью;

разработка методики проектирования сложных распределенных приложений, использующей графический подход и позволяющей в рамках одного проекта создавать программную систему и размещать ее подсистемы на различных серверах;

создание визуальных шаблонов, реализующих J2EE-образцы, и разработка методики их использования в процессе проектирования корпоративных приложений;

автоматизация процесса размещения приложений на различных серверах с учетом их специфики;

практическая реализация полученных теоретических результатов, а именно, создание инструментальных программных средств, ориентированных на программистов средней квалификации и позволяющих автоматизировать моделирование, кодирование и размещение на серверах сложных программных систем.

Методы исследования

В диссертации используется теоретический аппарат, основанный на концепции объектно-ориентированных моделей, объектно-ориентированного проектирования и дизайна, а также теория графов для описания моделей. Основными критериями являлись технологичность и простота разработки (usability), возможность повторного использования (reusability). Применялись архитектура, управляемая моделью (Model Driven Architecture, MDA), технологии платформы J2EE, стандарты Unified Modeling Language (UML), идеология построения CASE-средств, а также Constructive Cost Model II (COCOMO II) для оценки эффективности процесса разработки программного продукта.

Достоверность полученных результатов подтверждается следующим образом:

1. В процессе создания абстрактной модели, отвечающей требованиям универсальности и расширяемости, и абстрактного API, а также конкретной модели и конкретного API для платформы J2EE последовательно применялись принципы MDA.
2. Разработка новых визуальных элементов производилась в строгом соответствии с требованиями общепринятого стандарта UML, зарегистрированного группой OMG.
3. Осуществлялась постоянная проверка полноты разработанных визуальных элементов, их адекватности набору компонентов J2EE и соответствия спецификациям фирмы-разработчика платформы J2EE “Sun Microsystems”.
4. Образцы, реализованные автором, разрабатывались в контакте с ведущим специалистом Джоном Круппи, под руководством которого разрабатывались J2EE-шаблоны в “Sun Microsystems”. Производились экспертные оценки визуальных шаблонов.
5. Корректность исходного кода и дескрипторов поставки, генерируемых на основе визуальной модели, а также их взаимная синхронизация проверялись полномасштабным тестированием.
6. Достоверность технических решений подтверждается
 - a. сравнением с другими продуктами и подходами,
 - b. анализом экспериментальных показателей результатов практического использования.

7. Для оценки достоверности полученных результатов привлекались независимые эксперты. Получены положительные отзывы крупных предприятий, таких, как SAP (“Systems, Applications, and Products in Data Processing”) и RBS (“Royal Bank of Scotland”).

Научная новизна работы состоит в следующем:

Предложен и теоретически обоснован новый графический подход к разработке сложных распределенных программных систем уровня предприятия.

Разработана новая абстрактная модель корпоративного приложения и соответствующий абстрактный программный интерфейс (Application programming Interface, API).

Предложен метод построения моделей и программных интерфейсов, поддерживающих конкретные компонентные технологии на основе абстрактной модели и абстрактного программного интерфейса.

Предложена методика автоматизации основных проектных процедур: моделирования, кодирования и размещение приложений на сервере. В рамках UML разработаны спецификации новых J2EE-диаграмм, в том числе обобщенная диаграмма размещения.

Для поддержки типовых проектных решений разработаны новые визуальные элементы, реализующие J2EE-образцы.

Основные положения, выносимые на защиту:

1. Расширяемая абстрактная модель для представления иерархически-связанных элементов и абстрактный программный интерфейс (API) к ней, а также метод построения конкретных моделей и конкретных программных интерфейсов для проектирования корпоративных приложений, поддерживающих выбранную компонентную технологию (J2EE, .NET, CORBA и пр.), на основе абстрактной модели и абстрактного API.
2. Методика визуального моделирования корпоративных приложений с использованием технологий J2EE. На основе стандарта UML разработаны и специфицированы новые типы визуальных элементов для графического представления программной системы рассматриваемого класса, адекватно описывающие его свойства и служащие основой для последующей автоматической генерации кода и размещения приложения на сервере.
3. Методика проектирования распределенных программных систем с использованием обобщенной диаграммы размещения, позволяющая в рамках одного проекта реализовать автоматическое размещение подсистем на различных серверах.
4. Новые визуальные элементы, реализующие шаблоны J2EE. Использование этих элементов в процессе проектирования облегчает и ускоряет решение типовых задач, часто встречающихся при разработке корпоративных приложений.

