

ОБЗОР ТЕХНОЛОГИЙ СОЗДАНИЯ ИНТЕРАКТИВНЫХ WEB-УЗЛОВ

В статье содержится обзор технологий создания интерактивных Web-узлов. Рассматриваются основные элементы интерактивности, методы реализации интерактивности в Web. Дана краткая характеристика технологий реализации интерактивности (CGI, ISAPI, технологий создания интерактивных страниц: ASP, PHP, SSI), а также их сравнительный анализ.

Ключевые слова: интерактивный Web-сайт, методы реализации интерактивности, интерактивные технологии, технологии создания интерактивных страниц.

I.S. Shuvalova

SURVEY OF TECHNOLOGIES OF INTERACTIVE WEB-SITE CONSTRUCTION

This article is devoted to technologies of interactive Web-sites development. It explores the basic interactive components and describes methods of working out interactivity (CGI, ISAPI) and interactive pages (ASP, PHP, SSI). There is their comparative analysis as well as.

Keywords: interactive Web-site, methods of working out interactivity, interactive technologies, technologies of working interactive pages ASP, PHP, CGI.

Web-документы создаются не только для печати. Повсеместно идет смещение акцента с Web-узлов как набора страниц на Web-узлы как наборы программ, например Web-узел, обрабатывающий заявки на вакантные места. Он должен оперативно обновлять список вакансий, поддерживать поиск по ключевым словам, уметь принимать резюме, составлять расписание собеседований и выполнять множество других задач. Причем делать все это необходимо с учетом различий в браузерах, которыми могут пользоваться его посетители.

Теперь уже повсеместно Web-узлы делают интерактивными. Несмотря на то что акцент на программы усилился, это вовсе не значит, что язык гипертекстовой разметки HTML вскоре канет в Лету. Еще до появления клиент-серверных технологий HTML пересекался с идеями программирования. В создании серверных программ в Web всегда присутствовал незримый дух HTML, в особенности при появлении серверных вставок и условных HTML-конструкций – типа активных серверных страниц компании Microsoft (ASP – Active Server Pages) или программного пакета Cold Fusion компании Allaire. Данные технологии стирают грань между HTML и программированием, так как в них используются специальные теги разметки, которые либо выводят определенную информацию, либо выполняют определенные действия.

По своей сути Web – это форма взаимодей-

ствия клиента и сервера. Web-браузеры делают запросы к Web-серверам, которые выполняют какое-либо действие, либо возвращают файл, отправляемый обратно браузеру, который его и выводит на экран. Схема этого взаимодействия изображена на рис. 1.

По данной схеме Web-сервер выступает как обычный файловый сервер, который доставляет HTML-файлы Web-браузеру.

Самыми основными элементами интерактивности в Web, помимо выбора ссылок, являются формы, которые обрабатываются программами. Обычно это CGI-программы, которые запускаются на Web-сервере и возвращают пользователю результат своей работы. Возникает вопрос: где должны осуществляться вычисления и с помощью какой технологии? На ранних стадиях развития Web браузер практически никакими вычислениями не занимался. Его задачей было только вывести страницу на экран. Теперь с появлением клиентских технологий, например Java, ActiveX и JavaScript, многие вычисления можно делать прямо в браузере. Если бы весь этот груз вычислений перенести на сервер, пользователи стали бы жаловаться на медленную реакцию сервера на свои действия. Тем не менее, при переносе вычислений на сторону клиента могут возникнуть проблемы с совместимостью, так как очень трудно предугадать, какой программой пользуется клиент. Вопросы безопасности тоже надо учесть. Наилучшим решением является сочетание: некоторые вещи хорошо делать на стороне клиента, другие – на сервере.

¹ Кандидат технических наук, доцент НОУ ВПО «Российский новый университет».

