

Администрирование содержимого информационной системы непрофессионалами в области ИТ

А.А. Степанов

Comwell Systems Limited Company, Рига, Латвия

alex@comwellsys.com

В.С.Шитиков

Рижский технический университет, Рига, Латвия

vss@latnet.lv

АННОТАЦИЯ

Рассматривается решение, обеспечивающее как достаточную гибкость и модифицируемость информационной системы, так и относительно невысокую стоимость ее разработки.. В основе решения лежит трех-модульная архитектура, где одна часть системы жестко закодирована и не подлежит изменениям, вторая требует изменений крайне редко и может быть изменена только ИТ-профессионалом, третья же – самая динамичная, может изменяться самим конечным пользователем с помощью специального Средства Управления Содержимым. Система Управления Пользователями, предусматривает различные типы Администраторов (Редакторы, Публикаторы и т.д.) с четко определенными правами доступа, ролями и ответственностью. Приводится пример конкретной системы, построенной по указанным принципам.

Ключевые слова

информационная система, содержимое, пользователь, управление.

Введение

Современные методы создания и сопровождения интерактивной информационной системы, основанной на Web-технологии (в дальнейшем именуемой в данном документе “сайт”), безусловно, не ограничиваются простой разработкой дизайна сайта и созданием страниц будущего сайта при помощи HTML-редакторов. Необходимость оперативного обновления информации на сайте и требуемая для этого высокая квалификация персонала вынуждают прибегать к внедрению комплексных программных решений поддержки сайта, где наиболее квалифицированный ИТ-персонал (для простоты именуемый в дальнейшем “программисты”) занят, в основном, лишь на стадиях разработки и внедрения проекта. В то время как текущее сопровождение сайта (распределение прав и обязанностей, обновление информации и т.д.) выполняется людьми (в дальнейшем именуемыми “пользователями”), не являющимися “программистами”, зато имеющими непосредственное отношение к содержимому сайта. Чтобы обеспечить возможность самостоятельной поддержки содержимого информационной системы непосредственно владельцами информации, не являющимися профессионалами в области ИТ, необходимо предоставить им дружелюбное, простое, безопасное и в то же время, обладающее достаточными изобразительными возможностями средство. Это средство оперирует такими понятными для пользователя категориями как Дерево структуры содержания, Формы ввода, Шаблоны и Редактор Шаблонов. Средство должно также обеспечивать эмуляцию внешнего пользователя (Посетителя информационной системы) для

просмотра информации перед ее опубликованием.

Поскольку разные компоненты информации могут иметь разных ответственных лиц, то в целях обеспечения безопасности необходима гибкая Система Управления Пользователями, предусматривающая различные типы Администраторов (Редакторы, Публикаторы и т.д.) с четко определенными правами доступа, ролями и ответственностью.

Термин “посетители” зарезервируем за теми, для кого собственно и предназначена информация на сайте.

Теоретическая часть

Хорошо реализованная система управления сайтом должна обеспечивать:

- возможность оперативного обновления или исправления содержимого каждой страницы;
- интуитивно понятную систему навигации;
- модификацию иерархии элементов сайта (контекста) и элементов дизайна (компонент).

Система должна быть максимально защищенной от ошибок пользователя, как преднамеренных, так и непреднамеренных – вызванных недостаточной квалификацией.

Защита от ошибок в первую очередь реализуется строгим разграничением прав и ролей - использованием шаблонов (templates) и системой авторизации (администрированием).

Для гарантии логической целостности и связи содержания в разных частях сайта, а также для экономии времени пользователя желательно предусмотреть возможность создаваемых автоматически элементов сайта. Это может быть:

- программная организация ссылок на множественный одностипный контекст (заголовки новостей, поисковая система, карта сайта);
- автоматические системы замены содержания после истечения определенного промежутка времени или чрезмерного наполнения заданного ресурса.

Система управления сайтом при этом должна обеспечивать централизованное хранение данных сайта и по возможности избавить пользователей от задачи размещения данных непосредственно в файловую систему.

