

Masarykova Univerzita
Fakulta Informatiky



Diplomová práce

Hostování business procesu

Vypracoval: Bc. Radim Drgáč
Vedoucí práce: RNDr. Jan Pavlovič

Brno, 2009

Prohlášení

Prohlašuji, že tato práce je mým původním autorským dílem, které jsem vypracoval samostatně. Všechny zdroje, prameny a literaturu, které jsem při vypracování používal nebo z nich čerpal, v práci řádně cituji s uvedením úplného odkazu na příslušný zdroj.

Poděkování

Chtěl bych poděkovat vedoucímu mé práce panu RNDr. Janovi Pavlovičovi za jeho rady a připomínky. Dále pak panu Branislavovi Hudcovi z IBM, se kterým jsem konzultoval problematiku hostování procesu.

Shrnutí

Obsahem této práce je popsat problematiku business proces managementu a jeho namapování na informační technologie. Navrhnout a implementovat jeden business proces a popsat problematiku hostování procesu.

Klíčová slova

Business proces, BPEL, BPMN, SOA, Hostování procesu, BMPS, BPM.

Obsah

I 1 ÚVOD	6
II2 BUSINESS PROCES MANAGEMENT.....	7
2.1 BUSINESS MODEL.....	7
2.2 PROCESNĚ ORIENTOVANÝ PŘÍSTUP.....	7
2.2.1 PROCES.....	7
2.2.2 DEFINICE PROCESU.....	7
2.2.3 INSTANCE PROCESU.....	8
2.2.4 AKTIVITA, ÚKOL.....	8
2.3 MOTIVACE PRO POUŽITÍ PROCESNĚ ORIENTOVANÉHO PŘÍSTUPU.....	8
2.4 ŽIVOTNÍ CYKLUS PROCESU.....	8
2.4.1 ANALÝZA A NÁVRH PROCESU („MODEL“).	9
2.4.2 IMPLEMENTACE PROCESU („ASSEMBLE“).	9
2.4.3 NAsAZENÍ A BĚH PROCESU („DEPLOY“).	9
2.4.4 ADMINISTRACE A MONITORING PROCESU („MANAGE“)	
	9
2.5 MODEL ZRÁNÍ PROCESŮ („PROCESS MATURITY MODEL“).	10
2.6 BPMS (BUSINESS PROCES MANAGEMENT SYSTÉM).....	11
III3 TECHNICKÉ ASPEKTY BPM.....	12
3.1 PLATFORMA WEBOVÝCH SLUŽEB („WEBSERVICES“).	12
3.2 ESB („ENTERPRISE SERVICE BUS“).	12
3.3 SOA („SERVICE ORIENTED ARCHITECTURE“).	13
IV TECHNICKÝ DESIGN PROCESU „SLUŽEBNÍ CESTA „.....	17
3.4 POPIS PROCESU V PŘIROZENÉM JAZYCE.....	17
3.5 UŽIVATELSKÉ ROLE VYSKYTUJÍCÍ SE V PROCESU.....	17
3.6 UŽIVATELSKÉ ROZHRANÍ.....	18
3.7 DATOVÝ MODEL PROCESU.....	18
3.7.1 POPIS DATOVÝCH ENTIT A JEJICH ATRIBUTŮ:.....	19
3.7.2 MODIFYCOMMENT.....	20
3.7.3 TRIPDETAIL.....	20
3.7.4 TRIPDETAILITEM.....	20
3.8 BUSINESS PROCES DIAGRAM.....	21
V4 REALIZACE BUSINESS PROCESU „SLUŽEBNÍ CESTA“.....	27
4.1 IMPLEMENTACE PROCESU SLUŽEBNÍ CESTA	
.....	27

4.2 IMPLEMENTACE KLIENTSKÉ WEBOVÉ APLIKACE	27
.....	
4.3 NAsAZENÍ NA PROCESNÍ SERVER.....	28
VI5 HOSTOVÁNÍ BUSINESS PROCESU.....	29
5.1 POPIS CÍLE DIPLOMOVÉ PRÁCE V OBLASTI HOSTOVÁNÍ PROCESU	
.....	29
5.2 ARCHITEKTURA BPMS.....	29
5.3 DEFINICE POJMU „HOSTOVÁNÍ PROCESU“.....	31
5.4 DATOVÁ VRSTVA („DATA ACCESS LAYER“.....)	32
5.5 VRSTVA APLIKAČNÍ LOGIKY („BUSINESS LOGIC LAYER“.....)	32
5.6 VRSTVA PREZENTAČNÍ LOGIKY („PREZENTATION LOGIC LAYER“.....)	33
5.7 IBM BPMS KOMPONENTY.....	33
VII6 ZÁVĚR.....	35
VIII7 LITERATURA.....	36
IX8 PŘÍLOHY.....	37

1 ÚVOD

Úvodní kapitola „Business Proces Management“ mé diplomové práce pojednává obecně o business proces managementu. Definuji pojem business proces management, business model organizace. Business proces management je založen na procesně orientovaném přístupu. Zabývám se výhodami tohoto přístupu, popisuji definici procesu, instance procesu, úkolu a aktivity. Business procesy mají svůj životní cyklus. Popisuji jednotlivé fáze tohoto cyklu. Stručně zmiňuji model zrání procesu a systémy podporující business proces management.

Kapitola „Technické aspekty BPM“ popisuje namapování business proces managementu na informační technologie. Uvádím základní obecné informace o platformě webových služeb, sběrnici služeb ESB. Detailněji popisuji SOA architekturu. Uvádím hlavní rysy, popisují její jednotlivé vrstvy a na závěr kapitoly přikládám schéma s komentářem SOA architektury implementované pomocí webových služeb.

V kapitole „Technický design procesu služební cesta“ rozepisují jednotlivé části návrhu procesu služební cesty. Uvádím a komentuji datový model, procesu. Popisují jednotlivé entity datového modelu včetně jejich atributů. Definuji uživatelské role a způsob interakce uživatelů s procesem. Uvádím a popisují procesní BPMN diagram.

V kapitole „Realizace business procesu služební cesta“ stručně shrnuji implementační fázi procesu služební cesty. Dále shrnuji implementaci webové klientské aplikace. Rovněž uvádím důvody, proč nebyla některá funkcionalita plně naimplementována. Uvádím kroky nutné k nasazení procesu na běhové prostředí.

V Kapitole „Hostování business procesu“ se zabývám možnostmi hostování procesů. Uvádím definici „hostování procesu“. Popisuji možné komponenty BPMS. Procházím jednotlivé vrstvy architektury BPMS a hledám možnosti pro hostování procesu. Celou problematiku vztahuji na BPMS technologie IBM.

Závěrečná kapitola shrnuje poznatky a výsledky mé diplomové práce. Zamýšlím se zde nad možnostmi pokračování, případně budoucího navázání na mou práci.

2 BUSINESS PROCES MANAGEMENT

Ve své práci vycházím z teoretického podkladu, který považuji za nutné alespoň stručně popsat. Bez jeho přiblížení a následného porozumění čtenářem není možné porozumět celkovému kontextu, do kterého je tato práce jak po teoretické, tak praktické stránce zasazena. Považuji za nezbytné zmínit, definovat základní pojmy, které dále v textu používám. Stejně tak popíšu základní mechanismy a přístupy, ze kterých čerpám.

BPM – Business Proces Management

Jedná se manažerskou disciplínu a současně i technologii, která využívá business model pro procesně orientované řízení podniku.

2.1 Business model

Je konzistentní a vzájemně provázaný popis strategických cílů, měřitelných ukazatelů výkonnosti, procesů, pravidel podnikání, znalostí, organizačních struktur, rizik a IT služeb.



Obrázek č.1 - schéma business modelu. Zdroj: skripta z předmětu PV206 na FI MUNI.

2.2 Procesně orientovaný přístup

2.2.1 Proces

Procesem rozumíme opakující se sekvenci kroků transformující vstupy do procesu na výstupy z procesu. Takovýmto procesem může být například jednoduchý proces rezervace míst k sezení v kině. Vstupem je požadavek zákazníka na rezervaci a vybrání volného místa. Výstupem rezervace zvoleného místa, případně oznámení, že jsou již všechna místa rezervována. Výkon procesů by měl být měřitelný a výsledky měření využívány pro vlastní zlepšování procesů. Procesy a jejich infrastruktura musí poskytovat flexibilitu ke změnám, vznikajícím na základě strategického zadání.

2.2.2 Definice procesu

Popsaný algoritmus procesu. Popisuje chování a sémantiku procesu. Pro vyjádření definice procesu je vhodné použít některý z procesních diagramů.

2.2.3 Instance procesu

Konkrétní instance daného procesu. Instance představuje běh procesu. Proces často běží ve velkém množství instancí. Instance se od sebe liší vstupními daty, která do procesu vstupují.

2.2.4 Aktivita, úkol

Jednotlivý krok v procesu. Například zaslání seznamu volných míst v kině. Aktivita může být vykonána automaticky (krok, který provede nějaká aplikace automaticky) nebo manuálně některým uživatelem.

2.3 Motivace pro použití procesně orientovaného přístupu

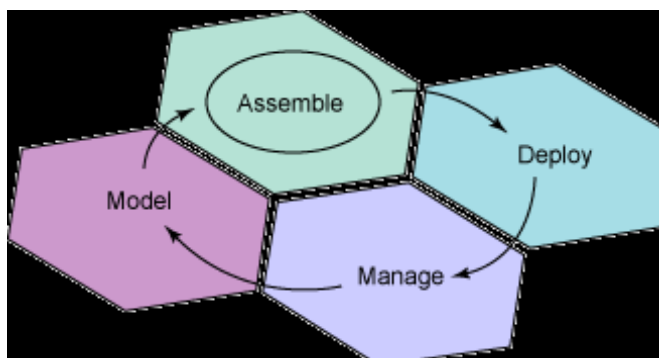
Zde uvádím seznam výhod, které použití procesně orientovaného přístupu přináší:

- **Zvyšování výkonnosti a efektivity**
Formalizace, popsání a rozbor procesu na základě monitorovacích dat z běhu instancí procesu, mohou vést k nalezení úzkých míst, zbytečných kroků a celkově k optimalizaci a zefektivnění procesu.
- **Podpora automatizace**
Při rozboru procesu můžeme nalézt kroky, které mohou být vykonány automaticky místo toho, aby byly vykonávány manuálně. Dochází tak ke zvyšování produktivity a nabízí možnost snižování počtu zaměstnanců.
- **Podpora uživatelského zásahu do procesů**
Navzdory skutečnosti, že procesně orientovaný přístup umožňuje redukovat počet zaměstnanců. Poskytuje taky prostředky pro flexibilitu řešit vzniklé problémy na uživatelské úrovni.

