

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

TRANSFORMACE MODELŮ BYZNYS PROCESŮ

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

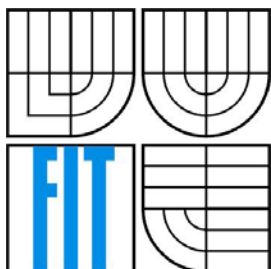
AUTOR PRÁCE
AUTHOR

JAN WOHLGEMUTH

BRNO 2008



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

TRASFORMACE MODELŮ BYZNYS PROCESŮ

TRANSFORMATION OF BUSINESS PROCESS MODELS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JAN WOHLGEMUTH

VEDOUCÍ PRÁCE
SUPERVISOR

ING. PETR WEISS

BRNO 2008

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav informačních systémů

Akademický rok 2007/2008

Zadání bakalářské práce

Řešitel: **Wohlgemuth Jan**

Obor: Informační technologie

Téma: **Transformace modelů byznys procesů**

Kategorie: Softwarové inženýrství

Pokyny:

1. Seznamte se s problematikou modelování byznys procesů, především s jazykem Business Process Modeling Notation (BPMN). Dále se seznamte s XML a XMI.
2. Seznamte se s metodikou převodu modelů byznys procesů do UML.
3. Zkoumejte možnosti volně dostupných modelovacích nástrojů podporujících BPMN vzhledem k exportu grafických modelů do XML nebo XMI. Zhodnoťte použití XML exportů vzhledem k bodu 4 zadání, případně navrhněte vlastní schéma exportování.
4. Navrhněte a následně implementujte nástroj, který bude provádět výše uvedený převod na úrovni XML (XMI) popisů.
5. Zhodnoťte dosažené výsledky a diskutujte možná rozšíření.

Literatura:

- Quin, L.: *Extensible Markup Language (XML)*. W3C. 2006. Dostupný z: <http://www.w3.org/XML/>
- White, S. A.: *Introduction to BPMN*. IBM Corporation. Dostupný z <http://www.bpmn.org/>
- OMG: *MOF 2.0 / XMI Mapping Specification*. 2005. Dostupný z <http://www.omg.org/technology/documents/formal/xmi.htm>
- Amsden, J.: *Business services modeling*. IBM, 2005. Dostupný na http://www.ibm.com/developerworks/rational/library/05/1227_amsden/

Při obhajobě semestrální části projektu je požadováno:

- Body 1 až 3 zadání.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese <http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Weiss Petr, Ing.**, UIFS FIT VUT

Datum zadání: 1. listopadu 2007

Datum odevzdání: 23. ledna 2008

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav informačních systémů
602 00 Brno, Božetěchova 2

doc. Ing. Jaroslav Zendulka, CSc.
vedoucí ústavu

Abstrakt

Tato práce se zabývá studiem byznys procesů, možnostmi jejich modelování a zejména využití notace BPMN při jejich modelování. Dále se zde dočtete o možnosti jejich reprezentace v textové formě pomocí rozšiřujícího značkovacího jazyka XML a výhody a nevýhody jeho použití. Hlavní náplní této práce je transformace modelu zapsaného v notaci BPMN do standardu jazyka UML. Tato transformace se provádí právě na úrovni XML popisků. Její součástí je jednoduchá aplikace, která dokáže daný převod provádět na již zmíněné úrovni.

Klíčová slova

Byznys proces, BPMN, XML, XMI, transformace, UML, byznys model

Abstract

This work deals with research of business processes modeling, especially with using of the BPMN notation. Further, this work discusses representation of business processes in the text form using the extensible markup language – XML; its advantages and disadvantages. The main content of this work is the transformation of a model denoted in the BPMN notation into the UML language standard. This transformation is made at the level of XML specification. A simple application that can perform the given transformation is a part of this work.

Keywords

Business process, BPMN, XML, XMI, transformation, UML, business model

Citace

Wohlgemuth Jan: Transformace byznys procesů. Brno, 2008, bakalářská práce, FIT VUT v Brně.

Transformace modelů byznys procesů

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Petra Weisse.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Jan Wohlgemuth
9. 5. 2008

Poděkování

Rád bych na tomto místě poděkoval vedoucímu své bakalářské práce Ing. Petru Weissovi, který mi ochotně radil a pomáhal v průběhu celého zpracování.

© Jan Wohlgemuth, 2008.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

Obsah	1
Úvod	3
1.1 Motivace projektu	3
1.2 Cíle projektu	3
1.3 Rozdělení kapitol	4
2 Byznys procesy	5
2.1 Modelování byznys procesů	5
2.1.1 Business Process Modeling Notation	6
3 Extensible Markup Language	12
3.1 Vývoj značkovacího jazyka	12
3.2 Vznik XML	12
3.3 Struktura a formát dokumentů	12
3.4 Aplikace XML	13
3.4.1 RDF	13
3.4.2 WML	13
3.5 XMI	14
4 Transformace BPMN modelu do UML	15
4.1 Důvody transformace	15
4.2 Jednotlivé entity byznys procesu	17
4.2.1 Vlastní projekt	17
4.2.2 Proces	17
4.2.3 Role	17
4.2.4 Úkol	18
4.2.5 Propojení	18
4.2.6 Začátek a konec procesu	18
4.2.7 Rozhodnutí / spojení	18
4.2.8 Rozvětvení / sdružení	18
4.3 Výsledek transformace	18
4.4 Modelovací programy	19
4.5 Úroveň XML/XMI popisků	20
5 Popis vlastní implementace	22
5.1 Volba prostředí	22
5.2 Volba modelovacího programu	22
5.3 Modularita programu	22

5.4	Vlastní řešení.....	23
Závěr.....		27
Literatura		28
Seznam příloh		29

Úvod

Procesy jsou všude kolem nás. Vždyť i psaní této práce je proces, do kterého vstupují rozličné role (moje osoba, vedoucí bakalářské práce). Jeho jasně definovaným cílem je vytvoření tohoto dokumentu. Zároveň se jedná o sub-proces v procesu mého vysokoškolského vzdělání, jehož cílem je stát se vzdělanějším, moudřejším a lépe připraveným pro svou budoucí praxi. V neposlední řadě také získat vysokoškolský titul.

V následujícím textu se ale nebudeme věnovat tak triviálním a známým procesům, jako je moje studium. Zkusíme se podívat na procesy z pohledu obchodních společností, odtud označení byznys procesy.

1.1 Motivace projektu

Hledal jsem zadání, jimž bych se mohl co nejvíce přiblížit světu, který je mi blízký, tedy obchodu a službám, ale zároveň využil poznatků a dovedností získaných na domovské fakultě. Hledal jsem propojení byznysu s informačními službami.

S nápadem zabývat se byznys procesy přišel právě vedoucí mé práce Ing. Petr Weiss. První myšlenka vedla k tomu, vytvořit možný zápis byznys procesu v textové podobě, ve formě popisků XML. Tato činnost měla navázat na jeho zaměření na webové služby a celkově na architektury orientované na služby (SOA). Vzhledem ke skutečnosti, že jsme společně vypátrali programy, jež uměly BPMN modelovat a zároveň tyto modely exportovat do XML, ztrácel by význam mé práce smysl.

Z tohoto důvodu se zrodil úplně nový cíl mého projektu, který ale neodbíhal od původního tématu, tedy byznys procesů.

Poznámka: více o systémech SOA (Service-Oriented Architecture) se dočtete v [4].

1.2 Cíle projektu

Novým, ještě zajímavějším a snad i užitečnějším cílem se stala transformace byznys procesu z jedné možné notace do jiné. Transformaci nelze provádět jinak, než právě na úrovni textové reprezentace – tedy XML. V tuto chvíli jsem tedy veškeré své síly vynaložil k získání co nejvíce poznatků o samotných byznys procesech, o notaci, ve které se zakreslují, ale také o jazyku, do kterého jsem chtěl byznys procesy převádět, tedy UML.

