

STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST
2008 / 2009
OBOR 18 - INFORMATIKA

V3-CMS

System správy obsahu

Adam Staněk, Jiří Zavadil

<http://soc.v3net.cz>

Gymnázium, Praha 6, Arabská 14
Arabská 14, 160 00, Praha 6
Hlavní město Praha

Čestné prohlášení autorů

Prohlašujeme tímto, že jsme soutěžní práci vypracovali zcela samostatně a dodrželi jsme autorského práva všech našich zdrojů, které jsou pravdivě uvedeny.

V Praze 23. března 2009

Adam Staněk

Jiří Zavadil

Anotace

V3-CMS je webová platforma pro budování a správu webových projektů. Obsahuje nástroj pro administraci projektů a soubor knihoven sloužící k efektivnímu generování webových rutin. Následující dokumentace popisuje činnost systému V3-CMS a ukazuje jeho reálné nasazení na jednom z projektů.

Motivace

Po více než pětileté praxi v oblasti webových technologií a práci s několika interními systémy webových studií jsem došel k závěru, že v současné době stále ještě neexistuje kvalitní systém správy obsahu (dále CMS = Content Management System). Trh doslova překypuje spoustou open-sourcových řešení a řešení psaných na koleni, každému však něco chybí. Mým cílem bylo vytvořit konkurence schopné řešení, pro nasazení ve stávajících projektech a usnadnění implementace projektů budoucích.

Annotation

V3-CMS is a web platform for building and administration of web projects. It contains a tool for an administration of projects and a set of libraries serving an effective generation of web routines. The following documentation describes the operation of system V3-CMS and displays its real implementation within one of the projects.

Motivation

After more than a five-year practice in the area of web technologies and work with several internal systems of web studios, I have come to a conclusion that nowadays there still does not exist a quality content management system (CMS). The market obviously abounds in the number of open-source solutions and for the occasion solutions, yet there is something lacking in all of them. My goal was to create a competitive solution to be put in practice in the current projects and to facilitate the implementation of future projects.

Obsah

Čestné prohlášení autorů	2
Anotace	3
Motivace.....	3
Annotation.....	3
Motivation	3
Obsah.....	4
Vysvětlení pojmů.....	5
Analýza problému.....	7
Specifikace požadavků.....	8
Bližší pohled na návrh.....	9
Návrh	10
PHP Smarty	11
Administrace.....	13
Implementace.....	16
Testování	17
Zhodnocení.....	18
Porovnání s konkurenčními systémy.....	19
Seznam použité literatury	20
Seznam příloh.....	21

Vysvětlení pojmů

AJAX	„ <i>Asynchronous JavaScript And XML</i> “ – souhrnný název pro způsob komunikace mezi prohlížečem a serverem pomocí XML a JavaScriptu
API	„ <i>Application Programming Interface</i> “ – sbírka funkcí frameworku, které jsou využitelné zvenčí
Aplet	Samostatná část aplikačního celku
BB kód	Odlehčený webový jazyk používaný zejména na diskuzních fórech
C++	Programovací jazyk administrace
Captcha	Druh ochrany proti robotů založený na opisování textu z obrázku
CMS	„ <i>Content Management System</i> “ – Systém správy obsahu
CSS	„ <i>Cascading StyleSheet</i> “ – jazyk stylopisů používaných spolu s jazykem HTML k formátování webových stránek
DB	Databáze
Deployment	Proces finalizace stabilní verze aplikace a distribuce ke klientům
Desktop	Souhrné označení pro pracovní prostředí uživatele
ePlatby	Platební systém Raiffeisenbank a.s.
Framework	Aplikační vrstva, často soubor knihoven a funkcí
GTK	„ <i>Gimp ToolKit</i> “ – framework pro tvorbu GUI
GUI	„ <i>Graphical User Interface</i> “ – grafické uživatelské rozhraní
HTML	„ <i>HyperText Markup Language</i> “ – V současnosti nejrozšířenější jazyk pro tvorbu webových stránek
HTML5	Nástupce HTML/X-HTML
HTTP	„ <i>HyperText Transfer Protocol</i> “ – protokol používaný pro webovou komunikaci mezi prohlížečem a serverem
HTTPS	Je HTTP protokol doplněný o SSL vrstvu, která se stará o šifrování
Imagick	Knihovny pro práci s obrázky
InnoDB	Pokročilý úložný engine systému MySQL
Java	Programovací jazyk původní administrace
JS	„ <i>JavaScript</i> “ – skriptovací jazyk prohlížečů
Layout	Rozvržení stránky
MS Word	Textový procesor
MyISAM	Úložný engine systému MySQL
MySQL	Databázový systém
Netdogs	CMS systém firmy Beneta spol. s r.o.
O2 SMS Connector	GSM gateway operátora O2 Telefonica
Open-source	Označení pro vývoj aplikace s veřejně dostupným zdrojovým kódem

