Využití technologie Flex pro tvorbu webových aplikací a komponent

Usage of the Flex technology to produce Web applications and components

Lukáš Malý

Bakalářská práce
2010

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Rozsah bakalářské práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. **GILMORE W. JASON, Velká kniha PHP a MySQL, Zoner Press, Brno, 2007, ISBN: 80-86815-53-6.**
2. **WELING L. a THOMSON L., MySQL, Computer Press, Brno, 2005, ISBN: 80-251-0671-3.**
3. **GASSNER DAVID, Adobe Flex 3 Bible, John Wiley & Sons, United Kingdom, 2008, ISBN: 978-04-702-8764-4.**
4. **ŠPINAR DAVID, Tvoříme přístupné webové stránky, Zoner Press, Brno, 2004, ISBN: 80-86815-11-0.**
5. **CEDERHOLM DAN, Webdesign s webovými standardy, Zoner Press, Brno, 2004, ISBN: 80-86815-15-3.**
6. **PAPA JOHN, Silverlight datové služby, O'Reily/Zoner Press, Brno, 2009, ISBN: 978-80-7413-041-0.**
7. **ULLMAN L., PHP a MySQL, Computer Press, Brno, 2004, ISBN: 80-251-0063-4.**

Vedoucí bakalářské práce: **Ing. Zuzana Oplatková, Ph.D.**
Ústav informatiky a umělé inteligence

Datum zadání bakalářské práce: **5. března 2010**

Termín odevzdání bakalářské práce: **1. června 2010**

Ve Zlíně dne 5. března 2010

prof. Ing. Vladimír Vašek, CSc.
děkan

UNIVERZITA TOMÁŠE BATI VE ZLÍNĚ
Fakulta aplikované informatiky
L.S.
①

doc. Ing. Ivan Zelinka, Ph.D.
ředitel ústavu

ABSTRAKT

Práce přináší pohled na technologii Adobe Flex a její využití k tvorbě interaktivních klientských součástí informačních systémů společně s obvyklými prostředky internetu/intranetu. Zkoumá i jiné alternativy k Flex a srovnává je. V praktické části se zaměřuje na tvorbu publikačního systému a jeho rozšíření o komponentu vytvořenou ve Flex. Popisuje moderní přístup k tvorbě webových aplikací.

Klíčová slova: CMS, Adobe Flex, MXML, ActionScript, PHP, MySQL

ABSTRACT

This Bachelor thesis gives insight on Adobe Flex technology and its use for producing interactive clien-side parts of information systems, together with the usual means of Internet / intranet. Examines the alternatives to Flex and compares them. The practical part of thesis focuses on the creation of the Web publication system and its extension by component created in Flex. Describes a modern approach to building Web applications.

Keywords: CMS, Adobe Flex, MXML, ActionScript, PHP, MySQL

Vážím si podpory, které se mi dostávalo při studiu, mé poděkování za pochopení a trpělivost patří rodině, blízkým i ostatním lidem v mém okolí. Zejména bych zde rád poděkoval svému vedoucímu bakalářské práce Ing. Zuzaně Oplatkové, Ph.D. za vstřícné jednání z její strany, cenné rady a připomínky.

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně …….

podpis diplomanta

OBSAH

ÚVOD

Prostředí internetu a intranetu prošlo v posledních letech velkým rozvojem, ať už z pohledu rozsahu využití nebo možností nových technologií. Z původního, majícího spíše informativní charakter, se internet stal nejen součástí reálného světa, ale i vlastním světem, který umožňuje překonat mnohá naše omezení. S pronikáním k různým skupinám uživatelů se otevřel prostor do oblastí, v kterých vznikají podněty také k použití nových postupů a proměn jeho podoby.

Ve své práci popisuji jednu z novějších technologií používaných na internetu - Adobe Flex. Je perspektivním doplněním klasických technologií a v něčem i alternativou. Flex převzal mnoho užitečných vlastností z Flashe, nabízí však více než poutavé prezentace s animovaným obsahem. Umožňuje programátorský přístup k práci a má velký potenciál využitelný v oblasti zpracování dat. Uvádím i některé další technologie a porovnávám je z pohledu na jejich prostředky k tvorbě webových aplikací.

V praktické části bylo mým úkolem vytvořit jednoduchý publikační systém rozšířený o některé další komponenty vztahující se k činnosti soukromého subjektu (například k organizaci jím provozovaných aktivit). Nabízí se celá řada systémů správy obsahu, které stačí nainstalovat a přizpůsobit jejich účelu. Vytvoření vlastního, byť jednoduchého systému, však může být cennou zkušeností s realizací projektu tohoto typu. Jako platformu na straně serveru jsem zvolil PHP v kombinaci s databází MySQL, mohl jsem tak využít svých zkušeností s programováním v tomto jazyce. Věnuji se zde dále návrhu a realizaci aplikace s využitím frameworku Flex.

I. TEORETICKÁ ČÁST

1 ADOBE FLEX

1.1 Co je Flex

Adobe Flex je platforma pro vývoj směřující do prostředí Flash Playeru, umožňuje tvorbu interaktivních aplikací pro internet, intranet nebo aplikací samostatně spouštěných v počítači nad Adobe Integrated Runtime (AIR).

V souvislosti s Flexem se často hovoří o RIA (Rich Internet Applications). Koncepce RIA vychází z požadavků zajištění větší funkcionality moderních aplikací. U jejího zrodu byla myšlenka tvorby aplikací spouštěných v prostředí internetového prohlížeče, které nabízí podobné možnosti jako programy z prostředí operačního systému. Takto ji původně uvedla společnost Macromedia, dnes je však poněkud širším pojmem který zahrnuje i desktopové aplikace cross-platformního charakteru a začíná pronikat do oblasti mobilních aplikací.

RIA technologie tedy úzce souvisejí s internetem a pro jejich aplikace je typické: snadné spouštění bez nutnosti instalace nebo se snadnou instalací, interaktivní rozhraní, rychlá odezva. Ve většině případů je potřeba mít v prohlížeči instalovaný specifický zásuvný modul pro spouštění těchto aplikací.

Mezi hlavní zástupce RIA technologií patří:

- Adobe Flex, JavaFX, MS Silverlight
- Ajax – k vytvoření interaktivního rozhraní využívá běžných webových technologií, nepotřebuje vlastní prostředí k interpretaci programů

Na rozdíl od Adobe Flash je Flex zaměřený programátorsky. Jeho významným rysem a předpokladem interaktivity je událostmi řízený chod programů. Interakce uživatele společně s událostmi přicházejícími z okolního prostředí jsou zachytávány a vyhodnocovány, dále pak zpracovány na pokyny k vykonání jim přiřazené odezvy (obslužného podprogramu). Příznivý dopad to má také na rychlost – například po odeslání nějakého požadavku serveru může program vykonávat další činnost, příchodu odpovědi je definována událost s její obsluhou.

Základem je Flex SDK (Software developers kit) skládající se z objektových knihoven a command line kompilátoru. Od r. 2007 je open-source projektem, který spadá pod Mozilla Public Licence. Licence umožňuje upravovat a rozšiřovat jeho zdrojový kód, distribuovat dále komponenty (nebo celé SDK) s popisem provedených změn. Zdrojový kód vytvořené

aplikace je však vlastnictvím jejího autora. Jako dobrá volba se jeví použití vývojového prostředí Flash Builderu (Flex Builder 3 - jeho předchozí verze). Jedná se o komerční grafické prostředí firmy Adobe Systems, nabízí mnoho nástrojů usnadňujících práci. Je k dispozici také ve verzi pro studenty, tato není určená ke komerčnímu využití a je zdarma po předložení dokladu o studiu. Mezi volně dostupnými prostředími Flash Develop.

1.2 Stručně k historii Flex:

Ve své první verzi se objevil Flex pod značkou společnosti Macromedia v roce 2004. Tato verze obsahovala vývojový kit, vývojové prostředí a Flex Data Services (nástroj zprostředkování výměny dat založený na platformě J2EE). Zdrojový kód aplikace byl uložen na aplikačním serveru. Na požadavek uživatele byl zkompilován a doručen prohlížeči s Flash Playerem.

V roce 2005 Adobe Systems odkoupila společnost Macromedia a převzala vývoj flashové platformy. Flex 2 již nese značku značkou Adobe, byl vydán v roce 2006. Změnilo se vývojové prostředí – Flex Bulider 2 je postaven na volně šiřitelném vývojovém prostředí Eclipse, tak jak je tomu v současnosti. Flex 2 SDK byl uvolněn zdarma ke stažení. Výměna dat přestala být vázána na použití Flex Data Services. Action Script vyšel ve třetí verzi.

Flex 3 SDK byl uvolněn pod licencí MPL v r. 2007, nástroj Flex Data Services se oddělil jako samostatný produkt LiveCycle Data Services. Vstupuje do popředí Adobe Integrated Runtime – běhové prostředí, dříve Apollo.

Flex 4 vydaný v roce 2010 prošel změnami, které vedly ke zlepšení výkonu jeho aplikací a zjednodušení tvorby vzhledu. Flex SDK 4 obsahuje nové a přepracované komponenty, vývojové prostředí nese označení Flash Builder.

1.3 Flex, Flash

Způsob, jakým jsou vytvářeny aplikace pro Flash Player, je rozdílný, i když jsou ve výsledku soubory stejného formátu.

Flex je vhodnější pokud jsou kladeny vyšší požadavky na:

- úroveň interaktivity s uživatelem
- výměnu dat s aplikačními servery jako ColdFusion, ASP.NET, PHP nebo J2EE

Flash se více hodí pro:

- vizuální prezentace

- zobrazení videa šířeného přes internet

Flash je především určen pro prezentaci animovaného obsahu, protože využívá časové linky k řízení běhu programu. Podporuje řadu animačních technik, animace mohou být také programovány čistě v Action skriptu, ale tuto možnost nabízí i Flex.

