

ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA APLIKOVANÝCH VĚD
KATEDRA INFORMATIKY A VÝPOČETNÍ TECHNIKY

BAKALÁŘSKÁ PRÁCE

ROZŠIŘUJÍCÍ MODUL „TÉMATA PRACÍ“ PRO WEB KATEDRY

PROHLÁŠENÍ

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 24. 4. 2012

Václav Rajtmajer

PODĚKOVÁNÍ

Rád bych poděkoval zejména vedoucímu práce Ing. Přemyslu Bradovi, MSc., Ph.D. za trpělivost a vstřícnost při konzultacích, panu Ing. Martinovi Dostalovi za cenné rady a v neposlední řadě mému otci za motivaci.

ABSTRACT

This thesis focuses on MODx system, which is used as a web content management system for the Department of Computer Science and Engineering. First the MODx system is properly described with components and procedures introduced that are applied in this system.

Secondly, a new extension module called “project topics” is created, which is ready for integration into the department website. It will demonstrate the theory that is published in this thesis.

This document describes the basic components of MODx; the two basic ways of developing web applications in this system and the principles for creating modules for the website of the Department of Computer Science and Engineering.

1 OBSAH

2	Úvod.....	1
2.1	Cíl	1
2.2	Konvence pro psaní textu	1
3	System pro správu obsahu (CMS).....	2
4	MODx teoreticky.....	4
4.1	Historie MODx-u	5
4.2	Verze MODx-u.....	5
4.2.1	MODx Evolution.....	5
4.2.2	MODx Revolution	5
4.2.3	„Evolution vs. Revolution“	6
4.3	MODx nastavení.....	6
4.4	MODx architektura	7
4.4.1	Dokument	7
4.4.2	Šablona	8
4.4.3	Template Variable	8
4.4.4	Chunk.....	9
4.4.5	Snippet.....	10
4.4.6	Plugin	11
4.5	Přehled tagů.....	11
4.6	Příklad použití	13
4.7	xPDO.....	13
4.8	Verzování a přenositelnost	14
4.9	Alternativa	14

5	Rozšiřující modul „Témata prací“	16
5.1	Analýza provedení webu katedry	16
5.2	Analýza současného modulu „Témata prací“	17
5.2.1	Chyby a nedostatky modulu	18
5.3	Návrh nové architektury systému.....	19
5.3.1	Model.....	19
5.3.2	View	20
5.3.3	Controller.....	20
5.4	Role databáze	20
5.4.1	Datový model	21
5.5	Implementace komponent pro MODx - vyvinutý modul „Témata prací“	21
5.5.1	Nastavení modulu.....	21
5.5.2	Popisy stránek.....	22
5.5.3	Změny oproti předchozí verzi modulu	23
5.5.4	Popis použitých funkcionalit.....	24
5.5.5	Zásady při tvorbě modulů pro web katedry KIV.....	27
5.5.6	Vzhled stránek	28
5.6	Konverze dat, používaných modulem	28
5.7	Testování nově vyvinutého modulu	28
5.7.1	Testy nad vrstvou Controller/View.....	29
5.7.2	Testy nad vrstvou Model	29
6	Závěr	30
	Příloha A - Základní a běžně používaná šablona	35
	Příloha B - Uživatelská dokumentace	36
	Příloha C - Datový model.....	42

2 ÚVOD

Webové stránky Katedry informatiky a výpočetní techniky (dále jen KIV) a obecně všechny webové stránky je třeba stále aktualizovat a modernizovat, aby si udržely svou informační hodnotu v množství rychle přibývajících nových webů. Pokud mluvíme o nějaké malé doméně se statickými stránkami (například prezentace živnosti), není problém jednou za čas stránkám zmodernizovat vzhled, či aktualizovat obsah, aby stránky vyhovovaly novým standardům. Lehkost této akce se ale vytrácí, mluvíme-li o stránkách objemných, často aktualizovaných, v nejhorším případě psaných více programátory.

Pro vnesení řádu je dobré pro tyto weby používat správně zvolený redakční systém (detailní popis v kapitole 3).

Webové stránky KIVu byly od roku 2004 tvořeny v OpenCMS. Tento systém je založen na jazycích Java a XML. Tato volba se ale ukázala jako nevyhovující a bylo zapotřebí vybrat redakční systém nový. Zvolen byl MODx (detailní popis v kapitole 4).

2.1 CÍL

Cílem této práce je tedy sestavit základní popis systému MODx, vytvořit stručný návod pro práci s ním, doplněný o vzorové příklady. Jelikož je MODx systém mnoha možností, je třeba zveřejnit zásady pro psaní modulů přímo do webu KIVu.

Pro demonstrování zde popsané teorie pak naprogramujeme zadaný modul pro evidenci témat prací, který bude kompatibilní s již existujícím webem.

Tato práce tedy může být použita jako návod k vývoji modulů pro web KIVu.

2.2 KONVENCE PRO PSANÍ TEXTU

```
<html>
  <head>Hlavička</head>
  <body>Tělo stránky</body>
</html>
```

Takto je psán zdrojový kód.

```
/plugins/plugins_ready
```

Takto je psán název souborů.

3 SYSTÉM PRO SPRÁVU OBSAHU (CMS)

Webové stránky jsou stále důležitějším zdrojem informací a na jejich tvorbu jsou kladeny stále náročnější požadavky. Weby jsou obsáhlejší, propracovanější, využívají většinou více provázaných technologií. Psaní takovýchto webů se stává obtížnější a často se na jejich vývoji musí podílet i více vývojářů.

CMS (Content Management System) je systém pro správu obsahu (v našem případě webového obsahu), někdy nazýván jako redakční, či publikační systém [1].

Jeho úkolem je dělat vytváření a správu webového obsahu přehlednější, rychlejší, bezpečné a přístupné i pro uživatele neznalé webových technologií. Hlavní předností všech CMS je oddělení obsahu stránek od jejich vzhledu [31].

Většina CMS podporuje následující funkce [2]:

- Vytváření a úprava webových stránek.
- Vytváření položek menu.
- WYSIWYG, tj. textový editor, který převádí vzhledové formátování na HTML tagy.
- Správa souborů.
- Administrace uživatelů a přidělování práv.
- Archivace obsahu a řadu dalších funkcí.

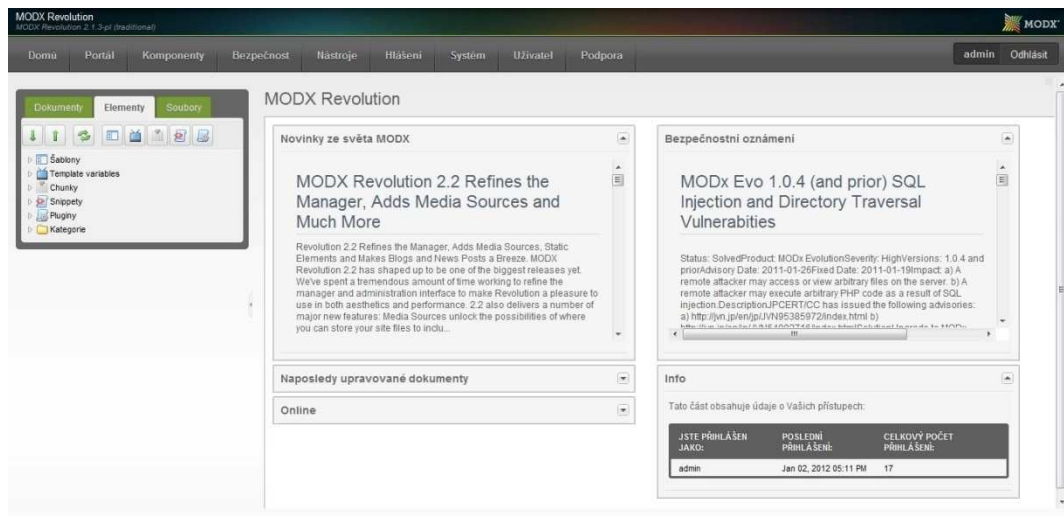
CMS existuje dnes desítky a dělí se hned z několika pohledů [3]. Například podle:

- Technologie, na které jsou sestaveny. (Java, PHP, MySQL, PostgreSQL, Python, .NET, Perl)
- Licence. (MIT, BSD, GPL)
- Rychlosti.
- SEO.
- Zabezpečení.
- Možnosti doplňků a řady dalších kritérií.

CMS je výhodné používat hned z několika důvodů:

- U dynamických webů je levnější.
 - o Pořizovací cena je sice vyšší, ale při aktualizaci stránky si webdesignové firmy účtují poplatky, zatímco s CMS se dá aktualizace stránky přirovnat k psaní dokumentu v aplikaci Microsoft Word a může ji provést i nezkušený uživatel.
- Vývoj je rychlejší.
 - o Nezdržujeme se vývojem standardních balíčků, které už napsal někdo před námi.
 - o Děláme méně chyb v kódu, protože většinu kódu nepíšeme my.
- Stránky jsou optimalizované pro vyhledávače (SEO).
 - o Pokud použijeme dobrý CMS.
- Máme kontrolu nad zabezpečením stránek.
 - o Většina systémů je zabezpečena proti útokům. (např. Injection, krádeže session, XSS útoky, atd.)
 - o Administrátor má pod kontrolou práva uživatelů a uživatelských skupin.

4 MODX TEORETICKY



Obrázek 4.1 - Výchozí stránka MODx-u po přihlášení do manageru.

MODx je mocný a komplexní CMS, psaný v PHP, s veškerými daty uloženými v MySQL databázi. Vývojáři ho někdy nazývají spíše CMF (Content Management Framework), protože v něm lze vytvářet i nové CMS.

