

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Využití Microsoft Windows SharePoint Services jako DMS či CMS

Jan Keller

Bakalářská práce

2008

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Jan KELLER**

Studijní program: **B2646 Informační technologie**

Studijní obor: **Informační technologie**

Název tématu: **Využití Microsoft Windows SharePoint Services jako DMS
či CMS**

Z á s a d y p r o v y p r a c o v á n í :

Cílem práce je zhodnocení možností použití MS SharePoint Services jako DMS a CMS.
Teoretická část:

1. Pojem CMS a DMS - možnosti využití, požadavky na řešení, vlastnosti.
2. Microsoft Windows SharePoint Services - možnosti konfigurace, možnosti propojení s jinými systémy (portály, ...).

Praktická část:

1. Instalace a konfigurace MS SharePoint Services.
2. Vytvoření struktury pro řízení práce s určitými typy dokumentů např. směrnice apod.
3. Implementace propojení s jinými systémy pomocí standardizovaného rozhraní.

Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

Kutěj Tomáš, Technologie Microsoft Office SharePoint 2003. ISBN: 80-251-0976-3

Vedoucí bakalářské práce:

Ing. Lukáš Slánský

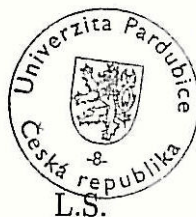
Ústav elektrotechniky a informatiky

Datum zadání bakalářské práce:

30. listopadu 2007

Termín odevzdání bakalářské práce:

16. května 2008

A handwritten signature in black ink, appearing to read "Simeon".

doc. Ing. Simeon Karamazov, Dr.

děkan

V Pardubicích dne 29. dubna 2008

PODĚKOVÁNÍ

Děkuji vedoucímu bakalářské práce Ing. Lukáši Slánskému za cenné rady a připomínky při vedení mé bakalářské práce.

SOUHRN

Práce je věnována problematice Content Management System a Document Management System se zaměřením na Microsoft Windows SharePoint Services 3.0. Popisuje možnosti vývoje nad technologií Microsoft SharePoint ve vývojových nástrojích Microsoft SharePoint Designer 2007 a Microsoft Visual Studio 2008. Zkoumá možnosti propojení Microsoft Windows SharePoint Services 3.0 s jinými portály přes standardní rozhraní.

KLÍČOVÁ SLOVA

SharePoint, DMS, CMS, portály, Web Services, XML-RPC, ASP.NET 2.0, C#

TITLE

Using Microsoft Windows SharePoint Services as a DMS or CMS

ABSTRACT

This work deals with problematic of Content Management System and Document Management System with focus on Microsoft Windows SharePoint Services 3.0. The work describes possibilities of developing under the Microsoft SharePoint technology in Microsoft SharePoint Designer 2007 and Microsoft Visual Studio 2008. It explores the possibilities of connection of Microsoft Windows SharePoint Services 3.0 with other portals through standard interface.

KEYWORDS

SharePoint, DMS, CMS, portals, Web Services, XML-RPC, ASP.NET 2.0, C#

Obsah

1	Úvod	10
1.1	Motivace pro výběr tématu.....	10
1.2	Obsah práce.....	10
2	Pojem CMS a DMS.....	12
2.1	Content Management System.....	12
2.1.1	Životní cyklus CMS.....	12
2.2	Web Content Management System.....	13
2.3	Document Management System.....	14
2.4	CMS a DMS – nástroje k realizaci	15
2.4.1	Open source řešení	15
2.4.2	Komerční řešení	16
3	Technologie Microsoft SharePoint	18
3.1	Microsoft Windows SharePoint Services	19
3.1.1	Základní členění WSS 3.0	19
3.2	Microsoft Office SharePoint Server.....	21
3.2.1	Standart edice	21
3.2.2	Enterprise edice	22
4	Vývoj nad SharePoint	23
4.1	Vývojové nástroje.....	23
4.1.1	Microsoft SharePoint Designer 2007	23
4.1.2	Microsoft Visual Studio	24
4.2	Technologie	24
4.2.1	ASP.NET	24
4.2.2	C#.....	25

4.2.3	XML-RPC	25
4.2.4	SOAP	26
4.2.5	Web Services	26
4.3	Objektový model	26
4.3.1	Vstupní bod	26
4.3.2	Serverová architektura.....	27
4.3.3	Webová architektura	29
5	Návrh řešení.....	31
5.1	Intranetový portál	32
5.1.1	Umístění portálu	32
5.1.2	Uživatelské role	32
5.2	Jednotný vizuální styl.....	33
5.3	Knihovna dokumentů	33
5.4	Propojení s jinými portály.....	34
6	Tvorba portálu	35
6.1	Operační prostředí	35
6.1.1	Počítačová struktura	35
6.2	Instalace.....	36
6.3	Konfigurace.....	37
6.3.1	Vytvoření Webové aplikace	37
6.3.2	Vytvoření site collection.....	38
6.3.3	Vytvoření základní struktury stránek.....	38
6.3.4	Nastavení práv	39
7	Implementace propojení s jinými systémy	41
7.1	Programová hierarchie	41
7.1.1	BakalarkaWebConrol.dll.....	41

7.1.2	BakalarkaWebService.dll.....	42
7.2	Klientská část	42
7.2.1	Aplikační vrstva	43
7.2.2	Tvorba Web Partů	44
7.3	Serverová část	45
7.3.1	Web Services	45
7.3.2	Metody třídy BakalarkaSample.....	46
7.3.3	Zprovoznění Web Services.....	46
7.4	Externí portál	46
7.4.1	Popis řešení	47
7.4.2	Poskytované služby	48
8	Závěr.....	49
	Použité zdroje.....	50
	Přílohy.....	52
	Příloha A – grafický návrh portálu.....	52
	Příloha B – grafický návrh testovacího webu.....	53

Seznam použitých symbolů a zkratk

<i>Zkratka</i>	<i>Anglický význam</i>	<i>Český význam</i>
AJAX	Asynchronous JavaScript and XML	<i>Technologie pro vývoj interaktiv. web. aplikací</i>
ASP	Active Server Pages	Technologie pro tvorbu webových stránek
CLR	Common Language Runtime	Prostředí pro běh .NET aplikací
CMS	Content Management System	Systém pro správu obsahu
DMS	Document Management System	Systém pro správu dokumentů
HTML	HyperText Markup Language	Jazyk pro tvorbu webových stránek
HTTP	Hypertext Transfer Protocol	Internetový protokol
HTTPS	Hypertext Transfer Protocol over Secure Socket	Zabezpečený HTTP
IIS	Internet Information Services	Webový server pro systém Windows
MOSS	Microsoft Office SharePoint Server	Microsoft Office SharePoint Server
RPC	Remote Procedure Call	Vzdálené volání procedur
SOAP	Simple Object Access Protocol	Protokol pro snadný přístup k objektům
SQL	Structured Query Language	Strukturovaný dotazovací jazyk
WCMS	Web Content Management System	Systém pro správu webového obsahu
WSDL	Web Services Description Language	Jazyk pro popis webových služeb
WSS	Windows SharePoint Services	Windows SharePoint Services
XML	eXtensible Markup Language	Rozšiřitelný značkovací jazyk

1 Úvod

1.1 Motivace pro výběr tématu

Toto téma jsem si vybral, protože se rád učím nové technologie. S Microsoft Windows SharePoint Services mám možnost naučit se nové programovací jazyky, seznámit se s novými webovými technologiemi a možnostmi vývoje aplikací pro platformu .NET a rozšířit si znalosti v oblasti správy obsahu a dokumentů.

Motivací také byla možnost budoucího využití získaných zkušeností. Microsoft SharePoint Services je dnes hojně využívaným nástrojem v mnoha velkých firmách a poptávka po vývojářích, kteří tuto technologii ovládají, stále stoupá.

1.2 Obsah práce

Náplní této bakalářské práce bude v první části vysvětlení pojmů Content Management System a Document Management System. Podrobnější popis těchto dvou systémů a jejich srovnání. Dále bude pozornost zaměřena na několik konkrétních nástrojů k realizaci. Zastoupeny budou jak systémy open source, tak komerční.

Následující část popisuje již samotnou technologii Microsoft SharePoint. Obecný popis technologie, důvody vzniku, možnosti využití a rychlá rekapitulace starších verzí. V následujících podkapitolách bude popis dvou konkrétních produktů z rodiny SharePoint, jejich vlastnosti a odlišnosti.

Před samotným návrhem aplikace je třeba vědět něco o vývoji nad technologií SharePoint. Budou popsány vývojové nástroje jako Microsoft Visual Studio a také technologie, které se budou v našem řešení objevovat. Zabývat se budeme těmi nejdůležitějšími, jako jsou ASP.NET 2.0, programovacím jazykem C#, a také Web Services či XML-RPC. Samostatná podkapitola bude věnována objektovému modelu SharePoint, bez jehož znalosti nelze naše řešení realizovat.

V kapitole Návrh řešení si popíšeme základní podobu naší práce. Řekneme si, kde bude náš portál umístěn, jaké prvky do něj chceme zabudovat a jaké budou uživa-

telské role v systému. Podíváme se také na grafický návrh portálu a jeho sladění s jednotným vizuálním stylem univerzity. Na závěr kapitoly navrhne, jakým způsobem bude řešeno propojení portálu WSS¹ s jinými systémy. Zde se podíváme jak na Web Services, tak na XML-RPC.

Další kapitola se zabývá již samotnou tvorbou portálu. V popisu operačního prostředí si ukážeme, s jakými počítači budeme v našem řešení pracovat a k čemu je budeme využívat. Poté krok za krokem projdeme instalaci. V podkapitole Konfigurace vytvoříme webovou aplikaci, kolekci webů a popíšeme konfiguraci nejdůležitějších prvků.

Předposlední kapitola pojednává o programové implementaci propojení WSS s jinými systémy. Popisuje se zde programová hierarchie, knihovny a třídy se objevují v našem řešení. Pozornost je pak věnována tvorbě klientské části a s tím spojená tvorba Web Partů a práce s objektovým modelem. V serverové části je pozornost zaměřena hlavně na Web Services. Nechybí ani popis Externího portálu, který slouží k testování námi implementovaných funkcí.

V závěrečné kapitole najdeme zhodnocení celé naší práce.

¹ WSS – Microsoft Windows SharePoint Services

2 Pojem CMS a DMS

2.1 Content Management System

Pod pojmem Content Management System rozumíme soubor softwarových nástrojů navržených pro snadné vytváření, spravování a prezentování informací. Těmito informacemi mohou být elektronické dokumenty, obrazové soubory, zvukové soubory nebo zdrojové kódy programů. CMS² vznikl kvůli stále se zvětšujícím nárokům na shromažďování informací. Myšlenkou bylo vytvořit systém, ve kterém by se informace snadno uchovávaly, upravovaly a vyhledávaly.

Pro firmy a nejen pro ně má nasazení CMS spoustu výhod (12):

- autentizace uživatelů,
- zvýšení bezpečnosti,
- snadná orientace v systému,
- redukce duplicitních informací,
- dostatečné kapacity pro budoucí růst,
- jednoduchá správa všech informací prostřednictvím jediného systému.

2.1.1 Životní cyklus CMS

V této části si popíšeme, jaké fáze má každý CMS.

Vytváření obsahu

Na prvním místě je poskytnutí jednoduše použitelného autorského prostředí obsahující WYSIWYG editor. Neboli poskytnout komukoliv, bez nutnosti znalosti jakýchkoliv programovacích technik, možnost vytvářet či upravovat obsah. CMS také umožňuje spravovat strukturu stránek, tzn. určovat, kde jaká stránka bude umístěna a jak budou navzájem propojené. Samozřejmostí je možno přihlášení se do systému ze vzdáleného místa a spravovat obsah pod svým účtem (12).