Rozdíly mezi Flex a Flash z pohledu na vývoj aplikací [2]:

Flex	**Flash**
Animace	
Flex používá Action skript třídy zvané efekty k přehrání animace.	Časová osa ve Flash umožňuje animovat obrázek po obrázku nebo vytvářet přechody mezi klíčovými snímky (tweening).
Práce s daty	
Flex má několik nástrojů pro práci s daty a aplikačními servery, RPC komponenty (HTTPService, WebService a RemoteObject), lze ho také použít s LiveCycle Data services.	Flash může komunikovat se stejnými RPC zdroji jako Flex, jeho programátorské nástroje nejsou však na takové úrovni.
Design	
Flex umožňuje vyvářet vzhled aplikace v okně WYSIWYG editoru, nemá však žádné jiné nástroje pro tvorbu grafiky.	Flash má dobré nástroje pro práci s grafikou při vytváření vzhledu, ačkoli nejsou tak uceleným souborem jako je tomu v Illustratoru. Nicméně velmi dobře dokáže importovat grafiku vytvořenou v Photoshopu a Illustratoru.
Programovací jazyky	
Podporuje ActionScript 3 a MXML	Podporuje všechny verze Action skriptu, i když jen jednu v každém flashovém dokumentu a nepodporuje MXML.
Správa zdrojového kódu	
Flexové aplikace jsou tvořeny ze zdrojového kódu v textových souborech,	Flashové dokumenty jsou v binárním tvaru, co může představovat problém v prostředí

jejich zápis je tak plně podporován systémy správy zdrojového kódu.	vývojových týmů, které vyžadují nástroje pro správu zdrojového kódu .

2 PŘEHLED PROSTŘEDKŮ

2.1 Programovací jazyky

Aplikace jsou psány s využitím dvou programovacích jazyků – ActionScriptu a MXML:

- ActionScript 3 (zkráceně AS3) je současná verze programovacího jazyka, který se rozvíjel společně s nástroji pro tvorbu Flashe. Je plně objektově orientovaný.
- MXML je jazyk vycházející z XML, je použit při definování Flex aplikace a mnoha jejích komponent.

Při kompilaci je MXML kód přepsán do ActionScriptu, MXML je vlastně jazykem který usnadňuje a urychluje psaní programů v AS3. MXML se obvykle využívá při zapisování vizuálních prvků, záleží však do určité míry na programátorovi, který ze způsobů zápisu zvolí.

Jako dobrý příklad poslouží deklarace instance třídy Label. Třída Label je součástí knihovny tříd z Flex SDK, je určená pro zobrazení jednoho řádku textu. V MXML může být třída Label zapsána značkou <mx:Label /> takto:

```
<mx:Label id=˝label˝ text=˝Zobrazený text˝/>
```

V AS3 je potřeba nejdříve vytvořit objekt pomocí konstruktoru třídy a následně jej připojit k zobrazeným prvkům aplikace.

```
var label: Label = new Label();
label.text = ´Zobrazený text´;
this.addChild(label);
```

ActionScript nabízí širokou podporu formátu XML. Obsahuje nástroje EcmaScript for XML(E4X) založené na standardu ECMA-357 (Ecma International). Používá jednoduchý zápis volání E4X API pro přepis, získání a úpravy dat[2].

2.2 Práce s daty

Velká výhoda frameworku Flex oproti klasickým webovým aplikacím je v trvalém uchování dat do doby než uživatel ukončí činnost v programu. Data mohou přicházet z mnoha zdrojů: XML soubory, databáze nebo další prostředky na straně serveru. Podobně je tomu s jejich formou. Po přijetí jsou uchována v objektu určitého datového modelu. Instance objektu jedné datové entity tak může představovat řádek databázové tabulky nebo

prvek XML souboru. Je to způsob reprezentace dat, u kterých nemusí být předem definovaná struktura[2].

Obecným datovým modelem pro netypová data je <mx:Model>. Při zpracování údajů může být implementován jako dynamický datový objekt:

```
<mx:Model id=˝kontakty˝>
  <data>
    <id>{zdroj.id}</id>
    <jmeno>{zdroj.jmeno}</jmeno>
    …
  </data>
</mx:Model>
```

2.2.1 Vázání k datovým zdrojům

Ke zprostředkování dat mezi jednotlivými vrstvami aplikace na straně klienta slouží vázání na datové zdroje (data binding). Data z vlastnosti jednoho objektu jsou automaticky kopírována do vlastnosti jiného, následkem události vyvolané změnou zdrojového objektu. Je možné určit směr, kterým mají data být propojena, zda se změna zdrojových dat projeví v připojeném objektu nebo naopak.

2.3 Integrace se serverovým prostředím

Zpočátku byl vývoj ve Flex frameworku vázaný na nasazení aplikačního serveru. Flex Server, původní produkt společnosti Macromedia, převzal technologie vzdálených služeb z ColdFusion, ty mu byly přizpůsobeny pro práci s objekty Javy [2]. V současné době se nabízí více způsobů, jakými lze Flex propojit s informačními systémy různých platforem.

Metody přístupu:

- HTTP POST nebo GET prostřednictvím komponenty HTTPService
- SOAP s použitím komponenty WebService
- použitím AMF (Adobe Action Mesage Format) volání vzdálené služby a komponenty RemoteObject

SOAP (simple object access protokol) je základem systémů navržených jako webové služby (tyto systémy jsou označovány jako SOA - Service Oriented Architecture). Principem je komunikace pomocí zpráv XML, k jejichž přenosu se zpravidla využívá protokolu HTTP. K popisu služby slouží WSDL (web services description language), XML formát dat s informacemi o metodách, parametrech, apod.

Výhodou SOAP je nezávislost na jazycích používaných na straně serveru. Podstatné je, aby tyto systémy uměly zpracovat zprávy tohoto formátu. Podporu má širokou: v ASP.NET jako XML Web Services, Apache AXIS jej implementovalo pro klientské a serverové aplikace Javy, Adobe ColdFusion umožňuje volání funkcí ColdFusion komponenty (CFC) z většiny SOAP serverů, je dostupný skriptovacím jazykům jako je PHP, Python, Ruby [2].

AMF je binární formát, s jehož pomocí jsou přenášeny objekty, pole objektů, grafická data a třeba i vlastní datové struktury. Je komprimovaný, navržený pro úsporný přenos dat. Jeho implementace existuje pro většinu jazyků používaných na serveru.

2.3.1 J2EE

Informační systémy navržené na platformě Java bývají řešení směřující do firem, často zákaznicky orientovaná, která využívají služeb práce s dokumenty (tvorba PDF souborů apod.), formuláři, databázemi. Jedná se o infrastruktury zaměřené na výměnu zpráv v heterogenním prostředí označované MOM (Message-oriented middleware). Snižují složitost vývoje mezi více operačními systémy a síťovými protokoly, používají API pro překonání jejich odlišností. Komunikace obvykle probíhá asynchronně.

Patří sem Adobe LiveCycle Data Services, BlazeDS (open-source alternativa k LCDS) a ColdFusion.

2.3.2 ASP.NET

Využití ASP.NET jako middleware při vývoji ve frameworku Flex může mít své výhody:

- implementace služeb založených na SOAP
- dosažení nejlepší integrace s operačním systémem MS Windows a MS SQL Serverem
- výběr programovacích jazyků

U .NET komunity lze navíc najít velké množství komponent usnadňujících integraci s touto platformou. Jsou k dispozici volně nebo jako placené.

2.3.3 PHP

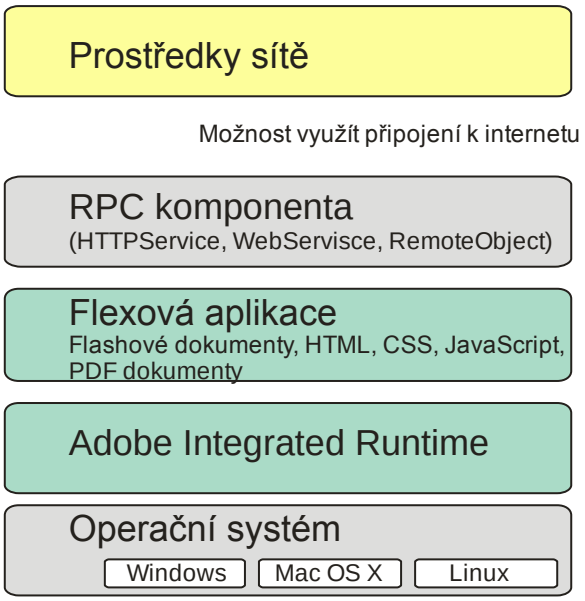
PHP je velmi rozšířeným skriptovacím jazykem používaným na serverech různých platforem. Po otevření specifikace AMF (Action message format) je kromě běžného

způsobu komunikace možné použít také tohoto formátu při volání vzdáleného objektu. Je k tomu využívána součást frameworku Zend - Zend_AMF.

2.4 Integrace s klientským počítačem - AIR

Flashová platforma umožňuje spouštět aplikace samostatně a částečně jim zpřístupňuje některé prostředky systému díky Adobe Integrated Runtime (AIR). Ke spouštění AIR aplikací je třeba mít v klientském systému instalovanou runtime knihovnu.

Jeho základem je jádro prohlížeče WebKit které používá také Safari nebo Google Chrome. WebKit zobrazuje HTML stránky na počítačích s různými operačními systémy dost podobně a vynikající rychlost společně s nízkými paměťovými nároky jej činí ideálním pro zamýšlené využití v mobilních zařízeních. Druhou jeho podstatnou součástí je JavaScriptový interpret. Do JavaScriptu je přidán jmenný prostor (objekt) air, ten obsahuje funkce knihoven pro práci se soubory, okny, apod. Pro uživatelská data je v AIR integrována databáze SQLite [15].



