

п р о ф е с с и о н а л ь н ы й ж у р н а л

# ПРОГРАММИСТ

№ 1\_2003

## Как написать шахматную программу

стр. 81

**ПРОГРАММА СЕРТИФИКАЦИИ MCAD**

**ПРИМЕНЕНИЕ ТЕХНОЛОГИИ FastCGI**

**УКАЗАТЕЛИ И ССЫЛКИ В C++**

# PHP: ОПЫТ ПРИМЕНЕНИЯ

МИХАИЛ ЕМЕЛЬЧЕНКОВ / [Michael@Emelchenkov.net](mailto:Michael@Emelchenkov.net) / [www.Emelchenkov.net](http://www.Emelchenkov.net) / [www.UnusualWorks.com](http://www.UnusualWorks.com)

## Почему я ушел от SSI?

Мне стало сложно поддерживать многоязыковый сайт, потому что при изменении даже одной строки текста приходилось менять эту строку и в других частях сайта, редактируя множество HTML-файлов в разных папках. До настоящего момента сайт был построен на SSI, но его возможностей оказалось недостаточно для отделения языкозависимых строк от HTML-разметки. Важно понимать, что современный сайт – это компьютерная программа, а не набор файлов. Современный сайт должен взаимодействовать с посетителем, помогать ему принимать решения, заставлять приходить еще и еще. Кроме того, хотелось максимально упростить изменение цветовой гаммы и сопутствующих ей элементов, чтобы менять внешний вид сайта на праздники, в соответствие со временем суток или временами года.

## Почему PHP?

Как я уже сказал, SSI оказался неподходящим средством для реализации моих идей. А PHP подходит идеально. Конечно, существует множество других технологий, например, Better Markup Language (BML; <http://www.bradfitz.com/bml>) или Parser (<http://www.parser.ru>). Но PHP широко распространен, лучше документирован и отлажен по сравнению с менее известными средствами. Кроме того, PHP – это язык (в отличие от того же SSI), соответственно, на нем можно написать любую программу или web-сайт. Альтернативой PHP является ASP, но ASP создан специально для Windows (хотя частично он портирован на другие платформы), а у нас используется Unix.

## Каков результат?

1. Все языкозависимые строки текста расположены в отдельном

файле, и теперь для изменения текста на сайте достаточно отредактировать только этот файл.

2. Гибкое управление внешним видом – наш сайт будет менять свое оформление по праздникам, радуя глаз посетителю.
3. Единый адрес для любого языка. Если раньше, например, было <http://www.UnusualWorks.com/about.htm> для английского языка и <http://www.UnusualWorks.com/rus/about.htm> для русского, то теперь существует единый адрес <http://www.UnusualWorks.com/about.php> для всех языков (т.к. содержимое страницы генерируется «на лету» в зависимости от выбранного пользователем языка).

## Мастерство PHP

Самое важное – определить задачи, которые вы хотите решить с помощью сайта. От них будет очень сильно зависеть, какой подход и какие инструменты выбрать. Именно проектирование программной логики сайта является наиболее сложной задачей.

Существует несколько методик создания сайта на PHP: CMS (Content Management System), внедрение островков PHP-кода в HTML-сайт для реализации определенного функционала, полная реализация сайта на PHP (на базе единого класса).

Решение на базе CMS требует огромных инвестиций – полнофункциональные CMS-системы стоят дорого, а их техническое сопровождение – еще дороже. Этот подход применяется лишь для больших проектов, где из CMS-системы можно получить стопроцентную отдачу (порталы издательств наподобие «Компьютерры» или «Домашнего компьютера», новостные сайты). Для небольших сайтов исполь-

зовать CMS даже вредно – излишнее усложнение может погубить проект. Естественно, научиться программировать на PHP при использовании CMS невозможно. Внедрение островков PHP-кода – наиболее распространенный вариант использования PHP. Обычно его используют при модификации существующих HTML-сайтов для придания им динамичности (например, с помощью подобных островков можно добиться сохранения информации о пользователе или установить динамически обновляемую колонку новостей или загрузить необходимую информацию из базы данных).

Полная реализация сайта на базе единого класса – оптимальный вариант для сайтов средних размеров, как с точки зрения динамической функциональности, так и легкости последующего сопровождения. Он обладает преимуществами больших CMS-систем (единообразие управления, воплощение истинной динамичности), так и лишен их недостатков (сложность и дороговизна). Наш сайт [www.UnusualWorks.com](http://www.UnusualWorks.com) построен именно по этому принципу. Нельзя сказать, что построение сайта на базе единого класса хуже или лучше внедрения островков PHP-кода, просто это разные подходы, и выбирать тот или иной подход следует исключительно из задач (ну и, конечно, ориентируясь на внешние факторы, как-то: требования заказчика, личная симпатия, оптимизация нагрузки на сервер и т.д.), а не ориентируясь на заказные статьи, неаргументированные советы или профессиональные течения. Основной прием создания сайта – разделять дизайн, код и информационное наполнение. Целью отделения HTML-разметки от PHP-кода является полное разграничение дизайна и логики приложения. Таким образом, дизайном может зани-

маться один человек, программированием – другой, заведовать информационным наполнением – третий.