Zajímavé jsou velmi optimistické prognózy nárůstu používání tohoto přístupu. Pan Václav Kalenda na svých přednáškách, který je jedním ze zástupců serveru www.procesy.cz, hovoří o meziročním růstu o 40% do roku 2011.

2.4 Životní cyklus procesu

Proces, včetně jeho projekce do business modelu konkrétní organizace, prochází jednotlivými etapami vývoje. První etapou je analýza a návrh procesu („model“), druhou implementace („assemble“), třetí nasazení a běh procesu („deploy“). Poslední čtvrtou fází, je administrace a monitoring procesu („manage“). Jednotlivé fáze popíšu podrobněji.



Obrázek č.2 - schéma životního cyklu procesu. Zdroj: www.ibm.com

2.4.1 Analýza a návrh procesu („model“)

Výstupem této fáze je intuitivní diagram procesu, který popisuje jednotlivé kroky procesu reflektující business požadavky, které jsou na proces kladeny. Tento diagram je vytvářen jak business analytiky, tak technickými architekty. Účast business analytiků při vytváření diagramu je žádoucí, neboť nejlépe rozumí danému business procesu. Technicky zaměření architekti jejich postřehy doplní o technicky orientované aspekty procesu. Pro efektivní součinnost těchto skupin při vytváření modelu procesu, je klíčové vytvářet diagram, který je postaven na prvcích, kterým rozumí obě skupiny. Pro business analytiky poskytuje notifikaci business orientovaného pohledu a technikům notifikaci pro systémově orientovaný pohled. Takovouto podporu nabízí BPMN („Business Process Modeling Notification“), který je na tomto poli nejrozšířenějším standardem.

2.4.2 Implementace procesu („assemble“)

V současné době chybí plynulý a hlavně automatický přechod od návrhu procesu k jeho běhu na běhovém prostředí. Přes veškeré invence velkých hráčů na poli IT není tento požadavek současnými technologiemi zcela naplněn. Proto je mezi fází designu a běhu procesu uvažovat o fázi implementace. Jedná se převod analytického modelu do spustitelného kódu, který pak běží na procesním serveru.

2.4.3 Nasazení a běh procesu („deploy“)

Implementovaný proces ve spustitelné podobě je nasazen na procesní server. Podobným způsobem jako jakákoliv jiná aplikace. Běh procesu představuje vytváření a průběh životní cyklů jednotlivých instancí na základě požadovaných vstupů.

2.4.4 Administrace a monitoring procesu („manage“)

- **Administrace**

Je důležité, aby běžící instance procesů mohly být administrovány. Klíčové je obzvlášť uživatelsky vidět aktuální stav konkrétní instance procesu, řešit chybové stavy procesu, zrušit nebo vytvořit instanci procesu, modifikovat instanci procesu, zavést novou definici procesu, zrušit definici procesu.

- **Monitoring**

Na monitoring business procesů se můžeme dívat jako rozšíření administrátorského přístupu. Jedná se zejména o managerský pohled na procesy z hlediska efektivity a výkonnosti jednotlivých kroků procesu i procesu jako celku. Tento uživatelský pohled je často reprezentován formou grafů, tabulek a schémat.

2.5 Model zrání procesů („Process Maturity Model“)

Model zrání procesů vyjadřuje úroveň zralosti procesů o povědomí o nich v dané organizaci. Stejně tak úroveň, jak se s těmito procesy pracuje. Tento model byl vytvořen organizací Software Engineering Institute (SEI) na základě nasbíraných zkušeností. Cílem tohoto modelu bylo identifikovat klíčové aspekty a postupy vedoucí k celkovému snížení nákladů vynaložených na procesy, zvýšení produktivity a kvality výstupů z procesů.

Model má následujících 5 úrovní:

1. *Ad hoc procesy („initial“)*

Pro procesy na této úrovni je typické, že jsou nedokumentovány, jsou ve stavu neustálé změny, jsou nekontrolovatelné. Tato charakteristika má za důsledek nestabilní procesní prostředí jako takové.

2. *Opakující se procesy („repeatable“)*

Pro procesy na této úrovni je charakteristické, že se opakují a že jsou relativně neměnné. Objevují se náznaky zavedení procesní disciplíny u některých z procesů.

3. *Definované procesy („defined“)*

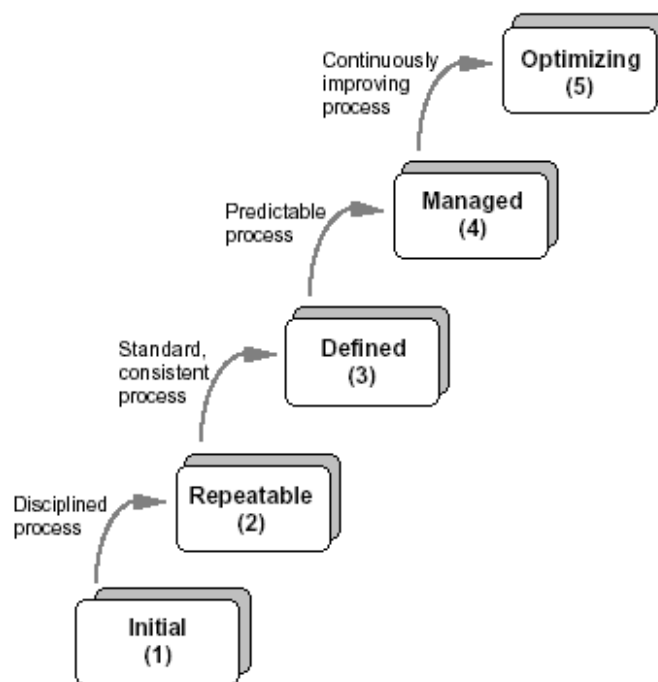
Procesy jsou definovány a popsány některým ze standardů. Příležitostně dochází ke zlepšování procesů.

4. *Řízené procesy („managed“)*

U procesů jsou používány metriky pro potřeby managementu určující výkonnost a efektivitu procesů.

5. *Optimalizované procesy („optimizing“)*

Pro procesy je charakteristické kontinuální zlepšování z hlediska výkonnosti a efektivity a to zejména na poli technologických a inovativních změn.



Obrázek č.3 - schéma modelu zrání procesů. Zdroj: www.wikipedia.com

2.6 BPMS (Business Proces Management Systém)

Jedná se systémy, které podporují management business procesů. Nabízí funkcionalitu potřebnou v jednotlivých fázích životního cyklu procesu. (design, implementace, běh a monitoring procesu).

Použité komponenty BPMS pro účely této diplomové práce:

- **Design procesu**
 - Visual Paradigm, modulu Business Process Visual ARCHITECT 2.4 Analyst Edition
 - Enterprise Architect 5.0
- **Implementace procesu**
 - IBM Websphere Integration Developer V 6.2
- **Běh procesu**
 - IBM Websphere Process Server V 6.2

3 TECHNICKÉ ASPEKTY BPM

V kapitole o BPM jsem se zabýval obecným pohledem na BPM a procesy jako takové. Tento úhel pohledu je postačující pro business analytiku a managery. Aby mohli být procesy implementovány, je zapotřebí rozšířit tento pohled o jednotlivé technické aspekty a namapovat procesy do konkrétního technologického řešení. Považuji za vhodné zmínit se obecně o webových službách, ESB, SOA orchestraci webových služeb.

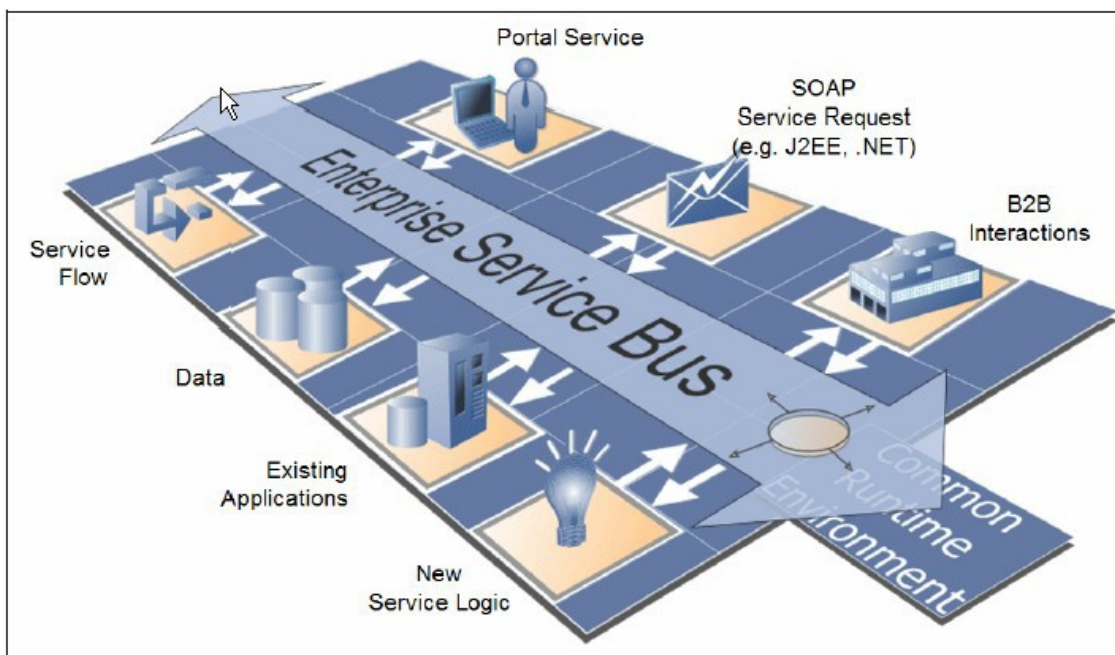
3.1 Platforma webových služeb („Webservices“)

Architektura webových služeb zahrnuje řešení vzájemné komunikace jednotlivých aplikací, případně aplikačních komponent (služeb). Jedná se o poslední distribuovanou technologii. Interaktivita účastníku je realizována formou přenášení zpráv ve formátu XML. Webové služby umožňují synchronní a asynchronní komunikaci. Webové služby jsou na rozdíl od objektově orientovaného paradigmatu bezstavové. Využívají pro transport zpráv standardní internetové protokoly jako HTTP, FTP, MIME, SMTP. Webové služby jsou implementací SOA architektury, proto informace obsažené v části o SOA se budou ve velké míře týkat i webových služeb.

