



24-25 мая 2007, г. Москва

**6-я международная конференция
“Современные технологии
эффективной разработки веб-приложений
с использованием PHP”**

PHP&Flex - новая альтернатива для создания RIAs

Иванников Андрей
Улич Дмитрий

Что такое Flex?

Adobe Flex 2 – это решение, позволяющее разрабатывать Rich Internet Applications на основе родственной Flash технологии.

Описание интерфейса, и не только, основано на диалекте XML – MXML.

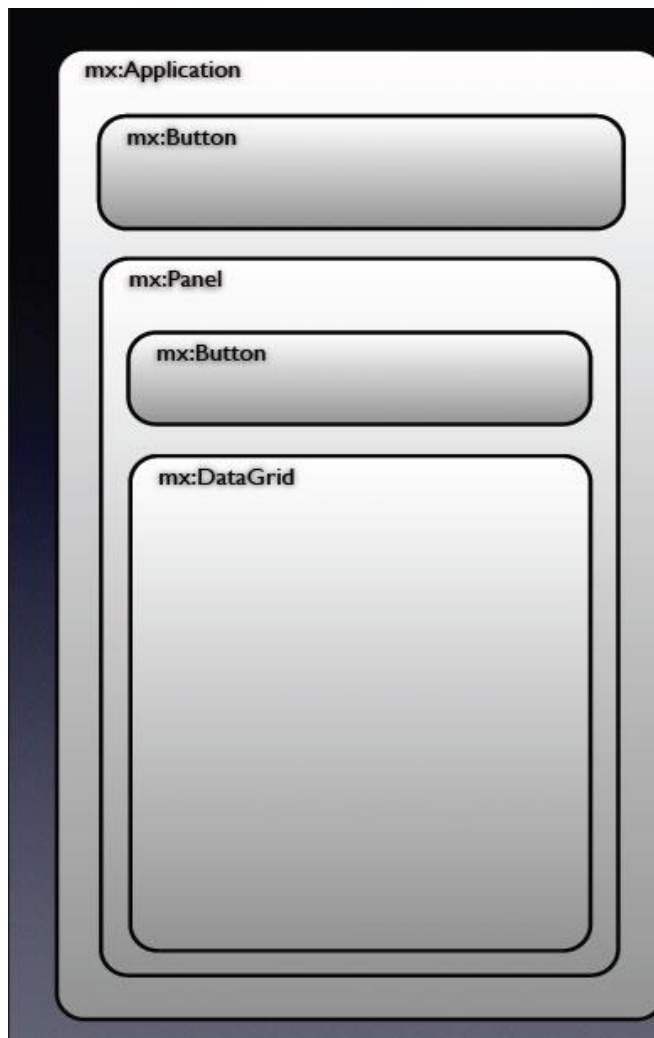
Описание клиентской логики на ActionScript3 (AS3)

В результате MXML->AS3->swf файл.

Линейка продуктов Flex 2

- Flex Builder 2 – IDE основанная на eclipse.
- Flex framework and SDK
- Flex Charting 2
- Flex Data Services 2

Структура Flex приложений



The diagram illustrates the structure of a Flex application. It shows a hierarchy of components: an outer `mx:Application` container, followed by an `mx:Button`, then an `mx:Panel` containing another `mx:Button` and an `mx:DataGrid`.

```
<?xml version="1.0" ?>  
<mx:Application>  
  <mx:Button width="100" height="50"/>  
  <mx:Panel width="100%" height="100%">  
    <mx:Button width="100%" />  
    <mx:DataGrid width="100%" height="100%" />  
  </mx:Panel>  
</mx:Application>
```

Пример

```
<WebService id="ws" wsdl="some.wsdl">
```

```
<Button label="Get Data"  
  click="ws.getData()">
```

```
<DataGrid  
  dataProvider="{ws.getData.result}">
```

```
<LineChart  
  dataProvider="{ws.getData.result}">
```

MXML <-> ActionScript

```
<andry:CustomButton label="Click Me"/>
```

```
package{  
    import mx.controls.Button;  
    public class CustomButton extends Button{  
        public function CustomButton(){  
            supper();  
        }  
    }  
}
```

ActionScript 3

ООП! Основан на стандарте ECMAScript 4.

- Строгая и динамическая типизация
- Наследование, интерфейсы, модульность
- Спецификаторы доступа
- Reflection API
- ECMAScript for XML (E4X)
- DOM3 модель событий
- Динамические классы (runtime добавление методов и свойств)

Другие аспекты разработки

- ASDoc – документирование в коде
- Mercury QuickTest – приемочные тесты
- FlexUnit – модульные тесты
- Flex Stress Testing – нагрузочные тесты
- Cairngorm - архитектурный фреймворк
- Flex-Ajax Bridge – AS <-> JS
- Apollo - desktop приложения

Из рубрики “Лучше один раз увидеть, чем сто раз услышать”!