PayMUZO	Platební systém využívající karet MasterCard a VISA
PDF	„ <i>Portable Document Format</i> “ – rozšířený formát pro výměnu dokumentů v digitální podobě
PHP	„ <i>Hypertext PreProcessor</i> “ - Programovací jazyk serverové části
PHP Smarty	Šablonovací systém
PSPad	Textový editor
PyQt	Vrstva knihoven pro podporu Qt frameworku v jazyce Python
Python	Programovací jazyk
Qt	Framework pro tvorbu GUI
RSS	„ <i>Really Simple Syndication</i> “ – jazyk postavený na XML sloužící k distribuci novinek
SOAP	Univerzální jazyk pro práci s Web Services
SQL	Dotazovací jazyk relačních databází
SQL Injection	Druh útoku při kterém dochází k nalezení slabiny v ošetřování uživatelského vstupu nad SQL dotazy
SSL	„ <i>Secure Sockets Layer</i> “ – systém bezpečné komunikace založené na asynchronním šifrování
Subversion (SVN)	Verzovací systém
Texy	Odlehčený formátovací jazyk s důrazem na jednoduchost
TinyMCE	Známý webový WYSIWYG editor
Total Commander	Souborový manažer
TXT	Textový dokument
VBS	„ <i>VisualBasic Script</i> “ – skriptovací jazyk prohlížečů
Web 2.0	Souhrné označení pro moderní webové aplikace
Webmaster	Člověk starající se o tvorbu webových projektů
WYSIWYG	„ <i>What You See Is What You Get</i> “ – Způsob editace obsahu podobný např. MS Word, uživatel přímo vidí, jak bude výstup naformátován
X-HTML	„ <i>eXtensible HyperText Markup Language</i> “ – jazyk HTML, který respektuje standardy XML a je podle nich definován
XML	„ <i>eXtensible Markup Language</i> “ – univerzální značkovací jazyk
XML-RPC	Způsob vzdáleného volání procedur pomocí jazyka XML
XSL	„ <i>eXtensible Stylesheet Language</i> “ – jazyk stylopisů pro XML
XSLT	„ <i>eXtensible Stylesheet Language Transformations</i> “ – jazyk pro transformaci XML do jiného výstupního formátu (HTML, PDF, TXT, aj.)
XSS	„ <i>CrossSite Scripting</i> “ – druh útoku pomocí infiltrace webových stránek vlastním kódem

Analýza problému

Jako uživatel systému i jako kodér starající se o implementaci jsem se potýkal s několika problémy:

- **Nevyhovující editační prostředí pro koncového uživatele**

Drtivá většina současných řešení je pro uživatele bez pokročilé znalosti systému uzavřená. CMS nutí uživatele k znalosti HTML či k používání vlastního nestandardního jazyka (BB tagy, Taxy, aj.). Existují i systémy využívající WYSIWYG jako editoru obsahu, nicméně výstupní kód je nevalné kvality a často rozhodí celý webový layout, o validitě raději ani nemluvě. Všechny CMS hledají jakýsi kompromis mezi funkčností a čistotou. Existují nástroje, které vám dovolí upravovat pouze formát písma a existují nástroje, které vám poskytnou komfort srovnatelný s Wordem, bohužel za cenu katastrofálního výstupního kódu – příkladem budiž TinyMCE.

- **Webové administrační rozhraní**

Zde se dá polemizovat nad tím, zda je to klad nebo zápor. Takové webové administrační prostředí je pak přístupné odkudkoliv. Bohužel díky současnému stavu implementace webových technologií v prohlížečích je dosažení tohoto stavu spíše sen než realita. Webové rozhraní je potřeba zvláště optimalizovat pro několik různých prohlížečů a zdaleka ne u všech webových administrací je toto dotaženo do konce. Ať je však webové rozhraní jaké chce, ať využívá AJAX a pyšní se visačkou Web 2.0, nikdy nemůže dosáhnout komfortu desktopové aplikace.

- **Nevhodná implementace**

Dosud jsem se nesetkal s CMS systémem, který by byl efektivně navržen pro potřeby kodéra. Cena za implementaci je mnohdy až příliš vysoká a vyžaduje velké časové i finanční prostředky.

