**Статичные и динамичные сайты**

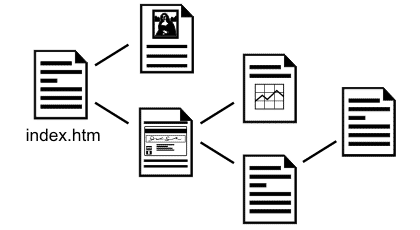
Статичный сайт

В основе статичного сайта лежат обычные HTML-страницы. Его основные достоинства — предельная простота и достаточно высокая скорость загрузки. Главный недостаток — невозможность полноценного использования многих сетевых технологий.

Работу статичного сайта можно упрощённо описать так:

* Клиент набирает адрес(URL) веб-страницы в браузере, и кликая отсылает запрос серверу.
* Получив команду, сервер проверяет наличие данного документа, и если он присутствует — направляет файл браузеру.
* Полученные данные браузер представляет на монитор пользователя, согласуя со своим внутренним алгоритмом.

Всё происходит достаточно быстро, т. к. на сервере находятся уже готовые веб-страницы.

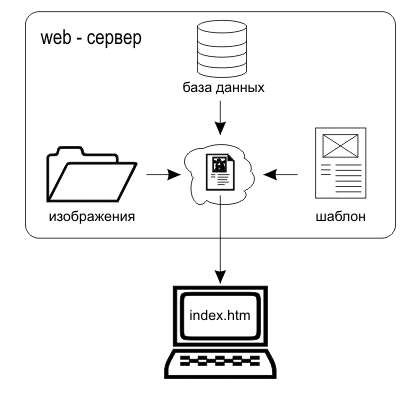
  
Связанные html-страницы. Каждая страница — отдельный файл.

Динамичный сайт

По мере разрастания простого HTML-сайта усложняется и замедляется работа по его обслуживанию. А устройство на статичном сайте голосования, оставление комментариев другого «интерактива» практически слабо осуществимо.

Появляется острая необходимость перехода на динамическое управление содержимым сайта.

Динамичный сайт работает на стороне пользователя (и его браузера) идентично статичному сайту. Отличия начинаются в работе с сервером, где динамичный сайт (сильно упрощая) работает примерно так:

* Приняв запрос, CGI-программа сервера обращается к базе данных за необходимым документом.
* Получив и проанализировав запрашиваемую информацию (если таковая имеется), программа формирует HTML код
* документа согласно установленному шаблону.
* Затем готовая страничка отправляется браузеру посетителя, который выводит её на монитор.
*   
  Формирование страницы «на лету».

Создание динамичных сайтов довольно сложная работа и требует знаний по программированию. В связи с этим, большой популярностью (да и это просто очень удобно) для создания динамичных сайтов пользуются CMS (системы по управлению содержимым сайтов). По своей сути CMS (их ещё называют движками) являются уже готовыми сайтами, но без наполнения содержимым.

**Технологическая основа CMS**

Для того, чтобы детально рассматривать отдельные возможности CMS для создания и поддержки сайтов коммерческих организаций и различия между ними, следует дать определение CMS и ее функциональности.

Система управления контентом – это программное обеспечение, которое позволяет публиковать и изменять опубликованную на сайте информацию самостоятельно, без привлечения разработчиков сайта. При этом 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 изучение новых технологий.

Большинство CMS можно разделить на back-office, т.е. инфраструктурную систему, обеспечивающую функциональность и хранение информации, и front-office, интерфейс с пользователем. В большинстве современных CMS back-office базируется на той или иной СУБД, может включать сервера приложений и портальное решение, а front-office 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 back-office, а пользовательские свойства в front-office.

Благодаря возможности конструировать внешний вид страниц из заранее заготовленных дизайн-шаблонов, пользователи CMS (систем управления контентом) теперь меньше зависят от разработчиков их сайтов. Администратор CMS может расположить различные информационные блоки в рамках заданной страницы с помощью мыши, задать их размеры, цвет и прочие атрибуты и сохранить состояние страницы, чтобы пользователи сайта видели ее в заданном виде.

**Структура сайта, основанного на CMS**

В системе CMS присутствует 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 контента, определяют внешний вид сайта, настраивают шаблоны представления информации. Этот инструментарий обычно не полностью автоматизирован. Для настройки сайта разработчики частично работают через средства CMS, часть информации размещается напрямую.

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.

