

С.В. Юрьев

УНИВЕРСАЛЬНАЯ СИСТЕМА ПОСТРОЕНИЯ И АДМИНИСТРИРОВАНИЯ ЛАБОРАТОРНЫХ ВЕБ-САЙТОВ

ВВЕДЕНИЕ

Раньше основная часть ресурсов в сети строилась с использованием статических страниц, написанных на HTML. Эти страницы редактировались в каком-нибудь текстовом или HTML-редакторе, после чего закачивались на хостинг-сервер (чаще всего по ftp).

Однако со временем в таком способе администрирования обнаружились серьезные недостатки. По мере увеличения количества страниц утрачивалась прозрачность структуры сайта, все труднее становилось находить и редактировать нужные страницы. Возрастала вероятность появления так называемых «мертвых ссылок», указывающих на несуществующие более страницы. Кроме того, назрела необходимость использования различных интерактивных элементов, форм регистрации пользователей, поиска, каталогов и т.д. Тогда и стали появляться первые «бэкофисы»¹ — системы администрирования веб-ресурсов.

Первые системы создавались под конкретные проекты. Основным их достоинством была высокая степень индивидуализации, однако они также обладали рядом серьезных недостатков. В них, как правило, намертво вшивался дизайн сайта, а возможности администрирования были ограничены некоторым набором программных функций, не поддающимся самостоятельному масштабированию и расширению возможностей без участия его разработчиков. Чтобы добавлять новые функции или тиражировать уже существующие, владельцам сайта снова приходилось привлекать команду программистов, что было крайне нецелесообразно с точки зрения временных и материальных затрат.

Именно тогда назрела необходимость в универсальных CMS² — системах управления содержанием ресурса. К концу 90-х гг. именно такие сис-

¹ От англ. back office — вспомогательный офис

² От англ. content management system — в разных источниках переводится как система управления содержанием или система управления контентом

темы стали новым этапом в индустрии веб-разработок. Появился рынок CMS для малых, средних и крупных веб-порталов.

В данный момент на российском рынке представлены более шестидесяти широко известных профессиональных систем управления [2,3] и еще сотни менее раскрученных пакетов. Их цена колеблется в среднем от \$1000 до \$3000, они ориентированны на малый и средний бизнес. Различные enterprise-решения от таких гигантов как Microsoft и IBM, стоящие сотни тысяч долларов, подходят лишь для промышленных гигантов и потому в поле зрения нашего исследования не попадают.

Основной задачей этих систем является разделение содержания сайта и его представления, а также предоставление пользователю удобных инструментов для публикации и изменения содержания при условии, что навыки пользователя не превосходят умения работать с Интернетом (точнее браузером) и офисными приложениями, такими как Microsoft Word и т.п. При этом представление содержания чаще всего ограничено форматом HTML. Само содержание имеет неоднородную структуру. Помимо обычных текстовых документов пользователи сайта работают с такими объектами, как ленты новостей, форумы и т.п. CMS предоставляют пользователям набор инструментов для работы с наиболее типичными объектами содержания, что вполне достаточно для поддержки большинства сайтов. Однако для поддержки лабораторных сайтов предоставляемых инструментов зачастую недостаточно, поскольку в лабораториях работают с информацией, структура которой зачастую кардинально отличается от структуры объектов, которые представлены в системах. Публиковать эту информацию на сайте с помощью таких систем крайне неудобно.

Целью работы являлась разработка CMS, предоставляющей возможность удобной работы с различной информацией, нетипичной для среднестатистических сайтов. Кроме того, система должна иметь возможность публиковать информацию не только в HTML, но и в других форматах, чтобы осуществлять её автоматическую обработку. При этом вновь разрабатываемая CMS не должна потерять тех преимуществ, которыми обладают уже существующие системы.

В результате проделанной работы была разработана CMS, удовлетворяющая всем поставленным задачам. Она позволяет более гибко работать с информацией, публиковать ее в различных форматах. При этом выгодные свойства других CMS не только не утрачиваются, но и реализуются еще более удачно. В частности, задача отделения информации от представления решается без наложения всяческих ограничений на последующую изменяемость структуры и дизайна сайта, чего нельзя сказать о других систе-

мах, которые все же такие ограничения накладывают. На базе разработанной CMS создается веб-сайт лаборатории. С его помощью система была проверена на соответствие условиям, описанным в постановке задачи.

Текст работы построен следующим образом. В разделе «Обзор предметной области» рассказывается о том, что такое CMS и какие задачи она решает. Также в ней описывается общая схема работы большинства CMS. Далее следует постановка задачи, выделяется ряд требований к разрабатываемой системе. Требования разделяются на общие для всех CMS и специфичные для конкретной системы ввиду того, что она должна быть оптимизирована для создания лабораторных сайтов. В разделе «Описание решения задачи» вначале дается сжатое изложение решения, его удовлетворение требованиям, рассмотренным в постановке задачи, кратко описываются преимущества предложенных механизмов работы по сравнению с существовавшими ранее. После этого следует подробное описание решения, рассматриваются все объекты системы и их взаимодействие между собой и с администраторами сайта. Рассматриваются инструменты, предоставляемые пользователю для управления сайтом. В разделе «Механизм генерации готовой страницы раздела» подробно описывается процесс сборки страниц, соответствующих разделам, запрашиваемым посетителем сайта. Заканчивается глава описанием функций, входящих в ядро системы, содержащее всю логику работы сайта. В заключение описываются результаты проведенной работы, их практическое применение и направления дальнейшей работы.