Смотрим и про себя думаем, как бы мы все это сделали на JS?! %)

HTML&JavaScript vs MXML&ActionScript

	JS	AS
• Кроссбраузерность	-	+
• Функционал языка	-	+
• Распространенность	+	-
• Размер приложений	+	-
• Индексация в поисковиках	+/-	-
• Скорость разработки RIAs	-	+
• Производительность	-	+

Обмен данными

- Использовать фреймворки, которые сериализуют данные в формат AMF
- SOAP запросы
- Сокеты (в том числе и бинарные)
- Просто использовать GET и POST запросы

Реализована поддержка AMF

- PHP
- JAVA
- .NET
- ColdFusion
- Ruby

Фреймворки для связки PHP и Flex

- AMFPHP
- WEBORB
- SabreAMF

Общее

- Поддержка AMF3
- Сопоставление классов (class mapping) между Flex и PHP
- Поддержка PHP5

Сравнение

	AMFPHP	WEBORB	SabreAMF
Поддержка PHP4	+	-	-
Авторизация	+	+	-
Документация	-	-	-
Простота установки	+	+	-
Ведение логов	+	+	-
php_amf_extension	+	-	-
Браузер для служб	+	-	-

Flex

```
<mx:Script><![CDATA[
    import mx.controls.Alert;
    import mx.rpc.events.FaultEvent;
    import mx.rpc.events.ResultEvent;
    //обрабатываем результат
    private function resultHandler(event:ResultEvent):void{
        dg.dataProvider = event.result;
    }
    //обрабатываем ошибки
    private function faultHandler(event:FaultEvent):void{
        Alert.show(event.fault.faultString, "Error");
    }
]]></mx:Script>
<mx:RemoteObject id="ro" result="resultHandler(event)"
    fault="faultHandler(event)" destination="users" source="dummy.Users"/>
<mx:DataGrid id="dg" x="10" y="40" width="746" height="500"/>
<mx:Button x="10" y="10" label="Загрузить данные" click="ro.getUsers()"/>
```


services-config.xml

```
<default-channels>
  <channel ref="my-amf"/>
</default-channels>
<destination id="users">
  <properties>
    <source>*</source>
  </properties>
</destination>
<channel-definition id="my-amf"
  class="mx.messaging.channels.AMFChannel">
  <endpoint uri="http://localhost/amfphp_1.9/gateway.php"
  class="flex.messaging.endpoints.AMFEndpoint"/>
```

PHP

```
class Users {  
    public function getUsers(){  
        $users = array();  
        $query = mysql_query("select * from users");  
        while ($row = mysql_fetch_object($query)){  
            $users[] = $row;  
        }  
        return $users;  
    }  
}
```

НИЧЕГО!

Class mapping

```
package vo{  
    [RemoteClass(alias="dummy.vo.UserVO")]  
    public class UserVO{  
        public var userId:int;  
        public var userName:String;  
    }  
}
```

```
class UserVO {  
    public $_explicitType = "dummy.vo.UserVO";  
    public $userId;  
    public $userName;  
}
```

```
import vo.UserVO;  
private var user:UserVO;  
private function createNewUser():void{  
    var newUser:UserVO = new UserVO();  
    newUser.userName = uName.text;  
    ro.createNewUser(newUser);  
}
```

```
<mx:Panel x="618" y="40" width="279" height="155" layout="absolute"/>  
  <mx:Form x="10" y="10">  
    <mx:FormItem label="Имя:" required="true">  
      <mx:TextInput id="uName"/>  
    </mx:FormItem>  
  </mx:Form>  
  <mx:Button x="89" y="83" label="Добавить"  
click="createNewUser()"/>  
</mx:Panel>
```

```
require_once("vo/UserVO.php");
```

```
public function getUsers(){
    $users = array();
    $query = mysql_query("SELECT user_id, user_name FROM users");
    while ($row = mysql_fetch_object($query)){
        $user = new UserVO();
        $user->userId = $row->user_id;
        $user->username = $row->user_name;
        $users[] = $user;
    }
    return $users;
}

public function createNewUser (UserVO $newUser){
    mysql_query("INSERT INTO users SET user_name = '" .
    $newUser->userName . "'");
}
```

Спасибо за внимание!
Вопросы