Výsledkem projektu měla být aplikace, která transformuje XML zápisky byznys procesu do XMI, tedy XML jazyka pro popis diagramů UML. Taková aplikace měla být naprogramována hlavně

s důrazem na co největší modularitu, znovupoužitelnost, a to zejména z důvodu neustálého vývoje standardů pro tyto služby.

1.3 Rozdělení kapitol

Práce obsahuje celkem šest hlavních kapitol. První z nich je tento stručný úvod a nastínění obsahu práce. Následuje stěžejní část teoretické přípravy – kapitola věnovaná samotnými byznys procesům, jejich modelováním a seznámení s notací BPMN. Třetí kapitola je taktéž teoretická a seznámí nás se základy značkovacích jazyků, XML a XMI a jejich praktickému využití. V pořadí čtvrtá kapitola nám představuje, jakým způsobem se byznys procesy převádějí, problémy s tím spojené a způsob, jakým je možné tuto transformaci provést. Pátá kapitola popisuje vlastní implementaci této transformace, objevují se v ní informace, jak jsem tento převod prováděl, jaké technologie a software k tomu použil a s jakými problémy jsem se při této práci setkal. Poslední šestou kapitolou je stručný závěr, jenž obsahuje informace o možném dalším vývoji tohoto projektu.

2 Byznys procesy

Byznys proces, možná raději v anglickém originále business process, je jakási sekvence kroků, která ve větší či menší míře abstrakce vede k dosažení určitého předem definovaného cíle. Ve své podstatě proces je pouze označená a řádně pojmenovaná organizovaná činnost.

Zjednodušeně nám popsány a zavedeny procesy popisuje kdo, co a v jakém pořadí má dělat, aby společnost uspokojila zákazníka, potažmo dosáhla zisku. A to zisku ať hmotného či nehmotného. Činnost, která se zabývá popisováním a analyzováním procesů, se nazývá Business Process Management (BPM). Je to tedy snaha ovlivnit předmět našeho zájmu (podnikání) žádoucím způsobem.

Integrace, jak se popsání a zavedení těchto procesů říká, je také nutným předpokladem pro efektivní zavedení informačních a komunikačních technologií (ICT). Správně popsané firemní procesy jsou nezbytným předpokladem pro vývojáře, pro vytvoření správných a efektivních prostředků pro ICT podporu podnikání.

Systémy založené na byznys procesech využívají další z jazyků, z nichž některé jsou zmíněny v této práci. Tyto systémy se využívají jak pro návrh a tvorbu podpůrného softwaru, tak pro řízení a management všech procesů.

Pokud se zaměříme na menší společnosti, kde nemusí být tato podpora hlavním podpůrným prostředkem, tak i zde najdeme více než jeden pádný důvod, proč popisovat kroky vedoucí k vytváření zisku společnosti. Pokud například nebude mít malá společnost popsané firemní procesy, jedná se zde pouze o činnost založenou na jednotlivých konkrétních osobách, které na utváření zisku pracují, a společnost na těchto lidech „stojí a padá“. Tudíž s jejich odchodem již nemá společnost téměř žádnou hodnotu.

Pokud se tomuto chceme vyvarovat nebo pokud se chystáme společnost prodat, jsou pro nás vhodně popsané firemní (interní i externí) procesy velmi důležité.

2.1 Modelování byznys procesů

V této kapitole bych se chtěl podívat jak, takové firemní procesy ve skutečnosti vypadají, jak se vytvářejí – modelují a zakreslují.

Jak jsme se již dozvěděli, namodelovaný proces popisuje kdo, co a v jaké časové posloupnosti dělá pro dosažení cíle. Z toho nám vyplývají základní prvky, které v takovém procesu vystupují.

- role
- události
- aktivity
- rozhodovací brány

K samotnému zakreslení procesů je nejdříve důležité zvolit vhodnou notaci, tedy konkrétní popis pravidel a zobrazovaných značek, které budeme používat pro naše procesy. Pokud již máme vybranou notaci (v našem případě BPMN), nezbývá, než se poohlédnout po programovém nástroji, který bude námi zvolenou notaci podporovat. Pro budoucí zpracování v této práci bylo důležité najít nástroj, který podporuje export obrazové prezentace procesu do textového formátu, nejlépe XML.

2.1.1 Business Process Modeling Notation

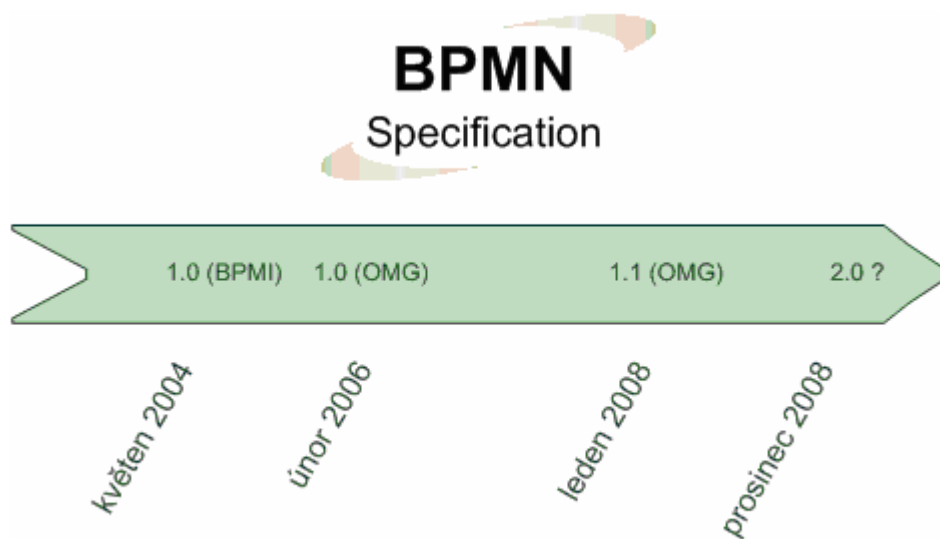
Business Process Modeling Notation (BPMN) je standardizovaná grafická reprezentace pro kreslení obchodních procesů ve firemní sféře. BPMN se snaží zjednodušit čitelnost pro všechny účastníky životního cyklu procesu, ať už se jedná o vrcholné manažery, systémové analytiku, business analytiku nebo vývojáře.

2.1.1.1 Vznik

Za vznikem této notace stála iniciativa BPMI (Business Process Management Initiative), která se snažila zacelit komunikační díru mezi návrhem a následnou implementací procesu. Původně vyvinutý jazyk BPML (jazyk založený na XML) potřeboval pro svou realizaci grafickou podobu, jímž se právě BPMN stala. Jazyk BPML byl později nahrazen jazykem BPEL určeným výhradně pro realizaci byznys procesu. BPMN notaci následně v roce 2006 přebírá OMG (Object Management Group), respektive BPMI se stává součástí OMG.

2.1.1.2 Verze

Aktuální verze BPMN je 1.1, která byla oficiálně přijata v lednu letošního roku (2008). Od původní 1.0 se však příliš neliší a právě proto je zatím daleko rozšířenější její předchůdkyně z února 2006. V chystané specifikaci 2.0, která má být představena koncem roku, by se měly projevit zásadnější změny.



