

Integrace Java EE aplikace s Enterprise Content Management systémem Alfresco

Java EE application and Enterprise Content
Management system Alfresco integration

Bc. Tomáš Hoférek

Diplomová práce
2010



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

**!!! NASCANOVANY OBOUSTARNY ORIGINAL NEBO KOPIE ORIGINALU 1.
STRANA!!!**

!!! KOPIE ORIGINALU 2. STRANA !!!

ABSTRAKT

Cílem diplomové práce je navázat na bakalářskou práci „Distribuovaná aplikace založená na architektuře J2EE“ a dále rozvinout poznatky získané během realizace. Hlavním tématem pro zpracování diplomové práce je integrace informačních systémů. Informační systémy jsou reprezentovány referenční aplikací bakalářské práce a Enterprise content management systémem Alfresco, ze kterých vznikne nová implementace systému. Stávající referenční aplikace jsou rozšířeny o nově implementované funkce díky integraci modulu JBI s podporou BPEL technologie v GlassFish aplikačním serveru.

Klíčová slova:

Java, Java Enterprise Edition, EJB, Alfresco, JBI, BPEL, webové služby

ABSTRACT

The point of my thesis is to study with theoretical knowledge's Java Enterprise Edition during the implementation of distributed application in thesis „Distributed application based on J2EE“. Main topic of this thesis is the enterprise integration of information systems. Information system was re-engineered on bachelor's applications and Enterprise Content Management system's Alfresco that is used in the new conception of integrated system. The new implementation of integrated system has been extended by new functionality due to JBI environment and BPEL technology deployed to GlassFish application server.

Keywords:

Java, Java Enterprise Edition, EJB, Alfresco, JBI, BPEL, web services

Chtěl bych poděkovat svému vedoucímu Ing. Pavlovi Pokornému, Ph.D. za konzultace, cenné připomínky a rady při zpracování diplomové práce.

Prohlašuji, že

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

Prohlašuji,

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

Ve Zlíně

.....
podpis diplomanta

OBSAH

I. TEORETICKÁ ČÁST.....	9
ÚVOD.....	10
1 OBECNÝ ÚVOD DO ARCHITEKTURY SYSTÉMŮ.....	12
1.1 INTEGRACE SYSTÉMŮ.....	12
1.2 ARCHITEKTURA JAVA EE.....	13
1.3 ECM SYSTÉM ALFRESCO.....	16
2 INTEGRACE SYSTÉMŮ.....	19
2.1 TYPY INTEGRACÍ.....	21
2.1.1 Datová integrace.....	21
2.1.2 Aplikační integrace.....	22
2.1.3 Business process integrace.....	23
2.1.4 Prezentační integrace.....	23
2.1.5 Business-to-Business integrace.....	24
2.2 INTEGRAČNÍ TECHNOLOGIE.....	24
3 ECM ALFRESCO.....	30
3.0.1 Architektura.....	31
3.0.2 Repository.....	32
3.0.3 Automatizace business procesů.....	33
3.0.4 Integrace s externími systémy.....	33
3.1 VLASTNOSTI ECM SYSTÉMU.....	33
3.1.1 Adresářová struktura.....	33
3.1.2 Definice Content Model.....	36
3.1.3 Bezpečnost.....	38
3.1.4 Funkce Alfresco ECM systému.....	41
3.1.5 Implementace business procesů.....	44
3.1.6 Workflow.....	46
3.1.7 Vyhledávání.....	47
3.1.8 Zpracování elektronické dokumentace.....	48
3.1.9 Správa systému.....	50
4 POUŽITÉ TECHNOLOGIE.....	52
II. PRAKTICKÁ ČÁST.....	54
5 ARCHITEKTURA INTEGROVANÉHO SYSTÉMŮ.....	55
5.1 SEZNÁMENÍ SE ZADÁNÍM PROBLEMATIKY.....	55
5.2 SPECIFIKACE SYSTÉMŮ.....	55
6 ANALÝZA SYSTÉMU.....	58

6.0.1	Diagram nasazení.....	58
6.0.2	Funkční specifikace.....	59
6.0.3	Studie případu užití.....	61
6.0.4	Analýza procesů.....	63
6.0.5	Diagram tříd.....	64
6.0.6	Datový model systému.....	66
6.0.7	Databázová vrstva knihovny.....	66
7	IMPLEMENTACE.....	68
7.1	INTEGRACE SYSTÉMU ALFRESCO.....	68
7.1.1	Instalace.....	69
7.1.2	Definice typů.....	69
7.1.3	Adresářová struktura.....	71
7.1.4	Centrální úložiště.....	72
7.2	IMPLEMENTACE BUSINESS PROCESŮ V SYSTÉMU ALFRESCO.....	73
7.3	IMPLEMENTACE BPEL PROCESU.....	76
7.4	IMPLEMENTACE DESKTOP APLIKACE.....	78
7.4.1	Schválení uživatelských online registrací.....	79
7.4.2	Katalog online knih.....	79
7.4.3	Fulltext vyhledávání v katalogu knih.....	79
7.4.4	Zobrazení dokumentů v externím prohlížeči.....	80
7.5	IMPLEMENTACE WEB APLIKACE.....	81
7.5.1	Registrace uživatelů.....	81
7.5.2	Zobrazení elektronických knih.....	81
7.6	IMPLEMENTACE IMPORTNÍ APLIKACE.....	82
	ZÁVĚR.....	83
	CONCLUSION.....	84
	SEZNAM POUŽITÉ LITERATURY.....	85
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	88
	SEZNAM OBRÁZKŮ.....	89
	SEZNAM TABULEK.....	90

I. TEORETICKÁ ČÁST

ÚVOD

Diplomová práce je založena na již zpracovaném tématu bakalářské práce „Distribuovaná aplikace založená na architektuře J2EE“.[11] Mimo uvedené přednosti Java Enterprise platformy v bakalářské práci je záměrem dále detailněji zpracovat problematiku integrace systémů. Většina aplikací v podnikových prostředích není izolována v době rozšířených počítačových sítí od IT infrastruktury, která se skládá z různých systémů sloužících např. pro zpracování bilance skladu, mezd zaměstnanců nebo zajišťující např. pouze emailovou komunikaci.

Řada firem již běžně využívá v IT infrastrukturách různé implementace dokument management systému (DMS) pro zpracování elektronické dokumentace, proces schvalování nebo pouze ve formě pasivního datového úložiště. Uvedené příklady jsou poměrně častou variantou pro implementaci DMS a využití systému je možné poměrně v širokém spektru nasazení. Referenční aplikace založená na Java Enterprise Edition byla integrována s ECM systémem Alfresco primárně pro prezentaci předpokladů již stanovených v bakalářské práci. Hlavním benefitem zmiňovaným pro využití Java platformy byla její nezávislost a poměrně snadné další rozšíření založené na standardech. Původní vrstva byla integrována dle doporučených integračních vzorů s Alfresco systémem. Vytvořené rozhraní mezi systémy omezuje redundantní uložení dat a rozšiřuje funkcionalitu a možnosti implementovaných systémů. Požadavky zákazníků se mění nebo rostou v čase na základě získaných znalostí:

- vlastnosti získané z aktuálního implementovaného systému – vždy je prostor pro již optimalizované řešení,
- technologie zaváděné v organizaci – bezpečnostní standardy, aktualizace a vývoj technologií, komunikačních protokolů.

V úvodní části této práce budou teoreticky popsány možnosti integrace systémů v různých ohledech. Jelikož je celkový proces integrace velmi náročný na realizaci, jsou úvodní části uvedeny doporučené postupy vedoucí k usnadnění vývoje projektu. V další kapitole bude pouze zkráceně popsána architektura distribuovaných systémů na platformě Java Enterprise Edition (Java EE). Další kapitola teoretické částí popisuje systém Alfresco se všemi funkcionalitami, které nabízí a jsou využívány ve funkčních požadavcích na systém. Závěrečná kapitola popisuje použité technologie a softwarové nástroje, které byli využity během implementace a analýzy integrovaného systému.

V úvodu praktické části bude popsán návrh architektury systému, detailní specifikace serverových stanic a jejich konfigurace. Další kapitolou je analýza implemen-

tovaného systému v jazyce UML, která specifikuje požadavky kladené na nově implementovaný systém a jasné hranice systému. Analýza je dále doplněna o datový model referenční aplikace. Implementovaný systém vytvořený na platformě Java Enterprise Edition je integrován s JBI architekturou a dále systémem Alfresco. Mezi systémy je stanovené komunikační rozhraní pro vzájemnou distribuci informací a dat. Rozhraní je realizováno na middleware prostředí pomocí technologií Enterprise Java Beans 3.0, webových služeb a BPEL procesu. Uvedené technologie jsou flexibilním nástrojem zahrnutým ve specifikaci SOA, která je v současné době preferovanou technologií pro integraci enterprise systémů.

Každý z celků systému je specifikován v podkategoriích popisující vlastnosti získané během analýzy. Implementovaný systém je prezentován s ukázkami a příklady zpracované problematiky na referenčních aplikacích. Ukázky referenčních aplikací jsou doplněny do příloh pro ilustraci, v jaké míře byl rozšířen původní systém zpracovaný v bakalářské práci.

1 OBECNÝ ÚVOD DO ARCHITEKTURY SYSTÉMŮ

1.1 Integrace systémů

S nárůstem počtu informací v počítačových systémech se zvyšují požadavky a nároky na dostupnost a prezentaci výsledků uživatelům. Stand-alone aplikace jen stěží dokáží růst a poskytnou komplexní informace, pokud se nejedná o velmi specifický proces nebo problematiku. Existují dvě varianty integrace systémů, první z nich je integrace v enterprise prostředí, s dalším růstem je to „business-to-business“ integrace.[14]

Většina organizací využívá již z historicky daných důvodů starší aplikace, rozdílné platformy architektur a technologií, které ve většině případů nebyly připraveny pro vzájemnou integraci. Náklady spojené s jejich nahrazením nejsou zanedbatelné. Proces změny nebo aktualizace systému obsahující kritická data pro organizaci jsou velmi nákladné na časové, personální a samozřejmě finanční nároky.

Vývoj systému a poskytovaných služeb se v čase rozšiřuje a mnohdy přináší řešení business požadavků a nemalé finanční úspory. Nová řešení jsou založena na modernějších architekturách odlišných od již existujících ve firemních prostředích. Novější vydané aplikace podporují předem definované a otestované konektory, plug-in dále rozšiřuje využití.

Na první pohled jsou patrné klady a zápory integrace aplikací, především náročnost celkového procesu až po finální nasazení v produkčním prostředí. K naplnění uvedených cílů integrace existují různé metodiky, návrhových vzorů a technologií, které se v průběhu let vyvíjely.

Motivace

Motivací pro zavedení integrace je především požadavek na aktuální dostupné informace uložené v různých systémech. Schopnost okamžitého přístupu k informacím nemalou měrou ovlivňuje úspěšnost firmy. Neefektivní řešení, které lze využít v omezených případech je ukládání duplikátních dat. Z uživatelského pohledu je změna aplikací během pracovního procesu, zadávání identických dat do každého systému a doba zpracování jednotlivých operací velkým nedostatkem projevujícím se ve výsledném počtu realizovaných operací během pracovní doby. Důsledkem je hůře vyhodnocená efektivnost a v této souvislosti dražší náklady na zpracování úkolů. Ideálním řešením je informační infrastruktura založená na integrovaných systémech poskytujících dostatečnou množinu dat pro konkrétní business proces.

Problematická je zejména složitost a náklady spojené s integrací systémů, kdy v některých případech projekt prodražuje nedostatečné zmapování procesů a finalizace požadavků. Ať už se jedná o nedostatečně dotaženou specifikaci či naopak přehnané nároky uživatelů, vždy je následkem zbytečné prodražení daného projektu. Terminologie a náhled manažerů zadávající projekt se v mnoha případech zcela liší od architektů realizující integraci.

Přístupy firem lze rozdělit na:

- úspěšné projekty – stanovení reálných cílů, zejména přesná specifikace kritických částí nutných pro integraci,
- nerealizované projekty – již známá potřeba integrace aplikací, projekt je odložen z důvodů nedostatečných finančních prostředků nebo nekončení IT infrastruktury,
- neúspěšné projekty neakceptované v termínech – nedostatečná specifikace požadavků v analýze projektu, zásadní změny v průběhu integrace, do integrace vstupují další systémy v průběhu vývoje atd.

Integrace je pro většinu organizací nejdůležitějším strategickým bodem, zejména díky inovativním obchodním řešením a sdílením podnikových dat. Integrované informační systémy poskytují jednotný efektivní přístup k relevantním informacím. V následující kapitole jsou popsány platformy, které jsou v praktické části implementovány.

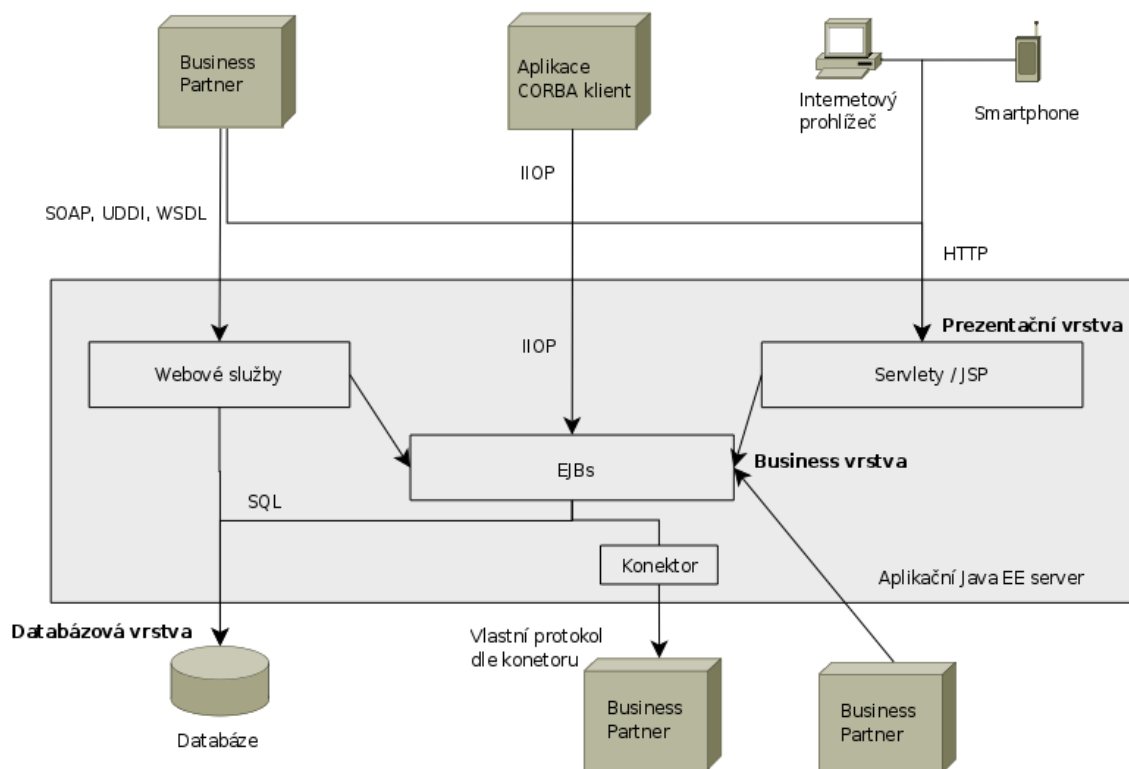
1.2 Architektura Java EE

Architektura Java EE platformy (Obr. 1) byla detailněji popsána v bakalářské práci „Distribuovaná aplikace založená na J2EE“. Popis architektury je proto omezen na soupis nejdůležitějších částí v návaznosti na integraci systémů. Základní znalost problematiky je nutná k porozumění názvosloví a získání obecného přehledu o návrhu aplikace, technologiích využitých během integrační fáze. Samostatnou kapitolou je řešerše systému Alfresco pojednávající o možnostech a variantách pro nasazení ECM systému Alfresco.[2]

Specifikace Java EE je standardem řešení založená na jazyce Java modelu klient-server. Koncept serverových komponent je založen na robustní architektuře poskytující aplikační framework pro tvorbu serverových aplikací. Hlavním důvodem pro nasazení je distribuované a transakční zpracování poskytované Java EE kontejnerem. Serverové aplikace zpravidla dělíme do oddělených vrstev:[3]

- databázová nebo-li perzistentní – data uložená převážně v relační nebo objektové databázi,
- business logika (middleware) – objekty provádějí business funkcionalitu,

- prezentační vrstva – část uživatelského rozhraní přijímající a rozesílající odezvy uživateli.



Obr. 1: Schéma architektury

Pro rozdělení na jednoznačně ohraničené vrstvy hovoří zejména fakt nasazení serverových komponent na odlišný aplikační server a dosažení distribuovanosti systému. Jednotlivě provázané komponenty lze nasadit na různé servery a tím optimalizovat zatížení a celkovou škálovatelnost systému. Nejen uvedené postupy dále potvrzují názor, že pokud se jednotlivé vrstvy prolínají a získávají oboustrannou závislost – patrně se jedná o chybný návrh a koncept systému.[2]

Perzistentní vrstva

Trvale uložená data jsou uchována převážně v databázových systémech optimalizovaných pro ukládání strukturovaných dat. Původní varianta namapování záznamu databázové tabulky na EJB Entity Bean objekt byla nahrazena Java Persistence API (JPA) s novým přístupem specifikace EJB 3.0. JPA definuje relační mapování entit na objektu, od původní specifikace se jedná výrazné vylepšení přístupu k perzistentně uloženým datům. Java Persistence API poskytuje komplexní nástroj pro mapování databázových tabulek včetně relací a dalších funkcí pro práci s databázovými daty.[3]

Business vrstva

Implementace business procesů je hlavní a stěžejní částí implementovaného systému. Veškeré návrhy během analýzy jsou soustředěny do specifikace funkcí požadovaných zákazníkem. Jednotlivé vrstvy mezi sebou navzájem interagují, komunikace je např. běžně realizována na základě události z prezentační vrstvy a požadavek v podobě business funkcí přechází do business vrstvy a dle implementované logiky provede kód např. aktualizace dat zákazníka, kde je následně vyvolána perzistentní vrstva. Uvedený postup je pouze ilustrací jak komunikace probíhá mezi vrstvami.[11]

Implementace business vrstvy je zpravidla vyvíjena pomocí objektů Enterprise Java Beans (EJB). Komponenty jsou nasazeny do EJB kontejneru poskytující běhové prostředí systému. Kontejner tak řídí zátěž, transakce (pokud není implicitně zpracována vlastní logikou), resource pooling (uchování často volaných komponent do lokální cache, vyvoláním se předá objekt z poolu a odpadá tím režie spojená s voláním a inicializací objektu) a cache pro optimalizaci systému.[1]

EJB komponenty jsou v zjednodušené formě rozděleny na:

- lokální rozhraní, které je omezeno na volání pouze v identickém Java Virtual Machine procesu,
- vzdálené rozhraní poskytuje jak napovídá název možnost vzdáleného vyvolání rozhraní mimo instanci Java Virtual Machine nebo odlišné serveru v síťovém prostředí.

Dále dělíme dle typu EJB komponenty:

- Entity Beans nahlíží na objekty jako databázové entity,
- Session Beans zajišťují komunikaci mezi voláním a ve většině případů obsahují business logiku publikovanou ve formě služeb,
- Message-Driven Beans slouží pro asynchronní zpracování dat, vyvolané na základě přijaté nebo odchozí zprávy.

Prezentační vrstva

Prezentační vrstva je zodpovědná za interakci s uživatelem a na základě požadavků dále komunikuje s business vrstvou. Uživatelské rozhraní těžkého desktop klienta je postupně nahrazováno webovým kontejnerem, zajišťujícím prostředí pro nasazení webové aplikace. Těžký klient (desktop aplikace) je využívána ve specifických případech, zejména v případech kdy není vhodné využít vlastnosti http protokolu. Jedná se zejména o kompli-

kovanější uživatelské rozhraní vyžadující častou aktualizaci dat nebo reference na další dialogová okna nebo ovládací prvky.

Tenký klient v podobě webové aplikace je preferován z důvodu klientské nezávislosti, klient nemusí instalovat software na další pracovní stanice a aplikace je tak ve většině případů dostupná po 24 hodin denně. Omezením je pouze dostupnost síťové adresy serveru.

Základní komponenty webové aplikace jsou:

- Servlet,
- Java Server Pages (JSP).

Servlet je základní specifikací pro práci s http protokolem, která je založena na metodách post a get zajišťující získání dat včetně parametrů a odeslání odpovědi. Servlety jsou zpravidla využívány pro zpracování dat. Generování html výstupů není vhodné zejména díky nepřehlednému kódu obsaženému ve zdrojovém kódu servletu. Ideální řešení v servletu provádí konkrétní akci v podobě vytvoření nového zákazníka a předání výsledku JSP stránce informující o výsledku požadované operace.

Java Server Pages jsou optimalizovány pro prezentaci dat a práci s formuláři, struktura vychází z HTML formátu webové stránky. JSP stránky lze kombinovat se značkami zpřístupňujícími Java kód a objekty JSP stránek. Mimo uvedené postupy lze využít tag knihovny obsahující např. ovládací prvky nebo definici značek pro zápis HTML prvků.

Java Server Faces je frameworkem založeným na technologii JSP a posunuje koncept vývoje uživatelského rozhraní. JSP stránky jsou rozšířeny o JSF komponenty zajišťující funkcionalitu např. komponenta datagridu sloužící pro zobrazení dat s možností třídit, vkládat a aktualizovat atd.

1.3 ECM systém Alfresco

Alfresco je open source alternativou k DMS systémům provozovaných v enterprise prostředí a poskytuje následující okruhy pro zpracování elektornické dokumentů:

- Document management,
- Web Content management,
- Records management,
- Image management,
- Content Repository.

Document management je v dnešní době nutností pro organizace, ve kterých dochází ke schvalování a obecně k práci s dokumenty. Oběh dokumentů v papírové formě je neefektivní řešení, nespĺňující pro většinu organizací základní požadavky pro práci s dokumenty. ECM systémy převádějí problematiku zpracování a sdílení dokumentů do elektronické reprezentace. Document management je tak orientován na zpracování a transformaci dokumentů v elektronické podobě.[10]

Web Content Management je implementací portálového systému, užívaný pro publikaci dat na webový server. Web Content Management poskytuje identické nástroje jako document management. Portálové řešení je rozšířeno o editor webových stránek, správu web projektů a dále např. publikaci dat na cílový webový server.

Records management je orientován na uchování dokumentů po celou dobu jejich životního cyklu, od vytvoření až po skartaci. Primárně se jedná o dokumenty, které nejsou v každodenním procesu využívány. Na základě legislativních nebo firemních požadavků musí být vedeny v systému.

Image management zpracovává problematiku transformace dat do elektronické podoby. Mimo základní snímání dokumentů a vytěžování textů pomocí OCR je dále podporováno zpracování dat po transformaci do odlišných formátů po přidání watermark nebo identifikačních kódů.

Content Repository je úložištěm dokumentů poskytující základní sadu služeb pro práci. Primární úložiště metadat a obsahů dále poskytuje fultextové vyhledávání obsahů, standardizované rozhraní pro interakci s ostatními systémy.

Základní jednotkou je počítačový systém, používaný k uchování, archivaci a sdílení elektronických dokumentů. Reprezentace dokumentů je uložena v textové, binární nebo obrázkové podobě. Pokud již existuje papírová předloha, je dokument převeden pomocí skeneru do elektronické podoby a uložen v systému. Výhodou dokumentů uložených v elektronické formě je lépe zálohovaný obsah a dále více možností pro práci. Dokument poskytuje verzování a během pracovního postupu je vždy možné se navrátit k předchozí uložené verzi. Další možností je přehled o zapracovaných změnách v revizích. Metadata a obsah dokumentů je zabezpečen pomocí přístupových práv, která definují pro danou fázi uživatelská oprávnění pro zápis a sdílení informací o dokumentu. Přístupová práva se v čase mění na základě předem definované události, kterou je změna životního cyklu. Dokument je zařazen do typu, na který lze následně definovat příslušný životní cyklus. Životní cyklus obsahuje nastavení akcí pro jednotlivé stavy, ze kterých se skládá. Pro ilustraci je dokument běžně členěn do následujících stavů:

- Draft – inicializace dokumentu,
- WIP – vývoj dokumentu,
- Effective – platnost dokumentu po schválení,
- Obsolete – dokument s ukončenou platností.

Seznam stavů reprezentuje vývoj dokumentu od počátečního stavu až po ukončení platnosti dokumentu.

Workflow je aktivní proces zpracovávající řešení konkrétní business problematiky. Dynamický proces vychází ze šablony workflow. Spuštěná instance workflow na základě uživatele nebo události vytvoří instanci workflow. Workflow následně vykoná seznam aktivit definovaných v šabloně. Mimo aktuální instanci je další variantou workflow ukončit a vyvolat další subworkflow. Výsledkem spuštěného workflow je zpracovaný úkol, kterým je např. schválený dokument, faktura nebo dokončené připomínkování překladu.

Uvedené vlastnosti poskytují komfortní nástroj pro vedení dokumentace v elektronické podobě. Výhodou je méně náročná agenda spojená s nároky na prostor běžných archivů. Uložená data však podléhají chybovým stavům způsobenými havariemi disků, hardwarem nebo přerušáním dodávky elektrické energie. Vhodným návrhem systému lze ovšem z velké části chybové stavy eliminovat. Elektronická reprezentace dokumentů poskytuje komfortní nástroje uživatelům pro práci s dokumenty:

- Fulltext vyhledávání – importovaný obsah je indexován a tím připraven pro dotaz, zda je obsažen text v souboru,
- Vyhledání dle metadat – obsah je označen atributy, které lze využít jako vyhledávací podmínku,
- Třídění – dokumenty lze řadit dle kritérií,
- Organizace dokumentů do skupin – dokumenty lze členit do stromové struktury nebo přidat informaci o dané kategorii.