Obr. 1. Uspořádání vrstev AIR architektury

2.5 Vývojové prostředí

Ačkoli lze aplikace pro web vyvíjet i jen s použitím Flex SDK, pro rozsáhlejší projekty se více hodí programátorský nástroj Flash Builder který nabízí pokročilé programátorské

nástroje pro usnadnění psaní kódu, vizuální návrh uživatelského rozhraní, ladění. Hlavními výhodami Flash Builderu jsou:

- podpora různých platforem na straně serveru (využití nachází při propojování aplikací s datovými službami)
- integrovaný debugger

Při ladění s tímto nástrojem odpadá nutnost trasování zapsaného v přímo v kódu programu. V průběhu vykonávání kódu aplikace spuštěné z lokálního nebo i vzdáleného umístění lze sledovat z perspektivy debuggeru hodnoty proměnných a hlášení upozornění nebo chyb.

3 DALŠÍ TECHNOLOGIE

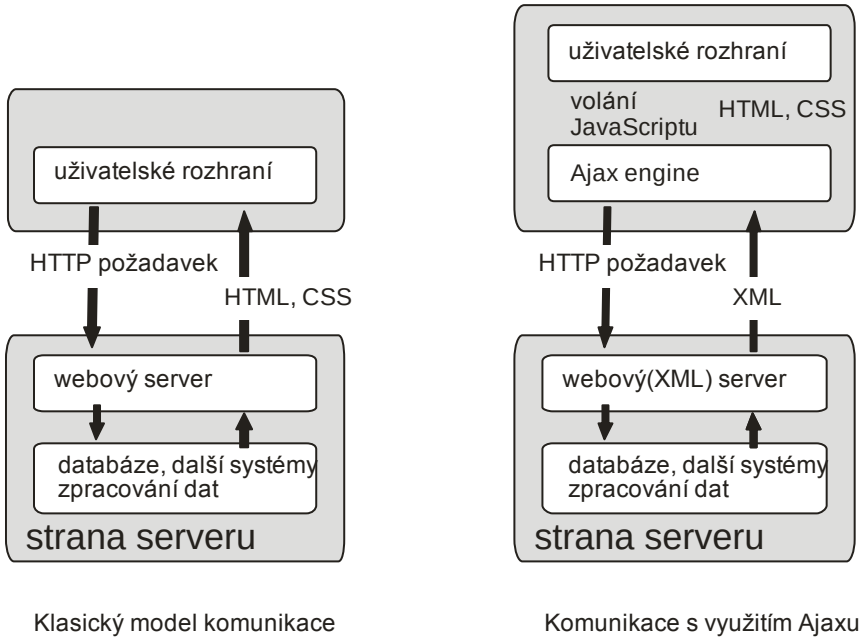
Klasické webové aplikace založené na HTML, PHP, ASP, ASP.NET mají samy o sobě omezené možnosti tvorby interaktivního rozhraní. Hlavním jejich omezením je synchronní způsob komunikace, při kterém je se změnou stavu nutné odeslat požadavek na server a čekat na odezvu než se může zobrazit nový obsah.

V minulosti se objevovala různá řešení, zpočátku však často měla problémy s kompatibilitou v konkurenčním prostředí (např. technologie iframe, definující v dokumentu oblast kterou je možné překreslovat nezávisle). Přenesení části aplikační logiky na stranu klienta umožnil JavaScript (původně LiveScript) společně s DOM v podobě DHTML. Toto spojení je jednou ze základních součástí koncepce Ajax. Jiné technologie jsou založené na specifickém běhovém prostředí a podmíněné instalací doplňků prohlížeče či frameworku.

3.1 Ajax

Ajax (Asynchronous JavaScript and XML) je označením pro spojení několika technologií použitých k dosažení záměru vytvoření interaktivního uživatelského prostředí. Jeho charakteristickým rysem je asynchronní komunikace se serverovým prostředím probíhající na pozadí aplikace, při které jsou zpracovány jen potřebné části dokumentu. Využívá HTML, XHTML a CSS pro prezentaci obsahu, DOM umožňující provádět změny prezentovaného obsahu, JavaScriptu k vytvoření aplikační logiky a na straně klienta. K výměně informací slouží formát XML nebo JSON.

Poprvé se tento pojem objevil v roce 2005 v článku Jesse James Garreta: A New Approach to Web Applications (Ajax: Nový přístup k webovým aplikacím) [13]. Garrett v něm píše o Ajaxu jako o jiném modelu využití stávajících technologií. Definoval v něm „Ajax engine“ – jde o aplikační vrstvu zprostředkující výměnu dat mezi uživatelskou částí a serverem.



Obr. 2. Pohled na schémata komunikace

Lze se s jeho využitím setkat poměrně často, aniž by si návštěvník stránek tuto skutečnost uvědomoval. K jeho popularizaci přispěla společnost Google, která ho využívá například v Gmail, Google Maps nebo Google Suggest.

3.1.1 Prostředky Ajaxu

DOM (Document Object Model)

Objektový model dokumentu je objektově orientovaná reprezentace XML nebo HTML dokumentu. DOM je API, umožňuje přístup a úpravy dokumentu nebo jeho částí. Původně měl každý prohlížeč vlastní rozhraní pro manipulaci s prvky dokumentu pomocí JavaScriptu, později bylo sjednoceno a standardizováno konsorciem W3C. Ke struktuře dokumentu přistupuje jako ke stromu, je zřejmá podoba s XML.

XMLHttpRequest (XHR)

Objekt XMLHttpRequest je rozhraní umožňující komunikaci mezi klientem a serverem. Komunikace probíhá obvykle asynchronně přes protokol HTTP. Vychází z technologie Remote Scripting představené Microsoftem v roce 1998, jednalo se o Javový aplet komunikující se serverem, tento aplet poskytoval služby JavaScriptovým funkcím. Stal se součástí páté verze IE a také byl použit pro rozhraní přístupu k e-mailu [19]. Prohlížeč Mozilla jej integroval se stejným voláním jeho metod a vlastností.

Před začátkem komunikace je tento objekt potřeba vytvořit pomocí JavaScriptu a inicializovat. Při iniciaci lze zvolit metodu odeslání (GET nebo POST), uvést zda má být komunikace asynchronní, dále uvést adresu kam požadavek směřuje a údaje pro autentizaci. Stav vyřízení požadavku lze sledovat a případně jej i zrušit:

- **0** (*UNINITIALIZED*) - objekt XMLHttpRequest byl vytvořen, ale ještě ne inicializován metodou *open()*
- **1** (*LOADING*) – před odesláním HTTP požadavku
- **2** (*LOADED*) - metoda *send()* již byla volána, přijaty HTTP hlavičky, nikoliv však tělo odpovědi
- **3** (*INTERACTIVE*) - část dat již byla přijata, vlastnost *responseText* obsahuje získaná nekompletní data
- **4** (*COMPLETED*) - všechna data odpovědi byla přijata, všechny operace byly dokončeny [19]

3.1.2 Programovací jazyk

JavaScript

Při programování Ajaxových aplikací je zapotřebí kromě znalosti HTML a CSS také znalost JavaScriptu. Tento jazyk, jehož vznik je spjatý se společností Netscape, je multiplatformní objektově orientovaný skriptovací jazyk. Využití našel především jako interpretovaný programovací jazyk pro WWW stránky, kde jsou jím obvykle ovládány interaktivní prvky, efekty. Jeho syntaxe vychází z jazyka C, patří tedy do rodiny jazyků C/C++/Java, více však s Javou společného nemá. JavaScript byl v červenci 1997 standardizován asociací ECMA a v srpnu 1998 ISO. Standardizovaná forma se označuje jako ECMAScript, z ní byly odvozeny další implementace jako JScript (v prohlížečích Internet Explorer), VBScript (pro skriptování HTML pomocí Visual Basicu) nebo ActionScript [18]. Používá se také při skriptování dalších aplikací jako je například Adobe Acrobat.

U JavaScriptu se setkáme s dynamickým přiřazováním typů proměnných - podobně jako je tomu u jiných skriptovacích jazyků, typy jsou asociovány s hodnotami. Stejná proměnná může být jednou celočíselným typem, podruhé řetězcem. Má rysy objektově orientovaného jazyka, klasickému konceptu tříd se však vymyká. Jeho objekty jsou asociativními poli rozšířenými o tzv. prototypy. Přes tyto prototypy lze vytvořit odkaz na nadřazený objekt

s jeho vlastnostmi a metodami – prototypová dědičnost. Takto vytvořený objekt zděděné vlastnosti a metody neobsahuje, avšak může se na ně odkazovat. Funkce mají kromě své běžné role význam jako konstruktory objektů. Mezi zápisem funkce a metody není rozdíl, s použitím ukazatele `this` mohou být funkce volány jako metody objektu.

3.1.3 Vývoj aplikací

K vývoji pro platformu Ajax stačí jakýkoliv editor s podporou JavaScriptu. Výrazným usnadněním je využití nějakého JavaScriptového frameworku (Google Toolkit, Echo 2, Rico). Může pomoci odstranit problémy s implementací Ajaxu v různých prostředích a také nabídnout postupy při vytváření aplikací orientovaných na určitou serverovou platformu.

V závěru jsou pak především důležité nástroje pro ladění. Je jich celá řada, ladění je však třeba nejen z hlediska funkčnosti ale i výkonu.

- **JSLint** – praktický nástroj pro kontrolu syntaxe a analýzu JavaScritového kódu, ještě před jeho spuštěním. Pomůže odstranit mnoho chyb před tím, než se projeví.
- **Microsoft skript debugger** – doplněk IE, je určený pro ladění skriptů běžících u klienta i skriptů které běží pod Internet Information Serverem (IIS).
- **Firebug** – Doplněk do prohlížeče Mozilla FireFox určený vývojářům pro ladění internetových a intranetových aplikací. Umožňuje zobrazit, upravit a ladit CSS, HTML a JavaScriptové kódy. Umožňuje trasovat XMLHttpRequest. Je dispozici také pro jiné prohlížeče [4].