Создавая любой код, целесообразно применять объектно-ориентированный подход, и PHP не является исключением – при использовании обычного процедурного программирования можно очень быстро потерять контроль над проектом. Естественно, это утверждение верно не всегда, в некоторых случаях ООП только усложняет процесс разработки.

Информационное наполнение, особенно когда оно представлено на нескольких языках, лучше всего расположить в отдельном файле или в базе данных, тогда его будет проще редактировать как автору, так и переводчикам. Желательно, чтобы доступ к этому файлу был организован через соответствующий класс, чтобы не пришлось переделывать весь сайт при изменении формата файла или переноса информационного наполнения в базу данных.

Для отделения дизайна используются шаблоны. Они не должны содержать информационного наполнения или программного кода, только HTML-разметку. Каждый шаблон – это логически завершенная часть страницы, поэтому нельзя, например, в одном шаблоне расположить заголовок `<HTML>...<BODY>...`, в другом – `...</BODY></HTML>`, а затем объединить их при выводе. Правильным решением будет создать единый шаблон страницы `<HTML>...<BODY>...</BODY></HTML>`, который уже сам будет включать шаблоны составных частей (например, `header.tpl`, `body.tpl` и `footer.tpl`).

Довольно распространенная система шаблонов в PHP – класс `FastTemplate` (<http://www.thewebmasters.net>). Но при близком рассмотрении `FastTemplate` тоже не является панацеей. В этом плане мне понравился `Smarty – the PHP compiling template engine` (<http://smarty.php.net>). Удобен он тем, что позволяет внедрять в шаблон свои собственные функции и фильтры (естественно, `Smarty` имеет обычные функции шаблонных

систем: компиляция, кэширование, вывод повторяющихся блоков и т.д.). Но тут надо быть очень внимательным: на `Smarty` очень легко реализовать практически любую возможность, а поэтому можно отойти от смысла применения шаблонов, перегрузив их кодом, относящимся к реализации, а не к дизайну (а вот код, относящийся к дизайну, как раз нужно хранить вместе с шаблонами, потому как за него отвечает скорее дизайнер, чем программист).

Некоторые разработчики вообще не применяют классы шаблонов, включение файлов функциями `include(...)` и `require(...)` для них и есть шаблоны. И это правильно, если разработка в конечном счете упрощается (разве не к этому стремится каждый программист?.. ведь чем код проще – тем он надежнее). Что должно содержаться в HTML-файле? Инициализация глобальных объектов (вернее, подключение соответствующего файла, в котором инициализируются глобальные объекты), загрузка шаблонов и вывод их на экран. То есть HTML-файл – это сплошной PHP-код. Например, можно загружать языкозависимые сообщения, или получать размер определенного файла (на нашем сайте таким образом вычисляется размер изображений и вставляется в HTML-тег `<IMG HEIGHT="..." WIDTH="...">`). Также хотелось бы рассмотреть наиболее типичные технические ошибки, встречающиеся при работе с PHP:

- Закачав файлы на Unix-сервер, им необходимо выставить атрибут `Execute` (если иного не сказано в FAQ вашего хостера), иначе сервер может выдавать ошибку.
- Если что-то не работает, не надо сразу кидаться отлаживать код. Вполне вероятно, что у вашего хостера установлена старая версия PHP, которая не поддерживает нововведений, произошедших в последней версии. Кстати, часто бесполезно просить хостера обновить PHP, т.к. на новых версиях интерпретатора старые скрипты могут работать неправильно, со-

ответственно, пришлось бы обновлять скрипты всем клиентам, что весьма затруднительно.

- Функции `include(...)` и `require(...)` скрывают в себе потенциальные ошибки. Включая один файл из одного каталога в другой файл из второго каталога, а эти два файла в еще один и т.д., происходит их объединение в один общий код, который выполняется последовательно. Поэтому, если функция `include(...)` работает в одном файле, то совсем не обязательно, что она будет работать, включи мы этот файл в какой-нибудь еще из другого каталога. Это значит, что надо четко знать местоположение всех файлов и по возможности не использовать относительные пути.

Динамические сайты иногда получают меньший рейтинг в поисковиках, чем статические, т.к. динамический сайт предполагает постоянное обновление находящейся на нем информации, обратную реакцию на действия пользователя. Конечно, рейтинг зависит от множества факторов, а не только от типа сайта, поэтому не стоит рассматривать это как повод для отказа от применения динамических технологий, но и забывать об этом тоже не стоит.

Я против того, чтобы PHP становился единственным средством создания сайта. Если проще смешать несколько технологий – обязательно сделайте это! Например, форум нашего сайта построен на скрипте `YaBB` (<http://www.yabbforum.com>), который, в свою очередь, написан на Perl. И это хорошо, потому что нам не пришлось разрабатывать собственный форум, а было достаточно найти подходящий из уже существующих. Безусловно, смешение разных технологий требует гораздо большей квалификации программиста, чем применение какой-либо одной. Но если процесс разработки, или сам сайт, или его сопровождение становятся проще – обязательно сделайте это!

*Выражаю отдельную благодарность Андрею Ревво и Валентину Озерову за помощь в написании статьи.*