1. ОБЗОР ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. Понятие CMS систем

Система управления контентом [1], иначе называемая CMS, — это программное обеспечение, которое позволяет публиковать и изменять опубликованную на сайте информацию самостоятельно, без привлечения разработчиков сайта. Подразумевается, что от пользователей такой системы не требуется специальных знаний технологий, отличающихся от обычно используемых в офисных процессах (текстовый редактор, Интернет и т.п.). При этом не следует считать, что такая система не требует обучения персонала, но это обучение касается порядка работы в системе, а не изучения новых технологий.

Большинство CMS можно разделить на среду разработки, т.е. инфраструктурную систему, обеспечивающую функциональность и хранение ин-

формации, и среду редактирования, интерфейс работы с пользователем. В большинстве современных CMS среда разработки базируется на той или иной СУБД, может включать сервера приложений, а среда редактирования имеет веб-интерфейс и допускает использование стандартных офисных пакетов редактирования документов (текстовые редакторы, электронные таблицы, средства создания презентаций, почтовые системы и т.п.). При этом вся функциональность, сложность разработки и администрирования сосредоточены в среде разработки, а пользовательские свойства — в среде редактирования. Обобщенная типичная схема работы сайта, использующего CMS, представлена на рис. 1.

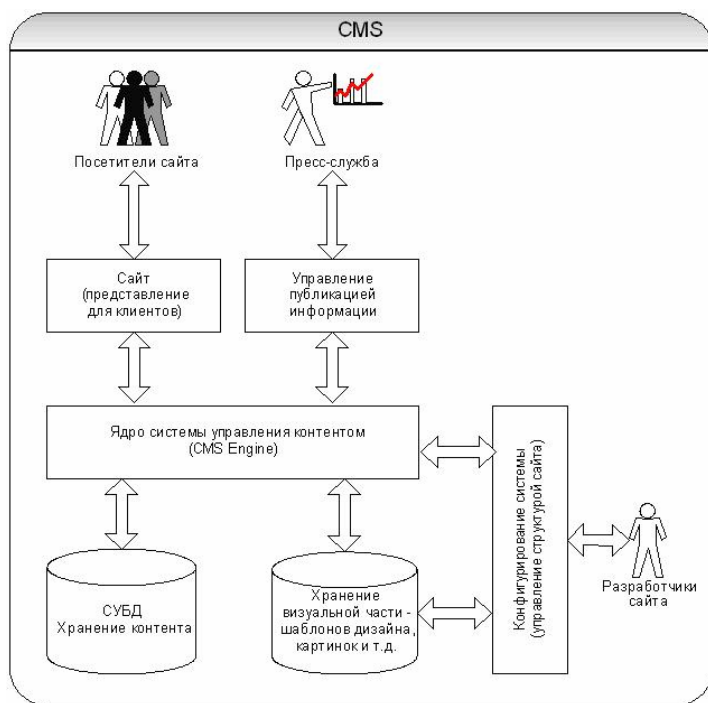


Рис. 1. Схема работы сайта с использованием CMS

В системе присутствуют два хранилища. В первом (обычно реляционная СУБД) хранятся все данные, которые публикуются на сайте. Во втором (обычно файловая система) хранятся элементы представления — шаблоны,

графические изображения и т.д. Кроме внешнего представления сайта, каким его видят все пользователи, есть как минимум два специализированных рабочих места. Первое рабочее место — для разработчиков сайта. С его помощью они задают структуру сайта, структуру содержания, определяют внешний вид сайта, настраивают шаблоны дизайнера. Этот инструментарий обычно не полностью автоматизирован. Для настройки сайта разработчики частично работают через средства CMS, часть информации размещается напрямую (т.е. идет работа с файлами на сервере и с СУБД). Второе рабочее место — для владельцев сайта. Оно позволяет сотрудникам компании самостоятельно размещать информацию на сайте, без участия разработчиков. Менеджеры заказчика работают только через специализированное рабочее место. Прямого доступа к файлам и СУБД они не имеют.

Использование CMS предоставляет следующие преимущества.