Vlastnosti MODx-u, které ho dělají výjimečným [4]:

- Podléhá licenci GNU GPL.
- Pracuje s PHP 4.3.11 a vyšší.
- Může být instalován na webové servery IIS, Apache, lighttpd, nginx a Zeus
- Kód jádra je objektově orientovaný a rozdělený do architektury MVC (Model-view-controller)
- Instalátor s grafickým rozhraním.
- K veškeré práci se používá přehledný AJAXový manager (zobrazen na obrázku 4.1), který lze spustit v jakémkoliv webovém prohlížeči. Pro úpravu textů je možné doinstalovat WYSIWYG editor.
- Silná podpora SEO, jako například přátelské URL, správa všech metadat, validita kódu, atd.
- Podpora všech AJAXových knihoven, jako například jQuery, MooTools, ExtJS, Prototype, atd.

- Správa uživatelů, uživatelských skupin a jejich práv.
- Systém je přeložen do více než 20 různých jazyků.
- Oddělené sessions pro připojení do administrace a pro prohlížení webu.

MODx je ale stále vyvíjen a tak se tyto vlastnosti stále rozrůstají.

4.1 HISTORIE MODX-U

Projekt MODx CMS začali programátoři Raymond Irving a Ryan Thrash v roce 2004, když vyvíjeli doplněk do systému Etomite. V roce 2005 se pak MODx odtrhl od Etomite a začal svou vlastní dráhu. Tak vznikl MODx Evolution. V roce 2010 byl MODx Evolution kompletně přepsán a tato nově vzniklá vývojová větev byla pojmenována MODx Revolution [5].

V roce 2007 vyhrál MODx cenu za nejslibnější Open Source CMS [4].

Česká komunita, zajímající se o MODx, je velice malá a rozšiřuje se jen velmi pomalu. Překlady částí MODx-u vznikaly už od dubna roku 2006, ale kompletní verze překladu vyšla až na začátku roku 2009 a české fórum bylo spuštěno až v říjnu 2010 [6].

4.2 VERZE MODX-U

Jak jsem psal v předchozí kapitole, MODx má od roku 2010 dvě vývojové větve. Obě se od sebe liší hlavně syntaxí a aplikační logikou, nicméně jsou si velmi podobné a snadno převoditelné. V následujících kapitolách budeme uvažovat nejnovější verzi MODx Revolution 2.1.5-pl.

4.2.1 MODX EVOLUTION

Je první verze, vyvíjená z „prázdného papíru“. Aktuální stabilní verze je 1.0.6 (vydaná 8. 3. 2012) a autoři se chlubí celkem šesti patchi.

Jádro je neobjektové, parser rekurzivní, nicméně nedokonale a tak existuje limit objemu stránky na cca 5000 dokumentů [7].

4.2.2 MODX REVOLUTION

MODx Evolution přepsaný do objektové podoby. Jako objekt je nyní reprezentován každý dokument i všechny ostatní části systému.

Parser opraven, takže žádný limit v podobě počtu dokumentů již neexistuje. Autoři se prezentují celkem sedmnácti patchi, z nichž poslední stabilní verze má značení 2.1.5-pl a je vydaná 9. 11. 2011 [8].

4.2.3 „EVOLUTION VS. REVOLUTION“

Evolution je verze starší a tak i technologie je zastaralejší. Větší pozornost se dnes tedy upíná k novější verzi Revolution. Na druhou stranu jako starší je stabilnější a má zatím větší podporu již hotových doplňků.

„Upgrade“ z MODx-u Evolution na MODx Revolution ve většině případů nemá za následek žádnou ztrátu informace. Jen při převodu vlastních snippetů (pojem popsán v kapitole 4.4.5), či jiných částí systému, které si sami napíšeme, může dojít k dysfunkci dané části.

Základní myšlenky obou systémů si jsou velice podobné a na první pohled se liší jen podle syntaxe tagů. Dále budeme popisovat MODx Revolution.

4.3 MODX NASTAVENÍ

Nastavení systému je většinou první krok ihned po nové instalaci. Vývojáři MODx-u to nejspíše vzali na vědomí a snažili se udělat nastavování systému přehledné a příjemné pro každého. Volby jsou skvěle kategorizovány, okomentovány a umožňují rychlou filtraci podle klíče.

Je potřeba vyzdvihnout snadnou možnost přidávání nových položek konfigurace, což dává vysoký komfort uživateli. Stačí k tomu vyplnit krátký formulář (viz. Obrázek 4.2). Tato možnost se na první pohled nejeví nijak důležitě, ale v následujících kapitolách si ukážeme, že každou „value“ nastavení lze vypsát pomocí tagu `[[++key]]`. To nám pak umožní vytvořit například slogan webu nebo cokoliv jiného, potřebného globálně pro celý web.

Je třeba říci, že systém hned po instalaci má velice příjemné defaultní nastavení a pro jednodušší weby si většinou vystačíme jen s přenastavením některých hodnot v kategorii „Portál“. Tam najdeme nastavení, jako jsou: „auto_menuindex“, „default_content_type“, „default_template“, „error_page“, „site_name“ a mnoho dalších.

Vytvořit novou položku konfigurace

Klíč:

Klíč položky nastavení. Položka nastavení bude dostupná v obsahu jako placeholder `[[++key]]`.

Název:

Název položky nastavení. Můžete zadat také klíč slovníku.

Popis:

Popis položky konfigurace. Můžete zadat také klíč slovníku.

Hodnota

Typ políčka:
Textové políčko
Políčko typu nastavení. Může být pouze: textfield, textarea nebo boolean.

Jmenný prostor:
core
Jmenný prostor, ke kterému tato položka nastavení patří. Výchozí téma slovníku bude načteno pro tento jmenný prostor při dotazu na nastavení.

Oblast záznamu slovníku:

Zadejte klíč záznamu slovníku pro tuto oblast. Pokud ve slovníku záznam není, zobrazí se pouze klíč oblasti.
Oblasti jádra: authentication, caching, file, furis, gateway, language, manager, session, site, system

Zpět Uložit

Obrázek 4.2 - Vytváření nové položky konfigurace.

Ostatní kategorie slouží pak k nastavení například cachování, přátelských URL, proxy, WYSIWYG, a mnoho dalších.

Některé doplňkové balíčky mají také po instalaci svou vlastní kategorii v nastavení.

4.4 MODX ARCHITEKTURA

MODx se snaží co nejvíce ušetřit čas programátorům a tak na zobrazování stránek používá tzv. „šablonování“ (v originále Templating) [11]. To je způsob, jak oddělit ze stránky co nejvíce znovupoužitelného kódu a do praxe se převádí částmi, které jsou popsány v následujících podkapitolách. Každá část má svá pravidla, atributy a funkce. [10]

Komunikace mezi jednotlivými částmi systému probíhá za pomoci tzv. MODx tagů. To jsou tagy, podobné řetězci: „`[[tag]]`“, které parser nahradí daným atributem instance (dokumentu, chunku, atd.). Každý typ objektu má svou vlastní syntaxi tohoto tagu [9, 11].

4.4.1 DOKUMENT

Dokument (v originále Resource) je část HTML (samozřejmě problém s HTML odpadá při použití WYSIWYG editoru) kódu, která je jedinečná pro každou stránku. Může obsahovat pouze čisté HTML nebo MODx tagy.

Dělí se na 4 základní typy

1) Dokument

- Může obsahovat HTML kód, doplněný o MODx tagy.

2) Webový odkaz

- Nemá žádný obsah a proto ho parser ignoruje. Jen nás přesměruje na zadanou URL (může to být i na naší stránce).

3) Symbolický odkaz

- Podobné jako webový odkaz s tím, že zachová URL adresu.

4) Statický zdroj

- Funguje stejně jako obyčejný dokument s tím rozdílem, že zdroj není uložen v databázi, nýbrž v souboru na disku.

Každý dokument je jednoznačně identifikován svými atributy - například `[[*id]]`, `[[*pagetitle]]`, `[[*content]]`, `[[*template]]`, `[[*deleted]]` a mnoho dalších [12].

4.4.2 ŠABLONA

Šablona (v originále Template) je část HTML kódu, která určuje základní rozložení stránky a která se většinou opakuje pro více dokumentů. Typicky obsahuje HTML tagy `<html>`, `<head>`, `<body>`, do kterých pak vkládá dynamické části kódu (jako například části dokumentů, chunky, snippets, atd.) pomocí MODx tagů.

Při vytváření nového dokumentu mu přiřadíme jednu konkrétní šablonu a při zobrazení stránky pak parser zařídí, aby vložil určité části dokumentu do šablony podle situovaných tagů [13].

4.4.3 TEMPLATE VARIABLE

Template Variable (dále jen TV) vznikly k rozšíření atributů, které obsahuje každý dokument. Jsou to uživatelsky definovaná pole (klíč – hodnota), která fungují stejně, jako jakýkoliv jiný atribut dokumentu (například atribut `[[*id]]` je také TV, ale defaultně vytvořen pro každý dokument).

TV může nabývat hodnot, které mají předem definovaný „datový typ“, jako je například E-mail, Obrázek, Soubor, Tag, Datum, WYSIWYG, atd. Po implementaci ji pak

můžeme nastavit vlastnosti, jako je výchozí hodnota, maximální délka, minimální délka. Všechny tyto vlastnosti jsou volitelné. Pro vznik proměnné je nutné zadat jméno a přiřadit ji alespoň k jedné šabloně, která ji pak bude používat. Ve vlastnostech lze pak omezit přístup konkrétním uživatelům nebo skupinám uživatelů.

Pokud vytváříme dokument, který používá šablonu s přiřazenou TV, v editačním menu se navíc objeví záložka Template variable, ve které můžeme změnit hodnotu naší TV tohoto konkrétního dokumentu.

Do stránky ji parser vkládá jako každý jiný tag a to náhradou tagu za hodnotu TV 1:1 [14]. Možnost předání parametrů je ukázána v příkladu [15].