Obrázek 1 Verze BPMN

2.1.1.3 Prvky BPMN diagramu

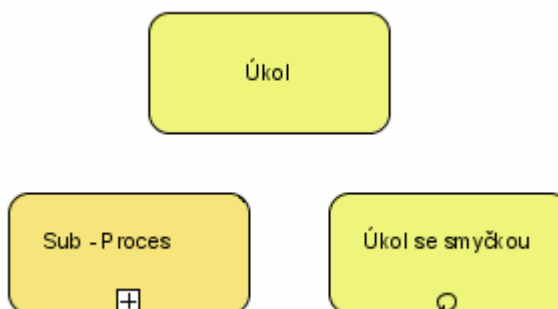
Nejdříve popíšeme základní prvky diagramu. Jedná se o aktivity, události, brány a spojení. Viz obrázek 2.



Obrázek 2 BPMN - základní prvky

Aktivita

Aktivita je činnost vykonaná během procesu. Může se vyskytovat jako nejmenší jednotka práce, která se již nedá dále specifikovat, nebo činnost složená z dalšího procesu. Potom se jedná o sub-proces, který je specifikovaný v dalším diagramu nebo jiném projektu, na který se odkazuje. Aktivita se označuje jako obdélník se zakulacenými rohy. Pro případ sub-procesu se používá značka plus. Aktivita nebo sub-proces může být vykonán jednou nebo může mít definované opakování. Potom označíme aktivitu šipkou v kolečku, značící smyčky, ve kterých je daný úkol definovaný.



Obrázek 3 BPMN – aktivity

Události

Událost je něco, co nastane v průběhu byznys procesu. Události mají vliv na běh procesu. Mohou jej začínat, přerušit nebo ukončit. Události se v BPMN označují kolečkem. Pokud je okraj jednoduchý, jedná se o počáteční stav. Dvojitý okraj značí událost v průběhu procesu. Kolečko s tučným okrajem je koncová událost.



Obrázek 4 BPMN – události

Počáteční událost neboli „start event“ nám značí, kde proces začíná. Existuje několik druhů počátečních událostí navzájem odlišených pomocí obrázku uvnitř kolečka. Tyto specifické počáteční události přesněji specifikují, co způsobí počátek procesu.



Obrázek 5 BPMN - počáteční události

Pokud se v kolečku nenachází žádný obrázek, jedná se buď o počátek sub-procesu nebo o počátek, který není blíže specifikovaný. Zpráva, časovač nebo pravidlo znázorňuje událost, která daný proces spouští. Speciální počáteční událost „link“, která slouží pro napojení procesu do jiného místa diagramu, bude odstraněna v nové verzi BPMN, takže pro nás již nemá význam. Poslední značka s hvězdičkou v kolečku označuje počátek větvení procesu.



Obrázek 6 BPMN - události v průběhu procesu

Tyto události se nacházejí kdekoli mezi počátkem a koncem procesu. Mohou být normálně v průběhu proudu procesu (jako standardní prvky) nebo mohou být „nalepeny“ na okraj aktivity, čímž se značí začátek vykonávané aktivity.

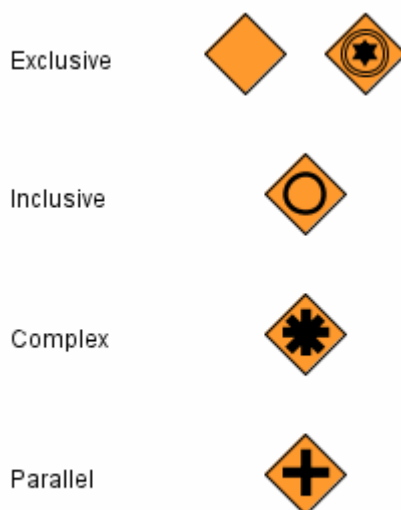


Obrázek 7 BPMN - koncové události

A konečně „end event“ – koncové stavy indikují, kde bude proces končit. Mohou nastat různé výsledky, které značí rozlišné okolnosti, za kterých proces končí. Obdobně jako u počátečních stavů, se neoznačený koncový stav (bez obrázku) používá pro sub-proces nebo pokud není blíže specifikovaný. O události označené jako „link“ zase jen tolik, že již v příští verzi BPMN bude nahrazena jinou alternativou.

Brány

Brány neboli „gateways“ jsou modelovací prvky, které se používají k řízení toku událostí při rozbíhání nebo sbíhání více vláken procesu. Pro všechny typy bran se používá čtverec natočený o 90 stupňů. Obrázek uprostřed tohoto čtverce značí různé typy chování. Všechny typy bran tok procesu buď rozbíhají do více nebo ho naopak sdružují do jednoho jediného.



Obrázek 8 BPMN - rozhodovací brány

Rozdělení druhů rozhodovacích bran není ještě v aktuální verzi úplně stoprocentně definováno. Pokusím se nastínit pravidla pro použití jednotlivých druhů.

První dvojice bran (bez obrázku a s obrázkem mnohonásobné události) se používá všude tam, kde je nutné se rozhodnout mezi dvěma nebo více možnými cestami. Z těchto cest se vydáme výlučně po jedné, která odpovídá buď nějaké podmínce (prázdný čtvereček nebo v alternativní notaci velké tiskací „X“) nebo na základě nějaké události (s obrázkem hvězdičky ve dvojitém kruhu). Jedná se o nejčastější typ brány. Zjednodušeně se o něm dá říct, že se jedná o „křižovatku na cestě“ a my se vydáme pouze po jedné z možných cest.

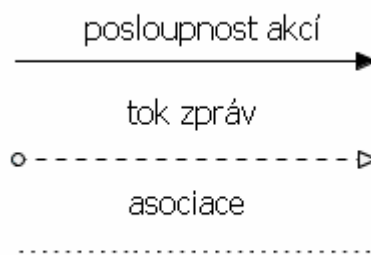
Druhou v pořadí je brána označená uvnitř čtverečku kolečkem a v BPMN je popsána jako „inclusive“. Odvození je jednoduché, tato brána umožňuje vydat se po více možných cestách současně. Zpravidla bývá později následována odpovídající „inclusive“ branou pro opětovné sjednocení možných cest.

„Complex“ označuje bránu, která rozhoduje o více možných variantách. Je možné jít po jedné nebo více cestách, podle toho, kterému pravidlu odpovídá. Tato brána je opět použitelná jak pro rozbíhání, tak pro následné sbíhání více větví procesu.

Poslední zbývající brána je určená pro začátek nebo konec paralelního procesu – označení „parallel“. Je označená velkým tučným znaménkem „+“. Ve většině situací se však bez něj obejdeme, notace BPMN totiž umožňuje rozbíhat více procesů i bez použití jakékoliv brány. Její použití slouží spíše pro větší kontrolu a synchronizaci těchto paralelních cest.

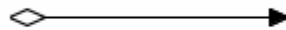
Spojení

BPMN nám nabízí tři možnosti propojení objektů v diagramu.



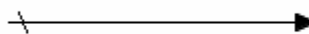
Obrázek 9 BPMN – spojení

Posloupnost akcí, možná lépe anglicky „sequence flow“, popisuje v jaké posloupnosti jsou všechny úkoly vykonávány. Start a cíl musí být přiřazen vždy k jednomu z těchto objektů: události, aktivity a brány. Posloupnost akcí nesmí nikdy přetnout hranice sub-procesu nebo hranice „poolu“ (pojem pool je vysvětlen později). Dále je také v této notaci umožněno přiřazovat jednotlivým cestám podmínky, za kterých se mají vykonat. To je možné pouze tehdy, pokud existuje více než jedna možná cesta. Taková podmínka se označí malým kosočtvercem na straně aktivity, ze které daná cesta vychází, jak je zobrazeno na obrázku 10.



Obrázek 10 BPMN - cesta s podmínkou

Zároveň pokud používáme „inclusive“ nebo „exklusive“ bránu, měla by mít definovanou výchozí cestu. Tuto cestu označíme malou čárkou, přesně jak zobrazuje následující obrázek.