Практическая значимость работы

1. Предложенный метод визуального проектирования корпоративных приложений на основе технологии J2EE доведен до практической реализации в реальных продуктах, которые уже длительное время используются разработчиками.
2. Предложенный программный интерфейс и реализованная на его основе абстрактная модель могут быть с успехом использованы для разработки других программных продуктов.
3. Использование данной методики позволяет сократить затраты на проектирование и существенно снизить их при внесении каких либо изменений в технические требования на разрабатываемую систему, а также на начальном этапе проектирования (по экспериментальным оценкам от 20% до 50%, в зависимости от объема и сложности проекта).

Апробация работы

Основные результаты диссертационной работы обсуждались в рамках научно-практических конференций СПбГПУ (2001г. - 2004г.), на 2-й международной научно-практической конференции «Информационные технологии в моделировании и управлении» (Санкт-Петербург, 2000г.), на конференции «Bea User's Conference» (Даллас, США, 2002г), на семинаре для отдела маркетинга фирмы «Togethersoft» (Рэйли, США, 2002г.), на семинаре для отдела поддержки фирмы «Togethersoft» (Рэйли, США, 2003г.), на конференции «Технологии Microsoft в теории и практике программирования» (Санкт-Петербург, 2004г.), на 5-й международной научно-технической конференции «Компьютерное моделирование 2004» (Санкт-Петербург, 2004г.), на семинаре «Royal Bank of Scotland» (Лондон, Великобритания, 2005г.) и на научно-техническом семинаре профессорско-преподавательского состава кафедры «Информационные и управляющие системы» СПбГПУ (Санкт-Петербург, 2005г.).

Автором проводились также теоретические и практические занятия для разработчиков интеграционной команды фирмы «Togethersoft» (Рэйли, США, 2002г.), для разработчиков фирмы «Vorland» (Прага, Чехия, 2004г.), для студентов СПбГПУ (Санкт-Петербург, 2004г.).

Внедрение

Результаты диссертационной работы использовались при разработке диаграмм и компонентов J2EE, вошедших в интегральную среду разработки линии Borland@R Together@R ControlCenter@R: Borland@R Together@R ControlCenter@R 6.x, Borland@R JBuilder, Borland@R JBuilder Eclipse Edition.

Методика автоматизированной разработки корпоративных приложений внедрена в ООО «Интеллект-ФТК» при разработке учебно-методического комплекса.

Практическое использование представляемых на защиту результатов подтверждается соответствующими актами о внедрении.

Публикации

По материалам диссертации опубликовано 6 работ.

Структура и объем работы

Диссертация содержит 178 страниц основного текста, 71 рисунок, 12 таблиц и состоит из введения, 5 глав, заключения, списка сокращений и списка использованных источников.

СОДЕРЖАНИЕ РАБОТЫ

Во введении обосновывается актуальность работы, формулируются цель, задачи, объект и предмет диссертационного исследования, показана научная новизна и практическая ценность полученных результатов. Приводятся краткие сведения о структуре диссертации.

Первая глава посвящена изучению процесса проектирования корпоративных приложений, исследованию архитектуры, управляемой моделью (MDA), унифицированного языка моделирования (UML) и технологий J2EE с точки зрения их применимости для автоматизации разработки сложных программных систем.

В настоящее время наиболее перспективным направлением проектирования новых программных продуктов является применение MDA. При этом подходе первоначально строится абстрактная модель, не зависящая от платформы. Затем абстрактная модель может быть преобразована в модель для любой конкретной платформы. Архитектура проекта определяется моделью, что делает процесс проектирования технологичным и обоснованным.

Компонентные технологии J2EE, представленные в спецификациях «Sun Microsystems», полностью поддерживают принципы MDA и предоставляют широкие возможности для автоматизации. Однако реальное применение архитектуры J2EE вызывает трудности, связанные с непомерно большим объемом информации, знание которой необходимо для применения данных технологий, а также чрезмерно высокими требованиями к квалификации специалистов, участвующих в процессе разработки.