Příklad: TV pro vypsání počtu nových a přečtených zpráv.

Kód TV, pojmenované „zpravy“:

```
Máte [[+novych]] nových a [[+prectenych]] zpráv!
```

Volání TV:

```
[[ *zpravy?novych=`8` &prectenych=`93` ]]
```

4.4.4 CHUNK

Nejprimitivnější element systému je chunk (v překladu kus). Je to kus statického HTML kódu, který se opakuje na více stránkách, jako například kontaktní informace, copyright, často celá patička stránky atd.

Tento statický HTML kód nesmí obsahovat PHP, ale může obsahovat volání snippetu, takže striktní omezení na statický obsah je relativní [16, 17].

Příklad: Chunk vypíše větu: Už je tu zase Josef Nový!

Kód chunku, pojmenovaném „privitani“:

```
<p>Už je tu zase [[+jmeno]]!</p>
```

Volání chunku:

```
[[ $privitani? &jmeno=`Josef Nový` ]]
```

4.4.5 SNIPPET

Snippet je obdoba chunku v dynamické podobě. Obsahuje jen PHP nebo JavaScript a nesmí se v něm vyskytovat žádné HTML [18].

Většina snippetů je cachovaná. Pokud se při volání před název snippetu napíše vykřičník, funkce se cachovat nebude a bude zpracovaná až po všech ostatních, které jsou v cachi uloženy.

Snippets často pracují s xPDO (popsáno v kapitole 4.7) a tak přistupují k celému systému (tedy i do databáze). Čisté SQL příkazy se nedoporučují, nicméně fungují [18].

Web bez snippetů by byl jen statický výpis informací a poptávka po takových webech není nikterak výrazná. Aby ale mohl používat snippets i obyčejný uživatel (pro které je vlastně CMS vytvořeno), musel by mít alespoň základní znalosti některých dynamických jazyků (PHP, JavaScript). To MODx řeší zavedením tzv. balíčků. Každý programátor může po vytvoření vlastního snippetu tento zabalit do balíčku a zveřejnit do MODx knihovny, kde bude dostupný pro všechny ostatní. Často používané snippets (jako třeba dynamické menu, formuláře, kalendář a mnoho dalších) jsou už vytvořeny a volně dostupné přes import balíčku. [20]

V následujícím příkladu je vidět, že snippet lze volat i s parametry, ke kterým se pak v PHP funkci přistupuje jako k obyčejným lokálním proměnným [19].

Příklad: Snippet, který funguje stejně jako PHP funkce include() [19]:

Kód snippetu, pojmenovaném „include“:

```
if (file_exists($file)) {  
    $o = include $file;  
} else { $o = 'File not found at: '.$file; }  
return $o;
```

Volání snippetu:

```
[[!include? &file=`/absolute/path/to/my/snippet.php`]]
```


4.4.6 PLUGIN

Plugin je velice podobný snippetu. Možnosti má naprosto totožné. Jediný rozdíl je v tom, že plugin nemá žádný zástupný tag, takže nemůže být zavolán uživatelem. Plugin je totiž asociován s určitou událostí systému a pouze a jen při výskytu této události je vykonán jeho kód. Takových událostí existují desítky [22].

To přináší řadu možností, aniž bychom museli přepisovat jádro systému. Můžeme tak systém snadno aktualizovat, což bychom nemohli, pokud by byl plugin dopsán přímo do jádra.

Příklad: Plugin po zaznamenání událostí `OnWebPageInit` nebo `OnWebPagePrerender` vypíše, co se děje [21].

Kód pluginu, který je připsán události `OnWebPageInit` a `OnWebPagePrerender`.

```
$eventName = $modx->event->name;
switch($eventName) {
    case 'OnWebPageInit':
        $modx->event->output('Stránka je načtená do $modxu!');
        break;
    case 'OnWebPagePrerender':
        $modx->event->output('Stránka se odesílá klientovi!');
        break;
}
```

4.5 PŘEHLED TAGŮ

Na příkladech uvedených výše jsme mohli vidět většinu syntaxí tagů, které lze v praxi použít. V tabulce 4.1 je vypsán jejich přehled.

Příklady, kde neexistují tagy:

- Šablona nemá svůj tag, jelikož k ní není třeba přistupovat z jiných míst. Každému dokumentu je jedna přiřazena pomocí svého ID a jiné použití pro ni není.
- Plugin nemá svůj tag, jelikož není nikdy manuálně volán. Jak jsme zjistili v kapitole 3.5.6, plugin je snippet, který reaguje na událost systému.

Obsahový element	Syntax tagu	Příklad
Šablona	---	---
Pole dokument	[[*field]]	[[*pagetitle]]
Template Variable	[[*templatevar]]	[[*tags]]
Chunk	[[\$chunk]]	[[\$header?lang=`en`]]
Snippet	[[snippet]]	[[!Wayfinder? &startId=`0`]]
Plugin	---	---
Placeholder	[[+placeholder]]	[[+modx.user.id]]
Odkaz	[[~link]]	[[~[[*id]]? &scheme=`full`]]
Nastavení systému	[[++system_setting]]	[[++site_start]]
Jazyk	[[%language_string_key]]	---
Komentář	[[-this is a comment]]	---

Tabulka 4.1 - Přehled tagů [9]

V tabulce 4.1 jsou použity některé speciální kombinace znaků. Jak je tedy možno konkrétní tag „složit“:

[[otevírací značka
!	(volitelný) tag se nebude cachovat
elementToken	(volitelný) pokud se nejedná o snippet, pak token určuje typ tagu (,\$' = chunk, +' = placeholder, atd.)
elementName	jméno tagu
@propertyset	(volitelný) nastavení vlastností
:filterName=`modifier`:...	(volitelný) jeden nebo více výstupních filtrů
?	(volitelný) indikuje, že budou následovat parametry
&propertyName=`propertyValue` &...	(volitelný) pokud předcházel znak otazník, pak musí následovat jeden nebo více parametrů, které jsou uváděné znakem &
]]	zavírací značka

Poznámky:

- Musí být použity zpětné apostrofy.

4.6 PŘÍKLAD POUŽITÍ

Pro názornou ukázkou použití všech těchto tagů by bylo dobré demonstrovat jejich spolupráci na názorném příkladu. V praxi použitelnou šablonu, ve které je znázorněna většina placeholderů, najdeme v příloze A.

4.7 xPDO

xPDO (eXtensions to PDO) je ORB (Object-relational bridge) knihovna implementovaná pro PHP 4.3.x a novější. ORB je programovací technika, která má za úkol automatickou konverzi mezi relační databází a objektově orientovaným jazykem. To nám dává k dispozici „virtuální objektovou databázi“, která může být používána bez programovacího jazyka [23].

xPDO je tedy rozšíření nástroje PDO, který „poskytuje ideální unifikované rozhraní pro komunikaci s různými databázovými systémy. PDO nabízí vrstvu abstrakce dat. Můžeme tedy pro práci s databází používat stále stejnou sadu příkazů - bez ohledu na to, v jakém databázovém systému je vaše databáze uložena.“ [31]

Jak už bylo naznačeno, celý systém MODx je převeden do virtuálního objektového modelu a ke všem jeho částem se dá přistupovat jako k objektům. To se dělá pomocí proměnné `$modx`, která existuje jen díky xPDO. xPDO totiž po převodu databáze do „objektové databáze“ přiřadí do proměnné `$modx` ukazatel na celou tuto databázi a ta nám postačí pro další práci se systémem. (Ještě existuje proměnná `$xpdo`, ale tu využívá nejvíce jádro a my jsme jí využili jen pro vytvoření proměnné `$modx`) [24].

Jen díky funkcím proměnných `$modx` a `$xpdo` [25] lze z MODx-u vytvořit nejen CMS, ale i CMF [24].

Naši „objektovou databázi“ použijeme jen ve snippetech a pluginech. Například:

- Pro vytvoření chunku:

```
$object = $modx->newObject('modChunk');
$object->set('name', 'ChunkName');
$object->setContent('Tady bude obsah chunku.');
```

- **Pro výpis názvu aktuálně zobrazené stránky:**

```
echo $modx->resource->get('pagetitle');
```

- **Pro výpis TV:**

```
echo $modx->resource->getTVValue('nazev_promenne');
```

- **Po zavolání tohoto kódu se aktuální dokument posune v menu o index níže:**

```
$id = $modx->resource->get('id');  
$poradi = $modx->resource->get('menuindex')  
$object = $modx->getObject('modResource', $id);  
$object->set('menuindex', ++$poradi);  
$object->save();
```

4.8 VERZOVÁNÍ A PŘENOSITELNOST

Při vytváření webu je třeba dbát na zálohování práce a často i její přenositelnost (web se může vyvíjet na jiném serveru, než na kterém se zveřejňuje po dokončení).

Verzování lze do systému doplnit například balíkem VersionX [26], který uchovává veškeré změny všech dokumentů, chunků a snippetů. Alespoň to je v popisu balíku napsáno, já osobně jsem balík netestoval nikterak dlouho, ale jako běžný uživatel jsem ho zprovoznit nedokázal. Na instalaci MODx Revolution 2.2.0-pl2 (traditional) vznikla v pořádku položka v menu, pojmenována „VersionX“ a area nastavení „VersionX“. Historie dokumentů se však nikde nezaznamenává a v nastavení nejsou žádné možnosti. Jelikož balík ale pro práci nepotřebuji, nebudu to dále rozvádět.

Přenositelnost by měl mít systém také jednoduchou, bohužel ale po praktickém vyzkoušení se tyto jednoduché postupy stanou velice zdoluhavými a nepříjemnými. Při přenosu stránek to není jen čisté překopírování souborů ze serveru na server a export/import databáze. Je potřeba změnit ještě několik nastavení a postupovat přesně podle návodu [27], což je velice pracné a zdoluhavé.