Obrázek 11 BPMN - výchozí cesta

„Message flow“ neboli tok zpráv se používá k zobrazení toku zpráv mezi dvěma entitami, které jsou připraveny ji odeslat a přijmout. Tyto entity bývají zpravidla dva účastníci procesu, jež v BPMN znázorňují dva oddělené bazény. Tok zpráv se může tedy vázat buď na jednotlivé bazény nebo na objekty (aktivity) mimo bazén.

Asociace potom slouží k přiřazení určitých dat, informací nebo výrobků k jinému objektu.

Modelování rolí

Názvy pro oddělení jednotlivých rolí trochu vybízí k letnímu dovádění v bazénu, ale nebude tomu tak. „Swimlanes“ v BPMN pomáhají rozlišit jednotlivé části. Swimlane se dělí na dva typy, jednou je „pool“ (bazén) a každý bazén může obsahovat jednu nebo více „lane“ (dráha). Bazén reprezentuje účastníka působícího v procesu. Dráhy potom označují dílčí účastníky pro objekty v jednom konkrétním bazénu. Často se k tomuto dělení přiřazuje konkrétní fyzické dělení dané organizace, pro kterou jsou byznys procesy modelovány. Potom bazén většinou odpovídá sekci organizace a dráhy jednotlivým rolím uvnitř této sekce.



Obrázek 12 BPMN – pool (bazén)

Další prvky

BPMN obsahuje ještě další prvky, které jsou určeny ke zvýšení přesnosti popisu daného procesu. Můžeme použít například textovou asociaci k popisu některých špatně srozumitelných částí. Dále zmíním ještě důležité datové objekty, kterými jsou například dokumenty nebo obecně produkty daných aktivit (faktura, dopis, email, apod.).

2.1.1.4 Shrnutí

Notace BPMN nabízí spoustu možností rozšíření, které přesahuje rámec této práce. Více se o ní můžete dočíst v [1]. Aktuální specifikaci můžete získat na www.bpmn.org. I toto mu pomáhá být stále aktuální a použitelné. V současné době je jakýmsi standardem v modelovací notaci byznys procesů.

3 Extensible Markup Language

Proč je důležitý? Žijeme v informační době a nedílnou součástí informační společnosti je i praktické a rychlé sdílení dat a vyhledávání v nich. Dokud jsme pro ukládání dat používali pouze proprietární formáty jednotlivých aplikací, bylo takové sdílení nekomfortní, pomalé a náročné z důvodu nekompatibility formátů jednotlivých softwarových aplikací. Při studiu této kapitoly jsem čerpal z [2] a [3].

3.1 Vývoj značkovacího jazyka

Potřeba mít jazyk umožňující popsat strukturu a obsah textu bez jeho grafické podoby vznikla hlavně rozvojem internetu. Kde více je potřeba, aby stejný obsah bylo možné zobrazit v různých grafických úpravách, případně jako čistý text pro účely tisku. Svým způsobem to již umožnil jazyk HTML (HyperText Markup Language), ale tento jazyk obsahuje spoustu nevýhod, které právě Extensible Markup Language (XML) nemá. Největší nevýhodou je absence možnosti vytvářet vlastní značky.

Asi prvním známým značkovacím jazykem byl GML (Generalized Markup Language), který si vytvořil tým lidí z IBM pro lepší ukládání a následné využití právních textů uvnitř společnosti. Za přispění organizací ANSI a GCA (Graphics Communications Association), jež měly podobné cíle, vznikl jazyk SGML (Standard Generalized Markup Language), který byl definován již v normě ISO z roku 1986.

Standard SGML byl opravdu hodně obecný, podporoval volbu vlastních značek i mnoho dalších předvoleb. Asi nejznámějším využitím je právě jazyk HTML, který pomocí DTD (definice typu dokumentu) určoval značky, které může dokument obsahovat. U HTML se typ dokumentu měnil s každou novou verzí.

3.2 Vznik XML

XML vznikl odnoží té nejpoužívanější části rozsáhlé definice SGML. Jejím duchovním otcem je konsorcium W3C, které se zabývá vývojem standardů pro web. Jejich cílem je co největší rozšiřitelnost všech služeb WWW. Díky tomu je standard XML volně k dispozici, využitelný ve vlastních aplikacích.

3.3 Struktura a formát dokumentů

Dokument XML obsahuje specifické instrukce, nazývané „tagy“, „elementy“ a „entity“, které ohraničují části dokumentu.

```

<kniha>
  <nazev>Nayev knihy</nazev>
  <kapitola>
    <nazev>Kapitola 1</nazev>
    <obasah>Obsah kapitoly 1 muze obsahat jakykoliv text</obasah>
  </kapitola>
</kniha>

```

Obrázek 13 Ukázka XML

XML je ve skutečnosti metajazyk, který se používá k popisu dalších jazyků. Neexistuje seznam předdefinovaných tagů, každý si může pro svou vlastní aplikaci popsat své vlastní značky a strukturu, ve které bude pomocí těchto značek ukládat informace. V takovýchto aplikacích se XML nejlépe hodí právě pro export a následný import dat. Dokumenty ve formátu XML můžeme upravovat pomocí speciálních programů pro jejich čtení, ale i pomocí jakéhokoliv textového editoru. Zároveň již existují nástroje pro různé programovací jazyky podporující tento standard a umožňující s ním pracovat, jako s funkčním DOM modelem, tedy vnořovat či mazat celé elementy dokumentu, řadit je, apod.

3.4 Aplikace XML

Kdykoliv je potřeba výměna komplexních dat mezi aplikacemi, nabízí se XML jako nejvhodnější formát dat. Dalším oborem, kde je použití výhodné, jsou relační databázové systémy. V jejich případě jsou XML tagy použity během přenosu záznamů, polí a vztahů mezi systémy jako jakési data interface.

3.4.1 RDF

RDF (Resource Description Framework), jenž je výsledkem více zájmových skupin opět pod záštitou konsorcia W3C, nabízí robustní a flexibilní architekturu pro výměnu metadat na internetu. RDF jako aplikace XML poskytuje významný prostředek k zajištění vzájemné součinnosti metadat v podobě různých XML namespace (jmenných prostorů). To umožňuje použití různých typů značek a jejich atributů v jednom dokumentu.

3.4.2 WML

Doba pokročila dále a internet již není doménou jen stolních počítačů obřích rozměrů (hlavně obrazovek velkých rozměrů). S postupem času začal vstupovat do většiny elektronických zařízení, mobilní telefony nevyjímaje. Pro potřeby menších obrazovek a nižších možností datových přenosů bylo zapotřebí mít jazyk k tomuto účelu upravený. Protokol WAP, který se o přenos internetu na

mobilní telefony stará, využívá právě jazyk WML (Wireless Markup Language), který je založen na XML a definován pomocí DTD [3].

3.5 XMI

XMI je zkratkou výrazu XML Metadata Interchange. XMI je standard pro výměnu informací ve formě metadat založený na Extensible Markup Language, tedy XML. Můžeme ho použít pro jakákoliv metadata, která popisují metamodel vyjadřující Meta-Object Facility (MOF). Nejčastější využití nachází jako formát pro výměnu dat UML modelů.

Za vytvořením jazyka stojí skupina OMG a několik předních softwarových firem. XMI umožňuje přenášet data UML popisující návrhy informačních systémů, softwarové komponenty, schémata databází mezi různými systémy, což dříve nebylo možné z důvodu velké rozlišnosti formátů jednotlivých systémů.

4 Transformace BPMN modelu do UML

Tématem mé bakalářské práce je právě samotný převod diagramu zakresleného v notaci BPMN do standardizované a v praxi používané UML verze 2.0. Před samotným započítím práce bylo nejdříve potřeba prostudovat teorii tohoto převodu. Bohužel do této doby neexistoval žádný přesný předpis na to, jak převádět přímo notaci BPMN do jazyka UML. Při mé práci jsem vycházel z jiných dostupných postupů od konkrétních společností. Příklad společnosti IBM [6].