3.2 ESB („Enterprise Service Bus“)

Podniková sběrnice služeb je spolehlivou, robustní a škálovatelnou softwarovou infrastrukturou plnící funkci vrstvy prostředníka („middleware“) v komunikaci mezi službami. ESB přináší flexibilitu do komunikace mezi službami, usnadňuje proces integrace a znovupoužitelnost služeb. ESB umožňuje připojení nejenom webových služeb, ale také služeb implementovaných na dalších technologiích. (EJB, CORBA komponenty, stávající aplikace a jiné middleware produkty) Slouží jako prostředník mezi vzájemně nekompatibilními middleware produkty a komunikačními protokoly. Umožňuje konfigurovat QoS („Quality of Service“) u služeb. Zahrnujíce spolehlivost služeb, ošetření chybových stavů, a bezpečnost služeb. ESB poskytuje management nad komunikací jako je směrování zpráv, vzájemná interakce služeb a transformace zpráv do požadovaného formátu. Na základě výše zmíněného ESB přináší následující benefity:

- Zvýšená flexibilitu služeb
- Nižší režije při zavádění a vývoji služeb
- Snížené náklady na údržbu služeb
- Zvýšení spolehlivosti a managementu služeb



Obrázek č.3 - schéma modelu zrání procesů. Zdroj: skripta z předmětu PV206 na FI MUNI.

3.3 SOA („Service Oriented Architecture“)

Službově orientovaná architektura podporuje volné svázání služeb („loose coupling“) za účelem flexibility a interoperability businessu z technologického hlediska. SOA se skládá z business relevantních služeb. SOA není chápána jako balík technologií. Není přímo vztažena k žádné technologii. Nicméně v praxi je SOA často implementována pomocí webových služeb, neboť webové služby jsou pro realizaci SOA nejvhodnější. SOA jako architektura přesahuje architekturu webových služeb. Webové služby jsou pak používány k naplnění následujících konceptů:

- **Služby („services“)**

Služby nabízejí business funkcionalitu jako je například žádost o hypotéku. Nejedná se tedy o technologicky orientovanou funkcionalitu jako je získávání obsahu a update tabulek v databázi. Služby SOA musí poskytovat business hodnotu, skrýt implementační detaily a být autonomní.

- **Rozhraní služeb („interfaces“)**

Ke službám se přistupuje přes jejich rozhraní. Rozhraní definuje množinu popisů veřejných operací služby („operation signatures“). Rozhraní se stává kontraktem mezi producentem a konzumentem služby. Rozhraní je odděleno od implementace, je sebedopisující a platformově nezávislé.

- **Zprávy („messages“)**

Zprávy jsou vstupní nebo výstupní data, která si služby při své interakci vyměňují. Obsah

zpráv a popis vstupních a výstupních parametrů operací je platformově nezávislý. Tím se liší od objektově a komponentově orientovaného přístupu.

- **Synchronnost a asynchronnost operací**

Konzumenti služeb přistupují ke službám přes sběrnici služeb („service bus“). Touto sběrnici může být například transportní protokol SOAP nebo ESB. Operace služeb jsou volány synchronně nebo asynchronně. V synchronním módu operace vrátí odpověď na volání („response“) konzumentovi, který službu volal, jakmile je operace provedena. Konzument čeká na provedení dané operace. Synchronní volání je používáno pro operace, jejichž provedení není příliš časově náročné. V asynchronním módu konzument nečeká na provedení operace. Maximálně získá jako odpověď na volání potvrzení, že byla operace vyvolána úspěšně. V případě, že je po provedení operace potřeba předat nějaký výstup konzumentovi, služba volá zpět klienta. („callback“)

- **Volné vázání („loose coupling“)**

Volné vázání služeb znamená, že jsou služby vystavovány minimálnímu a pouze nezbytnému množství závislostí. Tato vlastnost je důležitá zejména pokud služby čelí častým změnám. Minimální počet závislostí zaručuje, že dochází k minimálnímu počtu změn ostatních služeb, při modifikaci jedné. Takovýto přístup zvyšuje robustnost celého systému a jeho přizpůsobivost změnám.

- **Registry („registries“)**

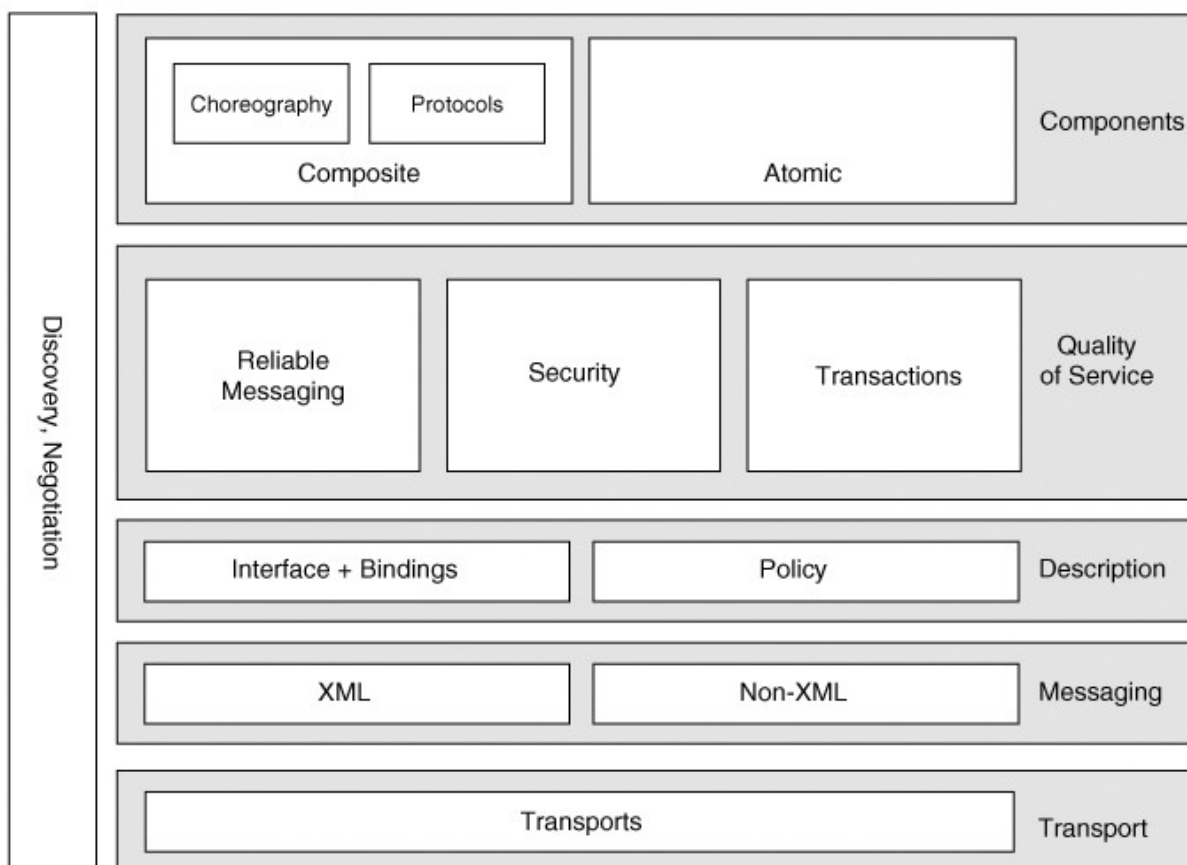
Služby jsou uvedené v registrech služeb, které fungují na způsob adresářové struktury. Tím je usnadněno vyhledávání vhodné služby. Poskytovatelé služeb publikují služby do registru, konzumenti pak tyto služby v registrech vyhledávají. UDDI je příkladem takového registru služeb.

- **Kvalita služeb („quality of services“)**

Na služby jsou kladeny požadavky z hlediska kvality. Jedná se například o požadavek na bezpečnost, spolehlivého doručení zprávy, transakčního zpracování a další.

- **Kompozice služeb do business procesů**

Jedná se o finální a hlavní koncept SOA. Služby jsou v konkrétním pořadí a souslednosti vkomponovávány do business procesů. Kompozice služeb poskytuje podporu pro procesy v jejich flexibilitě. Takovéto procesy je pak možné rychle a jednoduše modifikovat. Pro kompozici jsou použity exekutivní jazyky a takto naimplementovaný proces pak běží ve svých instancích na procesním serveru. Nejrozšířenější a nejpoužívanější z těchto jazyků je jazyk **BPEL**. („Business Process Executive Language“).



Obrázek č.5 – jednotlivé vrstvy SOA architektury

Zdroj: kniha Prentice Hall – Web Services Platform Architecture

„Transport“ představuje transportní vrstvu pro přenos zpráv. Bývá zpravidla implementována internetovými protokoly HTTP, TCP/IP, RMI atd.

„Messaging“ představuje vrstvu zprávového frameworku („Message Framework“). Jako framework, kde jsou zprávy ve formátu XML, může být použit SOAP („Simple Object Access Protocol“). Jako non XML framework například JMS („Java Message Service“).