4.9 ALTERNATIVA

O něco jednodušší způsob vývoje je tedy nechat všechny naše texty v externích souborech na disku a do systému je pak vkládat pomocí jednoho jednoduchého snippetu (pojmenujme ho například „include“) [18]:

```
if (file_exists($file)) {  
    $o = include $file;  
} else { $o = ,File not found at: ,.$file; }  
return $o;
```

Následné volání snippetu:

```
[[!include? &file=`/absolute/path/to/my/snippet.php`]]
```

Tím docílíme toho, že soubory můžeme kdykoliv zálohovat obyčejnou kopií a při potřebě přenosu systému stačí provést novou instalaci, vytvořit v ní dříve zmíněný snippet „include“ a pak jen do každého dokumentu vložit jednu řádku s voláním tohoto snippetu s cestou požadovaného souboru.

Z toho vyplývá jeden problém, a to přenášení objemnějších portálů. Všechny dokumenty totiž musíme vytvořit ručně a do nich importovat správný soubor. To nemusí být vždy zrovna triviální operace. Nicméně při možnosti systém skládat postupně, je to celkem příjemný způsob tvorby.

5 ROZŠIŘUJÍCÍ MODUL „TÉMATA PRACÍ“

Tato aplikace je umístěná na stránkách katedry [29] a má za úkol shromažďovat a nabízet témata pro bakalářské a diplomové práce, oborové projekty a pro předmět Projekt 5.

Do této aplikace vkládají zadavatelé témata prací a tím je nabízejí studentům. Student má pak možnost mezi těmito tématy procházet, popřípadě si téma rovnou i zarezervovat.

5.1 ANALÝZA PROVEDENÍ WEBU KATEDRY

KIV má k dispozici subdoménu univerzity [29], kde prezentuje katedru, jako celek, její členy, výzkum, aktuální dění, informace pro studenty a mnoho dalšího. Vše je rozděleno do sekcí, které zachycuje obrázek 5.1:

- Uchazeči
 - o Zde se nacházejí informace pro zájemce o studium na KIVu, jako jsou záznamy o dnech otevřených dveří a popisy nabízených studijních programů.
- Studenti
 - o Tato sekce obsahuje informace o zahraničních studijních pobytech, o všeobecných povinnostech a možnostech studenta.
 - o Jsou zde zveřejňovány vyhlášky a harmonogramy, týkající se studentů.
 - o Právě zde se nachází modul „Témata projektů“, který je hlavním důvodem vzniku této práce.
- Výzkum
 - o Data o projektech, které na KIVu vznikly a vznikají.
- Katedra
 - o Kontaktní informace katedry.

FAKULTA APLIKOVANÝCH VĚD ZÁPADOČESKÉ UNIVERZITY V PLZNI

<KIV> KATEDRA INFORMATIKY A VÝPOČETNÍ TECHNIKY

English pages - FAV - ZČU | Login

Hledej

Domů Uchazeči Studenti Výzkum Katedra

KATEDRA

- Tradice už od roku 1970
- Individuální přístup
- Výzkumná orientace

STUDIUM

- Moderní studijní obory
- Účast na vědecké práci
- Spolupráce s firmami
- Zahraniční studijní pobyty

VÝZKUM A VÝVOJ

- Domácí i mezinárodní projekty
- Základní i aplikovaný výzkum
- Konference WSCG a TSD

Aktuality

Hovory s informatiky
Ustav informatiky AV ČR, v.v.i. pořádá dne 12. června 2012 v Brně diskuzní seminář v rámci cyklu [Hovory s informatiky](#), na němž vystoupí přední IT osobnosti z České republiky. [2012-04-20]

Stážový program IBM Intership
Firma IBM zahájila výběrové řízení do stážového programu [IBM Intership](#) pro nadcházející letní běh. Studenti mohou získat praxi, rozšířit své znalosti, uplatnit vlastní nápady a poznat práci v oboru svého studia. Přihlášku a více informací najdete na stránkách [IBM Smart University CZ](#). [2012-04-18]

Studentská vědecká konference
Každoročně pořádaná [Studentská vědecká konference](#) se v letošním roce bude konat dne 31.5.2012. Studenti všech vyučovaných oborů bakalářského, magisterského i doktorského studia mohou zasílat své příspěvky do 15.5.2012. Aktivní účast je odměněna mimořádným stipendiem. [2012-04-17]

Nová podoba těchto stránek byla podpořena ESF projektem CZ 1.07/2.3.00/09.0050 Systémová podpora studentů, výzkumných pracovníků a týmů v aplikovaných vědách (SPAV)

esf european social fund in the czech republic INVESTMENTS IN EDUCATION DEVELOPMENT

EUROPEAN UNION MINISTERSTVO ŠKOLNÍHO, MLÁDEŽNÍ A TĚLOVÝCHOVY OP Education and Competitiveness

KIV | Copyright | Webmaster | Design © 2008 Tomáš Krejčí | HTML+CSS2 layout

Aktualizováno 25.07.2011 H.Kalašová

Obrázek 5.1 - Úvodní stránka webu katedry [29].

Tyto stránky jsou v současné době psané právě v systému MODx (který v únoru 2012 nahradil OpenCMS [28]) a je to právě modul „Témata prací“, který jako poslední zbývá přepsat.

5.2 ANALÝZA SOUČASNÉHO MODULU „TÉMATA PRACÍ“

Současný modul pro evidenci témat vznikl v roce 2005 jako semestrální projekt a je psaný v PHP. Obsažená data jsou uložena v samostatné databázi. Od samého začátku modul fungoval separátně od zbytku webu katedry a nikdy nebyl členěn do žádného CMS. Během doby jeho provozu bylo zapotřebí několika menších úprav, které byly řešeny v podobě drobných „záplat“, čímž vznikl velice nepřehledný kód, nevhodný pro další modifikaci.

Grafická podoba výchozí stránky je zachycena na obrázku 5.2.

Obrázek 5.2 - Původní verze modulu „Témata prací“ [29].

Tento modul plní svůj účel a je třeba říci, že většina jeho základních funkcí pracuje jak má. Bohužel zde ale existuje značný počet „mrtvých“ odkazů (HTTP 404) a za dobu jeho provozu vznikl obsáhlý výčet chyb a možných vylepšení. I proto byla zadána tato bakalářská práce a tento modul v ní bude nahrazen.

Funkce modulu jsou odlišné pro zadavatele a pro studenta. Člen KIVu, jehož uživatelský účet má příslušná práva, může vypisovat všechny typy témat pro všechny studijní obory a dále může vlastní témata editovat a mazat. Student může mezi tématy vyhledávat filtrovacím formulářem, vybírat a posílat rezervace (požadavek k zadání tématu).

5.2.1 CHYBY A NEDOSTATKY MODULU

- Hned první největší nepříjemností je neperzistentnost vyhledávacího formuláře. To znamená, že po každém vyhledání se objeví formulář ve výchozím stavu a musíme jej znovu nastavovat. Tento problém se nachází ve vyhledávání témat i v editaci témat.
- Nefungují odkazy „modifikace výpisu“, kde jsou možnosti vypisovat podle prací nebo podle vyučujících.

- Ve vypisovaných tabulkách není uveden akademický rok, pro který je práce vypisána. Tím ztrácí výpis na informační hodnotě.
- Tabulky s výpisy se nedají řadit.
- Autentizace modulu je separovaná od zbytku webu.
- Nelze vypsat externě i interně zadané témata najednou.
- Systém umožňuje vypsat diplomovou práci pro studenta v bakalářském oboru, což je chyba.
- Neexistuje vyhledávání mezi tématy podle klíčových slov.
- Modul neumožňuje vypsat téma pro více studentů.
- O téma se nemůže ucházet více studentů („soutěž“ o jedno téma).
- Při přístupu přihlášené osoby se na stránkách s hromadným výpisem témat nevypisuje seznam právě této osoby.

5.3 NÁVRH NOVÉ ARCHITEKTURY SYSTÉMU

Po vzoru již existujících modulů webu katedry jsem zvolil architekturu, založenou na myšlence systému MVC (Model-View-Controller). Nicméně hranice mezi View a Controller byla odstraněna z jednoho, pro nás podstatného, důvodu. Na práci s vrstvou View, jako samostatnou vrstvou, je MODx velice dobře připravený. K tomu je ale potřeba mít připravené chunky (čisté HTML šablony, které by prezentovali vrstvu View) a z Controlleru je jen spustit s parametry, které potřebujeme vypsat. V kapitole 5.5.5 se ale dozvíme, proč chunky používat nebudeme. Tím pak přicházíme o možnost oddělení vrstev View a Controller. Samozřejmě by bylo možné vrstvy oddělit jen na úrovni dělení do souborů, ale to by jen navýšilo počet souborů, což by znehlednilo souborový systém a nemělo by to větší význam, než oddělení bloku kódu pomocí komentářů.

5.3.1 MODEL

Model je ta část kódu, která se stará o interakci s databází, všechny výpočty, manipulaci s daty a o další aktivní části kódu.

V mém případě je to jeden PHP objekt, který je vkládán do každého souboru. Tento objekt obsahuje funkce, jejichž úkolem je dotazování databáze a navracení výsledků zpět Controlleru. Jsou zde také funkce, které sice s databází nekomunikují, ale jsou používané globálně v celém modulu.

5.3.2 VIEW

Čisté HTML šablony, které převádí informace do podoby, která bude uživatelem čitelná. Je to jediná vrstva, obsahující HTML.

Jak jsme si zdůvodnili výše, tato vrstva bude úzce propojená s vrstvou Controller a popíšeme ji tedy v následující kapitole.