4.1 Důvody transformace

Důvod pro potřeby převodu modelu z jedné notace do druhé může být několik. Někdo může mít problém porozumět některé z nich, v některých případech nemusí jedna BPMN dávat jasný pohled na všechny aspekty procesu. Důvody mohou být rozličné. Mě osobně při této práci vyvstaly dva podstatné.

První důvod je spíše obecný. Jedná se o to, že mezi business komunitou a IT komunitou existuje stále dost velká bariéra, která zabraňuje lepší prosperitě obou. Lidé z té techničtější strany barikády mají problémy porozumět problémům předkládaným ze strany lidí z byznysu a naopak, lidé z byznysu nedokáží přesně specifikovat svoje zadání pro potřeby analytiků a návrhářů z řad IT pracovníků.

Jakýsi styčný bod v řeči těchto dvou stále přibližujících se skupin by se mohl nacházet právě v jazyce, kterému programátoři musejí dobře rozumět. Navíc nabízí řadu možností pro zobrazení dané problematiky ze všech stran a umožňuje efektivně pracovat s návrhem aplikací pro potřeby zadavatelů.

Druhý aspekt či spíše výhodu mi zprostředkoval Simon Perry [5]. Ve svém článku zabývajícím se modelováním byznys procesů a srovnáním následných pohledů na namodelovaný proces se věnuje právě srovnání procesu namodelovaného pomocí těchto dvou notací.

Toto jeho srovnání zpracoval do přehledné tabulky možných způsobů pohledu na proces. Zobrazím zde jeho původní podobu v anglickém jazyce. Hned za ní bude následovat můj český komentář.

View	UML Representation	BPMN Representation
Requirements view	✓ <i>Use case diagram</i> : requirements shown as use cases, stakeholders as actors	✗ <i>No representation</i> . Stakeholder information only found as swimlanes (pools and lanes) on business process diagram. No concept of requirements.
Process structure view	✓ Class diagram	✗ <i>No representation</i> .
Process content view	✓ <i>Class diagram</i> : processes shown as classes with artefacts shown as attributes and activities as operations	✗ <i>No representation</i> . Only way to understand content of each process is to look at business process diagram for each process.
Stakeholder view	✓ <i>Class diagram</i> : each stakeholder shown as a class	✗ <i>No representation</i> . Stakeholder information only found as swimlanes (pools and lanes) on business process diagram.
Process behaviour view	✓ <i>Activity diagram</i> : stakeholders shown as swimlanes, activities as activity invocations, artefacts as objects	✓ <i>Business process diagram</i> : stakeholders shown as swimlanes (pools and lanes), activities as activities, artefacts as data objects
Information view	✓ <i>Class diagram</i> : artefacts shown as classes	✗ <i>No representation</i> . Artefacts only found as data objects on business process diagram.
Process instance view	✓ <i>Sequence diagram</i> : each process shown as a lifeline	✓ <i>Business process diagram</i> : Sub-processes can be connected together to show executions of processes in sequence.

Obrázek 14 Simon Perry - srovnání pohledů na byznys proces v BPMN a UML[5]

Simon Perry rozdělil pohled na strukturu a chování uvnitř procesu do 7 náhledů:

- 1) **Zachycení podmínek procesu** – zachycuje podmínky procesu a účastníky, jež zahrnuje
- 2) **Informační náhled** – zachycuje vstupy a výstupy, které proces produkuje nebo spotřebovává, a také vztahy mezi nimi
- 3) **Přehled účastníků procesu** – zachycuje účastníky procesu
- 4) **Pohled na složení procesu** – umožňuje náhled na strukturu a použité názvosloví procesu, zároveň definuje základní bázi pro všechny základní procesy pro zachycení jejich odlišností, což je důležité zejména pro vnitřní audit a hodnocení
- 5) **Pohled na obsah procesu** – definuje obsah procesu pomocí podmínek vstupů a aktivit, jenž tvoří proces
- 6) **Náhled chování procesu** – definuje chování procesu, pořadí aktivit, kdy jsou spotřebovávány vstupy a kdy vytvářeny výstupy a kdy jednotlivé role do procesu vstupují
- 7) **Náhled instance procesu** – určuje pořadí jednotlivých procesů a jejich návaznosti, nabízí scénář

Pokud se na těchto 7 náhledů podíváme z pohledu informačního, které nám jednotlivé notace nabízejí, vychází skóre 7 : 2 ve prospěch UML. Její reprezentace se neomezuje na jeden jediný typ diagramu, v souhrnu nabízí přehlednější a kompletnější náhled na celý proces.

4.2 Jednotlivé entity byznys procesu

V této podkapitole postupně vyjmenuji jednotlivé entity, z nichž se celý model skládá a uvedu u nich jejich reprezentaci v notaci UML.

4.2.1 Vlastní projekt

Začínáme od toho jednoduššího. Top-level struktura business procesů je samotný projekt. V tomto projektu se budou nacházet všechny procesy popsané pomocí byznys procesů. Odpovídající UML element nebude nic jiného, než samotný projekt v té stejné podobě.

4.2.2 Proces

Samotný proces je v UML reprezentován diagramem použití a jeho realizací uvedenou v kolaboraci. Kolaborace je v diagramu užití navázána na „Use Case“ a vyjadřuje spolupráci jednotlivých členů v konkrétním procesu. Obojí se projeví v diagramu užití UML (Use Case diagram). Tento diagram v tomto případě reprezentuje záměry a cíle podnikání, cíle konkrétního procesu. Byznys proces zde popisuje, co přesně je potřeba vykonat, aby se kýžených cílů dosáhlo. Proces má za cíl koordinaci všech participujících osob, zdrojů a jejich povinností.

Proces je tedy reprezentován jako diagram použití a jeho realizace zapsaná pomocí diagramu aktivit. Diagram aktivit popisuje, jak musí jednotlivé zúčastněné osoby reagovat při plnění svých povinností k dosažení cíle.

4.2.3 Role

Všechny úkoly v procesu musí být přiřazeny konkrétním rolím, které jsou zodpovědné za vykonávání těchto úkolů. Byznys proces popisuje, kdo dělá co a v jakém pořadí. Role tedy odpovídá funkci, jakou má daná osoba v daném životním cyklu procesu (nemusí se shodovat s funkcí, kterou člověk zastává ve společnosti).

Každá role, která vykonává nějaké úkoly nebo do procesu nějak zasahuje, je diagramu užití UML reprezentována takzvaným „BusinessActor“, tedy jakousi zúčastněnou osobou obchodního procesu. Zároveň každá takto zúčastněná osoba má přiřazeno jakési rozhraní, které definuje jeho konkrétní úlohu v tomto procesu.

Ve zkratce je tedy každá role vystupující v procesu zapsaného v UML reprezentována osobou a třídou jejich povinností v kolaboraci v diagramu užití. Zároveň jsou jednotlivé role reprezentovány v diagramu aktivit jako jednotlivé swimlane.

4.2.4 Úkol

Úkol reprezentuje určitou část práce, kterou dále nemodelujeme. Procesy jsou tedy sekvencí propojených úkolů vykonávaných jednotlivými rolemi a můžou produkovat nebo spotřebovávat zdroje. Každý úkol přiřazený určité roli reprezentuje její povinnost vykonat přiřazený úkol.

Tyto povinnosti jednotlivých rolí jsou shrnuty v jejich rozhraní (třída) v kolaboraci a zobrazené jako jednotlivé operace této třídy. Zároveň v popisu aktivit – Aktivit diagramu jsou jednotlivé úkoly reprezentovány jako akce. Každá tato akce volá metodu z rozhraní dané role.