² CMS – Content Management System

Správa obsahu

Jakmile je stránka vytvořena, je uložena do centrální databáze. Zde jsou poté uchovávány všechny obsahy stránek a další informace k nim (12).

Díky této databázi je poté možné v CMS využít prvky, jako jsou:

- verzování,
- přidělování práv,
- integrace s existujícími informačními zdroji a systémy,
- workflow.

Publikování obsahu

Jakmile je obsah uložen, může být publikován na Internetu či intranetu.

Většina CMS používá k úpravě vzhledu šablony. Díky tomu lze použít stejný obsah na různých webech, vždy s odpovídajícím vzhledem. O tvorbu šablon se starají designéři a webový vývojáři. Autor obsahu se tak může plně soustředit na tvorbu obsahu.

Content management system také nabízí množství prvků ke zlepšení kvality a efektivity stránek samotných. CMS může například sám vytvořit navigační menu podle struktury obsahu. Celé stránky tak mohou být dynamické a interaktivní (12).

2.2 Web Content Management System

Pokud se bavíme o CMS a správě informací, jedná se ve většině případů o informace prezentované prostřednictvím webových stránek a o správu HTML³ obsahu. Mluvíme tedy konkrétně o správě obsahu webových stránek. Takový systém se anglicky nazývá Web Content Management System. Používá se ke správě a řízení velkých, dynamických souborů webových materiálů (15).

³ HTML – HyperText Markup Language je značkovací jazyk pro vytváření stránek v systému World Wide Web, který umožňuje publikaci dokumentů na Internetu.

Většina WCMS⁴ obsahuje tyto klíčové prvky:

- snadná správa obsahu přes webové rozhraní,
- šablony – při tvorbě nového obsahu se automaticky použijí,
- možnosti rozšíření – zásuvné moduly,
- možnost upgradu přes webové rozhraní,
- workflow.

2.3 Document Management System

Document Management System je systém softwarových nástrojů sloužících ke sledování a ukládání elektronických dokumentů. DMS⁵ je starší než CMS a byl vytvořen kvůli vzrůstající potřebě spravovat velký počet dokumentů v rámci jedné organizace. DMS řídí životní cyklus dokumentů ve firmě – to jak jsou dokumenty vytvářeny, publikovány a zpracovávány (3).

Klíčové vlastnosti dobře navrženého DMS (10):

- správa typů obsahu (šablony, metadata, atd.),
- možnost nastavení práv dokumentům v každém stadiu jejich životního cyklu,
- zálohování,
- archivace neaktuálních dokumentů,
- vyhledávání.

CMS a DMS jsou rozdílné systémy, navrženy k různým účelům. Rozdíl je hlavně v tom, že se samotným DMS nelze vytvořit webové stránky. Je zaměřený pouze na uchování, správu a archivování dokumentů, které většinou nebývají navzájem propojené. DMS většinou poskytuje kvalitní workflow pro řízení životního cyklu dokumentů. V praxi se dost často tyto dva systémy prolínají a vznikají tak produkty, které nabízejí jak kvalitní správu dokumentů, tak kvalitní správu webového obsahu.

⁴ WCMS – Web Content Management System

⁵ DMS – Document Management System

2.4 CMS a DMS – nástroje k realizaci

Pro realizace lze volit z bezpočtu dostupných řešení založených na různých technologiích. Všechny však mají společné to, že data ukládají do jedné či více databází a uživatel k nim přistupují za použití klientské aplikace. Ta může být v podobě tenkého klienta – webové aplikace, či tlustého klienta – desktopová aplikace. Vybírat lze mezi produkty komerčními a open source.

2.4.1 Open source řešení

Hlavním důvodem uživatelů pro nasazení open source řešení bývá často cena. Postavit celý systém na open source řešení může být ve výsledku značně levnější nežli komerční produkty. Ne všechna open source řešení jsou však zadarmo a jejich cena při zakázkové výrobě může být také velmi vysoká. Výhodou otevřených systému je bezpochyby také jejich možnosti vlastní modifikace přesně dle potřeb organizace. V současné době je k dispozici velké množství jak CMS, tak DMS a není problém najít to nejlépe padnoucí řešení pro dané účely (4).

Z hlediska bezpečnosti je open source dvojsečná zbraň. Chyby v programech může hledat mnohem širší skupina lidí a je proto naděje, že se snáze opraví. Na druhou stranu zranitelnosti mohou snáze najít i útočníci. V současné době se ovšem považuje za obecně výhodnější, když jsou informace dostupné všem, i za tu cenu, že jsou dostupné útočníkům. Alespoň u populárních programů s velkou základnou uživatelů a vývojářů lze předpokládat, že „uživatelská“ strana má výrazně větší prostředky než cracker (11).

Joomla!

Joomla! je jeden z nejpopulárnější open source CMS. Je používán po celém světě pro tvorbu jednoduchých stránek ale také k budování rozsáhlých firemních aplikací. Další užitečné prvky lze přidat pomoci stažitelných doplňků. Pokud by nestačila pouhá funkce CMS, jsou k dispozici další nástroje, se kterými lze snadno vytvářet fóra, foto galerie a další.

<http://www.joomla.org>

Drupal

Tento CMS má obrovskou uživatelskou základnu. Je podporován v mnoha jazycích. V Česku se taktéž těší velké popularitě. Drupal je určen pro budování dynamických webových stránek s širokou nabídkou služeb zahrnujících administraci, workflow, možnosti použití diskuzí, novinek, dále také funkce metadat a XML publishing pro účely sdílení obsahu.

<http://www.drupal.org>, <http://www.drupal.cz>

OpenKM

Document management systémy nejsou v open source sféře tak čteně zastoupeny jako CMS. Je to dáno tím, že správa velkého množství dokumentů je spíše výsadou velkých firem, které jsou také ochotny za kvalitní řešení zaplatit. I tak se dá však najít několik kvalitních open source systémů.

OpenKM je open source DMS, který může být díky svým charakteristikám použit ve velkých i malých firmách, jako užitečný nástroj informačního managementu, poskytující flexibilní a cenovou alternativu k jiným aplikacím. Dokumenty jsou přístupné přes webové rozhraní, které dovoluje jednoduše sdílet a vyhledat dokumenty, nastavovat potřebná práva a role a mnoho dalšího.

<http://www.openkm.com>

2.4.2 Komerční řešení

Komerční řešení jsou často nabízena jako hotová řešení přesně padnoucí potřebám firem a jsou tudíž mnohem rychlejší na implementaci nežli open source. Oproti otevřeným systémům nabízí často lepší dokumentaci a různé školicí kurzy. U velkých systému se také můžeme často setkat s integrací jiných komerčních řešení, které firmy běžně používají, jako jsou e-mailové služby, spolupráce s kancelářskými programy a další.

Nevýhodou je u komerčních řešení jasně cena. Komerční řešení, jak už vyplývá z názvu, jsou vždy placená. Komerční licence mohou stát velice vysoké částky (někdy

v řádech milionů korun) a s případnými úpravami a integrací do podnikového systému mohou ještě narůstat.

F-ART:CMS

F-ART:CMS je jeden ze zástupců českých CMS. Jako jeden z mála je dostupný také v open source podobě – BLOG:CMS. Toto řešení je však dostupné bez jakékoliv podpory. Struktura webu je definována pomocí šablon, a může být libovolně měněna a upravována. Celý web se skládá ze sekcí, k nimž můžete nastavovat práva neomezenému počtu redaktorů.

<http://www.hulan.cz>

Colibri CMS

Colibri umožňuje plnohodnotnou práci se strukturou webu, se šablonami stránek a jejich obsahem. Podporuje systém uživatelských práv a umožňuje verzování dokumentů. Přímo z prohlížeče lze nahrávat soubory ke stažení či upravovat obrázky.

<http://www.colibri-cms.cz>

Microsoft Office SharePoint Server

I společnost Microsoft nabízí svoje řešení pro správu obsahu a dokumentů. MOSS⁶ nabízí nejen výbornou správu dokumentů ale také široké možnosti v oblasti CMS a týmové spolupráce. S MOSS lze vytvářet portály, knihovny dokumentů, wiki weby a další.

⁶ MOSS – Microsoft Office SharePoint Server

3 Technologie Microsoft SharePoint

Základem každé firmy je v dnešní době tým pracovníků, ne jednotlivců. Aby firma byla produktivní a úspěšná, musí tyto týmy dobře pracovat, komunikovat a sdílet informace. Problémem u většiny firem je neexistence centrálního místa, kde by byli všechny informace dostupné. Ve skutečnosti situace mnohdy vypadá tak, že dokumenty jsou uloženy na souborovém serveru, informace o výrobcích v ekonomickém informačním systému. Pro členy týmu je pak složité vyhledat např. informace k určitému produktu, protože musí hledat v několika odlišných systémech, z nichž některé nemusí vyhledávání vůbec podporovat (9).

Jako řešení těchto problémů nabízí společnost Microsoft technologii SharePoint. Tato technologie slouží k vzájemné spolupráci mezi uživateli a sdílení dokumentů. Přístup k těmto sdíleným informacím je realizován hlavně skrze webové rozhraní ale také z aplikací Microsoft Office, a dalších. SharePoint umožňuje vytvářet rozsáhlé firemní portály – víceúčelová úložiště, které dovolují velice výhodně propojit všechny potřebné informace a jednoduše je zpřístupnit těm, kteří je potřebují.

V posledních letech se pro společnost Microsoft stala technologie SharePoint jedním z nejdůležitějších produktů. Svoji popularitu si získala již v roce 2003, kdy na trh spolu s kancelářskou sadou Office 2003 přišla druhá verze SharePoint (SharePoint Services 2.0 a SharePoint Portal Server 2003). Díky propojení právě se sadou Office, se stala tato technologie hojně využívanou v mnoha firmách. Hlavní výhody plynou již ze samotného konceptu, tedy CMS – všechny informace snadno dostupné na jednom místě. Tyto starší verze SharePoint byli ovšem stále zaměřeny hlavně na sdílení dokumentů (9).

Nová, v pořadí již třetí, verze SharePoint přináší spoustu nových a vylepšených prvků usnadňující práci. Z těch základních jmenujme možnost vytvoření kompletní firemní webové prezentace, synchronizace kalendáře, přidávání událostí, široké možnosti sdílení dokumentů (rezervace, verzování a další), vytváření blogů a wiki webů. Samozřejmostí je správa uživatelských účtů a s tím spojené přidělování pravomocí jednotlivým uživatelům a skupinám.

3.1 Microsoft Windows SharePoint Services

Základním produktem rodiny SharePoint je Windows SharePoint Services 3.0. Je to zdarma stažitelný doplněk pro Windows Server 2003 a 2008. Celé WSS 3.0 jsou postavené na ASP.NET 2.0 a využívají plně všech jeho prvků, jako jsou master pages nebo Web Parts.

Jako úložiště dat využívá WSS SQL Server 2005 (lze použít jakoukoliv edici). Při instalaci se vytváří konfigurační databáze, která je společná pro celou farmu, kde jsou uložena veškerá její nastavení. Pro každou webovou aplikaci se pak vytváří samostatná databáze obsahu. Další databáze je pak vytvořena pro službu vyhledávání. Zde se ukládají indexy obsahu.

3.1.1 Základní členění WSS 3.0

Farma

Je nejvyšší úroveň pro nasazení WSS. Je to logické uspořádání všech komponent, které obsahuje instalace WSS. Patří sem databázové servery, webové servery, vyhledávací servery ale také e-mailové servery a další podpůrné služby (14).

Na každém webovém serveru ve farmě běží webová aplikace.