3.1.4 Pohled na využití technologie

Ajax má širokou podporu mezi internetovými prohlížeči, nepotřebuje instalovat další doplňky. Problém může nastat s určitými odlišnostmi v interpretaci jeho kódu, lze však vytvořit knihovnu pro rozdílné prohlížeče a operační systémy. Zdrojový kód aplikací je distribuován společně s dokumentem HTML, většinou v samostatném souboru nebo je do dokumentu zapsán přímo. Uplatnění nachází především při realizaci méně náročných požadavků. Jeho výkon je omezen možnostmi JavaScriptu, při výpočetních operacích není příliš vysoký. S rostoucí členitostí uživatelského rozhraní vznikají potíže s přehledností zápisu kódu. Pro zpracování komplexnějšího řešení se jeví výhodné propojení s některou z dalších technologií.

3.1.5 Závěr a srovnání s Flex

Ajax a Flex nejsou tak úplně konkurenční technologie. V mnoha ohledech se liší, mohou však v některých případech nabízet podobná řešení. Při rozhodování zda použít jednu nebo druhou je zapotřebí znát jejich výhody či nevýhody. Uvedu zde srovnání důležitých rysů těchto frameworků. Navíc protože Flex patří společně s Flash do Flashové platformy – používají stejné běhové prostředí Flash Player, neliší se v mnohých srovnávaných vlastnostech.

Využití HTML, CSS a DOM

Ajax pro běh aplikací využívá pouze internetového prohlížeče a využití prostředků HTML, CSS a DOM je pro něj typické a přirozené. Flash Player je třeba doinstalovat jako doplněk, na podporovaných platformách však zobrazuje obsah shodně. HTML využívá omezeně (bez tabulek apod.), CSS ke stylování komponent také. Přístup k objektovému modelu dokumentu nemá, může jej realizovat přes volání JavaScriptu.

Vlastnosti programovacího jazyka z hlediska typové kontroly a podpory objektově orientovaného programování (OOP). Jazyk může být slabě nebo silně typovaný, silně typovaný provádí kontrolu datového typu a umí zjistit jeho chybné použití. OOP usnadňuje vytváření základních částí a programových vzorů.

JavaScript není klasickým objektově orientovaným jazykem, některé jeho frameworky OOP podporují, stále se však jedná o objekty se strukturou prototypů. ActionScript 3 Flashové platformy je rozšířen o přirozenou podporu XML (je E4X kompatibilní, definován standardem ECMA-357 pro XML). Má snadnější zápis a typickou objektovou strukturu.

Bitmapové operace umožňují provádět změny obrázků na straně klienta, poskytují prostředky pro vytváření efektů a přechodů za běhu programu.

Vektorová grafika – schopnost pracovat s vektorovou grafikou. Při využití pro tvorbu designu výrazně snižuje velikost aplikace.

Audio/video streaming – prostředky kterými disponuje pro přenos multimediálního obsahu.

V oblasti grafických a mediálních formátů má Ajax stále co dohánět. Operace s bitmapami mají nejednotnou podporu v prohlížečích Opera, Firefox a Safari, nejčastěji jsou zpracovány na straně serveru. Omezeně podporuje jednoduché lineární animace.

Z vektorových formátů může využít VML nebo SVG v IE9 a ostatních prohlížečích. Pro multimédia vyžaduje instalovat doplněk (QuickTime, Windows Media Player, Flash Player).

Bezpečnost

Obě technologie využívají k zajištění bezpečnosti mechanismus oddělení běžícího procesu Sandbox. Zatímco kód Ajaxu je snadněji přístupný ke zjištění bezpečnostních nedostatků, soubory pro Flash Player jsou kompilované a mohou být i šifrované.

Komunikace se serverem

Ajax i Flex mají přirozenou podporu komunikace s webovým serverem nebo službou, Flash Player navíc nabízí možnost využít socketů k vytvoření spojení pro obousměrnou komunikaci (PUSH a PULL).

SEO (Search engine optimization)

Optimalizaci pro vyhledávací stroje HTML umožňuje, problémem bývají JavaScriptové odkazy. Pokud je při programování v Ajaxu kladen důraz na SEO, je třeba tomu přizpůsobit zápis kódu. Flex ji podporuje velmi omezeně, pouze v meta tagu a tagu pro alternativní obsah v HTML stránce kam je vložený.

3.2 Silverlight

Silverlight je výsledkem snahy společnosti Microsoft prosadit se také v odvětví RIA technologií. Spojuje několik technologií do jedné vývojové platformy, která nabízí volbu mezi více programovacími jazyky a nástroji. Jeho tvůrci vycházeli z redukované verze frameworku .NET. Pro tvorbu uživatelského rozhraní používá deklarativního jazyka XAML. Skrze instalovaný doplněk ho podporují prohlížeče Internet Explorer, Firefox, Safari, Opera a Google Chrome na operačních systémech Windows a Mac OS X. Vývoj jeho podpory v operačním systému Linux svěřil Microsoft společnosti Novell, která ji vyvíjí pod názvem Moonlight. V současné 4. verzi otevírá cestu vývoji aplikací, které mohou běžet samostatně, mimo prohlížeč, podobně jako je tomu u Adobe AIR.

Ve verzi 1.0 byl programovacím jazykem JavaScript, podporoval práci s textem, 2D grafikou a audio/video formáty, komunikace probíhala s pomocí Ajaxu. JavaScript byl spouštěn v rámci .NET frameworku, vedlo to ke zvýšení výkonu oproti jeho vykonávání

prohlížečem. Od druhé verze už umožňuje použít další programovací jazyky jako je C#, Visual Basic, IronPython.

3.2.1 Na čem Silwerlight staví

Windows Presentation Foundation (WPF) je podmnožinou frameworku .NET, obsahuje technologie určené pro vykreslování grafického uživatelského rozhraní. Jednotlivé grafické prvky jde definovat pomocí jazyka XAML. WPF je součástí Windows Vista a Windows 7.

Windows Communication Foundation (WCF) rozhraní z frameworku .NET pro komunikaci se vzdálenými službami.

3.2.2 Závěr a srovnání s Flex

Technologie nabízí velké množství prostředků k vytvoření moderní aplikace s atraktivním vzhledem. Nachází podobné využití jako více rozšířená Flashová platforma. Nemá velký význam srovnávat co ve výsledku dokáží, obě technologie jsou stále vylepšovány a schopnost jedné z nich za čas může mít i druhá. Také Silverlight se snaží o sjednocení vývoje pro internet a desktop prostřednictvím OOB (Out of browser). Staví na různých základech, i z pohledu programátora jsou rozdílné. Flex se v určitých rysech se podobá klasickým webovým technologiím, konkrétně v zápisu jeho objektů pomocí MXML, úpravách vzhledu pomocí CSS (i když se nejedná o CSS implementované v HTML) a společném jazykovém základu ActionScriptu s JavaScriptem. Silverlight oproti tomu vychází z platformy .NET používané pro vývoj programů na plochu počítače, stejně tak XAML používají desktopové aplikace (ve Windows). Tento způsob, třeba s použitím jazyka C++, může být někomu bližší.

3.3 JavaFX

JavaFX je poměrně mladou technologií navrženou pro tvorbu interaktivních, multiplatformních aplikací označovaných často pojmem RIA. Společnost Sun Microsystems ji představila v roce 2008. JavaFX vychází z Javy, má nabídnout snadný vývoj, stabilitu a větší možnosti tvorby funkčního i atraktivního rozhraní aplikací určených pro internet, plochu počítače nebo mobilní zařízení. Podobně jako je tomu s Flex SDK, i Sun uvolnil klíčové části své technologie, kompilátor společně s částí grafických knihoven a nástrojů spadá pod licenci GPL 2.0.

3.3.1 Prostředky platformy JavaFX

JavaFX používá k vývoji vlastního skriptovacího jazyka JavaFX Script, se syntaxí podobnou JavaScriptu. Jde o statický deklarativní skriptovací jazyk založený na platformě Java. Tato skutečnost umožňuje použít velké množství jejích již vytvořených knihoven.

K vývoji je potřeba JavaFX SDK, který obsahuje knihovny, kompiler, běhové prostředí, emulátor mobilní platformy a dokumentaci. Jako vývojové prostředí společnost Sun poskytuje Netbeans for JavaFX které již SDK obsahuje. Nadstavbou je JavaFX Production Suite obsahující nástroje pro práci s grafickými materiály a spolupráci s dalšími prostředími při tvorbě designu designu jako jsou Adobe Illustrator a Adobe Photoshop..

3.3.2 Závěr a srovnání s Flex

Největší potenciál této platformy spatřuji ve využití v mobilních zařízeních. Nasvědčuje tomu rozšíření jejího předchůdce Javy, která se stala součásti většiny mobilních telefonů. JavaFX má předpoklady blíže propojit mobilní zařízení s počítačem, internetem i multimédii. Její využití na webu nebo ploše počítače v současné době nemá takovou uživatelskou základnu, Flashová platforma má velký náskok. V porovnání s Flexem nabízí jednu zajímavou inovaci, ta spočívá v běhu aplikace jako samostatného procesu, byť je spuštěná z prohlížeče. Umožňuje to instalovat ji do počítače jednoduše vytažením a umístěním na plochu. Je možné, že díky odvětví mobilní komunikace JavaFX získá lepší pozice i zde.

II. PRAKTICKÁ ČÁST

4 NÁVRH A REALIZACE PROJEKTU

V praktické části jsem se věnoval návrhu a programování webové aplikace s využitím běžných technologií společně s frameworkem Flex. Konkrétně se jedná o jednoduchý redakční systém internetové poradny doplněný o některé další aplikace, jehož základ tvoří PHP a databáze MySQL.

Návrh systému lze obvykle rozdělit do tří fází:

- systémová analýza
- systémový návrh
- technický návrh

Systémová analýza stanovuje požadavky na připravovaný systém. Na jejím základě je specifikován systémový návrh. Po zpracování specifikace systému je vytvořen technický návrh. Ten obsahuje hardwarové s softwarové systémové požadavky.