4.2.5 Propojení

Spojení jednotlivých objektů, anglicky sequence flow, reprezentuje načasování akcí v procesu konkrétním rolím. V UML jsou reprezentovány pomocí ControlFlow nebo DataFlow.

4.2.6 Začátek a konec procesu

V BPMN pro začátek procesu slouží start event a pro konec end event. Tyto uzly jsou určeny k signalizaci počátku nebo konci procesu. Počátek může být iniciován různými způsoby – zprávou (message), časem (timer), pravidlem (rule). V aktivit diagramu jsou tyto uzly reprezentovány pomocí InitialNode a FinalNode.

4.2.7 Rozhodnutí / spojení

Rozhodnutí (decision) a sloučení (merge) mají v byznys procesu stejnou reprezentaci jako v UML aktivit diagramu – DecisionNode / Merge Node.

4.2.8 Rozvětvení / sdružení

Model procesu může obsahovat „fork“ uzel pro rozvětvení procesu a zároveň „join“ pro synchronizaci paralelních procesů. Tyto uzly mají také shodnou reprezentaci v aktivit diagramu – ForkNode / JoinNode.

4.3 Výsledek transformace

Samotným výsledkem transformace procesu vznikne v UML projekt, který obsahuje 2 diagramy – diagram užití a diagram aktivit – kdy právě diagram aktivit popisuje realizaci diagramu užití, ve kterém se specifikují jednotlivé role v procesu.

4.4 Modelovací programy

Než model začneme převádět, je zapotřebí mít nějaký namodelovaný, případně si ho namodelovat. K tomu slouží programy – modelovací nástroje, které umožňují podle dané notace zakreslit různé diagramy. Nejčastěji se na internetu setkáte s aplikacemi podporujícími většinu základních diagramů spadajících do UML. Já jsem se musel zaměřit na ty dostupné modelovací programy, které podporují modelování v notaci BPMN.

Zdá se, že s postupným rozvojem této notace, která se v podstatě stává standardem (ikdyž jím ještě není), se počet těchto programů na internetu rozrůstá. Praxe je taková, že větší společnosti, které mají vlastní grafické aplikace na modelování různých diagramů, přidávají různými způsoby (rozšiřující moduly, nové verze, ...) právě BPMN do podporovaných diagramů.

Jako příklad uvedu 4 společnosti, jejichž programy BPMN podporují a různými způsoby se dají použít i zadarmo. Většinou pouze jako trial verze na určitou omezenou dobu.

Jako první bych zmínil TIBICO Business Studio společnosti Tibo, zabývající se řešeními SOA (Service-Oriented Architecture). Tato společnost nabízí svoje Studio jako free software, takže při registraci je možné ho plně využívat.

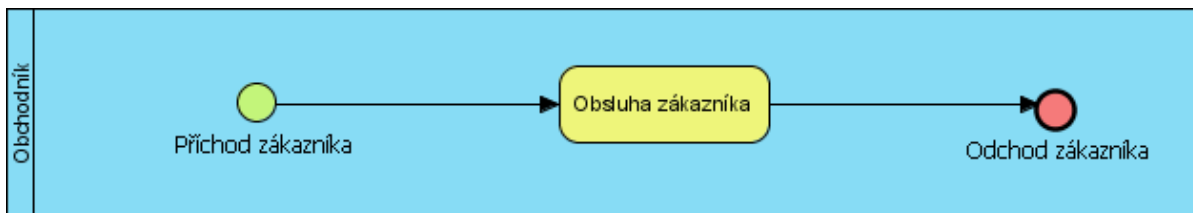
Dalším řešením, které je dostupné na internetu, je produkt společnosti No Magic, Inc., který nese název Magic Draw. Zde se jedná o software, který podporuje různé diagramy z UML standardu, mezi nimiž od verze standard edition (ta již není nabízena zdarma) je i BPMN.

Další společnost, jenž počítá s notací BPMN je ILOG. Společnost ILOG nabízí řadu produktů pro softwarové vývojáře a pro různé programovací jazyky. Řadu těchto produktů doplňují různé grafické podpůrné programy, z nichž jeden se týká právě naší problematiky byznys procesů – konkrétně ILOG JViews BPMN Modeler.

Společnost, kterou jsem si vybral k podpoře v mé aplikaci pro transformaci, je Visual Paradigma International. Tato společnost nabízí na trhu různé nástroje pro podporu vývoje a tvorby softwaru od prvopočátků návrhu, přes všechny vývojové stupně až po samotnou tvorbu software včetně podpory týmové práce. Nás bude zajímat balík programů pro podporu modelování diagramů – program Visual Paradigma Business ARCHITECT pro modelování BPMN diagramů. Pro zobrazení výsledných diagramů v UML budeme používat nástroj, který společnost nabízí pod názvem Visual Paradigma for UML. Tento program byl zvolen také z toho důvodu, že Fakulta informatiky, na které byla tato práce vypracována, vlastní školní licence na tyto programy a mohl jsem při práci tyto licence využít. Nicméně tyto produkty jsou k dispozici ke stažení na stránkách www.visual-paradigama.com a při registraci je možné stáhnout 30ti denní zkušební licenční klíč.

4.5 Úroveň XML/XMI popisů

Dalším aspektem, který jsem sledoval při výběru modelovacího programu, byla možnost exportu a následného importu zakreslených diagramů do formátu XML. Protože program Business ARCHITECT verze 2.2, který jsem si vybral pro modelování byznys procesů k transformaci pomocí mé aplikace, nepodporuje export přímo do UML metadata Interchange (formát XMI), spokojil jsem se se základními XML popisky.



Obrázek 15 Ukázkový proces obchodníka

Jako ukázkou zde vidíte triviální příklad procesu obchodníka. Obsahuje pouze jeden úkol – Obsluha zákazníka a počáteční a koncový uzel. Tento proces by se v normálním případě dále modeloval. My se však zaměříme na jeho reprezentaci v XML vygenerovanou programem Business ARCHITECT.

```
- <Diagrams>
- <Diagram connectorStyle="Round Rectilinear" diagramType="BusinessProcessDiagram" displayDiagramType="Business Process Diagram"
documentation="" id="7IcTChiGAqACAawCe" imageFileName="Diagram_Obchodnika.png" imageHeight="116" imageScale="1.0" imageWidth="634"
name="Diagram Obchodníka" rtfDocumentation="" trimmedHeight="128" trimmedWidth="28">
- <Shapes>
- <Shape height="110" id="fe_TChiGAqACAawCj" model="fe_TChiGAqACAawCk" name="Obchodník" shapeType="BPPool" width="628" x="2" y="2"
zorder="7">
  <ElementFont color="Cr:0,0,0,255" name="Dialog" size="11" style="0"/>
  + <Line cap="0" color="Cr:0,0,0,255" transparency="0" weight="1.0"/></Line>
  <Caption height="110" side="InsideNorth" visible="true" width="15" x="-28" y="-128"/>
- <ChildShapes>
  + <Shape height="20" id="dIybChiGAqACAawE2" model="dIybChiGAqACAawE3" name="Přijde zákazník" shapeType="BPStartEvent" width="20"
x="123" y="41" zorder="4"/></Shape>
  + <Shape height="20" id="JSVbChiGAqACAawNK" model="JSVbChiGAqACAawNL" name="Zákazník odchází" shapeType="BPEndEvent"
width="20" x="543" y="46" zorder="2"/></Shape>
  + <Shape height="40" id="WNQq.ZiGAqACAawdb" model="WNQq.ZiGAqACAawdc" name="Obsluha zákazníka" shapeType="BPTask"
width="110" x="292" y="32" zorder="0"/></Shape>
</ChildShapes>
  <FillColor color="Cr:135,220,245,255" style="1" transparency="0" type="1"/>
</Shape>
</Shapes>
- <Connectors>
+ <Connector connectorStyle="Follow Diagram" from="dIybChiGAqACAawE2" height="100" id="HkMq.ZiGAqACAawgz"
model="HkMq.ZiGAqACAawgy" name="" shapeType="BPSequenceFlow" to="WNQq.ZiGAqACAawdb" width="248" x="93" y="2"
zorder="-1"/></Connector>
+ <Connector connectorStyle="Follow Diagram" from="WNQq.ZiGAqACAawdb" height="100" id="Kkcq.ZiGAqACAawhk"
model="Kkcq.ZiGAqACAawhj" name="" shapeType="BPSequenceFlow" to="JSVbChiGAqACAawNK" width="241" x="352" y="2"
zorder="-1"/></Connector>
</Connectors>
</Diagram>
</Diagrams>
```

