



*Katedra technické a
informační výchovy*

*5. mezinárodní
konference*

*Nové technologie ve
výuce*

*PedF MU Brno
listopad 2011*

FLASH APLIKACE KOMUNIKUJÍCÍ S LMS MOODLE

Robin Horák
Masarykova univerzita, Pedagogická fakulta
robin.horak@seznam.cz

Jiří Hrbáček
Katedra technické a informační výchovy, PedF MU
hrbacek@posta.ped.muni.cz

Resumé: To, že se ve vývojovém prostředí Adobe Flash dají vytvářet kvalitní výukové materiály od animací až po výukové programy, není žádnou novinkou. Studijní materiály musíme samozřejmě někde zpřístupnit studentům, k čemuž je velmi hojně využívám LMS Moodle, který je uživatelsky přívětivý a i méně technicky zdatní pedagogové jsou schopni v něm vytvářet kurzy pro svoje studenty. Nicméně LMS Moodle na vše sám nestačí a někdy je nutné pro tvorbu specifických studijních materiálů sáhnout po něčem jiném – například po Adobe Flash. Naštěstí je možné aplikace vytvořené v Adobe Flash vložit do LMS Moodle a docílit i toho, že Flash aplikace a LMS Moodle spolu budou komunikovat. Tento příspěvek si klade za cíl poskytnout návod, jak je možné této vzájemné komunikace docílit.

Klíčová slova: Adobe Flash, Moodle, LMS, SWF Activity Module

1. ÚVOD

To, že se ve vývojovém prostředí Adobe Flash dají vytvářet kvalitní výukové materiály od animací až po výukové programy, není žádnou novinkou (Horák; Hrbáček, 2010). Studijní materiály musíme samozřejmě někde zpřístupnit studentům, k čemuž je velmi hojně využívám LMS Moodle, který je uživatelsky přívětivý a i méně technicky zdatní pedagogové jsou schopni v něm vytvářet kurzy pro svoje studenty. Nicméně LMS Moodle na vše sám nestačí a někdy je nutné pro tvorbu specifických studijních materiálů sáhnout po něčem jiném – například po Adobe Flash. Naštěstí je možné aplikace vytvořené v Adobe Flash vložit do LMS Moodle a docílit i toho, že Flash aplikace a LMS Moodle spolu budou komunikovat.

O možnosti komunikace mezi Adobe Flash a LMS Moodle, respektive o nějakém funkčním řešení se moc neví, protože tato možnost není ze strany LMS Moodle a Adobe Flash oficiálně podporována. Samozřejmě zde oficiálně máme SCORM, ale Adobe Flash na rozdíl od Adobe Captivate neobsahuje SCORM Packager a každý, kdo někdy zkoušel do vyhledávače zadat spojení „Flash and SCORM,“ našel v diskusních fórech mnoho otázek, ale málo odpovědí.

Jako vývojář výukových aplikací jsem se již mnohokrát setkal s požadavkem, aby bylo možné výsledky studentů z výukových programů vytvořených ve Flashi ukládat do LMS Moodle. Tento požadavek vyřešil SWF Activity Module, jehož vývojem se zabývá Sam Matbury a vše o tomto modulu si můžete přečíst na jeho [stránkách](#) v anglickém jazyce.

SWF Activity modul se osvědčil a začíná se využívat v LMS Moodle na VUT a Masarykově univerzitě v Brně.

2. MOŽNOSTI MODULU

S pomocí SWF modulu můžeme snadno pokročilým způsobem vkládat Flash aplikace. Při vkládání můžeme přímo v LMS Moodle nastavovat například rozměry aplikace, zarovnání aplikace, požadovanou verzi Flash Playeru, barvu pozadí aplikace (popřípadě průhlednost pozadí), povolovat či zakazovat možnost zobrazení na celou obrazovku apod.

Dále je možné nastavit maximální počet bodů a minimální počet bodů pro úspěšné splnění úkolu. Pokud budeme mít v našem kurzu více aplikací, je dokonce možné nastavit podmínku, že studenti budou muset úkoly v jednotlivých aplikacích plnit popořadě. Pokud tedy student nebude mít dostatečný počet bodů z úkolu v aplikaci 1, nebude moci spustit aplikaci 2.