В настоящее время большинство организаций имеет в том или ином виде собственный веб сайт. Большинство современных сайтов компаний созданы на основе статических страниц. В результате процесс размещения информации на таких сайтах скрыто от обычных пользователей. Вместе с тем, эта операция требует знания основ программирования и языка HTML. При этом вероятность ошибок, особенно для обычных пользователей весьма высока.

CMS предоставляет возможность оперативного обновления информации сотрудником – информацию публикует сотрудник, владеющий информацией, без дополнительных посредников в виде технических специалистов. Результатом является снижение стоимости поддержки – обновление информации производится самостоятельно, нет необходимости оплачивать труд собственного или внешнего web-мастера.

Так же в рамках CMS функционально реализованы многие сервисы – поиск, форумы, голосования и т.д., т.е. наиболее востребованная функциональность уже реализована в CMS и может быть сразу использована. При создании сайта на основе CMS полностью или частично используются готовые модули, которые уже прошли неоднократное тестирование. CMS разделяют данные и их представление, следовательно внешний вид сайта меняется с намного меньшими затратами, чем в случае со статическим сайтом.

Когда разработчики CMS заявляют о том, что после создания сайта в дальнейшем обновление информации может осуществлять нетехническими специалистами, это означает, что имеется определенный web-интерфейс, который позволяет размещать информацию, используя определенные визуальные инструменты редактирования, а также импортировать информацию из офисных документов. Под Web-интерфейсом понимается специализированное рабочее место, доступное через интернет, с использованием стандартной программы просмотра Интернет – Internet Explorer.

Ключевыми моментами является то, что возможно визуальное редактирование текста – есть WYSIWYG (What You See Is What You Get) – редактор, позволяющий размещать текст и выполнять простейшее форматирование документа, без наличия специальных технических знаний.

Возможно одновременно с текстом размещать и различные изображения – графики, диаграммы и т.д.

Возможен перенос данных из офисных приложений.

Поддержка визуального редактирования данных во всех системах реализована практически одинаково.

В состав Microsoft Internet Explorer входит элемент управления, который позволяет выполнять подобное редактирование. Разработчики осуществляют вызов этого элемента управления и далее пользуются результатами его труда.

Поскольку используются встроенные возможности Microsoft Internet Explorer (MSIE), то некоторые системы управления накладывают ограничение – для публикации информации желательно использовать Internet Explorer. При этом, поскольку в клиентской части визуальное редактирование не требуется, на него никаких ограничений не накладывается, и пользователи сайта могут использовать любую программу просмотра Интернет-страниц.

Достаточно часто возникает необходимость, одновременно с текстом разместить и изображения – картинку, график, диаграмму. Причем разместить не в фиксированном месте страницы – в начале или в конце, а непосредственно в тексте, рядом с определенным описанием.

Описанный в предыдущем разделе редактор не владеет информацией о том, где и каким образом хранятся изображения. Поэтому он позволяет размещать в произвольном месте текста ссылку на изображение.

При этом необходимо указать адрес на сервере (URL), где находится изображение. Как изображение попадет на сервер – находится вне компетенции данного редактора.

Разработчик может модифицировать данный функционал, позволив пользователю изначально разместить изображения либо в общую библиотеку, либо привязать их к конкретному документу, а далее уже при редактировании выбирать, какое изображение включить.

В случае использования общей библиотеки возможно повторное использование одних и тех же изображений, однако, при большом их количестве затруднен поиск, а также удаление ненужных. В случае связывания изображений с конкретным документом, повторного использования нет (для системы одно и то же изображение, присоединенное к двум документам, будет двумя разными), но это упрощает выбор и гарантирует автоматическое удаление изображений одновременно с самим документом.

Системы управления контентом позволяют нетехническому специалисту публиковать информацию на сайте и выполнять простейшие операции по форматированию текста. Выделить текст курсивом на сайте ничем не сложнее, чем выполнить аналогичную операцию в Microsoft Word. И данная возможность во всех системах реализована практически одинаково.

**Разделение данных и их представления**

Прежде всего, договоримся об используемой терминологии в применении к CMS:

* Документ (статья) – единица публикуемой информации. Документом может быть одного из типов документов, используемых на сайте – новость, статья, товар и т.д.
* Атрибут – один из признаков документа. Каждый документ обладает набором признаков – атрибутов. Например, для новости атрибутами являются название, автор, аннотация, текст, дата создания.
* Тип документа – обобщенное понятие документов, обладающих одинаковыми характеристиками. Например, «новость». Подразумевается, что ранее было описано, какими атрибутами обладают документы типа «новость» и далее создаем документы этого типа.
* Структурированные данные – представление данных, при котором каждый атрибут документа представлен отдельно и с ним могут выполняться независимые операции.