1. Оперативное обновление информации — информацию публикует человек (сотрудник компании), владеющий информацией, без дополнительных посредников в виде технических специалистов.
2. Возможность реализации дополнительных сервисов, таких как форумы, голосования и т.д. Они зачастую требуют интерактивного взаимодействия с пользователем.
3. Уменьшение сроков и затрат на разработку сайта — наиболее востребованная функциональность уже реализована в CMS и может быть сразу использована.
4. Повышение качества разработки — использование автоматизированных средств разработки позволяет сохранять прозрачность структуры сайта и избегать большинства ошибок и неудобств, сопутствующих работе с сайтами, не использующими CMS.
5. Упрощение дальнейших модификаций — CMS позволяют разделить и независимо редактировать данные и их представление.
6. Возможность совмещения работы нескольких человек над одним сайтом с разделением их прав и обязанностей.

Чтобы обеспечить данные преимущества, CMS должна решить следующие основные задачи.

1. Публикация информации человеком, не знакомым с технологиями разработки сайтов.
2. Разделение данных и их представления.
3. Организация работы нескольких человек над сайтом.

2. ПОСТАНОВКА ЗАДАЧИ

Поскольку областью использования разрабатываемой системы являются именно лабораторные сайты, то система должна быть способна реализовывать работу с динамическими информационными объектами, нетипичными для среднестатистических сайтов и систем. Например, если в лаборатории идет разработка некоторого программного проекта, то могут возникнуть нижеперечисленные задачи:

1. Публикация результатов работы в Интернете. При этом должна быть возможна публикация в различных форматах, отличных от традиционного HTML. Это дает возможность их точной автоматической обработки другими приложениями.
2. Предоставление сторонним лицам возможности воспользоваться программной функциональностью разрабатываемого проекта. Иными словами, предоставить возможность использования некоторого приложения через Интернет.

Система должна предоставлять инструменты решения этих задач. Кроме того, система должна обладать следующими свойствами.

1. Универсальность — система должна представлять собой универсальное средство для занесения и обновления информации, которое можно было бы применить к сайтам и разделам с различной визуальной и логической структурами.
2. Настройки безопасности, разграничение прав пользователей на редактирование сайта.
3. Возможность формирования динамической структуры сайта — структура сайта должна быть легко изменяемой, прозрачной и четко выраженной, что, в свою очередь, подразумевает следующие аспекты реализации:
 - гибкость и прозрачность архитектуры — наличие возможности беспрепятственно отслеживать и изменять структуру сайта, данных;
 - возможность беспрепятственного редактирования дизайна;
 - возможность внедрения приложений независимых разработчиков;
 - модульная структура подключения объектов функционала;
 - разделение данных и их представления.
4. Возможность хранения различных видов информации: не только одностраничных документов, но и динамических информационных

блоков, таких как ленты новостей и другие приложения как типичные, так и индивидуальные.

5. Простота в использовании. Для использования системы веб-мастеру должно быть достаточно знания офисных приложений. Здесь важно понимать различие между веб-разработчиком и веб-мастером, поскольку первый отвечает за установку сайта, его начальную настройку, подключение дополнительных модулей, управление дизайном, а второй — за добавление и изменение информации как статической, так и динамической.

3. ОПИСАНИЕ РЕШЕНИЯ ЗАДАЧИ

3.1. Краткое описание решения

Сайт, построенный на вновь разработанной системе, можно условно разбить на несколько глобальных объектов.

1. Дерево разделов — иерархически упорядоченное множество разделов сайта, каждый из которых имеет одного родителя и сколько угодно дочерних разделов. Исключение составляет корневой раздел, у которого нет родителя.
2. Множество пользователей/администраторов — людей, обладающих различными правами на просмотр и редактирование содержания и структуры сайта.
3. Множество информационных пакетов — объектов, несущих некоторую информацию, которую нужно разместить на сайте. Такими объектами могут быть, например, текстовые документы, ленты новостей и т.п. Каждый пакет имеет свой тип, которому соответствует ровно одно приложение, позволяющее его редактировать.
4. Множество шаблонов отображения — объектов, несущих информацию о том, как должна информация отображаться на сайте.
5. Множество приложений — программ, отвечающих за обработку содержания информационных пакетов. Число этих приложений равно числу типов информационных пакетов, поскольку одно приложение отвечает за один тип. Множество приложений не является фиксированным. В него можно добавлять новые приложения в соответствии с нуждами проекта.

Все эти объекты взаимосвязаны и образуют основное ядро сайта, обеспечивающее ему основу для работы и дальнейшего развития. Информационные пакеты размещаются в разделах сайта, которые отображаются посе-