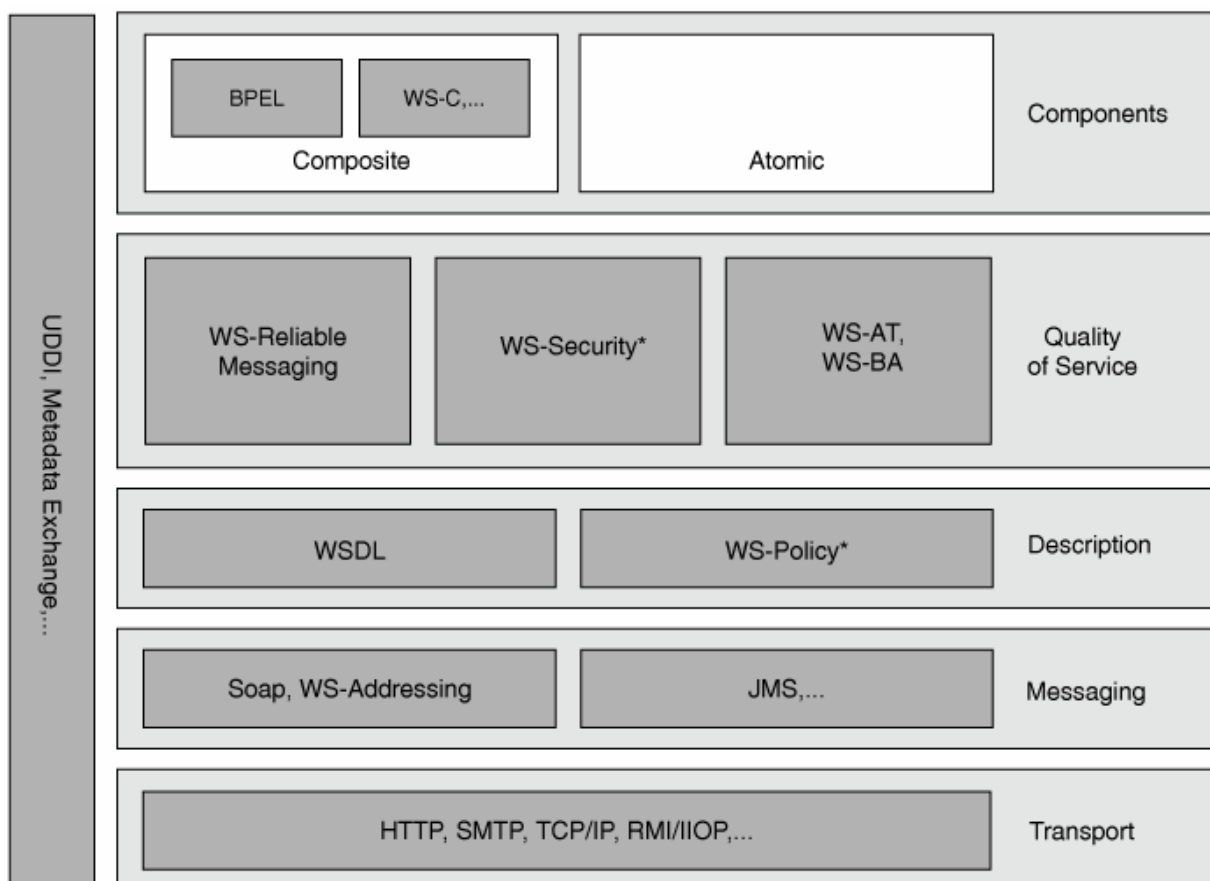
„Description“ je vrstva popisující podobu, vlastnosti, chování a umístění služeb. Poskytuje tedy metadata o službách. Pokud uvažujeme realizaci SOA pomocí webových služeb, je tato vrstva zastoupena WSDL („Webservice Description Language“), a WS-Policy, která specifikuje, jak jsou deklarovány politiky pro práci s webovými službami.

„Quality of Service“. Základními stavebními kameny této vrstvy jsou bezpečnost („security“), spolehlivé doručování zpráv („reliable messaging“) a transakční zpracování služeb („transactions“). V případě realizace webovými službami dochází k namapování na specifikace WS-Reliable Messaging, WS-Security, WS-AT, WS-BA.

„Components“ je nejvyšší a hlavní vrstvou SOA. Představuje služby atomického charakteru i služby

komponované z jiných služeb. Tyto služby jsou pak orchestrovány do vyššího celku – procesu. Orchestrace znamená, že jednotlivé služby neví, že jsou součástí procesu. Veškerou komunikaci, návaznost, řešení chybových stavů je v režii procesu. V současné době je nejrozšířenějším a tedy i obecně nevhodnějším jazykem pro implementaci procesů, deklarativní jazyk BPEL.

Mimo vrstvy stojí „**Discovery**“ představující registry popsané výše.



Obrázek č.6 – jednotlivé vrstvy SOA architektury namapované na konkrétní technologie

Zdroj: kniha Prentice Hall – Web Services Platform Architecture

TECHNICKÝ DESIGN PROCESU „SLUŽEBNÍ CESTA „

Podkladem pro analýzu a design procesu „služební cesta“ byl velmi abstraktní popis procesu v přirozeném jazyce. Tento popis jsem rozšířil o analýzu vycházející ze zaměření se na reálnou podobu tohoto procesu ve fiktivní společnosti. Sémantiku procesu jsem vyjádřil BPMN diagramem, doplněný komentářem vysvětlující jednotlivé kroky. BPMN diagram byl vytvořen v case nástroji Visual Paradigm v modulu Business Process Visual ARCHITECT 2.4 Analyst Edition.

Dále jsem identifikoval uživatelské role vyskytující se v procesu a navrhl způsob, jakým budou s procesem interagovat. BPMN diagram popisuje zejména funkční chování procesu. Tento funkční pohled jsem doplnil a datový model procesu. Datový model jsem vytvořil v case nástroji Enterprise Architect 5.0.

3.4 Popis procesu v přirozeném jazyce

verze: 1.0

autor: Jan Pavlovič

popis:

„Služební cesta je inicializovaná žádostí zaměstnance, který podá žádost. Zaměstnavatel žádost schválí, zamítne nebo odešle zaměstnanci k modifikaci. Po ukončení služební cesty zaměstnanec vloží fakturační informace včetně závěrečné zprávy. HR oddělení případně vyzve zaměstnance k doplnění údajů nebo schválí a vygeneruje report, který zaměstnanec podepíše“

3.5 Uživatelské role vyskytující se v procesu

Na uživatelské úrovni mohou s procesem z klientské aplikace interagovat uživatelé v následujících rolích.

1. **Role Zaměstnanec**

Jedná se o roli představující běžného zaměstnance firmy. Tento zaměstnanec si může zažádat o služební cestu, vložit detaily o služební cestě a v případě nutnosti je oprávněn modifikovat žádost o služební cestu případně vložené detaily.

2. **Role Zaměstnavatele**

Role je chápána jako vedoucí pracovník daných zaměstnanců. Mezi jeho činnosti v procesu patří schválení, zamítnutí žádosti o služební cestu a vrácení žádosti o služební cestu zpět zaměstnanci jako pokynu k modifikaci zaslaných dat.

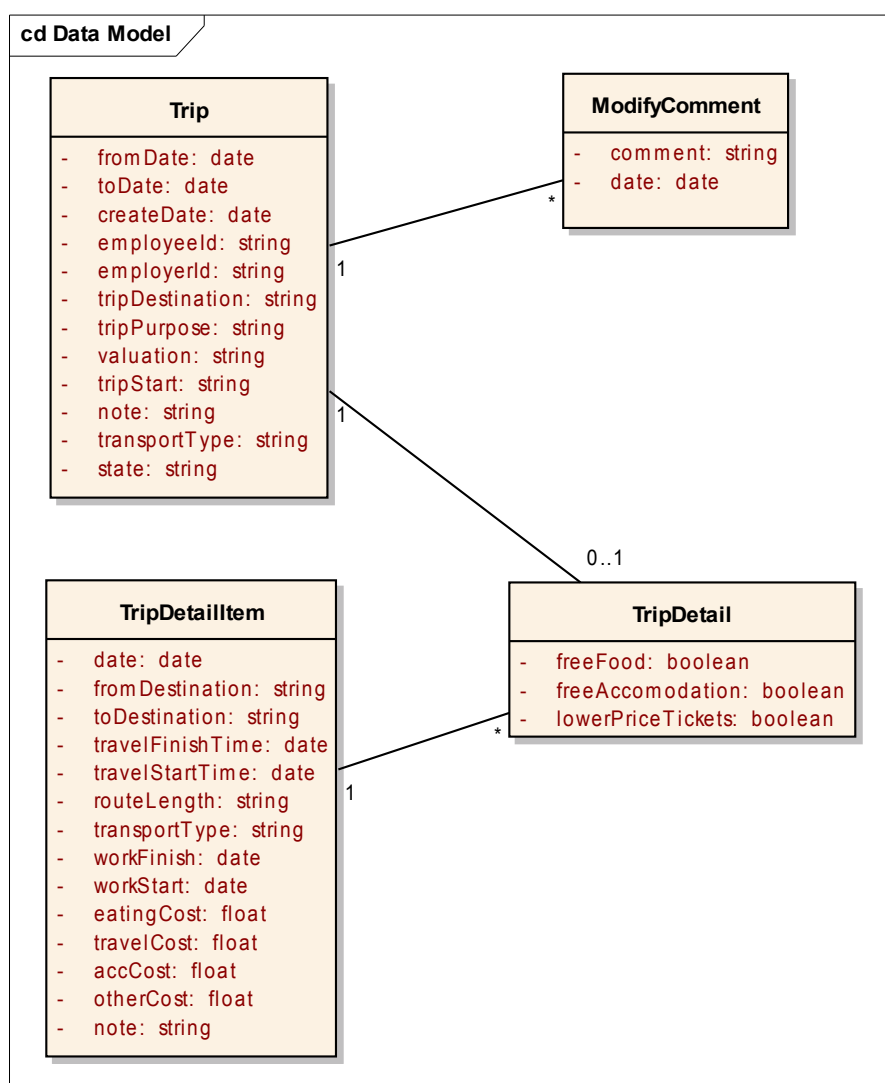
3. **Role HR**

V této roli se vyskytují pracovníci z oddělení pro lidské zdroje. Jejich úkolem v procesu je zkontrolovat vložené detaily zaměstnancem po absolvování služební cesty. Vložené údaje schválí nebo vrátí zaměstnanci k modifikaci.

3.6 Uživatelské rozhraní

Uživatelské rozhraní, ze kterého uživatelé v daných rolích interagují s procesem je realizováno formou webové java EE aplikace. Celkový uživatelský přístup je řešen formou přidělování úkolů. Jakmile je z této klientské webové aplikace zadán požadavek na služební cestu, vzniká instance procesu, která má automatické systémové kroky a manuální kroky, které jsou potřeba splnit požadovaným uživatelem. Pokud se proces posune ke kroku, který je manuální, v klientské webové aplikaci se tento krok zobrazí jako úkol uživateli, kterému je přiřazen. Úkol je přiřazen konkrétnímu zaměstnanci nebo libovolnému zástupci skupiny HR, případně zaměstnavateli. Po úspěšném zpracování úkolu, úkol z fronty úkolů opět zmizí.

3.7 Datový model procesu



V modelu se u atributů vyskytují následující datové typy:

- **string** – typ znakového řetězce
- **boolean** – typ pravdivostní hodnoty

- **float** – číselný typ desetinných čísel
- **date** – formát datumu. Kromě informace o datumu obsahuje rovněž informaci o čase s přesností na jednu milisekundu.