- **Závislost na výstupním formátu**

Řada systémů přímo generuje HTML kód, ten bývá často nevalidní a nese s sebou závislosti. Systém je pak velice těžce adaptovatelný v jiných podmínkách a často se setkáváme i se situacemi, kdy výstup nelze ovlivnit. Výsledný projekt pak např. nelze přeložit do jiných jazyků.

Jako konkurenční systém mohu uvést např. systém Netdogs, jehož provozovatelem je Beneta spol. s r.o. Tento systém jsem několik let implementoval a podílel se na jeho vývoji.

Specifikace požadavků

- **Vytvoření konkurence schopného řešení správy obsahu**

Důležitým aspektem bylo vytvořit produkt, který bude mít dostatečnou podporu a obstojí v náročném konkurenčním prostředí. Open-source řešení většinou neposkytují dostatečnou garanci – projekty se ruší a znovu zakládají, vývoj nereflexuje bezpečnostní chyby, není dostatečná dokumentace a často trpí nestabilním API.

- **Vytvoření snadno implementovatelného konceptu**

Bezesporu největším úkolem bylo navrhnout systém, který poskytne nástroj nejen uživateli, ale i administrátorovi. Řešení je tedy třeba optimalizovat pro minimalizaci prostředků při implementaci na jednotlivých projektech. V ideálním případě by doba potřebná k vytvoření nové webové prezentace neměla přesáhnout 6 hodin resp. 10 hodin u e-shopů, mluvíme-li o řešeních psaných na míru.

- **Vytvoření transparentního prostředí umožňující flexibilní růst aplikace**

Aby měl systém nějaký smysl, je potřeba ho navrhovat s ohledem na budoucnost. Systém by měl být absolutně nezávislý na výstupním formátu (HTML, X-HTML, XML, PDF, TXT, aj.) a měl by být poskytován transparentně přes libovolný protokol (HTTP, HTTPS, XML-RPC).

- **Možnost plné vícejazyčné implementace**

Systém by měl být snadno přeložitelný do cizích jazyků a neměl by nijak omezovat ve výběru jazyka (např. podpora čínských znaků). Neměly by existovat žádné „zakódované“ řetězce, které by nešly přeložit.

- **Zvolení kompromisu mezi funkcionalitou a efektivitou z pohledu uživatele**

Systém by měl být co nejpřívětivější pro uživatele a měl by být pevně stanoven poměr mezi užitečnými funkcemi a funkcemi, které jsou jen na obtíž. Typickou reprezentací takových funkcí je možnost uživatelsky ovlivnit řezy písma, barvu apod. Uživatelé takových metod pak užívají k svému rozmaru bez znalosti pravidel přístupnosti a typografie. Uživatelé by měli mít možnost využívat pouze kóděrem předdefinované volby.

- **Komfortní administrační rozhraní**

Využití potenciálu desktopových aplikací narozdíl od neinteraktivity webových rozhraní.

Bližší pohled na návrh

- **Z pozice kodéra**

Z předchozí zkušenosti vím, že největší zdržování při vývoji webového prezentace je v 90% absence vývojového prostředí, které by se bylo schopné přizpůsobit konkrétnímu redakčnímu systému. Rozhodl jsem se proto integrovat do administračního nástroje i funkce využitelné přímo webmasterem – přímá editace layoutu stránky, CSS, apletů apod. Díky tomu má kodér k dispozici několik automatizovaných nástrojů, které mu umožní uspořit hodiny trávené ve společnosti PSPadu a Total Commanderu.

- **Z pozice uživatele**

Pro administrační rozhraní je alfou omegou kvalitní WYSIWYG editor. Přeci jen je to komponenta, která bude uživateli využívána nejčastěji. WYSIWYG editor by měl v první řadě tvořit kvalitní a čistý kód, aby jeho výstup v žádném případě nemohl narušit layout výsledné stránky, jako se tomu často děje. Dále by uživatele neměl nijak omezovat, měl by nabízet pohodlné vkládání základních prvků jako jsou nadpisy, odstavce, seznamy, tabulky a obrázky. Obrázky by se měly dát snadno nahrát a spravovat, neboť to je často velké mínus webových administrací. Zároveň by ale WYSIWYG editor měl být přehledný a používat rozumnou množinu funkcí. Postupem času se prokázalo, že spousta funkcí je pro uživatele spíše nebezpečná. Typickým příkladem je již dříve zmíněná změna řezu a barvy písma. WYSIWYG editor by však neměl být jedinou komponentou, ale mělo by být umožněno editovat přímo výstupní kód v HTML nebo jiném jazyce.