- Публикация содержания сайта возможна в любом формате, основанном на технологии XML, поскольку внутри CMS все информационные объекты и документы, составляющие содержание сайта, описываются с помощью языка, также основанного на XML, а перед тем как быть отправленным пользователю любой документ претерпевает XSL-трансформацию в нужный формат. Фактически эта трансформация играет роль шаблона отображения информации. Это также позволяет публиковать документы не только для просмотра пользователем, но и для автоматической обработки документов другими приложениями.
- Гибкость и прозрачность архитектуры реализована за счет построения явного дерева разделов, хранящегося в базе данных. В обычных CMS структура сайта ассоциируется с расположением файлов, соответствующих разделам на жестком диске. Это делает структуру сайта менее прозрачной, поскольку явная иерархическая связь между разделами отсутствует.
- Система универсальна — за счет реализации независимой работы с разделами, информационными пакетами и шаблонами отображения, о которых речь пойдет ниже. Она позволяет реализовывать сайты с самой разнообразной структурой.
- Права пользователей на редактирование сайта строго разграничены за счет контроля прав пользователей на доступ к разделам и информационным пакетам.
- Работа с дизайном и вообще с форматами представления информации на сайте производится за счет шаблонов представления, которые можно беспрепятственно изменять, независимо от структуры содержания сайта.
- В систему можно внедрять приложения, реализующие работу с нетипичной информацией, как статической, так и динамической, с помощью механизма информационных пакетов, которые являются своего рода интерфейсом взаимодействия между ядром системы и другими приложениями. Благодаря этому механизму функциональность внедряемых приложений можно свободно тиражировать, например, создавать неограниченное количество лент новостей на сайте.
- Данные отделены от представления, поскольку всю работу с данными осуществляют подключенные приложения, а за представление отвечают шаблоны отображения.

- Разнообразие информационных объектов, работа с нетипичной информацией реализуется также при помощи специальных приложений, подключаемых к системе через информационные пакеты.
- Пользователю для публикации информации на сайте не нужно обладать широкими знаниями в области Интернет-технологий, поскольку вся работа, затрагивающая эти технологии, либо автоматизирована, либо проводится один раз разработчиком сайта в начале его работы.

Далее мы рассмотрим работу всех объектов системы более подробно.

3.2. Дерево разделов

Представляет собой множество объектов, у каждого из которых есть родитель, для которого он, в свою очередь, является дочерним объектом. Таким образом, у каждого объекта-раздела может быть только один родитель и любое количество дочерних разделов.

К каждому разделу можно привязать любое количество информационных пакетов, речь о которых пойдет ниже. Также к разделу привязывается шаблон отображения из базы шаблонов, который используется при генерации готовой страницы. Более подробно речь о шаблонах также пойдет ниже.

3.3. Множество пользователей и администраторов

Множество пользователей и администраторов — это база данных, содержащая следующие данные: id, логин, пароль, тип пользователя, информация о пользователе, его права. Пользователи могут быть шести типов.

1. *Рут* — может редактировать дерево и свойства разделов, базу пользователей, пакетов, приложений, шаблонов, а также содержание информационных пакетов. Вообще говоря, это пользователь, которому доступен полный контроль над сайтом. Хотя система не запрещает наличия нескольких пользователей такого типа, все же в идеале Рут в системе должен быть один.
2. *Креатор*³ — может редактировать дерево и свойства разделов, базу пакетов, шаблонов, а также содержание информационных пакетов. Иными словами, этому пользователю доступен весь инструментарий системы, за исключением работы с пользователями.

³ От англ. Creator — создатель.

3. *Конструктор* — может редактировать только дерево и свойства разделов, подключать уже существующие шаблоны и пакеты, а также создавать новые. Фактически Конструктор может полностью редактировать структуру сайта, однако работа с информационной частью ему недоступна. Такую роль в системе можно при необходимости выделять разработчику, перед которым поставлена задача изменения структуры сайта.
4. *Редактор* — может просматривать и редактировать содержание информационных пакетов в зависимости от прав доступа к ним. Эту роль можно выделять веб-мастерам, которые отвечают за публикацию информации в тех или иных информационных пакетах.
5. *Обычный пользователь* — может просматривать различные разделы сайта в зависимости от прав доступа к ним.
6. *Гость, или незарегистрированный*, — может лишь просматривать содержание открытых для всех разделов.

Права на доступ пользователей четвертой и пятой категории к разделам и пакетам сайта редактируются Рутот. Именно эти пользователи составляют основное множество людей, работающих с сайтом (исключая гостей, права которых минимальны, но которые при желании самостоятельно могут стать пользователями пятой категории, зарегистрировавшись на сайте). Рут может разрешать/запрещать просмотр/редактирование пользователями информации на сайте, а также менять статус пользователя в системе. Однако Рут не может разрешить пользователю пятой категории редактировать сайт. Для этого сначала необходимо присвоить ему статус редактора (4-я категория). Работа с пользователями сайта также не требует знания дополнительных технологий и может быть освоена обычным «офисным» работником.

Каждому *редактору* назначаются права на работу с пакетами и разделами. Редактор может иметь право на просмотр некоторого раздела и на редактирование конкретного пакета в конкретном разделе, например, ленты новостей только на главной странице. Необходимость такого определения прав пользователя обусловлена тем, что содержание одних и тех же пакетов в разных разделах может отличаться. Например, информация, содержащаяся в информационном пакете типа «обычный текст»⁴, — своя для каждого раздела.

⁴ Пакет этого типа соответствует приложению под названием «Менеджер Текстов», о котором речь пойдет ниже.

