

МЕТОДИКА ИСПОЛЬЗОВАНИЯ ИНТЕРАКТИВНЫХ ЭЛЕМЕНТОВ В СИСТЕМЕ УПРАВЛЕНИЯ САЙТОМ

Алтухов А. В.

Широкое использование систем управления сайтами с типовыми шаблонами не позволяет создавать привлекательные сайты. Применение различных современных технологий, таких как AJAX, JQuery, частично решают проблему привлекательности интерфейса. Выполнен анализ подходов по интеграции Flash-элементов с базой данных и языком программирования PHP. Рассмотрена методика замены стандартного интерфейса CMS Joomla! Flash-элементами. Проанализированы возможности интеграции Flash-анимации и языка программирования PHP. Приведены фрагменты кода с реализацией вывода информации из PHP во Flash-элемент и код на языке ActionScript.

Широке використання систем управління сайтами з типовими шаблонами не дозволяє створювати привабливі сайти. Застосування різних сучасних технологій, таких як AJAX, JQuery, частково вирішують проблему привабливості інтерфейсу. Виконано аналіз підходів щодо інтеграції Flash-елементів з базою даних і мовою програмування PHP. Розглянуто методику заміни стандартного інтерфейсу CMS Joomla! Flash-елементами. Проаналізовано можливості інтеграції Flash-анімації та мови програмування PHP. Наведено фрагменти коду з реалізацією виведення інформації з PHP в Flash-елемент і код на мові ActionScript.

The widespread use content management system with standard template does not allow you to create attractive sites. The use of various modern technologies such as AJAX, JQuery, partially solve the problem of attractive interface. The analysis of the approaches to the integration of Flash-elements to the database and the programming language PHP. The method of replacing the standard interface CMS Joomla! Flash-elements. The possibilities of integrating Flash-animation and programming language PHP. Code fragment to the implementation of information output from PHP into Flash-element and the code in ActionScript.

Алтухов А. В.

ст. преп. каф. КИТ ДГМА
tsl@ukr.net

ДГМА – Донбасская государственная машиностроительная академия, г. Краматорск.

УДК 004.415.2

Алтухов А. В.

МЕТОДИКА ИСПОЛЬЗОВАНИЯ ИНТЕРАКТИВНЫХ ЭЛЕМЕНТОВ В СИСТЕМЕ УПРАВЛЕНИЯ САЙТОМ

Широкое использование систем управления сайтами с типовыми шаблонами не позволяет создавать привлекательные сайты. Использование даже дорогих профессиональных шаблонов не всегда позволяет получить то, что хочет заказчик или пользователи. Наиболее характерно это для сайтов-визиток, которые содержат немного информации, но она должна быть красочной и динамичной. Применение различных современных технологий, таких как AJAX, JQuery, частично решают проблему привлекательности интерфейса.

Создание сайта полностью на основе Flash-анимации является хорошим решением, но не всегда устраивает заказчика по причине сложности обновления информации и имеет ряд проблем, связанных с долгой загрузкой сайта и его плохой индексацией в поисковых системах.

Решением проблемы может быть замена стандартного интерфейса CMS-систем Flash-анимацией. Такие разработки уже можно найти в сети, например: MotoCMS, fCMS2, UnidefCMS, Flaril CMS, Qubix. Часть из этих систем бесплатна. Главные недостатки – низкая популярность и отсутствие дополнительных модулей, иногда необходимых для создания сайта. Кроме этого можно заменить стандартный интерфейс в широко распространенных CMS Joomla!, Drupal, WordPress и других [1–3]. Рассмотрим возможности замены Flash-анимацией стандартного интерфейса в CMS Joomla!.

Целью работы является анализ подходов по интеграции Flash-элементов с базой данных и языком программирования PHP. Наиболее распространенный подход заключается в использовании передачи данных через XML-файл.