Логичным решением данной проблемы является использование графических образов в процессе проектирования, базирующееся на нотации UML. Вопросы создания автоматизированных систем разработки программных систем на основе MDA,

использующих одновременно преимущества технологий J2EE и визуальные методы моделирования UML, ранее не рассматривались. Настоящая работа восполняет этот пробел.

Во второй главе содержится описание теоретических основ методики визуального проектирования корпоративных приложений, разработанных автором на основе анализа концепций MDA, стандартов UML и технологий J2EE. Рис.1 иллюстрирует процесс разработки корпоративного приложения с использованием предлагаемой методики.



Рисунок 1. Процесс разработки корпоративного приложения на основе MDA

Отметим следующие основные моменты:

1. Спецификации UML предлагают набор диаграмм для визуального отображения проектируемой системы. Диаграммы создаются с помощью стандартизованного набора визуальных элементов (узлов, соединенных связями).

2. Специализированные диаграммы J2EE являются расширением UML, наследуя от метамодели UML и приобретая новые свойства, связанные со спецификой приложений J2EE. При этом эффективно использование разработанных автором образцов.
3. Абстрактная модель системы справедлива для любых компонентных технологий.
4. На основе абстрактной модели строится модель J2EE.
5. Далее абстрактная платформенно-независимая модель трансформируется в модель выбранной платформы (платформенно-зависимая модель).
6. Заключительное преобразование модели осуществляется с учетом спецификаций конкретного сервера приложений J2EE.

На базовой модели (Рис. 2) можно представить бизнес-объекты, которые каким-то образом связаны между собой. Эта модель изображает архитектуру приложения без привязки к какой-либо платформе: .NET, CORBA, J2EE и т.д. Далее эта модель может быть трансформирована в специфичную для конкретной платформы модель, в нашем случае - в J2EE-модель, отображающей объекты EJB, сервлеты, страницы JSP и т.п.