Webová aplikace

Jedná se o IIS⁷ webovou aplikaci rozšířenou o podporu WSS. Webová aplikace je součástí farmy a obsahuje samotné WSS.

Webová aplikace má dvě úložiště dat. Jedno přímo v souborovém systému a jedno v databázi. Data v souborovém systému jsou statická a jsou shodná na všech webových serverech ve farmě. Jsou to funkční součásti webové aplikace – skripty, knihovny, Web Services a další. V databázi je ukládán obsah. Všechny webové

⁷ IIS – Internet Information Services. Webový server pro operační systém Windows Server a Windows Vista

stránky a jejich rozložení, upravené master pages, soubory z knihoven dokumentu. Každá webová aplikace má svoji vlastní databázi obsahu (14).

Site collection

Jedná se o logické seskupení více sites do jednoho objektu. Každá webová aplikace obsahuje nejméně jednu site collection, která tvoří základní strukturu portálu (14).

Site

V terminologii SharePoint představuje site funkční jednotku, která zastřešuje webové stránky a jejich nastavení. Každý site je dostupný z webové adresy, na které se fyzicky nachází. Samotné stránky jsou však přístupné přes web collection.

Web collection

Každý site vždy obsahuje web collection, což je pouze logické seskupení jednotlivých webových stránek. Je to objekt, pomocí něhož lze přistupovat ke stránkám (anglicky webs). Web je nejnižším článkem ve struktuře SharePoint.

Seznam

Je jeden ze základních prvků, který se může nacházet na stránce WSS. Jedná se o obsah ukládaný po položkách. Každý seznam podporuje workflow, verzování a nastavování práv. Jeho grafický výstup je reprezentován tabulkou.

Knihovna dokumentu

Jedná se o speciální typ seznamu, který slouží ke sdílení dokumentů. Všechny soubory v knihovně dokumentů jsou ukládány do databáze obsahu. Fyzicky tedy nejsou uloženy v souborovém systému. Díky tomu lze pro jednotlivé soubory nastavovat práva, rezervace a metadata.

Web Parts

Web Part je základním stavebním prvkem portálu. V terminologii webových portálů by se dal přirovnat k portletu. Každá stránka portálu je složena z Web Partů, které slouží k zobrazení dat a práci s nimi. Knihovny dokumentů a seznamy jsou také Web Party.

3.2 Microsoft Office SharePoint Server

Druhým produktem rodiny SharePoint je Office SharePoint Server, který je postaven na Windows SharePoint Services a obsahuje tedy všechny jeho funkcionality. Tyto možnosti jsou pak dále rozšiřovány dalšími nástroji pro lepší správu obsahu, dokumentů, integrace externích systémů a další.

MOSS existuje ve dvou základních edicích: Standart a Enterprise.

3.2.1 Standart edice

Obsahuje vylepšené nástroje pro práci s obsahem a správu dokumentů. Také rozšiřuje možnosti vyhledávání a obsahuje vylepšené uživatelské profily.

Sklad záznamů

Slouží k archivaci dat z různých zdrojů, kterými mohou být knihovny dokumentů, šablony stránek, e-maily, kontakty a další. Součástí skladu záznamů jsou i konvertory dokumentů. Základní sada konvertorů umožňuje převod formuláře InfoPath na obrázek (PNG, TIFF, EMF) nebo webovou stránku, XML na webovou stránku a také dokumenty Word (DOCM, DOCM) na webové stránky (6).

Vyhledávání

Oproti vyhledávání ve WSS, kde bylo možné vyhledávat obsah pouze na jednom serveru a pouze v rámci jedné site collection, nabízí MOSS mnohem více možností. Vyhledávání včetně indexace probíhá ve všech site collections na všech serverech ve farmě. Lze tak získat data z celé podnikové sítě v rámci jednoho dotazu. V Enterprise

edici navíc nabízí možnost vyhledávání i v externích zdrojích a Business Data Catalog (6).

3.2.2 Enterprise edice

Kromě všech možností obsažených ve Standart edici, obsahuje Enterprise edice navíc hlavně nástroje pro spolupráci s externími systémy.

Business Data Catalog

Slouží k integraci externích aplikací jako účetnictví, finanční software nebo správa objednávek do prostředí MOSS. Data z těchto aplikací je možné publikovat na stránkách portálu a indexovat je pro vyhledávání. Data jsou z externích aplikací získávána buď přes Web Services (podporovány jsou aplikace BizTalk, SAP, Siebel a další) nebo přímo z jejich databáze (SQL Server, Oracle). Pro zobrazení dat jsou k dispozici Business Data Webparts (6).

Excel 2007 Services

Je tenký klient pro sdílení tabulek programu Excel 2007. Umožňuje nahrávat Excel soubory na MOSS a nastavovat práva pro jednotlivé části tabulek. Uživatelům jsou pak přístupné ty části Excel tabulek, na které mají oprávnění. Obsah těchto sdílených tabulek lze otevřít v aplikaci Excel 2007 nebo přímo na webové stránce, uložit jako nový soubor Excel nebo pomocí Web Services zpracovávat v externí aplikaci (6).

InfoPath Forms Services

Slouží jako tenký klient pro práci s formuláři vytvořenými v aplikaci InfoPath⁸. Umožňuje tyto formuláře otevírat na stránkách MOSS, upravovat je a exportovat. Pomocí Web Services je možno s těmito formuláři pracovat také v externí aplikaci (6).

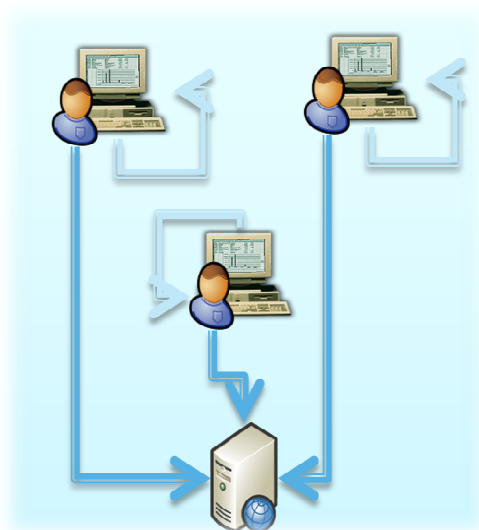
⁸ Microsoft Office InfoPath je aplikace pro tvorbu formulářů v jazyce XML.

4 Vývoj nad SharePoint

Každá organizace, která bude chtít naplno využívat potenciál WSS, zřejmě nevystačí pouze se základními funkcemi. Často je potřeba doprogramovat vlastní Web Parts, Web Services, Workflow a další specifické funkce. Objektový model SharePoint je plně zdokumentovaný a za použití vhodných vývojových nástrojů není problém vytvořit jakoukoli z těchto funkcí.

Vývojář, který chce vytvářet aplikace pro WSS musí mít administrátorské oprávnění na Windows Server. Musí znát souborovou architekturu WOSS a jeho objektový model. Musí ovládat jeden z jazyků platformy .NET, tj. C# nebo Visual Basic. Dále je nutná znalost infrastruktury .NET Framework 3.0/3.5, Microsoft Workflow Foundation, případně i technologií AJAX a Silverlight, které lze do stránek WOSS také implementovat. V neposlední řadě je nutná znalost ASP.NET 2.0 na kterém jsou celé WSS postaveny.

Při vývoji se pracuje v lokálním vývojovém prostředí (*Obrázek 1 – Lokální vývojové prostředí*). Kvůli bezpečnosti není doporučován klasický přístup klient-server. Doporučována jsou dvě schémata: server v roli workstation nebo práce ve virtuálním počítači. Tento přístup přináší výhody jako možnost lokálního testování bez přerušení provozu hlavního serveru a větší produktivita práce, kdy může více vývojářů pracovat paralelně (8).



Obrázek 1 – Lokální vývojové prostředí (8)

4.1 Vývojové nástroje

4.1.1 Microsoft SharePoint Designer 2007

SharePoint Designer je WYSIWYG HTML a ASPX editor a v současné době hlavním programem pro tvorbu webových stránek od Microsoftu. SharePoint Designer je

nástupce programu FrontPage. Ke změně názvu došlo kvůli zaměření na SharePoint (8).

SharePoint Designer je doporučován jako primární nástroj pro úpravu stránek WSS 3.0. Dovoluje pohodlně vytvářet jak grafický design, tak měnit rozložení komponent na stránce. Obsahuje většinu ASP.NET 2.0 funkcí a je možné v něm vytvářet jednoduché aplikace pro WSS.

4.1.2 Microsoft Visual Studio

Pokud chcete vytvářet rozšíření nebo pokročilejší řešení je nutné použít Microsoft Visual Studio. K dispozici je starší Visual Studio 2005 Professional pro které je dostupné rozšíření Windows SharePoint Services Extensions, které obsahuje speciální prvky pro pohodlný vývoj aplikací pro WSS i WOSS.

Je možné také použít novější Visual Studio 2008 Professional, které ovšem zatím neobsahuje WSS Extensions. Nabízí však již v základu SharePoint Workflow Template, které dovoluje velmi efektivně vytvářet workflow pro WSS 3.0 včetně kódu na pozadí v jazyce C# nebo Visual Basic. Visual Studio 2008 Professional také poskytuje možnost ladění klientských skriptů (8).

4.2 Technologie

Microsoft Windows SharePoint Services využívá mnoha moderních technologií, které programátorům usnadňují vývoj a rozšiřují možnosti využití. Některé tyto technologie, na kterých je WSS nebo které jsou použity v této práci, si podrobněji popíšeme.

4.2.1 ASP.NET

ASP.NET je webový framework vyvinutý společností Microsoft. Jeho pomocí lze vytvářet dynamické webové stránky, webové aplikace a webové služby. První verze byla vydána v roce 2002 spolu s vydáním .NET Framework 1.0 jako nástupce starší

technologie Active Server Pages (ASP). Oproti ASP je však tato technologie velmi odlišná (1).

ASP.NET je založen na Common Language Runtime, který je sdílen všemi aplikacemi postavenými na .NET Frameworku. Lze tak realizovat projekty v jakémkoli jazyce podporujícím CLR⁹. Mezi tyto jazyky patří především C# a Visual Basic.NET. Aplikace založené na ASP.NET jsou rychlé, neboť jsou již předkompilovány do DLL knihoven.

4.2.2 C#

C# je objektově orientovaný programovací jazyk vyvinutý firmou Microsoft, jako součást prostředí .NET. C# byl později schválen jako standart ECM a ISO. Syntaxe jazyka je založena na C++ a je také ovlivněna dalšími jazyky jako Delphi¹⁰ a Java.

Cílem jazyka C# je vyvinout jednoduchý, moderní, víceúčelový, objektově orientovaný jazyk. Úmyslem jazyka je také možnost vyvíjet softwarové komponenty, které mohou využívat výhod distribuovaných prostředí. C# také umožňuje psát programy pro embedded systémy (2).

4.2.3 XML-RPC

XML-RPC je protokol pro vzdálené volání procedur (Remote Call Procedur). Jedná se o velice jednoduchý protokol obsahující pouze ty nejnужnější datové typy a příkazy. Jeho implementace je velice snadná. Žádost i odpověď jsou ve formátu XML a musí přesně odpovídat specifikacím. Díky tomu se XML-RPC často používá při komunikaci mezi dvěma technologicky odlišnými systémy. Jako transportní protokol slouží HTTP. V současné době existují implementace téměř pro všechny programovací jazyky (16).

⁹ CLR – Common Language Runtime

¹⁰ V C# se objevuje několik prvků (názvy funkcí, syntaxe) shodných s jazykem Delphi a to díky hlavnímu designérovi a vedoucímu architektovi Microsoftu Adresi Hejlsbergovi, který se dříve podílel na vývoji Borland Delphi a Turbo Pascalu (2).