Как было отмечено ранее, одним из преимуществ использования CMS является структурирование информации и разделение содержимого и его представления (данных и дизайна):

* Разделение содержимого и его представления предоставляет гарантию сохранения внешнего вида – при разделении информации и ее представления оператор вводит значение каждого атрибута в отдельное поле и может быть уверен, что оно отобразиться в нужном месте, как определено в настройках отображения. Например, мы хотим, чтобы имя автора статьи всегда отображалось под названием, было выровнено вправо и выделено жирным шрифтом. Если оператор вводит только значение, то он уверен, что данные будут отображены где нужно и как нужно. Иначе он обязан помнить о принятом стиле и вручную выполнять данное форматирование.
* Разделение содержимого и его представления предоставляет возможность применения дополнительного сервиса – когда атрибуты хранятся отдельно, с ними можно выполнять дополнительные операции. Например, если у нас автор статьи хранится как отдельное поле, то очень легко просмотреть все статьи данного автора. Если имя автора хранится в тексте статьи, то формирование списка статей автора является отдельной ручной операцией.
* Разделение содержимого и его представления предоставляет возможность интеграции с внешними системами – быстрый экспорт новостей или товаров для обмена с другими ресурсами – сайтами, внутренними системами и т.д.
* Разделение содержимого и его представления позволяет снизить стоимость смены дизайна – для изменения внешнего вида сайта нет необходимости в ручной переработке каждого документа. Изменяются только шаблоны отображения и вся информация может быть быстро представлена в другом виде.
* Разделение содержимого и его представления предоставляет возможность использования одной информации в разных дизайнах – это особенно используется последнее время при создании информационных систем холдингов, когда вся информация хранится в центральной системе, но может быть показана, как на сайте холдинга, так и на сайте отдельного предприятия, но в разных дизайнах.

Исходя из вышесказанного сформулируем основные задачи, которые решает CMS (система управления содержимым сайта):

* Публикация информации нетехническим специалистом.
* Разделение данных и их представления.
* Организация совместной работы при публикации информации.
* Поисковые возможности.
* Дополнительные сервисы – форумы, голосования, анкеты и т.д.

**Модели представления данных CMS**

Объектная модель представления данных – оперирует такими понятиями, как класс и объект. Классы определяют структуру данных и представляют собой набор атрибутов (текстовая строка, целое число, изображение и т.д.). Представители класса (объекты) имеют определенную структуру и могут содержать другие объекты, образуя произвольную иерархическую структуру. Объекты могут наследовать свойства, содержание и поведение объектов, которые в них содержатся. Примерами объектов служат документы, картинки, папки и учетные записи пользователей. Класс контента не хранит в себе реальных данных – такую информацию содержат объекты (экземпляры класса). Определив один класс, можно создать множество его представителей (контент объектов).

В CMS-системах данные обычно хранятся в реляционной базе данных. В этом случае объектная модель данных отображается на реляционную модель базы данных. Связи между объектами создаются, например, с помощью таблиц вида id, from\_object, from\_object\_version, to\_object.

Как правило, системы, основанные на объектно-ориентированной модели данных, наиболее функциональные, гибкие, но, в то же время, и наиболее сложные.

Сетевая модель представления данных в CMS – опирается на теорию графов: структура информации представляется в виде узлов с помеченными связями между ними. Фундаментом системы может служить как сетевая, так и традиционная реляционная СУБД, на которую отображена сетевая модель описания данных. В реляционных таблицах хранится информация об узлах, их атрибутах и связях между ними. Связь отличается от атрибута тем, что в ней хранится ссылка на другой узел, а в атрибуте – собственно значение. Для извлечения данных из направленного графа обычно используются рекурсивные процедуры обработки, такие как составление списков узлов, определение атрибутов узла по атрибутам родителя и др. Пример системы такого типа – Communiware. Эта система представляет собой инструментарий нового поколения для создания вертикальных Web-сайтов и порталов.