- **Z pozice systému**

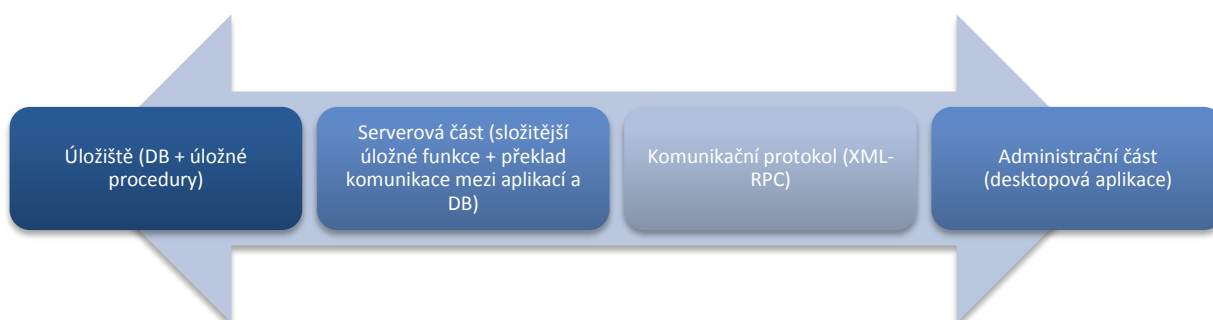
Systém musí být od počátku správně navržen, aby nedocházelo k nekonzistenci v API a problémům v databázové struktuře. Je velice důležité, aby staré projekty mohly být snadno updatovatelné novou verzí z důvodu bezpečnostních záplat. Zároveň systém musí být snadno rozšiřitelný a co nejvíce modulární. Dalším neméně důležitým aspektem je správné rozvrstvení prezentační a aplikační logiky.

Návrh

Díky předchozímu vývoji Javového ekvivalentu nebyl návrh příliš složitý a jelo vše v zajetých kolejkách. Bohužel z důvodu ztráty dat nebylo z této fáze zachováno mnoho. Doba návrhu byla téměř celý měsíc.

Jak již bylo nastíněno v předchozí části, celý systém se dá rozdělit do několika vrstev. Nejdůležitější vrstvy jsou serverová a administrační část. Více napoví následující diagram.

Diagram aplikačních vrstev



Návrh serverové části byl nejsložitější. Serverová část se musí starat o komunikaci s administračním nástrojem, patřičné ukládání souborů nebo předávání úložných příkazů databázi. Navíc se stará o verzování ukládaných změn a autorizaci všech prováděných akcí. U prezentační části je vytvořena speciální aplikační podvrstva, která má na starosti zobecnit API pro práci s DB, autorizování uživatelů, práci s obrázky a jiné důležité funkce každého CMS systému. U této vrstvy je kladen velký důraz na modularitu, a tak je systém snadno rozšířitelný např. o specifické e-shopové funkce (platba kartou, správa objednávek, aj.). Pod aplikační vrstvou již jsou samotná data, která jsou navíc u „výstupních“ jazyků parsována systémem PHP Smarty.

Diagram funkcí serverové části



PHP Smarty

Specialitou systému smarty je mnou dopsaná podpora pro tvorbu layoutů stránky. Systém umožňuje vykreslovat stránku v přirozené stromové struktuře. Udržuje se tak přehlednost a zpracovávají se pouze ty části, které jsou potřeba. Vykreslování se rozdělí na několik dílčích úloh a dojde k jednoduchému rozvrstvení. Nemusíme se tak starat např. o vykreslování hlavičky stránky, když vykreslujeme produkt. O to se postará stránka, která je mu nadřizená. To ve výsledku vede ke zpřehlednění kódu a zrychlení vývoje. Více znázorní následující diagram.

Diagram layoutovacího systému



Toto však není jediné řešení správy vykreslování, které systém nabízí. Kód je mimo jiné možné strukturalizovat do tzv. apletů. Aplety jsou části kódu, které se v našem projektu opakovaně využívají – např. webová galerie. Nechceme, aby kód webové galerie byl znovu psán pokaždé, když budeme chtít vypsát pole obrázků. Napíšeme proto aplet, který pokaždé zavoláme s patřičnými parametry. Je to jakási obdoba procedurálního programování. Ukázkou jednoduchého apletu můžete vidět na následující stránce.