4.2.4 SOAP

SOAP je protokol pro výměnu XML zpráv přes síť za použití HTTP/HTTPS. SOAP se používá jako náhrada XML-RPC. Je složitější, ale zato nabízí více možností než XML-RPC. Existuje několik způsobů komunikace. Nejvíce používaným způsobem je RPC, kdy klient pošle žádost serveru a ten zašle okamžitě odpověď (13).

SOAP často slouží jako základ pro jiná řešení, např. Web Services.

4.2.5 Web Services

Web Services jsou stejně jako předešlé technologie řešením problému nekompatibility dvou systémů. Ve většině případů fungují Web Services jako webové rozhraní dostupné např. přes Internet a vykonávající na vzdáleném systému požadované služby. Dle definice W3C je komunikace mezi klientem a serverem vedena za použití XML zpráv, které odpovídají standardu SOAP.

Každá Web Service obsahuje WSDL¹¹. Díky WSDL lze v prostředí .NET Frameworku pracovat s Web Services stejně jako s lokálními službami.

4.3 Objektový model

Windows SharePoint Services poskytují vysoce strukturovaný server-side objektový model, který usnadňuje přístup k objektům reprezentujícím různé vlastnosti webových stránek SharePoint. Od objektů nejvyšší struktury můžeme procházet objektovou hierarchii a nalézt objekty, které potřebujete pro získání potřebných hodnot nebo funkcí v našem kódu (7).

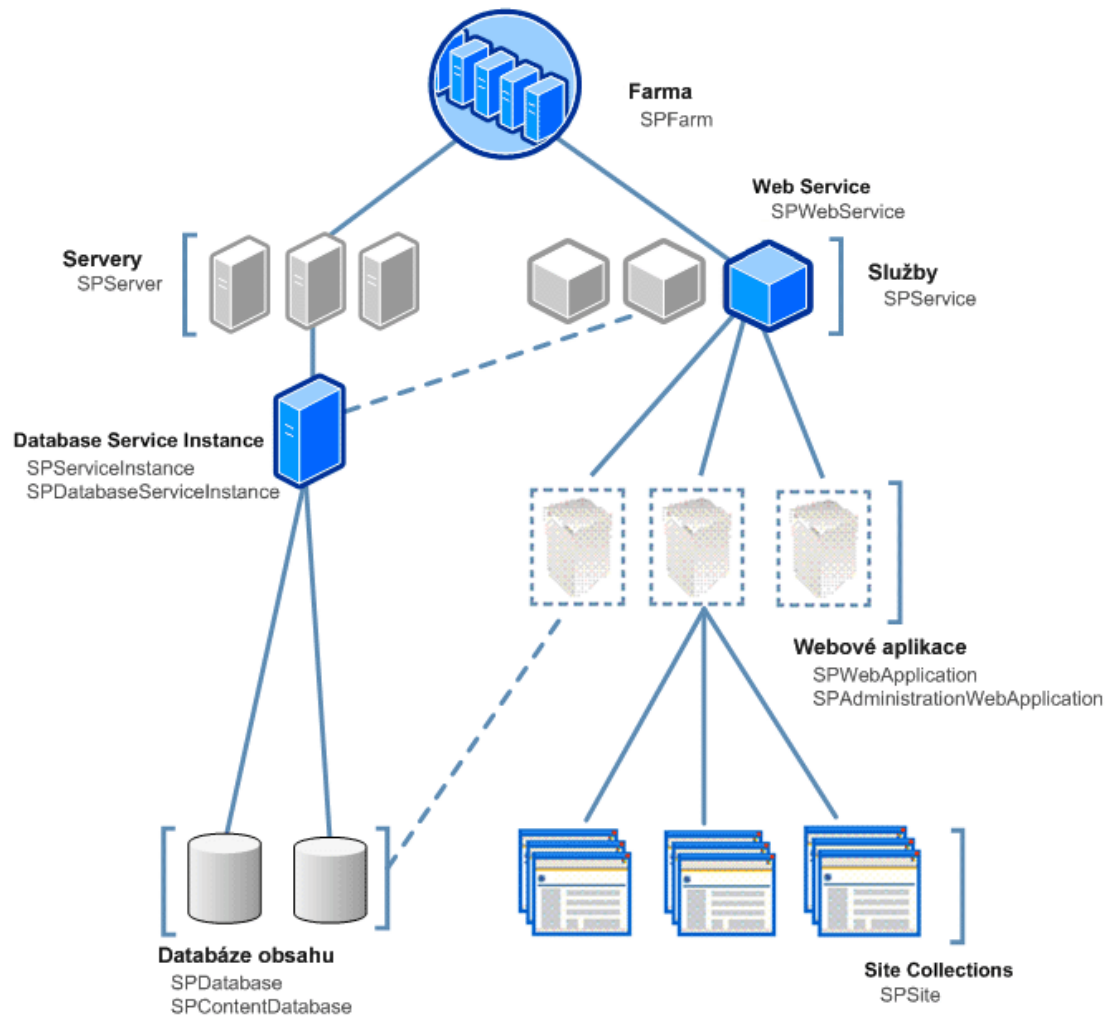
4.3.1 Vstupní bod

Výběr vstupního bodu neboli objektu nejvyšší úrovně, přes který budeme vstupovat do objektové hierarchie, závisí na typu vytvářené aplikace. Pro různé případy volíme jiné vstupní body. Pokud například vytváříme Web Part, Web Service nebo webovou

¹¹ WSDL – Web Services Description Language je XML dokument, který popisuje rozhraní a dostupné služby dané Web Services. Jsou zde popsány použité datové typy, formáty zpráv, vstupní a návratové parametry.

aplikaci, která bude pracovat se site collection, jednotlivými stránkami nebo seznamy, použijeme členy třídy Microsoft.SharePoint.SPContext abychom získaly stávající site collection, web nebo seznam (7).

4.3.2 Serverová architektura



Obrázek 2 – Serverová architektura SharePoint ve vztahu s objektovým modelem (7)

Na *Obrázek 2 – Serverová architektura SharePoint ve vztahu s objektovým modelem (7)* vidíme schematické zobrazení serverové hierarchie. V následujícím textu popíšeme objekty vztahující se k tomuto schématu.

Nejvyšším objektem v objektovém modelu Windows SharePoint Services je **SPFarm**. Odtud lze získat kolekce všech serverů a služeb ve farmě.

Objekt **SPServer** reprezentuje fyzický server. Vlastnost `ServiceInstance` umožňuje přístup k seznamu jednotlivých instancí služeb, které běží na daném počítači.

Každý objekt **SPService** reprezentuje logickou službu nebo aplikaci instalovanou v serverové farmě. Typy, které **SPService** dědí, jsou např. objekty pro služby Windows jako služba časovač, vyhledávání, Microsoft SQL Server, databázové služby a další. Také dědí objekty pro Web Services jako Windows SharePoint Services nebo služby v systému Microsoft Office.

Prostřednictvím **SPWebService** můžeme přistupovat k nastavením pro specifické logické služby nebo aplikace. Vlastnost `WebApplications` nám poskytuje seznam webových aplikací, na kterých běží daná služba.

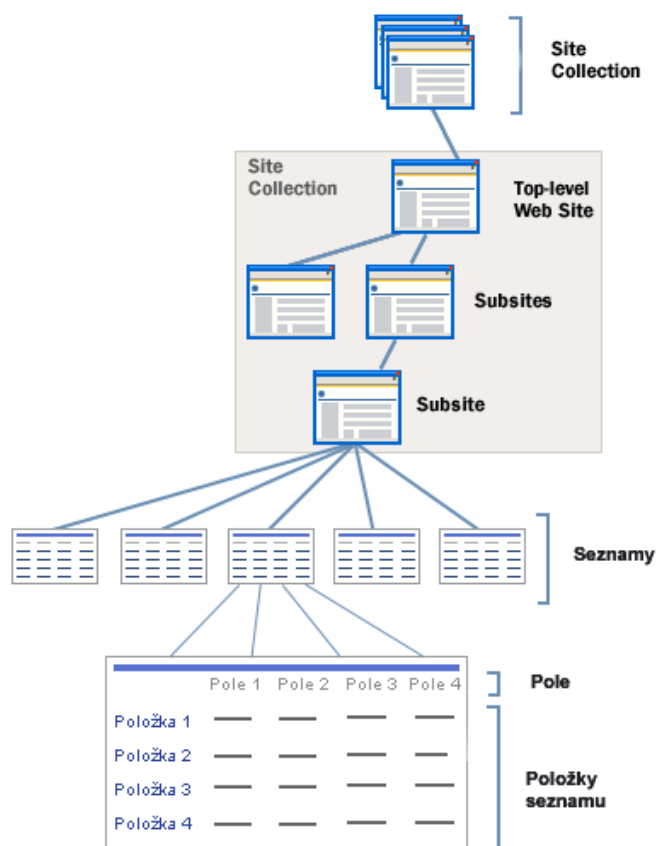
Jednotlivé databázové instance služeb běžící na serveru reprezentuje objekt **SPDatabaseServiceInstance**. Třída **SPDatabaseServiceInstance** dědí ze třídy **SPServiceInstance** a ta dědí z třídy `Service`. Vlastnost `Databases` vrací kolekci databází obsahu použité v dané službě.

Každý **SPWebApplication** představuje webovou aplikaci v IIS. Tento objekt umožňuje přístup k širokým aplikačním nastavením serverové farmy. Vlastnost `Sites` vrací seznam `site collection` uvnitř webové aplikace a `ContentDatabases` vrací seznam databází obsahu použitých ve webové aplikaci.

Objekt **SPContentDatabase** dědí ze třídy **SPDatabase** a reprezentuje databázi, která obsahuje uživatelská data pro webovou aplikaci SharePoint. Vlastnost `Sites` vrací seznam `site collection`, pro které jsou v databázi uložena data.

SPSiteCollection reprezentuje seznam `site collection` uvnitř webové aplikace. Vlastnost `Item` vrací určenou `site collection` ze seznamu. Pomocí metody `Add` můžeme vytvářet nové `site collection`.

4.3.3 Webová architektura



Obrázek 3 – Webová hierarchie SharePoint ve vztahu s objektovým modelem (7)

Každý objekt *SPSite*, nehlédě na jeho jméno, reprezentuje sadu logicky spojených objektů *SPWeb*. Taková sada se nazývá site collection. Vlastnost *AllWebs* umožňuje přístup do objektu *SPWebCollection*, který reprezentuje seznam stránek uvnitř site collection, včetně webu nejvyšší úrovně. *Microsoft.SharePoint.SPSite.OpenWeb*, metoda třídy *SPSite*, vrací určenou webovou stránku.

Každý site collection obsahuje jakékoliv množství objektů *SPWeb* a každý objekt má členské funkce ke správě webu, včetně jeho šablony a motivu a také zpřístupňuje soubory a složky příslušného webu. Vlastnost *Webs* vrací objekt *SPWebCollection* který reprezentuje všechny subweby konkrétního webu. *Lists* vrací objekt *SPListCollection* který reprezentuje všechny seznamy webu.

Každý objekt *SPList* má atributy a metody pro práci se seznamem a přístupu k jednotlivým položkám seznamu. Metoda *GetItem* může být použita k vykonání query příkazu, který vrátí požadované položky. Vlastnost *Fields* vrací *SPFieldCol-*

lection, který reprezentuje všechna pole nebo sloupce v seznamu. Vlastnost `Items` vrací objekt **SPListItemCollection**, který reprezentuje všechny položky nebo řádky v seznamu.

Každý objekt **SPField** má členy obsahující nastavení pro pole.