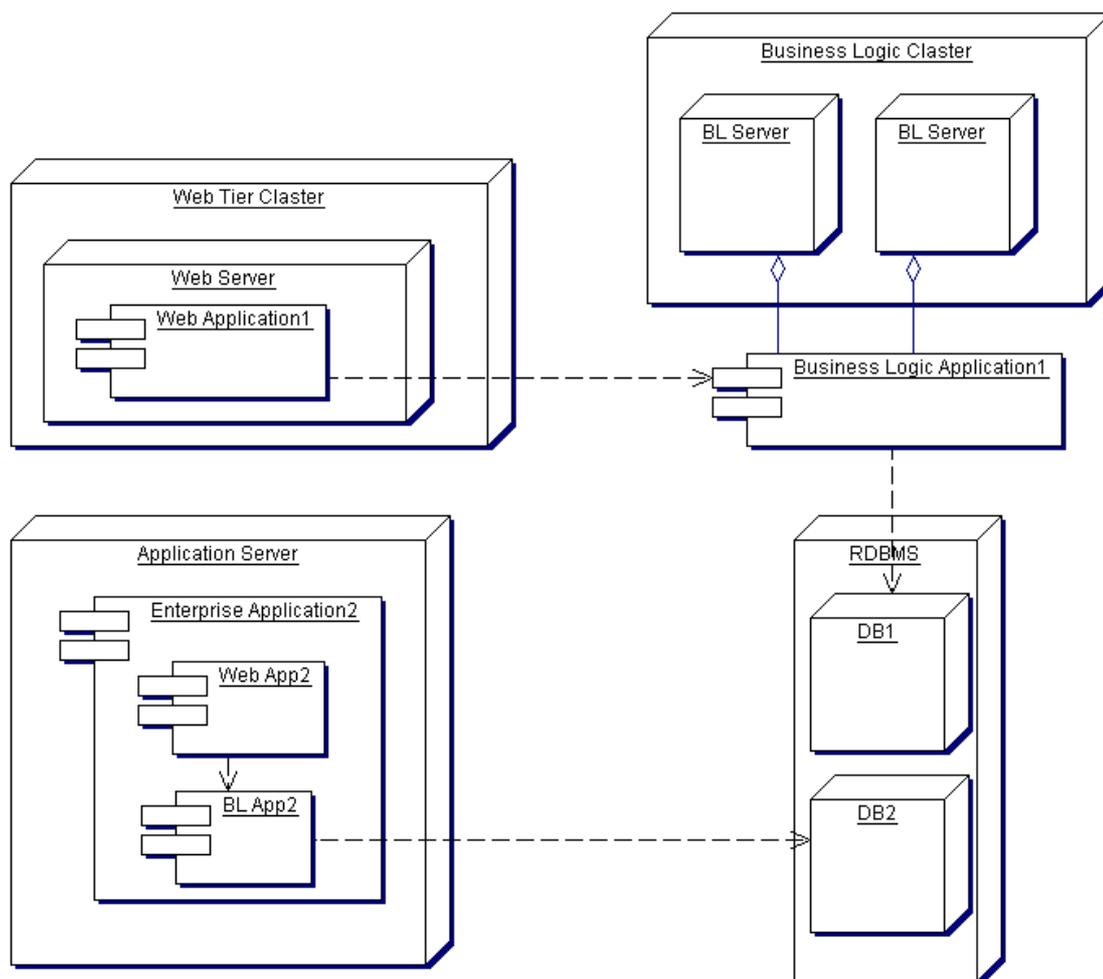


Рисунок 2. Абстрактная модель системы

Визуальный элемент Web Server представляет сервер, на котором размещены Web-приложения. Web Server может быть самостоятельной единицей (например, элементы RDBMS и Application Server непосредственно на диаграмме) или располагаться в кластере для распределения нагрузки и обеспечения устойчивости системы.

На Рис.2 для примера показано, что кластер Web Tier Cluster содержит один Web-узел. Это, вообще говоря, не имеет особого смысла, но вполне возможно в целях дальнейшего расширения. Кластер Business Logic Cluster содержит два узла, на которых размещено приложение с бизнес-логикой. При использовании технологий J2EE узел Business Logic Application представлен модулем EJB (EJB Module), а элементами типа BL Server являются серверы приложений, на которых оно размещено. RDBMS - сервер баз данных, который также может быть разбит на кластеры, DB – база данных.

Эта диаграмма может быть трансформирована в специфичную для платформы диаграмму, на которой кластеры и узлы будут представлять конкретные серверы с конкретными свойствами конфигурации, что позволяет автоматически размещать разрабатываемую систему на соответствующих серверах.

Узлы Web Application1 и Business Logic Application1 могут быть развернуты в соответствующие абстрактные диаграммы Web-приложений и приложений бизнес-логики, которые впоследствии могут быть трансформированы в конкретные приложения, в нашем случае, - приложения J2EE. Чтобы показать это все вместе, раскроем узел Enterprise Application (Рис. 3):

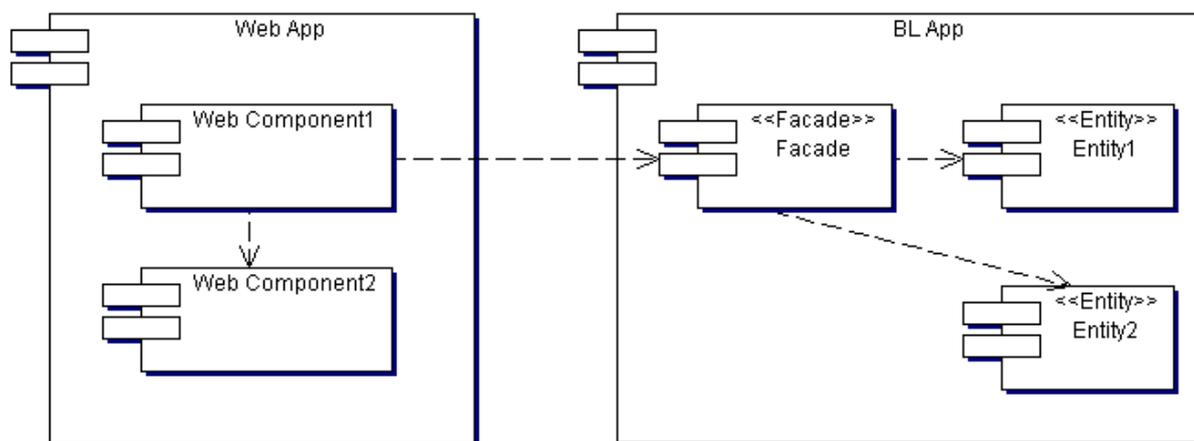


Рисунок 3. Развертывание узла Enterprise Application

Узел Enterprise Application трансформируется в соответствующее корпоративное приложение (J2EE Enterprise Application). Узел Web App трансформируется в Web-приложение (J2EE Web Application), где Web-компонентами являются сервлеты. Узел BL App трансформируется в модуль EJB, где компоненты будут преобразованы в соответствующие элементы Session и Entity EJB. Кроме того, специально обозначенные

компоненты могут быть преобразованы с использованием соответствующих образцов проектирования.