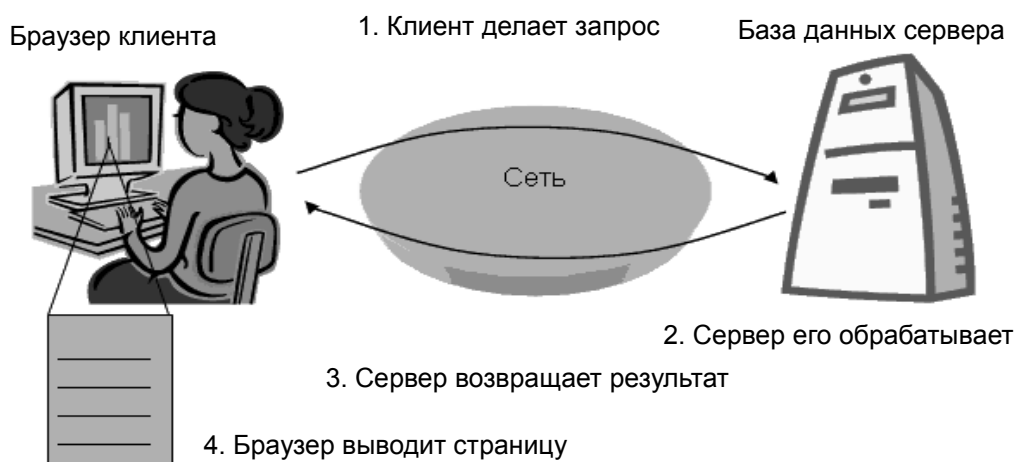


Рис. 1. Схема взаимодействия клиент – сервер

Для проверки содержимого формы перед отправкой данных CGI-программе лучше всего воспользоваться программой на JavaScript, чем осуществлять проверку в самой CGI-программе. Но на сервере все равно должна производиться проверка на тот случай, если клиент пользуется старым браузером или отключил в браузере обработку сценариев. Проверка нужна и на тот случай,

когда неправильные данные отправляются намеренно каким-либо злоумышленником. При построении программы на основе Web разработчик должен сделать выбор, где размещать программу (на стороне сервера или клиента) и какой технологией надо воспользоваться для ее написания. По таблице 1 можно судить, насколько широк выбор программных технологий в Web.

Таблица 1

Вариант программных технологий в Web

Клиент	Сервер
Вспомогательные программы	CGI-программы
Модули <i>plug-in</i>	Программы <i>ISAPI</i> (библиотеки <i>DLL</i>)
Элементы <i>ActiveX</i>	Серверные сценарии
Java-апплеты	Серверные включения (<i>SSI</i>)
Языки сценариев	Активные серверные страницы (<i>ASP</i>)
JavaScript	Серверный JavaScript (<i>LiveWire</i>)
VBScript	Обработчики баз данных
Динамический HTML	<i>Cold Fusion</i> компании <i>Allaire</i>

При добавлении интерактивности в Web-страницу часто имеет смысл перенести выполнение всех операций на сервер. Для этого существуют две причины. Первая: если браузер будет выполнять только функции вывода HTML-страниц, то гораздо проще контролировать все вопросы безопасности и правильности вычислений на сервере. Вторая причина: именно на сервере находится большая масса данных, или база данных. Например, существует Web-узел, к которому пользователь обращается с запросом, есть ли в наличии то или иное изделие. Чтобы пользователь мог оформить свой запрос, разрабатывается форма. Содержимое формы передается на сервер программе, которая разберет полученные данные

и сформирует запрос к базе данных. Такие программы называют посредниками (*middleware*), так как они стоят между сервером базы данных и Web-сервером, с которым взаимодействует клиент. CGI-программа и является таким посредником. После того как система управления базами данных выполнит запрос, результаты возвращаются обратно посреднику, который преобразовывает их в HTML-страницу и передает результат обратно в браузер. Координатором всего этого процесса выступает Web-сервер, он взаимодействует с программой-посредником и даже запускает ее. Данная идея схематически изображена на рис. 2.