Tou nejdůležitější věcí, kterou SWF Activity Module umožňuje, je ukládat výsledky z Flash aplikace do prostředí LMS Moodle. Tyto výsledky pak budou k dispozici stejným způsobem, jako výsledky z jakéhokoliv jiného testu vytvořeného přímo v LMS Moodle.

Další užitečnou možností je zasílat různá data a informace z LMS Moodle do Flash aplikace. Můžeme si sami definovat až 3 parametry, které Flash aplikaci zašleme. Navíc LMS Moodle zasílá Flash aplikaci informace o adrese kurzu, ID kurzu apod., což je užitečné především pro tvůrce Flash aplikací.

Pokud nás tedy tento modul svými funkcemi zaujal, téměř nic nám nebrání si jej nainstalovat.

3. INSTALACE MODULU

Požadavky na systém

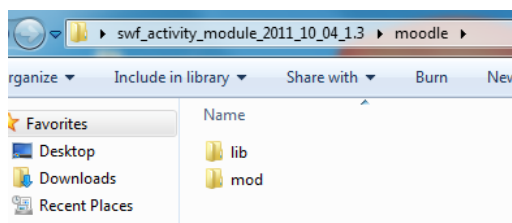
Než začneme se samotnou instalací, je dobré si ověřit, zda splňujeme následující požadavky

- LAMP server (Linux, Apache, MySQL, PHP)
- PHP 5+
- Moodle 1.9+ *

*Je možné, že SWF Activity module poběží i na starší LMS Moodle, nicméně ve starších verzích nebyl testován. Moodle 2.0 není v době psaní tohoto příspěvku podporován.

Instalace

SWF Activity Module můžete stáhnout [zde](#). Po stažení rozbalte zip archiv a v rozbalené složce otevřete složku moodle a v ní uvidíte dvě složky s názvy lib a mod. (viz Obr. 1)



Obr. 1 Obsah rozbalené složky

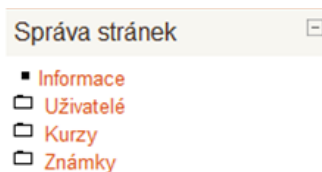
Obsah složky mod nakopírujte do složky mod v kořenovém adresáři LMS Moodle. Na serveru s LMS Moodle bude tedy následující sktruktura složek – kořenový adresář LMS Moodle/mod/swf/.

Tímto jste do LMS Moodle nahráli SWF Activity Module.

Dále obsah složky lib nakopírujte do složky lib v kořenovém adresáři LMS Moodle (kořenový adresář LMS Moodle /lib/ amfphp/).

Tímto jste nainstalovali službu AMFPHP (Action Message Format PHP) do LMS Moodle, která je potřebná pro správnou funkci SWF Modulu. Tato služba umožňuje Flash aplikacím běžícím na straně klienta přímo volat PHP metody. Služba je již předkonfigurovaná a není potřeba nic měnit. Více o AMFPHP se můžete dozvědět [zde](#).

Poté je nutné se přihlásit do LMS Moodle jako administrátor a v „nabídce správa“ stránek zobrazit „informace“ (viz Obr. 2) a potvrdit instalaci SWF modulu.

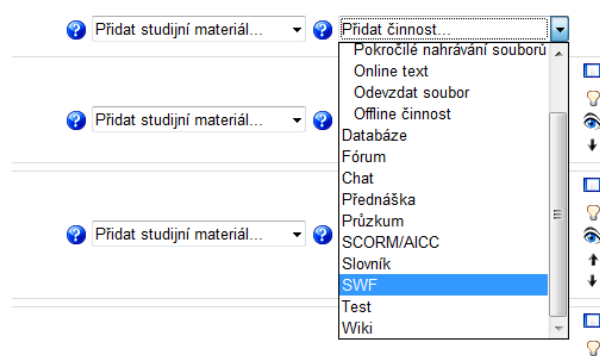


Obr. 2 Správa stránek a informace

Nyní je již možné SWF Modul v prostředí LMS Moodle začít využívat.