Každý **SPListItem** reprezentuje jednotlivý řádek seznamu.

5 Návrh řešení

V této části se budeme zabývat teoretickým návrhem a rozbořem našeho řešení.

Na začátku si musíme vysvětlit, co vlastně chceme dělat. Naším úkolem je vytvořit intranetový portál pro správu dokumentů, běžící na univerzitním serveru. Tento server je připojen do univerzitní domény a přihlašování do systému je přes službu Active Directory. Stejně přihlašování budeme požadovat také na našem portále. Po vytvoření portálu je nutné si určit, které uživatelské role budou přítomny v systému a jaká budou mít práva. Poté rozebere další možná nastavení, která se váží k danému problému.

Aby se mohl portál začlenit do univerzitní struktury, je potřeba přizpůsobit jeho vzhled tak, aby vyhovoval jednotnému vizuálnímu stylu univerzity. Nastíníme, jak by mohl vypadat budoucí vzhled a jaké kroky bude nutno provést.

Jelikož chceme Microsoft Windows SharePoint Services využívat hlavně jako DMS, je nutné vytvořit ve struktuře portálu potřebné knihovny dokumentů. Každá z knihoven bude obsahovat jiný typ dokumentů a bude mít nastavená rozdílná přístupová práva.

K závěru kapitoly popíšeme opravdové jádro problému a hlavní cíl naší práce – propojení našeho portálu, založeného na technologii SharePoint, s jiným portálem. Pod pojmem propojení je myšleno možnost získávat data z jednoho portálu a zpracovat je v druhém. Vzhledem k tomu, že každý portál může být založen na zcela odlišné technologii, je třeba toto propojení realizovat přes nějaké univerzální rozhraní. V našem případě to bude implementace XML-RPC pro klientskou část a Web Services pro serverovou část našeho portálu. Jak již bylo zmíněno, jde hlavně o využití portálu jako DMS, tudíž data, která budeme poskytovat, se budou týkat uložených dokumentů.

Podle těchto návrhů bude poté možné vytvořit samotné řešení.

5.1 Intranetový portál

Potřebujeme vytvořit intranetový portál, který bude primárně zaměřen jako DMS, s možností rozšíření o další služby v budoucnosti. Vzhledem ke koncepci technologie SharePoint není budoucí rozšíření větším problémem.

V naší práci vytvoříme site collection z jedním site nejvyšší úrovně obsahujícím weby pro testování navržených řešení a jeden site obsahující knihovny dokumentů.

5.1.1 Umístění portálu

Chceme, aby byl portál přístupný pouze z vnitřní sítě. Toto nám zajišťuje již samotné umístění serveru na vnitřní doméně univerzity (upceucebny.cz) a také nastavení zón v IIS 7. I přesto, že portál bude dostupný pouze z vnitřní sítě, hrozí zde rizika např. ve formě odposlechu komunikace. Z tohoto důvodu bude spojení probíhat šifrovaně přes protokol HTTPS. Znamená to, že na serveru je nutné umístit portál do webové aplikace naslouchající na portu 433 a vytvořit pro ni příslušný certifikát. V našem případě, kdy se budeme k serveru připojovat ze stejné domény, na které je umístěn, nám postačí self-signed certifikát.

5.1.2 Uživatelské role

Abychom omezili přístup do knihoven dokumentů obsahující soubory pouze pro oprávněné osoby, vytvoříme v systému specifické uživatelské role. Tyto role budou celkem čtyři:

- ***Vlastníci webu*** – uživatelé s plným přístupem. Mají přístup do všech částí portálu a mohou měnit jakákoliv nastavení.
- ***Správci webu*** – mají přístup do všech částí portálu, mohou přidávat, měnit, mazat soubory ale také schvalovat nové soubory, vytvářet workflow, atd.
- ***Zaměstnanci*** – mají omezená práva, nemohou měnit nastavení portálu.
- ***Studenti*** – mají omezená práva, nemohou měnit nastavení portálu.

5.2 Jednotný vizuální styl

Při vytváření designu portálu se budeme řídit jednotným vizuálním stylem univerzity. Zde ovšem není žádný předepsaný design ani layout pro webové stránky, pouze barvy a univerzitní logo. Abychom nevytvářeli něco designově zcela odlišného, než co již na univerzitě existuje, budeme se inspirovat novým univerzitním intranetovým portálem. Kompletně převezmeme design a částečně i layout.

Uspořádání stránky bude intuitivní a snadno použitelné. V horní části se nachází informace o přihlášeném uživateli, včetně uživatelského menu a vyhledávání. Nalevo se pak nachází logo Univerzity Pardubice a pod ním vertikální menu. Hlavní menu je horizontální a nachází se napravo od loga. Obsahuje odkazy na úvodní web, knihovnu dokumentů, externí testovací portál a menu Akce webu (viditelné jen pro uživatele s příslušnými právy). Pod hlavním menu se nachází informační řádek s aktuální cestou. Pod tímto informačním řádkem následuje již samotný obsah stránky.

Celkový grafický návrh založený na originálním designu univerzitního intranetu je v příloze A.

5.3 Knihovna dokumentů

Pro ukládání dokumentů slouží v SharePoint Services 3.0 knihovna dokumentů. V naší práci budeme potřebovat tři knihovny dokumenty pro tři cílové skupiny. Každá z knihoven bude mít nastavená práva přístupu pro jinou skupinu.

Knihovny dokumentů v našem projektu:

- ***Veřejné dokumenty*** – Právo přístupu mají všichni uživatelé včetně anonymních.
- ***Dokumenty pro studenty*** – Přístup do této knihovny je zakázán pro anonymní uživatele. Přístup mají uživatelé ze všech uživatelských skupin.
- ***Vnitřní dokumenty*** – Přístup do knihovny mají pouze uživatelé ze skupiny Zaměstnanci, Správci webu a Vlastníci webu.

Každá knihovna bude obsahovat několik podadresářů a bude naplněna testovacími soubory. Práva pro všechny podadresáře jsou děděna z příslušné knihovny dokumentů.

5.4 Propojení s jinými portály

Výměna informací mezi portálem WSS a externími systémy bude probíhat prostřednictvím Web Services a protokolu XML-RPC.

Web Services jsou dnes brány jako standardní propojení mezi dvěma různými systémy a jsou také standardním rozhraním WSS, proto je využijeme pro poskytování dat z knihovny dokumentů WSS. Vytvoříme vlastní Web Service, která bude poskytovat informace o adresářích v knihovně dokumentů, informace o souborech v určeném adresáři, bude umožňovat vyhledávání souborů podle jejich jména a bude také umožňovat nahrávat soubory do knihovny dokumentů.

Pro připojení do externího systému budeme využívat XML-RPC. Tento způsob propojení je u webových aplikací stále hojně používán, hlavně kvůli jeho snadné implementaci. Jelikož knihovny pro práci s protokolem XML-RPC nejsou v jazyce C# standardně obsaženy, je nutné použít řešení třetí strany. Charles Cook píše již několik let XML-RPC knihovnu pro .NET prostředí. Díky jeho knihovně nazvané XML-RPC.NET lze snadno implementovat a používat tento protokol v .NET aplikacích. S využitím této knihovny vytvoříme Web Party, které budou sloužit jako klientská část a budou zobrazovat data z externího portálu.

6 Tvorba portálu

6.1 Operační prostředí

Jako operační prostředí pro provoz serverů, jak vývojového, tak provozního, byl zvolen virtualizační nástroj VMWare. Použití virtuálního počítače nám poskytuje nezávislost na konkrétním hardwaru, možnost vytvořit na vývojovém počítači totožné prostředí jako na provozním serveru a pohodlný vývoj na stávajícím operačním systému.

6.1.1 Počítačová struktura

Vzhledem k tomu, že naše aplikace nebude příliš vytěžovaná, ale bude složit pouze pro demonstrační účely, postačí nám ve farmě jeden aplikační server. Jako databázový server využijeme SQL Server 2005 Embedded Edition dodávaný s instalací WSS 3.0. Autentizace uživatelů je realizována prostřednictvím doménového řadiče Active Directory v síti UPCE. Klientské počítače jsou všechny počítače připojené v doméně upcecebny.cz a upce.cz.

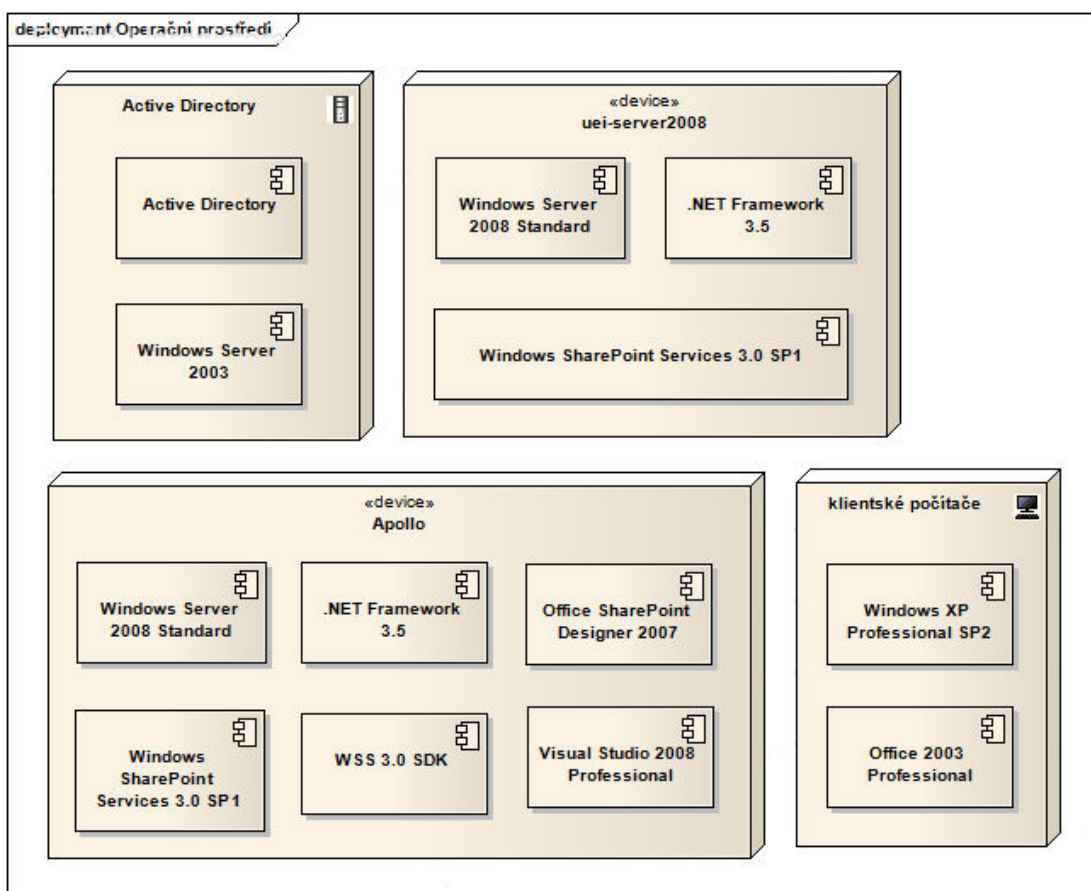
V našem prostředí (Obrázek 4 – Operační prostředí) máme dva WSS servery, doménový kontrolér Active Directory a několik desítek klientských počítačů. Doménový kontrolér Active Directory spravuje všechny uživatelské účty v doméně upce.cz. Díky tomu, že jsou WSS navrženy, aby fungovaly v prostředí Active Directory, nemusíme se starat o zřizování uživatelských účtů přímo ve WSS.