Каждому *обычному пользователю* можно разрешить/запретить просматривать тот или иной раздел. При этом, если просмотр данного раздела запрещен, то запрещен просмотр всех его потомков. Если же просмотр раздела разрешен, то права на просмотр дочерних разделов определяются отдельно.

Права пользователей остальных категорий определяются самими категориями.

3.4. Множество информационных пакетов

Множество информационных пакетов представляет собой систему обработки и хранения всей информации, отображаемой на сайте. Информация хранится в виде информационных пакетов. Каждый такой пакет представляет собой объект, свойства которого определяются приложением, за него отвечающим. Это может быть некоторый набор текстов, лента новостей, меню разделов, некоторая база данных и т.п. Каждый такой пакет зависит лишь от своего приложения, которое предоставляет пользователю полный набор средств для его просмотра и редактирования. Чтобы отобразить пакет на сайте, его необходимо привязать к некоторому разделу.

Любой пакет — это интерфейс взаимодействия ядра системы и приложения, обеспечивающего некоторую функциональность. Через него приложение предоставляет ядру доступ к объектам пакета, их свойствам и методам их обработки.

Редактирование содержания пакетов — наиболее часто выполняемая на сайте операция. Именно с помощью нее на сайте происходят публикация и изменение документов и прочего контента, как статического, так и динамического. Также на сайте доступна настройка некоторых свойств пакетов, например, количество новостей, отображаемых на одной странице для пакета-ленты новостей. Работа с пакетами в системе напоминает работу с файлами. Их можно удобно рассортировать по виртуальным папкам, что повысит удобство их использования.

3.5. Множество шаблонов отображения

Множество шаблонов отображения представляет собой набор шаблонов, использующихся для описаний структуры, вида и дизайна конечного документа, отсылаемого пользователю. Каждый такой шаблон представляет собой заранее подготовленный XSLT-документ, преобразующий внутренний XML-документ, соответствующий некоторому разделу, во внешний

документ некоторого формата (XHTML или другой формат, созданный на основе технологии XML).

При отображении на сайте именно этот документ отсылается пользователю. Ниже будет описан механизм генерации системой внутреннего XML-документа и его преобразования во внешний.

Из сказанного выше следует, что редактирование шаблонов требует знания технологий XML/XSLT. Поэтому подразумевается, что работа с шаблонами осуществляется разработчиком сайта при его установке. Множество шаблонов достаточно создать один раз вначале использования сайта. Дальнейшее же изменение сайта, в том числе изменение структуры и масштабирование функционала, не требует редактирования шаблонов. В последующем может возникнуть потребность в изменении шаблонов лишь при редизайне сайта, добавлении новых приложений и интеграции нового формата данных, основанного на XML, для последующего отображения документов на сайте в этом формате. Однако все эти задачи нетривиальны и решаются только командой разработчиков, и потому требование знания технологий XML/XSLT вполне оправданно.

Работа с шаблонами также напоминает работу с файлами. Для большего удобства также присутствует работа с директориями.

3.6. Набор приложений

Набор приложений — это набор классов, отвечающих за логическую структуру и содержание информационных пакетов, его редактирование и отображение. Таким образом, эти классы отвечают фактически за все содержимое сайта. Подключение приложений к системе — действие нетривиальное, и потому осуществляется только разработчиком сайта. После того как приложение было подключено, необходимо во все шаблоны занести информацию о том, как будут отображаться объекты и методы, создаваемые данным приложением. Рассмотрим работу приложений на примере некоторых из них.

Менеджер новостей

Это приложение, позволяющее создавать неограниченное количество лент новостей на сайте. Каждая лента — это информационный пакет, который можно привязать к одному или нескольким разделам. Лента новостей имеет одинаковое содержание во всех разделах, к которым она подключена.

Каждая новость имеет дату публикации, заголовок, фабулу (иначе предисловие, тезис или краткое содержание) и содержание. Все эти данные отображаются на сайте в зависимости шаблона отображения.

Пакет предоставляет следующие методы для пользователей.

1. Просмотр списка последних n новостей (где n задается в настройках пакета).
2. Поскольку все новости пакета разбиты на страницы по n новостей в каждой, то возможен просмотр какой-то из страниц.
3. Подробный просмотр конкретной новости с возможностью возвращения к списку новостей.

Кроме того, для редакторов существуют такие методы, как:

- создание новой новости,
- редактирование существующей новости,
- удаление существующей новости.

Менеджер голосований

Это приложение, позволяющее создавать неограниченное количество лент голосований на сайте. Каждая лента — это информационный пакет, который можно привязать к одному или нескольким разделам. Каждая лента содержит максимум одно активное голосование и неограниченное количество завершенных. Каждый пользователь имеет право проголосовать один раз. Ленты во всех разделах имеют одинаковое содержание.

Каждый объект голосования имеет дату публикации, заголовок, описание и варианты ответа.

Пакет предоставляет следующие методы для пользователей.