В большом количестве случаев от сайта требуется также обеспечение таких функциональных возможностей, как:

- обработка форм посетителя;
- обеспечение возможности “диалога” с посетителем (форумы, гостевые книги, листы рассылки);
- создание файловых хранилищ;
- системы скрытого слежения за посетителем на сайте (время нахождения на сайте, маршрут продвижения, периодичность посещения и т.д.).

Система авторизации предусматривает в первую очередь:

- различные категории пользователей и посетителей;
- защиту от несанкционированного доступа - разграничение доступа к отдельным страницам и элементам сайта, будь то внешние (посетительские) страницы сайта или внутренние (управленческие) в соответствии с категорией пользователей и посетителей.

Функционально подобные системы можно разделить на две части:

- Посетительская (направленная вовне) часть, отвечающая за форматирование страниц в соответствии с шаблонами, непосредственное отображение страниц сайта, и т.д.;
- административная (направленная внутрь) часть, отвечающая за наполнение сайта конкретной информацией и ее обновление, за обеспечение связей и контроль за используемыми компонентами сайта.

Структурно схема сайта может выглядеть как набор вложенных друг в друга элементов:

- страниц;
- управляющих элементов-ссылок;
- элементов дизайна;
- отдельных функциональных элементов (форм, непосредственно частей HTML кода и т.д.).

Каждый из элементов сайта подчинен некоторому шаблону. Шаблоном элемента сайта в данном случае можно назвать формализованный каким-либо образом набор правил для описания (форматирования) визуального представления этих элементов. Применением различных шаблонов к одним и тем же элементам достигается автоматизация и целостность представления информации на сайте.

Процесс создания подобных систем можно разделить на несколько основных этапов:

- создание “инструмента” для будущей системы - комплекса программных средств для анализа и форматирования исходных данных в понятную для конечного пользователя форму. Этот этап в свою очередь состоит из обычных этапов программирования (анализа и оценки проекта, выбора основных программных и аппаратных средств, непосредственно программирования);
- создание макета–дизайна сайта;
- создание шаблонов элементов сайта;
- непосредственное наполнение сайта информацией и создание соответствующих связей.

Важным моментом является соблюдение баланса в разделении полномочий по управлению функциональностью. Можно выделить двух основных “участников” процесса управления:

- автоматическая программная (компьютерная) часть;
- ручная операторская (человеческая) часть.

По мере смещения баланса ответственности за функционирование системы в сторону операторской части повышаются требования к квалификации обслуживающего персонала. Перекалывание же обязанностей обслуживающего персонала на программную часть ведет к усложнению кода и, соответственно, к увеличению стоимости разработки, а зачастую и к снижению гибкости системы. Поэтому кажется логичным введение промежуточного звена – административной части, ответственной за:

- построение элементов сайта;
- их встраивание в дизайн;
- изготовление шаблонов страниц;
- управление доступом к системе.

Для изменения этой части системы нужна высокая “программистская” подготовка. Однако эти элементы системы меняются крайне редко, и требуется лишь эпизодическое (при существенной реорганизации сайта) участие ИТ-персонала. В обычном же режиме сайт эксплуатируется самими пользователями, работающими с созданной для них административной частью. Тем самым становится возможным сохранить приемлемую стоимость разработки системы, и, в тоже время, не выдвигать чрезмерных требований к обслуживающему сайт персоналу.

Реализация

Далее следует описание системы, разработанной в компании Comwell Systems. Система управления сайтом реализует описанные выше принципы и построена на базе технологии Java-сервлетов [Hunter J., et al., 1998]. Java-сервлеты представляют собой расширение Web-сервера. Это Java-класс, который, загружаясь динамически, увеличивает функциональность сервера. Сервлеты могут служить заменой CGI скриптов. Сервлет похож на обычное расширение Web-сервера, однако сервлет исполняется внутри Виртуальной машины Java (JVM - Java Virtual Machine), что повышает безопасность продукта [Hunter J., et al., 1998]. Другое преимущество сервлетов – переносимость, как в отношении операционных систем, на которых может функционировать Java, так и в отношении Web-серверов. В настоящее время все основные Web-сервера поддерживают сервлеты, либо как встроенную возможность (Standalone Servlet Engines), либо как дополнительно встраиваемую возможность (Add-on Servlet Engines).

