

УДК 658.012.011.56

СРЕДСТВО БЫСТРОЙ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ WEB-ОРИЕНТИРОВАННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ (WADK).

В.М. Левыкин, Е.А. Моспан.

Харьковский национальный университет радиоэлектроники.

Статья посвящена проблемам, возникающим при разработке ПО WEB-ориентированных ИС, и рассматривает применение средств быстрой разработки ПО (framework) как возможный путь их решения. Указаны основные требования, которыми должен обладать framework для его эффективного применения. В связи с тем, что существующие framework не удовлетворяют указанным требованиям, было создано средство быстрой разработки ПО WEB-ориентированных ИС WADK (Web Application Development Kit), которое удовлетворяет большинству поставленных требований.

Ключевые слова: WEB-ориентированная ИС, WADK, MVC, framework, средство быстрой разработки ПО, инструментарий.

Введение

На сегодняшний день при разработке ПО информационных систем часто применяют готовые инструментальные средства, которые позволяют снизить трудовые и производственные затраты на разработку. Применение подобных инструментариев особенно актуально, когда система проектируется в несколько стадий и разработчик не знает о тех изменениях, которые должны будут произойти в системе на следующем этапе. В этом случае ПО ИС должно быть достаточно гибким и расширяемым.

С развитием глобальной сети Internet, а также ростом количества и размеров корпоративных и локальных сетей WEB-ориентированные ИС становятся все более популярными. Данные системы являются клиент-серверными и в качестве входного интерфейса используют любой WEB-браузер, не требующий значительных системных ресурсов. Однако использование WEB-браузера создает также и массу дополнительных проблем для разработчика, поскольку такой интерфейс не дает гарантии достоверности присылаемых запросов в систему. Из основных проблем, с которыми сталкивается разработчик WEB-ориентированной ИС, можно выделить следующие:

- организация отправления запросов пользователя на сервер и обработка ответа;
- разработка динамических пользовательских интерфейсов;
- управление переходами пользователя по HTML-страницам с учетом политики безопасности ИС;
- организация безопасности передачи данных;
- контроль целостности передаваемых данных;
- обработка ошибок и исключительных ситуаций, которые возникают в процессе функционирования системы.

Существующие технологии и методы

Для решения указанных выше проблем применяют Java-базированные технологии, такие как JavaServer Pages, Servlets, JavaBeans и другие [1]. Использование данных технологий дает возможность разработчику ускорить процесс разработки WEB-ориентированных ИС. Однако проблема заключается в том, что эти технологии не содержат в себе базового инструментария для разработчика, который вынужден начинать каждую разработку с нуля. Решение заключается в использовании средств быстрой разработки ПО – framework. Наиболее популярными framework являются свободно распространяемые, Java-базированные продукты компании Jakarta, такие как Struts, Velocity и другие, основанные на модели MVC (Model-View-Controller) [2]. Модель MVC делит программу на три части. В её состав входит три компонента Model, View, Controller. В обязанности компонента Model входит хранение данных и обеспечение интерфейса к ним. View ответственен за представление этих данных пользователю. Controller управляет компонентами, получая сигналы в виде реакции на действия пользователя и уведомляя об изменениях модель компонента.

Постановка задачи

В ходе анализа основных проблем, возникающих при разработке WEB-ориентированных ИС, был определен тот инструментарий, которым должен обладать framework, используемый для проектирования их ПО:

- средство должно быть основано на модели MVC;

- средство должно обладать, настраиваемым компонентом Controller;
- наличие инструментария по работе с БД;
- встроенная политика безопасности, которая предоставляет средства для настройки её под конкретную систему;

- контроль перехода пользователя по WEB-страницам системы;

- организация контроля целостности входной информации и предоставление средств для кэширования результатов выполнения расчетов, требующих значительных вычислительных затрат.

Анализа существующих framework компании Jakarta Struts [3] и Velocity [4], которые являются наиболее качественными разработками, позволил установить, что они не обладают необходимым инструментарием для удовлетворения всех выше поставленных требований. Поэтому было принято решение о создании такого framework, который учитывал бы эти требования и облегчал решение основных проблем, возникающих при разработке ПО. В связи с этим возникла проблема разработки WADK (Web Application Development Kit). Предлагаемое средство базируется на модели MVC с использованием кроссплатформенного языка программирования Java, что делает его малозависимым от системного ПО.