Nezanedbatelnou službou je archivace dat, v pravidelných intervalech je provedena záloha systému včetně databáze, dat a konfigurací. V případě havárie lze snadno obnovit systém k datu poslední realizované zálohy. Systém je dle potřeby provozován na zrcadlených discích nebo diskových polích. Nastavení bezpečnostních politik pro práci se systémem dále ovlivňuje aplikace fyzického zabezpečení systému. Vhodnou kombinací omezeného přístupu k serverové části klientským stanicím lze vybudovat komplexní zabezpečení ECM systému.

2 INTEGRACE SYSTÉMŮ

Organizace působící na trhu řadu let ve většině případů provozují integrované informační systémy, složené z řady heterogenních existujících aplikací. Aplikace jsou nasazeny různými způsoby, mezi hlavní cesty patří:[14]

- vývoj řízený vlastním IT oddělením,
- uživatelsky sestavené outsource řešení,
- komerční a ERP aplikace.

Aplikace jsou postaveny na odlišných platformách, různých technologiích a programovacích jazycích. V průběhu vývoje v nesourodých časových intervalech jsou aplikace běžně založeny na odlišných programovacích modelech:

- Kombinace monolitických, klient/server a vícevláknových operací,
- Kombinace procedurálního, objektově orientovaného, modelu založeného na komponentách,
- Různé programovací platformy využité během implementace,
- Rozdílné typy databázových systémů – relační, hierarchický a objektový,
- Rozdílné užití middleware řešení pro komunikaci – systém řízený dle zpráv, komunikace s brokerem a volání vzdálených procedury,
- Různé využití pro asynchronní komunikaci – model publish/subscribe, request/reply,
- Rozdílné využití transakcí a bezpečnosti pro middleware
- Rozdílné způsoby sdílení dat,
- Možné využití obecných modelů strukturovaných dat pro komunikaci.

Dalším aspektem pro vyhodnocení integrace je návaznost na již existující realizovanou integraci mezi systémy.

Důvodem pro integraci aplikací z podnikatelského hlediska je maximalizovat výhody informačního systému. Primárním požadavkem je realizace informačního systému jako globálního celku, především z uživatelského hlediska. Samostatně implementované aplikace se zabývají specifickou problematikou, zpravidla využívající odlišný pohled na realizované funkce. Výsledkem popsané implementace je nově implementovaný systém s nepříliš zřejmým uživatelským ovládním bez globálního pohledu.

Rozšíření funkcionality informačního systému je možné realizovat několika způsoby. Na první pohled nejjednodušším způsobem se může jevit nahrazení původní aplikace aplikací nově vyvinutou. Nároky na popsanou aktualizaci jsou extrémně vysoké, proti hovoří zejména extrémně vysoké náklady, časový harmonogram nasazení během dnů z původní na novou aplikaci, migrace stávajících dat a vyškolení uživatelů pro práce s novým systémem. Mnohaletý vývoj v existujících aplikacích nelze v blízké době snadno nahradit bez potřebného časového harmonogramu pro analýzu, vývoj, testování a migraci dat.

Zavedení komerčního finálního ERP řešení vyžaduje také nemalé investice. Každá společnost se v čase vyvíjí jako živý organismus a s tím se mění i požadavky na aplikaci. Všechny faktory se následně projevují v řešení podnikového informačního systému. Uživatelská konfigurace a přizpůsobení komerčního řešení vyžaduje nemalé nároky na zdroje. Požadavky v současné době kladené na ERP systémy pokrývají dle předních konzultačních společností pouze 30% až 40%. Přední IT společnosti např. SAP, Oracle proto zpřístupňují rozhraní k systémům pro uživatelskou konfiguraci.

Požadavky na integraci systémů

Základním stanoveným požadavkem je jednotný vstup dat do systému bez uživatelského opětovného zadání do jiných systému a řešení duplikace dat. Hlavním požadavkem business manažerů je přístup k datům s nízkou latencí a realizace komplexního informačního systému snadno přizpůsobitelného v podnikových procesech.

Integrační proces není založen pouze na technické problematice. Personální obsazení vyčleněné pro integraci systému vyžaduje nároky na vrcholné vedení společnosti včetně senior managementu, business pracovníků až po cílené uživatele pracující s aplikací. Koordinace uvedených členů je poměrně náročná na časový harmonogram a celkovou organizaci. V integračním týmu není podmínkou začlenění IT pracovníků společnosti připravující integraci. Zejména v případech, pokud společnost nevlastní centralizované IT oddělení.

Proces integrace je složen z:

- Definice integrační architektury – zahrnuje identifikaci a plán architektury s využitím stávajících aplikací s minimalizací závislostí mezi aplikacemi. Definice obsahuje plán kompletního nahrazení stávajících aplikací. Výsledkem je definice datového modelu, specifikace otevřených bodů nutných k řešení, definice rozhraní mezi systémy a formáty zpráv,

- Volba integrační architektury a technologií – obsahuje definici konzistentní integrační infrastruktury s využitím interoperabilních technologií. Zohledněny by měly být veškeré aspekty existujících aplikací s prioritou na optimalizaci řešení,
- Realizace integrační dokumentace – dokumentace zahrnuje datový model systémů, výčet rozhraní využitých během integrace, popis komunikace mezi aplikacemi.

Proces integrace je poměrně náročný na časový odhad a na samotnou realizaci. Z výše uvedených kroků proces dále pokračuje specifikací primárních (klíčových) a sekundárních aplikací. Primární aplikace využívají běžně stovky až tisíce uživatelů pro každodenní práci. Sekundární aplikace jsou naopak využívány omezeným počtem uživatelů, velmi časté je využití pouze jedinou osobou pro zvýšení vlastní produktivity bez informování firemního technického IT oddělení. Do uvedené kategorie lze zahrnout kancelářské aplikace bez jakékoliv podpory. Během návrhu je nutné zohlednit obě kategorie aplikací a po vyhodnocení je vhodné zahrnout sekundární aplikace do integrace, protože uživatelé je pro práci potřebují a varianta uchování vlastních řešení na lokálních discích není vhodná. Analýza integrace proto musí obsahovat veškeré požadavky kladené na systém. Po sběru dat následují dva různé přístupy pro integraci:

- Bottom-up Approach – sestavuje integrační systém s identickou funkčností původních implementací. Aplikace jsou rozděleny do detailních funkčních bodů, které jsou spojeny do větších subsystémů až do úplného sestavení systému. Strategií je postupný růst malých celků až do úplného řešení,
- Top-Down Approach – naopak využívá lámání systému z méně konkrétních pohledů do detailnějšího mapování až po získání dostatečného množství informací. Na počátku je obecná formulace systému na detailnější podkategorie až do rozložení na základní prvky.

2.1 Typy integrací

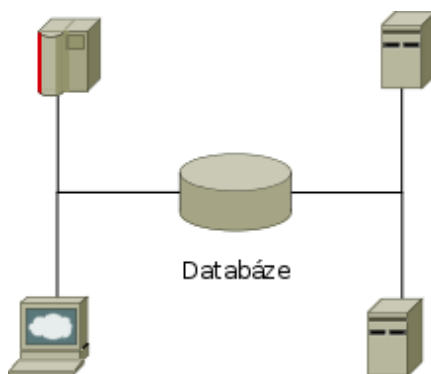
Architektura je zpravidla sestavena do systematicky rozdělených vrstev. Identicky s architekturou ISO OSI v síťovém prostředí je architektura dělena do celků skládajících se z dalších podcelků.

2.1.1 Datová integrace

Technické zajištění datové integrace viz. Obr. 2 je relativně snadná operace, skládající se z snadného přístupu k datům a poměrně velkému počtu dostupných nástrojů pro mapování. Sdílení a publikace určitého výčtu nevyžaduje změny v aplikační logice aplikace.

Problematickým bodem je během sdílení dat zejména význam jednotlivých polí. Každá aplikace má na konkrétní datovou informaci svůj náhled a ten nemusí zcela odpovídat datovým informacím v jiných aplikacích. Důležitým faktorem je přesná specifikace datového modelu během integrační analýzy.

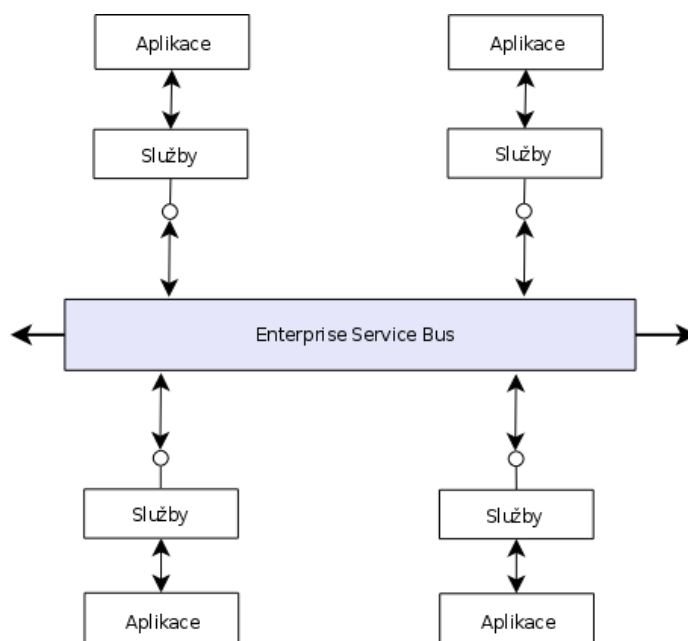
V některých případech je přístup k datům limitován, ať už se jedná o software licencí nebo omezení dodavatelem aplikace. V takových případech se jako vhodná alternativa jeví využití API rozhraní aplikace nebo publikací textových souborů.



Obr. 2: Datová integrace

2.1.2 Aplikační integrace

Aplikační integrace viz Obr. 3 je zaměřena na sdílení funkcionality a business logiky, nejedná se tedy pouze o čistá data jako v datové integraci. Běžně je v integraci využíván API přístup k aplikacím, programovým způsobem lze přistoupit k aplikaci bez využití uživatelského rozhraní.



Obr. 3: Aplikační integrace sběrnicí ESB

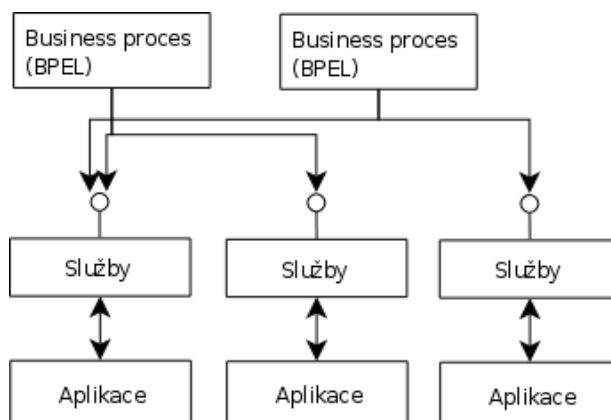
Dříve byl upřednostňován přístup pouze z uživatelské rozhraní, proto většina starších aplikací postrádá publikované API rozhraní a tím přístup k aplikační funkcionalitě.

V poslední době je preferovaná orientace na služby a publikované rozhraní, kdy integrace v takovém případě je lépe proveditelná než v předchozím modelu. Omezením je rozdílnost v přístupu k API rozhraní a využití technologii.

Prioritou pro aplikační integraci je opět stanovení jasných rozhraní pro vzájemnou komunikaci.

2.1.3 Business process integrace

Business process integrace viz. Obr. 4 je orientována na zpracování business procesů v enterprise prostředích, kde je problematika složena z kroků využívajících funkcionalitu dalších aplikací. Zpracování je zaměřeno na abstraktní business metody využívané z aplikačních rozhraní zejména pro integraci systémů. Základem je využití již existující funkcionality aplikací pomocí rozhraní. Uvedená architektura je poměrně novinkou v aplikačních architekturách založena na procesním jazyce BPEL (Business Process Execution Language).



Obr. 4: Business proces (BPEL) integrace

SOA a BPEL poskytují nový přístup k integrovaným informačním systémům, jsou flexibilní a lépe adaptabilní na změny podnikových procesů. Důraz je kladen především na snadnou podporu měnících se požadavků ve podnikovém prostředí.

2.1.4 Prezentační integrace

Prezentační integrace navazuje na dosud vyjmenované integrace s řešením prezentovat informace uživateli v přehledné formě. Presentace výsledků integrovaného systému poskytuje jednotné prezentační vrstvy, pomocí nichž mohou uživatelé přistoupit k funkčnosti integrovaného systému. Ideálním řešením je oddělení prezentační vrstvy od závislosti na aplikační vrstvě.

Rozvoj jednotné prezentační vrstvy skrývá skutečnost, že prezentované výsledky jsou složeny z různých aplikací běžících na pozadí. Do budoucna lze využít výhodu jednotného uživatelského rozhraní k nahrazení existující aplikace, která v současné době poskytuje pouze data a funkční business operace. Integrace se zabývá problematikou definice a implementace uživatelského rozhraní a prezentace business operací integrovaného systému. Poměrně často je využíváno portálové řešení.

2.1.5 Business-to-Business integrace

Integrace pouze na aplikační vrstvě není v mnoha případech dostatečná. S rostoucími potřebami na podnikové řešení jsou podnikové integrace označovány jako business-to-business (B2B). Realizovaná integrace v podnikovém řešení musí primárně poskytnout spolehlivé aktuální informace z informačního systému.

2.2 Integroční technologie

Komplexní řešení integrační infrastruktury vyžaduje ve většině případů využití různých technologií vhodných pro konkrétní situaci. Vybraná technologie musí být nezávislá a založená na otevřených standardech. Technologie používané pro integraci jsou často označovány jako middleware.

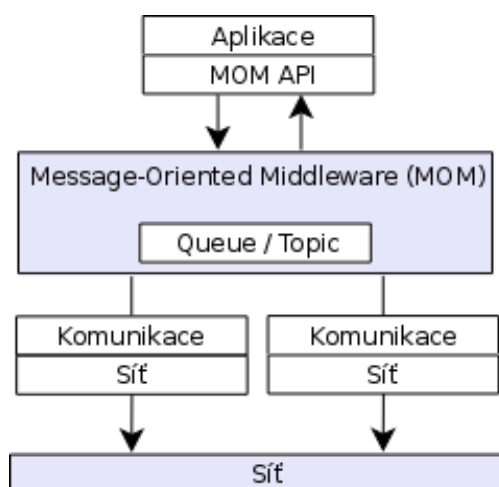
Middleware je software řešení poskytující systémové služby spadající do aplikační vrstvy, vhodný zejména jako základ pro stanovení komunikace mezi systémy do centrální infrastruktury. Využití představuje abstrakci vrstvy architektury s následkem vlivu na výkon integrovaného systému, škálovatelnost, propustnost a další faktory. Uvedené důsledky je potřebné vzít v potaz během paralelních zpracování integrovaných aplikací.

Databázový přístup

Databázové technologie poskytují přístup k databázovým systémům pomocí abstraktní vrstvy bez úpravy kódu klientské aplikace. Identicky lze příkazy definované v abstraktní vrstvě využít k různým databázovým zdrojům. Mezi hlavní výhody jsou obecné operace s databázemi snadno využitelné pro práci s dalšími databázemi. Rozdílnost abstraktního přístupu je v technologii využíváné pro připojení. Mezi neznámější zástupce patří Java Database Connectivity (JDBC) a Java Data Objects (JDO) na platformě Java, dále Open Database Connectivity (ODBC) a Active Data Objects (ADO.NET) pro platformu Microsoft.

Message-Oriented Middleware

Systém orientovaný na zpracování zpráv viz. Obr. 5 klient/server infrastruktury zvyšující interoperabilitu, flexibilitu a přenos aplikací. Uvedené vlastnosti umožňují distribuovanou komunikaci v abstraktní vrstvě middleware. Implementace a využití MOM je založeno na API operacích, které mohou obsahovat jakýkoliv typ dat. Preferovaný způsob integrace je vhodný zejména pro asynchronní zpracování zpráv. Middleware zpravidla poskytuje obecnou frontu, po které probíhá komunikace mezi systémy. Dle zvoleného typu se pro cílového uživatele ukládají zprávy ve frontě pro konkrétní systém. Pokud systém není na příjmu a zpráva je nakonfigurována jako perzistentní, informace není odstraněna a čeká na příjem od cílového systému.



Obr. 5: Integrace zpracování zpráv (MOM)

Nevýhodou je asynchronní zpracování požadavků a tím možná ne-konzistence během zpracování zpráv. Komunikace není řízena formou požadavek – odpověď (request/reply). Využití architektury je závislé na dodavateli middleware software řešení, ověření je nutné zejména pro specifikaci operačního systému, použitých komunikačních protokolů a Java platformy. Java poskytuje vysokou nezávislost na předchozích závislostech k přístupu k middleware produktům pomocí standardizované komunikace Java Messaging Service (JMS).

Remote Procedure Calls

Vzdálené volání procedur Remote Procedure Call (RCP) je využíváno pro synchronní volání vzdálených služeb architektury klient/server. Komunikace stejně jako v případě MOM architektury není závislá na použité platformě. Komunikace je řízena synchronním přístupem, kdy na odeslanou žádost je očekávána odpověď. RPC nezatěžuje v takovém rozsahu síť jak bylo uvedeno u příkladu MOM. Pro komunikaci existuje také asynchronní

varianta, která je využívána spíše výjimečně.

Výhoda vzdáleného volání je zejména v nárocích kladených na serverovou část systému, kde lze mnohem lépe systém škálovat a celkově plánovat zatížení. Klientské části systému tak mohou být pouze tenký klient prezentující například výsledky operací. RPC umožňuje vzdálenou komunikaci bez znalosti síťových adres. Nevýhodou plynoucí ze synchronní komunikace je nutnost vždy dostupného cílového systému. Pokud uvedená podmínka není splněna nastane chybový stav informující o důvodu neuskutečněné operace.

Transaction Processing Monitors

Transaction processing (TP) je důležité zejména pro aplikace s vysokou prioritou v middleware technologiích. TP monitory jsou založeny na koncepci transakcí koordinující běh transakcí v různých zdrojích, dále správu výkonu a bezpečnosti. V případech, kdy je nutné zachovat konzistentní data.

Object Request Brokers

Object Request Brokers (ORBS) je middleware technologií řídící komunikaci mezi distribuovanými objekty a komponentami. ORBS zajišťuje kompaktní interoperabilitu mezi distribuovanými objekty a komponentami bez nutnosti implementovat síťovou komunikaci. Komunikace mezi distribuovanými objekty a komponentami je založena na definici rozhraní. Implementace je odstíněna od klientů pomocí definovaných rozhraní. Podporovány jsou dva typy komunikací, jak synchronní tak asynchronní. ORBS obsahuje lokalizační služby odkazující na prvky uložené v síti.

Idea komunikace je založena na lokálních objektech, které jsou ve skutečnosti volány se vzdálených systémů.

ORBS existují hlavní tři druhy standardů:

- OMG CORBA ORB podpora,
- Java RMI a RMI-IIOP,
- Microsoft COM/DCOM/COM+/.NET Remoting/WCF.

ORB produktů existuje celá řada v souladu se specifikací CORBA ORB. Důležitá je zejména RMI-IIOP využívající stejný protokol pro komunikaci jako CORBA ORB, tím je zaručena interoperabilita s CORBA standardem.

Aplikační servery

Aplikační servery slouží k interakci mezi klientem a prezentační vrstvou. Poskytují sadu služeb v middleware prostředí společně s koncepcí business logiky a komponentami. Podpora služeb je standardně pro technologii webových služeb, ORB a MOM technologií, podporu transakcí, bezpečnosti, load balancing a správu zdrojů. Aplikační servery tak poskytují komplexní řešení pro podnikové informační systémy, vhodné zejména jako základní stavební prvek pro integraci systémů.

Webové služby

Webové služby se řadí mezi nejnovější technologie založené na technologických základech pro poskytnutí interoperability na různých softwarových platformách, operačních systémech a programovacích jazycích. Webové služby jsou tak evolučním krokem v distribuovaných architekturách.

Webové služby jsou první technologií globálně podporovanou všemi hlavními výrobci softwaru. Tím se řadí mezi první technologii opravdu splňující interoperabilitu mezi integrovanými aplikacemi. Základní specifikace webových služeb je založena na protokolu SOAP (Simple Object Access Protocol), WSDL (Web Service Description Language), UDDI (Universal, Description, Discovery and Integration). Všechny uvedené technologie jsou založeny na XML formátu, čímž dále zobecňuje možnost využití pro různé aplikace a programovací jazyky. Významná je především orientace na služby a výměnu dat, čímž se model odlišuje od distribuovaného objektového modelu. Služby lze vyvolat ve více druzích volání:

- jednosměrný,
- požadavek požadavek/odpověď (request/reply),
- vyžádat odpověď,
- pouze notifikace.

Výhodné pro řešení funkcionality webovou službou jsou oddělené služby podporující synchronní a asynchronní volání. Komunikace je založena na standardu internetové komunikace a jedná se obecně o vhodné řešení pro připojení přes firewall a proxy servery. Nevýhodou je poměrně nízký výkon v porovnání s distribuovanou architekturou, která pro komunikaci využívá binární protokol. Další nevýhodou je kvalita služeb (QoS), bezpečnost a transakční zpracování. Z tohoto důvodu byly představeny rozšiřující specifikace webových služeb:

- WS-Security – definuje bezpečnost a ověřování, zabezpečenou komunikaci,
- WS-Coordination – definuje koordinační rámec,
- Transaction – podporuje distribuované transakce pro webové služby,
- WS-Reliable Messaging – poskytuje podporu pro spolehlivou komunikaci a doručování zpráv,
- WS-Addressing – určuje směrování,
- WS-Inspection – poskytuje podporu pro dynamickou introspekci webových služeb,
- WS-Policy – definuje pravidla pro výměnu dat mezi komunikujícími webovými službami,
- WS-Eventing – model podporuje asynchronní volání webových služeb.

Enterprise Service Buses

Enterprise Service Bus (ESB) dále rozšiřuje vlastnosti middleware vrstvy o propracovaný model závislostí, robustnosti, bezpečnosti, řízení a kontrolu služeb a její komunikaci. Nasazením ESB zjednodušuje integraci, komplexnost komunikace podporované v různých druzích technologií EJB, CORBA, webové služby atd. ESB je ideálním řešením pro univerzální komunikační sběrnici mezi zpravidla nekompatibilními protokoly.

Mimo komunikační rozhraní dále poskytuje monitoring a kontrolu nad využitím služeb. Uvedená funkcionality ESB řadí do role prostředníka, který efektivně dokáže směřovat požadavky na jednotlivé rozhraní a odpověď využít pro další zpracování.

ESB mimo monitoring dále obsahuje služby pro logování, profilování, vyvažování zátěže, ladění výkonu a distribuované nasazení.

SOA

Service Oriented Architecture je aktuálně hojně užívaným pojmem ve spojení s integrací systémů. Infrastruktura je založena zejména na komponentách a službách, které lze opětovně využít na základě definice rozhraní. Ideálním řešením je tak adresace a komunikace dle standardizovaných komunikačních protokolů. Uvedený přístup je flexibilní vlastností využívaných dále v integraci a zpracování business procesů.[13][15]

Historicky vznikla SOA architektura ze stále rostoucího nasazení distribuovaných architektur přibližně od roku 1996. Většina implementací distribuovaných systémů byla dříve pevně svázána na dodavatele např. CORBA, DCOM. Následně proto vznikla iniciativa implementace obecného flexibilní architektury s minimalizací závislostí na platformě

implementovaného systému.[15]

Architektura SOA je založena na obecném formátu XML a v případě webových služeb je komunikace řízena pomocí SOAP protokolu. Volba vyjmenovaných technologií byla využita s důvodu nezávislosti na jakékoliv software implementace třetích stran. Simple Object Access Protocol (SOAP) slouží ke komunikaci a výměně dat v technologii webových služeb.[12]

Základ SOAP zprávy je definován ve formátu XML a obsahuje tři elementy:

- obálka (envelope) – informuje jak se má zpráva zpracovat,
- hlavička (header) – obsahuje informace o kódování a případně informace týkající se bezpečnosti,
- tělo (body) – obsahuje informace o strukturovaných datech.

BPEL

Business Process Execution Language je nástroj pro modelování business procesů interagující pomocí rozhraní s dalšími enterprise systémy nebo uživatelskou aktivitou. BPEL proces je jasně ohraničený sled akcí s počátkem v inicializační stavu až po ukončení v závěrečném bodě. Definice procesu je založena na univerzálním formátu XML nebo grafickém návrháři podporovaným v software designeru. Proces je složen z definice procesu, portace služeb nutných pro realizaci nebo odkazu na další BPEL procesy. Běhové prostředí je aplikační server se specifikací Java EE a rozšiřujícího modulu service engine v případě GlassFish řešení.