Проанализировав существующие компоненты для CMS Joomla!, использующие Flash-анимацию, были выделены два компонента, подходящие для изменения стандартного интерфейса. Компонент `com_flashmenu` позволяет создать динамическое меню на Flash взамен стандартного меню Joomla!. С помощью данного компонента можно создавать привлекательные горизонтальные или вертикальные меню в виде пиктограмм. Второй компонент J-AMFPHP версии 0.3.2 [4]. Данный компонент имеет большую функциональность и позволяет создавать Flash-элемент, в котором содержание меняется из PHP. В компоненте J-AMFPHP используется возможность Flash читать данные из XML-файлов. Для передачи информации из базы данных во Flash используется наиболее простой способ, данные извлекаются из базы, конвертируются в формат XML и с помощью PHP информация об XML-файле передается во Flash-элемент. Для извлечения данных Flash-элементом используется Flash Remoting Gateway, благодаря которому Flash-элемент может получить данные из разных источников в сети Интернет. Данные передаются во внутреннем формате данных Action Message Format (AMF). Flash-плеер обрабатывает данные и внедряет их во Flash-элемент [5–6].

Для реализации Flash-элементов в интерфейсе CMS Joomla! можно воспользоваться свободно-распространяемым компонентом J-AMFPHP, который включает 3 пакета: J-AMFPPHPCoreComponent, J-AMFPPHPAuthenticaitonService и J-AMFPPHPExampleServicesPlugin. После установки компонента его необходимо настроить. Компонент включает административную часть, с помощью которой выполняется управление Flash-элементом.

Пример работы AMFPHR показан на рис. 1 и 2. Для отображения информации используется один Flash-элемент, в котором меняется информация.



Рис. 1. Flash-элемент для вывода изображений и текста



Рис. 2. Изменение содержания при выводе Flash-элементом

Ниже приведен фрагмент кода на языке PHP, в котором показано создание и формирование объекта AMF для вывода данных во Flash-элемент.

```
$amf = new AMFObject($GLOBALS["HTTP_RAW_POST_DATA"]);
header(AMFPHP_CONTENT_TYPE);
$stoAddPos = strpos($output, "\301\260\0\0\1\0\0\0");
$time = (int) ((microtime_float() - $GLOBALS['amfphp']['startTime'])*1000);
$b = pack("d", $time); //
if (AMFPHP_BIG_ENDIAN) { //
    $r = strrev($b);
} else { //
    $r = $b;
}
$output = substr($output, 0, $stoAddPos) . $r . substr($output, $stoAddPos + 8);
```

Ниже приведен фрагмент кода на языке PHP, в котором показано формирование информации об Flash-элементе для вывода в html-код.

```
$amfbody->setValue($body[0]->body);
$amfbody->setSpecialHandling("RemotingMessage");
$amfbody->setMetadata("clientId", $body[0]->clientId);
$amfbody->setMetadata("messageId", $body[0]->messageId);
$GLOBALS['amfphp']['lastMessageId'] = $body[0]->messageId;
$methodName = $body[0]->operation;
$classAndPackage = $body[0]->source;
$amfbody->setSpecialHandling("Ping");
$amfbody->setMetadata("clientId", $body[0]->clientId);
$amfbody->setMetadata("messageId", $body[0]->messageId);
$amfbody->noExec = true;
```

Ниже приведен фрагмент html-кода, в котором показано внедрение Flash (swf)-элемента. На всех страницах используется один Flash-элемент, который выводит разную информацию.

```
<object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000" code-
base="http://fpdownload.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=7,0,0,0"
width="1000" height="690" align="middle">
    <param name="movie" value="main.swf?langval=2" />
    <param name="quality" value="high" />
    <param name="bgcolor" value="#352D01" />
    <embed src="main.swf?langval=2" quality="high" bgcolor="#352D01"
width="1000" height="690" name="main_69" align="middle" type="application/x-shockwave-
flash" pluginspage="http://www.macromedia.com/go/getflashplayer" /></object>
```