Изложение основного материала

Компонент Model WADK представляет собой базовый инструментарий, на основании которого возможно интерпретировать поведение того или иного бизнес-процесса, автоматизируемого в рамках конкретной разработки ИС. В задачи компонента Model WADK входит интерпретация атрибутивных сущностей реального мира, принимающих непосредственное участие в бизнес-процессе, организация связей между ними, а также работа с БД системы. Для компонента Model разработана объектная модель, благодаря которой возможно единое управление сущностями из компонента Controller, а также переносимость и расширяемость разработанных элементов.

WADK предполагает работу с тремя видами сущностей:

- *простые сущности* – предоставляют возможность разработчику обработки пользовательских данных и использование их в ходе выполнения того или иного бизнес-процесса. Простые сущности содержат в себе набор атрибутов, характеризующих их текущее состояние;

- *сущности БД* – являются частным случаем простых сущностей. Их основным отличием является то, что каждая такая сущность является отражением одной строки таблицы БД системы, а атрибуты сущности – столбцов этой таблицы;

- *сущности-списки*, предназначены для хранения краткой информации о наборе сущностей первых двух типов.

Объектная модель компонента Model WADK представлена на рис. 1. Она содержит в себе базовые классы для создания сущностей, описанных выше. Благодаря базовому классу AFormBean, наследниками которого они все являются, возможна организация управления сущностями в компоненте Controller. Сущности БД (класс ADBFormBean) содержат инструментарий для работы с БД системы и предоставляют разработчику единый подход для работы с ней. Сущности-списки (класс AListBean) предоставляют возможность осуществлять работу с результатами запросов к БД (сортировку, выделение блоков данных и т.д.) без дополнительного обращения к ней. В WADK предоставляется возможность вложения сущностей в более крупные, что позволяет проектировать сложные объекты реального мира, а также является достаточно удобным при работе с БД системы. Поведение вложенной сущности определяется сущностью-контейнером.

Контейнером для атрибутов сущностей выступает ассоциативный список, который заполняется на этапе инициализации сущности. Атрибуты сущностей WADK предоставляют разработчику возможность хранить информацию в строковом и объектном виде. Необходимость такого представления связана с тем, что все данные, поступающее вместе с входными запросами, передаются в строковом виде, что предполагает их дальнейшее преобразование. WADK позволяет определить правила для преобразования значений атрибутов, благодаря чему достигается контроль целостности передаваемых данных. Framework содержит в себе набор стандартных правил для преобразования форматов, состав которых может быть при необходимости расширен. Стандартизированная работа с атрибутами осуществляется благодаря тому, что все они являются наследниками одного абстрактного класса Field. В основную поставку WADK входят некоторые реализации атрибутов, которые представляют простые типы данных (целые и вещественные числа, строки и даты и т.д.), а также такие, которые могут быть использованы при построении пользовательских интерфейсов. При необходимости разработчик может создать произвольный атрибут сущности, разработав класс-наследник класса Field и определив его поведение. Атрибуты сущностей WADK содержат инстру-

ментарий для кэширования результатов преобразования форматов представления данных. Решение о необходимости выполнения преобразования принимает сам атрибут на основании анализа измене-

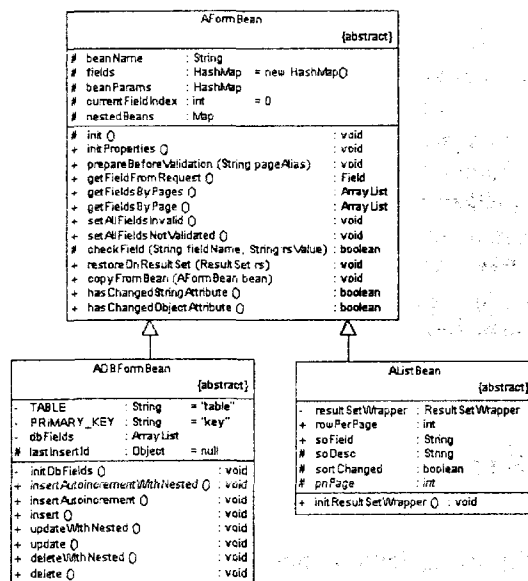


Рис. 1. Объектная модель компонента Model.