Hlavní funkcí BPEL je orchestrace webových služeb, jedná se o řízení souhry funkcionalit. Funkce lze dále rozložit do služeb a jednotlivých operací realizující požadovanou funkcionalitu. Výsledný BPEL proces je dostupný jako služba s definovanými vstupními a výstupními parametry a proměnnými. Během procesu tak lze běžně vyhodnotit aktuální datum, realizovat např. logické operace pro získané data z volaných operací. Počátkem BPEL procesu je vyvolání vstupní zprávy s parametry, ukončen zasláním výstupu vyvolaného rozhraní.[26]

JB1

Java Business Integration specifikací pro vytvoření standardizovaného integračního na architektuře SOA. Komponenty jsou propojeny do prostředí JBI a mohou poskytnout nebo konzumovat služby prostřednictvím volně vázaných rozhraní. JBI zajišťuje komunikační platformu.[27]

3 ECM ALFRESCO

Enterprise content management je nejrychleji rostoucím odvětvím v enterprise prostředí. Společnosti zavádějící nebo aktualizující ECM systém se nejčastěji setkávají s řadou komplikací. Během implementace jsou převážně uváděny omezení dodavatelem, vysoké poplatky podpory a problematikou standardizovaných rozhraní. Mezi omezení dodavatelem lze jmenovat výčet podporovaných operačních systémů/aplikací a uzavřený kód systému. Enterprise prostředí vyžaduje vysoké nároky na dostupnost systému. Hlavním důvodem pro řešení havárií je platba podpory systému. Podpora je dále členěna do úrovní podpory od vysoce dostupných systémů po tolerované nedostupnosti vyjádřené převážně v jednotkách hodin. Aplikace a systémy nejsou vždy založeny na standardizovaných řešeních. Během implementace se společnost potýká s problematikou uzavřených standardů, uzavřených formátech či strukturách dat.[4]

Alfresco bylo založeno v roce 2005 John Newton, spoluzakladatel společnosti Documentum, která lídrem na poli komerčních implementací ECM systémů. Spoluzakladatelem Alfresca je dále John Powell, bývalý ředitel společnosti Business Objects. Alfresco se odlišuje v kombinaci open-source a placené podpory systému.

Open-source varianta poskytuje přehled o inovovaných technologiích plánovaných k zavedení v dalších verzích komerční varianty. Vydání nových verzí je přizpůsobeno v častějších časových intervalech. Licence lze zdarma využívat v komerčním prostředí. Open-source varianta je vhodná zejména pro seznámení s novinkami v systému Alfresco tak i pro otestování nastavení systému pro plánovanou migraci na novější verzi. Obecně nelze doporučit pro využití u systémů s kritickou úrovní dostupnosti.

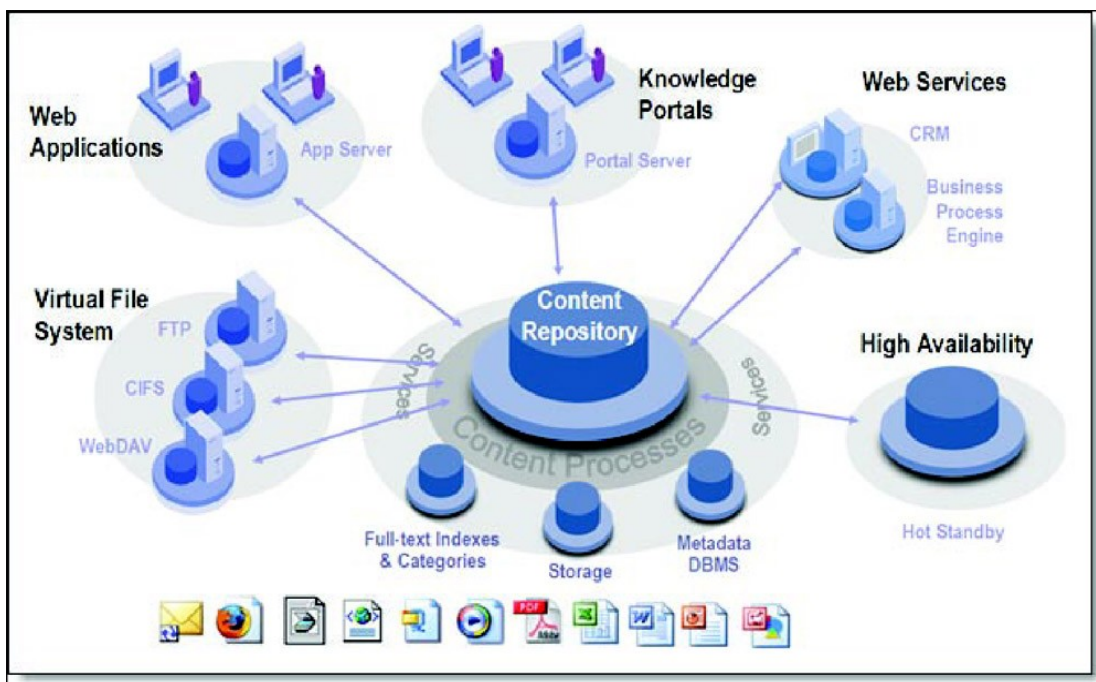
Komerční varianta je naopak vydávána v delších časových intervalech, založená na stabilnější variantě zdrojových kódů. Vydané verze jsou důkladněji otestovány, obsahují performance optimalizace a certifikaci pro uvedené prostředí. Další podmínkou pro enterprise prostředí je dostupnost placené podpory systémů dle různých úrovní. Licence jsou standardně placeny po dobu 2 dvou let dle zvolené podpory. Úrovně zvolené podpory je Silver a Gold. Liší se pouze v době reakce na hlášený případ podpory a tím i ceně licencí. Komerční verzi je možné otestovat po dobu 30 dnů.

Enterprise prostředí využívá vlastnosti systému Alfresco díky své modulární a vysoce škálovatelné architektuře. Systém není platformě závislý a funkcionalita kterou poskytuje plně odpovídá nárokům kladených v enterprise prostředích zpracování dokumentů v médiích, farmaceutickém, zdravotnickém prostředí, kde dochází každoročně exponenciálnímu růstu nároků na zpracování obsahů.

3.0.1 Architektura

Hlavním aspektem ECM systému je aspektově orientována architektura založená na open-source standardech Spring, Hibernate, Lucene, MyFaces, JSR 168, JSR 170 a JSE5.

Alfresco architektura podporuje škálovatelné s vysokým faktorem dostupnosti vyžadována pro kritickou úroveň dostupnosti. Alfresco podporuje model cluster, distribuované cache, replikaci pro více serverů.[4] Schéma architektury je zobrazeno na obrázku Obr. 6.



Obr. 6: Alfresco schéma architektury. Zdroj:[4]

Architektura je založena na ověřených open-source projektech, nasazení není platformě závislé na operačním systému, relační databázi nebo aplikačním serveru. Na klientské stanice nejsou vyžadovány mimořádné nároky, prohlížeče Firefox a Internet Explorer jsou podporovány.

Alfresco nabízí škálovatelnou architekturu s komplexním řešením vyhledávání, strukturovatelných a klasifikovaných informací. Uživatelské rozhraní a dokumentace je dostupná ve většině světových jazyků. Standardní uživatelské dialogy lze dále konfigurovat a upravovat dle funkcionality požadované zákazníkem. Důležitým faktorem pro implementaci ECM je bezpečnost implementovaného systému. Ochrana vůči neautorizovanému přístupu s definicí přístupových práv pro chránění objektů je základním požadavkem v enterprise prostředí. Bezpečnost v systému Alfresco je vymezena aplikací přístupových práv na adresář a objekty, práva je dále možné aplikovat dle rodičovského adresáře na podložky. Definice uživatelských účtů a skupin přístupujících do systému Alfresco je podporována datanázem, LDAP, NTLM, Kerberos a Active Directory.

3.0.2 Repository

System Alfresco je založen na content repository. Repository je hlavní částí systému, kombinací s externími technologiemi jakými je virtuální souborový systém, webové aplikace, portálové řešení nebo webové služby vzniká komplexní škálovatelný systém. Repository uchovává veškeré informace o metadatech a obsazích.

Repository poskytuje výčet služeb využívaných pro úložiště, vyhledávání, sdílení informací a kontrolu nad metadaty a obsahy. Aplikace vyžadující správu obsahů, webových redakčních systémů, ukládání obrázkových informací, vyhledávání a uchovávání rozsáhlého množství dat vyžadují systém document management systém (DMS). Operace podporující DMS systémem je uchování obsahu, který je ve většině implementací uložen na diskové jednotce nebo datovém poli s metadaty. Pro uložení metadat je využívána relační databáze. Dokumenty jsou do systému ukládány v různými způsoby. Ve většině případů se jedná o rozhraní s různými systémy využívající technologii např. Webových služeb, sdíleních síťových jednotek.

DMS systém podporuje pro obsah základní operace:

- uložení obsahu – základní operace uchovávající informace o metadatech a obsahu v systému,
- import – vstupní operace pro uložení obsahu do systému,
- export – export informací pro externí zpracování obsahu,
- klasifikace zdrojů – obsahy lze rozčlenit do kategorií a skupin,
- bezpečnost obsahů – na obsahy lze aplikovat bezpečností práva,
- check-in – operace zařazující nový obsah k objektu. Operace check-in odstraní zámek na dokumentu,
- checkout-out operace – operace vyřazení zajistí zámek nad dokumentům, ostatní uživatelé mohou číst jen poslední verzi,
- verze – operace podporuje vývoj dokumentu rozčlenit do verzí nesoucí informací o stavech dokumentu,
- content query – jazyk vycházející z lucene query language podporuje vyhledávání dle metadat a indexovaných obsazích.

3.0.3 Automatizace business procesů

Optimalizace a automatizace business procesů ve společnosti zvyšuje produktivitu práce, snížení nákladů a přínos pro přehledné a snadno dohledat informace. Řešení je zapracováno do automatizovaných procesů zpracovaných v cyklech s integrací systémů třetích stran. Alfresco pro zpracování business procesů využívá implementaci JBoss Business Process Manager (JBPM). Implementace workflow v prostředí JBPM řídí procesy s využitím komponent systému, lifecycle, auditování operací.

3.0.4 Integrace s externími systémy

Aplikace v dnešní době již nejsou omezeny heterogenním prostředím, a proto v dnešní době rozšířeného internetového připojení a rozsáhlých sítí se stále rozvíjejí komunikační technologie. V současné době se jedná o technologie SOA, MQ series messaging, EJB, webové služby.

Alfresco poskytuje rozhraní pro integraci pomocí technologie webových služeb a Java Content Repository Application Programming Intergace (JCR API). Alfresco je konfigurovatelné s Kofax Ascent Capture a nabízí tak řešení transformace papírových dokumentů do elektronické formy, včetně automatické klasifikace, extrakce a validace dat. Kofax nabízí řešení lokálního skeneru nebo centralizovaného řešení. Dalším faktorem pro publikaci a zpracování dat je portálové řešení, Alfresco zahrnuje podporu pro integraci standardně v instalovaném systému.

Alfresco lze integrovat s externími systémy jakými jsou LDAP a Active Directory a podporuje centralizované řešení bezpečnosti single sign-on.

3.1 Vlastnosti ECM systému

3.1.1 Adresářová struktura

Adresářová struktura v ECM systému Alfresco využívá velkou část ideologie při práci se soubory a složkami z operačních systémů. Operace nad objekty jsou rozšířeny o pokročilé operace indexování, tvorbu odkazu atd. Mimo objekty typu složka a soubor je definován pojem prostor a kabinet. Prostor uchovává informace o specifické skupině dat.

Podporované druhy prostorů:

- Space Security – definuje uživatelská oprávnění, které lze realizovat oprávněným uživatelem.

- Space Business Rules – jedná se o akce vyvolané nad obsahem přiřazené na úrovni prostoru.
- Space Workflow – dokumenty spravovány ve Workflow lze rozčlenit na úroveň prostoru. V praxi se jedná o definici prostoru pro dokumenty ve stavu Draft, Review atd.
- Space Events – definuje události např. nově příchozího dokumentu a následně na událostí lze vyvolat další akci.
- Space Aspects – aspekty rozšiřují vlastnosti objektu, identicky s Space Workflow typem prostoru lze členit např. uložení dokumentu na základě vyvolaného aspektu.
- Space Search – umístění nebo cesta pro vyhledávání.
- Space Content – objekt lze verzovat, uzamknout a realizovat další operace.

Kabinet označuje kořenové složky, které se dále větví do podsložek tvořící hierarchickou strukturu. Složka mimo názvu dále obsahuje informaci vlastníkovu složky, nastavených oprávněních a obsahu. Adresářová struktura je tedy tvořena složkou a souborem. Pokročilé vlastnosti DMS dále podporují operace kopírovat, přesunout či odkaz na daný soubor/adresář. Uvedené vlastnosti jsou ve většině případů využity pro běh Workflow procesů.

Z pohledu hierarchie adresářové struktury je návrh a finální podoba stěžejní částí implementovaného DMS systému. Kořenová složka jak již bylo uvedeno obsahuje podsložky které se větví v stromové struktuře. Hloubka stromové struktury není limitována, vhodné je přihlídnout během návrhu k optimálnímu vyvážení složek. Příliš složitá struktura komplikuje orientaci uživatelů, naopak méně složek obsahující velký počet souborů způsobí neuspokojivé uživatelské odezvy systému. Ideální je kombinace dle tématických typů objektů s využitím dědičnosti pro nastavení přístupových práv. Obrazovka adresářové struktury elektronických knih je v příloze Obr. 16.

Vlastnosti složky

Na dialogu je zobrazena organizace vlastností složky, horní část dialogu obsahuje kompletní cestu k vybranému adresáři. Název složky je uveden pod absolutní cestou, pravá strana obsahuje akční komponenty pro definici:

- Content rule – zobrazuje počet content rule definici pro označený adresář,
- Add Content – volba vložit soubor odkazující se na dialog pro import souboru,
- Create – podmenu obsahuje volbu pro vytvoření podsložky,

- More Actions – obsahuje výčet akcí,
 - View details – zobrazí detailní informace objektu,
 - Delete – spustí odstranění objektu,
 - Cut – vloží soubor do schránky pro přesun do jiného umístění,
 - Copy – vloží objekt do schránky,
 - Paste All – vloží veškeré objekty ze schránky,
 - Import – spustí importní dialog,
 - Manage Space Users – nastavení přístupových práv,
 - Manage Content Rules – nastavení content rule pravidel,
- Icon view – mění pohled zobrazených ikon,
- Browse space – seznam podsložek,
- Content Items – seznam obsahu pro složku.

Pro složky lze definovat pravidla tzv. Content rule zajišťující funkce DMS systému na základě události. Výběrem typu objektu, lze specifikovat přesný název typu nebo využít pravidel pro všechny příchozí objekty. Nastavením typu přechází dialog k výběru předem definované akce, z výčtu uvedeme například start Workflow procesu, operace checkou/check-in, spuštění custom Websript(implementace Alfresco API v javascript jazyce), odeslání emailu nebo spuštění transformace obsahu.

Add content z názvu odpovídá importnímu dialogu pro vložení souboru do ECM systému. V náváznosti na vytváření obsahu funkce Create space vytváří adresář.

Funkce view details zobrazuje detailní popis objektu rozšíření o seznam spuštěných Workflow nad objektem, přiřazené kategorii nebo content rule pravidla. Nastavení oprávnění lze definovat pomocí akce Manage Space Users, pro objekt je zobrazen seznam oprávnění včetně nastavení dědičnosti pro složky.

Náhled na obrazovku vlastnosti složky ve webové aplikaci Alfresco je dostupný v příloze Obr. 17.

Vlastnosti souborů

Vlastnosti souborů jsou zobrazeny v identickém dialogu stejného pro složky, většina funkcí je stejného významu jako již bylo uvedeno u vlastností složky. Obsah dokumentu je zaměřen odlišnou funkcí na rozdíl od složky, proto jsou uvedeny operace checkou, check-

in, download, podpora vícejazyčného obsahu, diskuze nebo spuštění Workflow.

Náhled na obrazovku vlastnosti složky ve webové aplikaci Alfresco je dostupný v příloze Obr. 18.

3.1.2 Definice Content Model

Problematika Content Model vlastností v ECM systému obsahuje poměrně širokou škálu nastavení, které je poměrně často využíváno pro implementaci požadavků zákazníka. Content Model v systému Alfresco definuje mimo standardní strukturu systému vlastnost vytvoření vlastních typů, implementace aspektu, definice business pravidel, definice relací pro objekty, nastavení šablon pro objekty a audit změn provedených na objektu.

Definice typů

Systém obsahuje definici standardních atributů využívaných pro uložení vlastností dokumentu. Mimo název, titulek a vlastnosti obsahují atributy datum, vlastník dokumentu atd. Rozšířit vlastnosti typu dokumentu lze vytvořením nového uživatelského typu (vlastní Content Model konfigurace) nebo přiřazením vlastního aspektu.

Konfigurace vlastního modelu vyžaduje vytvoření vlastního konfiguračního souboru, referenční nastavení systému obsahuje konfigurace typů knihovny a lze dohledat v adresáři Alfresco serveru:

```
/opt/alfresco/tomcat/shared/classes/alfresco/extension/customModel.xml
```

Typ je vytvořen jedinečným namespace, který určuje identifikaci a dále seznam políček pro typ. Mimo seznam lze využít dědičnost a rozšířit tak již definované vlastní nebo systémové Alfresco typy.

Konfigurace typu se musí registrovat v seznamu Content Model, uložený v cestě:

```
/opt/alfresco/tomcat/shared/classes/alfresco/extension/custom-model-context.xml
```

Definice aspektu

Aspekt dále rozšiřuje nastavení vlastností stejně jako v případě rozšíření definice typu. Na rozdíl od fixního nastavení v případě definice typu, aspekt je dynamicky přiřazená vlastnost objektu. Rozšířené vlastnosti lze v případě nutnosti odebrat. Implementace aspektů je flexibilním řešením pro rozšíření definice objektů, požadované zákazníkem. Na aspekty lze uplatnit závislosti, definující závislost na vybrané aspekty.

Konfigurace aspektu je stejná s definicí uživatelského typu.

Vlastnosti typu a aspektu

Vlastnosti nastavitelné pro typ a aspekt jsou shodné pro uvedené komponenty, uvedené v následující tabulce 1.

Vlastnost atributu	Popis
text	proměnná řetězec
content	binární dokument
int	celočíslná hodnota
long	celočíslná hodnota zaměřena na rozsáhlé číselné údaje
float	reálná hodnota
date	datum
datetime	datum proměnná včetně údajů o čase
boolean	logická hodnota pravda, nepravda
category	odkaz na typ kategorie
path	URL cesta

Tabulka 1: Definice typu atributů

Publikace Content Model dat

Uživatelsky definované typy a aspekty je nutné publikovat v konfiguraci webové aplikace. Bez uvedené změny a restartu systému uživatelské typy a aspekty existují, ale webové rozhraní neposkytuje ovládací prvky k nastavení hodnot. Konfigurací uživatelského rozhraní se pro typ dialogu a typ/aspekt objektů nastavuje viditelnost položek.

Definice relací

Objekty uložené v repository lze svázat relací, rodičovský objekt ve vlastnostech obsahuje informaci o svázaných objektech. Příkladem z praxe je například hlavní dokumentační dokument svázaný např. s technickými výkresy.

Typ relace lze rozlišit na:

- reference association – obsahuje pouze referenci na objekt,
- child association – rodičovský objekt je svázán s dalšími objekty.

Relace lze stejně jako datový model dále konfigurovat a vytvářet vlastní uživatelské relace, důvodem je poměrně časté projení objektů, pro příklad schvalování dokumentace, jazykových překladů.

Audit systému

Objekty včetně obsahů jsou v ECM systému nejdůležitější částí, uchovávající informace o vypracované problematice a vývoji dokumentu uložený v revizích. Změny provedené na objektech je možné auditovat v systému a uložit tak informaci, který uživatel s konkrétní verzí pracoval, a které operace s objektem realizoval.

V instalaci Alfresco je auditování systému vypnuto, konfigurací souboru:

```
/opt/alfresco/tomcat/webapps/alfresco/WEB-INF/classes/alfresco/auditConfig.xml
```

Lze detailně nastavit auditování, interní audit a ovlivnit mód uchovávání informací. Audit je rozdělen do služeb např. FileFolderService (změny ve složce), NodeService (prostory, správa aspektů, vlastnosti objektů) a dalších.

Report audit změn je možné dohledat v webové aplikaci Alfresco, označením složky a výběrem volby Preview in Template a označením šablony show_audit.ftl.

3.1.3 Bezpečnost

Model bezpečnosti v systému Alfresco je realizován konfigurovatelným frameworkem. Lze využít vestavěné konfigurace modelů zabezpečení nebo realizovat pomocí externího systému jakými jsou např. LDAP nebo Active Directory. Model zabezpečení je vysoce škálovatelný pro rozsáhlý počet uživatelů a skupin. ECM systém požaduje pro aplikování bezpečností definicí uživatelů do skupin a rolí, definicí přístupu k objektům, omezení přístupu nejen k administrátorským funkcím. Nelze omezit základní vlastnosti a funkcionalitu uživatelů, možnost sdílení informací a práci se systémem.

Uživatelé a skupiny

Uživatelé jsou individuálními členy skupin identifikováni jednoznačným uživatelským ID. Pouze administrátor je definován jako super user s rozšířenými právy konfigurace a přístupu. Uživatelé nepřihlášení do systému Alfresco jsou identifikováni jako guest s omezeným přístupem k obsahu content repository. Alfresco uživatelské skupiny jsou výčtem uživatelských členství ve skupinách. Skupiny jsou dále dělitelné dle významu (systémový administrátor) a hierarchie do pod skupin.

Přístupová práva a role

Práva definují přístupové práva k adresářové struktuře a obsahům, Alfresco podporuje základní úroveň nastavení. Práva lze možné rozšířit dle konfigurace uvedené v kapitole 3.1.3. Práva jsou uložena v řetězcové hodnotě, jednotlivé práva jsou povoleny nebo omeze-

ny uživateli, skupině. Potomky objektů v adresáři mohou přejímat nastavení od rodičovských objektů. Standardní instalace obsahuje přednastavenou vlastnost přejímání souborů od adresářů. Konfigurací lze ovlivnit nastavení na adresářích či kabinetech.

Každá z definice práv je aplikovatelná na uzel, kabinet, vlastnosti kabinetů, pod-kabinety, obsahy, vlastnosti obsahu a business pravidel.

Skupiny práv jsou:

- číst,
- editovat,
- vložit,
- odstranit.

Role je kolekce práv přiřazených každému uživateli v systému. Každá role zahrnuje výčet přístupových práv. Role dělíme do skupin:

- Consumer – právo číst objekt,
- Editor – právo číst a editovat objekt,
- Contributor – právo číst a vložit objekt,
- Collaborator – právo číst, editovat a vložit objekt,
- Coordinator – právo číst, editovat, přidat a odstranit obsah – definice plného přístupu.

Role a přístupová práva lze dále rozšiřovat na základě požadavků.

Autentifikace

Alfresco pro autentifikaci uživatelů využívá kombinaci uživatelského jména a hesla. Autentifikace je podporována následující částí systému:

- webová aplikace,
- CIFS,
- FTP,
- WebDAV,
- Webové služby,
- Spring bean dostupné jako public služby v jazyce Java.

Klient realizující požadavek na repository pomocí veřejné služby API musí být nejdříve autentifikován. Proces autentifikace je ve standardní instalaci proveden ověřením kombinací uživatelského jména a hesla. Další režimy jsou realizovány na základě použitého modelu ověřování uživatelů single sign-on, LDAP nebo Active Directory.

Rozšířený model práv a skupin

Alfresco v podstatě podporuje standardní režim přednastavených práv s možností bezpečnostní politiku dále rozšířit dle požadavků zákazníka. Standardní sadu zabezpečení je možné konfigurovat dle XML souborů nebo implementací Java třídy.

Definice práv je uvedena v konfiguračním souboru:

```
/opt/alfresco/tomcat/webapps/alfresco/WEB-INF/classes/alfresco/model/permissionDefinitions.xml
```

Práva podporovaná pro objekty kabinetu a adresáře uvádí tabulka 2:

Název práva	Popis
ReadProperties	Právo číst vlastnosti objektu
ReadChildren	Právo číst obsah objektu v adresáři
WriteProperties	Právo aktualizující vlastnosti objektu
DeleteNode	Právo odstranění kabinetu
DeleteChildren	Právo odstranění obsahu a adresáře
CreateChildren	Právo vytváření adresáře

Tabulka 2: Definice práv pro kabinetu a adresáře

Práva podporována pro objekty souboru uvádí tabulka 3:

Název práva	Popis
ReadContent	Právo číst soubor
WriteContent	Právo zápisu obsahu souboru
ReadProperties	Právo číst vlastnosti objektu
WriteProperties	Právo zápisu vlastností objektu
DeleteNode	Právo odstranění souboru
ExecuteContent	Právo spuštění souboru
SetOwner	Právo změny vlastníka souboru

Tabulka 3: Definice práv pro soubory

Definice rolí

Definice role (viz tabulka 4) je kolekcí práv přiřazených uživateli v konkrétním prostoru úložiště. Od děděné podprostory mohou převzít data z nadřazeného složek úložiště. Role mohou být aplikovány na jednotlivé položky, složky či prostory v systému Alfresco.