Модульная модель представления данных в CMS – в подобных системах контент разделен на отдельные модули по типам содержимого. Структура данных зависит от модуля, и вся работа с контентом сосредоточена внутри модуля. Модули независимы и полностью отвечают за работу с документами данного типа. Документы описываются с помощью фиксированного набора характеристик – типы документов строго фиксированы. Расширять функциональность можно за счет добавления нового модуля, замены или редактирования существующего кода. Чаще всего нет никакой системы связей между документами разных модулей и между документами одного и того же модуля. Стандартный набор типов контента (модулей) таков: ссылки, статьи, файлы, новости, разделы, форум. Несмотря на очевидную ограниченность модели данных, системы на ее основе наиболее популярны благодаря своей простоте. В качестве примера можно упомянуть такие системы, как PHPNuke, Joomla! и Xoops. У модульных CMS-систем есть общий недостаток – строго фиксированная в пределах модуля структура содержимого. Однако для расширения их функциональности можно воспользоваться внешними модулями. Очевидное преимущество модульных систем – возможность получения почти полностью готового к использованию портала за короткое время.

**Системы типа Модель – Вид – Управление**

Схема работы Web – приложений накладывает свои ограничения на архитектуру систем управления сайтами. При работе в режиме “запрос-ответ” без сохранения состояния стандартные модели построения интерактивных пользовательских систем (которыми обычно являются Web – системы) не могут применяться. Однако, с некоторыми ограничениями, эти модели могут быть адаптированы. Наиболее распространенной, удобной, и признанной архитектурой интерактивных пользовательских систем является архитектура Модель – Вид – Управление (Model – View – Controller или MVC).

В данной архитектуре выделяются три основные подсистемы:

Модель данных – подсистема, представляющая уровень логических объектов, их взаимодействие, основную функциональность системы.

Вид или подсистема отображения – отвечает за отображение модели данных в удобном для пользователя виде, обычно графический пользовательский интерфейс.

Подсистема управления – обрабатывает события, получаемые от пользователя и других частей системы, инициирует действия и изменения в модели.

Архитектура широко распространена и применяется в таких популярных системах построения пользовательского интерфейса как Microsoft Foundation Classes (MFC) и Windows Forms для операционных систем Windows, Java Swing для Java платформы, X Window System и Motif для UNIX систем. Несмотря на несомненные достоинства архитектуры, такие как разделение данных и их представления, удобство в разработке и возможности изменения отдельных частей системы без изменения остальных, она является непростой для реализации в Web – системах. Особенную сложность представляет система управления, поскольку инициатором и получателем событий и команд являются различные программы, соединенные негарантированной линий связи. Тем не менее удобство данной архитектуры заставляет искать пути ее наилучшей адаптации к разработке Web приложений.

В качестве реализации подсистем модели и отображения обычно предлагается сочетание шаблонов страниц с отдельным программным кодом, при котором работа программиста и дизайнера может быть максимально разделена. При построении на шаблонной технологии, например PHP, они позволяют компоновать на одной странице различные шаблонные блоки, а также вводить их содержимое в административном интерфейсе. При этом взаимное расположение блоков, основной дизайн заложены в так называемых темах сайта и не могут быть гибко изменены, хотя и позволяют проводить самостоятельную разработку. При использовании более гибких технологий ядро системы позволяет определять шаблоны и программные компоненты, заполняющие содержимое, обеспечивает их взаимодействие. В наиболее высокоуровневых моделях строится объектная модель сайта, его страниц, участков страниц. К сожалению в подобных системах визуальное построение страниц и оперативное управление через Web – интерфейс отсутствуют, поскольку системы ориентированы на разработку на сервере. В качестве подсистемы управления присутствует объектная модель, позволяющая взаимодействовать с пользователем в режиме, напоминающем обычную интерактивную модель с событиями, обработчиками, передачей управления, и т.д.

Поиск оптимальной реализации архитектуры Модель – Вид – Управление для систем управления сайтами является актуальным и находит свое применение в различных существующих системах построения сайтов.