Obrázek 16 Část exportovaného XML z procesu obchodníka

Zobrazil jsem zde jako ilustrační příklad vygenerovaného XML popisujícího předcházející jednoduchý byznys proces. Z vygenerovaného XML dokumentu jsem odstranil rozšiřující informace popisující grafický vzhled diagramu specifický právě pro Business ARCHITECT. Dále zde vidíte pouze výřez popisující diagram BPMN. Vygenerované popisky XML obsahují navíc informace o modelu, ale tyto informace nejsou pro moji aplikaci důležité, vycházím z grafické podoby modelovaného diagramu. Navíc si můžete všimnout u některých elementů znaménko + značící nerozbalený element. Tyto elementy obsahují pouze grafické informace o daných prvcích, a proto jsem je nechal nerozbalené, aby zde zbytečně nezabíraly místo.

V XML exportu jsou jednotlivé diagramy uzavřeny do tagu `<diagrams>`. V něm již vidíme náš diagram procesu obchodníka. Jeho jediný kořenový shape je právě pool značící proces vykonávaný obchodníkem. Všechny ostatní prvky diagramu již patří do tohoto poolu, takže jsou v něm vnořeny jako `childShapes`. Jedinou výjimkou jsou značky `<connectors>` popisující propojení prvků v diagramu. U nich nezáleží, jestli jsou nějak vnořeny, jelikož jsou jednoznačně určeny pomocí ID atributů prvků, které spojují.

5 Popis vlastní implementace

Mým úkolem v této práci bylo vytvořit prostředek, který bude schopen provést výše zmíněnou transformaci byznys procesu na úrovni XML. Jinými slovy bylo zapotřebí navrhnout a vytvořit aplikaci, která z modelu zapsaného v notaci BPMN a její textové reprezentaci v XML vytvoří tentýž proces zapsaný pomocí XMI popisků reprezentujících tento proces v UML.

Dalším dílčím požadavkem zadání bylo klást důraz na modifikovatelnost a znovupoužití mé práce, které by v dnešní době mělo být samozřejmostí.

5.1 Volba prostředí

První a základní volbou pro vytvoření této aplikace bylo rozhodnout se o prostředí, ve kterém se bude aplikace používat a zároveň zvolit její implementační jazyk. Jelikož jazyk k tomuto tématu nebyl přímo specifikován, zvolil jsem pro mě oblíbený a využívaný skriptovací jazyk PHP (Hypertext Preprocessor), konkrétně jeho aktuální verzi PHP 5. Výběr PHP byl opodstatněn a schválena mým vedoucím z důvodu možného nasazení aplikace do prostředí webových aplikací přístupných všem.

Zároveň jsem při uvažování o programovacím jazyku počítal s využitím nějakého prostředku pro lepší práci s XML soubory. V tomto ohledu se v PHP 5 objevila spousta novinek a jednou z nich je DOM Extension. Jedná se o plně objektovou nástavbu pro práci s XML soubory.

5.2 Volba modelovacího programu

Jak se již objevilo v předchozí kapitole, jako modelovací programy jsem zvolil software od společnosti Visual Paradigma International. Pro modelování vstupního diagramu v BPMN notaci slouží program Business ARCHITECT a pro následné načtení transformovaného modelu v UML program Visual Paradigma for UML. Na oba dva programy má Fakulta informačních technologií licenci, takže nebyl problém s jejich plným využitím.

5.3 Modularita programu

Hlavním požadavkem bylo vytvořit aplikaci, která bude lehce modifikovatelná. Nároky na modifikovatelnost jsou kladeny zejména ve spojení se změnou nástroje ve kterém by byl transformovaný model zakreslen a exportován a zároveň změnou nástroje, ve kterém bychom chtěli transformovaný model načíst.

Tento požadavek byl zdrojem největšího počtu problému, se kterými jsem se během práce setkal. Zejména v UML modelech a jejich reprezentaci v XMI používá program Visual Paradigma for UML spoustu rozšíření, které jsou nutné pro správné zobrazení diagramu.

Aby se dalo snadno změnit podobu přetransformovaného XMI dokumentu, je skript napsán tak, že při sestavování UML diagramů „nevytváří“ popisky XML jednotlivých elementů, ale tyto elementy načte z již vytvořených a pouze poskládá nový dokument z těchto elementů. Jediné, co při tomto procesu upravuje, jsou atributy těchto elementů, popisující například názvy prvků, jejich ID nebo odkazy a propojení (tedy ID prvků na které se váže). Všechny tyto elementy jsou obsaženy ve složce `/src` a jejich popis lze najít také v programové dokumentaci. Musím se v tomto bodě zmínit o tom, že díky pár záležitostem a rozšíření, které Visual Paradigma používá při ukládání svých diagramů do XMI, není přechod na jiný program triviální v tom smyslu, že bychom pouze upravili nebo změnili zdrojové elementy XMI ve složce `/src`. Ale toto řešení spolu s lehkou úpravou zdrojových souborů by tento přechod mělo umožnit.

5.4 Vlastní řešení

Vzhledem k tomu, že aplikace je naprogramována v PHP, nejpohodlnější přístup k ní je přímo z webového prohlížeče. Připomínám, že je zde uvažováno o nasazení jako webová služba, kdy stačí zvýšit bezpečnost při nahrávání souborů na server, přidat nějaký grafický styl (například pomocí kaskádových stylů) a je ihned možno sdílet tuto aplikaci po síti.