Кроме того, использование данной технологии позволяет унифицировать доступ к внешним источникам информации. Практически любая СУБД, поддерживающая интерфейс JDBC (Java DataBase Connectivity) [Shah R., 1996], может быть использована в качестве хранилища данных сайта.

Исходные данные для построения системы можно разделить в зависимости от ролей обслуживающего персонала.

Роль администратора - создание каркаса сайта, описывающего элементы дизайна и расположение в нем отдельных функциональных модулей (управляющих элементов, внутренних страниц сайта и т.д.). Наилучшим средством реализации этой части системы является HTML (HyperText Markup Language) [W3C, 1999]. Эта часть должна быть описана так, чтобы в последствии подвергаться наименьшим изменениям. Именно за счет этой части снижаются требования к „программистской” подготовке оператора.

Роль оператора - наполнение сайта конкретной информацией и координация управляющих элементов. Сюда же относится создание шаблонов для визуального представления информации.

Содержимое сайта хранится на SQL-сервере, связанном с системой посредством интерфейса JDBC. Для представления шаблонов используется XML (eXtensible Markup Language) [Bray T., et al., 2000], так как этот формат наилучшим образом подходит для иерархического представления данных. Оператор предельно отстранен от непосредственного изменения данных сайта в базе (физическом хранилище данных) - всё необходимое обеспечивается системой управления сайтом.

Программно сайт состоит из набора компонентов (Java-классов), которые на основании исходных данных и шаблонов строят соответствующий элемент сайта. В данной реализации представлены следующие компоненты:

- простая страница – внутренняя страница сайта, форматированная в соответствии с определенным шаблоном;
- повторяющаяся страница – набор простых страниц;
- ссылка на страницу;
- набор ссылок на страницу;
- форум – по существу, набор иерархических ссылок на страницы;

- компонент обработки данных посетителя при заполнении им HTML-формы, которая может быть описана шаблоном;
- компонент контроля прав пользователя для доступа к защищенным элементам сайта;
- меню.

По мере необходимости, возможно динамически добавлять компоненты.

Базовая структура административной части, необходимая для обеспечения описанных выше функциональных возможностей, приведена на рис. 1. Визуально она выполнена в виде многоуровневого меню.

Наполнение каждого элемента информационного дерева может динамически управляться оператором. Изменение же самого дерева может быть выполнено только специалистом на уровне HTML-кода.

Для секций, где информация меняется часто (новости, статьи) или изменения имеют прикладную ценность, может быть организован архив.