4. IMPLEMENTACE A POUŽÍVÁNÍ MODULU

V prostředí LMS Moodle vložíme Flash aplikaci přes „Přidat činnost...“ a z nabídky vybereme „SWF“ (viz Obr. 3)



Obr. 3 Vkládání Flash aplikací přes SWF Modul

Tímto způsobem můžeme Flash aplikaci jednoduchým způsobem vložit do prostředí LMS Moodle. Navíc máme k dispozici mnoho různých možností vložení, z nichž některé již byly nastíněny výše v textu. Věřím, že není třeba popisovat každou z těchto možností, jelikož kdo umí základy angličtiny, nebude mít ani s pokročilejším vkládáním Flash aplikací nejmenší problém.

Nicméně pro to, aby bylo možné vložit složitější aplikaci obsahující externí soubory, předávat data z LMS Moodle do Flash aplikace nebo posílat data z Flash aplikace do LMS Moodle (například známky a komentáře), je nutné přizpůsobit i Flash aplikaci, což už je práce pouze pro vývojáře Flash aplikací.

Vkládání aplikací využívajících externí soubory

Při vkládání Flash aplikace, která pro svůj chod využívá externí soubory, musíme myslet na to, že v případě využití SWF Activity Modulu v systému Moodle nemůžeme zadávat cestu k souborům způsobem, jako bychom ji zadávali například v případě desktopové Flash aplikace v prostředí MS Windows.

Pokud bychom ve Flash aplikaci vložené přes SWF Activity Module do prostředí Moodle zadali relativní adresu „data/data.xml“, tak se nám XML soubor nenačte.

Správný formát adresy by tedy byl například:
„<https://www.vutbr.cz/elearning/file.php/1000099/data/data.xml>“.

Absolutní adresu můžeme ve Flash aplikaci samozřejmě „napevno“ zadat, ale tato aplikace by nám pak nefungovala v jiném kurzu, protože číslo 1000099 z adresy je ID pouze jednoho kurzu v LMS Moodle. Na druhou stranu pro nás tohle řešení může být i výhodné, protože tímto způsobem máme možnost do jisté míry ochránit naši aplikaci před použitím v jiném kurzu nebo mimo LMS Moodle.

Samozřejmě ne vždy budeme chtít, aby nám aplikace fungovala pouze v jednom kurzu, a určitě nebudeme chtít při změně kurzu přepisovat adresu ve Flash aplikaci.

Naštěstí máme možnost adresu kurzu zjistit díky parametrům, které LMS Moodle předává Flash aplikaci.

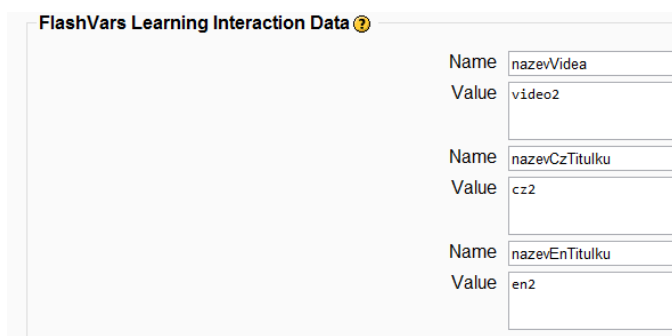
Vytvoříme si tedy ve Flash aplikaci proměnnou

```
private var moodledata:String;  
a do této proměnné uložíme parametr předaný z Moodle  
moodledata = this.root.loaderInfo.parameters.moodledata;  
pak už jen budeme proměnnou moodledata přidávat před  
cestu k našim souborům, například:  
var cestaSouboru:String;  
cestaSouboru = moodledata + "data/data.xml";
```

SWF modul nám nabízí ještě možnost posílat vlastní, námi definované, parametry ze systému Moodle.

Představme si situaci, že do našeho kurzu chceme vložit 30 videí a každé video bude mít navíc přepínatelné české a anglické titulky. Samozřejmě máme možnost vytvořit 30 různých složek, které budou obsahovat vždy stejně pojmenovaný videosoubor a stejně pojmenované titulky. Náš přehrávač vytvořený v Adobe Flash umístíme do každé této složky a přehrávač bude vždy hledat titulky a video s pevně zadaným názvem. V konečném výsledku tedy skončíme se 30 složkami a 30 stejnými přehrávači, což není zrovna ideální.