Počítač uei-server2008 je provozním serverem. Je připojen v doméně upcecebny.cz a také k Active Directory. Zde je instalováno pouze to nejnutnější pro běh WSS: Windows Server 2008, .NET Framework 3.5 a WSS 3.0 se Service Pack 1.

Na serveru Apollo je kromě základního softwaru nainstalováno co nejvíce nástrojů ulehčujících práci s WSS. Zde je prováděn veškerý vývoj a testování. Na provozní server jsou data přesunuta až po konečném odzkoušení. Nejdůležitějšími nástroji jsou Visual Studio 2008, Office SharePoint Designer 2007 a WSS 3.0 SDK. Server není

připojen k vnější síti ani k Active Directory. Data na tomto serveru jsou pouze testovací.

Klientské počítače jsou zastoupeny všemi pracovními stanicemi v síti UPCE. Všechny tyto stanice jsou připojeny do domény Active Directory a mají přístup k serveru uei-server2008. Jejich softwarová výbava může být rozdílná, většina však jako operační systém používá Windows XP Professional SP2, obsahuje kancelářský balík Office 2003 a také Internet Explorer 7 jako klientskou aplikaci.



Obrázek 4 – Operační prostředí

6.2 Instalace

Před samotnou instalací musíme ve VMWare vytvořit nový virtuální počítač. Jeho parametry nastavíme pokud možno alespoň na minimum dle hardwarových požadavků. Je potřeba mít k dispozici dostatek místa na pevném disku. Na vývojovém

stroji je to minimálně 13 GB. Na provozním serveru již nepotřebujeme vývojové nástroje, proto postačí méně volného místa.

Při instalaci Windows Server 2008 projdeme grafickým instalátorem. Při celém procesu nemusí uživatel téměř zasahovat. Pouze při zadávání hesla pro administrátora. Na novém systému je pak třeba zprovoznit Internet Information Services a nainstalovat .NET Framework 3.5.

Nyní máme připravený systém pro instalaci WSS 3.0. Pro instalaci na Windows Server 2008 je nutné stáhnout verzi WSS 3.0 s integrovaným Service Pack 1. Na začátku instalace máme na výběr ze dvou možností instalace: Základní a Rozšířená. V rozšířené instalaci lze zvolit instalaci na jeden počítač či připojení do existující farmy nebo její vytvoření. Můžeme zvolit cestu, kam se budou ukládat data a index pro vyhledávání. Dále je nutné nastavit informace pro připojení k databázovému serveru. Tento server musí být samozřejmě již nastaven. Pro naše účely však zvolíme základní instalaci. Zde se kromě umístění dat a indexu, které ponecháme v základním stavu, nenastavuje nic. Jako databáze je použit SQL Server 2005 Embedded Edition, který je obsažen v instalaci. Po dokončení instalace je spuštěna automatická konfigurace, která v IIS vytvoří Webovou aplikaci na náhodném portu pro centrální administraci a druhou Webovou aplikaci na portu 80. V této Webové aplikaci také vytvoří základní site collection.

6.3 Konfigurace

6.3.1 Vytvoření Webové aplikace

Po úspěšné instalaci máme na portu 80 vytvořenou webovou aplikaci. Pro náš projekt je toto umístění však nevhodné. Potřebujeme, aby naše webová aplikace naslouchala na portu 443. Nejjednodušší cesta je v našem případě odstranění stávající webové aplikace a vytvoření nové přes *Centrální správu serveru SharePoint*. V sekci *Správa aplikací* zvolíme *Odstranit Webovou aplikaci*. Odstraníme jak její databázi obsahu, tak jí samotnou z IIS. Poté v té samé sekci zvolíme vytvořit nebo rozšířit webovou aplikaci, pokračujem odkazem *Vytvořit webovou aplikaci*. V nastavení no-

vé webové aplikace změním pouze port, povolím anonymní přístup a povolím zabezpečení SSL. Potvrdím vytvoření.

Aby adresa, na níž se budou nacházet naše stránky, byla dostupná, je nutné vytvořit pro ni certifikát. Vzhledem k tomu, že provozní server běží v doméně upceucebny.cz a klienti kteří se k němu budou připojovat taktéž, postačí vytvořit v IIS self-signed certifikát. Ten poté připojíme k příslušné webové aplikaci.

6.3.2 Vytvoření site collection

Do nové aplikace musíme ještě vytvořit site collection. V sekci *Správa aplikací* přejdeme na odkaz *Vytvořit kolekci webů*. Zde vybereme jako šablonu pro vytvoření *Prázdný web* a doplníme jméno *Primárního správce kolekce webů*. Tímto máme vytvořeny prázdné stránky pro další úpravy.

6.3.3 Vytvoření základní struktury stránek

Celý portál se bude skládat ze dvou sites. První se bude jmenovat Domovská stránka a bude obsahovat veškeré Web Party, které vytvoříme, přehledně rozdělené na stránkách. Druhý site se bude jmenovat Knihovna dokumentů a bude obsahovat tři knihovny dokumentů dle návrhu.

Úvodní site s úvodní stránkou je již vytvořený. Nyní mu musíme změnit nadpis. To provedeme přes menu *Akce webu >> Nastavení webu* a v sekci *Vzhled a chování* vybereme položku *Nadpis, popis a ikona*. Poté musíme vytvořit stránky pro Web Party. Opět přes menu *Akce webu* ale tentokrát zvolíme *Vytvořit* a dále *Stránka webové části*. Zde vyplníme název stránky. Jako šablonu rozložení vybereme *Celá stránka visle* a nakonec vybereme knihovnu dokumentů, do které se stránka uloží. Takto vytvoříme celkem pět stránek: *Sample.GetEmployeesAll*, *Sample.GetFilesAll*, *Výpis složek*, *Výpis dokumentů* a *Hledání dokumentů*. Nakonec upravíme levé menu tak, aby odpovídalo našemu návrhu, viz příloha 1A. Menu nastavíme přes *Nastavení webu >> Snadné spuštění*. Pomocí tlačítek *Nový nadpis* a *Nový odkaz* vytvoříme menu v požadované podobě.

Site Knihovna dokumentů vytvoříme pomocí *Akce webu >> Vytvořit >> Weby a pracovní prostory*. Na stránce pro tvorbu nové site vyplníme jako nadpis Knihovna dokumentů. Adresa site bude *https://uei-server2008.upceucebny.cz/knihovna*. Jako poslední vybereme šablonu *Pracovní prostor dokumentů*. Do prázdného site nyní vložíme tři knihovny dokumentů: *Veřejné dokumenty*, *Dokumenty pro studenty* a *Vnitřní dokumenty*. *Akce webu >> Vytvořit >> Knihovna dokumentů*. Zde pouze vyplníme název, zapneme verzování dokumentů a vybereme výchozí šablonu pro tvorbu dokumentů. V knihovnách dokumentů vytvoříme ukázkové adresáře a nahrajeme několik ukázkových dokumentů. Na závěr musíme opět ručně přeformátovat levé menu.

6.3.4 Nastavení práv

V Knihovně dokumentů máme několik knihoven dokumentů, každou s jiným obsahem určeným pro jiné uživatele. Musíme zajistit, aby se členové příslušné skupiny dostali pouze tam, kam mají přístup. To zajistíme přidělením práv pro jednotlivé knihovny dokumentů.

Na webu chceme povolit možnost anonymního přístupu. Můžeme tento přístup povolit u jednotlivých částí webu nebo najednou pro celý web. V menu *Akce webu* zvolíme *Nastavení webu*. Přejdeme na stránku *Rozšířená oprávnění* a v menu *Nastavení* zvolíme *Anonymní přístup*. Vybereme nastavení *Celý aktuální web*. Nyní máme povolený anonymní přístup pro celý web.

Aby se mohli nyní nastavit práva pro jednotlivé knihovny, je nutné vytvořit odpovídající uživatelské skupiny. Na stránce *Nastavení webu* přejdeme na *Osoby a skupiny* a z menu *Nový* vybereme *Nová skupina*. Takto vytvoříme pět skupin:

- ***Vlastníci webu*** – mají veškerá práva; do této skupiny přidáme jednotlivé uživatele.
- ***Správci webu*** – do této skupiny přidáme vybrané uživatele nebo skupinu, kteří budou moci schvalovat a upravovat položky; nemohou měnit strukturu a hlavní nastavení webu.

- **Zaměstnanci** – jelikož zaměstnanci jsou v Active Directory rozděleni do skupin podle fakult, je jedinou společnou skupinou pro všechny zaměstnance skupina *UPCE\g_wifi_employees*. Tuto skupinu využijeme pro přihlášení zaměstnanců na portál. Zaměstnanci mohou zobrazovat, přidávat, aktualizovat a odstraňovat položky.
- **Studenti** – do této skupiny přidáme studenty, jejichž účty se v Active Directory nacházejí ve skupině *UPCE\all_students*. Mohou zobrazovat, přidávat, aktualizovat a odstranit položky.

Po vytvoření uživatelských skupin přiřadíme tyto ke knihovně dokumentů a tím zabráníme anonymnímu přístupu. Na stránce *Nastavení webu* přejdeme na *Seznamy a knihovny webu*. Vyberme příslušnou knihovnu a v sekci *Oprávnění a správa* zvolíme *Oprávnění pro tuto knihovnu dokumentů*. Nyní dědí knihovna dokumentů oprávnění z nadřazeného webu. Toto změním v menu *Akce >> Upravit uživatelská oprávnění*. Pro knihovnu Dokumenty pro studenty ponecháme všechny skupiny. V případě vnitřních dokumentů odebereme ze seznamu skupinu Studenti. Tímto máme nastavený anonymní přístup na celý web s výjimkou Dokumenty pro studenty a Vnitřní dokumenty.

7 Implementace propojení s jinými systémy

7.1 Programová hierarchie

Implementační část se bude skládat ze tří hlavních assemblies: *BakalarkaWebControl.dll*, *BakalarkaWebService.dll*, *CookComputing.XmlRpcV2.dll*. Aby mohli být tyto assemblies použity v prostředí SharePoint, musí splňovat několik požadavků:

- Všechny assemblies musí být podepsané – mít tzv. strong name. Každá podepsaná assembly má Public Key Token, což je veřejný klíč, podle kterého může být identifikována. Public Key Token lze získat pomocí utility sn.
- Assemblies musí být umístěny v Global Assembly Cache nebo v adresáři *bin*, který se nachází v kořenovém adresáři webové aplikace WSS.
- Musí být přidány do seznamu bezpečných prvků v souboru *web.config*, nacházejícím se taktéž v kořenovém adresáři webové aplikace WSS. Příklad takového záznamu:

```
<SafeControl Assembly="BakalarkaWebControl, Version=1.0.0.0, Culture=neutral, PublicKeyToken=62bc0b560ec0a097" Namespace="BakalarkaWebControl" TypeName="*" Safe="True" />
```
- V souboru *web.config* musí být také přidány do seznamu assemblies. Příklad:

```
<add assembly="BakalarkaWebControl, Version=1.0.0.0, Culture=neutral, PublicKeyToken=62bc0b560ec0a097" />
```
- V souboru *web.config* je třeba upravit parametr *level* vlastnosti *trust*. V našem případě, kdy máme assemblies umístěny v adresáři *bin*, nastavíme parametr *level* na Full:

```
<trust level="Full" originUrl="" />
```

. Toto nastavení nám dovoluje používat ve všech třídách objektový model bez omezení.

V této podkapitole si popíšeme, co je součástí každé assembly. *CookComputing.XmlRpcV2.dll* je knihovna třetí strany a byla popsána dříve, proto se o ní již nebudeme zmiňovat.