ния входных данных. Это позволяет сократить вычислительные ресурсы при необходимости выполнения сложных расчетов и преобразований. Между атрибутами сущностей WADK можно построить отношения на основании определения зависимостей для преобразования строкового значения в объектное.

WADK содержит полностью готовый к использованию компонент Controller, который может быть расширен разработчиком в рамках конкретной разработки. В основные задачи Controller входит перехват пользовательских запросов к системе, определение необходимой последовательности действий для обработки поступившего запроса и передача управления компоненту Model для выполнения бизнес-логики, необходимой для обработки запроса. В задачи Controller также входит формирование ответа пользователю, сформировавшему запрос. На рис. 2 показана объектная модель, представляющая собой компонент Controller framework WADK.

Средством перехвата пользовательских запросов, отправляемых на сервер, выступает фильтр ManagingFilter, который перенаправляет их в ядро компонента Controller, представленное классом BeanManager. BeanManager представляет собой инструмент для анализа запросов, управления сущностями системы и её политикой безопасности, а также формирование ответа пользователю, который определит дальнейший интерфейс с ним. В ядре WADK хранятся описания тех WEB-страниц, для которых фильтр осуществляет перехваты запросов. Это позволяет определять источник запроса, его правильность и автоматически отсекают те запросы, которые не могут быть обработаны системой.

При перехвате запроса пользователя к системе Controller автоматически осуществляет проверку легальности выполнения такого действия на основании политики безопасности. Предлагаемые в WADK средства, для определения политики безопасности конкретной системы, позволяют определять профили пользователей системы, на основании которых осуществляется проверка уровня доступа. В framework существует возможность описания ролей (roles), групп (groups) и способностей (abilities) пользователя, на основании которых формируются различные виды профилей пользователей системы. Источником информации для политики безопасности системы выступает конфигурационный xml-файл, формат которого регламентирован DTD-файлом, входящим в основную поставку WADK. В ходе дальнейшего анализа запроса пользователя определяется последовательность действий, благодаря которым будет сформирован ответ пользователю в ходе исполнения бизнес-логики, размещаемой в компоненте Model.

Для управления сущностями в WADK предлагается использовать действия над ними, в рамках которых возможен вызов бизнес-логики и определение взаимодействия их между собой. Единое управление действиями в BeanManager осуществляется благодаря тому, что все они являются наследниками абстрактного класса ABeanAction. В основную поставку WADK входит разработанный

механизм вызова проверки целостности входной информации, который может быть использован для любой проектируемой системы. Для реализации собственного действия над сущностью в WADK необходимо создать класс-наследник `ABeanAction`, определить над какой сущностью будет осуществляться действие и каким образом оно должно быть выполнено.

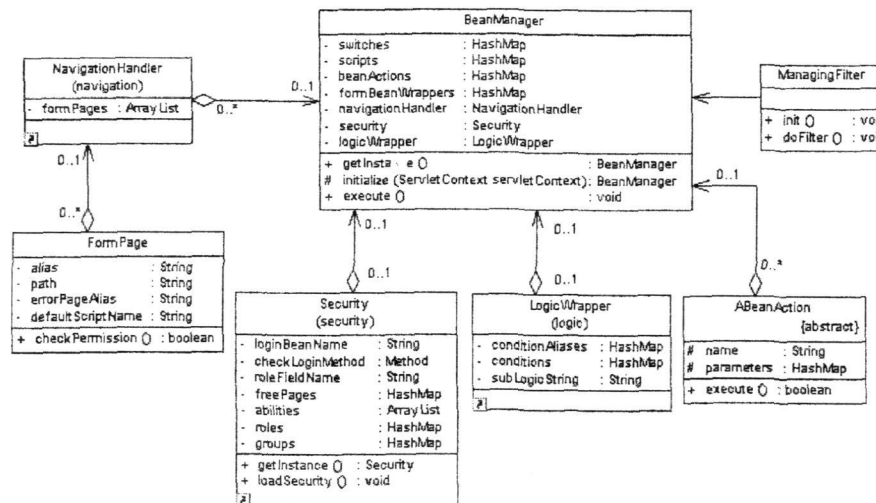


Рис. 2. Объектная модель компонента Controller.