V současné době systém obsahuje už několik desítek předpřipravených apletů, které zvládají pokrýt často se opakující úkony, jako je např: diskuzní fórum, kontaktní formulář, anketa, přihlašovací formulář a registrace nebo třeba již zmíněná fotogalerie. Implementace je pak často pouhým voláním již předpřipravených komponent, ale stále při zachování plné flexibility jelikož **všechny aplety jsou plně editovatelné skrze administrační rozhraní.**

Ukázka volání apletu

```
{* Vykreslí fotogalerii *}
{applet name="gallery"}
  {param name="thumbnailWidth"}300{/param}  {* Šířka náhledu: 300px *}
  {param name="columnsCount"}2{/param}  {* 2 obrázky na řádek *}
{/applet}
```

Ukázka obsahu apletu

```
{* Vykreslim tabulku s obrázkama *}
{foreach from=$galleryFiles item="file" name="gallery"}
  {if in_array($file.type, array("JPG", "PNG", "GIF"))}
    {* Zacetky radku (tabulky) *}
    {if $smarty.foreach.gallery.iteration % $columnsCount == 1}
      {if $smarty.foreach.gallery.first}<table class="galleryTable">{else}</tr>{/if}
      <tr>
      {/if}

      <td class="galleryItem">
        <div class="galleryThumbnail">
          {contentLink file=$file.name fromPage=$fromPage rel=$rel}
          {img file=$file.name fromPage=$fromPage width=$thumbnailWidth}
          {/contentLink}
        </div>
        <div class="galleryTitle">{$file.name}</div>
        <div class="galleryDescription">{$file.description}</div>
      </td>

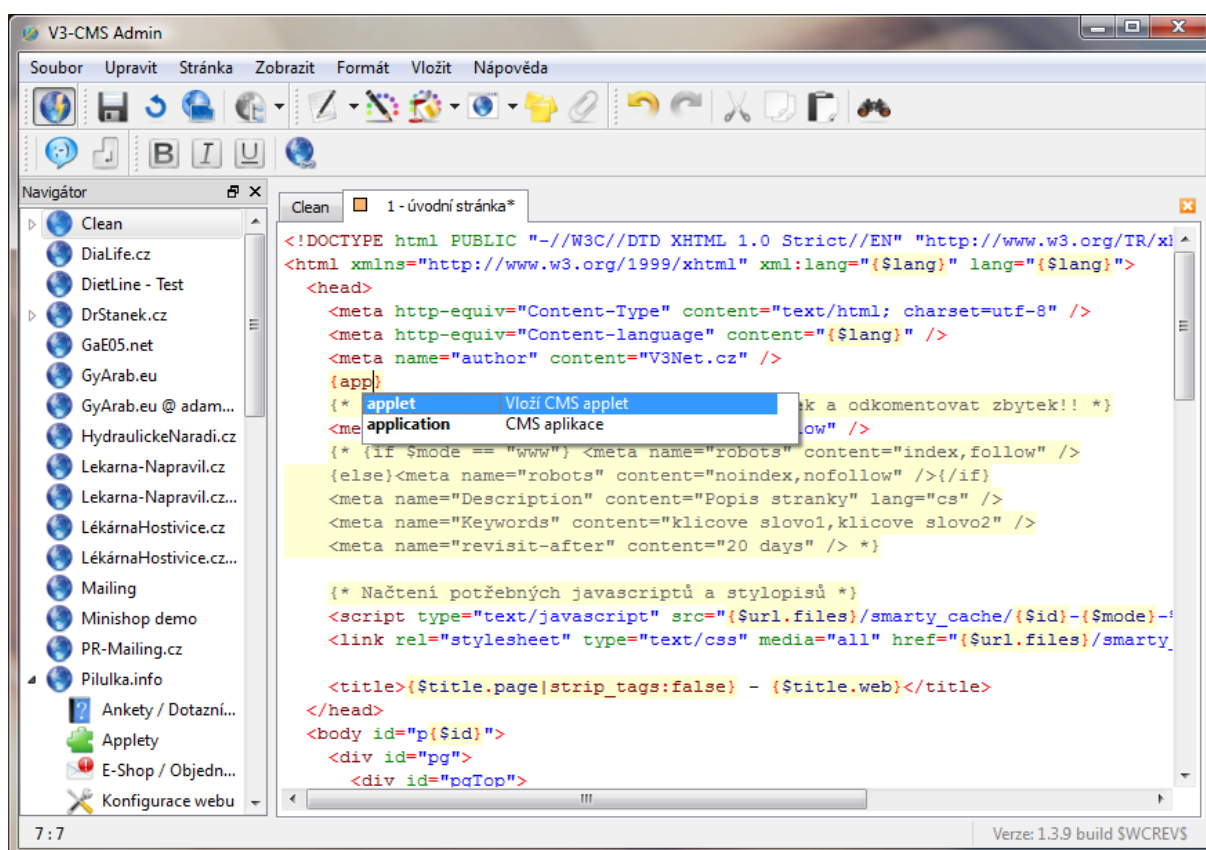
      {* Konec tabulky *}
      {if $smarty.foreach.gallery.last}</tr></table>{/if}
    {/if}
  {/foreach}
```