Proces systémové analýzy a systémového návrhu je často nazýván konceptuální model. Takový model popisuje data v databázi zcela nezávisle na jejich fyzickém umístění. Při jeho návrhu (tento proces je označován jako konceptuální modelování) se pozornost zaměřuje na aplikační logiku – avšak z pohledu člověka, nikoli z pohledu později použitých hardwarových a softwarových technologií. Při tvorbě konceptuálních návrhů jsou zpravidla vnímány objekty reálného světa, vztahy mezi nimi, jakož i funkce, jejichž prostřednictvím jsou tyto vztahy realizovány. Konceptuální modelování je tedy v podstatě objektově orientovaný proces. Kromě objektů vstupuje do hry rovněž hierarchické uspořádání – například dědičnost. To znamená, že objekty mohou být vytvořeny na základě jiných objektů, přičemž v tomto procesu zdědí část charakteristických rysů [5].

4.1 Co je redakční systém

Redakční systém (RS) je nástroj pro tvorbu internetových stránek bez potřeby programátorských znalostí. Spravovat obsah může oprávněný uživatel z jakéhokoliv místa s připojením k internetu i bez účasti odborníka na informační systémy. Často je označován také jako publikační systém nebo CMS (Content Management System).

Nejčastěji je využíván:

- zpravodajskými servery
- informačními severy státních i soukromých organizací (zpravidla kombinovaný s prezentací)
- komunitními servery (doplněný o diskuzní fórum, galerii)

Existuje mnoho systémů s podobnou základní funkčností, liší se spíše v pokročilých možnostech konfigurace nebo ve vhodnosti k nasazení pro jednotlivé typy využití. K dispozici jsou zdarma nebo placené, ke kterým je zpravidla poskytována technická podpora.

Základní funkce a vlastnosti redakčního systému:

- systém správy uživatelů, použití přístupových práv
- systém pro vkládání a úpravy textu
- konfigurace obsahových částí
- správa souborů
- podsystémy komentářů, novinek, anket apod.
- rozšiřitelnost o další součásti

4.2 Konceptuální model

Před programováním rozsáhlejšího projektu je třeba udělat koncepční návrh řešení. Znamená to vzít v úvahu účel, jakému má sloužit a požadavky, které jsou na něj kladeny.

Internetové stránky mají sloužit k prezentaci aktivit soukromého subjektu a rozšíření jeho činnosti i na prostor internetu.

Mají obsahovat:

- stránku s informacemi o nabízených službách, s možností objednání/přihlášky v případě zájmu,
- publikační systém pro prezentaci literární tvorby,
- část pro další témata a diskuze,
- galerii výrobků určených k prodeji s možností objednat je přes internet,
- aplikace lunárního kalendáře a výkladu karet.

Požadavky kladené na jeho vlastnosti:

- být přístupný širokému spektru uživatelů, s jednoduchým, interaktivním uživatelským rozhraním,
- umožňovat komunikaci prostřednictvím elektronické pošty,
- používat správu přístupových práv uživatelů a vhodnou úroveň zabezpečení před neoprávněným přístupem.

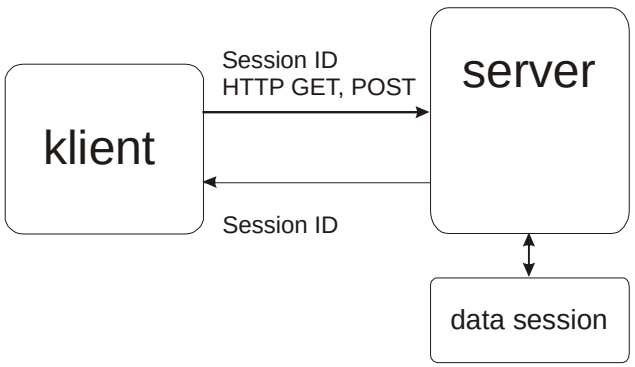
4.3 Logický a fyzický model řešení

Na rozdíl od konceptuálního modelu, kdy vlastní implementace zůstala stranou, v této části už je tomu jinak. Je třeba se zamyslet a udělat si představu jak má vypadat provoz a správa takové součásti informačního systému. Může k tomu dobře posloužit vytvoření prototypu, jejich základní podoby.

Stránky budou generovány dynamicky na základě požadavků uživatele. Důležitou úlohu proto má databáze. Její vhodně navržená struktura je základem při manipulaci s daty, kladně se projeví nejen optimální rychlostí vyhledávání, ale především už při psaní skriptů. Využita bude k uložení: obsahu stránek, informací o lidech přispívajících v publikační části, údajů k plánovaným akcím v kalendáři a zájemcích, dat aplikace pro prodej výrobků, dále pro data diskusí.

4.4 Realizace projektu

V prvních krocích jsem vytvořil základní návrh úvodní stránky a stránky rozhraní správy, kterému postupem času přibývaly položky jeho funkcí nebo nástrojů. Pro zabezpečení před neoprávněným přístupem jsem zvolil autentizaci s využitím PHP session bez zabezpečeného přenosu. Jedná se o stránky, které pokládají uvedené opatření za dostatečné. Uživatel se přihlásí do rozhraní správy pomocí hesla a jména, při vstupech na další stránky se jeho oprávnění ověřuje pomocí identifikace v session. Tohoto mechanizmu, při kterém se s přihlášením uživatele vyhradí na serveru určitý prostor jeho činnosti, jsem využil také později při publikování, k ukládání informací které mají být odeslány databázi společně s textem upravovaným v klientské části.



Obr. 3. Identifikace uživatele v PHP session.

Pro odesílání zpráv prostřednictvím elektronické pošty jsem zvolil volně dostupnou třídu phpmailer, především z důvodu častého omezování funkce mail() na straně serveru různých hostingových služeb. Phpmailer využívá autentizovaného přístupu k SMTP serveru, přihlašovací údaje získává ze souboru pristup.ini v kořenové složce adresáře stránek. V tomto souboru jsou zapsány i informace pro přístup do databáze.

Rozhraní správy

Systém je navržen pro menší okruh publikujících. Po přihlášení do rozhraní správy se v levém panelu zobrazí nabídka, která je výchozí při vkládání a úpravách příspěvků, zobrazení přehledu, úpravách osobních informací apod.

Rozlišuje dva typy oprávnění:

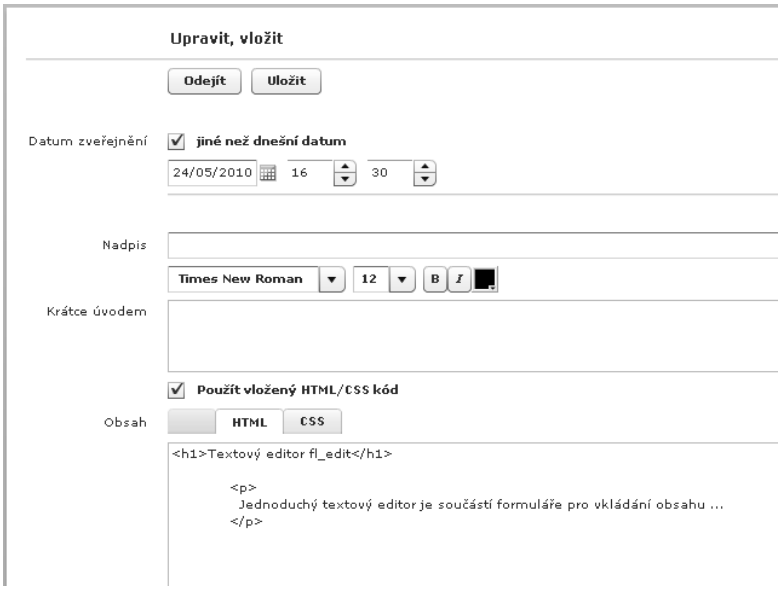
- uživatel

  Jeho články a témata budou zveřejněny po schválení uživatelem s oprávněním spravovat obsah stránek (redaktorem).

- uživatel s oprávněním administrátora (správce a redaktor)

  Správce může zakládat kategorie s různými atributy (zda do nich může vkládat kdokoliv, zda se mají zobrazovat ve společné části nebo třeba v jiném oddílu stránek). Má přístup k dalším funkcím pro organizaci prezentovaných aktivit a vkládání fotografií do galerie.

Úvahy nad úpravou prezentovaného textu vedly spíše k jeho jednotné podobě. Použil jsem jednoduchý textový editor vytvořený ve Flex. Je součástí formuláře, do kterého se uvádí také datum zveřejnění, zda má mít diskuzi, případně je přes něj vložen vlastní HTML/CSS kód. Text je vkládán do databáze prostřednictvím PHP skriptu.



Obr. 4. Vkládaní textu v publikační části – formulář s jednoduchým editorem vytvořený ve Flex

Organizátor

Jedná se o nástroj, který má pomoci při organizování aktivit prezentovaných na stránkách. Podrobněji je popsán v následující kapitole „Tvorba aplikace ve Flex“.

Galerie

Pro účely galerie je k dispozici celá řada již vytvořených šablon nebo aplikací. Existuje také velké množství programů, které galerii umí vygenerovat (např. Photoshop elements, Zoner Photo Studio, Picasa a další). Využil jsem šablony pro flashový prohlížeč snímků Simple Viewer, k šabloně jsem připojil skripty a rozhraní potřebné při vkládání fotek a obrázků. Rozhraní je využito jak pro účely dokumentace, tak i pro vkládání informací o vlastních produktech určených k prodeji. Poskytuje základní funkce připojení popisu a případných doplňujících informací k prodeji, dále k organizaci pořadí snímků nebo jejich mazání.

Obr. 5. Rozhraní prokládání snímků do galerie

Lunární kalendář a aplikace výkladu karet

Jsou specifickými požadavky tohoto projektu, vycházejí z jeho povahy. Sestavení přesného výpočtu pohybu těles sluneční soustavy je však složité, používá také empirických hodnot, proto lunární kalendář využívá dat generovaných programem.

4.5 Závěr

Projekt publikačního systému je dokončený, před uvedením do běžného provozu. Tato poslední část jeho vývoje počítá s případnými úpravami nebo odstraněním nedostatků. Pro účely mé bakalářské práce je systém umístěný na stránkách http://www.lukase.eu

Publikovaná data jsou ilustrační. K práci přiložené CD obsahuje zdrojové kódy, Flexové projekty a složku se snímky komponent projektu i z vývojového prostředí. Uživatelskou příručku s instrukcemi k instalaci a používání publikačního systému obsahuje příloha.