5.3.3 CONTROLLER

Řadič nebo také řídí logika. Jeho úkolem je naslouchat uživateli a reagovat na jeho pokyny. Jako reakci na uživatelskou akci pak upravuje Model nebo View (nebo oboje).

V našem případě je propletený s vrstvou View a tak se skládá jak z dynamického PHP, tak i z grafických prvků (myšleno převážně HTML).

Pro interakci s uživatelem je zvolen způsob přes URL adresu a tedy pole `$_GET[]`. Tento způsob byl zvolen kvůli možnosti uchovávat nebo sdílet odkazy na vypisované stránky [31]. Controller tedy reaguje na změny tohoto pole. Po jeho změně pak začíná komunikaci s Modelem a vypisuje data. Tyto výpisy (vrstva View) jsou ale od Controlleru oddělené jen blokovými komentáři.

5.4 ROLE DATABÁZE

Databáze je pro MODx hlavním úložištěm uživatelských dat (i když jsme si v kapitole 4.9 ukázali, že tomu tak být nemusí), a proto bychom jí měli věnovat i více pozornosti. Tato kapitola se ale netýká databáze - úložiště MODx-u, ale databáze - úložiště dat pro náš modul.

Pro testovací účely byl použit externí MySQL server, nikoliv databáze MODx-u. Změna tohoto rozhodnutí je umožněna skrze nastavení modulu.

V tomto ohledu práce s databází nebyla naši starostí a datový model v následující kapitole byl součástí zadání. Naším úkolem je do této databáze zapisovat data a upravovat je. Z databáze nikdy data neodstraňujeme. Pokud bude chtít zadavatel smazat některé své téma, převede jej pouze do stavu „Smazané“.

5.4.1 DATOVÝ MODEL

Změna datového modelu byla navržena v roce 2009 a verze E-R-A modelu je zveřejněna v příloze C. Jedním ze zadaných požadavků bylo, aby nový modul pracoval již nad tímto modelem.

Při závěrečném ladění modulu jsme zaznamenali absenci atributu v tabulce KIV_STUD_OBORY, která obsahuje informace o studijních oborech. Tento atribut by vyjadřoval, zařazení daného oboru do typu studia. Například obor SWI je v magisterském studiu, nikoliv v bakalářském. To by usnadnilo dynamické mapování typů témat na studijní obory.

Další nutná změna datového modelu by byla potřeba, pokud bychom chtěli splnit požadavek, aby se o téma mohlo ucházet více zájemců a dále aby bylo možné vypisovat témata pro více studentů.

Potřebné změny zde popsané ale nejsou obsahem této bakalářské práce.

5.5 IMPLEMENTACE KOMPONENT PRO MODx - VYVINUTÝ MODUL „TÉMATA PRACÍ“

Jak už jsme několikrát naznačili v textu, jeden z výsledků této práce je i nově vytvořený modul pro evidenci témat prací, který nahradí současnou verzi, popsanou v kapitole 5.2.

Při jeho vývoji je důraz kladen zejména na kompatibilitu se zbytkem webu katedry, snadnou přenositelnost dat, jejich zálohování a zabezpečení. V neposlední řadě musí být modul funkční i po aktualizacích celého systému MODx.

5.5.1 NASTAVENÍ MODULU

Ve složce */rozsirujici_modul_temata_praci/plugins/plugins_ready/tema_simple* na přiloženém CD se nachází soubor *tema_settings.inc.php*, který obsahuje konstantní proměnné, používané k nastavení. Je zde blok s nastavením připojení k databázi, dále je zde možnost nastavit URL aliasy všech dokumentů a poslední blok ovlivňuje názvy tabulek v databázi.

Všechny tyto konstanty jsou přístupné z jakéhokoliv souboru.

5.5.2 POPISY STRÁNEK

Název stránky	Jméno souboru
Výchozí stránka	<i>/tema_simple.inc.php</i>
Detail tématu	<i>/tema_simple_detail.inc.php</i>
Editační stránka pro hromadnou úpravu	<i>/edit/tema_edit_all.inc.php</i>
Stránka pro přidání nového tématu	<i>/add/tema_add.in.php</i>
Editační stránka pro úpravu jednoho tématu	<i>/edit/tema_edit_one.inc.php</i>

Tabulka 5.1 - Spojení stránek se soubory.

Jedna stránka (myšleno jako část vrstvy View/Controller) je vždy obsažena pouze v jednom souboru. Propojení stránek se soubory je znázorněno v tabulce 5.1. Tyto soubory jsou zveřejněné na přiloženém CD, ve složce */rozsirujici_modul_temata_praci/plugins/plugins_ready/tema_simple*.

Pro lepší orientaci na stránkách vznikla uživatelská dokumentace, zveřejněná v příloze B.

VÝCHOZÍ STRÁNKA

Obsahuje postupy a rady pro studenty, jeden vyhledávací formulář a na něj navázaný výpis témat, organizovaný do tabulek.

DETAIL TÉMATU

Obsahuje veškerá známá a veřejně dostupná data uložená v databázi, která se týkají daného tématu. Například údaje o garantovi, název tématu, popis a mnohá další. Pokud je téma ve stavu „volné“, pak je na stránce zobrazen i rezervační formulář, pomocí kterého může zájemce kontaktovat přímo garanta tématu.

STRÁNKA PRO PŘIDÁNÍ NOVÉHO TÉMATU

Stránka pouze pro přihlášené uživatele. Obsahuje vstupní formulář (znázorněný v uživatelské dokumentaci v příloze B), pomocí kterého se запиše nové téma do databáze.

EDITAČNÍ STRÁNKA PRO HROMADNOU ÚPRAVU TÉMAT

I tato stránka je chráněna heslem. Vyskytuje se zde stejný filtr, jako ve stránce výchozí. Výpisy však obsahují jen témata vypsaná právě přihlášenou osobou. Navíc jsou zde možnosti pro hromadné úpravy témat. Za pomoci akčních tlačítek pod každou neprázdnou tabulkou s tématy, která jsou spjatá se sloupečkem „Upravit“ dané tabulky, můžeme témata hromadně upravovat:

- Po filtrování volných témat se zobrazí tlačítko pro hromadný přesun témat na aktuální akademický rok.
- Po filtrování rezervovaných můžeme témata hromadně označit za dokončená.
- Po filtrování zadaných témat pak za dokončená.
- A po filtrování smazaných témat můžeme témata označit za volná.
- V každém předchozím stavu filtru lze témata hromadně mazat.

EDITAČNÍ STRÁNKA PRO ÚPRAVU JEDNOHO KONKRÉTNÍHO TÉMATU

Je také chráněna heslem. Obsahuje stejný formulář, jako stránka pro přidávání nových témat, avšak prázdná pole jsou nahrazena předvyplněnými. Ta obsahují data konkrétního tématu.

5.5.3 ZMĚNY OPROTI PŘEDCHOZÍ VERZI MODULU

Nový modul přebírá všechny funkce modulu starého a navíc by v něm měly být vyřešeny všechny chyby a některé nedostatky, které jsou popsány v kapitole 5.2.1.

Z požadavků modul splňuje:

- Perzistence formulářů pro vyhledávání témat.
- Všechny odkazy jsou funkční.
- Do vypisovaných tabulek byl přidán sloupec „Pro akad. rok“.
- Tabulky lze řadit podle názvu sloupců.
- Do filtrovacího formuláře byla přidána možnost pro vypisování témat s externími i interními zadavateli dohromady.

- Byly namapované studijní obory na typ práce a nelze tak zadat například diplomovou práci pro studenta v bakalářském oboru.
- Pokud se na stránky dostane přihlášená osoba, defaultní výpisy výchozí stránky a stránky pro hromadnou úpravu témat se změní na vyfiltrovaný seznam témat na aktuální akademický rok, která jsou vypsána právě touto osobou.
- Na editaci jednoho tématu se lze dostat přímým odkazem z jeho detailu. (Funkce pouze pro přihlášené uživatele.)

5.5.4 POPIS POUŽITÝCH FUNKCIONALIT

PERZISTENCE FORMULÁŘŮ

Význam a místa použití

Po použití vyhledávacího formuláře chceme, aby tento formulář zachoval své nastavení a nevykresloval se vždy ve výchozím stavu.

Řešení najdeme na výchozí stránce a na stránce pro hromadnou úpravu témat.

Realizace

Perzistence formulářů je zrealizována pomocí JavaScriptu, který je spuštěn ihned po jeho vykreslení. S pomocí knihovny jQuery [32] nastavíme například položku „Autor:“ podle proměnné obsažené v URL adrese tímto příkazem:

```
echo "$('#form_autor').val('".$_GET['autor']. "')";
```

KONTROLA ODESLANÉHO FORMULÁŘE

Význam a místa použití

Při vytváření nebo editaci tématu zadává uživatel některá data do formuláře ručně. Tato data je doporučováno vždy pečlivě kontrolovat [31].

Realizace

Tato kontrola probíhá pomocí rozsáhlého kódu v jazyce JavaScript (k nahlédnutí například v souboru *tema_add.inc.php*, který je k nahlédnutí ve složce *rozsirujici_modul_temata_praci/plugins/plugins_ready/tema_simple/add/*) a po

úspěšném splnění této kontroly ještě probíhá testování obsahu ve vrstvě Modelu při zápisu do databáze.

DEFAULTNÍ VÝPIS VLASTNÍCH TÉMAT NA AKTUÁLNÍ AKADEMICKÝ ROK PRO PŘIHLÁŠENÉ

Význam a místa použití

Při přístupu přihlášené osoby na výchozí stránku nebo na stránku pro hromadnou úpravu témat se automaticky vypíše vyfiltrovaný seznam témat na aktuální akademický rok, vypsaný danou osobou.