Administrace

Administrační nástroj je bezesporu největším kamenem úrazu všech CMS řešení. Systém může být navržen sebelépe, a i přesto může být nepoužitelný z důvodu špatné správy. Pro běžné uživatele jsou administrační nástroje často nepřívětivé a celá správa webu se tak stává značně neefektivní. Jak jsem naznačil v předchozích kapitolách, administrace systému V3-CMS probíhá zcela jiným způsobem, než jak jsme tomu byli doposud zvyklí. Místo běžné webové administrace nabízí správu v podobě desktopové aplikace, která nabízí o mnoho přítulnější rozhraní. Zároveň je administrace určena i pro samotný vývoj webového projektu a proto nás ušetří mnoha s tím spojených nepříjemností.

Administraci si budete moci vyzkoušet na k tomu speciálně vytvořeném projektu, viz. příloha č. 4. Pro úplnost zde uvedu několik obrázků pořízených z aplikace.

Ukázka editace šablony



Ukázka správy objednávek

Minishop Objednávky

Filtr

Zobrazit všechny objednávky
 Zobrazit pouze nové objednávky

Zobrazit objednávky od

ID	Uživatel	Jméno	Příjmení	Čas	Cena	Druh platby
200806 / 5		Jan	Noha	17.6.2008 19:12:37	220.00 Kč	Dobírkou
200806 / 4		Jan	Noha	17.6.2008 17:37:54	650.60 Kč	Dobírkou
200806 / 3		Jan	Noha	17.6.2008 17:37:10	1145.90 Kč	Dobírkou
200806 / 2		Adam	Staněk	16.6.2008 9:51:28	1511.80 Kč	Dobírkou
200806 / 1		Adam	Staněk	16.6.2008 9:46:38	650.60 Kč	Dobírkou

Ukázka správy uživatelů

Clean 1 - úvodní stránka* Uživatelé

Hledat

Klíčové slovo:

Username	Name	Surname	Groups
admin	Admin	Admin	Administrátor, Správce obchodu
admin2008	Admin	Admin	Uživatel webu
admin21	Admin	Admin	Uživatel webu
admin22	Admin	Admin	Uživatel webu
admin23	Admin	Admin	Uživatel webu
admin24	Admin	Admin	Uživatel webu
admin25	Admin	Admin	Uživatel webu
admin26	Admin	Admin	Uživatel webu
admin27	Admin	Admin	Uživatel webu
admin28	Admin	Admin	Uživatel webu
admin29	Admin	Admin	Uživatel webu
admin30	Admin	Admin	Uživatel webu

The screenshot displays the V3-CMS Admin interface. At the top, a menu bar includes options: Soubor, Upravit, Stránka, Zobrazit, Formát, Vložit, and Nápvěda. Below the menu is a toolbar with icons for text formatting (bold, italic, underline, strikethrough, text color, background color), alignment, indentation, bulleted and numbered lists, link management, and other editing tools. The main content area is titled 'Content' and contains several text blocks:

- Text:** A block with the text 'lorem ipsum (test)'.
- Text:** A block with the text '3 - lorem ipsum (test)'.
- Text:** A block with the text 'Thulek'.
- Text:** A block with the text 'lorem ipsum (test)'.
- Text:** A block with the text 'Content'.
- Text:** A block with the text 'lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam hendrerit, turpis a gravida? ultricies, quam nunc sodales nulla, semper interdum quam mauris quis urna. In tristique viverra nunc. Praesent id ligula. Proin et lectus. Duis vel nulla non metus molestie suscipit. Ut rhoncus eros faucibus neque. **Maccenas** mi *mauris*; sagittis vulputate, mattis vitae, fringilla vel, enim. ~~Aliquam erat volutpat. Nam vitae- ipsum at pede malesuada venenatis. Vivamus condimentum ullamcorper felis. Vestibulum trndunt aliquam nulla. Etiam npportitor. Cras ullamcorper arcu ac purus.~~
- Section Header:** A blue block with the text 'Podnadpis 1'.
- Text:** A block with the text 'Etiam a velit. Duis erat duj, rhoncus et, feugiat eget, adipiscing eu, nisi. Praesent a velit. Cras placerat tellus vitae felis. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Nulla facilisi. Pellentesque sit amet odio nec risus sagittis viverra. Vestibulum du. Nulla facilisi. Sed fermentum pharetra ligula. Cras est enim, venenatis nec, dictum in, egestas non, sapien.' (Note: 'sapien' is underlined in the image).
- List:** A block containing a bulleted list:
 - Lorem ipsum 1
 - Lorem ipsum 2
 - Lorem ipsum 3
- Text:** A block with the text 'Praesent lectus maona, ultricies vel, suscipit ut, semper euismod, ipsum, Pellentesque faucibus consequat risus. Sed diam sapien, tristique sed, dapibus nec, vehicula vitae, turpis. Ut venenatis nibh in'.
- Section Header:** A block with the text 'PereX'.
- Text:** A block with the text 'Etiam a velit. Duis erat duj, rhoncus et, feugiat eget, adipiscing eu, nisi. Praesent a velit. Cras placerat tellus vitae felis. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Nulla facilisi. Pellentesque sit amet odio nec risus sagittis viverra. Vestibulum du. Nulla facilisi. Sed fermentum pharetra ligula. Cras est enim, venenatis nec, dictum in, egestas non, sapien.'