Источником информации для компонента Controller WADK выступает дескриптор развертывания, представленный в виде xml-файла. Благодаря дескриптору развертыванию разработчик может производить гибкую настройку системы, указывая следующую информацию: описание WEB-страниц, которые входят в состав пользовательского интерфейса к системе, описание применяемых сущностей в системе, а также действий выполняемых над ними, определение последовательностей действий для обработки того или иного запроса и т.д. Формат дескриптора развертывания представлен в соответствующем DTD файле, входящим в поставку WADK.

В WADK существует возможность формального описания альтернативных последовательностей действий для обработки запроса. Это достигается благодаря разработанной в WADK внутренней логики, которая дает возможность разработчику создавать формальные логические описания и использовать их при определении последовательности действий, необходимых для обработки поступившего входного запроса, а также при формировании пользовательских интерфейсов. Использование внутренней логики в WADK позволяет использовать логические конструкции в компоненте Controller, которые могут быть заданы формально в дескрипторе развертывания WADK. Объектная модель внутренней логики WADK представлена на рис. 3.

Внутренняя логика WADK позволяет сравнивать значения атрибутов сущности с заданными (класс `FieldCondition`), проверять атрибуты контекста пользователя при обращении к системе (классы `RequestCondition` и `SessionCondition`), использовать профиль пользователя (классы `RoleCondition`, `AbilityCondition` и `GroupCondition`) и т.д. Все выше описанные элементы могут быть объединены в логическое условие (класс `LogicalCondition`), которое в свою очередь может быть использовано в качестве простого элемента. Это позволяет описывать условия произвольной сложности, что расширяет возможности при определении поведения системы.

Компонент View предназначен для создания пользовательского интерфейса, благодаря которому осуществляется диалог пользователя с системой. В качестве основы для компонента View в WADK была выбрана технология `JavaServer Pages (JSP)`, которая позволяет смешивать программный код с HTML, что позволяет эффективно разрабатывать динамические пользовательские интерфейсы. Технология JSP также позволяет разрабатывать собственные тэги, которые могут быть использованы для облегчения процесса создания WEB-страницы. Однако в возможности тэгов входит не только формальное представление структурных элементов страницы, они также могут выступать в качестве логических элементов, которые реагирует на изменения параметров системы и формируют интерфейс по заранее определенным правилам. Как правило такие тэги объединяются в библиотеки и могут быть использованы на любой JSP, после выполнения соответствующих правил по их подключению к ней. Для WADK были разработаны тэги для представления элементов ввода благодаря формализованной структуре компонента Model, настраиваемые тэги для представления

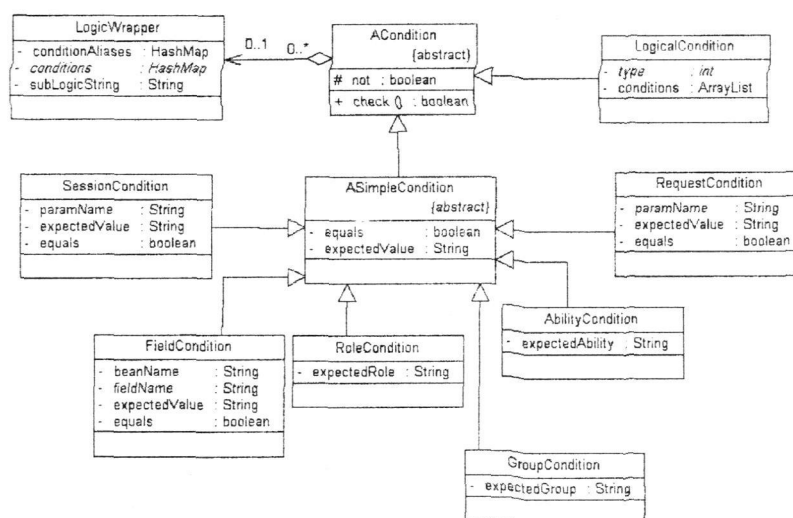


Рис. 3. Внутренняя логика WADK.

пользователю ошибок возникших в ходе проверки целостности входных данных, а также тэги логических элементов (при использовании описанных условий внутренней логики WADK), которые позволяют эффективно создавать динамические пользовательские интерфейсы.