**Классификация CMS**

За несколько лет системы управления содержимым веб-ресурсов значительно усовершенствовались. Классифицируем CMS по областям применения:

* Сайт-визитка — содержит самые общие данные о владельце сайта (организация или индивидуальный предприниматель). Вид деятельности, история, прайс-лист, контактные данные, реквизиты, схема проезда. Специалисты размещают своё резюме. То есть подробная визитная карточка.
* Представительский сайт — так иногда называют сайт-визитку с расширенной функциональностью: подробное описание услуг, портфолио, отзывы, форма обратной связи и т. д.
* Корпоративный сайт — содержит полную информацию о компании-владельце, услугах/продукции, событиях в жизни компании. Отличается от сайта-визитки и представительского сайта полнотой представленной информации, зачастую содержит различные функциональные инструменты для работы с контентом (поиск и фильтры, календари событий, фотогалереи, корпоративные блоги, форумы). Может быть интегрирован с внутренними информационными системами компании-владельца (КИС, CRM, бухгалтерскими системами). Может содержать закрытые разделы для тех или иных групп пользователей — сотрудников, дилеров, контрагентов и пр.
* Каталог продукции — в каталоге присутствует подробное описание товаров/услуг, сертификаты, технические и потребительские данные, отзывы экспертов и т. д. На таких сайтах размещается информация о товарах/услугах, которую невозможно поместить в прайс-лист.
* Интернет-магазин — сайт с каталогом продукции, с помощью которого клиент может заказать нужные ему товары. Используются различные системы расчётов: от пересылки товаров наложенным платежом или автоматической пересылки счета по факсу до расчётов с помощью пластиковых карт.
* Промо-сайт — сайт о конкретной торговой марке или продукте, на таких сайтах размещается исчерпывающая информация о бренде, различных рекламных акциях (конкурсы, викторины, игры и т. п.).
* Тематический сайт — сайт, предоставляющий специфическую узкотематическую информацию о какой-либо теме.
* Тематический портал — это очень большой веб-ресурс, который предоставляет исчерпывающую информацию по определённой тематике. Порталы похожи на тематические сайты, но дополнительно содержат средства взаимодействия с пользователями и позволяют пользователям общаться в рамках портала (форумы, чаты) — это среда существования пользователя.
* Веб-сервис — услуга созданная для выполнения каких либо задач в рамках сети WORLD WIDE WEB.

**Существующие решения CMS**

CMS – один из наиболее конкурентных рынков приложений сегодня. В мире существует несколько тысяч или десятков тысяч подобных приложений.

Все использующиеся CMS можно разделить на три основные категории:

* Системы, созданные и развиваемые внутри студии (внутренние CMS) . Это внутренний инструмент, доступный лишь сотрудникам конкретной студии.
* Бесплатные свободно распространяемые CMS. Разрабатываются и развиваются сообществами пользователей на добровольной основе. Использование бесплатной системы не требует никаких выплат, кроме гонорара программисту, осуществляющему разработку проекта с использованием данной CMS. Наиболее известные представители этой категории: Joomla, Drupal, Wordpress, TYPO3.
* Коммерческие CMS. Каждая система имеет конкретного разработчика, который берёт на себя вопросы развития системы и технической поддержки пользователей. Пользователи коммерческих систем оплачивают приобретение лицензии на использование системы и, часто, техническую поддержку свыше определённого срока. Веб-студии, являющиеся партнёром того или иного производителя CMS, имеют скидку на приобретение, за счёт чего могут предоставлять систему своему клиенту по цене производителя. Наиболее известные коммерческие системы управления: 1С-Битрикс, NetCat, UMI.CMS, S.Builder, HostCMS.

Помимо выделенных типов все CMS можно делить по их назначению (универсальные и специализированные на конкретных типах сайтов), но, поскольку в нашем случае речь идёт о создании конкретного типа проекта (корпоративный сайт), то это деление несущественно.

Центр маркетинговых решений Андрея Коновалова» представляет первое открытое маркетинговое исследование рынка услуг по разработке веб-сайтов.

Исследование проведено в августе 2010 года. В ходе исследования был проведён анализ данных от 653 веб-студий трёх крупнейших российских городов: Москвы, Санкт-Петербурга и Екатеринбурга.

На момент проведения, данное исследование является единственным подробным открытым маркетинговым исследованием рынка услуг по разработке интернет-сайтов.

Полученные результаты максимально справедливы для Москвы, Санкт-Петербурга и Екатеринбурга. Для страны в целом картина будет несколько иная, это наглядно заметно по изменениям от Москвы к Екатеринбургу (в первую очередь – уменьшение общей доли коммерческих систем).

В качестве оцениваемого проекта выступал корпоративный сайт средней сложности. Для других типов сайтов картина может меняться – в первую очередь это будет касаться некоммерческих проектов, сайтов-визиток, интернет-магазинов, сложных коммьюнити-проектов.

Популярность CMS среди веб-студий (сводный отчёт)