7.1.1 BakalarkaWebControl.dll

Jsou zde definovány dva datové typy pro práci s informacemi souborů a adresářů. Tyto typy jsou reprezentovány třídami *FileInfo* a *FolderInfo*.

Dále jsou zde všechny námi vytvořené Web Party, respektive jejich zástupné třídy, které se starají o jejich načítání v prostředí SharePoint a třídy pro zpracování požadavků z User Web Control daného Web Partu. Jedná je o třídy:

- *FileList + FileListControl*
- *FolderList + FolderListControl*
- *SearchDocuments + SearchDocumentsControl*
- *XMLRPCDocuments + XMLRPCDocumentsControl*
- *XMLRPCEmployees + XMLRPCEmployeesControl*

Na závěr zde najdeme třídy tvořící v našem modelu aplikační vrstvu. Jsou to třídy, které zpracovávají data mezi Web Part či Web Services a databází obsahu WSS:

- *ExportFiles*
- *ImportFiles*
- *ExportFolders*

7.1.2 BakalarkaWebService.dll

Obsahuje Web Service pro práci s daty knihoven dokumentů. Tato Web Service je tvořena třídou *BakalarkaSample*, která je také jedinou třídou v tomto assembly.

7.2 Klientská část

Pokud chceme získávat data z externích systémů a zpracovávat je v prostředí SharePoint, musíme pro tento účel vytvořit rozhraní. Data zobrazovaná na portále WSS, musí být vždy součástí nějakého Web Partu. Vzhledem k tomu, že ve WSS žádný takový Web Part není, musíme ho napsat. Samotný Web Part však nezpracovává data z databáze WSS. K tomuto účelu musíme vytvořit tři třídy, které budou sloužit jako aplikační vrstva.

7.2.1 Aplikační vrstva

ExportFolder

Tato třída poskytuje informace o adresářích z knihovny dokumentů. Obsahuje hlavní metodu *GetAllFolders*, která vrací seznam všech adresářů z knihovny dokumentů, ke kterým má přihlášený uživatel přístup. Pro uchování těchto informací slouží *FolderArrayList* typu *ArrayList*. Adresářová struktura knihovny dokumentů se prochází rekurzivně pomocí metody *FindSubFolder*. Při nalezení podadresáře je tento uložen do *FolderArrayList*.

ExportFiles

Obsahuje metody *GetFiles* a *SearchDocument*, které slouží k získání informací o souborech z knihovny dokumentů. Argumentem *GetFiles* je adresář, jehož soubory budou vráceny v poli typu *FileInfo*. Metoda *SearchDocument* vrací výsledky hledání. Soubory jsou hledány v celé knihovně dokumentů podle jména, které je argumentem této metody. V metodě *GetQueryResult* je sestaven dotaz v jazyce CAML, podle které se vyhledá soubor v databázi obsahu. Dotaz v jazyce CAML vypadá takto:

```
SPSiteDataQuery query = new SPSiteDataQuery();
query.ViewFields = "<FieldRef Name=\"FileLeafRef\"/>" +
    "<FieldRef Name=\"FileRef\"/><FieldRef Name=\"Modified\"/>" +
    "<FieldRef Name=\"Editor\"/><FieldRef Name=\"FileSizeDisplay\"/>";
query.Lists = "<Lists ServerTemplate=\"101\"/>";
query.Query = "<Where><Contains><FieldRef Name=\"FileLeafRef\" />" +
    "<Value Type=\"Computed\">" + fName + "</Value></Contains></Where>";
```

Import Files

Obsahuje metodu *UploadFile*, která slouží k nahrání souboru do knihovny dokumentů. Vstupními parametry jsou *fileName*, *fileContents* a *destinationPath*, které představují jméno souboru, pod jakým bude uložen v knihovně, dále obsah souboru v poli typu *byte* a cílový adresář, do kterého se má soubor uložit. Návrátová hodnota představuje řetězec, který informuje o úspěchu/neúspěchu operace.

7.2.2 Tvorba Web Partů

Web Part se tvoří jako třída, která dědí z *WebPart*, ze jmenného prostoru *Microsoft.SharePoint.WebPartPages*. K renderování dat na stránce je nutné přepsat funkce *CreateChildControl* a *RenderWebPart*. V těchto funkcích lze ručně definovat obsah, který se vyrenderuje na stránce.

Pro ulehčení práce však můžeme použít prvek ASP.NET 2.0 zvaný User Web Control. Jedná se o jakýsi fragment webové stránky, který může obsahovat HTML a ASP.NET prvky a má také svoji obslužnou třídu. Veškerý webový obsah je definován v User Web Control a programová logika v jeho obslužné třídě. Třída Web Partu pak slouží pouze jako kontejner, který za pomoci funkce *Page.LoadControl* v metodě *CreateChildControl* nahrává User Web Control jako součást Web Partu.

Samotný Web Part budeme tedy tvořit hlavně v obslužné třídě, která se nachází v souboru s příponou *.ascx.cs*. Všechny Web Party mají velice podobnou strukturu. Všechny obsahují metody *Page_Load*, *WriteData* a *MakeTable*. Třídy, které pracují s XML-RPC mají navíc interface *IBakalarkaSample*, který zpřístupňuje XML-RPC metody.

Page_Load

Je to metoda, volána při načítání stránky. Obsahuje většinou pouze volání jiných metod. Jsou zde ošetřeny výjimky, které mohou při těchto voláních nastat.

WriteData

Tato metoda slouží pro výpis dat na webovou stránku. Obsahuje ošetření možných výjimek a volá metodu *MakeTable*, která vrací zformátovanou tabulku, která se poté vypíše na stránce.

MakeTable

Tato metoda dělá jediné: ze vstupního argumentu typu *FileInfo[]* setaví HTML tabulku, kterou vrátí jako řetězec.

7.3 Serverová část

Jako prostředek pro poskytování dat ze serveru WSS jsme zvolili Web Services. Aby propojení bylo univerzální a použitelné ve všech systémech, bylo by dobré zaimplementovat do portálu také XML-RPC Services. Bohužel implementace XML-RPC Services do SharePoint je složitější a nepodařilo se mi tuto službu zprovoznit. Lze provozovat XML-RPC Services vně SharePoint Services a využít je pouze jako převodník mezi XML-RPC a Web Services. V takovém případě se na venek vše tváří jako standardní XML-RPC.

7.3.1 Web Services

Vytvoření SharePoint Web Services je složitější nežli tvorba klasické Web Services. Na začátku musíme vytvořit klasickou Web Service, tj. soubor *BakalarkaSample.asmx* spolu s kódem na pozadí. Kód na pozadí bude součástí assembly *BakalarkaWebService.dll*, kterou musíme nahrát do adresáře *bin*, jak bylo popsáno v předchozí kapitole. Assembly musí být samozřejmě podepsaná. K vytvoření strong name použijeme stejný klíč jako v případě tvorby *BakalarkaWebControl.dll*.

Třída *BakalarkaSample* dědí ze třídy *System.Web.Services.WebService* a obsahuje čtyři jednoduché metody. Kromě jiného používá třídy ze jmenného prostoru *BakalarkaWebControl*. Každá z metod je velice jednoduchá. Představují pouze jakousi mezivrstvu mezi aplikační vrstvou (třídy *ExportFiles*, *ExportFolders*, *ImportFile*) a prezentační vrstvou (aplikace využívající Web Service).

Takto např. vypadá metoda *GetFolders*:

```
[WebMethod]
public FolderInfo[] GetFolders()
{
    ExportFolder folders = new ExportFolder();
    return folders.GetAllFolders();
}
```

Celá metoda se skládá pouze z vytvoření instance třídy *ExportFolders* a volání její členské funkce *GetAllFolders*.

7.3.2 Metody třídy *BakalarkaSample*

Metoda *UploadDocument* má vstupní parametry *fileName*, *fileContents* a *destinationPath*. Návrátová hodnota je řetězec, který obsahuje informace o úspěšnosti provedení uploadu dokumentu na server.

Metoda *GetFolders* nemá vstupní parametry a vrací pole typu *FolderInfo*, viz předchozí kapitola.

Metoda *GetFiles* vrací pole typu *FileInfo* a jejím vstupním parametrem je *folderPath*, relativní cesta k požadovanému adresáři.

SearchDocumentByName vrací výsledky hledání v poli typu *FileInfo*. Vstupním parametrem je hledaný název dokumentu, *docName*.

7.3.3 Zprovoznění Web Services

Samotná Web Service, která bude fyzicky volána, tedy soubor *BakalarkaSample.asmx* musíme umístit do virtuálního adresáře *_vti_bin*, který je fyzicky mapován na adrese *Local_Drive:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\12\ISAPI*.

Pomocí programu *disco.exe* vytvoříme soubory DISCO a WSDL. V těchto souborech upravíme absolutní cesty na relativní a změníme jejich názvy na *BakalarkaSample-disco.aspx* a *BakalarkaSamplewsdl.aspx*. Tyto popisné soubory jsou umístěny rovněž ve virtuálním adresáři *_vti_bin*.

Na adrese *https://uei-server2008.upceucebny.cz/_vti_bin/BakalarkaSample.asmx* by měla být nyní dostupná funkční Web Service. Pokud nyní budeme vytvářet aplikaci v prostředí .NET, pomocí webové reference přidáme námi vytvořenou Web Service a můžeme používat její třídy, funkce a typy.

7.4 Externí portál

Externí portál je jednoduchý web sloužící k otestování funkční implementace našeho propojení. Stránky jsou psány v ASP.NET 2.0 a v IIS jsou umístěny v defaultní

webové aplikaci naslouchající na portu 80. Jsou tedy dostupné na adrese <http://uei-server2008.upceucebny.cz>.

Souborová struktura se skládá pouze z několika souborů. V hlavním adresáři jsou to *Default.aspx*, *Default.aspx.cs*, *web.config* a *XMLRPCServer.ashx*. V adresáři *App_WebReferences* se nacházejí reference na externí Web Services. Jedná se konkrétně o soubory *BakalarkaSample.wsdl* popisující danou Web Service a *BakalarkaSample.discomap* popisující cestu k této Web Service. Knihovny potřebné pro běh webové aplikace jsou umístěny v adresáři *bin*. Nalezneme zde knihovnu *CookComputing.XmlRpcV2.dll*.

7.4.1 Popis řešení

Aplikace obsahuje pouze jednu webovou stránku a tou je *Default.aspx*. Zde je definován vzhled a rozložení stránky. Stránka je rozdělena na tři oddíly. Horní oddíl obsahuje pole pro zadání přihlašovacích údajů. Dolní část je rozdělena na levý a pravý sloupec. V levém sloupci jsou popsány testované metody s odkazy na jejich použití. Levý sloupec slouží pro zobrazení výsledků, které vrátí daná metoda. Celkový grafický návrh je v příloze B.

Default.aspx je jednoduchá ASPX stránka. Na začátku jsou definovány kaskádové styly. Logické rozdělení prvků na stránce je řešeno elementy DIV. Použita je také technologie ASP.NET AJAX, díky které lze odesílat data z dolní části stránky bez nutnosti znovu načítat celou stránku. Tato vlastnost se zde využívá kvůli zachování přihlašovacích údajů v polích formuláře. Pro vykonání akcí jsou použity ASP prvky jako Button, LinkButton či DropDownList.

Samotný výkonný kód je umístěn v souboru *Default.aspx.cs*. Všechny atributy i metody se nachází ve třídě *_Default*, která dědí ze *System.Web.UI.Page*. Jediným atributem třídy je instance Web Service *BakalarkaSample*. Kromě metody *Page_Load*, ve které se po načtení stránky ověřují přihlašovací údaje, obsahuje třída metody pro obsluhu aktivních prvků na stránce a metody pro výpis dat. V každé metodě pro výpis se vždy vypíše tabulka s daty získanými s příslušné metody Web Service *BakalarkaSample*.