1. Просмотр последнего активного голосования.
2. Голосование с выбором одного из вариантов ответа.
3. Просмотр архива голосований.
4. Просмотр результатов активного голосования без выбора вариантов ответа.

Редакторам также предоставляются:

- добавление/удаление голосования,
- блокировка голосования — перевод из статуса активного в статус завершеного.

Менеджер меню

Это приложение позволяет создавать различные меню навигации по сайту. Меню может быть одного из следующих типов:

1. *Меню разделов первого уровня* — отображает ссылки на все верхние разделы.
2. *Меню «братьев»* — отображает ссылки на всех детей раздела, принадлежащего родительским для текущего раздела.
3. *Меню «родителей»* — отображает ссылки на братьев родительского раздела и на сам родительский раздел.
4. *Меню «детей»* — отображает ссылки на детей данного раздела.
5. *Цепочка навигации* — отображает полный путь до раздела, начиная с раздела первого уровня.

Содержание меню полностью зависит от раздела, в котором оно отображается и от типа меню. Оно генерируется автоматически по запросу на основе дерева разделов и не может быть отредактировано пользователем.

Менеджер текстов

Это приложение отвечает за вывод простых текстов. Его пакеты не имеют никаких дополнительных свойств. Однако через текстовые информационные пакеты можно очень удобно вставлять тексты в разделы. Каждый «текстовый» информационный пакет сопоставляет каждому разделу из тех, к которым он подключен, отдельный текстовый документ. Таким образом, в разных разделах этот пакет имеет разное содержание.

Менеджер семинаров

Это приложение, разработанное для администрирования базы данных семинаров. Эта база данных содержит в себе информацию о заседаниях семинаров. Семинар представляет собой информационный пакет. Объектами являются заседания семинаров. Каждое заседание обладает следующими свойствами: номер заседания, дата и время проведения, тема заседания, докладчик, содержание/описание.

Пакет предоставляет пользователям следующие методы.

1. Просмотр списка заседаний.
2. Подробный просмотр выбранного заседания.
3. Сортировка заседаний по номерам, датам, докладчикам.

Редакторам, в свою очередь, предоставляются методы добавления, удаления и редактирования свойств заседания.

3.7. Структура классов-приложений и их взаимодействие с ядром системы

Любое приложение разбивается на две независимые части — внутренний редактор и внешнее отображение содержимого пакетов. Они представлены в двух разных классах, подключенных соответственно к интерфейсу администрирования и программе, отображающей сайт. Редактор взаимодействует с интерфейсом администрирования, который, в свою очередь, обращается к соответствующему приложению при необходимости отредактировать свойства и содержание объектов пакета. Редактор хранит всю информацию о содержании и свойствах объектов в базе данных, содержащей одну или несколько таблиц, структура которых целиком определяется свойствами приложения, как было показано выше на примере реализованных приложений. Программа вывода соответствующего приложения использует эту же базу данных для вывода содержания пакетов. Её запускает основная программа отображения, чтобы затем вставить возвращаемый приложением код в генерируемый внутренний XML-документ.

В случае необходимости приложению через интерфейс пакета передается XML-документ, содержащий некоторые параметры. Приложение распоряжается этим документом по своему усмотрению. Подробнее об это речь пойдет ниже.

3.8. Встроенный текстовый редактор

Для расширения возможностей редактирования в ядро встроен текстовый редактор, который может быть использован приложениями при редактировании текстового содержания информационных пакетов для получения форматированного текста. В случае необходимости его использования, приложение обращается к специальному модулю, содержащемуся в ядре, и тот возвращает DHTML-код, содержащий все элементы управления и рабочие функции, написанные на JavaScript.

Помимо работы с текстом, его копирования, вырезки и вставки, этот редактор также позволяет моделировать таблицы, редактировать их основные свойства, после чего вставлять их в документ и в последующем заполнять их ячейки. Кроме того, он позволяет вставлять картинки из базы картинок, менять их размер и свойства, вставлять некоторые специальные символы, недоступные с клавиатуры, например € — евро. Также он умеет создавать нумерованные и ненумерованные списки, менять цвет и стиль написания текста, его шрифт и размер. Еще он умеет вставлять ссылки. В случае необходимости он также позволяет редактировать получившийся HTML-код.