Ниже приведен код модуля на языке ActionScript Main.as для подключения к сервису AMFPHP и получения данных.

```
package {

    import flash.display.Shape;
    import flash.display.SimpleButton;
    import flash.display.Sprite;
    import flash.events.MouseEvent;
    import flash.net.NetConnection;
```

```
import flash.net.Responder;
import flash.text.TextField;

public class Main extends Sprite
{

    private var gateway:NetConnection;
    private var txt:TextField;
    private var btn:SimpleButton;

    public function Main()
    {
        createSimpleView();
    }

    public function init():void {
        gateway = new NetConnection();
        gateway.connect("http://localhost/services/amfphp");
        var responder:Responder = new Responder(onResult, onFault);
        var arg:Array = new Array();
        gateway.call( "views.get", responder, 'techblog_all',"arg, 0, 10);

    }

    public function onResult(responds:Object):void {
        txt.text = "onResult: \n";
        reflect(responds);
    }

    public function onFault(responds:Object):void {
        txt.text = "onFault : " +responds.error ;
        reflect(responds);
    }

    public function createSimpleView():void {
        txt = new TextField();
        txt.x = txt.y = 10;
        txt.width = 390;
        txt.height = 200;
        txt.border = true;

        addChild(txt);
        btn = new SimpleButton();
        btn.x = 10;
        btn.y = 220;
        btn.upState = createRectangle( 0x00FF00 );
        btn.overState = createRectangle( 0xFF0000 );
        btn.downState = createRectangle( 0x0000FF );
        btn.hitTestState = btn.upState;
```

```
        btn.addEventListener( MouseEvent.CLICK, btnClick );
        addChild(btn);
    }

    private function btnClick(event:MouseEvent):void {
        init();
    }

    private function createRectangle(color:uint):Shape {
        var rect:Shape = new Shape( );
        rect.graphics.lineStyle( 1, color );
        rect.graphics.beginFill( color );
        rect.graphics.drawRect(0,0,150,50)
        rect.graphics.endFill();
        return rect;
    }

    private function reflect(obj:Object):void{
        for(var i:String in obj){
            txt.appendText("\n"+i+": "+obj[i].node_title);
        }
    }
}
```

ВЫВОДЫ

Таким образом, на основе рассмотренной методики по замене стандартного интерфейса CMS Joomla!, можно создавать динамичные сайты с привлекательным дизайном и простые в управлении. Кроме этого решается проблема с поисковыми системами, выбранный подход позволит создавать сайты, которые хорошо индексируются поисковыми системами за счет использования формата XML. В отличие от Flash, в котором информация закрыта и плохо доступна для индексации, формат XML открыт для чтения поисковыми системами, что дает возможность хорошей индексации и продвижения во многих поисковых системах. При реализации данного подхода был выявлен недостаток, который может ограничить область использования в сайтах различной направленности. При формировании вывода поисковыми модулями использование Flash-элементов практически невозможно.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Горнаков С. Г. *Осваиваем популярные системы управления сайтом (CMS)* / С. Г. Горнаков. – М. : ДМК Пресс, 2009. – 336 с. : ил.
2. Вандюк Д. *CMS Drupal: руководство по разработке системы управления сайтом* : пер. с англ. / Джон Вандюк, Мэтт Вестгейт. – М. : Вильямс, 2008. – 400 с. : ил.
3. Граф Хаген. *Создание web-сайтов с помощью Joomla! 1.5!* / Хаген Граф. – М. : Дом Вильямс, 2008. – 294 с.
4. *Suhreed Sarkar. Joomla! with Flash.* – Birmingham : Published by Packt Publishing Ltd., 2009. – 260 p.
5. *LeBlanc, Joseph. Learning Joomla! 1.5 Extension Development.* – Birmingham : Published by Packt Publishing Ltd., 2007.
6. *Kennard J. Mastering Joomla! 1.5 Extension and Framework Development* / James Kennard. – Birmingham : Published by Packt Publishing Ltd., 2007.