Realizace

Pokud na začátku stránky zaregistrujeme přihlášeného uživatele, ihned režii stránky přesměrujeme na stejnou URL adresu s vyplněnými proměnnými. K přesměrování slouží následující kód:

```
// $id je číslo přihlášeného uživatele
$act_aca_year = $tema_simple->get_actual_academic_year();
$this_document_id = $modx->resource->get('id');
$url_param = array( 'autor' => $id, 'branch' => '%', 'state' => '%',
    'type' => '%', 'year' => $act_aca_year, 'extern' => '%' );
$url = $modx->makeUrl($this_document_id, "web", $url_param, "http");
$modx->sendRedirect($url);
```

V tomto kódu je vidět například práce s vrstvou Model, která nám vrací aktuální akademický rok. Dále stojí za povšimnutí pole s parametry, které si MODx přeloží do URL adresy funkcí `makeUrl()`. Konstanta '%' v jazyce SQL zastupuje libovolný počet znaků.

ŘAZENÍ TABULEK PODLE NÁZVU SLOUPCE

Význam a místa použití

Ve výchozí stránce a ve stránce pro hromadnou editaci témat se seznamy vypsaných témat organizují do tabulek. Tyto tabulky lze řadit kliknutím na název sloupce. Se stisknutou klávesou Shift se dá tabulka řadit i podle více sloupců.

Realizace

K této funkci je použita jQuery knihovna Tablesorter. Detailní podrobnosti o použití této knihovny lze najít pod odkazem v seznamu použitých zdrojů [30].

NAMAPOVÁNÍ TYPU PRÁCE NA STUDIJNÍ OBOR

Význam a místa použití

Při vytváření nebo editaci tématu je třeba zamezit situaci, kdy zadavatel vypíše například bakalářskou práci pro studijní obor magisterského studia.

Realizace

Opět je zde použita knihovna jQuery [32], která hlídá změny nad položkou „Typ“. Jakmile se změní, zjistí id nově vybraného typu a podle dvourozměrného pole najde seznam povolených oborů. S touto znalostí projde všechny možnosti oborů a u kterého se jeho id nevyskytuje v seznamu, toho skryje.

Po každé změně pak označí za vybraný obor s posledním id, který je pro všechny typy stejný - „n/a“.

S ohledem na velký objem kódu je konkrétní provedení k nahlédnutí jen ve zdrojových kódech na přiloženém CD - například v souboru *tema_add.inc.php*, který je uložen ve složce *rozsirujici_modul_temata_praci/plugins/plugins_ready/tema_simple/add/*.

PRÁCE S DATABÁZÍ

Význam a místa použití

Data z databáze používáme naprosto všude, jen převážná menšina textů je vypisovaná staticky.

Realizace

Každý soubor (mimo souboru s nastavením a souboru s vrstvou Model) na prvních řádkách kódu integruje soubor *tema_simple.class.php* ze složky *rozsirujici_modul_temata_praci/plugins/plugins_ready/tema_simple/* a vytváří si instanci

objektu `tema_simple`, který se v tomto souboru nachází. Skrze tuto proměnnou pak můžeme přistupovat ke všem funkcím vrstvy Modelu, které už přímo komunikují s databází. Například:

```
$tema_list = $tema_simple->get_tema_list_by_type($i);
```

Zajistí, že v proměnné `$tema_list` bude uloženo asociativní pole se všemi tématy, která vyhovují filtraci podle proměnných v URL a která mají typ roven `$i`.

5.5.5 ZÁSADY PŘI TVORBĚ MODULŮ PRO WEB KATEDRY KIV

Pro tvorbu tohoto modulu a obecně všech modulů pro stránky KIVu je podstatné držet se standardů a postupů, ve kterých je psán zbytek webu, do kterého se bude modul integrovat. Tyto standardy nejsou vždy zcela srozumitelné a „standardní“, proto zde uvedeme nejdůležitější příklady pravidel. Takto si ozřejmíme některé složitější formulace, které se v modulu vyskytují.

1) Nový web katedry v systému MODx je vytvářen stylem vkládání externích souborů, který je popsán v kapitole 4.9. Z tohoto důvodu je třeba psát modul s použitím co nejmenšího počtu dokumentů. Ty se totiž musí při integraci vytvářet ručně a propojení s modulem nastává:

- Použitím správného parametru pro snippet include (který je popsán v kapitole 4.4.5).
- Správně zvoleným atributem `[[*alias]]`, který je používán pro adresování dokumentu uvnitř vkládaných souborů.

Proto by měl být v ideálním případě s modulem vytvořen také návod na implementaci modulu. Samozřejmě by se této zásadě neměl nijak přizpůsobovat rozsah balíku. Toto doporučení navrhujeme pouze s ohledem na méně náročnou integraci modulu.

2) Jak již bylo zmíněno, jednotlivé dokumenty budou adresovány za pomoci jejich atributu `[[*alias]]`. Je to cesta delší, zato univerzálnější. Při tvorbě dokumentu je totiž id generováno automaticky jako obyčejné zvyšující se (auto increment) číslo. Jeho hodnotu předem vědět nebudeme, zatímco atribut `alias` je povinný a ten ovlivnit dokážeme. Pro příklad zde uvedu dva způsoby pro výpis odkazu na dokument s `id=6` a URL aliasem „temata“.

- Verze při adresování za pomoci id:

```
echo '<a href="[[~6]]">text odkazu</a>';
```

- Verze při adresování za pomoci aliasu:

```
$document_temata = $modx->getObject('modResource',array(
    'published' => 1,
    'alias' => 'temata'));
$id = $document_temata->get('id');
echo '<a href="[[~'. $id. ']]">text odkazu</a>';
```

5.5.6 VZHLED STRÁNEK

V této kapitole je třeba říci, že veškeré grafické práce na modulu (CSS styly a rozložení stránek) byly přebrány z již existující šablony. Úmyslně jsme nevytvářeli nic kreativního, aby integrace do zbytku webu byla co nejméně nápadná.

5.6 KONVERZE DAT, POUŽÍVANÝCH MODULEM

Potřebná testovací data nám byla poskytnuta v podobě standardního exportu databáze, se kterou pracoval předchozí modul. (Jak již bylo zmíněno v kapitole 5.2, tato databáze byla kompletně separována od zbytku webu.)

Tato data nejsou kompatibilní s novým datovým modelem a ani s používaným kódováním. Proto je třeba data upravit. Tato úprava v praxi znamená přepsání struktury tabulek, jejich kódování a data pak do těchto tabulek nahrát.

S ohledem na to, že tabulka měla v původní verzi jen 19 sloupců a nová podoba tabulky má sloupců 25, vznikají tak prázdné sloupce. Některé z nich lze zaplnit například editací tématu, ale většinu z nich využijeme až u nově vzniklých témat.

Úprava kódování se řeší hromadným nahrazením problematických znaků v textovém editoru (tuto funkci obsahuje například PSPad).

5.7 TESTOVÁNÍ NOVĚ VYVINUTÉHO MODULU

Modul to není natolik rozsáhlý, aby si zasloužil speciální testovací dokumentaci v příloze, proto se zde zmíníme jen o základních opatřeních.

Testování modulu probíhalo nejintenzivněji během vývoje po částech. Chyby ve vrstvě Controller/View se hledaly velice snadno, protože většinou přímo ovlivňovaly

grafický výstup. Naproti tomu vrstva Model skrývá mnoho funkcí a většina z nich obsahuje MySQL příkazy pro komunikaci s databází. To je hlavním důvodem, proč byla tato vrstva vystavena převážně většině testů.

Všechny testy probíhali v prohlížečích Google Chrome 18.0.1025.162 m, Internet Explorer 8.0.7601.17514 a Mozilla Firefox 11.0 a ve všech je modul použitelný a funkční.

S ohledem na to, že specifikace zadavatele nebyla příliš konkrétní, většina testů byla prováděna jen na základě zdravého úsudku zpracovatele úkolu.

5.7.1 TESTY NAD VRSTVOU CONTROLLER/VIEW

V této vrstvě může nastat problém jen v komunikaci mezi uživatelem a systémem. Proto jsme zkoušeli:

- Vkládat testovací témata s různě vyplněnými i nevyplněnými údaji.
- Na stránky jsme přistupovali pod různými URL (které jsme určili pro vrstvu Controller/View jako zásadní).

Objevené chyby byly v programu opraveny.

5.7.2 TESTY NAD VRSTVOU MODEL

Tato vrstva - jak již bylo zmíněno výše - nám umožňuje komunikaci s databází. Funkce po zavolání provedou MySQL příkaz nad databází a vrací výsledky k výpisu. Pro testování MySQL příkazů bylo použito aplikaci phpMyAdmin 3.3.9. Byla vytvořena malá testovací databáze, abychom mohli porovnávat výsledky všech příkazů.

6 ZÁVĚR

V této práci jsme se snažili vyzdvihnout základní vlastnosti systému MODx a hlavně jeho všestrannost použití. S ohledem na požadavky v zadání práce byl vytvořen jen nástin jeho možností, který však nepostačuje ke komplexní práci se systémem MODx.

Představili jsme si všechny komponenty MODx-u a ukázali dva způsoby, jak s ním pracovat při vytváření webových aplikací.

Dále jsme dle zadání vytvořili rozšiřující modul pro evidenci témat, na kterém jsme provedli demonstraci základních pravidel pro vývoj doplňků, která je třeba dodržovat při práci pro web KIVu. Testování modulu bylo provedeno na datech, která byla získána zkonvertováním dat z dřívější verze aplikace.

Testováním bylo zjištěno, že navržený produkt je plně funkční a bez větších úprav jej lze aplikovat v praxi.