В третьей главе автор предлагает к рассмотрению спецификации визуальных элементов, предназначенных для моделирования и разработки приложений J2EE. Разработанные спецификации, реализующие теоретическую базу, представленную в главе 2, можно объединить в следующие группы:

- распознавание и выделение классов и интерфейсов, относящихся к J2EE;
- визуальные элементы для разработки Web-приложения;
- визуальные элементы для разработки модуля EJB;
- визуальные элементы для разработки корпоративного приложения.

Предлагаемые спецификации полностью поддерживают многоуровневую концепцию J2EE.

Каждый визуальный элемент J2EE, представляющий собой класс или интерфейс, имеет общие свойства (Properties), как и любой обычный класс или интерфейс. Установка общих свойств влияет на автоматическое формирование соответствующего программного кода. Кроме общих параметров, элементы J2EE обладают свойствами размещения, задавая которые можно фактически определить необходимые тэги при генерации дескрипторов поставки. Подробное описание параметров приводится в спецификациях визуальных элементов J2EE, представленных в тексте диссертации.

Web-приложения реализуют уровень представления приложения J2EE и обеспечивают взаимодействие между клиентами и уровнем бизнес-логики. Для визуального моделирования и размещения Web-приложения разработана спецификация диаграммы Web-приложения (Web Module diagram), а также спецификация, расширяющая диаграмму классов и позволяющая работать на ней со специализированными Web-элементами. Первоначально на диаграмме классов специальным образом выделялись классы, относящиеся к Web-приложению (такие, как сервлеты, страницы JSP, фильтры, ссылки на компоненты EJB, роли секретности, Web-ресурсы и т.п.). Эти элементы имели дополнительные свойства, используемые при размещении приложения на сервере. В дальнейшем было принято решение максимально приблизить визуальную модель приложения к дескриптору поставки (с использованием стандартных UML-средств с небольшими дополнениями). А именно, за основу была взята стандартная компонентная диаграмма UML.

В процессе моделирования с помощью диаграммы Web Module устанавливаются свойства сервлетов, страниц JSP, прочих Web-компонентов и Web-приложения в целом, необходимые при размещении на сервере, создаются фильтры и слушатели событий, определяется режим секретности и т.д. Для создания или подключения библиотек тегов

используется дополнительная диаграмма библиотеки тэгов (Taglib diagram). Добавление библиотек тегов к Web-приложению осуществляется с помощью элемента Taglib.

Диаграмма Web Module с расположенными на ней элементами является визуальной моделью Web-приложения. С нее осуществляются редактирование исходного кода компонентов, поиск возможных ошибок и отладка программы, автоматическая генерация дескрипторов поставки и размещение приложения на выбранном сервере J2EE.

Уровень EJB представляет собой уровень бизнес-логики. На этот уровень обычно выносятся вычисления, а также вся логика по работе с базами данных.

Визуальное моделирование и последующее размещение модуля EJB на сервере осуществляются с помощью диаграммы EJB Module. По аналогии с диаграммой Web Module и для удобства пользователя были расширены возможности диаграммы классов, а именно: все классы и интерфейсы, относящиеся к EJB, распознаются и объединяются в соответствующие EJB-компоненты. Это особенно актуально, так как EJB-компоненты, в отличие от Web-компонентов, состоят из 1-6 элементов. Была также добавлена возможность за один щелчок мыши создавать полноценный компонент EJB. Реализована техника автоматического создания простого клиента EJB на основе разработанных автором образцов.

Все EJB-элементы выделялись на диаграмме классов и имели дополнительные свойства для размещения на сервере. Разработаны визуальные элементы для всех типов компонентов EJB: session, entity и message-driven bean, а также способ визуального отображения отношений компонентов entity bean (EJB relationships). Как и в случае диаграммы Web Module, для диаграммы EJB Module с использованием стандартных средств UML была разработана визуальная модель приложения, максимально приближенная к дескриптору поставки. Диаграмма EJB Module также базируется на стандартной компонентной диаграмме UML, расширяя ее возможности.