Role	Popis bezpečnostních práv
Consumer	Právo číst objekty typu adresář a obsah
Editor	Oprávnění Consumer a editace existujícího obsahu
Contributor	Oprávnění Editor a právo vložit nový obsah
Collaborator	Kombinace oprávnění Editor a Contributor
Coordinator	Plné oprávnění

Tabulka 4: Definice rolí

Struktura a vlastnost rolí je konfigurovatelná pomocí konfigurace. Nově definované role dále rozšiřují vlastnosti dle klientských požadavků. Konfigurační nástroj je uložen v cestě:

```
/opt/alfresco/tomcat/webapps/alfresco/WEB-INF/classes/alfresco/model/permissionDefinitions.xml
```

```
<permissionGroup name="TestRole" allowFullControl="false"
expose="true">
  <includePermissionGroup permissionGroup="Read" type="sys:base" />
  <includePermissionGroup permissionGroup="AddChildren"
type="sys:base"/>
  <includePermissionGroup type="cm:lockable" permissionGroup="CheckOut"/>
</permissionGroup>
```

3.1.4 Funkce Alfresco ECM systému

Práce s dokumenty a organizace složek usnadní uživatelům práci, přidanou hodnotou ECM systému je bezpečnost a centrální řešení. Uchovávání veškeré dokumentace a disku lokálních stanic a výměnu velkých souborů emailem nelze považovat za v dnešní době ideální řešení. Uživatel je ve většině případů nucen k uchování různých verzí pro dohledání změn nebo připravit dokument v odlišných jazycích. Uvedené případy lze řešit efektivně v ECM systému Alfresco.

V popisu funkcí je vybrán souhrn prioritních funkcí s uvedeným příkladem v praxi.

Vytvoření obsahu

Obsahy lze vytvářet ve dvou odlišných variantách. V prvním případě se jedná o editaci v zabudovaném editoru textových souborů, editovatelné jsou formáty HTML, Text a XML. Naopak binární soubory lze uložit dle importní funkce, nejsou omezeny další funkce systému. Alfresco vestavěný editor pro soubory typu MS Office, obrázky, sken. Pro otevření a práci s binárními soubory lze využít externí aplikace asociované na úrovni operačního systému.

Editace vlastností dokumentu

Každý objekt v repository musí obsahovat informaci o typu, jelikož právě od něj se dále odvíjí seznam atributů. Typ objektu je poměrně důležitou informací, na typ dokumentu mohou být navázány další spustitelné akce, workflow atd. Seznam atributů je označován jako content metadata, jedná se o strukturovaná data, která popisují charakteristiku typu objektu.

Vlastnosti dokumentu lze v Alfresco systému načítat pomocí komponenty metadata extractor. Uvedené komponenta dle typu souboru načte z binárního souboru vlastnosti dokumentu. Podporované typy formátů jsou Microsoft Office, média soubory a PDF dokumenty.

Verzování

Verzování uchovává oddělené informace o objektu rozdělené do odlišných verzí objektu. Každá verze obsahuje jednoznačné reálné číslo, poslední verze je automaticky označena. Verze objektu dělíme na major a minor verzi. Označení major verze je využíváno pro změnu primárního čísla verze, v případě zařazení major vzniká z verze 1.1 verze 2.0. Využití je vhodné zejména pro zapracování nebo fixaci dokumentu. Minor verze naopak vzniká během vývoje dokumentu, označující nepříliš zásadní změny pro význam dokumentu. Verzování není přednastaveno pro nově vytvořené objekty. Povolení je možné v detailním dialogu kategorie „Version History“ nebo konfigurací na typu dokumentu.

Operace check-out, check-in a undo check-out

Využití operace verzování zpřehledňuje práci na dokumentech a zachovává změny vytvořené během editace. Pro operace zařazení je implementována funkce check-in, kde dochází k přenosu dat z klientského počítače do repository. Během operace check-in lze nastavit volbu zařazení (verze bude uložena v major nebo minor verzi) a další vlastnosti. Naopak operace vyřazení check-out, vytváří zámek nad objektem, pracovní verze označená „Working Copy“ je dostupná pouze uživateli, který provedl check-out operaci. Oprávnění uživatelé mohou přistoupit k poslední uložené verzi, není povoleno znovu vyvolat check-out operaci na již vyřazeném dokumentu.

Zrušení uzamčeného objektu je možné realizovat funkcí „Undo check-out“, pracovní verze zaniká, zruší se změny provedené nad dokumentem a v repository je dostupná poslední verze dokumentu před vyvoláním operace check-out.

Kategorie obsahu

Třídění kategorií klasifikuje objekty dle pravidel do hierarchické, taxonomické nebo ontologické struktury. Klasifikace tak určují rozdělení objektů do skupin, které lze snadněji rozčlenit a vyhledávat.

Kategorizace probíhá správou skupin kategorií, a tato role je povolena pouze uživateli v roli administrátor. Kategorie lze dále členit do pod kategorií, počet úrovní není omezen.

Obnovení dat

Alfresco v případě odstranění složky nebo souboru neodstraní fyzické data, ale přesune pouze odstraněná data do odlišného dočasného úložiště. Data lze pomocí administrátorského zásahu zpětně obnovit, konfigurace je nastavena k neodstranění smazaných dat. Alfresco nabízí konfigurovatelnou volbu periodického odstranění souborů starších n-dnů.

Sít'ová jednotka

Standardní instalace poskytuje uživatelům rozhraní webové aplikace pro připojení do repository. Mimo webovou aplikaci lze pro připojení k repository využít protokoly FTP, WebDAV a CIFS.

CIFS

CIFS je rozšíření protokolu Microsoft Server Message Block, jedná se o standardizovaný protokol pro komunikaci a sdílení dat v počítačových sítích a Internetu. CIFS poskytuje sadu základních operací open(otevřít), close(uzavřít), read(číst), write(zápis) a seek(vyhledání). Sdílení dat obsahuje ochranu konfliktů proti vzájemnému přístupu, protože protokol podporuje přístup více klientů. CIFS server podporuje anonymní a zabezpečený přístup k souborům.

FTP

FTP protokol je výhodný pro využití přenosu velkého objemu dat mezi lokálním klientem a vzdáleným serverem. Alfresco server podporuje připojení FTP včetně zachování nastavení bezpečnostních oprávnění a verzování.

WebDAV

WebDAV je primárně určen pro editaci a správu souborů na vzdáleném serveru. Rozhraní poskytuje pokročilé operace se soubory umístěnými na vzdáleném serveru. Mimo přímou adresaci vzdáleného souboru a získání tak odkazu na vzdálený server rozhraní WebDAV dále poskytuje upload a další varianty.

WebDAV klient je dostupný na adrese: <http://localhost:8080/alfresco/webdav>

Data dictionary

Data dictionary je interní složkou obsahující sdílené zdroje, převážně:

- emailové šablony – obsahuje šablony pro notifikaci vytvořené v FreeMarker jazyku,
- šablony prezentace - vytvořené v FreeMarker jazyku nebo formátu flt,
- RSS šablony – šablony RSS,
- uložené definice vyhledávání – nastavení vyhledávání lze uložit do definice, postup je výhodný zejména pro často opakované vyhledávání,
- skripty – skripty definovány uživateli,
- šablony adresářů – strukturu existujících adresářů lze uložit do šablon.

3.1.5 Implementace business procesů

Implementace business procesů využívá veškeré uvedené informace o adresářové struktuře, uživatelských, vlastnostech typů a objektů a ECM operacích. Během vývoje je na každou uvedenou část nahlíženo jako na komponentu, která je využitelná během vývoje a nastavení business pravidel. Během implementace lze využít seznam dostupných služeb pro nastavení konkrétního požadavku. Akce využitelné pro nastavení procesů lze využít implementací vlastních pravidel.

Alfresco framework využívá pro implementaci business pravidel technologii aspektově orientovaného programování, které je výhodné zejména pro dynamicky aplikované akce bez nutnosti změny původního objektu. Na objekt lze aplikovat aspekty rozšiřující vlastnosti objektu bez nutnosti měnit jakkoliv již realizovaný kód nebo konfiguraci.

Alfresco během události vložení, odstranění, aktualizaci objektu vždy kontroluje, zda pro objekt neexistuje business pravidlo. Pokud pravidlo je definováno, nastává kontrola podmínek dle definice. V případě úspěšného splnění podmínek je vyvolána akce, naopak v případě nesplnění podmínky je operace dokončena na pozadí systému.

Seznam podporovaných akcí pro realizaci business pravidel:

- Aplikovat aspekt objektu – aplikuje na objekt sadu nových vlastností/funkcí,
- Spustit workflow nad objektem – spustí proces schvalování nad objektem,
- Operace check-in nad objektem – pro objekt je vyvolána operace check-in,

- Operace kopírování od odlišného umístění – objekt je mimo současnou složku zkopírován do odlišné cílové složky,
- Spuštění skriptu – spuštění uživatelské JavaScript akce,
- Získání metadat z objektu – specifické formáty souborů obsahují metadat, které lze pomocí loaderu přenést na metadata objektu v Alfresco repository,
- Import Alfresco obsahu – import objektů do repository,
- Připojení objektu do kategorie – přiřazení objektu specifickou existující kategorií,
- Přesun objektu do odlišného umístění – objekt je odstraněn ze současné složky a přesunut od cílového adresáře,
- Odstranění aspektu objektu – odstraní objektu vlastnosti definované aspektem,
- Zaslání emailové notifikace – akce odešle emailovou notifikaci vybraným uživatelům nebo skupinám,
- Nastavení typu objektu – definuje objektu vlastností dle typu,
- Transformace a zkopírování objektu do specifické složky – transformuje objekt do vybraného formátu, výsledný objekt je přesunut do odlišné složky,
- Transformace a zkopírování obrázku do specifické složky – transformuje obrázek do vybraného formátu a upraví např. vlastnosti obrázku do specifické složky.

Business pravidla lze nastavit na složku vyvolané událostí nebo další variantou je vyvolání akce na základě uživatelského požadavku a naplánované akce na pozadí. Volba „Run Action“ je dostupná v menu „View Details“, pokud uživatel má nastaveno oprávnění pro spuštění akce.

Akce vyvolané v pravidelně plánovaném časovém okamžiku obsahují definici cron výrazu definovaný na úrovni operačního systému. Cron výraz obsahuje nastavení času a intervalu vyvolání akce. Následuje dotaz v repository označující objekty, na které bude aplikována akce, která na každém označeném objektu provede požadovanou funkci.

Definice plánovaných akcí je uložena v ukázkovém souboru:

```
/opt/alfresco/tomcat/webapps/alfresco/WEB-INF/classes/alfresco/config/alfresco/extension/scheduled-job-context.xml.sample
```

3.1.6 Workflow

Workflow označuje automatizovaný proces, kdy ve většině případů se jedná o sousledné kroky zpracované automaticky na úrovni systému nebo manuální uživatelskou aktivitou. Přední dodavatelé Enterprise Content Management systému poskytují nástroje pro zpracování workflow požadavků. Pro menší organizace postačí jednokrokové zpracování vztahu editor schvalovatel, kdy editor zpracuje elektronickou dokumentaci a schvalovatel se vyjádří k obsahu zpracovaného dokumentu. Schválením dokument přechází do jiného umístění v systému a proces je ukončen. Neschválením dokument je naopak navrácen nebo delegován ke zpětnému přepracování či doplnění. Workflow je konfigurovatelný nástroj pro realizaci business procesů zajišťující přehled o realizovaných úkolech a dokumentaci.

Alfresco podporuje simple workflow orientované na zpracování obsahu na rozdíl od advance workflow řešící zpracování úkolů. Simple workflow je vhodné využít pro zpracování dokumentace schvalované v jednokrokovém scénáři. Běžně se tak dokument posouvá v adresářové struktuře uživatelům v organizační struktuře. Realizace je řešená v případě ukončení úkolu vytvořením další instance workflow šablony.

Advance workflow je orientováno na úkoly, kde vytvořený úkol je svázán s dokumentem k provedení revize, schválení. Nastavení uživatelé zpracovávající úkoly jsou přiřazeni k úkolu automaticky nebo lze pro konkrétní případ nastavit předem definovanou skupinu.

Poslední skupinou je implementace custom workflow, což je nástroj pro zpracování vycházející z JBPM frameworku. Konfigurací workflow šablony lze vytvořit uživatelsky sled kroků složený opět z automatických a uživatelsky definovaných aktivit.

Simple workflow

Simple workflow je určeno zejména pro proces schvalování. Před spuštěním procesu je vytvořena adresářová struktura rozdělená na organizační jednotky a podsložky Review, Approve, Reject. Workflow je spuštěno na objektu a dle nastavení řídí cestu dokumentu v adresářové struktuře, která ještě před spuštěním workflow obsahuje nastavení oprávnění. Výsledkem je zabezpečený obsah dostupný editorovi nebo uživateli ve skupině administrátor (uživatelé z nadřazených jednotek mohou pouze číst pro verifikaci změn, popř. doplnit potřebné informace). Simple workflow je v systému nakonfigurováno pro jednokrokové schvalování s možností delegovat úkol na další uživatele v repository. Výsledkem je schválení či zamítnutí vytvořeného úkolu.

Advance workflow

Pokročilé vlastnosti workflow je vhodné implementovat v obsahově orientovaných procesech. Vlastnosti limitující nasazení jsou následující:

- nepodporuje více stavové workflow,
- omezeno pouze na dva výstupní stavy, schváleno a zamítnuto,
- nepodporuje definici paralelního workflow,
- závislost na adresářové struktuře pro více stavové workflow a trigger pravidel.

Alfresco pro práci s workflow rozšiřuje vlastnosti frameworku JBPM (JBoss Business Process Management) a začlenilo rozšířenou implementaci JBPM do jádra systému díky kterému plně podporuje komplexní nástroj na úkolově orientované procesy.

JBoss jBPM je flexibilní, rozšiřitelný workflow management s intuitivním procesní definicí s podporou grafického návrhu během implementace, triggeru, čekající komponenty, synchronní komunikací, časovači a automatizovanými akcemi.

Advance workflow je orientováno na úkoly, kde lze vytvořit uživatelský úkol, přiřadit dokument a spustit process workflow. V systému jsou nakonfigurovány dva druhy šablon:

- Adhoc Task workflow – úkoly přiřazené „ad hoc“, zjednodušená varianta využívána především pro vyjádření k dokumentu,
- Review and Approve workflow – využíváno pro schvalování a posouzení obsahu.

Custom workflow

Uživatelsky definované workflow je realizováno v případě nepokrytí vlastnosti workflow šablonami instalovanými v systému. Vlastní konfigurací lze implementovat specifické požadavky zákazníka. Šablona workflow je tvořena XML dokumentem, manuální editací nebo grafickým nástrojem JBoss jBPM generující finální šablonu. XML dokument popisuje vztah manuálních a automatických úkolů vzhledem k propojeným krokům. Mimo úkoly lze průchod workflow dále členit. Validní XML workflow šablona je nasazena do systému a nabízena iniciátorovi workflow.

3.1.7 Vyhledávání

Úspěšná implementace ECM systému z velké části závisí na schopnosti vyhledat objekty na základě vlastností či fulltext obsahu. Alfresco podporuje fulltext vyhledávání využitím ověřeného vyhledávacího frameworku Lucene <http://lucene.apache.org/>. Lucene je open-source vysoce škálovatelný framework s efektivním search engine. Dotaz na vyhledání je

podle konfigurace ověřen na metadata objektu a obsahu binárního souboru. Lucene podporované formáty indexuje, mezi podporované formáty se řadí PDF, HTML, Microsoft Office dokumenty a další.

Indexované informace jsou uloženy na souborovém systému v cestě:

```
/opt/alfresco/alf_data/lucene-indexes
```

Interní index dat na vyhledání je složen z Lucene search query nebo Xpath výrazu. Pro implementaci systému byly využity ve vlastní práci vlastnosti Lucene query, který nabízí široké možnosti pro vyhledání objektů.

Vyhledávací funkce ovlivňuje rozsah dat uložených v repository, nad kterými probíhá vyhledávání a podporované formáty v případě fulltext přes obsah objektů. Nastavením limitů lze ovlivnit výsledky zobrazené uživateli. V první variantě lze nastavit stránkování výsledků, čímž se vykreslování rozsáhlé sady omezí. Druhá varianta konfigurací minimálního počtu znaků hledaného řetězce, maximální velikost datové sady navrácené uživateli ve webové aplikaci.

Vyhledávací funkce lze rozdělit:

- simple search – slouží v hlavní obrazovce pro rychlou volbu vyhledávání, nelze detailněji specifikovat vlastnosti hledaného objektu,
- advance search – od předchozí varianty nabízí naopak detailní vlastnosti v podobě typů objektu, specifikaci místa a kategorií vyhledávání. Advance search dále podporuje konfiguraci uživatelských polí, požadované atributy lze nastavit v uživatelské konfiguraci,
- save reports – definice nastavení lze uložit a využít pro periodicky se opakující dotazy např. expirované dokumenty v úložišti.

Reporty

Pro reporting lze využít vlastností Advance search obrazovky, nastavení hodnot a parametrů uložit do sdílené konfigurace. Uživatelé tak volbou načíst zvolí jméno reportu, obrazovka se rozšíří o hodnoty definované ve vstupních polích.

3.1.8 Zpracování elektronické dokumentace

Uchování a transformace je dalším faktorem pro zavedení ECM systému a stává se tak jedním z hlavních požadavků zákazníka. Převod a transformace z papírové formy dokumentu do elektronické Alfresco podporuje skenovat a OCR technologii. Převedené doku-

menty jsou z velké části využívány dále systémy třetích stran.

Zpracování papírové dokumentace a transformace do elektronické formy dále zvyšuje hodnotu implementovaného ECM systému v organizaci. Dle požadavků zákazníka lze sken dokumentů plně automatizovat v kombinaci s podporovaným hardware zařízením. Převodem dokumentu do elektronické podoby neztrácí právní náležitosti.

Podpora integrace s různými skenovacími systémy poskytuje flexibilní a inteligentní zpracování dokumentů. Skenovaná data během procesu lze mapovat na konkrétní typ dokumentu a v systému tak dále využít funkce verzování a integrovat do procesu workflow. Popsané řešení lze dále použít pro archivaci dokumentů a interním auditem nad dokumentu uloženými v systému. Další důvodem pro nasazení popsané technologie je snížení nákladu na tisk, archivaci a distribuci v papírové formě.

Skenování a transformace do elektronických formulářů je využito zejména pro

- Zpracování faktur,
- Zpracování žádostí,
- Zpracování pojistných smluv,
- Uložení smluv.

Elektronické zpracování dokumentů

Správa dokumentů v papírové formě je poměrně náročné na prostor a celkové náklady s nimi spojené. Pokud je organizací distribuována pouze papírová forma dokumentu, může dojít k poškození či ztrátě dokumentu. Naopak pokud je např. během procesu schvalování využívána kopie, rostou opět náklady na tisk. Uvedený postup je nevhodné využít pro samostatné pobočky rozmístěné v různých městech, státech. Dalším nevýhodou je rychlost přidělení a distribuce, zde je jednoznačná nevýhoda od elektronické distribuce.

Koncept elektronického zpracování dokumentů je založený na skenování a digitalizaci dokumentů. Hlavním benefitem pro zpracování dokumentů v elektronické formě je redukce požadavků na prostor. Dokumenty jsou uloženy v magnetických nebo optických médiích, méně citlivých na stárnutí materiálu, teplotu a povětrnostní podmínky. Pro dokumenty lze poměrně snadně nastavit bezpečností oprávnění, na rozdíl od oběhu v papírové formě. Elektronické dokumenty lze sdílet a využít tak informace uložené pro více uživatelů současně, každý přístup nebo změna může být auditována – lze poměrně snadno získat informace o přístupu a změnách realizovaných v dokumentu.

3.1.9 Správa systému

Systém během funkčního období vyžaduje monitoring a podporu. V závislosti na definici kritického systému je navržen i plán monitoringu a celková podpora systému. Prvotní import dat z již nepodporovaných nebo ukončených systémů lze zařadit do podpory a je proveden ve většině případů během nahrazení nové implementace. Podpora a správa systému vyžaduje pravidelné zálohy a monitoring log souborů, podpora uživatelsky hlášených případů.

Alfresco Content Package (ACP)

Alfresco v uživatelském rozhraní webové aplikace nabízí importní a exportní funkci vhodnou zejména pro přesun určité množiny dat do odlišné Alfresco repository. Alfresco Content package je soubor s příponou acp nesoucí veškerá metadata a obsahy objektů pro přenos. Export na základě označeného adresáře vytvoří acp soubor popisující označené data. Importní funkce provede naopak na základě import struktury do Alfresco repository.

Export a import celku repository

Komplexní export obsahuje následující informace:

- Uživatelská data – jména, ID, hesla,
- Uživatelské skupiny – informace o definici skupin a jejich členstvích,
- Definice kategorií a podkategorií,
- Šablony adresářů, prezentací a emailových zpráv,
- Definice skriptů v systému,
- Kompletní adresářovou strukturu,
- Kompletní data o diskusních fórech,
- Veškeré data o dokumentech,
- Metadata – informace o aspektech a audit informacích,
- Definici business pravidel,
- Definice bezpečnostních oprávnění,
- Informace o procesech,
- Uložené definice vyhledávání.

Export a import repository je možný v administrátorské konzoli dostupné z webové aplikace Alfresco nebo pomocí command-line souborů definující přesné pořadí parametrů. Poslední variantou je využít přímo pravidlo business rule pro definici import / exportu.

Záloha systému

Nejdůležitějším bodem v podpoře systému je záloha systému. Ztráta dat a ne-konzistence, která hrozí během havárie je kritickou částí. Havárie může být způsobena hardwarovou chybou, napadení systému virem či malware, chybou operačního systému atd.

Alfresco uchovává informace o objektech v databázi a souborovém systému. Záloha systému vyžaduje zálohu souborového systému, s daty o obsazích a dále databázi nesoucí informace o objektech. Další nutnou zálohou jsou konfigurační adresáře systému.

Ideálním řešením je definice úkolu na úrovni systému, vytvářející pravidelnou zálohu datových a konfiguračních adresářů a dále nastavit pravidelný „dump“ databáze. Exportované data je vhodné přesunout na externí úložiště nebo médium, protože uchování na identickém serveru kde je běhové prostředí serveru je nedostatečné. Hlavním důvodem je především varianta, kdy systém není možné korektně spustit.

Monitoringem systému je především log soubor uložený na aplikačním serveru. Konfigurací lze ovlivnit úroveň logů a další parametry. Pokud je Alfresco nasazeno na aplikačním serveru Tomcat, log soubory jsou uloženy v instalačním adresáři.

4 POUŽITÉ TECHNOLOGIE

Debian Lenny 5.0

Debian je svobodný operační unixový systém založený na konzervativním přístupu, vhodný zejména pro serverové stanice. Systém nabízí téměř identické portace softwarových balíčků, porovnáme-li ho s enterprise verzí RHEL od společnosti RedHat. Od srovnatelného systému využívá APT (Advance Package Tool) balíčkovací systém na rozdíl od RHEL využívající RPM (RPM Package Manager). Přístup aktualizace nových software balíčků do systému prochází důkladné testovacím procesem. Systém zpravidla neobsahuje novinky z důrazem na stabilitu serverových stanic.[18]

GlassFish

GlassFish je pod GPL licencí dostupný aplikační server projektově navazující na aplikačního serveru SunApp. Architekturu a komponentami dále pokračuje v modulární architektuře určené pro podporu platformy Java EE. Základ operačního systému GlassFish tvoří open source zdrojové kódy společnosti SUN a relačního mapování Oracle TopLink, webový server je derivátem Apache Tomcat produktu. Instalace v diplomové práci navíc využívá komponentu ESB rozšiřující vlastnosti aplikačního serveru. Na aplikační server je mimo modulu ESB instalována portace komponent pro BPEL procesy např. Sun-http-binding.[19]

Netbeans ESB

Netbeans ESB je vývojové prostředí původně od společnosti Sun Microsystem. V současnosti po dokončení akvizice spadající pod společnost Oracle. Vývojové prostředí bylo vybráno především pro podporu definice SOA BPEL procesu s podporovanou integrací aplikačního serveru GlassFish. Mimo standardní vývojové rozhraní ESB verze nabízí grafický návrhář BPEL procesu a podporu SOA projektů.[20]

ArgoUML

ArgoUML je modelovací nástroj pro tvorbu digramů v softwarovém inženýrství podporující UML specifikaci 1.4. Použitá verze je vytvořená v jazyce Java a volně dostupná pod licencí open source BSD.[21]

Mercurial

Mercurial je distribuovaný platformě nezávislý správce zdrojových kódů primárně určený pro linuxové stanice. Funkcionalitou patří do nové generace správců zdrojových kódů.

Mimo operace pro uchování zdrojových kódů v textové a binární podobě dále podporuje uchování lokální historie, proto označení jako distribuovaný. Operace s mercurial lze ovládat pomocí příkazové řádky, existují však grafické nadstavby a integrace do vývojových prostředí. Pro diplomovou práci byla zvolena přímá integrace do Netbeans vývojového prostředí.[22]

Dia

Dia je grafický program pro tvorbu rozsáhlé řady typů diagramů a technických schémat. Dostupný je v GPL licenci se závislostí na GTK+ knihovnách. Implementace Dia programu byla inspirována komerční aplikací „Visio“ využívána především na operačních systémech Windows. Tvorba diagramů je podporována v řadě typů technické schémat, Cisco komponenty, UML diagramy, síťové schémata a mnoho dalších.[23]

VirtualBox

VirtualBox je virtualizačním nástrojem pro architektury x86 a AMD64/Intel64 v enterprise prostředích. Open Source varianta produktu je vhodná pro domácí virtualizované řešení. VirtualBox nabízí v základu komfortní uživatelské rozhraní s možností detailně konfigurovat vlastnosti virtualizovaného systému. VirtualBox je dostupný na platformách Windows, Linux, Macintosh a OpenSolaris a podporuje řadu virtualizovaných prostředí.[24]

GIMP

GIMP je multiplatformní grafický nástroj určený pro editaci grafických formátů. GIMP podporuje řadu grafických formátů a pokročilých funkcí pro práci s grafikou. Verze je volně dostupná pro řadu platforem pod licencí GPL.[25]

II. PRAKTICKÁ ČÁST

5 ARCHITEKTURA INTEGROVANÉHO SYSTÉMU

Kapitola pojednává o návrhu implementovaných systémů a doporučených postupech nastudovaných v teoretické části. Podkapitoly popisují návrh architektury, kde je popsána koncepce architektury dělené na serverové instalace až po vzájemnou komunikaci integrovaných systémů. V přehledné formě je tak prezentováno fyzické dělení systému a využití referenčních aplikací dle rolí uživatele.