3.7.1 Popis datových entit a jejich atributů:

3.7.1.1 Trip

Objektem typu „Trip“ je každý objekt představující Služební cestu. Objekt může obsahovat žádný nebo několik (shora neomezeně) objektů typu „ModifyComment“. Objekt může obsahovat žádný nebo jeden objekt typu „TripDetail“. Služební cesta může být v rámci procesu „Služební cesta“ v různých stavech, přičemž momentálně vždy v právě jednom.

Atributy:

Atribut	Poznámka
fromDate date	Datum počátku plánované služební cesty. Uvádí se při žádosti o služební cestu.
toDate date	Datum konce plánované služební cesty. Uvádí se při žádosti o služební cestu.
createDate date	Datum vytvoření žádosti o služební cestu.
employeeId string	Identifikátor zaměstnance, který žádá o služební cestu.
employerId string	Identifikátor zaměstnavatele, který schvaluje, zamítá nebo vrací žádost o služební cestu k modifikaci zaměstnanci.
tripDestination string	Cílová destinace služební cesty.
tripPurpose string	Účel služební cesty.
valuation string	Odhad nákladů na služební cestu
tripStart string	Počáteční (výchozí) bod služební cesty.
note string	Poznámka
transportType string	Způsob přepravy u služební cesty. Povoleny jsou následující hodnoty: „bus“ pro přepravu autobusem, „train“ pro přepravu vlakem, „carO“ pro přepravu vlastním autem, „carC“ pro přepravu služebním autem, „plain“ pro přepravu letadlem, „comb“ pro kombinovanou přepravu, „other“ pro jiný druh přepravy.
state string	Stav objektu „Trip“ v rámci procesu „Služební cesta“. Povoleny jsou následující hodnoty: „claimCreated“ – žádost o služební cestu byla vytvořena „claimRefused“ – žádost o služební cestu byla zamítnuta „claimApproved“ – žádost o služební cestu byla schválena

Atribut	Poznámka
	„claimToModify“ – žádost byla vrácena k modifikaci „claimModified“ – žádost byla modifikována „detailsIns“ – byly vloženy detaily o služební cestě „detailsToM“ – detaily o služební cestě byly vráceny k modifikaci „detailsM“ – detaily o služební cestě byly modifikovány „detailsApproved“ – detaily byly schváleny „tripClosed“ – služební cesta je uzavřena po úspěšném průchodu procesem.

3.7.2 *ModifyComment*

Objektem typu „ModifyComment“ je každý objekt, který představuje komentář při vrácení žádosti o služební cestu k modifikaci nebo při vrácení detailů o služební cestě k modifikaci.

Atributy:

Atribut	Poznámka
comment string	Popis důvodu vrácení k modifikaci
date date	Datum a čas vrácení k modifikaci

3.7.3 *TripDetail*

Objektem typu „TripDetail“ je každý objekt, který představuje údaje potřebné k vyúčtování služební cesty. Objekt může obsahovat žádný nebo několik (shora neomezeně) objektů typu „TripDetailItem“.

Atributy:

Atribut	Poznámka
freeFood boolean	Indikátor, zda bylo na služební cestě stravování zdarma.
freeAccommodation boolean	Indikátor, zda bylo na služební cestě ubytování zdarma.
lowerPriceTickets boolean	Indikátor, zda byla uplatněna sleva na jízdenky.

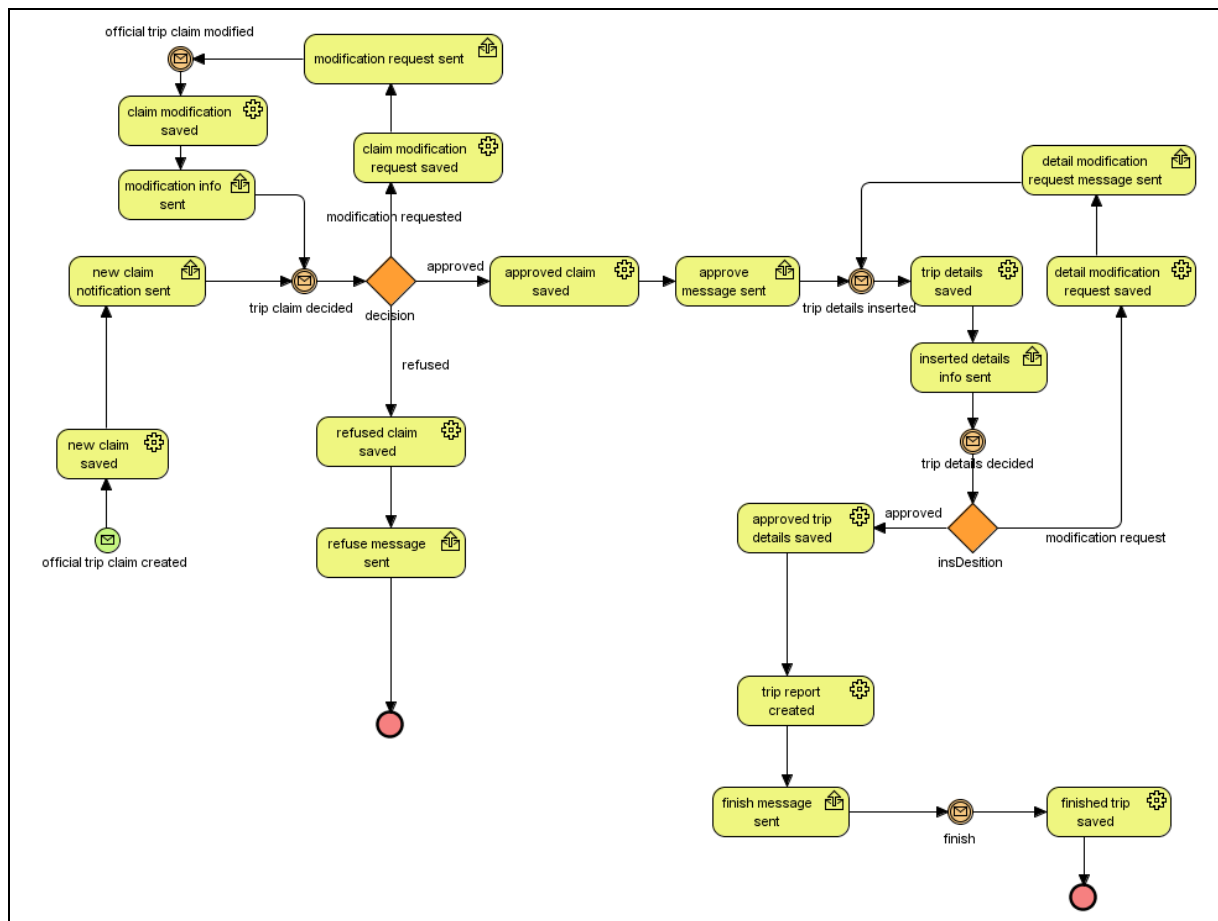
3.7.4 *TripDetailItem*

Objektem typu „TripDetailItem“ je každý objekt položky detailu služební cesty. Nese údaje o jednotlivých částech služební cesty. Služební cesta, tak může obsahovat zvlášť položku pro cestu do cílové destinace a zvlášť položku pro cestu nazpět. Služební cesta může vést přes několik destinací. Například z destinace A jedeme autem do destinace B. Z destinace B do destinace C letíme letadlem. Objekt typu „TripDetailItem“ obsahuje informace o těchto jednotlivých částech služební cesty.

Atributy:

Atributy	Poznámky
date date	Datum bez časového údaje o dané části služební cesty.
fromDestination string	Počáteční destinace dané části služební cesty.
toDestination string	Koncová destinace dané části služební cesty.
travelFinishTime date	Koncový čas absolvování dané části služební cesty.
travelStartTime date	Počáteční čas dané části služební cesty.
routeLength string	Délka trasy dané části služební cesty.
transportType string	Způsob přepravy dané části služební cesty. Povoleny jsou následující hodnoty: „bus“ pro přepravu autobusem, „train“ pro přepravu vlakem, „carO“ pro přepravu vlastním autem, „carC“ pro přepravu služebním autem, „plain“ pro přepravu letadlem, „other“ pro jiný druh přepravy.
workFinish date	Čas počátku pracovního úkonu.
workStart date	Čas konce pracovního úkonu.
eatingCost float	Stravovací náklady
travelCost float	Cestovní náklady
accCost float	Ubytovací náklady
otherCost float	Jiné výdaje
note string	Poznámka

3.8 Business Proces Diagram





Komentář k diagramu:








Pro účely správného pochopení jednotlivých kroků v diagramu a jednoznačnosti diagramu jako takového, popíšu význam jednotlivých částí diagramu.







V digramu se vyskytují následující typy grafických prvků vycházející přímo ze specifikace modelovacího jazyka pro business procesy BPMN:









- Sekvenční tok
Je znázorněn černou šipkou.
- Startovní událost procesu typu příchozí zprávy
- Koncový stav procesu
- Systémová aktivita v procesu typu volání webové služby.
- Systémová aktivita v procesu typu posílání notifikace uživateli






- Událost v procesu typu příchozí zprávy

- Strukturní element typu „case“ nebo „switch“. Na základě vstupní podmínky vybere jeden z výstupních sekvenčních toků.


