

Андрей МАЙОРОВ,

Федор СМЕРНОВ

Реанимация веб-сайтов с помощью «живых документов»

Кому-то словосочетание «живые документы», предложенное в заглавии этой статьи, может показаться чересчур смелой метафорой. Видимо, так устроены пользователи, что при работе с компьютером им постоянно хочется оперировать метафорами: складывать файлы на Рабочий стол, выбрасывать мусор в Корзину, а оставшиеся документы раскладывать по папкам. Когда я говорю в этой статье о документах, то имею в виду веб-страницы (или HTML-документы), совокупность которых и образует веб-сайт. Почему «живые»? Потому что Интернет предъявляет все более высокие требования к скорости обновления информации, а динамический контент становится практически стандартом де-факто. Однако «живут» динамические веб-страницы именно на сервере, а информация, представленная в браузере, неподвластна пользователю, мертва и нежизнеспособна до тех пор, пока мы вновь не обратимся к серверу, чтобы сохранить изменения. А как бы хотелось, чтобы веб-страницы «ожили» и на пользовательской машине! И в этом, как оказалось, нет ничего утопического.

Безвозвратно ушло то время, когда человечество довольствовалось небольшими сайтами со статическим содержанием. Что, похоже, очевидно даже энтузиастам, днями и ночами колдующим над собственными «малобюджетными» проектами. Современный сайт — это динамический контент, обеспечение работы различных групп пользователей, мониторинг посещаемости... Продолжать список требований можно до бесконечности, ясно одно — прежний способ создания и редактирования веб-страниц в Notepad или, того хуже, в Front Page потерял былую привлекательность. Что бы ни говорили о простоте языка разметки HTML, пользователи упорно отказываются верить и требуют более простых и понятных инструментов. К счастью, нелюбовь к рутинной работе отличает и разработчиков ПО. Альтернативой привычной схеме создания и редактирования веб-страниц стали системы управления контентом, также хорошо известные как CMS (Content Management System). Они позволяют даже неподготовленному пользователю быстро и без особых усилий создавать веб-страницы, редактировать навигационное меню, загружать на сайт картинки и делать многие другие фантастические вещи, одна мысль о которых раньше заставляла его вздрагивать. Как правило, современные CMS делают возможным управление доступом пользователей к различным разделам сайта, вести мониторинг посещаемости, отслеживать показы баннерной рекламы. В настоящее время количество заслуживающих внимания CMS измеряется сотнями: над их созданием и развитием трудятся и крупные компании, и гордые одиночки, ищущие самоутверждения.

В многочисленных публикациях на тему CMS авторы сетуют на то, что отечественный рынок систем управления содержанием веб-сайтов находится в стадии формирования, развивается хаотично. Однако все единодушны в том, что положительные традиции налицо [1, 3]. Традиционно выделяют две разновидности этих программных продуктов. Коробочные версии, т.е. достигшее определенной стадии зрелости ПО, могут поставляться как набор инструментов для самостоятельного создания динамического веб-сайта, чему очень способствует грамотно составленная документация и развитая служба технической поддержки. Среди таких продуктов — NetCat (<http://www.netcat.ru>), Bitrix (<http://www.bitrixsoft.ru>), Saitistika (<http://www.saitistika.ru>). Другая часть CMS поставляется вместе с веб-сайтом, разработанным на заказ, т.е. речь идет, скорее, об услугах, нежели о готовом решении. Такие услуги предоставляют многие отечественные веб-разработчики: RBC Contents (<http://www.rbcsoft.ru>), Optimizer (<http://www.optimizer.ru>), Dynasite (<http://www.reksoft.ru>), Q-Publishing (<http://quantumart.ru>). Любителям бесплатного сыра тоже есть чем поживиться (например, предлагаемые бесплатно Zope, PHP-Nuke). Западные системы (Vignette, WebSphere) пока не получили у нас серьезного распространения. В последнее время появляются публикации, посвященные профессиональному анализу различных CMS, из которых можно почерпнуть массу дополнительной информации [2].

Тема этой статьи — оригинальное решение, которое позволяет упростить процесс управления веб-сайтом. Его уникальность в трансформации привычной иерархической схемы клиент—сервер, которая используется большинством CMS, в сторону предоставления клиенту большей свободы в процессе редактирования веб-страниц. Кроме того, речь пойдет именно о технологии, которая не зажата в жесткие рамки системы и предоставляет разработчику большую свободу.

Как это работает?

Посмотрим, как работает традиционная система управления контентом.

Допустим, нам необходимо создать раздел сайта. Для этого мы переходим в режим редактирования, манипулируя графическими значками, ссылками или кнопками, выполняем эту нехитрую задачу. А что же происходит в системе?

При загрузке страницы из базы данных, расположенной на сервере, формируется контент страницы (XML). Представление данных в виде соответствующего XSL-файла хранится на сервере в готовом виде. Сервер преобразует полученную информацию в привычный для браузера HTML, после чего страница демонстрируется пользователю, который вносит изменения.