POUŽITÉ POJMY A ZKRATKY

AJAX	Asynchronous JavaScript and XML
BSD	Berkeley Software Distribution
CMF	Content Management Framework
CMS	Content Management System, překlad doslova: Systém pro správu obsahu, někdy se používá název redakční systém nebo publikační systém
E-R-A	Entity-Relationship-Attribute
GPL	General Public License
HTTP 404	chybový kód, který se vrací po nenalezení stránky
KIV	Katedra informatiky a výpočetní techniky
MIT	Licence, která vznikla na Massachusetts Institute of Technology
MVC	Model-View-Controller
ORB	Object-relational bridge
SEO	Search Engine Optimization, tedy optimalizace pro vyhledávače
SQL	Structured Query Language
URL	Uniform Resource Locator
WEB	World Wide Web (www) se někdy zkracuje jako: web; v této práci je ale zkratka použita ve významu: internetová stránka
WYSIWYG	What You See Is What You Get, tedy editor, který převádí vzhled dokumentu na formátovací značky
xPDO	eXtension PDO
XSS	Cross-site scripting je metoda narušení WWW stránek využitím bezpečnostních chyb ve skriptech

ZDROJE

K této kapitole je třeba podotknout, že MODx je systém relativně nový a byl začleněn do trhu, plného již starších a dobře fungujících CMS. To je zřejmě hlavní důvod, proč ještě nevznikla žádná česky psaná publikace na téma MODx-u. V období vzniku tohoto dokumentu existovaly knihy pouze 3. Jedna je psána japonsky a další 2 psané anglicky. S ohledem na jejich vysokou cenu jsou pro studenta vysoké školy nedostupné. Nicméně pan Bob Ray, který vydal poslední známou publikaci o systému MODx, zveřejnil část podstatných informací ze své knihy na svém blogu <http://bobsguides.com/>. Z toho bylo v tomto dokumentu často čerpáno.

- [1] Co je CMS / Redakční systém? *CMSSystémy* [online]. [cit. 2011-12-20]. Dostupné z: <http://www.cms-systemy.cz/o--portali/co-je-cms-redakcni-system--/>
- [2] Systém pro správu obsahu. *Wikipedie: otevřená encyklopedie* [online]. 2001-2012 [cit. 2011-12-20]. Dostupné z: http://cs.wikipedia.org/wiki/Syst%C3%A9m_pro_spr%C3%A1vu_obsahu
- [3] List of content management systems. *Wikipedia: the free encyclopedia* [online]. 2001-2012 [cit. 2011-12-20]. Dostupné z: http://en.wikipedia.org/wiki/List_of_content_management_systems
- [4] MODx. In: *Wikipedia: the free encyclopedia* [online]. 2001-2012 [cit. 2011-12-20]. Dostupné z: <http://en.wikipedia.org/wiki/MODx>
- [5] Background. *MODx* [online]. 2005-2012 [cit. 2011-12-20]. Dostupné z: <http://modx.com/about/media-center/background/>
- [6] Czech translation. *MODx* [online]. 2005-2012 [cit. 2011-12-20]. Dostupné z: <http://forums.modx.com/thread/51175/czech-translation-0-6>
- [7] MODX Evolution. *MODx* [online]. 2005-2012 [cit. 2011-12-20]. Dostupné z: <http://modx.com/software/open-source-cms/modx-evolution/>
- [8] MODX Revolution. *MODx* [online]. 2005-2012 [cit. 2011-12-20]. Dostupné z: <http://modx.com/software/open-source-cms/modx-revolution/>

- [9] Tag Syntax. *MODx Documentation* [online]. 2005-2012 [cit. 2011-12-20].
Dostupné z: <http://rtfm.modx.com/display/revolution20/Tag+Syntax>
- [10] The MODx CMS. *Bob's Guides* [online]. 2011 [cit. 2011-12-20]. Dostupné z:
<http://bobsguides.com/modx.html>
- [11] How MODx Templates Work. *MODx From the Inside* [online]. 2008-2012
[cit. 2011-12-20]. Dostupné z: <http://sottwell.com/how-templates-work.html>
- [12] Resources. *MODx Documentation* [online]. 2005-2012 [cit. 2011-12-20].
Dostupné z: <http://rtfm.modx.com/display/revolution20/Resources>
- [13] Templates. *MODx Documentation* [online]. 2005-2012 [cit. 2011-12-20].
Dostupné z: <http://rtfm.modx.com/display/revolution20/Templates>
- [14] Template Variables. *MODx Documentation* [online]. 2005-2012 [cit. 2011-12-20].
Dostupné z: <http://rtfm.modx.com/display/revolution20/Template+Variables>
- [15] Creating a Template Variable. *MODx Documentation* [online]. 2005-2012
[cit. 2011-12-20]. Dostupné z:
<http://rtfm.modx.com/display/revolution20/Creating+a+Template+Variable>
- [16] Chunks. *MODx Documentation* [online]. 2005-2012 [cit. 2011-12-20].
Dostupné z: <http://rtfm.modx.com/display/revolution20/Chunks>
- [17] MODx Chunks. *Bob's Guides* [online]. 2011 [cit. 2011-12-20]. Dostupné z:
<http://bobsguides.com/modx-chunks.html>
- [18] Snippets. *MODx Documentation* [online]. 2005-2012 [cit. 2011-12-20].
Dostupné z: <http://rtfm.modx.com/display/revolution20/Snippets>
- [19] MODx Snippets. *Bob's Guides* [online]. 2011 [cit. 2011-12-20]. Dostupné z:
<http://bobsguides.com/modx-snippets.html>
- [20] Category:Snippets. *MODx Wiki* [online]. 2008 [cit. 2011-12-20]. Dostupné z:
<http://wiki.modxcms.com/index.php/Category:Snippets>
- [21] Plugins. *MODx Documentation* [online]. 2005-2012 [cit. 2011-12-20].
Dostupné z: <http://rtfm.modx.com/display/revolution20/Plugins>

- [22] System Events. *MODx Wiki* [online]. 2011 [cit. 2012-02-05]. Dostupné z: http://wiki.modxcms.com/index.php/System_Events
- [23] Object-relational mapping. *Wikipedia: the free encyclopedia* [online]. 2012 [cit. 2012-02-05]. Dostupné z: http://en.wikipedia.org/wiki/Object-relational_mapping
- [24] XPDO 2.0. *MODx Documentation* [online]. 2005-2012 [cit. 2012-02-05]. Dostupné z: <http://rtfm.modx.com/display/xPDO20/Home>
- [25] API. *MODx Documentation* [online]. 2005-2012 [cit. 2012-02-05]. Dostupné z: <http://api.modx.com/revolution/2.1/>
- [26] VersionX. *MODx Documentation* [online]. 2012-01-23 [cit. 2012-04-13]. Dostupné z: <http://rtfm.modx.com/display/ADDON/VersionX>
- [27] Moving Your Site to a New Server. *MODx Documentation* [online]. 2011-12-30 [cit. 2012-04-13]. Dostupné z: <http://rtfm.modx.com/display/revolution20/Moving+Your+Site+to+a+New+Server>
- [28] WebKiv. *Wiki KIVu* [online]. 2006 [cit. 2012-04-22]. Dostupné z: <http://wiki.kiv.zcu.cz/WebKiv/HomePage>
- [29] *Katedra Informatiky a výpočetní techniky* [online]. [cit. 2012-04-22]. Dostupné z: <http://www.kiv.zcu.cz/>
- [30] *Tablesorter: JQuery plugin* [online]. [cit. 2012-04-23]. Dostupné z: <http://tablesorter.com/docs/>
- [31] BORONCZYK, Timothy, Elizabeth NARAMORE, Jason GERNER, Yann Le SCOUARNEC, Jeremy STOLZ a Michael K. GLASS. *PHP 6, MySQL, Apache: Vytváříme webové aplikace*. Brno: Computer Press, a.s., 20011. ISBN 978-80-251-2767-4
- [32] *JQuery Cookbook: Solutions & Examples for jQuery Developers*. United States of America: O'Reilly Media, Inc., 2009. ISBN 978-0-596-15977-1.
- .

PŘÍLOHA A - ZÁKLADNÍ A BĚŽNĚ POUŽÍVANÁ ŠABLONA

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="cs" lang="cs">
  <head><!-- ščřžýščřžý -->
    <meta name="description" content="[[*description]]"/>
  <!-- vlastnost dokumentu -->

  <base href="[[!++site_url]]" /> <!-- nastavení systému -->

  <title>[[++site_name]] - [[*pagetitle]]</title>
  <!-- vlastnost dokumentu -->

</head>
<body>

  <div id="page_whole">

    <div id="page_left_menu">

      <div id="flag">

        [[BabelLinks]] <!-- snippet -->

      </div> <!-- end of flag -->

      [[!Wayfinder? &level=`3` &rowTpl=`cssplay_row`]]
  <!-- necachovaný snippet s parametry -->