Popis jednotlivých kroků v diagramu:

Element diagramu	Popis
 official trip claim created	Do procesu přichází zpráva z klientské webové aplikace. Zpráva představuje požadavek zaměstnance na schválení služební cesty. Zpráva obsahuje elementy s vyplněními hodnotami. Názvy a význam elementů jsou popsány v datovém modelu. Příchozí zpráva obsahuje elementy: fromDate, toDate, createDate, employeeId, tripDestination, tripPurpose, valuation, tripStart, note, transportType.
 new claim saved	Pomocí volání webové služby backendu (systémového prostředí reprezentující datovou vrstvu) je v systému vytvořen perzistentní objekt typu „Trip“ a jsou uloženy hodnoty elementů příchozí zprávy do příslušných atributů tohoto objektu. Stav objektu se nastaví na „claimCreated“.
 new claim notification sent	Je zaslána notifikace zaměstnavateli daného zaměstnance, aby rozhodl o schválení, neschválení, případně vyzval daného zaměstnance k modifikaci žádosti. Notifikace je realizována formou objevení se nového úkolu v seznamu úkolů zaměstnavatele. Tento seznam je viditelný z klientské webové aplikace.
 trip claim decided	Jakmile zaměstnavatel daný úkol zpracuje. Přichází z klientské webové aplikace zpráva nesoucí informaci o tom, jak se zaměstnavatel rozhodl. Označme ji jako „desMess“. Zpráva obsahuje element status. Přípustné hodnoty tohoto elementu jsou „APPROVED“, v případě, že žádost schválena. „DISAPPROVED“ v případě, že je žádost zamítnuta a „MODIFY“ v případě, že je žádost vrácena danému zaměstnanci k modifikaci. Dalšími elementy jsou element note, který může obsahovat komentář k rozhodnutí a element date, který obsahuje datum a čas daného rozhodnutí.
 decision	Na základě hodnoty elementu status zprávy „desMess“ o rozhodnutí se určí, kterou sekvenční větví bude proces „Služební cesta“ pokračovat.
 approved claim saved	Objekt typu „Trip“ si přes webovou službu uloží stav „claimApproved“.
	Notifikace zaměstnanci, že jeho služební cesta byla

approve message sent	schválena. Zaměstnanec, teď může vložit detaily o služební cestě. Notifikace je realizována formou objevení se úkolu „vložení detailů o služební cestě“ v seznamu úkolů daného zaměstnance viditelného z klientské webové aplikace.
 claim modification request saved	Objekt typu „Trip“ si přes webovou službu uloží stav „claimToModify“. Dále je do objektu přidán nový objekt typu ModifyComment“. U tohoto objektu je atribut naplněn hodnotou elementu note z příchozí zprávy rozhodnutí. Atribut date objektu typu „ModifyComment“ je naplněn hodnotou elementu date z příchozí zprávy rozhodnutí
 modification request sent	Notifikace zaměstnanci, že jeho služební cesta byla vrácena k modifikaci. Zaměstnanec teď může opravit požadavek na služební cestu. Notifikace je realizována formou objevení se úkolu „Modifikace požadavku na služební cestu“ v seznamu úkolů daného zaměstnance viditelného z klientské webové aplikace.
 official trip claim modified	Do procesu přichází zpráva (modMessage) z klientské webové aplikace. Zpráva představuje modifikovaný požadavek zaměstnance na schválení služební cesty. Zpráva obsahuje elementy s vyplněními hodnotami. Názvy a význam elementů jsou popsány v datovém modelu. Příchozí zpráva obsahuje elementy: fromDate, toDate, tripDestination, tripPurpose, valuation, tripStart, note, transportType.
 claim modification saved	Pomocí volání webové služby backendu (systémového prostředí reprezentující datovou vrstvu) je v systému updatován perzistentní objekt typu „Trip“, který vznikl při prvním požadavku daného zaměstnance na danou služební cestu. Jsou do něho uloženy hodnoty elementů příchozí zprávy modMessage do příslušných atributů. Stav objektu se nastaví na „claimModified“.
 modification info sent	Je zaslána notifikace zaměstnavateli daného zaměstnance, aby rozhodl o schválení, neschválení, případně vyzval daného zaměstnance k modifikaci žádosti. Notifikace je realizována formou objevení se nového úkolu v seznamu úkolů zaměstnavatele. Tento seznam je viditelný z klientské webové aplikace.
 trip details inserted	Do procesu přichází zpráva (detMessage) z klientské webové aplikace. Zpráva představuje detail o proběhnuté služební cestě zaměstnance. Zpráva obsahuje elementy s vyplněními hodnotami. Názvy a význam elementů jsou popsány v datovém modelu. Příchozí zpráva obsahuje elementy: freeFood, freeAccommodation, lowerPriceTickets. Dále může obsahovat jeden nebo více elementů „TripDetailItem“, které představují jednotlivé položky detailu a služební cestě. Element „TripDetailItem“ obsahuje následující vlastní elementy. Jsou to: date, fromDestination,

	toDestination, travelFinishTime, travelStartTime, routeLengh, transportType, workFinish, workStart, eatingCost, travelCost, accCost, otherCost, note.
 trip details saved	Pomocí volání webové služby backendu (systémového prostředí reprezentující datovou vrstvu) je v systému updatován perzistentní objekt typu „Trip“, který vznikl při prvním požadavku daného zaměstnance na danou služební cestu. Jsou do něho uloženy hodnoty elementů příchozí zprávy detMessage do příslušných atributů. Stav objektu se nastaví na „detailsIns“.
 inserted details info sent	Je zaslána notifikace skupině pracovníků z oddělení péče o lidské zdroje. Libovolný člen skupiny úkol přijme a rozhodne o schválení, případně vyzve daného zaměstnance k modifikaci vloženého detailu. Notifikace je realizována formou objevení se nového úkolu v seznamu úkolů pro tuto skupinu. Tento seznam je viditelný z klientské webové aplikace.
 trip details decided	Do procesu přichází zpráva (insDetMessage) z klientské webové aplikace. Zpráva představuje rozhodnutí o schválení vloženého detailu o služební cestě. Zpráva obsahuje element status. Příпустné hodnoty tohoto elementu jsou „APPROVED“ v případě, že byla žádost schválena a „MODIFY“ v případě, že je detail vrácen danému zaměstnanci k modifikaci. Dalšími elementy jsou element note, který může obsahovat komentář k rozhodnutí a element date, který obsahuje datum a čas daného rozhodnutí.
 insDecision	Na základě hodnoty elementu status příchozí zprávy insDetMessage se určí, kterou sekvenční větví bude proces „Služební cesta“ pokračovat.
 detail modification request saved	Objekt typu „Trip“ si přes webovou službu uloží stav „detailsToMod“. Dále je do objektu přidán nový objekt typu ModifyComment“. U tohoto objektu je atribut naplněn hodnotou elementu note z příchozí zprávy insDetMessage. Atribut date objektu typu „ModifyComment“ je naplněn hodnotou elementu date z příchozí zprávy insDetMessage.
 detail modification request message sent	Notifikace zaměstnanci, že detail o služební cestě byl vrácen k modifikaci. Zaměstnanec teď může opravit detail o služební cestě. Notifikace je realizována formou objevení se úkolu „Modifikace detailu na služební cestu“ v seznamu úkolů daného zaměstnance viditelného z klientské webové aplikace.
 approved trip details saved	Objekt typu „Trip“ si přes webovou službu uloží stav „detailsApproved“.
 trip report created	Z objektu typu „Trip“ se vygeneruje report ve formátu .pdf.

 finish message sent	Je zaslána notifikace skupině pracovníků z oddělení péče o lidské zdroje. Libovolný člen skupiny úkol přijme, vytiskne report a nechá jej podepsat. Celý proces pak ukončí. Notifikace je realizována formou objevení se nového úkolu v seznamu úkolů pro tuto skupinu. Tento seznam je viditelný z klientské webové aplikace.
 finish	Do procesu přichází zpráva (finMessage) z klientské webové aplikace. Zpráva představuje uzavření procesu „služební cesta“. Zpráva obsahuje element note, představující komentář k uzavření.
 finished trip saved	Objekt typu „Trip“ si přes webovou službu uloží stav „tripClosed“.
 refused claim saved	Objekt typu „Trip“ si přes webovou službu uloží stav „claimRefused“.
 refuse message sent	Notifikace zaměstnanci, že jeho služební cesta byla zamítnuta.

4 REALIZACE BUSINESS PROCESU „SLUŽEBNÍ CESTA“

4.1 Implementace procesu služební cesta

Proces „Služební cesta“ je implementován v jazyku BPEL. Je obsažen v aplikaci „Trip“, která představuje integrační projekt ve vývojovém prostředí IBM Websphere Integration Developer. Buildem toho projektu vznikne modul nasaditelný na procesní server IBM Websphere Process Server.

Projekt obsahuje:

- Sémantika procesu popsána v XML jazykem BPEL
- Datové typy jednotlivých zpráv vyskytujících se v procesu. Podoba datových typů je odvozena z datového modelu procesu popsaného v kapitole technického designu. Podoba zpráv je popsána v jednotlivých krocích popisu procesního diagramu služební cesty.
- Popis rozhraní ve formátu WSDL. Proces vystupuje jako webová služba, je tedy popsán pomocí WSDL. Má pouze jednu operaci a tou je operace startu procesu. Ostatní operace procesu jsou představovány jednotlivými uživatelskými rozhraními pro jednotlivé uživatelské úkoly. Každé toto rozhraní má vlastní popis ve formátu WSDL.

Funkcionalita procesu byla implementována podle technického designu. Není cílem mé diplomové práce tento proces plně implementovat, aby byl bez dalších úprav nasaditelný a použitelný přímo v komerčním prostředí. Proto nebyla implementována vrstva backendu pro ukládání dat a všechny servisní kroky typu „ulož“ popsané v technickém designu nebyly při realizaci procesu implementovány. Proces si tak drží veškerá data pouze ve svých procesních datových strukturách v rámci procesního úložiště („repository“). Dále nebyl žádný důvod k implementaci procesního kroku pro vygenerování PDF reportu ze služební cesty.

4.2 Implementace klientské webové aplikace

Webová java aplikace je postavena na platformě JSF („Java Server Faces“). Aplikace představuje uživatelské rozhraní procesu. Název aplikace je ClientJSFEAR. Ve vývojovém prostředí IBM Websphere Integration Developer je reprezentována projektem typu webové aplikace. K jejímu vytvoření jsem použil nástroj ve vývojovém prostředí IBM Websphere Integration Developer pro automatické generování klientské aplikace. Takto vzniklou aplikaci jsem rozšířil o funkcionalitu nutnou pro uživatelskou podporu procesu „služební cesta“. Aplikace používá JSF komponenty, které používá například standardní aplikace IBM Business Process Choreographer Explorer pro uživatelskou práci s procesy. Tyto komponenty používají API Business Process Choreographer pro práci s procesy. Tyto komponenty je možné rozšiřovat.

Není předmětem diplomové práce poskytnout plně implementovanou webovou aplikaci pro uživatelskou podporu procesů. Aplikace ClientJSFEAR obsahuje pouze nezbytnou funkcionalitu, pro jednotlivé uživatelské kroky v procesu a podporuje jednotlivé role v procesu. Není lokalizována a není implementován požadavek uživatelské přívětivosti. („user friendly“).