5.1 Seznámení se zadáním problematiky

Jak již bylo uvedeno v úvodu, diplomová práce dále rozvíjí implementovaný systém, který byl již použit v bakalářské práci.[11] Jednalo se o referenční zpracování jednoduchého knihovního systému se základní funkcionalitou a důrazem na aplikaci Java EE architektury. Během návrhu a implementace byly kladeny požadavky na platformní nezávislost, snadnou údržbu a možnou rozšiřitelnost systému v dalších fázích provozu. Před implementací bakalářské práce byla platforma Java EE upřednostněna před využitím Spring Framework, z důvodu dostupné nové specifikace EJB 3.0. Od předchozí specifikace EJB 2.1 učinila specifikace zásadní změny v konceptu volání rozhraní. Spring Framework je naopak označován jako „lighweiht“ kontejner pro zjednodušenou implementaci enterprise aplikací v porovnáváme-li náročnost s EJB 3.0. Implementovaný systém knihovny obsahoval desktop aplikaci a tenkého klienta v podobě webové aplikace, nasazené na aplikačním serveru Sun App Server 7. Výhodou Java EE architektury byla centralizována komunikace na jednotné rozhraní EJB komponent nasazených na aplikačním serveru. Desktop dostupný pouze v interní síti knihovny využíval vzdálených rozhraní, webová aplikace nasazená na identickém stroji s EJB komponentami komunikovala pomocí lokálních rozhraní.

5.2 Specifikace systémů

Architektura systému realizována v bakalářské práci byla podrobena analýze jednotlivých částí systému. Původní systém byl založen na operačním systému Ubuntu 6.06 s aplikačním serverem Sun App Server 7 a perzistentním uložištěm databázového serveru MySQL. Operační systém již v dnešní době není podporován, proto byl nahrazen novějším operačním systémem Debian 5.0 Lenny. Z důvodu snadněji realizovatelné údržby a celkové podpory systému byl navržen oddělený server pro migraci middleware serveru a ECM Alfresco systém. Kritický je z pohledu návrhu middleware server, protože poskytuje veškerou implementovanou business logiku.

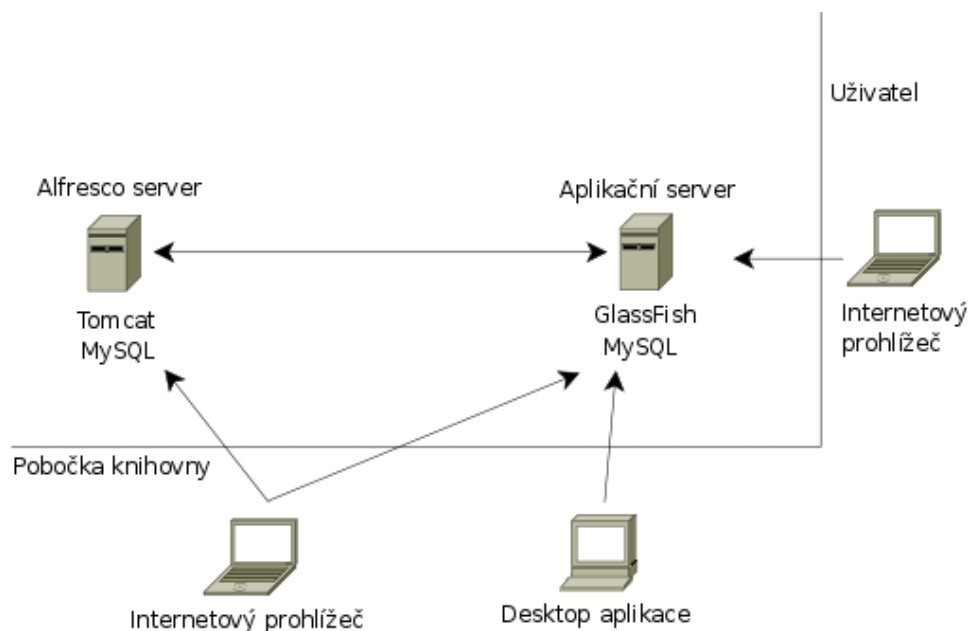
Běh operačních systémů byl vyhrazen ve virtualizovaném prostředí VirtualBox, který je dostupný v enterprise licenci pro komerční využití a také open source licenci pro

testovací nebo domácí použití. VirtualBox byl zvolen z důvodu vysoké výkonosti poskytující profesionální virtualizované řešení. V současné době sílí tlak na úsporu nákladů v IT infrastruktuře, především díky neustále rostoucím cenám energií. Uvedené faktory s výhodou lépe proveditelné správy serverů se stále častěji prosazují ve virtualizovaném řešení serverů. Výjimkou mohou být kritické systémy náročné na výpočetní výkon nebo zejména databázové servery, které jsou naopak náročné na vstupní/výstupní operace.

Operační systém instalovaný ve virtualizovaném prostředí byl pomocí klonovací utility pře-migrován se základní instalace middleware aplikačního serveru pro další cílovou platformu ECM serveru. Základní instalace operačního systému obsahovala:

- server instalace Debian Lenny 5.0,
- souborový systém EXT3 v Logical Volume Managementu pro správu diskového prostoru,
- Gnome grafické rozhraní 2.2.23,
- MySQL,
- VirtualBOX Guest Additions,
- OpenJDK 1.6.0_16.

Jednotlivé serverové instance budou popsány v následujících pod-kapitolách viz. Obr. 7.



Obr. 7: Schéma integrovaných systémů

Debian aplikační server

Standardní instalace s grafickým rozhraním byla rozšířena o databázový server MySQL. Pro základní potřeby byla instalována Java Open JDK 6.0 a aplikační server GlassFish 2.1.

Z nutnosti komfortního nástroje pro vývoj bylo nainstalováno specializované vývojové prostředí Netbeans IDE 2.2 s podporou ESB a JBI integrace. Pro správu informací o zdrojových kódech byl nainstalován správce zdrojových kódů source version control systém Mercurial. Konfigurace aplikačního serveru je znázorněna v tabulce 5.

<i>Popis</i>	<i>Nastavení</i>
Paměť	1400MB
Pevný disk- VirtualBox formát	14/16 GB
Grafické rozhraní	ANO

Tabulka 5: Konfigurace aplikačního serveru.

Debian Alfresco server

Instalace identicky s aplikačním serverem obsahuje databázový server MySQL a pro prvotní nastavení bylo využito grafické rozhraní. Alfresco 3.2r Community version pro konfiguraci mimo databázi vyžaduje instalaci Java Open JDK 6.0 a nástroje operačního systému a Open Office využívané pro transformaci obsahů. Operační systém pro konfiguraci byl nastaven pro zavedení pouze do textového režimu, z důvodu úspory operační paměti a především nepotřebného X server grafického rozhraní. Konfigurace Alfresco serveru je znázorněna v tabulce 6.

<i>Popis</i>	<i>Nastavení</i>
Paměť	800MB
Pevný disk- VirtualBox formát	7/16 GB
Grafické rozhraní	NE

Tabulka 6: Konfigurace Alfresco serveru.

Klientské stanice

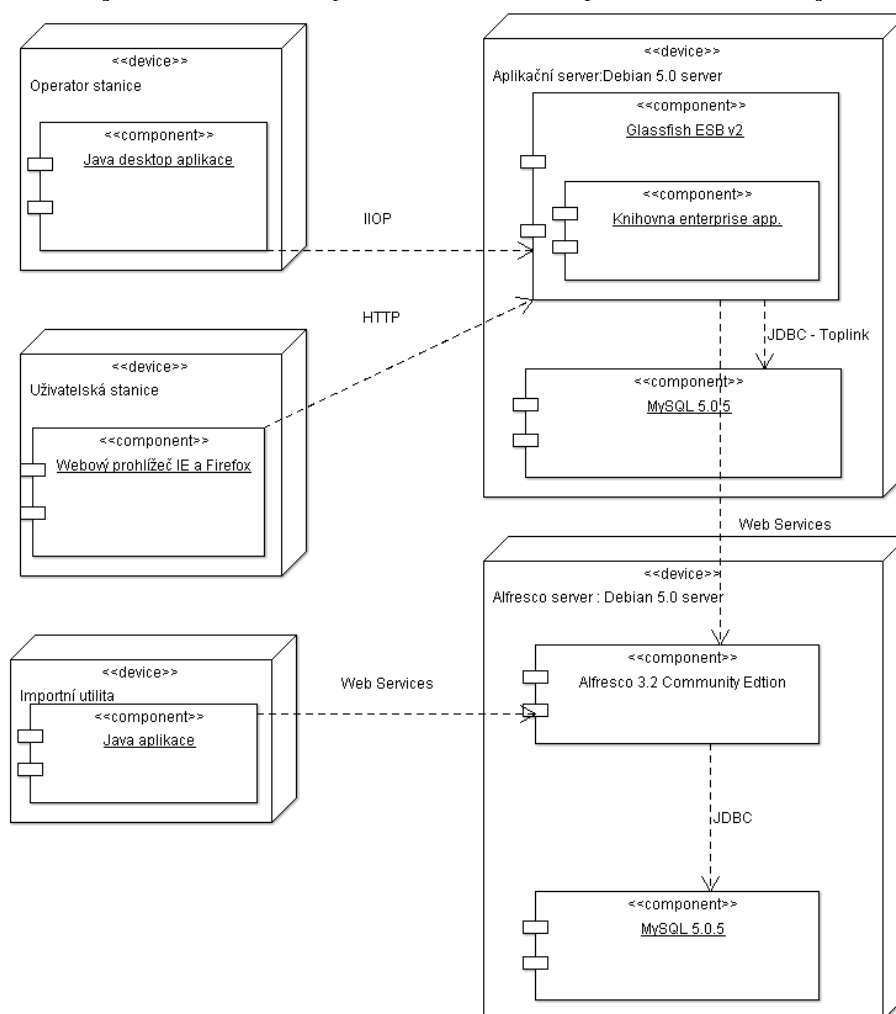
Klientské stanice přistupující k publikovaným službám na serverových stanicích se liší z pohledu využití. Zaměstnanci knihovny obsluhují ve většině případů Desktop aplikaci implementující běžné rutinní business operace nejčastěji využívané během dennodenního pracovního procesu. Požadavky na spuštění desktop aplikace je grafické rozhraní a Java Runtime prostředí ve verzi 1.6. Další variantou je přístup tenkým klientem pomocí Internetového prohlížeče k ECM systému Alfresco, popř. referenční webové aplikaci knihovny. Zaměstnanec přistupuje k Alfresco systému v případě pokročilých požadavků při práci dokumenty např. schválení dokumentu. Uživatelé registrovaní v integrovaném systému přistupují vždy pouze pomocí Internetového prohlížeče.

6 ANALÝZA SYSTÉMU

V následující kapitole je popsán cíl a účel práce s funkční specifikací systému v UML jazyce.

6.0.1 Diagram nasazení

Kapitola popisuje v jazyce UML – deployment diagram fyzické rozložení systému na detailnější části realizovaného systému. Skladba systému je založena na kontejnerech poskytujících funkcionalitu vnořeným komponentám. Systém je hardware zařízení s přesnou technickou specifikací. Komponenty nasazené v systému označují instance prostředků realizující funkcionalitu požadovanou od implementovaného systému.



Obr. 8: Diagram nasazení

Na Obr. 8 jsou uvedeny hlavní uzly v podobě virtuálních serverů Debian Lenny 5.0 a klientských stanic. Aplikační server GlassFish 2.1 je základním prvkem v integrovaném systému. Veškeré klientské požadavky jsou směřovány právě na rozhraní aplikačního serveru. Výhodou uvedeného řešení je zejména fakt, centrální „fasády“ poskytující uživatelům jednotné funkce. Datové části a funkce jsou sdíleny na middleware úrovni pro desktop

a web aplikaci. Uvedený postup je výhodný pro opětovné využití komponent systému a celkovou úsporu zdrojů během implementace. Aplikační server poskytuje základní sadu kontejnerů:

- Middleware kontejner – definice EJB komponent s poskytovanými službami systému,
- Webový server – definice uživatelského rozhraní pro webovou aplikaci.

Middleware přistupuje k datům pomocí objektového relačního mapování Oracle Toplink pomocí standardu JDBC rozhraní s podporou transakčního zpracování XA protokolem. Oracle Toplink je pro nekomerční studijní účely poskytován zdarma.

Rozhraní se systémem Alfresco 3.2r je realizováno technologií webových služeb, a to standardní sadou definovanou v Alfresco API. Centrální úložiště repository využívá vlastní souborový systém operačního systému a databázový systém pro ukládání metadat technologií JDBC.

Konečnou částí systému je specializovaný importní nástroj pro přenos volně dostupných knih na adrese http://www.gutenberg.org/wiki/Main_Page do úložiště ECM Alfresco systému. Aplikace komunikuje s Alfrescem stejně jako v případě middleware vrstvy pomocí standardní sady webových služeb.

6.0.2 Funkční specifikace

V následující podkapitole jsou rozepsány požadavky kladené na systém včetně provedené analýzy v jazyce UML. Kapitola stanovuje jednoznačné hranice systému a výčet požadavků. Požadavky kladené na systém dělíme na typy:

- funkční – stanovuje požadované funkce od implementovaného systému. Běžně se zde specifikují technické vlastnosti a postupy, kterých má systém dosáhnout. Výsledné funkce analýzy jsou ve většině případů použity do diagramu užití (use case).
- nefunkční – naopak od předchozí kategorie vymezují omezující podmínky implementovaného systému týkající se především kvality, odezvy systému, bezpečnosti a tolerovaného množství výpadků.

Do nefunkčních požadavků spadají:

1. Implementovaný systém bude založen na standardizovaných platformách, komunikačních protokolech, datových formátech.

2. Systém identicky s předchozí implementací zachová model komponent a tím umožní snadné rozšíření v dalších fázích implementace.
3. Systém nebude pevně svázán s platformou, musí být platformě nezávislý.
4. Uživatelské rozhraní desktop a webové aplikace bude lokalizováno do českého jazyka.
5. Systém bude založen na platformě Java EE a relační databázi MySQL.
6. ECM systém bude obsahovat sadu operací pro přístup k systému bez nutnosti realizovat operace pomocí uživatelského rozhraní.
7. ECM systém bude nasazen na identickém operačním systému pro aplikační middleware knihovny.
8. ECM systém bude obsahovat uživatelské rozhraní pro administrativní úkony.
9. Instalace serveru bude realizována pro aplikační middleware a ECM systém na oddělených hardware strojích.
10. Metadata ECM systému budou uložena ve stejné relační databázi, kterou využívá aplikační server knihovny. Obě instalace poběží na rozdílném hardware stroji.
11. Systém bude aplikovat bezpečnostní politiku, proti zamezení neoprávněného přístupu k informacím nebo elektronickým obsahům.
12. Grafické rozhraní implementované v předchozí verzi zůstane zachováno.

Dále definujeme funkčních požadavky:

1. Integrovaný systém umožní uchování interních dokumentů a dále elektronické úložiště knih.
2. Systém podporuje fulltextové vyhledávání pro standardní formáty produktů Open Office, Microsoft Office, PDF a další textové soubory.
3. Systém bude uchovávat informace o zapůjčených knihách.
4. Systém poskytne uživatelům přístup k elektronickým dokumentům dle oprávnění.
5. Systém podporuje vyhledávání dle specifických podmínek.
6. Systém umožní ve webové aplikaci online registraci uživatelů.
7. Pro administrativního pracovníka na pobočce je poskytována funkce registrace uživatele a evidence knih v archívu.
8. Administrativní pracovník spravuje uživatelské účty pro přístup k systému.

9. Systém poskytne interním uživatelům základní proces schvalování.
10. Systém poskytne administrátorovi importní / exportní funkci přístupnou z souborového systému.
11. Nově registrováni uživatelé budou informováni o úspěšném schválení emailovou notifikací.

6.0.3 Studie případu užití

Mapování studie případu užití bude popsána v jazyce UML a doplněna o use diagram. Jazyk UML slouží k unifikované vizualizaci softwarových systémů. Specifikace pomocí jednoznačných pravidel popisují systém v textové nebo grafické podobě (diagramy). Jednotlivé diagramy jsou určeny pro popis jednotlivých úloh s přesně danou vnitřní konzistencí a sémantikou.

Využití systému vychází a dále rozšiřuje body stanovené ve funkční specifikaci. Základním úkolem je stanovit hranici systému, určující přesně definované funkce, které systém poskytuje. Jednotlivé funkce lze ve specifikaci dále omezit o detailnější popis funkcí, který systém nebude podporovat. Stanovením aktörů, uživatelé či externí systémy spolupracující s implementovaným systémem. Úkolem je proto stanovit přesné rozhraní na úrovni funkcí odpovídající určitému uživateli.[16]

Definice hranice systému

Systém implementuje následující funkce:

- Správa výpůjček,
- Registrace uživatelů (online),
- Katalog knih,
- Zobrazení knih (online),
- Centrální úložiště organizace,
- Jednokrokové schvalování dokumentů.

Definice aktörů systému

Aktoři systému reprezentují obecného uživatele zahrnutého do univerzálních rolí. Jednotlivé skupiny aktörů v reálném systému mohou zahrnovat skupinu pracovníků nebo uživatelů.

Aktoři systému byly definovány na:

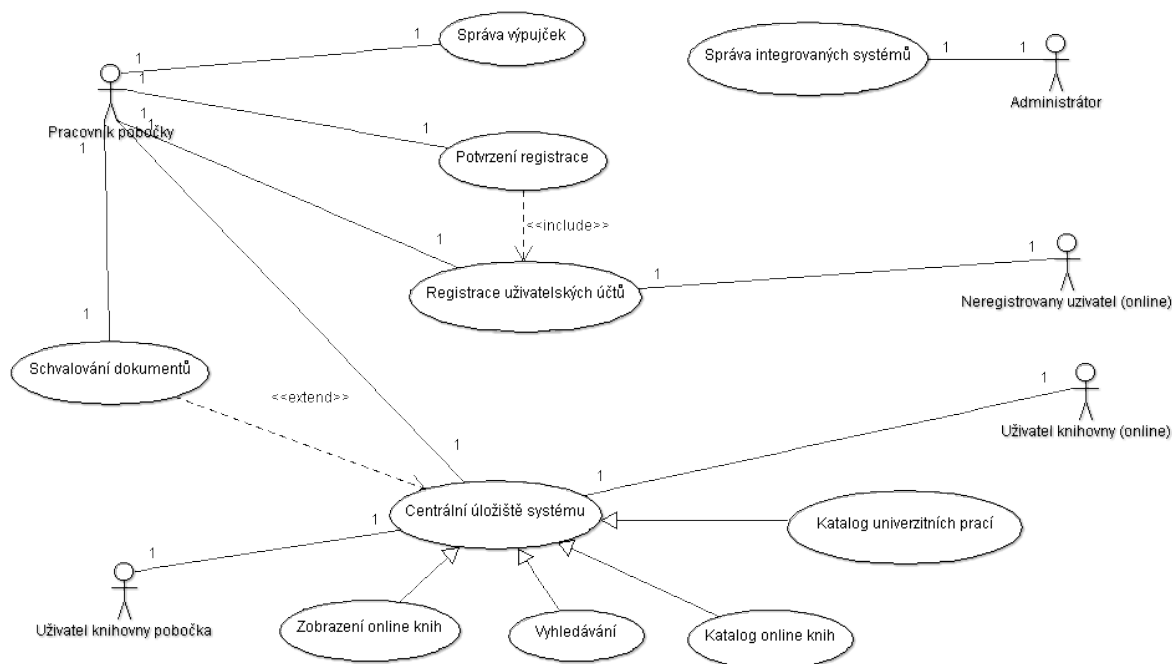
- Administrátor – správce systému,
- Pracovník na terminálu – zaměstnanec knihovny,
- Uživatel knihovny na pobočce – uživatel, který se nachází na pobočce knihovny,
- Neregistrovaný uživatel (online) – neregistrovaný uživatel bez oprávnění přistoupit k webové aplikaci knihovního systému,
- Registrovaný uživatel (online) – uživatel s oprávněním přistoupit pod uživatelským účtem k systému pomocí webové aplikace.

Definice případů užití

Diagram případů užití zachycuje vnější pohled na modelovaný systém a poskytuje tak přehled o rozsahu analyzovaného systému. Zásadním prvkem pro stanovení případu užití během analýzy před implementací je hranice určující poskytované funkce systému.[17]

V případě užití jsou definováni uživatelé v rolích:

- Administrátor,
- Pracovník pobočky,
- Uživatel knihovny na terminálu,
- Uživatel mimo pobočku online větev,
- Neregistrovaný uživatel.



Obr. 9: Diagram případu užití

Pro každou roli byly definovány funkce, které jsou popsány na diagramu Obr. 9. Speciálním typem v diagramu jsou klíčová slova *include*, zahrnující funkci navázanou na jiné aktivitě. Dále funkci *extend*, která vychází z určité aktivity a dále ji rozšiřuje.

6.0.4 Analýza procesů

Ideálním postupem dle metodiky UML je pro každý případ užití vypracovat sekvenční diagram popisující zúčastněné role začleněné do zpracovaného procesu. Proces je v sekvenčním diagramu popsán na úrovni interakce aktorů v časové ose. Popis v sekvenčním diagramu ve srozumitelné formě informuje čtenáře o komunikaci systémů a jejich vzájemné interakci.

Kompletní popis implementovaných procesů by byl rozsáhlý a přesáhl by tak rozsah diplomové práce. Proto byly definovány stěžejní procesy k detailnímu mapování a popisu pomocí sekvenčních diagramů. Každý z diagramů v horizontální ose obsahuje popis funkcí realizovaných v daném procesu. Počátek procesu probíhá od levé části diagramu přes další funkce zobrazené v pravé části diagramu. Operace typu požadavek a odpověď jsou zobrazeny plnou čarou, odpověď je naopak zobrazena přerušovanou čarou. Funkce vyvolané pouze požadavkem je zobrazena identicky bez zpětné reakce vyvolané funkce. Doba průběhu volání je zobrazena obdélníkem označující běžících operací v čase. Mapovaný proces tak obsahuje mimo přehledu využitých funkcí a přehled o časové náročnosti operací.

Zobrazení elektronických knih

Proces je složen z řady závislostí integrovaných systémů. Do procesu jsou zahrnuti klienti v podobě desktop nebo webové aplikace. Komunikace probíhající během procesu je složena z IIOP komunikace na EJB komponentu, transfer webovou službou a ze strany middleware webovou službu na rozhraní Alfresco API.

Proces vzniká na počátku zasláním požadavku uživatele v desktop nebo webové aplikaci na zobrazení elektronických knih. Uživatelské rozhraní následně vyvolá rozhraní na middleware systému knihovna s požadavkem navrácení dat o el. knihách. EJB komponenta nasazená v middleware aplikačním serveru poskytuje pouze informace identifikující objekt. Následuje proces autorizace pomocí webové služby, která na straně middleware vytvoří bezpečnostní token pro content webovou službu realizující přenos obsahu elektronické knihy. V uvedeném případě content webová služba pod předem definovaným uživatelem v systému Alfresco vyšle požadavek formě Lucene query systému Alfresco. Alfresco systém následně zpracuje požadavek a zpětně ve sledu volaných akcí dochází k transforma-

ci objektů z Alfresco reprezentace na interní objekt AlfContent. Transformace je realizována interně pomocí transformace uživatelsky definovaného modelu objektu knih na Java objekt AlfContent, který je následně transformován do modelu pro zobrazení v uživatelském rozhraní. Detailní popis procesu mapovaný na úroveň metod je dostupný na Obr. 19, který je přiložen v příloze.

Vyhledávání elektronických knih

Vyhledávání elektronických knih je od předchozího diagramu rozdílné v přístupu sekvencního zpracování. Proces je zjednodušen pouze o volání EJB komponenty, není zde závislost na webové službě přenášející obsah na klienta. Výsledkem operace je zobrazení datových záznamů splňujících zadanou informaci v podobě textové hodnoty. Inicializací procesu na straně klienta získá hledanou hodnotu, která je následně ve volání RMI předána ke zpracování AlfrescoSessionRemote EJB komponentě. Komponenta provede autentizaci do centrálního úložiště Alfresco a spustí sestavený dotaz na komponentu Lucene query search. Výsledkem je pole Alfresco objektů popisujících interní objekt elektronické knihy. AlfrescoSessionRemote objekt transformuje Alfresco objekty do interní Java reprezentace AlfBookEntity objektu. Sestavená kolekce objektů je dále předána klientovy pro další zpracování. Na straně desktop aplikace se jedná o transformaci kolekce objektů do Swing grid komponenty. Detailní popis procesu mapovaný na úroveň metod je dostupný na Obr. 20, který je přiložen v příloze.

6.0.5 Diagram tříd

Vývoj na platformě Java EE je založen na modelu komponent, které lze opětovně využít pro další implementovanou funkcionalitu nebo nasadit na odlišném aplikačním serveru. Pro implementaci byly vytvořeny oddělené projekty v jazyce Java SDK a Java EE na kterých jsou definovány vzájemné závislosti. Diagramy tříd jsou natolik rozsáhlé, že jsou v přílohách uvedeny pouze části diagramů. Kompletní diagramy tříd jsou součástí DVD média dokumentace k projektu.