    </div> <!-- end of page_left_menu -->
    <div id="page_right_content">

      [[*content]] <!-- vlastnost dokumentu -->

    </div> <!-- end of page_right_content -->
    <div id="page_footer">

      [[\$footer]] <!-- chunk -->

    </div>
  </div> <!-- end of page_whole -->
</body>
</html>
```

PŘÍLOHA B - UŽIVATELSKÁ DOKUMENTACE

Modul byl stavěn s největším důrazem na jednoduchost ovládání a na intuitivní pohyb v něm pro každého uživatele.

B.1 POŽADAVKY

Modul je funkční v každé instalaci MODx-u Revolution 2.1.3-pl a novější (starší verze nebyly testovány) a systémové požadavky na nejnovější verzi Revolution 2.2.1-pl jsou:

- Apache 2.2 nebo IIS7
- MySQL 5.1
- PHP 5.3 (s povoleným PDO)

Je možné se v praxi setkat s nutností nestandardního nastavení serveru, kde je třeba mimo jiné vypnout funkci `safe_mode`. Proto není pro instalaci MODx-u vhodný každý webhostingový poskytovatel a je dobré si před instalací zjistit požadavky aktuální verze MODx-u a nabídky a možnosti poskytovatelů.

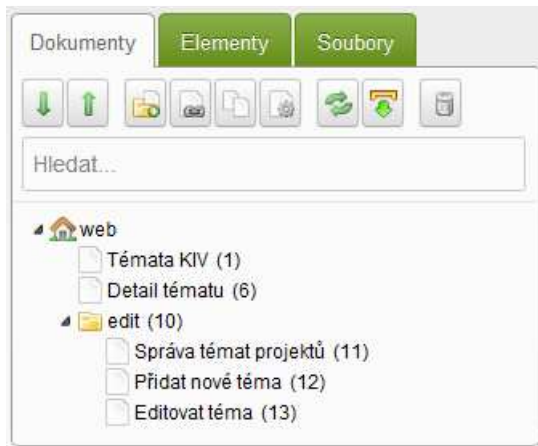
B.2 INSTALACE

Tento krok je určen jen pro administrátora stránek!

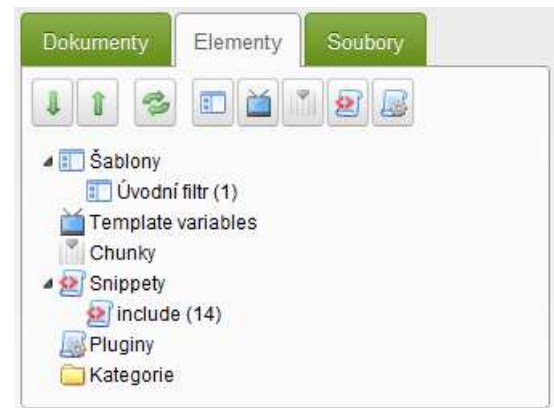
Pro instalaci systému MODx navštivte prosím oficiální stránky výrobce. Tento návod se týká pouze instalace zde popisovaného doplňkového modulu.

Složky se zdrojovými soubory ze složky *rozsirujici_modul_temata_praci* na přiloženém CD stačí překopírovat do kořenového adresáře MODx-u a do instalace doplnit všechny dokumenty a elementy, zobrazené na následujících obrázcích B.1 a B.2.

Strom dokumentů a elementů vypadá následovně (v kategorii edit jsou dokumenty, spadající pod zaheslovanou část aplikace):



Obrázek B.1 - Strom dokumentů.



Obrázek B.2 - Strom elementů.

Na obrázku B.2 stojí za povšimnutí, že aplikace ke svému provozu potřebuje pouze jeden snippet „include“ a jednu šablonu (v tomto případě pojmenovanou „Úvodní filtr“). Snippet „include“ i s jeho zdrojovým kódem jsou popsány v kapitole 4.9.

Povinným parametrem každého dokumentu je název a URL alias. Atribut „název“ se zobrazuje ve výstupu, ale nijak nezasahuje do funkčnosti aplikace. Zatímco URL alias musí být nastaven přesně dle výpisu níže. I při malé syntaktické chybě nebude fungovat většina odkazů v celém modulu. Do obsahu každého dokumentu musí být vepsána jedna řádka a ta musí striktně souhlasit:

- **Témata KIV** (URL alias = „temata“):

```
[[!include?
&file=`plugins/plugins_ready/tema_simple/tema_simple.inc.php`]]
```

- **Detail tématu** (URL alias = „detail-tematu“):

```
[[!include?
&file=`plugins/plugins_ready/tema_simple/tema_simple_detail.inc.php`]]
```

- **Správa témat projektů** (URL alias = „sprava-temat-projektu“):

```
[[!include?
&file=`plugins/plugins_ready/tema_simple/edit/tema_edit_all.inc.php`]]
```

- **Přidat nové téma** (URL alias = „pridat-nove-tema“):

```
[[!include?
&file=`plugins/plugins_ready/tema_simple/add/tema_add.inc.php`]]
```

- **Editovat téma** (URL alias = „editovat-tema“):

```
[[!include?
&file=`plugins/plugins_ready/tema_simple/edit/tema_edit_one.inc.php`]]
```

Všechny aliasy je možné změnit. Tato změna se ale musí provést navíc i v souboru */rozsirujici_modul_temata_praci/plugins/plugins_ready/tema_simple/tema_settings.inc.php*, který obsahuje proměnné nastavení. V tomto souboru je pak třeba nastavit i **údaje pro přihlášení do databáze!**

Po tomto nastavení databáze MODx-u a zdrojových souborů je ještě třeba doplnit data do databáze. To provedeme importem MySQL souboru *rozsirujici_modul_temata_praci/kiv_tema_simple_db.sql*, přiloženým na CD.

Po provedení všech výše zmíněných akcí a nastavení bude modul fungovat.

B.3 PŘEHLED FUNKCÍ

B.3.1 PRÁCE S MODULEM V ROLI STUDENTA

Student má možnosti listovat v seznamu témat, filtrovat ho a číst si jejich detaily jednotlivých témat.

Filtrovací formulář má jednoznačně pojmenované položky, jediné co stojí za vysvětlení, je položka určující původ vzniku. Téma může být vypsáno externím subjektem (právníčkou nebo i fyzickou osobou) nebo může být vypsáno interně, tzn. zaměstnancem KIVu.

Po vybrání konkrétního tématu se zobrazí detail, a pokud je to detail volného tématu, zobrazí se zde i rezervační formulář.

Rezervace tématu:

Rezervace proběhne vyplněním jednoduchého formuláře, zobrazeného na obrázku B.3. Kontrola formuláře probíhá pouze testem na prázdná pole, takže pokud bude zadána chybná e-mailová adresa, garant tématu nebude mít potřebný kontakt na zájemce. Je tedy v zájmu zájemce, vypsát data odpovídající realitě.

Po odeslání formuláře přijde garantovi e-mail s informacemi o zájemci.

Rezervace tématu

Toto téma je zatím volné. Pokud o téma máte vážný zájem, vyplňte prosím následující formulář, kterým si téma zamluvíte (všechny položky jsou povinné).

Jméno a příjmení:

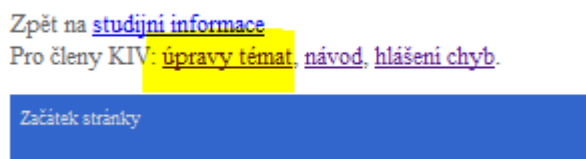
Studijní číslo: Email:

Obrázek B.3 - Rezervační formulář.

B.3.2 PRÁCE S MODULEM V ROLI ZADAVATELE

Zadavatel má práva téma vypsát, zadat, označit za dokončená, popřípadě i smazat. Veškeré změny smí provádět pouze na svých tématech.

Hromadná úprava vlastních témat:



Obrázek B.4 - Odkaz na hromadnou úpravu témat.

Přes odkaz dole na hlavní stránce témat „úpravy témat“ (který je zobrazen na obrázku B.4, ale je možná změna vzhledu při integraci modulu do celého webu!) se dostaneme do editačního formuláře. Ten je téměř shodný s defaultní stránkou modulu.

Pod neprázdnými tabulkami s výpisem témat najdeme tlačítka akcí, která jsou úzce spjatá s právě jednou tabulkou a jejím sloupečkem „Upravit“.

Projekt 5

2 záznamů

Název	Studijní obor	Stav	Pro akad. rok	Upravit
Nástroje pro správu požadavků	bc-INF	Volné	2009/2010	<input type="checkbox"/>
Testování kvality webových aplikací	bc-INF	Volné	2009/2010	<input type="checkbox"/>

Vybraná témata:

Obrázek B.5 - Hromadná úprava volných témat.

Po výpisu všech volných témat je možné tato témata hromadně převést na aktuální akademický rok (viz. Obrázek B.5).

Po výpisu všech rezervovaných témat je možné témata zadat jejich zájemcům.

Po výpisu všech zadaných témat je možné tato témata označit za dokončená.

Po výpisu všech dokončených témat nebo při jakémkoliv jiném stavu je vždy možné hromadně témata mazat.

Vypsání nového tématu:

Přidat nové téma

Kdy má být zveřejněno: rok měsíc den

Pro akademický rok:

Typ práce:

Studijní obor:

Název tématu:

Popis tématu:

URL k tématu:

Externím konzultantem k tématu je z firmy

Stav: Volné Rezervované Zadané Dokončené

Zájemce:

Jméno a příjmení: os. číslo: e-mail:

Odevzdáno dne: rok měsíc den

Obrázek B.6 - Vstupní formulář pro vypsání nového tématu.

Na stránce pro hromadnou úpravu témat (na kterou jsme se dostali přes odkaz „úpravy témat“, zobrazený na obrázku B.4) je v horní části stránky umístěné tlačítko „Přidat nové téma“. Toto tlačítko nás přesměruje na formulář pro vypsání nového tématu, který je zachycen na obrázku B.6.

Formulář má předvyplněné datum zveřejnění a pro úspěšné vypsání je ještě povinná položka „Název tématu:“. Celý formulář je dobře ošetřen proti chybám vstupu a uživatele vždy na chybu upozorní ihned po pokusu o odeslání. V případě prázdného popisu se pouze dotazuje, je-li popis nevyplněn úmyslně.

Editace vlastního tématu:

Do editačního formuláře nad jedním konkrétním tématem se dostaneme kliknutím na název tématu, které si vyfiltrujeme na stránkách pro hromadnou úpravu. (Na tyto stránky se dostaneme přes odkaz „úpravy témat“, zachycený na obrázku B.4.) Jde o téměř stejný formulář, jako při vytváření nového tématu (který můžete vidět na obrázku B.6), jen je předvyplněný.

Kontrola vstupů probíhá stejně jako při vkládání nového tématu a to po stisknutí tlačítka „Opravit údaje“.

PŘÍLOHA C - DATOVÝ MODEL