4.3 Nasazení na procesní server

Pro funkčnost procesu je nutné nasadit instalační balíky jednotlivých aplikací Trip a ClientJSFEAR na procesní server. Instalační balíky jsou ve formátu .ear. Jedná se tedy o:

- Trip.ear
- ClientJSFEAR

V konzoli procesního serveru je pak zapotřebí definovat skupinu zaměstnanců s názvem „FN1“, skupinu pracovníků pro péči o lidské zdroje s názvem „HR“ a skupinu zaměstnavatelů s názvem „managers“. Adresář uživatelů je implementován technologií VMM. („Virtual Member Manager“)

5 HOSTOVÁNÍ BUSINESS PROCESU

5.1 Popis cíle diplomové práce v oblasti hostování procesu

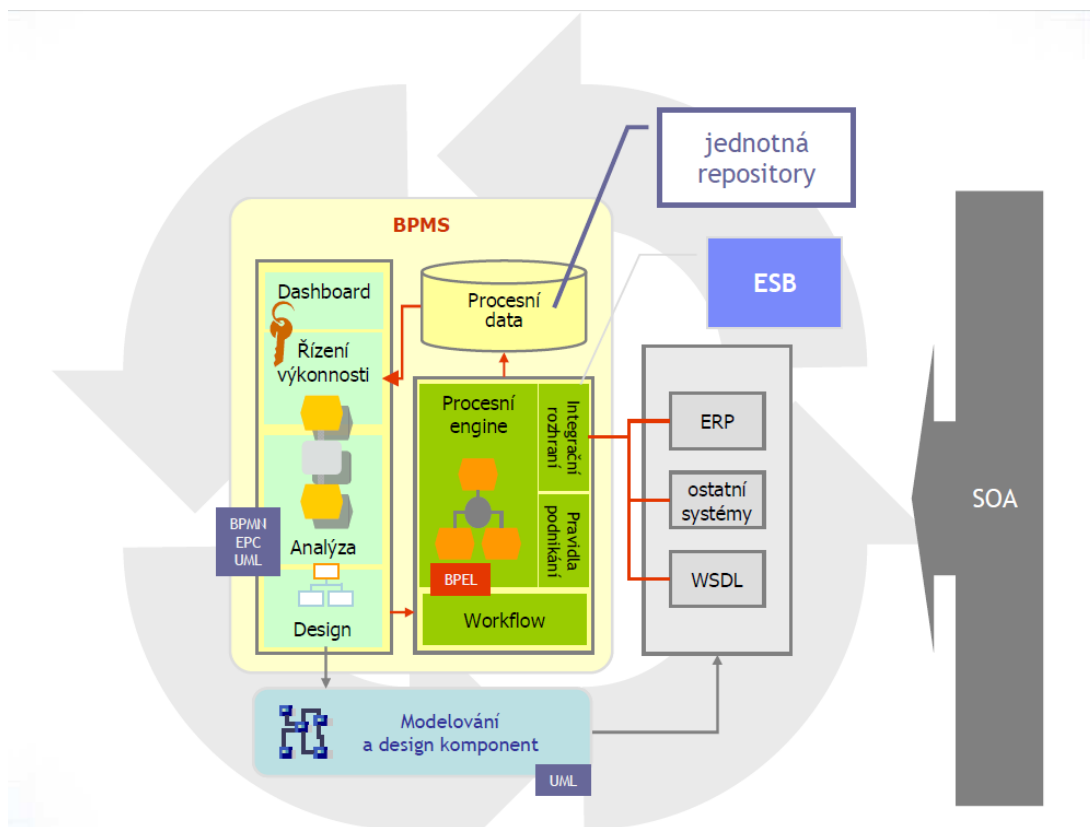
Hlavním cílem v oblasti hostování procesu, vyplývajícím ze zadání diplomové práce a konzultace s vedoucím diplomové práce RNDr. Janem Pavlovičem, bylo probádat možnosti lokálního BPMS hostovat business procesy běžících v rámci jiných BPMS. Hostováním procesu nebylo myšleno pouze volání webové služby vzdáleného procesu. Uvažujeme případ, kdy jsou jednotlivé BPMS na IBM technologiích. Danou problematiku vztaženou k technologiím IBM jsem emailově konzultoval s kompetentní osobou v oblasti BPMS za stranu IBM, panem Branislavem Hudcem.

5.2 Architektura BPMS

Považuji za nutné podrobněji popsat architekturu BPMS. Jedná se o systémy, které podporují vývoj business procesu ve všech fázích životního cyklu. Tedy ve fázích analýzy a designu, implementace, nasazení a monitoringu. Jednotlivé komponenty takového BPMS jsou:

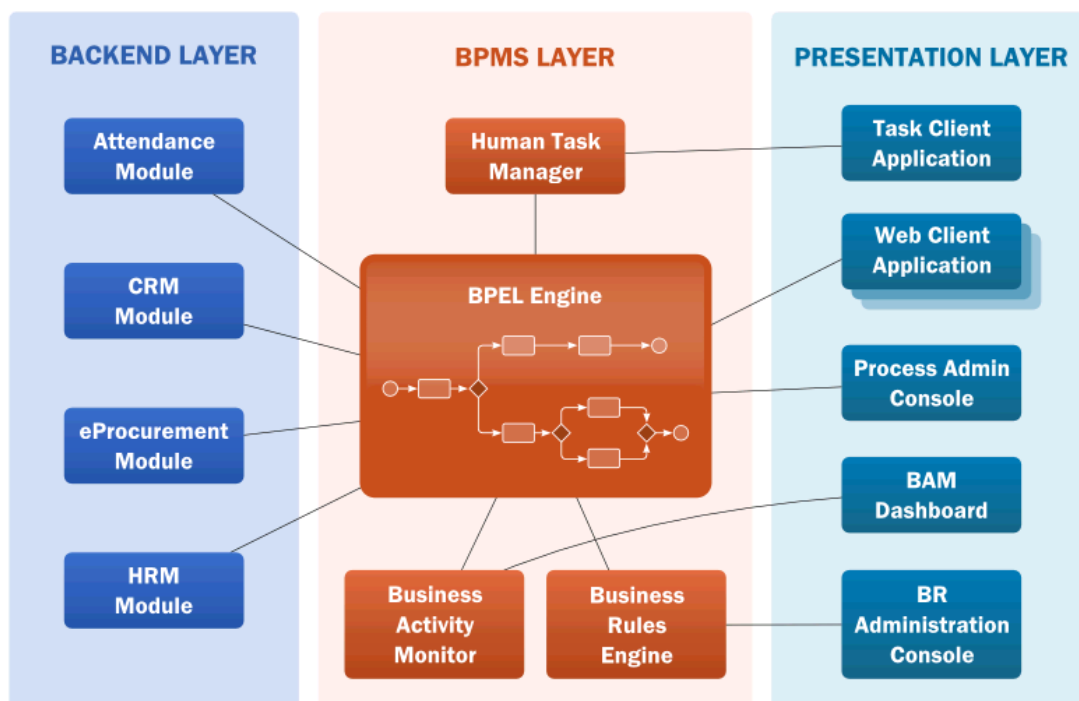
- Modelovací prostředí procesů („Process designer“) Slouží k návrhu a vytváření procesního modelu. Podporuje práci se statistickými daty a umožňuje simulaci procesu před fází implementace.
- Procesní server („Process Engine“) poskytující běhové prostředí procesům. Zároveň poskytuje administrační konzoli tohoto prostředí.
- Komponenta pro podporu uživatelských kroků v procesu („workflow“). Obecně se tato komponenta označuje „Human Task Manager“.
- Komponenta pro business pravidla („Business Rules Engine“). Business pravidla jsou výraz nebo omezení některého z business aspektů. Cílem používání těchto pravidel je řídit nebo ovlivňovat chování businessu, konkrétně v rámci business procesů.
- Komponenta pro monitoring procesů („Business Activity Monitor“)
- Integrovaná vrstva ESB, BI („business integration“) atd.
- Repository metadat obsahující procesní data

„Analýza a design procesů, pravidel, dat, rolí a jiných zdrojů včetně výjimek je přímo propojeno na jednotnou repository - srdce BPMS, které oživuje řízení workflow, pravidel a chování volaných služeb. Navržené změny se tak přes integrační rozhraní přímo promítají do nastavení a chování celého IS. Současně jsou v reálném čase dostupná data o jednotlivých průbězích procesu (kdo, co, kdy, kde, s čím), která se jednak využívají pro management výkonnosti, jednak slouží pro analýzy a simulace při návrzích nových scénářů a testování změn. V případě potřeby nové služby jsou data z modelování přímo čitelná pro vývojové nástroje pro design komponent. Padá tak potřeba duplicit modelů a dat, rolí i nástrojů“



Obrázek č.7 – jednotlivé části BPMS architektury.

Zdroj: skripta z předmětu PV206 na FI MUNI



Obrázek č.8 – BPMS vrstva, Zdroj: skripta z předmětu PV206 na FI MUNI

Procesní data mohou být následujícího charakteru:

1. Režijní data instancí daného procesu
2. Data k monitoringu
3. Data business pravidel
4. Statistická data k simulaci procesu

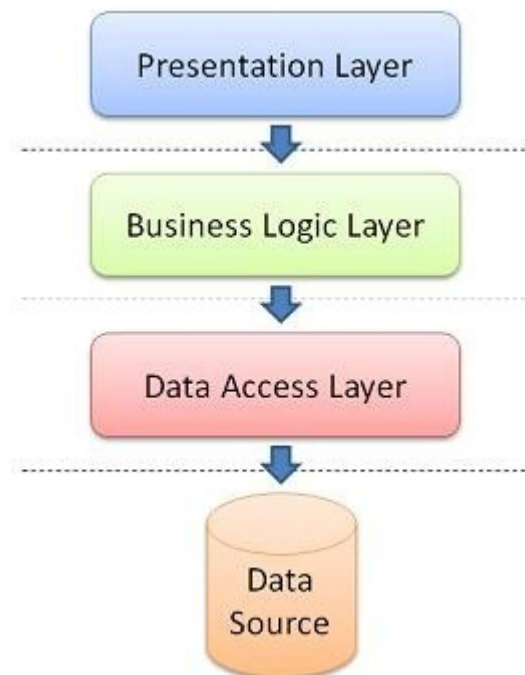
5.3 Definice pojmu „hostování procesu“

Považuji za nutné definovat pojem „hostování procesu“ v kontextu hostování mezi jednotlivými BPMS.

Definice:

Proces A běží v rámci svého BPMS. Nazvěme tento BPMS „BPMS-A“. Dalším BPMS, nazvěme jej „BPMS-B“ je tento proces A hostován tak, že neběží v rámci BPMS-B, ale komponenty BPMS-B k němu mají podobný přístup a poskytují stejnou funkcionalitu, jako by běžel.