Pro vlastní spuštění aplikace slouží úvodní skript `index.php`. Ten se postará o vyvolání formuláře pro nahrání souboru s modelem k transformaci. Skript využívá nejnovější DOM model pro práci s XML soubory, pomocí něhož projde skript a přečte vstupní model. Pokud projde mým vlastním parserem, který zjišťuje, jestli se opravdu jedná o BPMN model, načte všechny prvky modelu do pomocného pole ve struktuře odpovídající struktuře, v níž jednotlivé prvky jsou obsaženy v BPMN diagramu. Parser, o kterém jsem se zmínil, zjišťuje přítomnost základních tagů, které jsou v BPMN modelu nezbytné. Pokud něco chybí, vypíše chybovou hlášku s popisem, v čem je chyba. Zároveň tento jednoduchý parser zjišťuje, kolik diagramů nahraný projekt obsahuje. Skript dokáže převádět libovolný počet diagramů BPMN uzavřených v rámci jednoho projektu, což je výhoda v případě, že máme nějaký konkrétní soubor procesů (například procesy v administrační firmě), určitě se nebude jednat o jeden diagram.

```

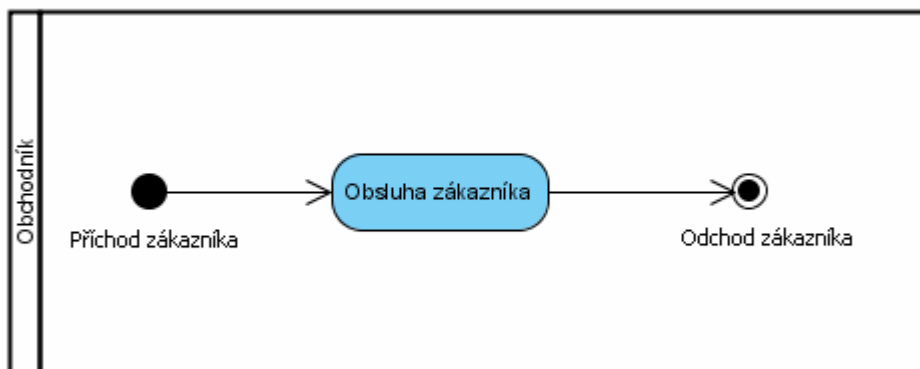
Array (
  [0] => Array (
    [name] => Business Process Diagram1
    [id] => 2Huo8JiGAqACAgAQ
    [0] => Array (
      [shapeName] => Pool
      [shapeType] => BPPool
      [shapeId] => tnyu8JiGAqACAgbg
      [0] => Array (
        [childShapeName] => Obchodník
        [childShapeType] => BPLane
        [childShapeId] => o5qu8JiGAqACAgc9
        [0] => Array (
          [childShapeName] => Příchod zákazník
          [childShapeType] => BPStartEvent
          [childShapeId] => 9Nxo8JiGAqACAgOB )
        [1] => Array (
          [childShapeName] => Odchod zákazníka
          [childShapeType] => BPEndEvent
          [childShapeId] => Gdpo8JiGAqACAg00 )
        [2] => Array (
          [childShapeName] => Obsloužit zákazníka
          [childShapeType] => BPTask
          [childShapeId] => Fa5o8JiGAqACAgPn ) ) )
    [1] => Array (
      [shapeType] => BPConnector
      [connectorName] => Sekvence1
      [connectorFrom] => 9Nxo8JiGAqACAgOB
      [connectorTo] => Fa5o8JiGAqACAgPn
      [connectorId] => PrNo8JiGAqACAgSz )
    [2] => Array (
      [shapeType] => BPConnector
      [connectorName] => Sekvence2
      [connectorFrom] => Fa5o8JiGAqACAgPn
      [connectorTo] => Gdpo8JiGAqACAg00
      [connectorId] => s.to8JiGAqACAgS3 ) ) )

```

Obrázek 17 Příklad pomocného pole obsahující diagram

Na předchozí ukázce (obrázek 17) vidíte digram zapsaný uvnitř programu v pomocném poli. Původní pole obsahuje nejdříve pole diagramů, potom pole bazénů (swimmingpool), následuje pole drah a teprve potom ve struktuře přichází na řadu jednotlivé prvky, jako počáteční a koncové události, úkoly a rozhodovací prvky. S tím pravidlem, že pokud některá z těchto tří vrstev v diagramu není obsažena, následuje hned ta další. Na konci výpisu jsou vidět dva elementy nezařazené do struktury, jedná se právě o propojení prvků, které nepřísluší dovnitř žádné dráze. Pouze položky ID popisují ke kterým značkám daný konektor patří.

Když jsem se zmínil o ID prvků, zmíním se také o problému, s kterým jsem musel pracovat. Program Visual Paradigma generuje všem prvkům exportovaným do XML nebo XMI popisků jednoznačná 10ti místná ID. Nejdříve jsem chtěl pracovat s těmito vytetovanými ID i v novém UML diagramu, tedy použít ty ID uložené ve struktuře v poli, jak je možné vidět na předchozí ukázce. Tato práce by však byla příliš nekomfortní, některým novým prvkům (zejména v Use Case diagramu),



Obrázek 19 UML Activity diagram

Jakmile jsou nové digramy sestaveny, jsou uloženy do dočasného souboru na serveru, na kterém skript spouštíme a prohlížeč nám nabídne odkaz k jeho stažení. Jméno souboru je možné změnit v konfiguračním souboru `constants.php`, který obsahuje několik dalších předvoleb popsaných v dokumentaci.

Závěr

Modelování byznys procesů je stále populárnější a žádanější. BPMN se stalo standardem jejich reprezentace. Nabízí na jedné straně velmi intuitivní řešení, pomocí kterého se však dá obsáhnout vše podstatné. Hierarchie sub-procesů umožňuje namodelovat chování celého fungujícího systému, zároveň ale odborníci upozorňují, že různorodost úrovně popisu se může při výstavbě procesů stát problémem jejich nekonzistentnosti.

Studovat a pracovat s byznys procesy mě velmi zaujalo a jistě budu v této práci pokračovat. Přeji všem podnikatelům, kteří stojí o úspěch ve svých aktivitách, aby firemní procesy nepodceňovali, věnovali se jim a uměli je řídit a spravovat.

Pokusím se zhodnotit výsledky své práce při studiu problematiky modelování procesů, jejich export na úroveň XML a transformaci do jiné notace. Všechny body zadání jsem splnil, jak se může čtenář přesvědčit ze studia teorie byznys procesů, notace BPMN, její reprezentace v XML popsané v této práci. Zároveň se podařilo navrhnout a naprogramovat aplikaci s jednoduchým ovládáním, která provádí transformaci procesu z jedné notace do druhé. Když se zaměřím na problémy, které mě při této činnosti potkaly, musím jednoznačně zmínit rozlišnosti modelovacích programů v tom, jakým způsobem reprezentují grafické modely do XML a XMI. Rozšiřování standardů o různé grafické a podpůrné informace, informace o modelovacích nástrojích, ale hlavně o vzájemné poloze a podobě jednotlivých prvků jsou jiné nejen u každého modelovacího programu, ale dokonce v rámci jednoho balíku, jakým je program Visual Paradigma. Zápisky v XMI podobě se liší u různých digramů (viz kapitola 5.4). Tento problém způsobil největší komplikace a tím i omezil použití pouze pro modely z tohoto programu. Z těchto omezení vyplývají možnosti dalšího vývoje tohoto programu, jimiž mohou být nalezení styčných bodů při popisování diagramů pomocí XML, případně určení standardu těchto popisů, tak aby aplikace mohla být používána s různými modelovacími prostředky.

O možnostech budoucího vývoje samotných byznys procesů nemůže být sporu. Já sám díky poznatkům, ke kterým jsem dospěl během jejich studia, se byznys procesy okamžitě začnu zabývat uvnitř společnosti, ve které pracuji a která je mým druhým domovem. Vypracování a popsání byznys procesů a jejich namodelování je nezbytnou součástí projektu další optimalizace služeb ve vztahu k zákazníkovi. A to je cíl nás všech.

Literatura

- [1] Stehen A. W., *Introduction to BPMN*, BPM Architect, IBM 2006
- [2] Bradley N., *XML kompletní průvodce*, GRADA Publishing, Praha 2000
- [3] Kosek J., *XML pro každého*, GRADA Publishing, Praha 2000
- [4] Rychlý M., Weiss P., *Architektura orientovaná na služby, návrh orientovaný na služby, webové služby*, FIT VUT v Brně, Brno 2007
- [5] Řepa V., *Podnikové procesy, procesní řízení a modelování*, GRADA Publishing, Praha 2006
- [6] Amsden, J.: *Business services modeling*. IBM, 2005. Dostupný na http://www.ibm.com/developerworks/rational/library/05/1227_amsden/

Seznam příloh

Příloha 1. Obsah přiloženého CD:

- tento text bakalářské práce ve formátu PDF a RTF
- dokumentace programu
- soubor readme.txt popisující seznam těchto součástí
- zdrojové skripty programu a další zdrojové soubory