Celý model je opět postaven jako trojvrstvý. ASPX stránka představuje prezentační vrstvu. Web Service vrstvu aplikační, která zpracovává data do požadovaného formátu. SharePoint, respektive jeho databáze obsahu, tvoří datovou vrstvu.

7.4.2 Poskytované služby

Externí portál obsahuje také XML-RPC Service. Ta se nachází v souboru *XMLRPCServer.ashx*.

K dispozici jsou dvě jednoduché metody v třídě *XmlRpcServer*. Třída *XmlRpcServer* dědí ze třídy *CookComputing.XmlRpc.XmlRpcService* a obsahuje struktury *EmployeeInfo* a *FileInfo*, které definují návratové hodnoty metod. Dva atributy *employees* a *files* představují pole těchto dvou struktur a jsou naplněny testovacími daty hned při inicializaci. Metody *BakalarkaSample.GetEmployeesAll* a *BakalarkaSample.GetFilesAll* pouze vrací tyto atributy.

8 Závěr

Práce se co nejvíce drží zadaných cílů. Byly popsány CMS a DMS, vysvětleny jejich rozdíly a krátce popsány také nástroje k jejich realizaci. Byly popsány možnosti konfigurace Windows SharePoint Services a také jejich instalace. Dále byla podrobně popsána praktická část týkající se návrhu a implementace propojení s jinými systémy.

Hlavním cílem práce byla praktická část, tedy vytvoření rozhraní pro komunikaci mezi dvěma technologicky odlišnými systémy. Cílem nebylo vytvořit rozsáhlou aplikaci s univerzálními možnostmi nasazení ale pouze ukázat, jakým způsobem se dá takový problém řešit. Možnosti implementace byly popsány jednak pro technologii Web Services a také pro technologii XML-RCP. Obě technologie byly úspěšně implementovány do prostředí Microsoft Windows SharePoint Services a na demonstračních příkladech byla ukázána funkčnost těchto řešení.

Jelikož pro mne byl Windows SharePoint Services a s ním spojené technologie novinkou, zabralo mi nejvíce času naučit se pracovat s platformou .NET, jazykem C# a ASP.NET 2.0. Při samotném vývoji bylo mnohdy obtížné najít řešení určitých problémů, ať už v internetových zdrojích nebo v dokumentaci WSS, protože ne vždy se jednalo o standardní řešení.

Z této práce jsem si odnesl hodně nových zkušeností, které chci i nadále rozvíjet. Toto téma pro mne bylo velkým přínosem a jsem rád, že sem si ho mohl zvolit.

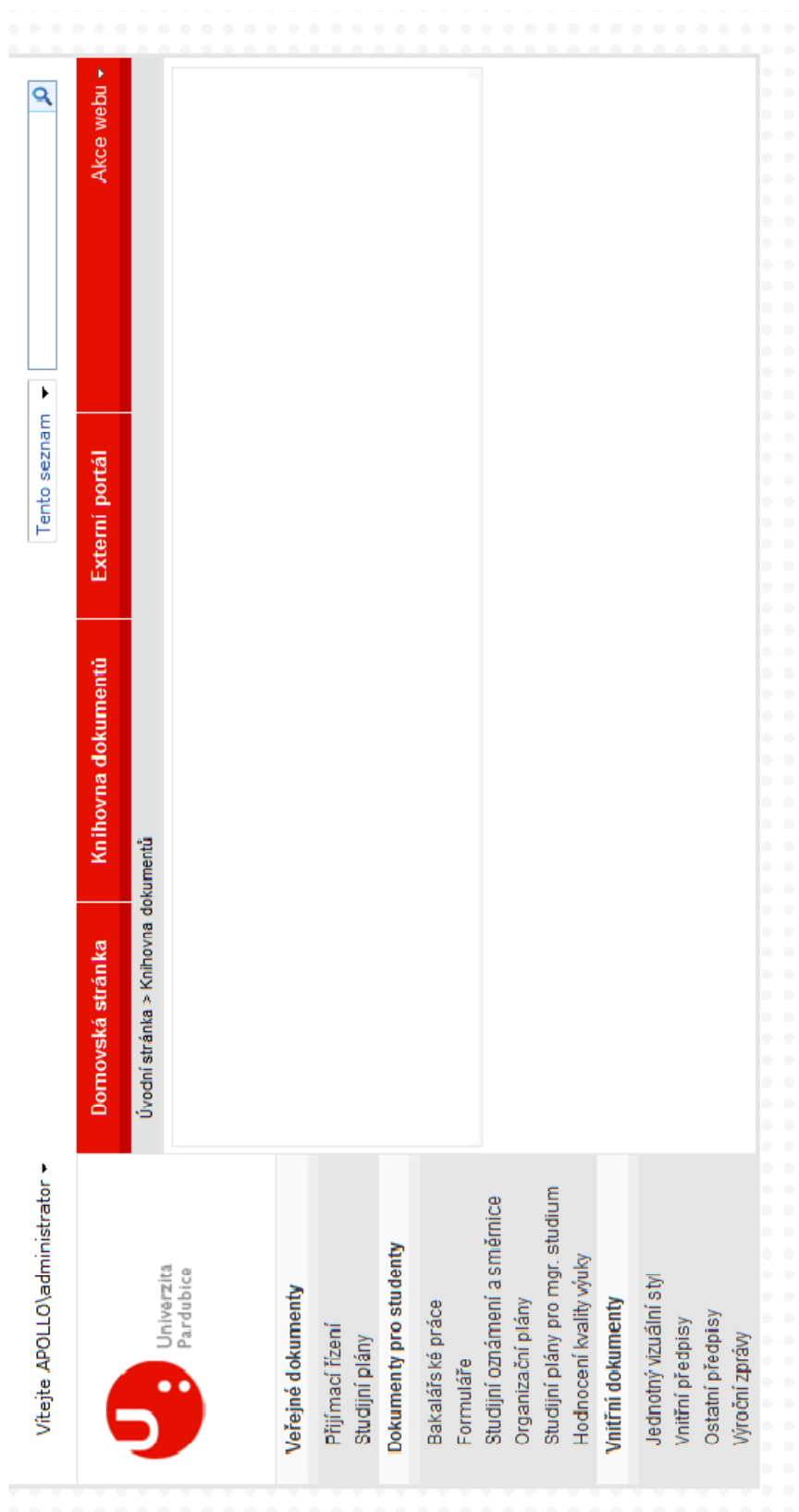
Použité zdroje

- 1) *ASP.NET* [online]. c2008, [cit. 2008-09-06]. Dostupné z:
<<http://en.wikipedia.org/wiki/ASP.NET>>.
- 2) *C Sharp (programming language)* [online]. c2008, [cit. 2008-09-06]. Dostupné z:
<[http://en.wikipedia.org/wiki/C_Sharp_\(programming_language\)](http://en.wikipedia.org/wiki/C_Sharp_(programming_language))>.
- 3) *Document management system* [online]. c2008, [cit. 2008-09-06]. Dostupné z:
<http://en.wikipedia.org/wiki/Document_Management>.
- 4) KERBER, Sean, Michael. *Choose between a commercial, open source, or customized CMS* [online]. 18. 7. 2003, [cit. 2008-09-06]. Dostupné z:
<http://articles.techrepublic.com.com/5100-10878_11-5054863.html>.
- 5) LÁVIČKA, Jan, SOBOTKA, Miloš, KUTĚJ, Tomáš. *Technologie Microsoft Office SharePoint 2003*. 1. vyd. Praha: Computer Press, 2006. 336 s. ISBN 80-251-0976-3.
- 6) MICROSOFT. *Microsoft Office SharePoint Server 2007* [webcast]. 18. 10. 2006, [cit. 2008-09-06]. Dostupné z:
<<http://download.microsoft.com/download/B/E/3/BE3FCEF7-8933-4957-912B-FA0F10A1726C/OFS01.zip> >
- 7) MICROSOFT. *Microsoft Windows SharePoint Services 3.0 SDK* [soubor CHM]. c2007, [cit. 2008-09-06].
- 8) MICROSOFT. *Windows SharePoint Services 3.0 a Visual Studio 2008* [webcast]. 15. 2. 2008, [cit. 2008-09-06]. Dostupné z:
<http://download.microsoft.com/download/a/c/3/ac3174a9-de33-47f8-81ff-dee8c39999a4/VS2008_WSS.zip>.
- 9) MICROSOFT. *Windows SharePoint Services 3.0 Overview* [online]. c2008, [cit. 2008-09-06]. Dostupné z: <[http://technet.microsoft.com/cs-cz/windowsserver/sharepoint/bb684453\(en-us\).aspx](http://technet.microsoft.com/cs-cz/windowsserver/sharepoint/bb684453(en-us).aspx)>.

- 10) MICROSOFT. *What is document management?* [online]. c2008, [cit. 2008-09-06]. Dostupné z: <<http://technet.microsoft.com/en-us/library/cc261933.aspx>>.
- 11) Open source software [online]. c2008, [cit. 2008-09-06]. Dostupné z: <http://cs.wikipedia.org/wiki/Open_source>.
- 12) ROBERTSON, James. *So, what is a content management system?* [online]. 3. 6. 2003, [cit. 2008-09-06]. Dostupné z: <http://steptwo.com.au/papers/kmc_what>.
- 13) SOAP [online]. c2008, [cit. 2008-09-06]. Dostupné z: <<http://en.wikipedia.org/wiki/SOAP>>.
- 14) SVOBODA, Jan. *Vývoj nad Microsoft Windows SharePoint Services*. Brno: Masarykova univerzita, Fakulta informatiky, 2007. 35 s., 5 s. příloh. Vedoucí bakalářské práce Mgr. Petr Steinmetzov.
- 15) *Web content management system* [online]. c2008, [cit. 2008-09-06]. Dostupné z: <http://en.wikipedia.org/wiki/Web_content_management_system>.
- 16) XML-RPC [online]. c2008, [cit. 2008-09-06]. Dostupné z: <<http://en.wikipedia.org/wiki/XML-RPC>>.

Přílohy

Příloha A – grafický návrh portálu



Příloha B – grafický návrh testovacího webu

Externí portál

Přihlašovací údaje (pouze zapíšete, nepotvrzujete)

Login:

Heslo:

[Přejít na stránky SharePoint Services](#)

Úvod

Tyto stránky slouží k testování funkčnosti Web Services a XML-RPC z portálu Windows SharePoint Services. Níže naleznete klientské metody z Web Service *BakalarkaSample*, které získávají data z externího WSS portálu. U každé metody je krátký popis, co přesně dělá.

Na odkaze pod tímto odstavcem naleznete XML-RPC metody umístěné na tomto webu. Pomocí těchto metod se naopak načítají data do WSS portálu. Jedná se o dvě jednoduché metody, které vrací pole struktur.

[Zobrazit dostupné XMP-RPC metody na tomto webu](#)

Web Services

Dostupné na adrese <https://uei-server2008.upceucebny.cz/vti/bin/BakalarkaSample.asmx>.

BakalarkaSample.GetFolders

Získá seznam adresářů, spolu s jejich nadřazeným adresářem, relativní adresou a počtem souborů. Jedná se o všechny adresáře ze všech knihoven dokumentů z adresy <https://server2008.upceucebny.cz/knihovna>.

[Zobrazit adresářový strom](#)

BakalarkaSample.GetFiles

Pomocí této metody se získá seznam souboru ve vybrané složce. Spolu se jménem obsahuje struktura i velikost, URL, verze a datum změny. Seznam složek je načítán metodou *BakalarkaSample.GetFolders*.

[Načíst seznam složek](#)