Na BPMS se lze dívat jako na aplikaci složenou z jednotlivých komponent. Architektura BPMS může být chápána jako obecná třívrstvá SW architektura. Jednotlivé vrstvy popíšu zvlášť.



Obrázek č.9 – obecná třívrstvá architektura, Zdroj: weblogs.asp.net

V souvislostech pojmu „hostování procesu“ vysvětleného výše provedu teoretickou úvahu nad možnostmi hostování v jednotlivých vrstvách BPMS třívrstvé architektury.

5.4 Datová vrstva („Data Access Layer“)

Datová vrstva je vrstvou přistupující k datovému zdroji. Tento datový zdroj může být představován databázemi, v případě BPMS nazýváme repository. Dalším příkladem datového zdroje je integrační vrstva, pomocí které získáváme data z jiných systémů, případně systémových komponent.

Jedním z požadavků na BPMS je požadavek jednotného repository pro procesy, které v procesním engine tohoto BPMS běží. Tudiž všechna data spjata s procesem by měla být umístěna v jednotném repository stejného BPMS jako je proces engine, na kterém proces běží. Nemá tedy smysl pro proces A mít například režijní data instancí daného procesu, data k monitoringu, statistická data sloužící k simulaci v repository BPMS-A a data business pravidel pro proces A v repository BPMS2.

Docházím tedy k závěru, že jediný správný způsob pro uložení procesních dat je ten, že proces A má všechna data v repository BPMS-A a nemá smysl (de-facto je přímo proti specifikaci BPMS 2. generace), jejich část převádět do repository jiného BPMS, ani nějak duplikovat.

Přes integrační vrstvu, například přes ESB je možné získat procesní data z jiných BPMS. Tento případ rozvinu v popisu dalších vyšších vrstev architektury.

5.5 Vrstva aplikační logiky („Business logic layer“)

Zahrnuje vrstvu aplikační logiky jednotlivých komponent BPMS. Uvažujeme-li o hostování procesu ze vzdáleného BPMS lokálním BPMS. Tzn. Komponenty lokálního BPMS budou moci pracovat se vzdáleným hostovaným procesem stejným způsobem jako s lokálními. K realizaci tohoto požadavku se nabízí 2 řešení:

1. ***Připojit komponenty lokálního BPMS mimo své lokální repository také ke vzdálenému repository vzdáleného BPMS hostujícího procesu.***

Jednotlivé komponenty lokálního BPMS budou mít pak potřebná data pro běh aplikační logiky lokálních i vzdálených procesů. Problémem takovéhoho přístupu je, že repository procesů lokálního BPMS se už nemusí jevit jako jednotné. Chápeme-li totiž repository BPMS jako úložiště procesních dat, která jsou potřebná pro funkčnost komponent daného BPMS, tak používáním repository vzdálených BPMS, dochází k rozšíření lokálně chápaného repository. Tato repository pak obsahuje segmenty repository vzdálených BPMS a nesplňuje tak požadavek na jednotnost repository.

2. ***Připojit komponenty lokálního BPMS ke komponentám vzdáleného BPMS hostujícího procesu.***

Jednotlivé komponenty mohou poskytovat svoji funkcionalitu prostřednictvím webových služeb. Uvažujeme tedy, že funkcionalita komponent vzdáleného BPMS hostující ho procesu je vystavena jako webová služba. Komponenty lokálního BPMS mohou být klienty těchto webových služeb. Tímto způsobem je realizováno požadované propojení komponent lokálního BPMS na komponenty vzdáleného BPMS

hostujícího procesu. Problémem tohoto řešení je fakt, že takovýmto propojení na úrovni komponent není tvořena žádná přidaná hodnota. Na vrstvu aplikační logiky se mimo jiné můžeme dívat jako na vrstvu připravující data pro prezentační vrstvu.

Uvažme příklad, kdy komponenta BPMS pro monitoring procesů představuje funkci **mon**. Tato funkce získává a transformuje data z procesního repository na uživatelsky požadovaná prezentační data ve formě grafů, diagramů, atd. Procesní data procesu A, běžícího na BPMS-A označme **a**. Prezentační data procesu A označme **p**. Platí $p = \text{mon}_A(a)$. V případě propojení komponenty monitoringu lokálního BPMS-B k vzdálené komponentě monitoringu vzdáleného BPMS-A pro hostující proces A, platí pro získání prezentačních dat procesu A rovnice $p = \text{mon}_B(\text{mon}_A(a))$. Jelikož platí $p = \text{mon}_A(a)$, získáme vztah $p = \text{mon}_B(p)$. Monitorovací komponenta BPMS-B, tedy k datům p, které získává na vstupu, nepřidává žádnou hodnotu a pouze je zprostředkovává prezentační vrstvě. Nevidím důvod, proč takového propojení komponent realizovat. Prezenční vrstva si totiž může sama volat $\text{mon}_A(a)$, neboť funkcionalita monitorovací komponenty na BPMS-A je vystavena jako webová služba. Nepotřebuje používat monitorovací komponentu BPMS-B jako zprostředkovatele.

5.6 Vrstva prezentační logiky („Presentation logic layer“)

V rámci BPMS chápeme prezentační vrstvu jako prostředí klientských aplikací, kterými se připojujeme a používáme komponenty daného BPMS. V této vrstvě ztrácí výše popsaná definice „hostování procesu“ význam. Definice:

Proces A běží v rámci svého BPMS. Nazvěme tento BPMS „BPMS-A“. Dalším BPMS, nazvěme jej „BPMS-B“ je tento proces A hostován tak, že neběží v rámci BPMS-B, ale komponenty BPMS-B k němu mají podobný přístup a poskytují stejnou funkcionalitu, jako by běžel.

V této vrstvě se ale žádné komponenty BPMS nevyskytují. Pouze jednotlivé klientské aplikace. Místo o hostování procesu se bavme tedy o integraci klientských aplikací na komponenty BPMS. Jak už jsem zmínil výše, funkcionalita těchto komponent může být vystavena pomocí webových služeb. Klientské aplikace prezentační vrstvy pak vystupují v roli klienta těchto webových služeb.

5.7 IBM BPMS Komponenty

Do teďka jsem pojednával o komponentách BPMS obecně, nicméně realizace procesně orientovaného řešení méj diplomové práce je postaveno na technologiích IBM.

Jednotlivé IBM BPMS komponenty:

- Procesní server („engine“) : IBM Websphere Process Server V6.2
- Komponenta pro business pravidla: Business Rules Manager
- Komponenta pro monitoring procesů : IBM Websphere Monitor Server V6.2

V současné době IBM technologie nepodporují hostování procesů popsaných v části datové vrstvy a v části vrstvy aplikační logiky. Jelikož jsem ve své práci nenalezl důvod k takovému přístupu, nevidím přínosné tento požadavek v IBM technologiích implementovat. IBM klientské aplikace jednotlivých BPMS komponent (Process server Console, Business Process Manager Client Application, BAM Dashboard) nemají vestavěnou uživatelskou podporu pro integraci na vzdálené BPMS.

Jediným řešením na IBM technologiích jak k těmto vzdáleným procesům přistupovat, je jako k outsourcovaným procesům. Proces běžící na vzdáleném BPMS vystaví svoji vlastní funkcionalitu, proces management funkcionalitu, monitoring funkcionalitu, atd., pomocí webových služeb. Klientské aplikace pak mohou k těmto službám přistupovat. Panem Branislavem Hudcem bylo toto řešení označeno jako řešení používané v praxi při implementaci požadavků zákazníku IBM.

6 ZÁVĚR

Prostudování obecné problematiky Business Proces Managementu mi pomohlo pochopit základy této manažerské disciplíny a motivaci, proč takovýto přístup používat. Tento pohled jsem rozšířil o technické aspekty a namapování problematiky Business Proces Managementu na SOA architekturu. Své nabyté poznatky shrnuji v prvních dvou kapitolách

Z domény docházkového systému jsem si vybral jeden proces. Proces „služební cesta“. Tento proces jsem zanalyzoval a provedl detailní technický návrh. Na základě tohoto návrhu jsem naimplementoval aplikaci tohoto procesu, včetně klientské webové aplikace představující uživatelské rozhraní procesu. Pro fázi implementace a běh procesu jsem použil IBM BPMS komponenty IBM Websphere Integration Developer a IBM Websphere Process Server. Popis návrhu a implementace popisují v dalších dvou kapitolách

Za jeden z hlavních přínosů své práce považuji zpracování problematiky hostování procesu. Abych mohl dojít k nějakým závěrům v této oblasti, musel jsem se zorientovat v architektuře BPMS a konkrétně pak v její realizaci společností IBM. Tuto architekturu detailně popisují a hledám vhodnou formulaci definice „hostování procesu“. Ve smyslu této definice procházím jednotlivé vrstvy BPMS a hledám způsoby realizace takového hostování. U všech nalezených způsobů jsem položil argument, proč není takovýto přístup vhodné realizovat. Proto přecházím od definice hostování procesu k outsourcingu procesu.

Jako další pokračování v mé práci vidím právě rozpracování oblasti outsourcingu procesu a dále pak prověření možností BPMS komponent IBM poskytovat svoji funkcionalitu prostřednictvím webových služeb.

7 LITERATURA

- Mike Havey: Essential Business Process Modeling, O'Reilly, August 2005
- Matjaz B. Juric: Business Process Execution Language for Web Services BPEL and BPEL4WS 2nd Edition, Packt Publishing, January 2006
- Sanjiva Weerawarana: Web Services Platform Architecture: SOAP, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging, and More
- Skripta předmětu PV206 na FI MUNI.
- Model zrání procesů
www.wikipedia.com
- Tutoriály na IBM Integration Developer a IBM Process Server
www.ibm.com
- BPM a IBM
www.procesy.cz

8 PŘÍLOHY

Příložené DVD obsahuje:

- Nasaditelný balík Trip.ear
- Nasaditelný balík ClientJSFEAR.ear
- Obrázek procesního diagramu procesu „služební cesta“
- Obrázek datového modelu procesu „služební cesta“
- Zabalený projekt Trip z vývojového prostředí IBM Integration Developer
- Zabalený projekt ClientJSFEAR z vývojového prostředí IBM Integration Developer
- Elektronická podoba této diplomové práce ve formátu .pdf a .doc