Mnohem elegantnější řešení je mít pouze jeden přehrávač, kterému budeme z LMS Moodle předávat název videa a názvy titulků, které má hledat. Budeme mít také pouze jednu složku pro videa a jednu složku pro titulky. Zní to jednoduše a jednoduché to také je, protože díky SWF modulu můžeme Flash aplikaci předat vlastní parametry, které máme možnost zadat při vkládání Flash aplikace pomocí SWF modulu (viz Obr. 4).



Obr. 4 Předávání vlastních parametrů Flash aplikaci
V naší Flash aplikaci si pak vytvoříme proměnné:

```
private var nazevVidea:String;  
private var nazevCzTitulku:String;  
private var nazevEnTitulku:String;
```

a uložíme si do nich parametry předané z LMS Moodle

```
FlashVars.vars = this.root.loaderInfo.parameters;  
nazevVidea = this.root.loaderInfo.parameters.nazevVidea;  
nazevEnTitulku=this.root.loaderInfo.parameters.nazevEnTitulku;  
nazevCzTitulku=this.root.loaderInfo.parameters.nazevCzTitulku;
```

Nesmíme však zapomenout naimportovat třídu pro práci s těmito parametry (tuto třídu jsme si již stáhli spolu s modulem).

```
import com.matbury.sam.data.FlashVars;
```

Poté bude naše cesta k video souboru například:

```
var cestaVidea:String;  
cestaVidea = moodledata + "video/"+nazevVidea+".flv";
```

A nyní to hlavní, co nám SWF modul nabízí – možnost odesílat výsledky a komentáře z Flash aplikace do systému Moodle.

Nejprve musíme naimportovat potřebné třídy, které byly staženy spolu s modulem. Kromě těchto tříd samozřejmě musíme importovat další, které budeme v naší třídě používat, např. Event, TextField apod.

```
import com.matbury.sam.data.Amf;  
import com.matbury.sam.data.FlashVars;
```

Do FlashVars uložíme parametry předané z LMS Moodle. Již jsme si tímto způsobem předávali vlastní parametry, ale kromě vlastních parametrů jsou předávány i parametry, z nichž některé jsou dále využívány pro ukládání známek a komentářů do LMS Moodle. Kompletní seznam předávaných parametrů je k dispozici [zde](#).

```
FlashVars.vars = this.root.loaderInfo.parameters;
```

Dále vhodně implementujeme funkci, která bude posílat známky a komentáře. Níže je uvedena abstraktní funkce přímo od tvůrce modulu, kterou si ale můžeme dle potřeby upravit a zjednodušit.

Nejdůležitější jsou pro nás žlutě zvýrazněné části kódu obj.feedback, což je komentář a obj.rawgrade, což je počet bodů nebo procent. Namísto pevně zadaného řetězce a čísla pouze přiřadíme vlastní proměnné s komentářem a počtem bodů z naší aplikace. Aktuální verze je k dispozici [zde](#).

```
private var _amf:Amf;  
private var _text:TextField;  
// Add the required code to create a text field.  
  
private function sendGrade():void {  
_amf = new Amf();  
// create Flash Remoting API object  
  
_amf.addEventListener(Amf.GOT_DATA, gotDataHandler);  
// listen for server response
```

```

_amf.addEventListener(Amf.FAULT, faultHandler);
// listen for server fault

var obj:Object = new Object();
// create an object to hold data sent to the server

obj.gateway = FlashVars.gateway;
// (String) AMFPHP gateway URL

obj.swfid = FlashVars.swfid;
// (int) activity ID

obj.instance = FlashVars.instance;
// (int) Moodle instance ID

obj.feedback = "Some text feedback for teachers and/or learners to read";
// (String) optional

obj.feedbackformat = Math.floor(getTimer() / 1000);
// (int) elapsed time in seconds

obj.rawgrade = 100;
// (Number) grade, normally 0 - 100 but depends on grade book settings

obj.servicefunction = "Grades.amf_grade_update";
// (String) ClassName.method_name

// send the data to the server
_amf.getObject(obj);
}

// clean up listeners
private function gotDataHandler(event:Event):void {
    _amf.removeEventListener(Amf.GOT_DATA, gotDataHandler);
    _amf.removeEventListener(Amf.FAULT, faultHandler);

    // check if grade was sent successfully
    if(_amf.obj != false) {
        _text.appendText("Your grade has been sent to the grade book.");
    } else {
        _text.appendText("There was a problem sending your grade to the grade book.");
    }
}

private function faultHandler(event:Event):void {
    _amf.removeEventListener(Amf.GOT_DATA, gotDataHandler);
    _amf.removeEventListener(Amf.FAULT, faultHandler);

    // Display server errors
    var msg:String = "Error: ";
    for(var s:String in _amf.obj.info) { // trace out returned data
        _text.appendText("\n" + s + "=" + _amf.obj.info[s]);
    }
}