5 TVORBA APLIKACE VE FLEX

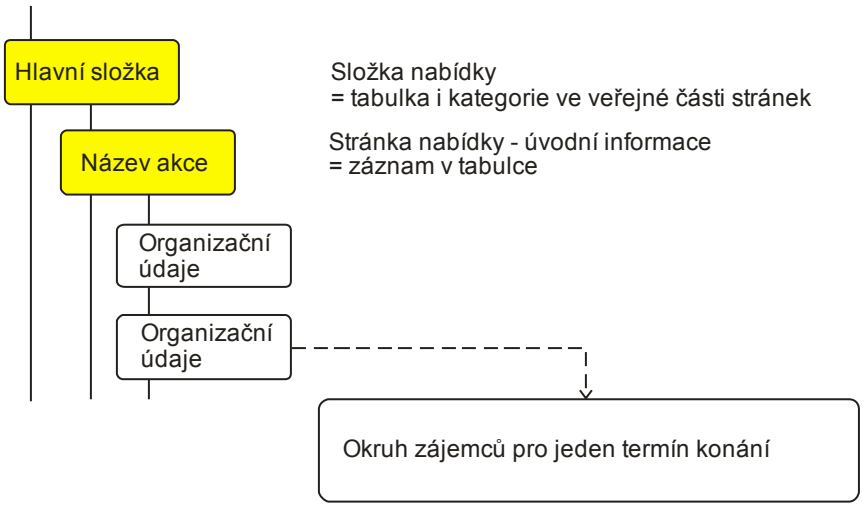
Flex nachází velmi dobré uplatnění v oblasti zpracování dat, poskytuje k tomuto účelu různé datové vizuální komponenty se základní funkčností. Z pohledu objektově orientovaného programování jsou to objekty, které lze rozšiřovat o další vlastnosti (datové, vizuální objekty) nebo metody a vytvářet tak vlastní komponenty.

Způsob práce a některé důležité principy, ze kterých jsem vycházel, ukážu v kontextu s tvorbou součásti realizovaného projektu – uživatelského rozhraní, které má majiteli stránek usnadnit organizaci aktivit.

5.1 Koncepce organizátoru

Rozhraní má poskytovat přehled pořádaných kurzů a jiných akcí, informace o přihlášených zájemcích. Má nabízet funkce úprav a vkládání nových stránek s informacemi, tisku seznamu zájemců a odeslání zpráv prostřednictvím elektronické pošty.

Základní koncept počítá s hierarchickou strukturou záznamů o probíhajících akcích, kterým odpovídají záznamy přihlášených zájemců. K databázi, která je rovněž využívána veřejnou částí, bude přistupováno prostřednictvím PHP.



Obr. 6. Struktura dat v organizátoru, výběr okruhu zájemců

5.2 Realizace

5.2.1 Projekt Flash Builderu

Při práci ve Flash Builderu je se založením nového projektu nutné určit zda se jedná o projekt určený na stránky internetu (nebo intranetu) nebo do prostředí AIR. Dále také, jakým způsobem mají být připojeny ke zkompilovanému souboru potřebné knihovny – zda budou přímo v souboru nebo jako sdílené runtime knihovny (RSL - runtime shared library, u rozsáhlejších projektů může být tato volba výhodná z hlediska velikosti přenášených souborů). Jsou to vlastně jedny ze základních voleb kompilátoru, další jdou ovlivnit ve vlastnostech projektu. Takto tedy vypadá zdrojový kód nové aplikace zapsaný v MXML souboru:

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application    xmlns:fx="http://ns.adobe.com/mxml/2009"
                   xmlns:mx="library://ns.adobe.com/flex/mx"
                   minWidth="955" minHeight="600">
 <fx:Declarations>
       <!-- Place non-visual elements (e.g., services, value
objects) here -->
 </fx:Declarations>
</mx:Application>
```

Základním elementem je <mx:Application/>, v atributech má uvedeny jmenné prostory objektových knihoven.

Abych mohl využívat vlastních komponent, připsal jsem další, s informací o umístění vzhledem ke kořenové složce projektu. Jmenný prostor `xmlns:renderer="renderer"` dovoluje v zápisu kódu deklarovat komponenty ze složky /renderer pomocí značky `<renderer:MáKomponenta/>.`

5.2.2 Základní vizuální návrh

V design módu Flash Builderu je vytvoření prototypu budoucího rozhraní rychlé. Nabízí se využití mnoha vizuálních prvků (formuláře, tabulky, grafy, uspořádání navigace, textové a grafické prvky). Vzhled těchto prvků lze dále upravovat nastavením stylů nebo také globálně volbou grafického schéma. Nejen vzhled, i ostatní vlastnosti tu je možné objektům přiřadit.

5.2.3 Tvorba aplikační logiky

Po návratu zpět ve zdrojovém kódu přibudou objekty, ke kterým je třeba přidat aplikační logiku. Zapisuje se ActionScriptem uvnitř bloku CDATA v elementu <fx:Script/>.

```
<fx:Script>
 <![CDATA[
       // aplikační logika
       …
 ]]>
</fx:Script>
```

Propojování jednotlivých kompnent vychází z modelu událostmi řízeného toku programu, je při tom důležité mít určitou představu o fungování programu. Událostmi se rozumí jednak ty způsobené interakcí uživatele, dále také datového charakteru. Hovoří se pak o aplikacích řízených daty.

Program organizátoru po spuštění v prohlížeči odešle serveru požadavek na data. Po jejich obdržení je uloží do datového objektu, ke kterému může přistupovat jako k asociativnímu poli (klíč – hodnota). Jejich uspořádání podle určitých kritérií znázorní stromovou strukturou. Na základě interakce uživatele filtruje položky zobrazované v tabulce.

Z pohledu provádění programu se po načtení do prohlížeče začne provádět inicializace vnitřní struktury, při které je odeslán požadavek serveru a objektu který jej vyslal je definována událost přijetí dat. Program zatím pokračuje v dalších krocích. Po příchodu dat je volána metoda obsluhy této události, pro jejich zpracování. Důležitým prvkem je zde tabulka, zobrazuje dynamická data a jednotlivým sloupcům přiřazuje odpovídající hodnoty z asociativního pole. Tento způsob využití datových zdrojů je užitečnou vlastností některých vizuálních datových komponent, zpravidla se jedná o dynamická data a může to být i volání jedné ze vzdálených služeb. Datový zdroj se váže na vlastnost dataProvider, k jeho obsahu je přistupováno jako k vlastnosti data.

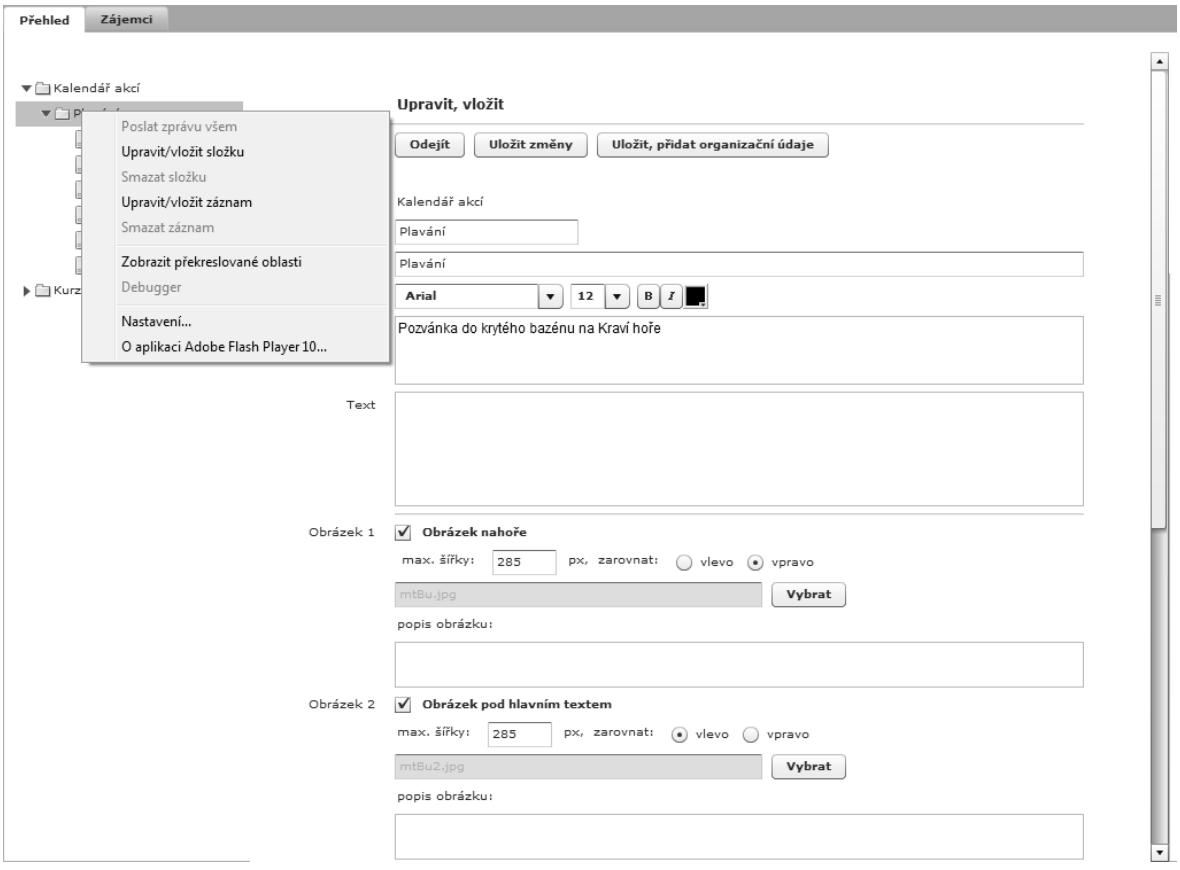
```
<mx:DataGrid id="tabulka" dataProvider="{zdroj}">
 <mx:columns>
       <mx:DataGridColumn id="sloupec1" dataField="data.datum"
headerText="Datum" />
       <mx:DataGridColumn id="sloupec2" dataField="data.text"
headerText="Událost" />
 </mx:columns>
</mx:DataGrid>
```

Použil jsem také vlastní komponenty, které bylo potřebné provázat s daty (například při vyobrazování obsahu polí tabulek nebo pro formuláře). Ne vždy však jde předchozí způsob použít tak, aby se projevily změny zpětně ve zdrojových datech. Lze však takové dceřiné komponentě definovat vlastnost představující rodiče, přes kterou je tato vazba realizována.