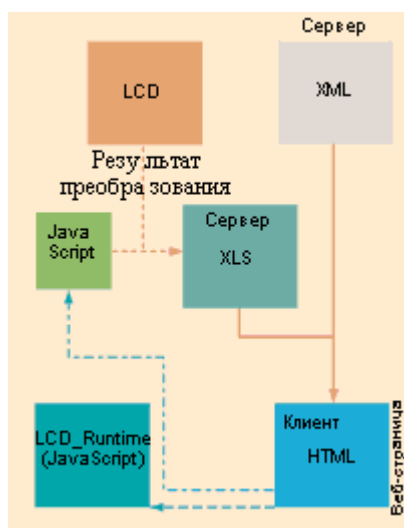


Схема генерации страницы с использованием LCD

Самый распространенный вариант дальнейшего развития событий — изменения передаются на сервер в виде имя поля = значение. Уверен, что каждому пользователю Интернета не одну сотню раз приходилось заполнять самые различные формы: будь то регистрация почтового ящика, обращение в Правительство РФ или размещение своего резюме. Если внимательно посмотреть на адресную строку браузера после того, как вы уже нажали кнопку «Отправить», можно заметить, что она заметно «подросла», в основном за счет конструкций типа: `/?first_name=Fedor&last_name=Smirnov&age=24...`

Очень часто именно в таком виде ваши данные поступают на сервер. Не требуется особых программистских талантов, чтобы написать скрипт, разбирающий подобные строчки и правильно записывающий информацию в базу данных.

Реже используется другой вариант: после сохранения изменений на сервер передается отредактированный XML-документ, который «разбирается» и записывается в базу данных. Чтобы пользователь мог увидеть результат сделанных изменений, страница перезагружается. При этом браузеру передается обновленный XML-документ и XSL-шаблон.

Таким образом, в традиционной системе управления контентом обращение к серверу происходит каждый раз, когда необходимо создать новый объект. Чтобы пользователь увидел результат своего труда, будь то новый раздел, страница или тема форума, данные должны пройти длинную цепочку всяческих преобразований.

О вреде излишеств

Я предвижу справедливый гнев пользовательских масс: «Ну и что? Почему нас должно беспокоить лишнее обращение к серверу? Самое главное, чтобы сайт можно было обновить быстро и надежно». Однако те, кому действительно приходилось пользоваться инструментами для управления информационным наполнением сайта, уже призадумались. Сервер — машина с ограниченными ресурсами: чем больше ее нагружаешь, тем медленнее она работает. Если добавить к этому, что на сайте не меньше 5000 страниц и его обслуживает несколько десятков редакторов, лишнее преобразование будет действительно серьезной проблемой.



Страница сайта с активным редактируемым элементом

Трафик — еще одна причина, по которой хотелось бы избежать дополнительного обращения к серверу. Полученный от провайдера счет может несколько отравить радость от активной работы над сайтом.

LCD: «клиент» всегда прав

Как же можно избежать лишних преобразований?

Ответ вполне очевиден. Если в системе управления контентом происходит взаимодействие между клиентом (браузером пользователя) и сервером, то имеет смысл переложить часть работы на клиента. Этот подход реализован в технологии Live Client Documents (LCD), разработанной компанией «BYTE-force» (www.byte-force.com).

Для того чтобы превратить обычную страницу сайта в интерфейс для редактирования, достаточно добавить несколько инструкций в XSL-шаблон. Нельзя сказать, чтобы эти модификации носили совсем тривиальный характер, однако человек, уже имевший дело с XSL, освоит их без труда.

Внешний вид веб-страницы после добавления инструкций не изменится, однако отдельные элементы при активации (щелчке мышью) становятся доступны для редактирования (например, можно ввести текст, добавить картинку и т.д.).

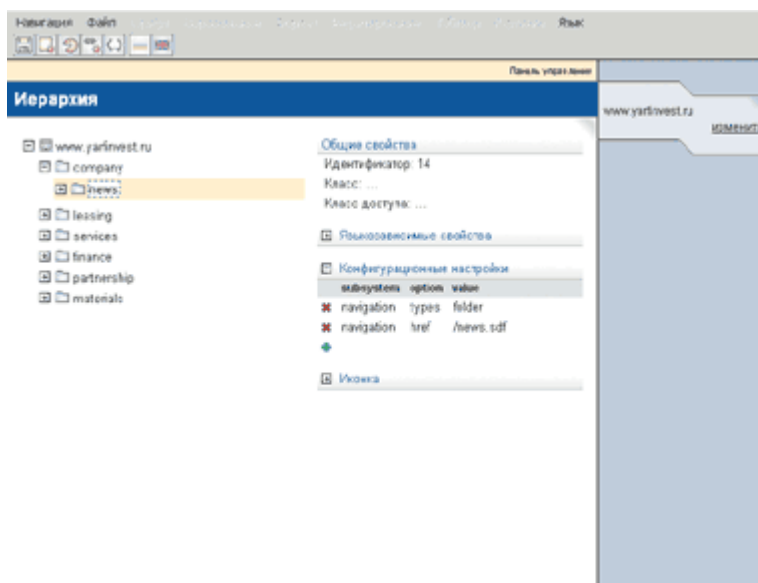
Принципиальным отличием такого редактора страниц сайта является возможность добавления новых объектов без перезагрузки. Захотелось, скажем, добавить раздел — достаточно нажать соответствующую кнопку, и новый объект появится на странице. С ним можно сразу же что-нибудь сделать, например переименовать или заполнить его свойства. К тому же не надо ждать, пока перезагрузится страница, и размышлять о возможных недостатках сервера или интернет-соединения.