3.9. Механизм генерации готовой страницы раздела

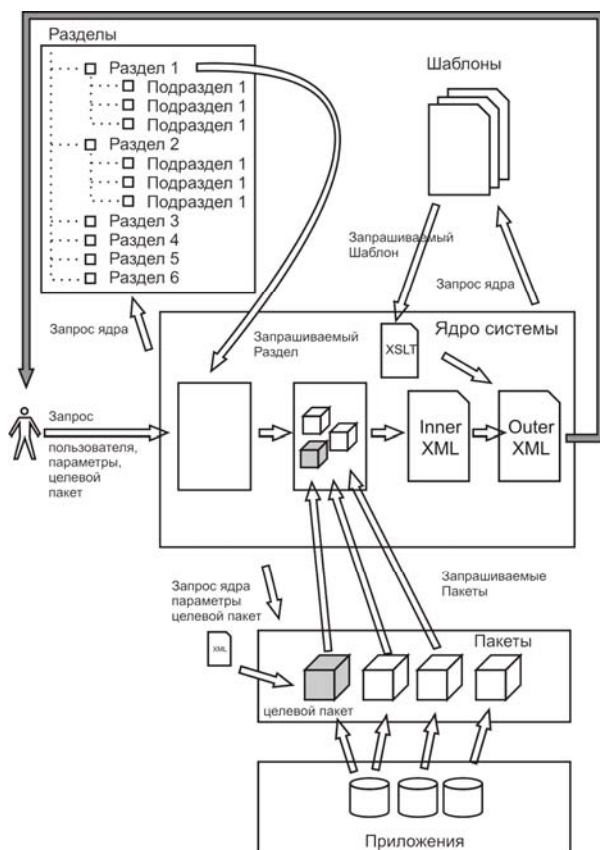


Рис 3. Механизм создания страницы сайта по запросу пользователя

Схема механизма генерации готовой страницы раздела представлена на рис. 3. Когда ядро системы получает запрос от пользователя на отображение некоторого раздела, она запрашивает свойства раздела из базы данных. Раздел обладает следующими свойствами:

- название,
- положение раздела в дереве разделов,
- шаблон отображения,

- привязанные пакеты.

После этого ядро запрашивает информацию о пакетах и соответственно узнает их названия и соответствующие приложения, отвечающие за их работу. После этого ядро обращается к этому приложению с запросом на генерацию содержимого пакета. Для каждого пакета, соответствующему приложению, передается несколько общих для всех пакетов параметров, таких как id раздела, id пакета. Также один из пакетов при необходимости помечается как *целевой пакет*. Для него ядро может передать приложению ряд индивидуальных параметров в виде XML-документа. Эти параметры индивидуальны для каждого приложения. Например, **Менеджеру Новостей** может быть передана команда на подробное отображение новости вместо общего списка новостей и соответственно id этой новости. Все остальные пакеты никаких дополнительных параметров не получают и отображаются, исходя из настроек по умолчанию.

В ответ ядро получает от приложения XML-документ, содержащий некоторый набор объектов данного пакета и набор действий, которые можно совершать над этими объектами и самим пакетом. Эти действия содержат те самые индивидуальные параметры, которые впоследствии передаются целевому пакету в случае, если это действие выполняется.

```
<package id="1" title="Новосту" package_index="1" package_type="news">
  <item type="new" id="29" mode="brief">
    <str name="title">Заголовок новосту 2</str>
    <datetime name="date">14.05.2005 13:45:00</datetime>
    <text name="fabula">
      Фабула 2, то есть краткое описание основной мысли текста.
    </text>
    <text name="content">
      Основной текст 2 идет здесь. Он может быть очень большим и
      длинным.
    </text>
    <package_action caption="Подробнее" value="view_item"
      partition_id="1" package_id="1">
      <parameters><![CDATA[
        <item_id>29</item_id>
      ]]>
    </parameters>
  </package_action>
</item>
<item type="new" id="28" mode="brief">
  <str name="title">Заголовок новосту</str>
  <datetime name="date">14.05.2005 13:40:00</datetime>
  <text name="fabula">
```

```

        Фабула, то есть краткое описание основной мысли текста.
    </text>
    <text name="content">
        Основной текст идет здесь. Он может быть очень большим
        и длинным.
    </text>
    <package_action caption="Подробнее" value="view_item"
    partition_id="1" package_id="1">
        <parameters><![CDATA[
            <item_id>28</item_id>
        ]]></parameters>
    </package_action>
</item>

<package_action caption="1" value="view_page" partition_id="1"
package_id="1">
    <parameters><![CDATA[
        <page_number>1</page_number>
    ]]></parameters>
</package_action>

<package_action caption="2" value="view_page" partition_id="1"
package_id="1">
    <parameters><![CDATA[
        <page_number>2</page_number>
    ]]></parameters>
</package_action>
</package>

```

Получив все документы от соответствующих приложений, ядро образует из них XML-документ, соответствующий вызываемому разделу.

```

<document title="Главная" id="1">
    <package ...>
        ...
    </package>
    <package ...>
        ...
    </package>
</document>

```

Перед отправкой пользователю этот документ претерпевает трансформацию, согласно XSLT-шаблона, привязанного к разделу. В итоге получа-

ется новый документ в некотором формате на основе XML (XHTML, SVG и т.п.), который отсылается обратно пользователю.