At the bottom of the editor, the text 'Verze: 0.9.9 build SWCREVS' is visible.

Implementace

Tento projekt s sebou nesl celou řadu otázek, na které jsem v době návrhu musel myslet. V první řadě je to volba použitých technologií. Jelikož, jak už jsem dříve naznačil, jsem se rozhodl pro desktopovou variantu administračního nástroje, můžeme celý projekt rozdělit na administrační a serverovou část. Serverová část se stará o publikaci dat uživateli pro webové klienty a o úložnou logiku. Administrační část je pak dedikovaná desktopová aplikace, která komunikuje se serverovou částí. Stará se tedy pouze o samotné uživatelské prostředí, data jako taková jsou zpracovávána až na serveru. Viz. kapitola Návrh.

Pro vývoj administrační části jsem zvolil jazyk Java. Bohužel po havárii disku se mi nepodařilo zachránit původní data a projekt jsem psal znovu od nuly. Vzhledem k problémům, které mi vývoj pod Javou způsobil („líné“ GUI, nedostatek volně dostupných potřebných knihoven) jsem se ale rozhodl sáhnout po C++. C++ mi nabídlo solidní objektový návrh, multiplatformní zázemí a díky knihovně Qt i vynikající API pro vývoj GUI. Jedním z kandidátů byl i Python + PyQt, ale po zkušenosti s líným GUI Javy, jsem raději zvolil kompilovatelný jazyk. Jako grafický framework jsem zvolil Qt, neboť ve verzi 4 doznalo spousty skvělých novinek. Zároveň je velice srozumitelné a perfektně dokumentované, narozdíl např. od GTK.

Pro serverovou část jsem zvolil, mně nejbližší, jazyk PHP. Bohužel již nyní vidím, že tato cesta nebyla z nejlepších. Očekávám, že projekt bude postupem času přepsán do Javy nebo Pythonu. Vývoj PHP sice velice pokročil a ve verzi 5, ve které je projekt napsaný, už poskytuje celkem solidní práci s objekty, ale stále bohužel nemůže příliš konkurovat. Uvidíme, co přinese verze 6.

Pro úložnou část posloužil systém MySQL 5 s tabulkami MyISAM. Postupem času však přecházíme na InnoDB kvůli potřebě transakcí a úložných procedur.

Jako protokol pro komunikaci mezi administrační aplikací a serverovou částí byl zpočátku využíván SOAP, ale kvůli problematické implementaci (tehdy se ještě vyvíjela administrační část v Javě) byl vytvořen jednoduchý vlastní protokol, založený na XML-RPC.

Jelikož CMS doslova vyžaduje úplné oddělení prezentační vrstvy od aplikační logiky, je využit šablonovací systém PHP Smarty, který jsem však postupem času výrazně modifikoval a rozšířil.

Dále projekt využívá knihovny Imagick pro tvorbu náhledů obrázků a captcha kontroly.

Testování

Projekt je před každou stabilní verzí pečlivě testován k tomuto účelu speciálně napsanými nástroji. Testuje se zejména výkon při krizových operacích a atomičnost. Systém jako takový je imunní vůči běžným formám útoků jako slovníkové útoky nebo útoky hrubou silou na autorizační tabulku. Dále je pravidelně testován na náchylnost k XSS nebo SQL injection. Zvláštní metodiku testování pak vyžaduje nasazování nových verzí PHP a dalších použitých technologií v produkčním prostředí.