Vývoj integrovaného systému byl tak rozdělen do projektů:

- prj.bc_ee_core – modul obsahuje obecné třídy využívané v dalších modulech. Implementované Java třídy obsahují transformaci Alfresco entity na interní reprezentaci knih a dále implementaci generování primárních klíčů.
- prj.dipl_ee_app-ejb – hlavní modul systému nasazený na middleware aplikačním serveru. Zde je implementovaná veškerá business logika systému knihovna.

- prj.bc_ee_app-app-client – klientské rozhraní pro komunikaci prj.bc_ee_app-ejb modulem.
- prj.bc-ee-app – definice Java EE aplikace obsahující moduly pro nasazení na aplikační server.
- prj.alf_bridge_ws – modul definující rozhraní mezi EJB komponentami a Alfresco systémem.
- prj.bc_ee_web – webová aplikace systému knihovna určená především návštěvníkům knihovny a online uživatelům.
- prj.desktop_app – desktop aplikace určená převážně interním zaměstnancům knihovny.
- prj.alf_import_app – ukázková aplikace importující volně dostupné knihy v textovém formátu do interního úložiště.

Modul prj.bc_ee_core

Knihovna core je sdílena skrze veškeré moduly. Jedná se o obecnou knihovnu s funkcionalitou generování primárních klíčů a mapování Alfresco entit na objekty. Část diagramu je přiložena v příloze Obr. 23.

Modul prj.dipl_ee_app-ejb

Projekt implementuje hlavní část business požadavků integrovaného systému. Funkce definované v rozhraních jsou využívány v desktop a webové aplikaci. Počet tříd a logika projektu je rozsáhlejší porovnáváme-li ji s ostatními projekty v diplomové práci.[6] Část diagramu webových služeb je přiložena v příloze Obr. 24.

Modul prj.alf_bridge_ws

Projekt obsahuje vygenerované proxy objekty webových služeb implementovaných v modulu prj.dipl_ee_app-ejb. Generované webové služby obsahují autentizační *AuthWebServiceService* a přenos obsahu pomocí webové služby *ContentWebServiceService*.

Modul prj.bc_ee_web

Webová aplikace obsahuje transformační třídy zpracovávající dat pro prezenční vrstvu. Další třídy jsou svázány s nově implementovanou logikou ve webové aplikaci. V prvním případě se jedná o BPEL procesem, kdy servlet aktualizuje po přístupu stav registrovaného uživatele. Další servlet je transfer třída pro přenos elektronických knih uživateli.

Modul prj.desktop_app

Desktop aplikace obsahuje rozsáhlou sadu tříd definujících EJB rozhraní a dále definici uživatelského rozhraní.

Modul prj.alf_import_app

Importní aplikace obsahuje sadu tříd pro průchod adresářovou strukturou a následně import do centrální repository. Alfresco třídy nejsou do modelu zahrnuty.

6.0.6 Datový model systému

Databázová vrstva je nejnižší vrstvou v uvedené vícevrstvé architektuře. Během integrace systémů zůstala ponechána bez zásadních změn. Hlavním důvodem je již komplexní navržený model v bakalářské práci a dále fakt, že sdílená data nejsou duplikována. Informace o elektronických knihách jsou uloženy pouze v Alfresco systému a na základě požadavku jsou navraceny.

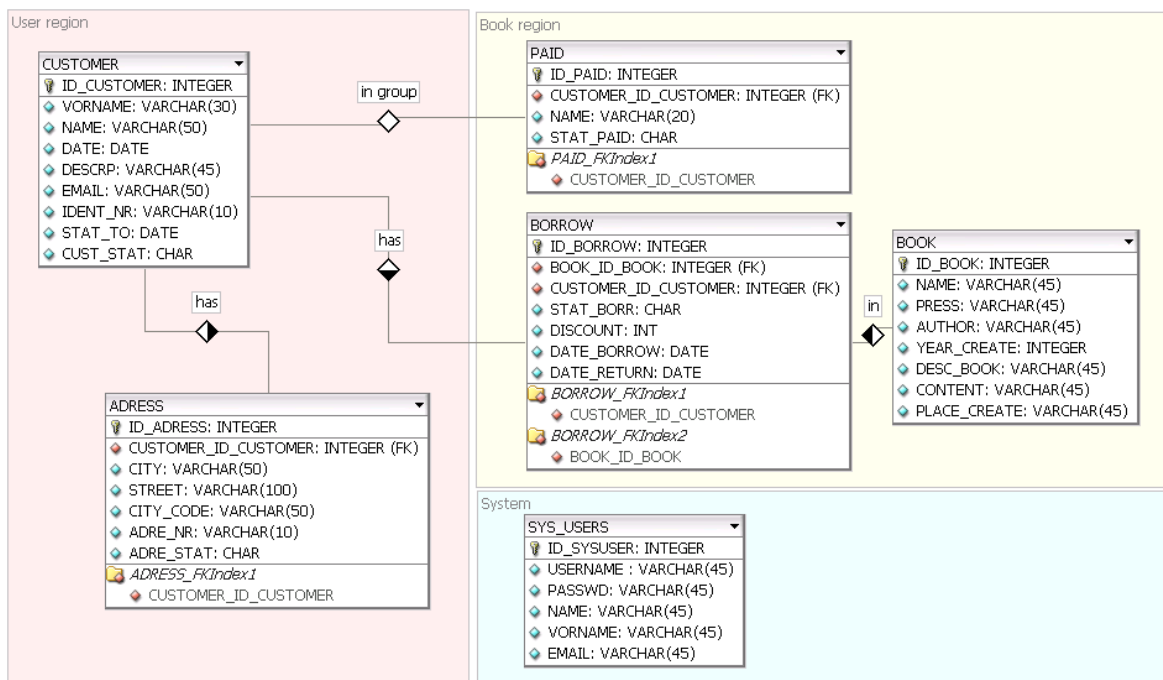
6.0.7 Databázová vrstva knihovny

Informace o uživatelích, knihách a výpůjčkách jsou perzistentně uložena v relační databázi. Připojení z aplikačního serveru do databáze je realizováno pomocí standardu databázového rozhraní JDBC. Aplikace již v bakalářské verzi obsahovala MySQL databáze verze 5.0 podporující procedury a trigery, které nebyly v implementaci využity. Migrace dat proběhla do novější verze MySQL 5.0.5. Transakční zpracování operací je realizováno plně v režii aplikačního serveru. Přístup k perzistentním datům z aplikačního serveru byl nastaven s podporou poolování. Tzn. aplikační server spravuje vytvořené databázové spojení, které na vyžádání předá k zpracování. Ukončením operace je databázové spojení zpětně zařazeno do pool, veškerá logika pro správu zdrojů je realizována na aplikačním serveru. Využití je vhodné zejména pro velkou časovou náročnost připojení k databázi vyvolané spuštěním každého SQL dotazu. Datový model byl navržen v DB Designéru vhodný právě pro zvolenou databázi MySQL.[11]

Definice entit popisující schéma Obr. 10:

- CUSTOMER – entita popisující zákazníka knihovny. Obsahuje relace do tabulky adres, vyúčtování a vypůjčených knih.
- ADRESS – entita popisující adresu klienta, lze definovat různé typy adres jakými jsou např. trvalá, přechodná.

- BORROW – entita popisující vztah vypůjčených knih a klienta knihovny. Relace jsou realizovány pomocí primárních klíčů tabulek CUSTOMER a BOOK.
- BOOK – entita popisující knižní díla uložené v knihovně.
- SYS_USER – entita systémových uživatelů.



Obr. 10: Datový model knihovny

7 IMPLEMENTACE

Při zpracování implementační kapitoly předkládané diplomové práce byl kladen důraz na aplikaci metodiky návrhových vzorů do praxe.

Implementovaná desktop aplikace spravovala základní evidenci klientů knihovny a seznam dostupných knih k zapůjčení. Informace o klientech a knihách bylo možné získat v detailním dialogu. Obrazovka zapůjčení knih obsahovala informace, které knihy a k jakému datu musí klient navrátit na pobočku knihovny. Desktop aplikace v novém konceptu integrovaného systému zachovává kompletní funkcionalitu předchozího systému, rozšířenou však o integraci do centrálního úložiště. Integrace tak zpřístupňuje pracovníkovi na pobočce informace o elektronických knihách a univerzitních pracích včetně zobrazení obsahu, vyhledávání s podporou fulltext a dále informace o dokumentech uložených v Alfresco systému. Pracovník v desktop aplikaci finálně schvaluje online registraci uživatele.

Uživatelům přistupujícím mimo pobočku knihovny je určena referenční webová aplikace. Informace vzdáleného přístupu jsou zabezpečeny ověřením validního uživatelského účtu. Po úspěšné autentifikaci získá uživatel přístup k informacím o dostupných knihách, o vypůjčených knihách přihlášeného uživatele, zobrazení obsahu elektronické knihy a seznam univerzitních prací. Registrace nového uživatele naopak z principu nevyžaduje ověřování, které je následně provedeno během dostavení uživatele na pobočku. První realizovaná zápůjčka a osobní navštívení pobočky obnáší verifikaci údajů dle zadaného registračního formuláře. Integrace systému tak nově zpřístupňuje online registraci uživatelů bez nutnosti osobně navštívit pobočku knihovny.

7.1 Integrace systému Alfresco

Alfresco je nasazeno v integrovaném systému pro softwarové řešení centrálního úložiště a navázaným procesem schvalování. V době přípravy práce byla volně dostupná verze Alfresco Community 3.2r, která přináší novinky a inovaci do ověřené architektury systému. Hlavní změnou je náhrada vnitřního relačního mapování Hibernate na produkt iBatis, dále změna přístupu k verzovací AVM službě. Kompletní přehled o změnách je uvedený na webové stránce: http://wiki.alfresco.com/wiki/Alfresco_Community_Edition_3.2r. Instalace proběhla na doporučené konfiguraci operačního systému Debian Lenny 5.0, relační databázi MySQL 5.0.5 a webovém serveru Tomcat 6.0.[5]

7.1.1 Instalace

Instalace proběhla dle standardních instalačních kroků a konfigurace složené z následujících částí:

- Konfigurace úložiště Lucene indexu dat – cesta v operačním systému obsahující úložiště Lucene indexovaných dat. Běžně se uvádí podadresář v hlavní instalaci Alfresco.
- Konfigurace relační databáze – jedná se JDBC konfiguraci databázového spojení k relační databázi.
- Externí aplikace – Alfresco pro transformace využívá projektů třetích stran, převážně se jedná o OpenOffice.

Typ community instalace již obsahoval webový server Tomcat nakonfigurovaný na webový kontext alfresco a port 8080.

7.1.2 Definice typů

Informace o typech dokumentů je uložena v meta modelu Data Dictionary. Prvním typem je definice typů objektů, Alfresco ve standardní instalaci již obsahuje definice pro adresář a dokumenty. Definice typů je založena na schématech a konfiguraci XML souboru, proto je možné poměrně snadno definovat dědičnosti na již existujících typech. Druhým typem je aspekt objektu, který byl popsán v teoretické části.

Dokumenty plánované pro ukládání v úložišti jsou online knihy a univerzitní práce převážně textového, Office nebo PDF formátů. Zbývající dokumenty jsou uloženy se standardní sadou atributů definovanou pro objekt dokument nebo adresář. Definice typů je uložena v cestě:

```
/opt/alfresco/tomcat/shared/classes/alfresco/extension/customModel.xml
```

Hlavnička definice vychází z Alfresco content modelu, zde je i konfigurace omezených podmínek pro typ objektu. V konfiguraci je pro typ univerzitní práce použit pevná sada vlastností pro typ práce. Pro prezentaci příkládám definici uživatelského modelu.

```
<!-- Custom Model -->

<!-- Note: This model is pre-configured to load at startup of the Repository. So, all custom -->
<!-- types and aspects added here will automatically be registered -->

<model name="customlibmodel:customModel" xmlns="http://www.alfresco.org/model/dictionary/1.0">

  <!-- Optional meta-data about the model -->
  <description>THO Library Custom Model</description>
  <author>Tomas Hoferek</author>
```

```

<version>1.0</version>

<imports>
  <!-- Import Alfresco Dictionary Definitions -->
  <import uri="http://www.alfresco.org/model/dictionary/1.0" prefix="d"/>
  <!-- Import Alfresco Content Domain Model Definitions -->
  <import uri="http://www.alfresco.org/model/content/1.0" prefix="cm"/>
</imports>

<!-- Introduction of new namespaces defined by this model -->
<!-- NOTE: The following namespace custom.model should be changed to reflect your own namespace
-->
<namespaces>
  <namespace uri="custom.lib.model" prefix="customlibmodel"/>
</namespaces>

<constraints>
  <!--
  THO thesis constraint
  -->
  <constraint name="customlibmodel:costrThesisList" type="LIST">
    <parameter name="allowedValues">
      <list>
        <value>Semestrální práce</value>
        <value>Bakalářská práce</value>
        <value>Diplomová práce</value>
        <value>Disertační práce</value>
      </list>
    </parameter>
  </constraint>
</constraints>

```

Typ kniha (*booktype*)

Typ kniha je primárně určen pro ukládání knih v elektronické podobě. Kniha vychází z obecného abstraktního modelu. Seznam atributů je uvedený v přiložené ukázce. Speciálním typem je atribut language, který může nabývat vícenásobných hodnot a postihuje tak variantu knihy ve více jazykových mutacích. Kniha byla definována s následujícími atributy.

```

<!--
  THO custom abstract type
  -->
  <type name="customlibmodel:booktype">
    <title>Book library model</title>
    <parent>customlibmodel:abstype</parent>
    <properties>

      <property name="customlibmodel:author">
        <type>d:text</type>
      </property>

      <property name="customlibmodel:publisher">
        <type>d:text</type>
      </property>

      <property name="customlibmodel:subject">
        <type>d:text</type>

```

```
</property>

<property name="customlibmodel:year">
<type>d:int</type>
</property>

<property name="customlibmodel:locnr">
<type>d:text</type>
</property>

<property name="customlibmodel:language">
<type>d:text</type>
<multiple>true</multiple>
</property>

<property name="customlibmodel:type">
<type>d:text</type>
</property>

<property name="customlibmodel:abstract">
<type>d:text</type>
</property>

<property name="customlibmodel:keywords">
<type>d:text</type>
</property>

<property name="customlibmodel:copyright">
<type>d:text</type>
</property>

<property name="customlibmodel:pages">
<type>d:int</type>
</property>

</properties>
</type>
```

Typ univerzitní práce (thesis)

Na typ thesis nebyla aplikována dědičnost od typu kniha, protože se nejedná o shodnou sadu atributů potřebnou uchovávat na univerzitních pracích. Na atributu publicationtype je definováno omezující pravidlo obsahující výčet typů prací. Omezující pravidla jsou definovány v hlavičce definice typu pod elementem <constraints>. Uživatel tak ve webové aplikaci musí specifikovat o jaký typ práce se jedná.

7.1.3 Adresářová struktura

Adresářová struktura je navržena pro uchování elektronických knih a univerzitních prací. Pro uvedené typy objektů byla nadefinována adresářová struktura s oprávněním dokumenty číst. Změna struktury a jakékoliv zásahy pod struktury je nutné provádět pod administrátorským účtem uživatele.

<i>Umístění složky v repository</i>	<i>Význam</i>
/Company Home/Libraries	Složka obsahuje prvotní import volně dostupných knih z www.gutenberg.org . Hlavní adresář je možné rozšířit o další adresáře svázané s konkrétním vydavatelstvím.
/Company Home/Theses	Složka obsahuje členění dle typu práce na bakalářské, diplomové a disertační práce. Další členění je hierarchicky rozděleno na rok vypracování a název práce včetně autora.

7.1.4 Centrální úložiště

Návrh úložiště byl realizován pouze pro využití zaměstnanci knihovny, kteří mají povinnost uchovávat veškerou pracovní dokumentaci v Alfresco systému. Uvedený postup je doporučen zejména z důvodu bezpečnosti, protože dokumenty ukládané na přenosných médiích nebo lokálních stanicích nevyhovují bezpečností politice. Další faktorem je pravidelná záloha systému ošetřující případnou hardware havárii. Definice uživatelů je opět ve správě administrátorského účtu Alfresca. Dokumenty lze ukládat dvěma typy:

- Webová aplikace Alfresco – komfortní uživatelská operace podporující kompletní sadu operací s dokumenty. Dokumenty lze prohlížet, vyhledávat, importovat/exportovat a zahájit proces schvalování.
- WEBDAV – technologie zpřístupňuje adresaci dokumentů na http adresu. Uživatel se tak odkazuje na URL, na které je dokument volně dostupný.
- Sdílená jednotka – v operačním systému se složka připojí jako nová síťová jednotka. Uvedený postup je vhodný zejména pro import a export dokumentů.

Webová aplikace je dostupná na adrese http://IP_ALF_SERVER:8080/alfresco a je hlavním rozhraním pro práci s dokumenty v systému Alfresco. Dostupné operace byly specifikovány v teoretické podkapitole 3.1.4.

WebDAV zpřístupňuje dokument pomocí adresace http protokolem. Stejně jako v ostatních případech jsou vyžadovány uživatelské údaje pro přístup k dokumentu, pokud není vysloveně nastaveno oprávnění číst neautentizovaným uživatelům „Guest“ WebDAV adresa je dostupná v detailních vlastnostech dokumentu pod odkazem „View“ in WebDAV.

Sdílená jednotka je založena na protokolu NETBIOS a údaje jsou identické stejně jako v případě připojení síťové jednotky v operačním systému Microsoft Windows nebo samba klientem v případě unixového systému. Pro připojení je nutný platný uživatelský účet do systému Alfresco.

7.2 Implementace business procesů v systému Alfresco

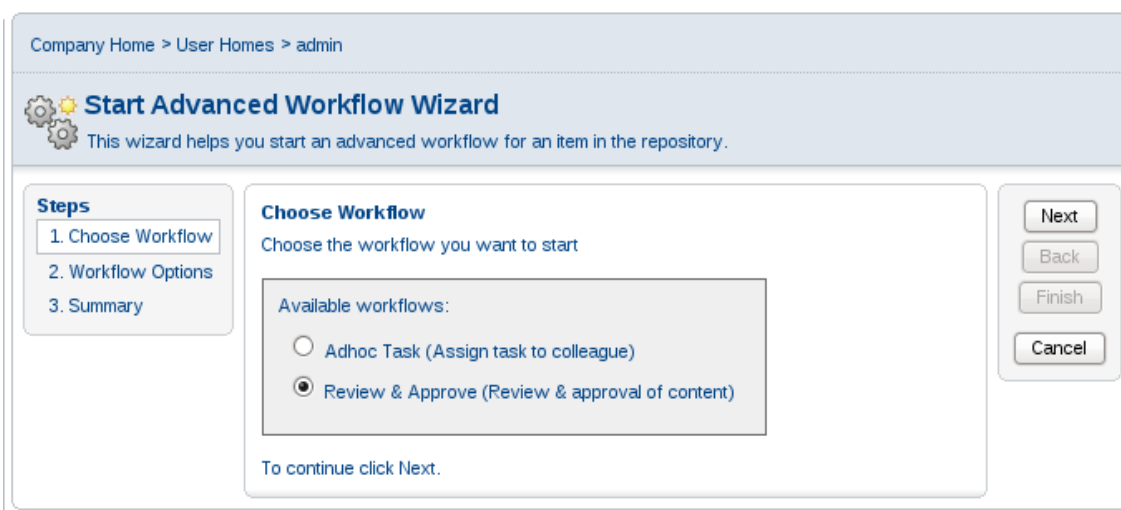
Alfresco systém mimo funkci hlavního úložiště je navrženo, ke zpracování elektronické dokumentace a to pomocí nástroje Workflow. Workflow je hlavním článkem pro organizaci procesů zpracovávající elektronickou dokumentaci. V běžném procesu je implementace navržena pro schválení dokumentů nebo uložení platných vyhlášek v organizaci. Uživatelé jsou informováni např. o nově platné vyhlášce knihovny. Uvedený postup se zpravidla řídí zpracování uživatelem v roli editor. V dalších fázích je dokument předán k schválení nebo lze dokument delegovat na vyjádření dalších uživatelů pro objektivní zpracování problematiky.

Proces schvalování

Proces schvalování od nasazení systému Alfresco bude probíhat pouze v DMS systému. Pro schvalování bude využita standardní šablona workflow „Review & Approve (Review & approval of content)“. Veškeré dokumenty ke schválení musí být uloženy v elektronické podobě, papírový originál bude členěn od lokálního archívu dle přiřazeného kódu pro snadné dohledání originálu v případě potřeby. Elektronická reprezentace dokumentu tak po inicializaci workflow a nastavení parametrů po vytvoření instance dále přechází zvolenému uživateli.

Proces schválení dokumentu je složen z kroků:

1. Iniciátor workflow je uživatel žádající o vyjádření či schválení dokumentu.
2. Nad elektronickým dokumentem je zahájen proces „Advance workflow“.



Obr. 11: Inicializace schvalování dokumentu

3. Uživatelské rozhraní uživateli poskytuje výběr typu standardních workflow. Pro schválení dokumentu je nutné zvolit volbu „Review & Approve“ zobrazenou na obrázku: Obr. 11.

4. Další krok obsahuje volbu vlastností workflow s informacemi o popsaném úkolu, do kdy je nutné se k úkolu vyjádřit a finálně pro kterého uživatele je úkol určen. Obrazovka pro specifikaci vlastností je uvedena na obrázku: Obr. 12.

Company Home > User Homes > admin

Start Review & Approve Workflow Wizard

This wizard helps you start an advanced workflow for an item in the repository.

Steps

1. Choose Workflow
2. Workflow Options
3. Summary

Workflow Options
Select options for the workflow

Properties

General

Description:

Review Priority:

Review Due Date:

Users and their Roles

Reviewer:

Resources

Name	Description	Path	Created	Modified	Actions
bpel.dia		/Company Home/User Homes/admin	13 May 2010 17:07	13 May 2010 17:07	<input type="button" value="D"/> <input type="button" value="U"/> <input type="button" value="R"/> <input type="button" value="X"/>

Page 1 of 1

To continue click Next.

Obr. 12: Vlastnosti schvalování dokumentu

5. Po odeslání je uživatel informován v sumarizačním okně o zadaných vlastnostech. Ukončení procesu je realizováno tlačítkem „Finish“.
6. Uživatel je o novém úkolu informován v seznamu úkolů po přihlášení do webové aplikace Alfresco.
7. Proces schvalování lze sledovat ze stavu zpracovaného dokumentu. Ihned po přijetí uživatel v položce „Status“ nalezne informaci o nezahájené aktivitě na úkolu viz. obrázek: Obr. 13.

8. Volbou ke schválení lze ukončit úkol tlačítkem „Approve“, zamítnutí naopak „Reject“. Uživatel během vypracování úkolu mění stav úkolu, informace nezbytná zejména pro iniciátora workflow.
9. Ukončením úkolu pověřené osoby úkol dále přechází iniciátorovi do seznamu úkolů. Přijatý úkol opět obsahuje sadu příznaků pro vyhodnocení zpracovaného dokumentu a zpětnou reakci uživatele. Potvrzením úkolu je proces ukončen a úkol odstraněn z seznamu.

The screenshot displays the 'Manage Task: Review' interface in Alfresco. The page title is 'My Alfresco' and the main heading is 'Manage Task: Review' with the subtitle 'Review Documents to Approve or Reject them'. The interface is divided into two main sections: 'Task Properties' and 'Resources'.

Task Properties:

- General:**
 - Identifier: 2
 - Description: Prosím o schválení přiloženého dokumentu
 - Due Date: 20 | May | 2010
 - Priority: 2
 - Status: In Progress
 - Comment: (empty text area)

Resources:

Name	Description	Path	Created	Modified	Actions
bpel.dia		/Company Home/User Homes/admin	13 May 2010 17:07	13 May 2010 17:34	

Page 1 of 1

On the right side of the 'Task Properties' section, there are three buttons: 'Save Changes', 'Approve', and 'Reject'. Below these is a 'Cancel' button.

Obr. 13: Schvalování přijatý úkol.

Vyhledávání s fulltext podporou

Funkční požadavek kladený na integrovaný systém je vyhledávání dokumentů dle specifikace atributu nebo obsahu dokumentu. Implementace je řešena v Alfresco systému funkcí pokročilého vyhledávání.

Proces vyhledávání je složen z kroků:

1. Uživatel označí menu „Advance search“.

2. V obrazovce upřesní vyhledávací podmínky např. typ dokumentu, časové údaje nebo umístění dokumentu.
3. Spuštěním dotazu se v obrazovce načte výsledek vyhledávání, které lze uložit pro pozdější použití.