```
// Deklarace komponenty před jejím zobrazením
up=new komponenta();
up.aplikace=this;
```

Události vyvolané interakcí uživatele s navigačními prvky nebo v kontextovém menu jde zpravidla definovat přímo v zápisu MXML, jejich obsluha má podobu metody naprogramované v ActionScriptu.

5.2.4 Výsledná podoba



Obr. 7. Aplikace organizátoru, formulář pro vložení záznamu o plánované akci.

ZÁVĚR

Událostí, která znamenala průlom ve vývoji moderních aplikací, byl přechod k programování s objektově orientovaným přístupem. Koncepce nových technologií a postupů z objektového modelu vychází a ve velké míře nachází uplatnění v prostředí internetu. Využití klasických technologií má stále své opodstatnění v případech, kdy slouží jejich původnímu záměru - zprostředkování informací. Framework flex výrazným způsobem rozšiřuje možnosti internetu o interaktivní grafická rozhraní zpracování dat nebo k prezentaci různých obsahových forem. Podobných technologií je více, Flex však zatím má vývojový náskok a výhodu plynoucí z jeho širokého rozšíření.

ZÁVĚR V ANGLIČTINĚ

Event, that maked a breakthrough in the development of advanced applications, the transition to object-oriented programming approach. The concept of new technologies and procedures based on the object model is a widely find application in the Internet environment. Use of conventional technology is still warranted when serving their original purpose - providing information. Flex framework significantly extends the capabilities of the Internet by an interactive graphical interface, data processing or presentation of different content forms. There are more technologies, Flex development has leading position in benefits and therefore is wide distributed.

SEZNAM POUŽITÉ LITERATURY

Monografie:

[1] CEDERHOLM, Dan. *Webdesign s webovými standardy*. Brno: Zoner Press, 2004. ISBN: 80-86815-15-3.

[2] GASSNER, David. *Adobe Flex 3 Bible*. United Kingdom: John Wiley & Sons, 2008. 978 s. ISBN: 978-04-702-8764-4.

[3] GILMORE, W. JASON. *Velká kniha PHP a MySQL*. Brno: Zoner Press, 2007. ISBN: 80-86815-53-6.

[4] LACKO, L. *Ajax hotová řešení*. 1. vyd. Brno: Computer Press, 2008. 268 s. ISBN: 80-251-2108-5.

[5] LACKO, L. *PHP a MySQL hotová řešení*. 1. vyd. Brno: Computer Press, 2005. 300 s. ISBN: 80-251-0397-8.

[6] PAPA, John. *Silverlight datové služby*. Brno: O´Reily/Zoner Press, 2009. ISBN: 978-80-7413-041-0.

[7] ŠPINAR, David. *Tvoříme přístupné webové stránky*. Brno: Zoner Press, 2004. ISBN: 80-86815-11-0.

[8] ULLMAN, L. *PHP a MySQL*. Brno: Computer Press, 2004. ISBN: 80-251-0063-4.

[9] WELING, L. a THOMSON L. *MySQL*. Brno: Computer Press, 2005. ISBN: 80-251-0671-3.

Internetové zdroje:

[10] *Adobe Live Docs: Flex 3* [Online]. Dostupný z URL: <http://livedocs.adobe.com/flex/3/html/>

[11] BERNARD, Borek. *Adobe Flex - Co je a co není* [Online]. Dostupný z URL: <http://interval.cz/clanky/adobe-flex-co-je-a-co-neni/>

[12] BERNARD, Borek. *Rich Internet Applications v roce 2008*[Online]. Dostupný z URL: <http://interval.cz/clanky/rich-internet-applications-v-roce-2008/>

[13] GARRETT, J. *Ajax: A New Approach to Web Applications* [Online]. Dostupný z URL: <http://www.adaptivepath.com/ideas/essays/archives/000385.php/>

[14] KRCHA, Tomáš. *Stručný průvodce po frameworku Flex 4 – Webové služby* [online]. Dostupný z URL: <http://zdrojak.root.cz/clanky/flex-4-webove-sluzby/>

[15] TILL, Michal. *Svěží vzduch pro vývoj: Adobe AIR* [Online]. Dostupný z URL: <http://connect.zive.cz/jaky-je-adobe-air/>

[16] *Userinterface: Flex vs. Ajax, Friends or foes?* [Online]. Dostupný z URL: <http://www.uiresourcecenter.com/ui-technologies/adobe-flex/white-papers.html/>

[17] *Wikipedia: Document object model* [Online]. Dostupný z URL: <http://cs.wikipedia.org/wiki/Document_Object_Model/>

[18] *Wikipedia: JavaScript* [Online]. Dostupný z URL: <http://cs.wikipedia.org/wiki/Javascript />

[19] *Wikipedia: XMLHttpRequest* [Online]. Dostupný z URL: <http://en.wikipedia.org/wiki/XMLHttpRequest />

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

.NET	.NET (z anglického „*dot*“ + *NET*) je zastřešující název pro soubor technologií tvořících celou platformu, která je dostupná nejen pro Web, Windows i Pocket PC
AIR	Adobe Integrated Runtime, běhové prostředí pro aplikace webových technologií (HTML/CSS, Ajax, Flex, Flash) integrované se systémem lokálního počítače.
AMF	Adobe Message Format je formátem využívaným pro výměnu dat mezi Flashovou aplikací a vzdálenou službou.
API	Application Programming Interface je soborem metod nebo tříd z knihoven, které může programátor využít.
AS3	ActionScript 3 je programovací jazyk Flashové platformy (Flex, Flash, jejich servery, služby a klientské programy). Vychází z ECMA skriptu, je silně typový a objektově orientovaný.
ASP.NET	je součást .NET Frameworku pro tvorbu webových aplikací a služeb. Je nástupcem technologie ASP (Active Server Pages) a konkurentem JSP (Java Server Pages).
CMS	Content Management System - systém správy obsahu nebo také publikační či redakční systém.
ECMA	European Computer Manufacturer Association je mezinárodní organizace vyvíjející standardy v oblasti informačních a komunikačních technologií.
ISO	International Organization for Standardization je mezinárodní organizace definující standardy.
J2EE	Java 2 Enterprise Edition (také Java EE) součást platformy Java určená pro vývoj a provoz podnikových informačních systémů a aplikací.
JSON	JSON (JavaScript Object Notation) - formát dat, který je zapisuje jako JavaScriptové objekty pomocí skládání polí, map a skalárních hodnot.
MXML	MXML (Magic eXtensible Markup Language) značkovací jazyk používaný

	k zápisu vizuálních nebo datových prvků v Adobe Flex, vychází z XML.
OOP	Objektově orientované programování.
RIA	Rich Internet Applications - koncepce moderních aplikací cross-platformního charakteru s funkčností podobnou klasickým počítačovým programům.
RPC	Remote procedure call – vzdálené volání procedur, pro vykonání procedury v jiném umístění než je program.
SDK	Software Development Kit je souborem nástrojů pro vývoj aplikací určité platformy (knihovny, kompiler apod.).
SEO	Optimalizace pro internetové vyhledávače při indexaci obsahu stránek (Search Engine Optimization).
SOA	Service Oriented Architecture je označení informační systémů navržených jako webové služby, využívají ke komunikaci SOAP (Simple Object Access Protokol).
W3C	World Wide Web Consortium – konsorcium, jehož členové společně s veřejností vyvíjí webové standardy.
WPF	Windows Presentation Foundation, je podmnožinou .NET Frameworku od verze 3.0, klade si za cíl sjednocení grafického uživatelského rozhraní. Technologie je použita ve Windows Vista a Windows7.
XML	XML (eXtended Markup Language) je standardizovaný značkovací jazyk.

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

SEZNAM PŘÍLOH

PŘÍLOHA P I: OBSAH PŘILOŽENÉHO CD

Příloha obsahuje informace o obsahu přiloženého CD

Přiložené CD obsahuje:

- bakalářskou práci ve formátu PDF
- soubory potřebné pro umístění publikačního systému z praktické části na internet, Flexové projekty pro něj vytvořených komponent
- složku se snímky jednotlivých součástí systému a pracovního prostředí

Stránky jsou dočasně umístěny na adrese http://www.lukase.eu/.

PŘÍLOHA P II: UŽIVATELSKÁ PŘÍRUČKA

Příloha obsahuje instrukce k instalaci a práci s publikačním systémem.

Instalace

Při instalaci je třeba zkopírovat soubory a složky internetové poradny do kořenového adresáře stránek na server poskytovatele hostingové služby. Nejlépe pomocí některého FTP klienta, např. FileZilla nebo Total Commander.

Postup ve FileZilla:

1) Otevřete Správce míst (v horním menu Soubor->Správce míst), zvolte „Nové místo“ a zde vyplňte údaje:

   hostitel - adresa FTP serveru uvedená poskytovatelem,

   port – pokud je třeba,

   typ serveru – podle poskytovatele, nejčastěji pouze FTP server,

   způsob přihlašování – normální,

   uživatelské jméno a heslo.

2) Po vyplnění těchto údajů je možné se připojit k serveru. Klepněte na tlačítko „Spojit“. V pravém okně se objeví adresářová struktura složek na serveru. Kořenový adresář bude tento, nejnižší úrovně, i když někde může být výše např. jako složka /wwwroot.

3) V levém panelu otevřete složku, ve které jsou soubory internetové poradny. Označte v ní všechny soubory a v kontextovém menu pravého tlačítka zvolte „Nahrát na server“.

4) Otevřete Nastavení (v horním menu Upravit->Nastavení), zde u Editace souborů označte volbu výchozího editoru „Use system default’s editor for text files“ (výchozí systémový editor pro úpravu textových souborů) nebo „Use custom editor“ ke které přiřadíte např. PSPad editor. Dále označte volbu „Vždy použít výchozí editor“. Klepněte na tlačítko „OK“.

5) Je potřeba editovat soubor s nastaveními připojení k databázi a údaji pro odesílání zpráv prostřednictvím e-mailu. Ta jsou uložena v souboru „nastaveni.ini“. V pravém panelu označte tento soubor a v kontextovém menu pravého tlačítka

vyberte editovat. Zde proveďte úpravy podle údajů od poskytovatele hostingu. Uložte v editoru (CTRL+s), potvrďte v dialogovém okně FTP klienta FileZilla.

6) V posledním kroku do okna prohlížeče vepište adresu Vašich stránek doplněnou o cestu ke skriptu, jenž vytvoří potřebné tabulky v databázi. Cesta k /databaze/mysql.php (např. http://má.doména**/databaze/mysql.php**). Po spuštění se v okně zobrazí zprávy o průběhu procesu.

Registrace uživatele do publikační části

Provádí správce z rozhraní správy z nabídky „Uživatelé“, která také slouží k zobrazení přehledu uživatelů. Nově registrovaný uživatel obdrží e-mail s vygenerovaným heslem pro přístup, které později může změnit.

Vkládání příspěvků

Příspěvky nebo texty k diskusím se vkládají přes formulář z nabídky „vložit“. Lze nastavit datum, kdy mají být zveřejněny. Správce zakládá kategorie a schvaluje zveřejnění textů.

Vkládání informací o aktivitách a jejich organizace

Tyto funkce jsou přístupné z aplikace organizátor jen správci. Nabízí vkládání úpravy a mazání těchto záznamů, dále zobrazení přihlášených zájemců, odesílání mailů a tisk jejich seznamu. Tato součást systému používá jednoduchého uživatelského rozhraní, její funkce jsou přístupné přes kontextové menu. Má spojitost s „kalendářem akcí“ a nabídkou „kurzy“ ve veřejné části.

Vkládání obrázků do galerie

Snímky jsou do galerie vkládány přes rozhraní, které umožňuje přidat jim popisky, změnit jejich pořadí a v případě prací určených k prodeji také další informace.

PŘÍLOHA P III: TVORBA DATABÁZE.

Příloha obsahuje instrukce k vytvoření tabulek MySQL databáze z praktické části.

Tabulky redakčního systému:

```
CREATE TABLE `autori` (
  `id_autor`       int(10) unsigned  NOT NULL auto_increment,
  `prihlaseni_jm` varchar(32)       NOT NULL,
  `prihlaseni_h`  varchar(32)       NOT NULL,
  `jmeno`          varchar(255       NOT NULL,
  `mail`           varchar(255       NOT NULL,
  `mail_zverejnit` varchar(4)       NOT NULL default 'ano',
  `profil_text`    text,
  `profil_fotka`  varchar(64)       NOT NULL default
'bez_fotky.jpg',
  `poznamky`       text,
  `opravneni`      varchar(1)        NOT NULL default 'u',
  PRIMARY KEY  (`id_autor`)
) ENGINE=MyISAM  DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci
AUTO_INCREMENT=1 ;

CREATE TABLE `kategorie` (
  `id_kategorie`  int(10) unsigned  NOT NULL auto_increment,
  `nazev`          varchar(255)      default NULL,
  `popis`          text,
  `verejna`        tinyint(4)        NOT NULL default '1',
  `zobrazovat`     tinyint(4)        NOT NULL default '1',
  `sablona`        varchar(255)      default NULL,
  `cesta`          varchar(255)      NOT NULL default '/',
  PRIMARY KEY  (`id_kategorie`)
) ENGINE=MyISAM  DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci
AUTO_INCREMENT=1 ;

CREATE TABLE `clanky` (
  `id_clanek`      int(10) unsigned  NOT NULL auto_increment,
  `datum`          timestamp         NOT NULL default
CURRENT_TIMESTAMP,
  `datum_zverejneni` datetime       default NULL,
  `id_autor`       int(10) unsigned  NOT NULL,
  `id_kategorie`  varchar(256)      NOT NULL,
  `nadpis`         varchar(256)      default NULL,
  `anotace`        text,
  `clanek`         text,
  `clanek_css`     text,
  `poznamky`       text,
  `schvaleny`      char(4)           NOT NULL default 'ne',
  `diskuze`        char(4)           NOT NULL default 'ne',
  PRIMARY KEY  (`id_clanek`)
) ENGINE=MyISAM  DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci
AUTO_INCREMENT=1 ;
```

```
CREATE TABLE `fora` (
  `id_prispevek` int(10) unsigned  NOT NULL auto_increment,
  `id`           int(10) unsigned  NOT NULL default '0',
  `id_clanek`    int(10) unsigned  NOT NULL default '0',
  `datum`        timestamp NOT NULL default CURRENT_TIMESTAMP,
  `jmeno`        varchar(255)      default NULL,
  `mail`         varchar(255)      default NULL,
  `predmet`      varchar(255)      default NULL,
  `text`         text,
  `spada_pod`    int(11)           NOT NULL default '0',
  PRIMARY KEY  (`id_prispevek`),
  KEY `id_clanek` (`id_clanek`)
) ENGINE=MyISAM  DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci
AUTO_INCREMENT=1 ;
```

Tabulky aplikace správy plánovaných akcí a zájemců:

```
CREATE TABLE `nabidka` (
  `odkaz`        varchar(64)       NOT NULL,
  `text_odkazu`  varchar(64)       NOT NULL,
  UNIQUE KEY `odkaz` (`odkaz`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_czech_ci;

CREATE TABLE `nabidka__kalendar_akci` (
  `id`           int(8) unsigned NOT NULL auto_increment,
  `stranka`      varchar(32)       NOT NULL,
  `nadpis`       varchar(256)      NOT NULL,
  `text_kratce`  varchar(2048)     NOT NULL,
  `text`         text              NOT NULL,
  `obrazek`      varchar(1024)     default NULL,
  `obrazek_popis` varchar(1024)    default NULL,
  `obrazek2`     varchar(1024)     default NULL,
  `obrazek2_popis` varchar(1024)   default NULL,
  `odkaz`        varchar(2048)     default NULL,
  `text_css`     varchar(2048)     default NULL,
  `cena_var1`    varchar(8)        default NULL,
  `cena_var1popis` varchar(512)    default NULL,
  `cena_var2`    varchar(8)        default NULL,
  `cena_var2popis` varchar(512)    default NULL,
  `cena_var3`    varchar(8)        default NULL,
  `cena_var3popis` varchar(512)    default NULL,
  `cena_var4`    varchar(8)        default NULL,
  `cena_var4popis` varchar(512)    default NULL,
  `volne1`       varchar(512)      default NULL,
  `volne2`       varchar(512)      default NULL,
  PRIMARY KEY  (`id`)
) ENGINE=MyISAM  DEFAULT CHARSET=utf8 COLLATE=utf8_czech_ci
AUTO_INCREMENT=1 ;
```

```
CREATE TABLE `nabidka_organizace` (
  `id`            int(8) unsigned   NOT NULL auto_increment,
  `tabulka`       varchar(64)       NOT NULL,
  `id_z_tabulky`  int(8)            NOT NULL,
  `kontakt_jmeno` varchar(64)       NOT NULL,
  `kontakt_mail`  varchar(64)       NOT NULL,
  `kontakt_adresa` varchar(256)     default NULL,
  `kontakt_tel`   varchar(16)       default NULL,
  `misto`         text              NOT NULL,
  `cas`           datetime          default NULL,
  `prihlasky_do`  datetime          default NULL,
  `dalsi_info`    text,
  `mapa`          varchar(2048)     default NULL,
  `mapa_at`       varchar(256)      default NULL,
  PRIMARY KEY  (`id`)
) ENGINE=MyISAM  DEFAULT CHARSET=utf8 COLLATE=utf8_czech_ci
AUTO_INCREMENT=1 ;

CREATE TABLE `nabidka_zajemci` (
  `id`            int(8) unsigned   NOT NULL auto_increment,
  `datum_z`       timestamp         NOT NULL default
CURRENT_TIMESTAMP,
  `kontakt_jmeno` varchar(32)       NOT NULL,
  `kontakt_prijmeni` varchar(32)    NOT NULL,
  `kontakt_mail`  varchar(64)       NOT NULL,
  `kontakt_ad_ulice` varchar(64)    default NULL,
  `kontakt_ad_mesto` varchar(64)    default NULL,
  `kontakt_ad_psc` varchar(8)       default NULL,
  `kontakt_tel`   varchar(16)       default NULL,
  `predmet`       varchar(512)      NOT NULL,
  `popis`         varchar(2048)     default NULL,
  `tabulka`       varchar(64)       NOT NULL,
  `id_z_tabulky`  int(8)            unsigned NOT NULL,
  `slozka`        varchar(64)       NOT NULL,
  `stranka`       varchar(32)       NOT NULL,
  `text`          text,
  `datum`         datetime          default NULL,
  `rezervovat`    varchar(512)      NOT NULL default 'Ano',
  `platba`        varchar(512)      default NULL,
  `platba_cena`   varchar(8)        default NULL,
  `prevzeti`      varchar(512)      default NULL,
  `pocet`         varchar(32)       NOT NULL default '1',
  `stav`          varchar(32)       NOT NULL default '1',
  `zasilat_info`  varchar(256)      default NULL,
  `poznamka`      varchar(1024)     default NULL,
  `volne1`        varchar(512)      default NULL,
  `volne2`        varchar(512)      default NULL,
  PRIMARY KEY  (`id`)
) ENGINE=MyISAM  DEFAULT CHARSET=utf8 COLLATE=utf8_czech_ci
AUTO_INCREMENT=1 ;
```