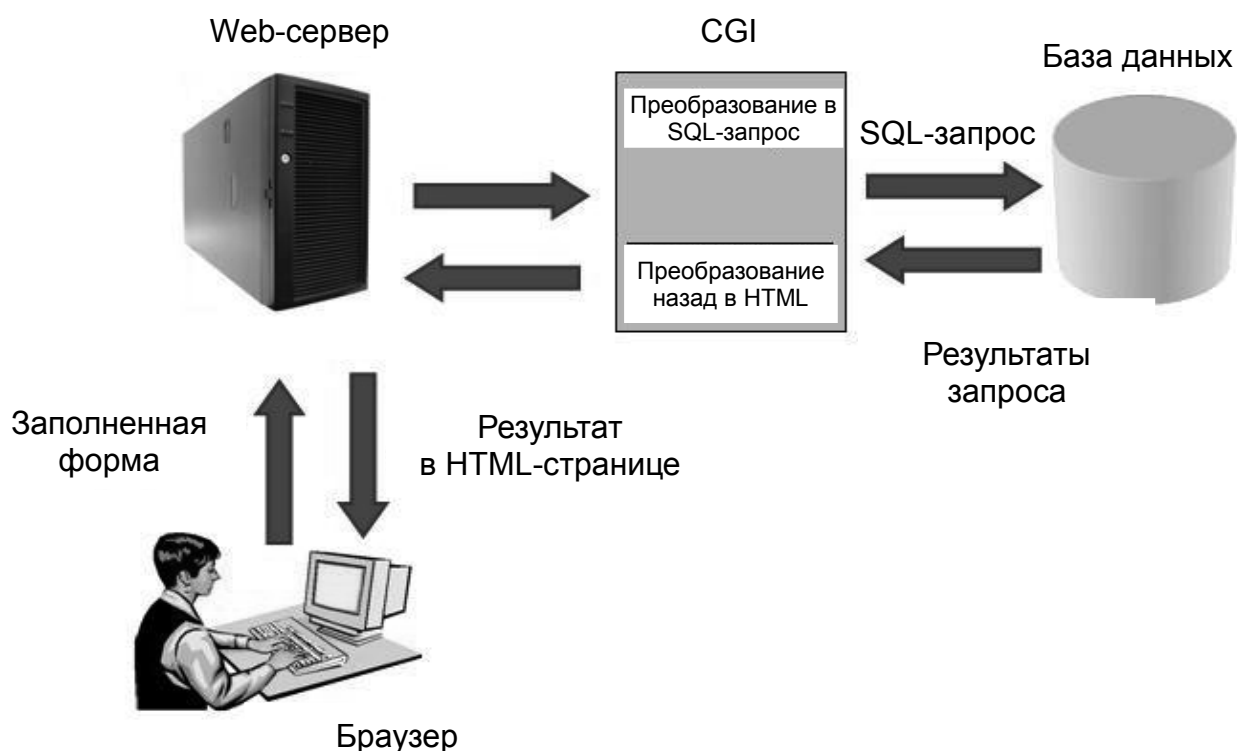


Рис. 2. Доступ к базе данных через Web

Главный недостаток переноса интерактивности на серверные программы – слишком много работы выполняется на сервере, в результате чего уменьшается его скорость и масштабируемость. Роль браузера относительно проста, так как вся вычислительная работа практически полностью перенесена на сторону сервера. Браузеру-клиенту нужно только вывести форму и страницу с результатами.

Методами реализации интерактивности являются:

- 1) единый (общий) шлюзовой интерфейс (CGI);
- 2) программный интерфейс (ISAPI);
- 3) технологии создания интерактивных страниц:
 - 3.1) технология ASP (активные серверные страницы);
 - 3.2) технология PHP (препроцессор гипертекста, или серверный мультиплатформенный язык описания сценариев, встраиваемый в HTML-текст);
 - 3.3) технология SSI (включения на стороне сервера)

Краткая характеристика данных технологий

CGI (Common Gateway Interface) переводится как *единый (общий) шлюзовой интерфейс*. Это протокол, который определяет, как информация передается из Web-страниц через Web-сервер в

программу, а затем обратно из программы в браузер в нужном формате. *CGI*-программа – это практически обычная, исполняемая программа, написанная, как правило, на языке *Perl*, по нотации сходным с *Cu*, но дополненным рядом специфических операторов. Транслятор *Perl* обычно построен по принципу интерпретатора и должен иметься на сервере, на котором планируется разместить *CGI*-программу. Переданный с локального компьютера (клиента) запрос принимается *CGI*-программой на вход, подобно вводу параметров в командной строке при ручном запуске.

ISAPI (Internet Server Application Programming Interface) – это программный интерфейс, позволяющий реализовать функции *CGI*-программы при помощи библиотек динамической загрузки *DLL*. Основное различие между *CGI* и *ISAPI* заключается в том, что когда на данный сайт обращаются одновременно несколько пользователей, пересылая на сервер свои информационные запросы, *CGI*-программа запускается на исполнение в нескольких экземплярах как параллельные вычислительные процессы; библиотека же *DLL* загружается в оперативную память лишь при самом первом обращении к ней и далее для обработки нескольких разных запросов используется только одна ее копия. На сервере подобные программные файлы, как правило, размещаются в специально отведенном каталоге *CGI-BIN*. Это единственный каталог, в котором допускается запуск программ

на исполнение (в остальных каталогах запуск исполняемых модулей обычно блокирован при помощи соответствующих настроек серверного программного обеспечения, чтобы исключить лазейки для непрошенных гостей – хакеров). Обращение к такой программе (например, в ссылке, щелчок на которой загружает страницу, сгенерированную CGI-программой) может выглядеть, например, так: *HREF* = “*http://www.domain.ru/cgi-bin/progen.exe*”