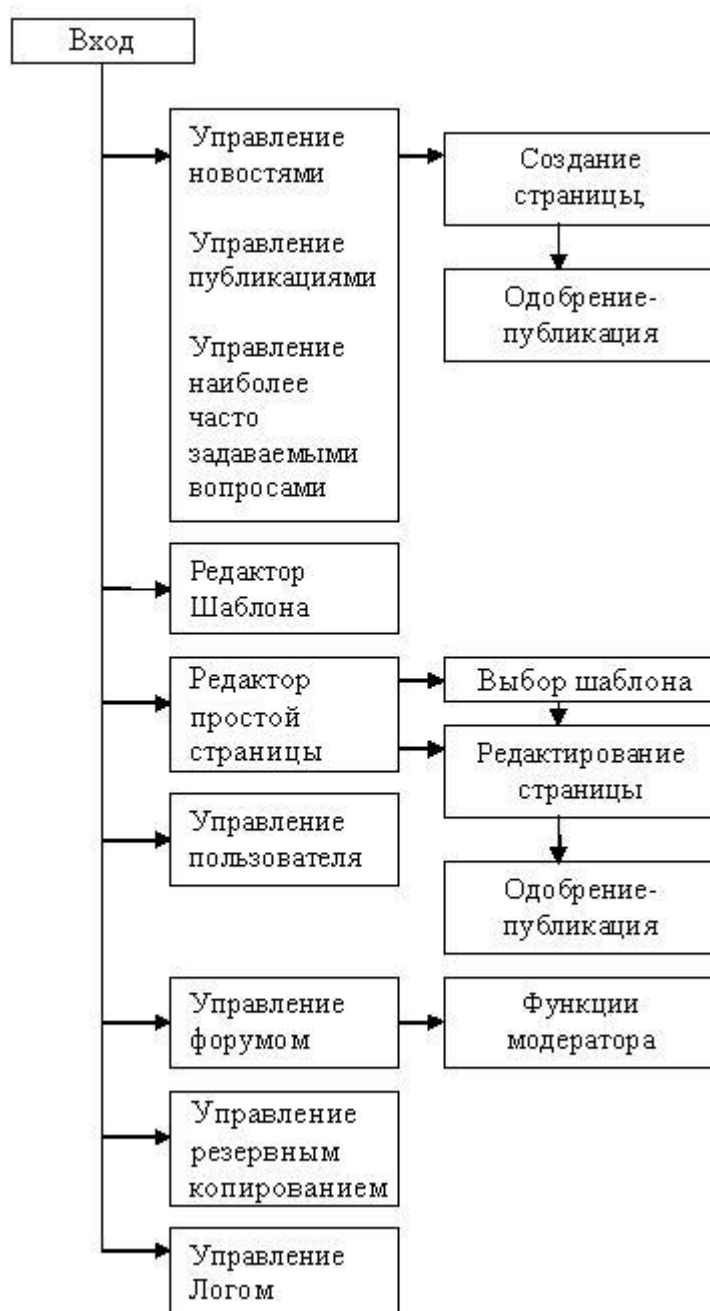


Рис. 1. Структура административной части.

Любой административный инструмент должен отвечать политике безопасности организации. Поскольку это внутренняя проблема организации, права и обязанности любого пользователя должны быть ясно определены и одобрены на процедурном уровне. Хорошая основа для объяснения и понимания ролей и обязанностей – диаграммы (flowchart) типичных действий. На рис. 2 показан пример очень простого действия – добавление нового элемента сайта. Для внедрения системы управления сайтом необходимо иметь определение процедурной роли каждого пользователя в системе, будь то простой посетитель, посетитель с правами доступа к защищенным элементам сайта, оператор или администратор.

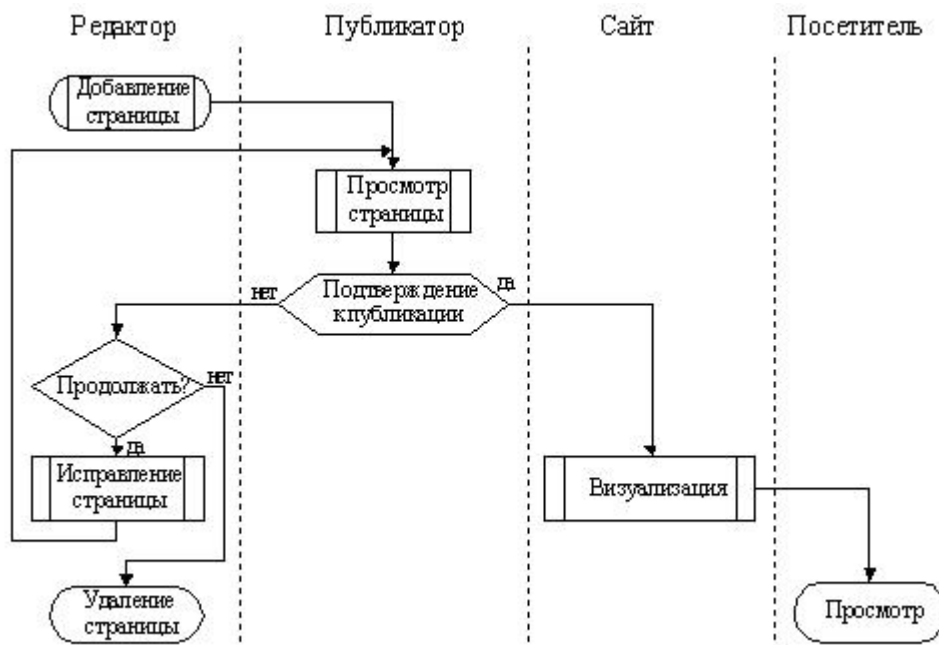


Рис. 2. Пример диаграммы деятельности (flowchart).