Диаграмма EJB Module с расположенными на ней элементами является визуальной моделью модуля EJB. С нее осуществляются редактирование исходного кода классов реализации и интерфейсов компонентов EJB, автоматическая генерация дескрипторов поставки и размещение модуля EJB на сервере. При разработке диаграммы EJB Module широко используется техника создания компонентов на основе образцов. Подробная спецификация диаграммы EJB Module и ее элементов представлена в тексте диссертации.

Диаграмма корпоративных приложений (Enterprise Module diagram) является основной диаграммой для формирования и размещения на сервере приложения J2EE, объединяющего Web-приложения, модули EJB, коннекторы и приложения-клиенты. Разработанная автором методика внедрена в инструментальные средства линии Borland@R Together@R ControlCenter@R, предназначенные для разработки программного обеспечения.

В процессе работы над диссертацией была также создана обобщенная диаграмма размещения, позволяющая моделировать корпоративное приложение, распределенное по разным серверам. На сегодняшний день ни в одном из существующих продуктов этого нет. Опрос потенциальных клиентов, менеджеров различных фирм и экспертов-аналитиков показал их заинтересованность в подобной диаграмме.

Четвертая глава содержит описание методики автоматизированного проектирования корпоративного приложения с использованием визуализации. Рассматриваются абстрактный и конкретный программные интерфейсы (API), классы и образцы, разработанные автором. Описывается последовательность действий, которые следует выполнить для моделирования, кодирования и размещения приложения на сервере. Приводятся сведения о графических инспекторах для установки свойств компонентов J2EE и эксперте размещения.

В пятой главе проводится анализ результатов внедрения разработанной методики и анализируется ее эффективность. Для оценки стоимости разработки с помощью предложенной автором методики применялся метод СОСОМО II Барри Боэма, опирающийся на знание специфики проектируемой системы и оценку команды разработчиков.

Номинальный объем трудозатрат рассчитывался по формуле (1):

$$PM = \prod_{i=1}^{17} (EM_i) \cdot A \cdot \left[\left(1 + \frac{BRAK}{100} \right) \cdot Size \right]^B + \left[\frac{ASLOC \cdot \left(\frac{AT}{100} \right)}{ATPROD} \right], \quad (1)$$

$$\text{где } B = 0.91 + 0.01 \cdot \sum_{i=1}^5 w_i; \quad (2)$$

PM – трудозатраты (в человеко-месяцах);

$A = 2,45$ – константа, значение которой рекомендовано в СОСОМО II;

$Size$ – размер проектируемой программной системы в в тысячах строк исходного кода;

w_i - масштабирующие коэффициенты, представленные в Таблице 2;

EM_i – дополнительные коэффициенты, учитывающие масштабирующие факторы, которые характеризуют проект в целом (представлены в тексте диссертации);

$BRAK$ – процент кода, выбрасываемого из программной системы из-за изменений требований в процессе проектирования;

$ASLOK$ – размер адаптируемых компонентов (количество строк);

$ATPROD$ – производительность автоматизированной трансляции;

AT – процент автоматически транслируемых компонентов;

Таблица 2. Масштабирующие коэффициенты метода СОСОМО

Коэффициент w_i	Назначение коэффициента	Значение (в СОСОМО)	
		Проектирование без визуализации	Визуальное проектирование
PREC	Наличие предшествующего опыта проектирования подобных систем	Нормальный 2.43	Высокий 1.62
FLEX	Гибкость разработки	Очень низкая 6.07	Очень высокая 1.21
RESL	Устойчивость к рискам при выбранной архитектуре	Низкая 3.38	Очень высокая 0.84
TEAM	Сработанность команды	Нормальная 2.97	Нормальная 2.97
PMAT	Коэффициент учета измеримости процессов	4	1.5
$\sum_{i=1}^5 w_i$		18.85	8.14
Результат расчета константы B		1.0985	0.9914

Анализировались три реальных проекта, для которых были получены оценки номинальных затрат. По полученным оценкам сокращение сроков разработки при использовании визуализации составляло от 25% до 50%, в зависимости от сложности проекта. Анализ результатов экспериментального параллельного проектирования одного и того же корпоративного приложения двумя равноценными группами разработчиков подтвердил правильность полученных теоретических оценок.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ РАБОТЫ