3.10. Функционалы, входящие в ядро системы

1. **Менеджер разделов** — специальная программа, входящая в состав ядра, отвечает за добавление, редактирование, удаление разделов, позволяет также устанавливать связи с шаблонами и пакетами через ячейки.
2. **Менеджер пользователей** — программа, входящая в состав ядра системы. Отвечает за редактирование базы пользователей, позволяет менять их свойства, добавлять, удалять, а также назначать права доступа к просмотру и редактированию разделов и пакетов.
3. **Менеджер пакетов** — отвечает за информационные пакеты, их создание, удаление и свойства. С его помощью осуществляется редактирование свойств пакетов через взаимодействие с соответствующими приложениями.
4. **Менеджер содержания** — отвечает за работу с информацией, содержащейся на сайте, осуществляет доступ к содержимому подключенных информационных пакетов через соответствующие разделы. Для редактирования содержимого пакетов он передает управление программе-приложению.
5. **Менеджер файлов и картинок** — отвечает за загрузку файлов и картинок на сервер для последующего их использования. Закачанные файлы можно выкладывать для скачивания пользователями. Картинки можно вставлять в документы через HTML-редактор, а также использовать в других областях, определяемых информационными пакетами.
6. **Менеджер шаблонов отображения** — отвечает за работу с шаблонами, их добавление, редактирование и удаление.

ЗАКЛЮЧЕНИЕ

В результате проведенной работы были изучены несколько существующих систем и методы их работы, а также была разработана CMS, удовлетворяющая, поставленной задаче.

- Система универсальна и позволяет создавать сайты с документами различных форматов, отличных от традиционного HTML, любым дизайном, любой структурой и любым информационным наполнением.

- Пользователи системы имеют различный статус и доступ к редактированию настроек, структуры и содержания сайта.
- В силу древовидного иерархического строения структура сайта прозрачна и легко изменяема, изменения не влияют ни на дизайн, ни на содержание сайта;
- Дизайн сайта легко изменяем и настраиваем и не влияет, в свою очередь, на структуру и информационное наполнение сайта.
- Благодаря разработанному интерфейсу взаимодействия между программами-приложениями и ядром системы через информационные пакеты, существует возможность модульного подключения дополнительных объектов/функционала в виде классов-приложений.
- Возможность хранения различных нетривиальных видов информации осуществляется за счет подключаемых приложений и встроенного текстового редактора.
- Для редактирования информации на сайте пользователю необходимо иметь лишь базовые знания о работе с офисными приложениями. Для разработки сайта необходимо также знание технологий XML, XSLT.

Среди преимуществ системы перед другими, описанными ранее, можно выделить следующие.

- Если в ранее описанных системах достигалось лишь отделение данных от их отображения, то во вновь построенной — достигнуто разделение структуры, содержания и отображения сайта. Это позволяет более гибко работать с каждым из этих трех аспектов, не опасаясь нарушить остальные. Например, можно изменять дерево разделов, независимо от содержания и дизайна, или менять дизайн, независимо от содержания и структуры.
- Структура сайта стала более прозрачной за счет использования явного дерева разделов.
- Отображение данных возможно не только в формате традиционного HTML, но и в любом другом формате, основанном на технологии XML, что делает возможным использование сайта не только людьми, но и различными программами.

На базе разработанной системы создается веб-сайт лаборатории конструирования и оптимизации программ ИСИ. С его помощью система была проверена на работоспособность и соответствие условиям, описанным в постановке задачи. С целью практического обоснования возможности подключения дополнительного нетипичного функционала был написан специализированный модуль для работы с базой данных семинаров.

В дальнейшем планируется исследование проблем реализации поиска, сбора статистики и алгоритмов кэширования страниц сайта для ускорения его работы, а также организации документооборота с участием пользователей внутри системы.

СПИСОК ЛИТЕРАТУРЫ

1. **Заостровцев Н. В.** Выбираем систему управления контентом для небольшого предприятия // Elashkin Research 2004. — 29с. — <http://business-site.ru/articles/cmsnp.pdf>
2. **Косяков И.** Стратегия выбора системы управления сайтом: сравнение систем по формальным параметрам // Информационный проект Business-Site.ru. — http://business-site.ru/articles/wsms_strat.htm
3. **Терехов А.** Сравнение Контент-Менеджеров // Информационный проект cmslist.ru. — <http://cmslist.ru/?ext=content&lang=1&pid=170>
4. Bitrix-управление сайтом: руководство пользователя / М: Битрикс, 2004 — 108с. — http://www.bitrixsoft.ru/download/BSM_Expert_UserGuide.pdf
5. Bitrix-управление сайтом: руководство по интеграции / М: Битрикс, 2004 — 40с. — http://www.bitrixsoft.ru/download/BSM_IntegrationGuide.pdf
6. Saitistika: руководство пользователя / М: Individ Company, 2003 — 155с. — <http://www.saitistika.ru/files/articles/2002/SaitistikaUserGuide.pdf>
7. Q-Publishing: руководство пользователя / М: Quantum Art, 2002 — 83с. — <http://www.quantumart.ru/doc/Q-Publishing%20User%20Guide%20Russian%20Revised.pdf>
8. Q-Publishing: руководство разработчика / М: Quantum Art, 2003 — 92с. — http://www.quantumart.ru/doc/site_developer_tutorial_rus.pdf