```

Výše uvedený kód můžeme zpracovat do naší Flash aplikace. Kdyby nám zaslání známek nefungovalo, doporučuji vyzkoušet [testovací aplikaci](#) přímo od autora, než začneme hledat chyby v kódu. Pokud by zaslání bodů nefungovalo ani v testovací aplikaci, může to znamenat, že jsme třeba správně nainstalovali službu AMFPHP.

Pokud testovací aplikace bude fungovat v pořádku, je chyba pravděpodobně v naší aplikaci. V případě, že by se chybu nepodařilo odstranit, doporučuji se podívat do diskuze [zde](#), kde se řeší problémy s odesíláním bodů do LMS Moodle.

5. ZÁVĚR

LMS Moodle přináší na první pohled určitá omezení pro Flash aplikace, avšak jak vyplývá z příspěvku, tato omezení lze překonat. Spojení s LMS Moodle však přináší i nespornou výhodu pro tvůrce Flash výukových aplikací – a to, že odpadá nutnost tvorby administračního systému pro Flash aplikace, pokud nějakým způsobem potřebujeme získávat a organizovat výsledky od studentů.

Věřím, že příspěvek bude přínosný pro tvůrce výukových Flash aplikací, kteří by rádi bez omezení využívali všech výhod Adobe Flash i výhod LMS Moodle.

6. ODKAZY

HORÁK, Robin; HRBÁČEK, Jiří. *Možnosti využití Adobe Flash ve výuce*. In *Nové technologie ve výuce 2010*. Brno : Masarykova Univerzita, 2010. s. 4. Dostupné z WWW: <<http://boss.ped.muni.cz/hrbacek/ktivkonf2010/prisp/nab1/prisp6.pdf>>.

MATBURY, Sam. *SWF Activity Module* [online]. 2011 [cit. 2011-12-20]. Dostupné z WWW: <<http://code.google.com/p/swf-activity-module/>>.

AUTOŘI



Bc. et Bc. Robin Horák

Student a absolvent Pedagogické fakulty Masarykovy univerzity. Probíhající studia – Učitelství technické a informační výchovy pro základní školy, Učitelství speciální pedagogiky pro základní školy a Učitelství anglického jazyka a literatury pro základní školy. V současnosti se autor nadále zabývá vývojem aplikací v Adobe Flash, PHP, HTML, JavaScript a MySQL.



Ing. Jiří Hrbáček, Ph.D.

Studium a kurzy – FEL ČVUT Praha obor technická kybernetika, FTSD – diagnostika mikroprocesorů, kurz využití počítače pro e-learning, kurz využití počítače v distančním vzdělávání, kurz pro autory, tutorů a administrátory distančního vzdělávání, rozšiřující studium speciální pedagogiky., PGS obor speciální pedagogika – využití distančních studijních opor v prezenční výuce. Od roku 1996 se intenzivně věnuje problematice distančního vzdělávání. Od roku 2002 vývoji a výuce systémů a programování ve Flashi, výuce a tvorbě multimediálních interaktivních výukových opor. Výuka elektroniky, elektrotechniky, digitální techniky.