Pro usnadnění deploymentu a testování je používán verzovací systém Subversion (SVN), který umožňuje rozdělení kódu na stabilní a vývojovou větev.

Zhodnocení

Po roce a půl vývoje systému a jeho implementaci lze říct, že byl tento projekt doveden k zdárnému konci. Webové stránky postavené na V3-CMS mohou denně využívat tisíce lidí. Systém může být nasazován od malých firemních webů až po velké e-shopové portály. Systém je otevřený k budoucímu rozvoji a je snadno rozšiřitelný.

Dosažené výsledky:

- **Systém je přívětivý pro koncového uživatele**
Využívá síly desktopové aplikace, která je uživatelům mnohem blíže a poskytuje komfort práce s lokálními soubory, rychlé odezvy a kvalitní WYSIWYG editor.
- **Systém je lehce implementovatelný**
Zkušený kodér dokáže implementovat jednoduchý firemní web do 2 hodin, doba implementace e-shopu se pohybuje kolem 10-20 hodin. V porovnání s konkurencí je tento čas až 4x menší.
- **Systém je snadno rozšiřitelný a není závislý na výstupním obsahu**
Díky aplikačním modulům a PHP Smarty lze snadno přidávat nové technologie. Zároveň díky kvalitnímu návrhu je k dispozici stabilní API, zaručující zpětnou kompatibilitu. Systém je zároveň zcela nezávislý na výstupním formátu, tedy je schopný formovat výstup v HTML, X-HTML, v budoucím HTML5 nebo například i XML.
- **Systém je snadno rozšiřitelný**
Díky modulárnímu návrhu lze systém snadno rozšířit o novou funkcionalitu. V současné době je např. rozšířen o moduly ePlatby, PayMUZO a O2 SMS Connector.
- **Propracovaný systém vrstev a apletů**
Se systémem se distribuuje množství apletů coby hotových řešení pro modelové úlohy. Vytvoření např. fotogalerie je pak hračkou na několik řádků a to vše při **plné konfigurovatelnosti**.

Porovnání s konkurenčními systémy

V porovnání s ostatními systémy těží V3-CMS zejména v administraci a snadné implementaci. Desktopová aplikace vylepšuje odezvu mezi uživatelem a systémem a nabízí komfortní WYSIWYG editor a správu šablon. Zároveň slouží jako silná zbraň administrátora při implementaci, kde notně pomáhá zredukovat čas potřebný k zhotovení díla.

	V3-CMS	Běžný CMS
Šablonovací systém	✓	✓
Vícejazyčný	✓	🟢
Plně ovlivnitelný výstup	✓	🟢
Verzovací systém	✓	✗
Apletovací systém	✓	🟢
WYSIWYG editor	✓	✓
Správa příloh	✓	🟢
Správa šablon	✓	✗
Správa stylopisů	✓	✗
Správa skriptů	✓	✗
Doba implementace běžné stránky	4-6 hodin	10-20 hodin
Doba implementace e-shopu	10-15 hodin	30 a více hodin

Seznam použité literatury

Qt 4.4.3 Reference Documentation, 2008, Nokia Corporation, Dostupné z URL:

<http://doc.trolltech.com/4.4/>

MySQL 5.0 Reference Manual, 2008, Sun Microsystems, Dostupné z URL:

<http://dev.mysql.com/doc/refman/5.0/en/index.html>

PHP Manual, 2009-03-20, PHP Documentation Group, Dostupné z URL:

<http://cz2.php.net/manual/en/>

Extensible Markup Language (XML) 1.0 (Fifth Edition), W3C Recommendation 26 November 2008,

Dostupné z URL: <http://www.w3.org/TR/2008/REC-xml-20081126/>

XML-RPC Specification, 1999, Dave Winer, Dostupné z URL: <http://www.xmlrpc.com/spec>

XHTML™ 1.0 The Extensible HyperText Markup Language (Second Edition), W3C Recommendation

26 January 2000, revised 1 August 2002, Dostupné z URL: <http://www.w3.org/TR/xhtml1/>

RFC1738, Uniform Resource Locators (URL), 1994, CERN, Dostupné z URL:

<http://www.faqs.org/rfcs/rfc1738.html>

RFC 2818, HTTP Over TLS, 2000, The Internet Society, Dostupné z URL:

<http://tools.ietf.org/html/rfc2818>

Seznam příloh

1. Případová studie GyArab.eu
2. Uživatelský manuál – Správa obsahu
3. Uživatelský manuál – WYSIWYG editor
4. Přístupové údaje k testovacímu webu