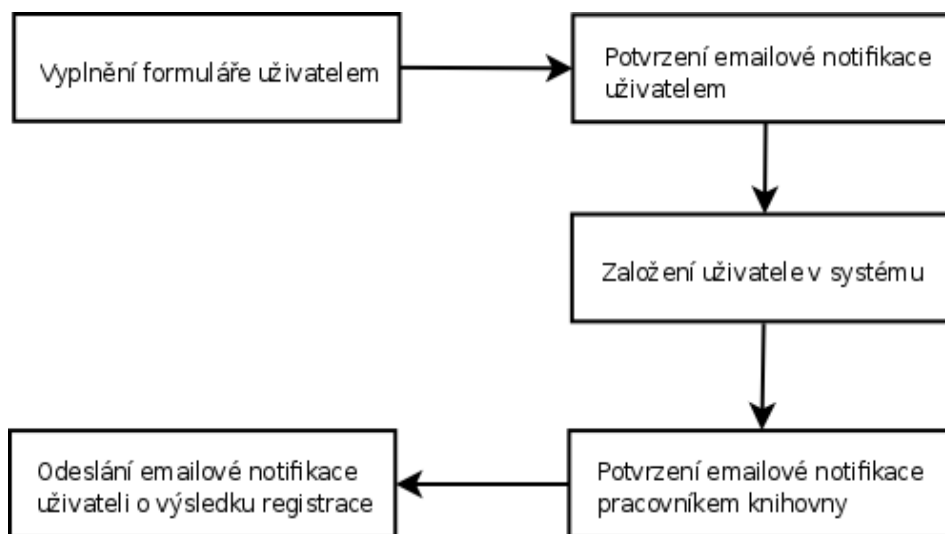
7.3 Implementace BPEL procesu

Implementace BPEL procesu slouží pro registraci nových uživatelů v online verzi webové aplikace. Během registrace se předpokládá, že uživatel se fyzicky nedostaví na pobočku knihovny. Z uvedených faktorů je patrné, že proces registrace uživatele nevyhovuje původně implementované funkcionalitě v desktop aplikaci, kde se jednalo o získání výsledku. Proces registrace je dlouhodobý jev se zavilostí např. verifikace uživatele emailem nebo finální schválení pracovníkem knihovny.

Online registrace je poměrně náročný proces na zpracování a významem se zcela odlišuje od předchozích analyzovaných procesů. Hlavní odlišnost spočívá v časovém horizontu zpracovávaného procesu. Předchozí procesy vždy řešily funkci typu požadavek-odpověď v řádu několika sekund nutných pro zpracování. Registrace je naopak dlouhodobý proces zahrnující interakci uživatelů a integrovaných systémů. Předpokládaná doba dokončení registrace se tak v běžném pracovním procesu odhaduje v řádu dnů či týdnů. Vstupem pro proces je vyplnění formuláře ve webové aplikaci, která ověří zadaná data a spustí instanci BPEL procesu pomocí technologie webové služby. V takovém případě je uživatel založen z příznakem nepotvrzující příslušnost k registrovaným uživatelům, jde o nepotvrzeného uživatele. V dalším kroku po uložení entity uživatele je vygenerována emailová notifikace informující uživatele o nutnosti potvrzení zadané URL pro dokončení registrace. BPEL proces tak v pravidelných časových intervalech kontroluje stav potvrzení klienta. Pokud uživatel přistoupí na základě zadané URL, servlet změní hodnotu stavu uživatele na stav Registrace. Dalším krokem je schválení pracovníkem knihovny pouze na základě zadaných hodnot zobrazených v desktop aplikaci. Pokud je uživatel úspěšně schválen, BPEL proces realizuje odeslání informačního emailu uživateli a ukončí instanci procesu. Hlavní devizou pro využití technologie BPEL a využití SOA architektury je nově vytvořená instance procesu perzistentně se uchovávajících. V případě výpadku systému tak nedochází ke ztrátě informací o právě probíhajícím procesu.[14]

Implementovaný proces využívá portované služby aplikačního serveru vyvinuté v rámci modulu prj.dipl_ee_app-ejb. Základní komunikační rozhraní obsahuje autentizační službu pro zabezpečení procesu pouze oprávněnými uživateli. Další služby implementující portaci emailové komponenty a vlastní implementace webové služby pro monitoring regis-

trovaného klienta. Původní objekt implementující logiku *CustomerSessionBean* objektu nemohl být využit z důvodu konfliktu s entitní třídou *CustomerEntity*, proto byl vytvořen stateless objekt implementující získání informací o klientovi.



Obr. 14: Proces registrace uživatele

Proces registrace je navržen s následujícími kroky:

1. Uživatel vyplní registrační formulář ve webové aplikaci,
2. Uživatel je vytvořen ve stavu „nepotvrzen“,
3. Data zadaná uživatelem budou transformována do emailové zprávy a odeslána uživateli s URL odkaz potvrzující registraci. Uživatel musí pro potvrzení otevřít ve webovém prohlížeči URL odkaz,
4. Data jsou následně zpracována ve middleware prostředí se stavem uživatele „registrace“,
5. Pracovník knihovny ověří zadaná data a buď schválí nebo zamítne registraci,
6. Výsledek registrace je zaslán uživateli emailovou notifikací.

V případě havárie během procesu registrace je definován chybový stav, který odešle zprávu administrátorovi o manuální registraci konkrétního uživatele.

Proces bude složen z BPEL komponent:

1. EJB komponenta registrace uživatele portována do webové služby
2. Portace emailové služby pomocí email-binding komponenty na ESB sběrnici

- Portace http protokolu pomocí http-binding nebyla využita z důvodu omezení POST metody. Bylo využito poměrně snadné řešení pomocí servlet implementace.

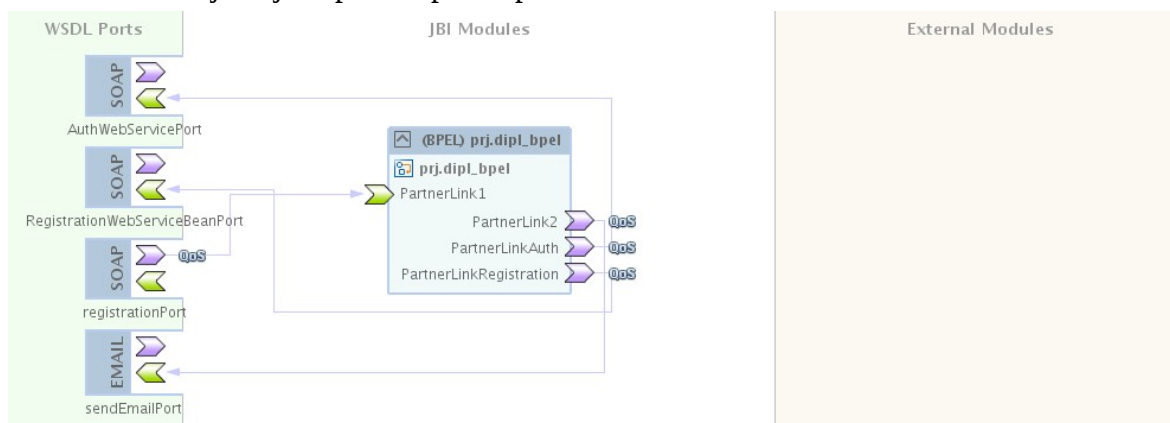
Mimo uvedené portace služeb GlassFish JBI dále nabízí funkce pro připojení do databáze, http, ftp protokolů, JMS a dalších.

BPEL diagram

BPEL diagram definuje postup kroků popsaných na Obr. 14. Na počátku procesu je příchozí požadavek na registraci nově uloženého uživatele obsahující identifikaci nově vytvořeného čtenáře. Proces dále sestaví informační email pro potvrzení validnosti zadaných dat od uživatele v emailové notifikaci. Střední část je řízena cyklem repeat-until dokud neproběhne počet povolených kroků registrace nebo uživatel nepřistoupí na vygenerovanou http adresu. Každý běh cyklu je uspána na 1 hodinu, následně dochází k otestování výše popsaných pravidel. V závěrečné části procesu je uživatel informovaný o stavu registrace registrován/registrace vypršela. Ukázka inicializace BPEL je zobrazena v přílohách Obr. 21, kontrola registrace uživatele je zobrazena v příloze Obr. 22.

CASA diagram

CASA diagram obsahuje informace o portovaných službách pro konkrétní BPEL procesy a slouží především pro tvorbu SOA aplikace snadno přenositelné na jiné prostředí, obdobné řešení jako je např. enterprise aplikace ear.



Obr. 15: CASA diagram

7.4 Implementace desktop aplikace

Jádro desktop aplikace byla rozšířena o novou funkcionalitu popsanou ve specifikaci 6.0.2. Implementovaná aplikace je založena na principu Model View Controller a grafické knihovně Swing, která byla použita již bakalářské práci.[11] Zdrojové kódy byly zrevi-
dovány a připraveny pro začlenění funkcionality integrovaného systému.

Aplikace v bakalářské práci poskytovala uživatelské funkce:

- Evidence čtenářů – evidence registrovaných čtenářů,
- Evidence knih – evidence katalogu knih,
- Výpůjčky – modul spravující výpůjčky knih.

Diplomová práce implementovala nové funkce integrovaného systému:

- Schválení uživatelských online registrací,
- Katalog knih,
- Fulltext vyhledávání v katalogu knih,
- Zobrazení dokumentů v externím prohlížeči.

7.4.1 Schválení uživatelských online registrací

Proces online schválení uživatele je spravován v implementovaném BPEL procesu. Proces je popsán v podkapitole 7.3. Pracovník na terminálu knihovny provádí změnu stavu ve vlastnostech uživatele ze stavu „Registrace“ na „Ověření“ až po finální schválení aktivního uživatele dle úředního dokladu (občanský průkaz, pas) uživatele s trvalým pobytem na území České republiky. Doposud není uživatel veden jako aktivní čtenář s možností realizace fyzického zapůjčení knih mimo knihovnu.

7.4.2 Katalog online knih

Katalog online knih byl pro prezentaci naplněn volně dostupnými knihami ze zdroje

<http://www.gutenberg.org>, nicméně návrh entity neomezuje využití elektronických knih od dalších zdrojů včetně komerčních elektronických knih. Záložka „Dokument knihovna“ poskytuje aktuální přehled o dostupných elektronických knihách v elektronickém úložišti. Informace o knihách jsou zobrazeny v tabulce s omezeným výčtem atributů. Detailní vlastnosti lze získat z rozšířeného dialogu vlastností při označení knihy. Data jsou získávána online komunikací pomocí technologie webových služeb. V případě nefunkčnosti úložiště není možné zobrazit informace o elektronických knihách. Referenční aplikace je zobrazena v příloze Obr. 26.

7.4.3 Fulltext vyhledávání v katalogu knih

Fulltext vyhledávání je podporováno u katalogu online knih. Na základě zadané hodnoty v filtrovacím dialogu jsou vyhledány objekty s identickými hodnotami atributů objektu nebo s obsahem elektronické reprezentace. Formáty indexovaných dat jsou plně závislé na Lucene engine, podporovány jsou formáty textového souboru, PDF, Office a další.

7.4.4 Zobrazení dokumentů v externím prohlížeči

Zobrazení v externím prohlížeči dále rozšiřuje funkcionalitu desktop aplikace o získání obsahu elektronické knihy uložené v úložišti. S požadavkem na začlenění externího prohlížeče bylo nutné rozšířit konfiguraci aplikace do externího textového souboru, aby nebylo nutné s každou změnou externího prohlížeče opětovně kompilovat Java třídu. Nově tak konfigurace obsahuje definici externích prohlížečů pro formáty uvedené v tabulce 7.

<i>Typ souboru</i>	<i>Typ obsahu souboru identifikován v Alfresco syst.</i>	<i>Nastavení aplikace</i>
Textový soubor	text/plain.	/usr/bin/gedit
PDF soubor	application/pdf	/usr/bin/evince
Office soubor	application/doc	/usr/lib/openoffice/program/soffice.bin
Ostatní soubory	ostatní	/usr/bin/gedit

Tabulka 7: Desktop aplikace - podporované formáty externího prohlížeče.

Implementace transformace byla otestována na využití RMI volání vzdálených metod. Uvedený postup není vhodný pro přenos textových či binárních dat. Proto musela být implementována další sada služeb poskytující zabezpečení *AuthWebService* a dále službu pro přenos souboru *ContentWebService* z balíku *cz.hoferek.alfbridge.ws*.

Grafické rozmístění prvků a návrh dialogových oken byl ponechán dle předchozího návrhu z důvodu využití již funkčního ovládacího rozhraní. Pokud není vznesen požadavek od uživatelů s návrhem na efektivní rozložení, není důvod zásadněji zasahovat do uživatelského rozhraní během vývoje integrované verze.

Komunikace s aplikačním serverem je založena na kombinaci volání vzdálených rozhraní EJB komponent a webové služby. V případě EJB byla ponechána lokální cache objektů v implementované třídě *cz.hoferek.bc.core.CacheServiceLocator*. Nastavení portu pro komunikaci s aplikačním serverem je uvedeno v vlastnostech spouštěného příkazu desktop aplikace.

```
-Djava.naming.provider.url=iiop://localhost:3700
```

Pro komunikaci pomocí technologie webových služeb je navázán modul *prj.alf_bridge* obsahující generované proxy objekty služby *cz.hoferek.alfbridge.ws.ContentWebService*. Pokud se změní lokace content webových služeb, je nutné vygenerovat nové proxy objekty a aktualizovat knihovnu *prj.alf_bridge* v Java classpath.

7.5 Implementace web aplikace

Webová aplikace slouží uživatelům pro získání informací o katalogu knihovny. V bakalářské verzi byly implementovány funkce rozdělené dle role uživatele. Pro neregistrované uživatele byla dostupná pouze informace o verzi aplikace a kontakt. Registrovaní uživatelé, kteří byli během registrace ověřeni na pobočce knihovny dle platného dokladu ve webové aplikaci naleznou informace o katalogu knih a realizovaných výpůjčkách. Diplomová práce rozšiřuje vlastnosti aplikace (viz. referenční aplikace Obr. 25) o následující funkce:

- Registrace uživatelů – registrace uživatelů realizuje proces založení uživatelského účtu a přístup do systému,
- Zobrazení elektronických knih – dostupnost informací o elektronických knihách a jejich obsazích.

Operace realizované nad prezenčními a elektronickými knihami:

- katalog prezenčních knih,
- informace o vypučených knihách uživatele,
- katalog elektronických knih,
- katalog univerzitních prací.

7.5.1 Registrace uživatelů

Registrace uživatelů provádí integrovaný proces založení uživatelského účtu ve spolupráci s BPEL procesem. Uživatel musí pro registraci vyplnit povinné platné údaje včetně emailové adresy a případně trvalé adresy bydliště. Proces je dále realizován mimo webové rozhraní aplikace BPEL procesem. BPEL proces nadále spravuje registraci až do ukončení úspěšné nebo nedokončené registrace. Pro ověření validity uživatele byla využita metoda potvrzení emailové notifikace. Uživatel tak po úspěšné registraci je informován emailovou notifikací s nutností přístupu na URL potvrzující zadané údaje a platnost vyžádané registrace. Proces je detailněji popsán v podkapitole 7.3.

7.5.2 Zobrazení elektronických knih

Zobrazení knih je předpokládána jako nejčastěji využívaná funkce přístupujících uživatelů. Pro tento účel byla implementovaná sada služeb na straně aplikačního serveru identicky využívána v Desktop aplikaci. Proces transferu dat je technicky popsán v podkapitole 7.4.4. V případě webové aplikace byl implementován servlet *cz.hoferek.web.servlet.DownloadServlet* zajišťující přenos dat v webovém prostředí.

7.6 Implementace importní aplikace

Importní aplikace byla vytvořena pro získání dostatečného množství elektronických dokumentů. Prvotní implementace pomocí sdílené jednotky postrádá rozšířené informace o entitách objektu. Po vzájemné dohodě s panem profesorem Pokorným byla vytvořena zdrojová báze dat. Import dat je vždy realizován na veškeré data umístěné v importním adresáři. Import neobsahuje aktualizaci dat s vedením záznamu o právě importovaném objektu. Import se skládá z následujících prvků:

- příkaz wget sestaví základní informace o struktuře do importního adresáře,
- Java aplikace následně prochází exportována díla uložená v importním adresáři na pevném disku,
- informace o attributech jsou získány pomocí regulárních výrazů z lokálního souboru, obsah elektronického dokumentu je načten z webového serveru <http://www.gutenberg.org> pomocí http protokolu,
- sestavený obsah je následně uložen do centrálního úložiště dokumentů pod složkou elektronických online knih.

Kombinace wget příkazu byla využita pro předpoklad využití XSL transformace dat o knihách, protože každý dotaz na http a vyhodnocení dat pomocí regulárního výrazu je náročný na přenos dat a především z časového hlediska. Proto byl uplatněn přístup získání základní informací o knihách pomocí příkazu:

```
wget -r -l 3 -k http://www.gutenberg.org/browse/languages/cs
```

Data uložená v externím adresáři jsou importní utilitou iterativně procházena, na každý soubor je aplikován regulární dotaz s vyhodnocením získaných dat. Pokud validace extrahovaných dat je úspěšná, získaná data obsahují informaci o obsahu knihy. Data jsou následně transformována do typu elektronické knihy a webovou službou importována do repository.

Dávka pro spuštění importní dávky:

```
$JAVA_HOME/bin/java cz.hoferek.dipl.imports.prjalfimportapp.Main
```

ZÁVĚR

Diplomová práce se zabývá reálnou implementací integrovaného systému v prostředí univerzitní knihovny. Od úvodu do problematiky je dále zpracována rešerše na použité softwarové platformy. V úvodní části je souhrnně zpracována platforma Java EE s odkazem na již detailněji zpracované téma v bakalářské práci. Následující kapitola popisuje teoretickou problematiku integrace systémů v doporučených postupech pro jednotlivé případy užití. Závěrečná kapitola teoretické části diplomové práce je věnována elektronickému úložišti Alfresco. Zpracování elektronických dokumentů je rozděleno na různé odvětví, které se vyskytují v enterprise prostředích. Dále jsou uvedeny vlastnosti a funkce systému.

Původní systém byl analyzován a nasazen do nového prostředí aplikačního serveru GlassFish 2.1, následně byla provedena migrace dat. Systém byl vydefinován ve funkčních bodech určující hranice a popsán diagramem případu užití. Implementační část obsahovala implementaci desktop, web aplikace a BPEL procesu. Podprojekty integrovaného systému byly popsány a doplněny diagramy tříd. Samostatná kapitola byla věnována BPEL procesu, jež zpracovává uživatelské registrace včetně grafické reprezentace během návrhu. Integrace byla provedena s webovými službami knihovního systému a portací BPEL email komponenty. Původní záměr využít http komponentu se neosvědčil z důvodu omezení funkcionality zpracování požadavků.

Od předchozí verze byla dotažena výkonost aplikačního serveru GlassFish, porovnávali-li výsledky se zkušenostmi s Sun App Server 7 v bakalářské práci. Migrace aplikace byla provedena v krátkém časovém horizontu a celkově lze hodnotit kladně opětovné použití již implementovaných komponent. Fakticky je GlassFish následovníkem ve vývoji aplikačního serveru řady Sun App, pokud by aplikace byla migrována na aplikační server JBOSS, lze předpokládat změny v konfiguracích komponent a celkově náročnější proces nasazení. Rozšířený JBI modul v aplikačním serveru GlassFish byl v roce 2009 omezen funkcionalitou pro implementace procesů a např. vývojem WS-handler projektu pro generování bezpečnostních tokenů. Nasazením aktualizovaného GlassFish bylo dosaženo požadovaných aktualizací a stability systému.

Integrace systému byla úspěšně dokončena a otestována na referenčních aplikacích v podobě desktop a web aplikace. Otevřeným bodem zůstal transfer binárních souborů a souborů s velikostí přesahující 2MB. I přes změnu kontextu a konfiguraci doporučení se nepodařilo vyladit optimální přenos. Další rozšíření systému je nadále možné díky platformě Java EE a nesporným výhodám, pomineme-li náročnost na hardware a náklady.

CONCLUSION

The main purpose of my work was to implement integrated library system. The introduction and common description of this elaboration continues with specification of Java EE in relation to chapter, which was described in bachelor thesis. Next chapter describes theoretical part of system's integration and common best patterns to use each type of integration. The last chapter is focused on ECM system Alfresco, because is used as main central repository in integration's architecture.

The previous release of bachelor's applications was analyzed and deployed to the new application server specification GlassFish 2.1, then followed data migration. The system has been defined in specification and described in use case diagram.

Implemented part contains creation of desktop application, web application and BPEL process. The sub projects of integrated system was specified in implementation chapter and shown in the class diagrams. BPEL process was described in a separately part of this thesis, which is used in customer registration process with a graphic visualization during the implementation. Integration was realized as connection between web services of library system and BPEL email component. Planned solution was not used due to http component request limitations.

Current GlassFish server reached better benchmark results then the previous application server with Sun App Server 7 specification, which was used in bachelor's thesis. Deployment of previous implemented applications were successful and ready in short time period. The main reason of easy deploying is the fact, that was use next specification of previous application server Sun App. If the application server was different e.g. JBOSS solution, then more time consuming application migration could be expected. Extension of JBI module was limited in year 2009 by provided functionality e.g. WS handler implementation and security token generation. The new GlassFish server specification fixed this issues and increased stability of the system.

Desktop and web applications tested and proofed that the server implementation was working, thus project was successful. The open issue is transfer of binary data and larger file then 2 MB, because recommended solution by set web service context does not work. Implemented system could be extended due to get all Java EE platform benefits, leaving aside time complexity and performance for hardware.

SEZNAM POUŽITÉ LITERATURY

- [1] MARINESCU, Floyd. *EJB Design Patterns*. New York: John Wiley & Sons, Inc., 2002. ISB 0-471-20831-0.
- [2] ASHMORE, Derek C. *The J2EE Architects Handbook*. Lombard : DVT Press, 2004. ISB 0972954899.
- [3] SRIGANESH, Rima Patel, BROSE, Gerald, SILVERMAN, Micah. *Mastering Enterprise JavaBeans 3.0*. Indianapolis : Wiley Publishing, 2006. ISBN 0-471-7854105.
- [4] AMITA, Bhandari, MUNWAR, Shariff, PALLIKA, Majumdar. *Alfresco 3 Enterprise Content Management Implementation*. [s.l.] : Packt Publishing , 2009. 600 s. ISBN 9781847197368.
- [5] POTTS, Jeff. *Alfresco Developer Guide*. [s.l.] : Packt Publishing, 2008. 556 s. ISBN 9781847193117.
- [6] R. HEFFELFINGER, David. *Java EE 5 Development using GlassFish Application Server*. [s.l.] : Packt Publishing, 2007. 424 s. ISBN 9781847192608.
- [7] MATJAZ, B., Juric, BENNY, Mathew , POORNACHANDRA, Sarang. *Business Process Execution Language for Web Services*. [s.l.] : Packt Publishing, 2006. 372 s. ISBN 9781904811817.
- [8] JENNINGS, Frank, SALTER, David. *Building SOA-Based Composite Applications Using NetBeans IDE 6*. [s.l.] : Packt Publishing, 2008. 301 s. ISBN 9781847192622.
- [9] CABRERA, F., Luis, KURT, Chris. *Web Services Architecture and Its Specifications: Essentials for Understanding WS-**. [s.l.] : Microsoft Press, 2005. 174 s. ISBN 9780735621626.
- [10] *Alfresco Community Edition 3.2r* [online]. 2009 [cit. 2010-01-21]. Dostupný z WWW: http://wiki.alfresco.com/wiki/Alfresco_Community_Edition_3.2r2.
- [11] HOFÉREK, Tomáš. *Distribuovaná aplikace založená na J2EE* [online]. Zlín, 2007. 57 s. Bakalářská práce. Univerzita Tomáše Bati ve Zlíně.
- [12] CHRISTUDAS, Binildas A. *Java Business Integration Service-Oriented Java Business Integration : Enterprise Service Bus Integration Solutions for Java Developers*. [s.l.] : Packt Publishing, 2008. 350 s. ISBN 9781847194404.
- [13] JURIC, Matjaz B; PANT, Kapil. *Business Process Driven SOA Using BPMN and BPEL*. [s.l.] : Packt Publishing, 2008. 312 s. ISBN 9781847191465.

- [14] JURIC, Matjaz, B. *SOA Approach to Integration : XML, Web Services, ESB, and BPEL in Real-World SOA Projects*. [s.l.] : Packt Publishing, 2007. 384 s. ISBN 1904811175.
- [15] ROSEN, Mike; LUBLINSKY, Boris; SMITH, Kevin T. *Realizing the Promise of SOA Applied SOA : Service-Oriented Architecture and Design Strategies*. [s.l.] : John Wiley & Sons, 2008. 696 s. ISBN 0470223650.
- [16] TIŠNOVSKÝ, Pavel. *Root.cz* [online]. 2005 [cit. 2010-06-01]. Nástroje pro tvorbu UML diagramů. Dostupné z WWW: <<http://www.root.cz/clanky/nastroje-pro-tvorbu-uml-diagramu/>>.
- [17] STEIN, René. *Interval.cz* [online]. 2004 [cit. 2010-06-01]. Návrh aplikací v jazyce UML - složitější diagram případů užití. Dostupné z WWW: <<http://interval.cz/clanky/navrh-aplikaci-v-jazyce-uml-slozitejsi-diagram-pripadu-uziti/>>.
- [18] *Debian.com* [online]. 2010 [cit. 2010-06-01]. Debian. Dostupné z WWW: <<http://www.debian.com/>>.
- [19] *OpenESB* [online]. 2010 [cit. 2010-06-01]. The Open Source ESB for SOA & Integration. Dostupné z WWW: <<https://open-esb.dev.java.net/>>.
- [20] *OpenESB* [online]. 2010 [cit. 2010-06-01]. Netbeans ESB. Dostupné z WWW: <<https://open-esb.dev.java.net/Downloads.html>>.
- [21] *Tigris.org* [online]. 2010 [cit. 2010-06-01]. ArgoUML. Dostupné z WWW: <<http://argouml.tigris.org/>>.
- [22] *Mercurial.selenic.com* [online]. 2010 [cit. 2010-06-01]. Mercurial. Dostupné z WWW: <<http://mercurial.selenic.com/>>.
- [23] *Projects.gnome.org* [online]. 2010 [cit. 2010-06-01]. Dia. Dostupné z WWW: <<http://projects.gnome.org/dia/>>.
- [24] *Virtualbox.org* [online]. 2010 [cit. 2010-06-01]. Virtualbox. Dostupné z WWW: <<http://www.virtualbox.org/>>.
- [25] *Gimp.org* [online]. 2010 [cit. 2010-06-01]. GIMP. Dostupné z WWW: <<http://www.gimp.org/>>.
- [26] *BPM prakticky* [online]. 2010 [cit. 2010-06-02]. Tvorba BPEL modulu. Dostupné z WWW: <<http://bpm-sme.blogspot.com/2008/04/5-tvorba-bpel-modulu.html>>.