Один из вариантов классификации пользователей с соответствующими правами приведен в таблице 1. Разделение на группы пользователей условно, существует возможность создавать дополнительные группы, изменять их полномочия для каждой функции.

Табл. 1. Ролевые функции.

Функция	Посетитель	Привилегированный посетитель	Редактор	Публикатор	Администратор
Просмотр сайта	•	•	•	•	•
Просмотр закрытых страниц сайта		•	•	•	•
Использование форума		•	•	•	•
Вход в систему администрирования			•	•	•
Добавление новых пользователей					•
Редактирование существующих пользователей					•
Удаление пользователей					•
Создание новых страниц			•		•

Редактирование неопубликованных страниц				•			•
Редактирование опубликованных страниц					•		•
Принятие страниц к публикации					•		•
Удаление неопубликованных страниц				•		•	•
Удаление опубликованных страниц					•		•
Создание шаблона							•
Редактирование шаблона							•
Удаление шаблона							•
Управление форумом					•		•

Разные сотрудники с различными идентификаторами пользователей могут быть ответственны за различные части сайта. Каждая категория информации должна иметь пользователя, ответственного за эту категорию (редактора). Без одобрения другого пользователя, наделенного правами разрешения публикации (публикатора), информация не должна появляться на сайте для публичного обозрения. Таким образом распределяются права и обязанности. Так как информация сайта может иметь иерархическую структуру, то возможно реализовать присвоение прав по умолчанию. Если какая-либо категория не имеет конкретного ответственного, то ответственность за нее автоматически делегируется ответственному на уровень выше. Для каждой ветви информационного дерева есть возможность создать множество редакторов и одного публикатора. Права доступа могут быть определены и для групп пользователей.

Анализ и оценка разработки

Программное средство, разработанное в соответствии с изложенными выше принципами, нашло применение в нескольких Web-приложениях, различающихся как по назначению, так и по структуре информации и внешнему дизайну. Скелетная часть информационного дерева настраивалась разработчиками и входила в комплект поставки. В каждом из приложений реализована своя политика распределения ролей и ответственностей. Эксплуатация внедренного продукта всюду проходит без обращений клиентов к разработчикам за консультациями или модификациями, что говорит либо об удачном первоначальном дизайне, либо о том, что местные программисты удачно справляются с чужим HTML-кодом, либо о том и другом.

Заключение

Описан подход к построению информационной системы, основанной на Web-технологии. Главная идея подхода заключается в поиске компромисса между гибкостью (модифицируемостью) системы и стоимостью ее разработки. Для достижения этого компромисса предлагается трех-модульная архитектура, включающая: а) неизменяемую часть, б) часть, пригодную для изменений только с привлечением специалистов (непосредственно в HTML-коде), с) часть, пригодную для изменений непосредственно самими конечными пользователями. Последняя часть реализована в виде Системы Управления Сайтом, которая

состоит из Системы Управления Пользователями и Системы Управления Содержимым. Первая из них обеспечивает возможность гибкого многоуровневого распределения ролей пользователей, вторая включает средства создания и редактирования шаблонов, наполнения шаблонов конкретной информацией и встраивания создаваемых элементов в структуру информационного дерева.

Литература

[Bray T., et al., 2000] Bray T., Paoli J., Sperberg-McQuenn C.M., Maller E. Extensible Markup Language (XML) 1.0 (Second Edition) Copyright © 2000 W3C®/ Internet. -

<http://www.w3c.org/TR/REC-xml>.

[Hunter J., et al., 1998] Hunter J., Crawford W. JAVA Servlet Programming ISBN 1-56592-391-XE First edition, October 1998.

[Shah R., 1996] Shah R. Integrating Databases with Java via JDBC// JavaWorld May 1996/ Internet.

- http://www.javaworld.com/javaworld/jw-05-1996/jw-05-shah_p.html.

[W3C, 1999] W3C. HTML 4.01 Specification. W3C Recommendation 24 December 1999/ Internet.

- <http://www.w3.org/TR/REC-html40/>.