Выводы

Для разработки WEB-ориентированной ИС с использованием framework WADK необходимо выполнить следующие этапы:

- разработать сущности, определяющие поведение системы, определив их атрибуты и правила работы с ними;
- разработать бизнес-логику выполняемых задач каждой сущностью;
- определить политику безопасности системы, на основании встроенной в WADK;
- разработать действия над сущностями, которые будут доступны Controller WADK и будут использованы для управления вызовами бизнес-логики системы с учетом политики безопасности;
- разработать JSP страницы, которые будут использованы в качестве входных интерфейсов к системе;
- определить конфигурационный файл и файл конфигурации политики безопасности.

Инструментальные средства framework WADK удовлетворяют всем поставленным выше требованиям и могут быть эффективно использованы при проектировании ПО WEB-ориентированных ИС. Данное средство было практически применено при создании информационно-справочной системы по продаже недвижимости. При помощи WADK было спроектировано такое ПО, которое оказалось устойчивым и легко расширяемым при поступлении новых требований для системы. Благодаря использованию WADK удалось уменьшить производственные и временные затраты при проектировании ПО системы, что и являлось основной целью его создания.

ЛИТЕРАТУРА:

1. Марти Холл. Сервлеты и JavaServer Pages. - СПб.: Питер, 2001. - 496 с.
2. Фаулер М. Архитектура корпоративных программных приложений: Пер. с англ. - М.: Вильямс, 2004. - 544 с.
3. <http://struts.apache.org/>
4. <http://jakarta.apache.org/tapestry/index.html>

Получено редакцией 30.01.2005.

© Левыкин В.М., 2005.

© Моспан Е.А., 2005.

Левыкин Виктор Макарович, доктор технических наук, профессор, директор Института компьютерных информационных технологий Харьковского национального университета радиоэлектроники, зав. кафедрой информационных управляющих систем.

Моспан Евгений Александрович, студент.

Харьковский национальный университет радиоэлектроники.

УДК 658.012.011.56

Методи реалізації взаємодії програмних модулів, реалізованих мовою Java, з функціональним програмним забезпеченням / Левикін В.М., Жигалов М.А. // Нові технології. – 2005. – № 3 (9). – С. 63.

Вирішена і обґрунтована задача організації взаємодії між програмними модулями, реалізованими за допомогою мови Java, та іншим функціональним програмним забезпеченням. Як приклад, розглядається один з найдосконаліших існуючих інструментальних засобів, що вирішують цю проблему. Базуючись на аналізі недоліків цього засобу, запропонована та обґрунтована схема взаємодії програмних модулів, яка більш відповідає висунутим вимогам та може бути основою функціонування аналогічного інструментального засобу.

Іл. 3. Бібліогр.: 4 найм.

УДК 658.012.011.56

Засіб швидкої розробки програмного забезпечення WEB-орієнтованих інформаційних систем (WADK) / Левикін В.М., Моспан Є.О. // Нові технології. – 2005. – № 3 (9). – С. 68.

Стаття присвячена проблемам, що виникають при розробці ПЗ WEB-орієнтованих ІС, та розглядає застосування засобів швидкої розробки ПЗ (framework), як можливий шлях їх вирішення. Визначені основні вимоги, якими повинен відповідати framework для його ефективного застосування. У зв'язку з тим, що існуючі framework не задовольняють ці вимоги, був створений засіб швидкої розробки ПЗ WEB-орієнтованих ІС WADK (Web Application Development Kit), який відповідає більшості поставлених вимог.

Іл. 3. Бібліогр.: 4 найм.

УДК 681.324

Вплив інформаційних технологій на реінжиніринг бізнес-процесів підприємства / Левикін В.М., Капустін М.Г. // Нові технології. – 2005. – № 3 (9). – С. 73.

Розглянуто проблему впливу інформаційних технологій на реінжиніринг бізнес-процесів. Досліджено фактори, що впливають на реінжиніринг бізнес-процесів, а також характеристики бізнес-процесів, по яких можна визначити необхідність проведення реінжинірингу. Запропоновано способи зменшення дисфункцій у бізнес-процесах з використанням інформаційних технологій, а також оцінки для визначення впливу інформаційних технологій на реінжиніринг бізнес-процесів.

Іл. 3. Бібліогр.: 2 найм.