[27] *Javaworld.com* [online]. 2010 [cit. 2010-06-02]. Use JBI components for integration. Dostupné z WWW: <<http://www.javaworld.com/javaworld/jw-07-2006/jw-0717-jbi.html>>.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

EJB	Enterprise Java Beans – serverové komponenty.
BPEL	Business Process Executional Language – jazyk na bázi XML sloužící pro zápis business procesů.
ECM	Enterprise Content Management – systém spravující dokumenty.
RMI	Remote Method Invocation – protokol pro vzdálené volání metod.
HTTP	HyperText Transfer Protocol – protokol použitý webovými servery a klienty pro odesílání požadavků a tvorbu odezvy.
IIOP	Internet Inter-ORB Protocol – protokol technologie CORBA, který se v J2EE používá v kombinaci s RMI, díky vlastnostem lze přistupovat k externím CORBA objektům. Další vlastností je možnost přístupu na Java komponenty z nejavové aplikace.
JB1	Java Business Integration – integrační standard.
Lifecycle	Životní cyklus definující stavy dokumentu.
Servlets	Java třídy generující dynamický obsah stránek.
JSP	JavaServer Pages – HTML stránky používající jazyk Java pro generování dynamického obsahu .
JNDI	Java Naming and Directory Interface – jednotný způsob práce s jmennými adresářovými službami .
WS	Web Service – komunikační standard.
Workflow	Šablona definující proces v ECM systému.

SEZNAM OBRÁZKŮ

Obr. 1: Schéma architektury.....	14
Obr. 2: Datová integrace.....	22
Obr. 3: Aplikační integrace sběrnicí ESB.....	22
Obr. 4: Business proces (BPEL) integrace.....	23
Obr. 5: Integrace zpracování zpráv (MOM).....	25
Obr. 6: Alfresco schéma architektury. Zdroj:[4].....	31
Obr. 7: Schéma integrovaných systémů.....	56
Obr. 8: Diagram nasazení.....	58
Obr. 9: Diagram případu užití.....	62
Obr. 10: Datový model knihovny.....	67
Obr. 11: Inicializace schvalování dokumentu.....	73
Obr. 12: Vlastnosti schvalování dokumentu.....	74
Obr. 13: Schvalování přijatý úkol.....	75
Obr. 14: Proces registrace uživatele.....	77
Obr. 15: CASA digram.....	78
Obr. 16: Adresářová struktura.....	92
Obr. 17: Vlastnosti složky.....	93
Obr. 18: Vlastnosti souborů.....	94
Obr. 19: Vyhledávání knih.....	95
Obr. 20: Získání obsahu.....	96
Obr. 21: BPEL úvod procesu registrace.....	97
Obr. 22: BPEL ověření registrace uživatele.....	97
Obr. 23: Projekt core (část diagramu).....	98
Obr. 24: Projekt EJB (část webových služeb).....	98
Obr. 25: Webová aplikace.....	99
Obr. 26: Desktop aplikace.....	99

SEZNAM TABULEK

Tabulka 1: Definice typu atributů.....	37
Tabulka 2: Definice práv pro kabinety a adresáře.....	40
Tabulka 3: Definice práv pro soubory.....	40
Tabulka 4: Definice rolí.....	41
Tabulka 5: Konfigurace aplikačního serveru.....	57
Tabulka 6: Konfigurace Alfresco serveru.....	57
Tabulka 7: Desktop aplikace - podporované formáty externího prohlížeče.....	80

SEZNAM PŘÍLOH

Příloha P I: Alfresco – adresářová struktura

Příloha P II: Alfresco – vlastnosti složky

Příloha P III: Alfresco – vlastnosti souborů

Příloha P IV: Vyhledávání knih

Příloha P V: Získání obsahu

Příloha P VI: Inicializace BPEL procesu

Příloha P VII: Diagram tříd

Příloha P VIII: Referenční aplikace



















PŘÍLOHA P I: ALFRESCO – ADRESÁŘOVÁ STRUKTURA


Company Home > Libraries > www.gutenberg.org

www.gutenberg.org
This view allows you to browse the items in this space.

(0) Add Content Create More Actions Icon View


▼ **Browse Spaces** Items Per Page 9

 Macaulay, Thomas Babington Macaulay, Baron, 1800-1859 - The History of England, from the Accession of James II, Volume 1, Chapter 04 ⓘ 9 January 2010 11:39 	 Macaulay, Thomas Babington Macaulay, Baron, 1800-1859 - Vierter Band ⓘ 9 January 2010 11:38 	 MacBeth, R. G. (Roderick George), 1858-1934 - Being the Real-Life Record of the Famous North-West Mounted Police ⓘ 9 January 2010 11:41 
 MacCarthy, Denis Florence, 1817-1882 - Poems ⓘ 9 January 2010 11:41 	 MacCauley, Clay, 1843-1925 - Government Printing Office, Washington, 1887, pages 469-532 ⓘ 9 January 2010 11:42 	 MacCulloch, J. A., 1868-1950 - The Religion of the Ancient Celts ⓘ 9 January 2010 11:42 
 MacDonald, George, 1824-1905 - A Book of Strife in the Form of The Diary of an Old Soul ⓘ 9 January 2010 11:43 	 MacDonald, George, 1824-1905 - A Hidden Life and Other Poems ⓘ 9 January 2010 11:44 	 MacDonald, George, 1824-1905 - A Study with the Text of the Folio of 1623 ⓘ 9 January 2010 11:47 

Page 7 of 30 

▼ **Content Items** Items Per Page 9






No items to display. To add an existing document click 'Add Content' action. To create an HTML or Plain Text file click 'Create Content' action.

Page 1 of 1 

Obr. 16: Adresářová struktura

PŘÍLOHA P II: ALFRESCO – VLASTNOSTI SLOŽKY


Company Home > Libraries > www.gutenberg.org > Cable, George Washington, 1844-1925 - The Flower of the Chapdelaines

Cable, George Washington, 1844-1925 - The Flower of the Chapdelaines   (0)  Add Content Create  More Actions  Icon View 



This view allows you to browse the items in this space.

▼ **Browse Spaces** Items Per Page




No items to display. Click the 'Create Space' action to create a space.


Page of 1 

▼ **Content Items** Items Per Page

 **The Flower of the Chapdelaines** 

325.82 KB
9 January 2010 11:24

Page of 1 

Obr. 17: Vlastnosti složky

PŘÍLOHA P III: ALFRESCO – VLASTNOSTI SOUBORŮ


▼ Custom View 📄 🗑️

Apply Template

▼ Links

View In Browser	View in WebDAV	View in CIFS
Download Content	Details Page URL	Alfresco Node Reference
















▼ Properties 📄

	Creator: admin
	Created Date: 9 January 2010 11:24
	Modifier: admin
	Modified Date: 9 January 2010 11:24
	Content Type: Plain Text
	Size: 325.82 KB
	author: Cable, George Washington, 1844-1925
	publisher:
	subject:
	year: 2005
	locnr:
	language: English
	type: 15881
	abstract:
	keywords:
	copyright: Not copyrighted in the United States. If you live elsewhere check the laws of your country before downloading this ebook.
	pages:

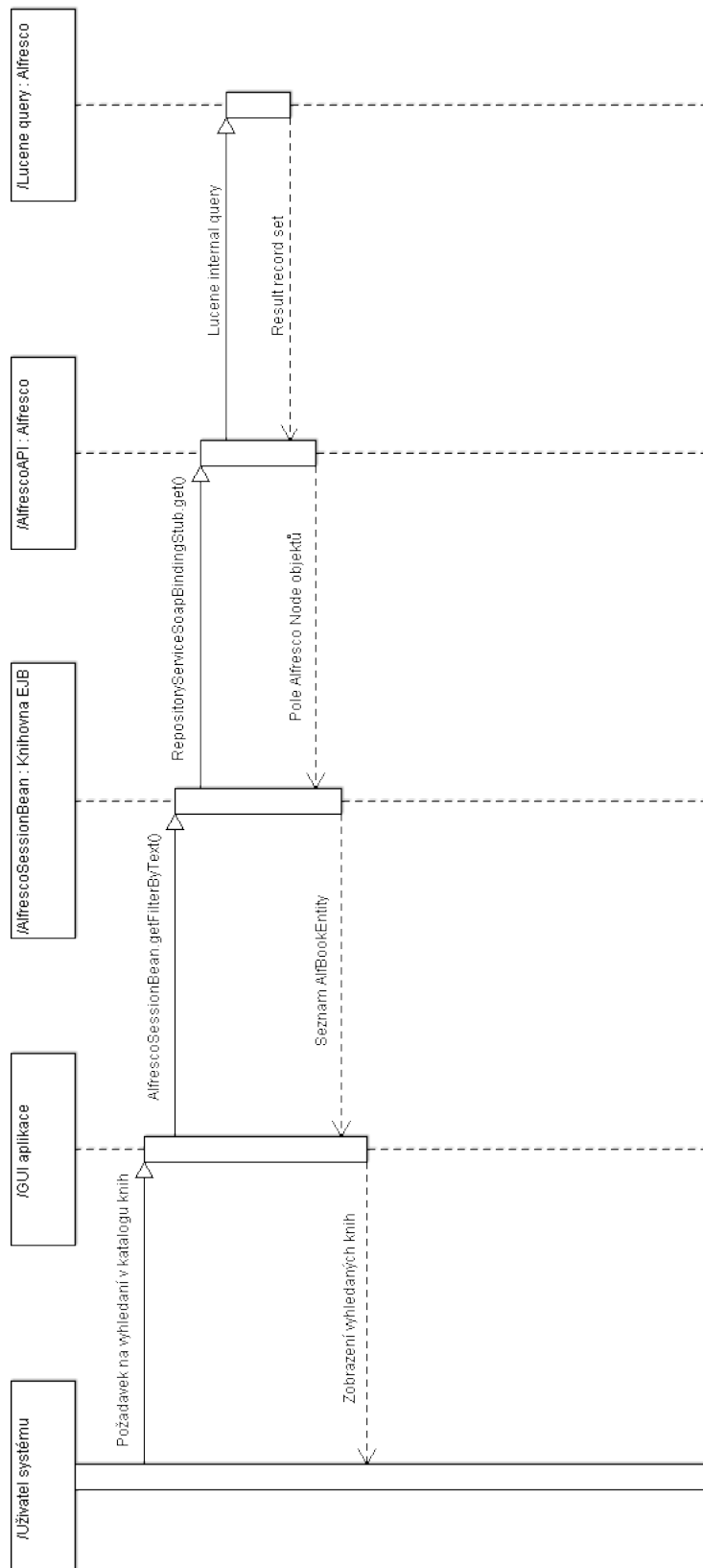
This document is not inline editable.

[Allow Inline Editing](#)

▼ Actions

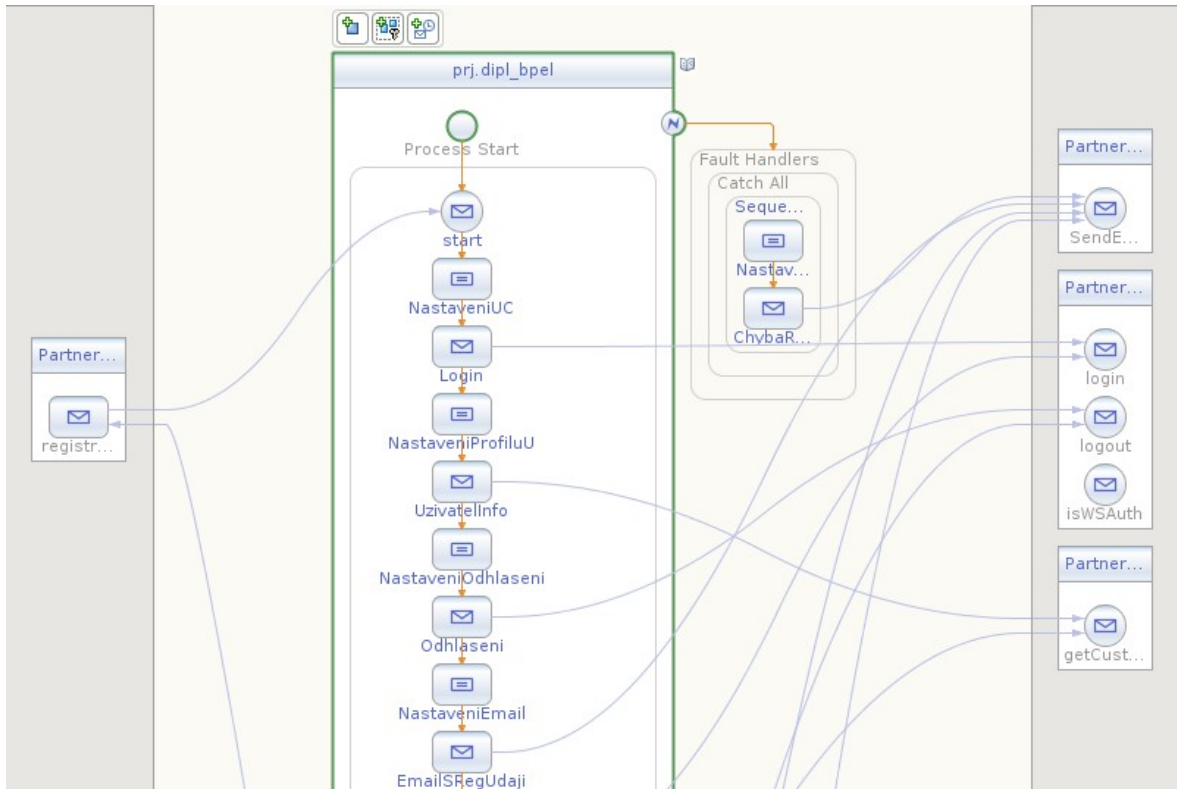
-  [Edit Offline](#)
-  [Check Out](#)
-  [Download](#)
-  [Update](#)
-  [Cut](#)
-  [Copy](#)
-  [Delete](#)
-  [Take Ownership](#)
-  [Manage Content Users](#)
-  [Create Shortcut](#)
-  [Preview in Template](#)
-  [Run Action](#)
-  [Make Multilingual](#)
-  [Start Discussion](#)
-  [Start Advanced Workflow](#)

Obr. 18: Vlastnosti souborů

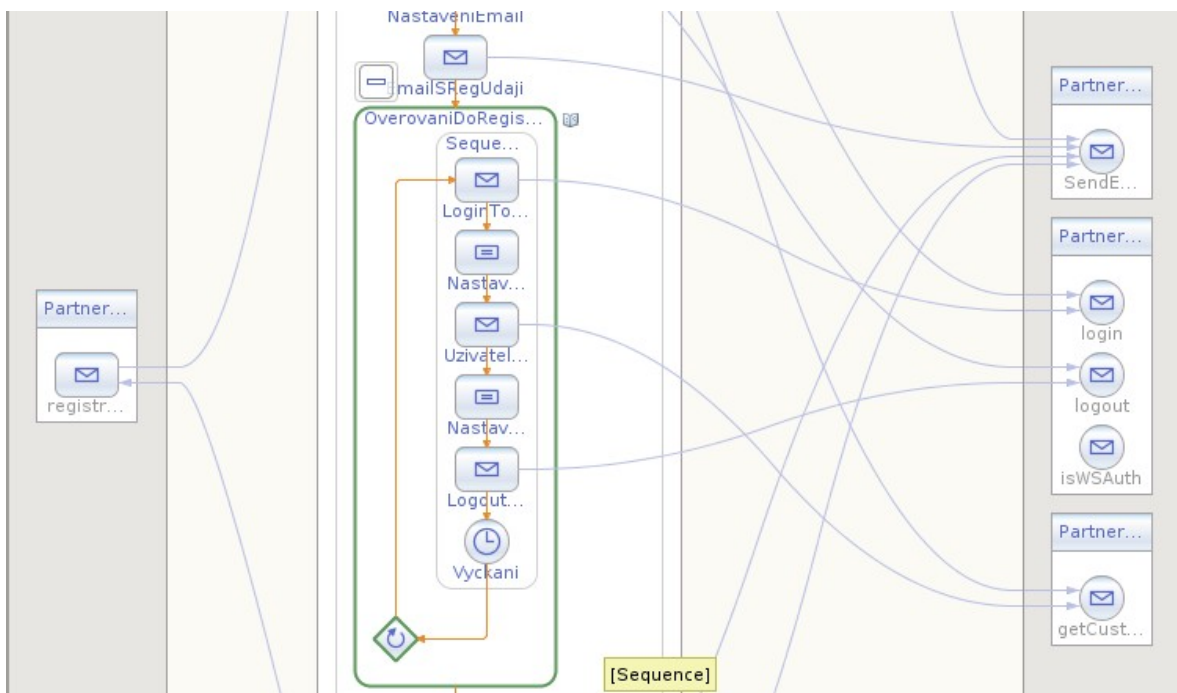


Obr. 19: Vyhledávání knih

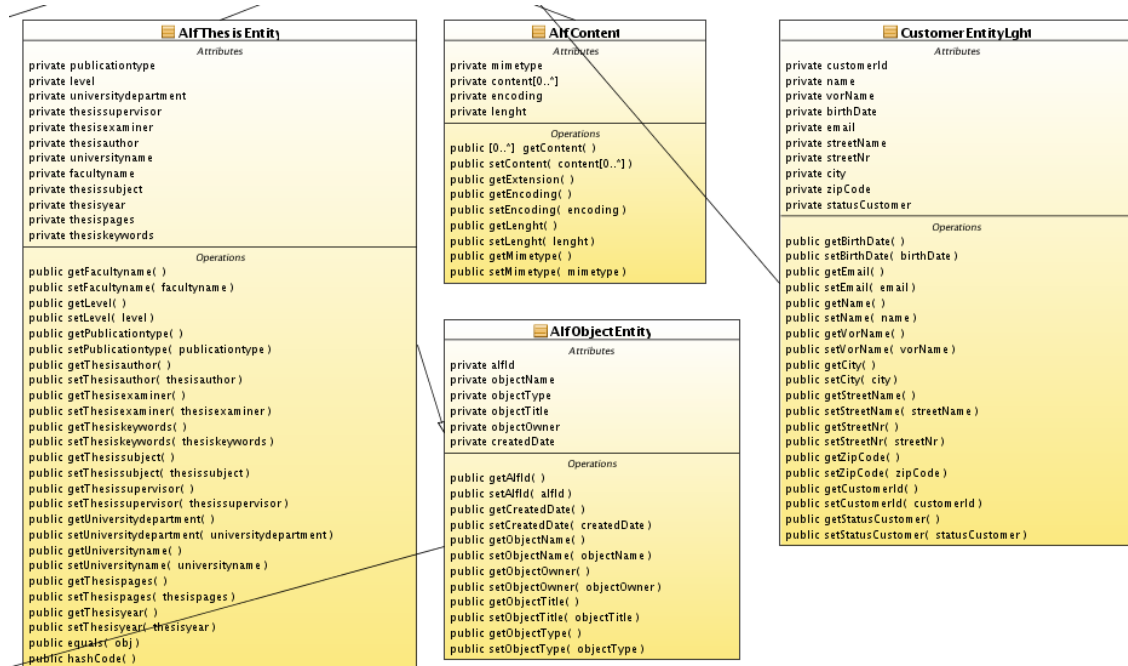
PŘÍLOHA P VI: INICIALIZACE BPEL PROCESU



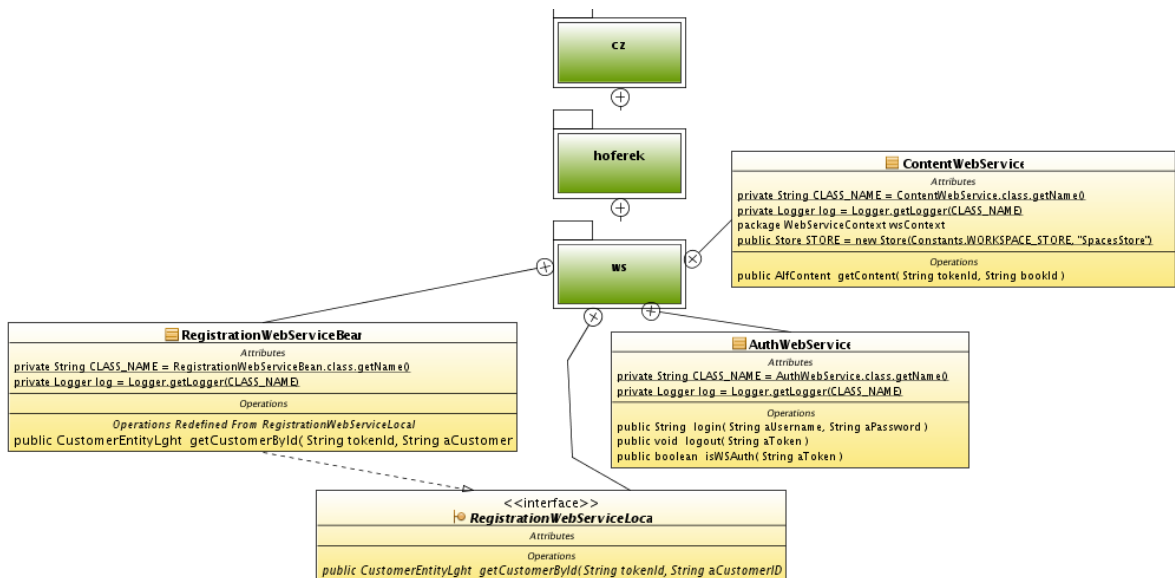
Obr. 21: BPEL úvod procesu registrace



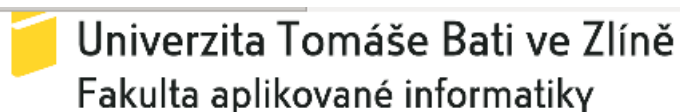
Obr. 22: BPEL ověření registrace uživatele



Obr. 23: Projekt core (část diagramu)



Obr. 24: Projekt EJB (část webových služeb)



Katalog elektronických knih

[Zpět](#)

Text:

BookId	Název	Autor	Title	Popis	Rok	Zobrazit
efc2e9e9-2fd7-42c4-afdd-71bc660416ea	Unspecialist	Yaco, Murray F.	Unspecialist		2007	<input type="button" value="Zobrazit"/>
1bb5e016-560f-4641-84c7-1d64413870b0	No Moving Parts	Yaco, Murray F.	No Moving Parts	Science fiction	2008	<input type="button" value="Zobrazit"/>
d6137130-81da-4a1d-8505-41ba447da584	The Travels of Marco Polo — Volume 1	Polo, Marco, 1254-1324	The Travels of Marco Polo — Volume 1		2004	<input type="button" value="Zobrazit"/>
4c95e786-579e-48d5-8781-eb29e3944862	The End of the Rainbow	MacGregor, Mary Esther Miller, 1876-1961	The End of the Rainbow		2009	<input type="button" value="Zobrazit"/>
af3566c5-698d-4b04-8e6d-b57081413114	The Book of Old English Ballads	Edwards, George Wharton, 1859-1950	The Book of Old English Ballads	Ballads, English -- England -- Texts	2005	<input type="button" value="Zobrazit"/>
e46748ca-b9ca-4b1d-94f2-e8a877aaaad3	Colonial Reports, Miscellaneous. No. 54. Newfoundland	MacGregor, William, 1846-1919	Colonial Reports, Miscellaneous. No. 54. Newfoundland	Micmac Indians	2006	<input type="button" value="Zobrazit"/>

Obr. 25: Webová aplikace

Název	Autor	Title	Popis	Rok
Unspecialist	Yaco, Murray F.	Unspecialist		2007
No Moving Parts	Yaco, Murray F.	No Moving Parts	Science fiction	2008
The Travels of Marc...	Polo, Marco, 1254-...	The Travels of Marc...		2004
The End of the Rain...	MacGregor, Mary Est...	The End of the Rain...		2009
The Book of Old Engl...	Edwards, George Wh...	The Book of Old Engl...	Ballads, English -- E...	2005
Colonial Reports, Mis...	MacGregor, William, ...	Colonial Reports, Mis...	Micmac Indians	2006
Or, The Story of the ...	Yechton, Barbara, [p...	Or, The Story of the ...		2006
A Tragedy	Young, Edward, 168...	A Tragedy	Tragedies	2009
or, The Quest for Na...	Youngusband, Fran...	or, The Quest for Na...	Natural history -- Hi...	2008
Library of the World'...		Library of the World'...	Literature -- Collecti...	2004
Library of the World'...		Library of the World'...	Literature -- Collecti...	2004
a2354e62-8a57-4f...	Various			2005
Being the Real-Life R...	MacBeth, R. G. (Rod...	Being the Real-Life R...	Northwest, Canadian...	2007
The Closet of Sir Ken...	Digby, Kenelm, 160...	The Closet of Sir Ken...	Liquors	2005
Who Left Their Nativ...	Dunderdale, George...	Who Left Their Nativ...	Frontier and pioneer...	2005
The Watchman of Gl...	MacGregor, Mary Est...	The Watchman of Gl...		2009

Filtr

Data načteny ... Stav synchronizace: STAV_SYNC

Obr. 26: Desktop aplikace