ASP, или *активные серверные страницы* (*вспомогательный процесс*), – это один из наиболее известных компромиссов между клиент-серверной и скриптовой технологиями. Основа *ASP* – скрипты, добавленные в HTML-текст страницы (такой файл обычно имеет расширение *ASP*) и выполняемые на сервере. Браузер пользователя получает модифицированный файл с «чистым» HTML, то есть сами *ASP*-скрипты пользователю недоступны, так как ему передается уже результат работы этих скриптов. В этом можно увидеть значительное сходство *ASP* с *Dynamic HTML (DHTML)*, но размещение скрипта на сервере позволяет избавиться от недостатков скриптового программирования (на стороне клиента). Сфера применения *ASP* – это прежде всего создание интерфейсов баз данных с использованием специальных *ActiveX*-компонентов – *ActiveX Data Objects* – и общепринятого стандарта формирования запросов к базам данных *SQL*.

PHP (Personal Home Page), или *язык программирования скриптов*, – это серверный, не зависящий от применяемого программно-аппаратного обеспечения язык описания сценариев, встраиваемый в HTML-текст. Название языка расшифровывается как «*препроцессор гипертекста*» (*Hypertext Preprocessor*). HTML-файл, содержащий в себе *PHP*-команды, должен носить расширение *PHP*. Команды *PHP*, заключенные в символы `<?php ?>` и размещаемые в требуемых местах HTML-кода, позволяют добавлять на Web-страницу те или иные элементы, обычно реализуемые вызовом *CGI*-программы, либо фактически превращают данный файл в шаблон, по которому строятся пересылаемые пользователям результирующие Web-страницы.

SSI (Server Side Includes), или *включения на стороне сервера*, – это тоже разновидность скриптовых команд, вставляемых в HTML-текст для передачи каких-либо указаний серверу. Выглядят они как обычные HTML-комментарии, содержимое которых начинается с символа `#`, состоит из имени команды и одного или нескольких параметров с их значениями. Всего таких команд предусмотрено около десятка; обычно они применя-

ются (аналогично *PHP*) для вставки в некоторый шаблонный Web-документ HTML-фрагментов из внешних файлов для генерации страниц в зависимости от значений переменных. HTML-файл, содержащий *SSI*-команды, имеет тип *shtml* (или *shtm*). Результирующая Web-страница формируется сервером непосредственно перед ее пересылкой на компьютер посетителя. И, как и в предыдущих случаях, сами эти команды конечному пользователю не «видны», то есть он просматривает в своем браузере страницу уже на «чистом» языке HTML, которую сервер скомпоновал в соответствии с данными ему указаниями.

Все рассмотренные технологии заметно упрощают жизнь Web-дизайнеру, особенно когда требуется реализовать сайт с динамически изменяющимся содержимым. Технологии создания интерактивных страниц гораздо проще использования *CGI*-программ и во многом сохраняют их преимущества относительно возможности накапливать на сервере поступающую от посетителей сайта информацию и делать ее доступной для других посетителей (что недоступно при использовании локально действующих скриптов *Javascript* и *VB-Script*). Если же сравнивать между собой три вышеописанные разновидности серверных скриптов, то наиболее универсальной по сфере применения и богатой по предоставляемым возможностям является технология *PHP*, тогда как *ASP* ориентировано главным образом на поддержку удаленных баз данных, а *SSI* можно рассматривать как упрощенный аналог *PHP*, более простой в освоении, но и менее богатый по возможностям.

Литература

1. Усенков, Д.Ю. Уроки Web-мастера. – М. : БИНОМ. Лаборатория знаний, 2004.
2. Веллинг, Л., Томсон, Л. Разработка Web-приложений с помощью PHP и MySQL. – М. : Вильямс, 2007.
3. Дари, К., Бринзаре, Б., Черчез-Тоза, Ф., Бусика, М. AJAX и PHP : разработка динамических веб-приложений. – М. : Символ-Плюс, 2006.
4. Шлоснейгл, Д. Профессиональное программирование на PHP. - М. : Вильямс, 2006.
5. Java 2. Библиотека профессионала. Том 1. Основы / Кей С. Хорстманн, Г. Корнелл. – М. : Вильямс, 2007.
6. Пауэлл, Т.А. Полное руководство по HTML. – Мн. : ООО «Попурри», 2001.
7. Хилайер, С., Мизик, Д. Программирование Active Server Pages. – М. : ИТД «Русская редакция», 2003.