УДК 681.332.8

Адаптивне використання різних методів зафарбовування при формуванні тривимірних графічних фігур / Романюк О.Н. // Нові технології. – 2005. – № 3 (9). – С. 78.

Запропоновано принцип зафарбовування тривимірних графічних об'єктів, у якому використовується трудомісткий метод Фонга тільки для тих ділянок поверхні, які мають локальну кривизну, або таких, в яких зосереджено зону відблиску або має місце суттєва зміна дифузної складової кольору. В усіх інших випадках використовуються значно простіший метод Гуро.

Табл. 1. Іл. 6. Бібліогр.: 6 найм.

УДК 002.66

Основні завдання управління IT-сервісами / Петренко В.Р., Лимаренко А.С. // Нові технології. – 2005. – № 3 (9). – С. 87.

В статті розглянута коротка історія розвитку управління IT-сервісами, описані такі центральні елементи IT-сервіс-менеджменту, як "надання послуг" та "підтримка послуг". Розглянуто склад цих елементів, основні задачі, які ними вирішуються. Намічені можливі шляхи вирішення цих задач, а також напрямки для подальших досліджень.

Іл. 2. Бібліогр.: 4 найм.

УДК 502.51(285):556.013:556.06

Моделювання і прогнозування екологічної ситуації Кременчуцького водосховища / Дядик Д.О., Зюман Б.В., Пасенко А.В., Підпорина

УДК 658.012.011.56

Methods of realization of the interaction between the Java software modules and functional software / Levikin V.M., Zhigalov M.A. // New technology. – 2005. – № 3 (9). – P. 63.

The task of realization of the interaction between the Java software modules and other functional software was solved and based. As example, one of the best of the existing tools, which solves this problem, was examined. Basing on the analysis of its shortcomings the scheme of the interaction of the software modules, which better corresponds to the demands, was suggested and based. This scheme can be used as the base of functioning of a similar tool.

Fig. 3. Ref.: 4 items.

УДК 658.012.011.56

Tool of rapid development of software of WEB-oriented informational systems (WADK) / Levikin V.M., Mospan E.O. // New technology. – 2005. – № 3 (9). – P. 68.

The article is devoted to the problems, which appear during the development of software of WEB-oriented informational systems, and consider use of tools of rapid development of software (framework) as a possible way for their decision. Main demands are pointed out, which framework must satisfy for its effective use. Because existing frameworks don't satisfy pointed demands, new quick developing means of programm software of WEB-oriented informational systems WADK (Web Application Development Kit) was created, which satisfies most formulated demands.

Fig. 3. Ref.: 4 items.

УДК 681.324

Influence of information technologies on reengineering of business-processes of the enterprise / Levikin V.M., Kapustin M.G. // New technology. – 2005. – № 3 (9). – P. 73.

The problem of influence of information technologies on reengineering of business-processes is considered. The factors influencing reengineering of business-processes, and also characteristics of business-processes on which it is possible to define necessity of carrying out of reengineering are investigated. Ways of reduction of dysfunctions in business-processes with use of information technologies, and also estimations for definition of influence of information technologies on reengineering of business - processes are offered.

Fig. 3. Ref.: 2 items.

UDC 681.332.8

Adaptive use of different shading methods while forming three-dimension graphic objects / Romanyuk O.N. // New technology. – 2005. – № 3 (9). – P. 78.

A principle of shading three-dimension graphic objects using labor-consuming Phong method solely for surface spots having local curvature or those concentrating reflection zone or having subtextional diffusion color component is presented. For other cases much simpler Gouraud method is used.

Tab. 1. Fig. 6. Ref.: 6 items.

UDK 002.66

The basic tasks of IT-service-management / Petrenko V.R., Limarenko A.S. // New technology. – 2005. – № 3 (9). – P. 87.

The article highlights the outlines of IT-services control history and describes such central IT services-management elements as "Service Support" and "Service Delivery". The authors discuss the components of these elements, basic tasks that are solved by means of these elements. The article suggests the possible ways of the given problems solving and directions for further investigations.

Fig. 2. Ref.: 4 items

UDC 502.51(285):556.013:556.06

Design and prognostication of ecological situation of the Kremenchug storage pool / Dyadik D.A., Zuyman B.V.,