Взгляд под капот

Как такое становится возможным? В распоряжении браузера постоянно находится толковый справочник — XSD-схема документа, с которой программа может свериться в случае необходимости. Кроме того, полученная веб-страница содержит ссылки на скрипты, формирующие некоторое виртуальное представление документа, которым, собственно, и манипулирует система. Сам браузер, наряду с HTML-документом, имеет XML- и XSL-составляющие. Казалось бы, мы располагаем всем необходимым для добавления нового элемента.

Чтобы пользователь увидел, скажем, новый раздел и его название, необходимо добавить еще один значок и сопроводительный текст. Заметить изменения мы сможем не раньше, чем соответствующий кусочек кода будет добавлен в HTML-файл. Нельзя забывать, что новый элемент должен быть добавлен и в XML, и в то виртуальное представление, которым манипулирует програм-

ма. Соответственно, необходим механизм, позволяющий связать все вышеперечисленные образы веб-страницы в единое целое.



Редактор структуры сайта, построенного на LCD

Среди скриптовых файлов, которые загружаются вместе с веб-страницей, есть неприметный LCD_Runtime. Он-то и обеспечивает согласованную работу. Сначала происходит инициализация объектов системы, т. е. каждый тип объектов, доступных программе, связывается с одним из корневых HTML-элементов веб-страницы. Например, объект Page связан с элементом body. В результате выстраивается многоуровневая иерархия объектов, каждый из которых связан с HTML-элементом. Так, поле для добавления текста новости связано с HTML-представлением и соответствующей секцией данных в XML.

Таким образом, когда пользователь добавляет новую страницу, изменения в структуре сайта затрагивают одновременно HTML, XML и внутреннее представление документа, которым оперирует система. При этом у LCD ни разу не возникает желания обратиться к серверу с дополнительными вопросами, перезагрузить страницу или еще как-то оторвать пользователя от работы над контентом и потратить его драгоценное время.

Безусловно, полностью избежать обращения к серверу невозможно. Когда-нибудь, вдоволь наигравшись со структурой веб-сайта, добавив десяток-другой страниц, вы вдруг решите, что было бы неплохо, чтобы результаты вашего труда стали доступны интернет-общественности. Теперь уже точно необходимо установить соединение с сервером для однократного сохранения всех выполненных изменений в базе данных. Переданный на сервер XML-документ не требует дополнительного разбора, поскольку его структура остается неизменной благодаря постоянному контролю процесса изменений со стороны программы.

В поисках несовершенства

На несовершенство нашего мира человечество жалуется уже довольно давно. Что уж говорить о совершенстве программного обеспечения, которое создается простыми смертными часто в сжатые сроки и под давлением требований несовершенного рынка?

Думается, у каждого программного продукта есть свои преимущества и недостатки, задача потребителя — суметь объективно взвесить все «за» и «против».

Аргументы в пользу LCD:

- Простота создания интерфейсов для редактирования веб-сайта.
- Возможность действительно визуального редактирования: редактируемая страница выглядит так же, как и предъявляемая посетителю сайта.
- Снижение нагрузки на сервер и уменьшение трафика.

- Единство структуры данных до и после редактирования.

Ограничения:

- Обязательное использование браузера Internet Explorer 6.0.
- Включенная поддержка JavaScript в браузере.

Ключевым для характеристики Live Client Documents является именно слово «технология». Речь не идет о коробочном продукте, который остается только установить, настроить и доверить ему свою информацию. В этом, наверное, заключается одновременно и недостаток, и важное преимущество LCD. Оригинальный подход, положенный в основу этой технологии, дополняет значительная свобода, предоставляемая разработчику в создании прикладных решений. Другими словами, «живые документы» принципиально возможны, но то, как будет использована их «жизненная сила» в рамках конкретного проекта, полностью зависит от разработчика.

Литература

Жарикова Е. Выбор системы управления контентом интернет-ресурса // Инфобизнес-Онлайн, 14.10.2002, <http://www.ibusiness.ru/marcet/tele/20905/>.

Савельева Н. Системы управления контентом // Открытые системы. 2004, №4. <http://www.osp.ru/os/2004/04/041.htm>

Терехов А. Российский рынок CMS, <http://www.webinform.ru/analyst/671.html>

Об авторах

Федор Смирнов — преподаватель ЯГПУ, технический писатель. Основная сфера интересов — развитие веб-технологий, документирование программного обеспечения, лингвистические аспекты применения информационных технологий. E-mail: sprache@inbox.ru

Андрей Майоров — исполнительный директор компании «BYTE-force». С ним можно связаться по адресу: xor@byte-force.com.