1. Предложен и теоретически обоснован новый подход к автоматизации проектирования корпоративных приложений, а именно, проектирование с помощью визуальной модели, построенной на основе архитектуры MDA и в соответствии со стандартами UML. Предлагаемая методика позволяет сделать процесс проектирования более технологичным и эффективным, а также снижает требования к квалификации разработчиков.
2. Разработаны и специфицированы новые типы диаграмм, предназначенных для моделирования приложений J2EE, а также визуальные элементы, представляющие компоненты корпоративных приложений. J2EE-диаграммы позволяют создавать модель корпоративного приложения, на базе которой осуществляется генерация скелета программного кода и дескрипторов поставки, а также автоматическое размещение приложения на сервере.

3. Предложены абстрактная модель для представления иерархически-связанных элементов и соответствующий абстрактный программный интерфейс, легко расширяемые и переносимые на различные компонентные технологии, такие, например, как .NET.
4. На основе абстрактного программного интерфейса разработан конкретный программный интерфейс, поддерживающий технологии J2EE.
5. Созданы и реализованы визуальные образцы, предназначенные для решения типовых задач проектирования корпоративных компонентов. Образцы, реализованные автором, разрабатывались в контакте с Джоном Круппи, под руководством которого разрабатывались J2EE-шаблоны в фирме “Sun Microsystems”.
6. Предложена новая обобщенная диаграмма размещения, автоматизирующая проектные процедуры для задач проектирования сложных распределенных программных систем.
7. Разработан метод автоматической генерации исходного кода и дескрипторов поставки на основе визуальной модели (J2EE-диаграмм), а также инструментальные средства (инспекторы и эксперт размещения), облегчающие работу пользователя.
8. Предложенный метод визуального проектирования корпоративных приложений на основе технологии J2EE реализован в конкретных программных продуктах (Borland@R Together@R ControlCenter@R 6.x; Borland@R JBuilder; Borland@R Together@R ControlCenter@R, Eclipse Edition).
9. Произведены экспериментальные и теоретические оценки эффективности использования графического подхода для моделирования, разработки и размещения корпоративных приложений. Сокращение трудозатрат по сравнению с традиционными методами проектирования составляло не менее 25%. Наиболее эффективно использование предлагаемой методики на начальных этапах разработки, при возможных изменениях требований в процессе проектирования, а также для автоматизации размещения приложения на сервере.

Содержание диссертации отражено в следующих публикациях:

1. Вершинин М. М. Иванова Е. Б. Разработка компонента ACTIVEХ для визуализации биржевой информации // Информационные технологии в моделировании и управлении: Труды II международной научно-практической конференции. – Санкт-Петербург: Издательство СПбГТУ, 2000. С. 84 – 86.
2. Вершинин М. М. Иванова Е. Б. Java 2, Enterprise Edition. Технологии проектирования и разработки. – Санкт-Петербург: БХВ-Петербург, 2003. – 1068 с. – 3000 экз.

3. Вершинин М.М. Поддержка Web сервисов в Borland@R Together@R ControlCenter@R с целью облегчения их разработки и использования // Технологии Microsoft в теории и практике программирования: материалы межвузовского конкурса-конференции студентов и молодых ученых Северо-Запада. – Санкт-Петербург, Издательство СПбГПУ, 2004. С. 15 – 16.
4. Вершинин М.М. Иванова Е.Б. Применение визуального моделирования для разработки приложений J2EE в среде Borland@R Together@R ControlCenter@R // Компьютерное моделирование 2004: труды 5-й международной научно-технической конференции. – Санкт-Петербург, ROSTE, 2004. С. 16 – 24.
5. Вершинин М. М. Иванова Е. Б. Моделирование корпоративных приложений // Научно-Технические Ведомости, № 1(35)/2004. – Санкт-Петербург: Издательство СПбГПУ, 2004. С. 182 – 187.
6. Вершинин М.М. Разработка образцов для автоматической генерации простых клиентов для компонентов EJB и WEB сервисов // XXXII Неделя науки СПбГПУ: материалы межвузовской научно-технической конференции 24-29 ноября 2003 года, часть V. – Санкт-Петербург, Издательство СПбГПУ, 2004, С.3 – 4.