



ZADÁNÍ BAKALÁ SKÉ PRÁCE

Název:	Možnosti p evodu DEMO procesních model do notace BPMN
Student:	Anežka Posoldová
Vedoucí:	Ing. Pavel Náplava
Studijní program:	Informatika
Studijní obor:	Informa ní systémy a management
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce letního semestru 2017/18

Pokyny pro vypracování

1. Definujte základní pojmy: proces, procesní ízení, modelování proces .
2. Seznamte se s metodikou DEMO a jejími možnostmi pro popis a modelování proces .
3. Prove te rešerši existujících publikací, týkajících se možností p evodu DEMO model do notace BPMN.
4. Porovnejte výsledky a záv ry z nastudovaných prací s výstupy diplomových prací Ond eje Mráze a Št pána Hellera.
5. Po dohod s vedoucím práce vytvo te sadu vzorových procesních diagram , které použijete pro praktické ov ení jak výstup uvedených diplomových prací, tak i záv r , u in ných v prostudovaných publikacích.
6. Vyhodno te kvalitu a použitelnost metod z diplomových prací, navrhn te a ov te jejich možná zlepšení, rozší ení nebo redukci.

Seznam odborné literatury

- [1] Mráz O., Možnosti využití princip metodiky DEMO pro zvýšení kvality BPMN model , Diplomová práce, VUT FIT, Praha, 2016
- [2] Heller Š, Možnosti využití metodiky DEMO pro tvorbu BPMN model , Diplomová práce, VUT FIT, Praha, 2016

Ing. Michal Valenta, Ph.D.
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.
d kan

V Praze dne 18. listopadu 2016

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
KATEDRA SOFTWAREVÉHO INŽENÝRSTVÍ



Bakalářská práce

Možnosti převodu DEMO procesních modelů do notace BPMN

Anežka Posoldová

Vedoucí práce: Ing. Pavel Náplava

15. května 2017

Poděkování

Chtěla bych poděkovat vedoucímu práce, Ing. Pavlovi Náplavovi za rady a připomínky při tvorbě práce. Dále děkuji manželovi a rodičům za jejich podporu během studia.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 15. května 2017

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2017 Anežka Posoldová. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.

Odkaz na tuto práci

Posoldová, Anežka. *Možnosti převodu DEMO procesních modelů do notace BPMN*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2017.

Abstrakt

Tato bakalářská práce se zabývá porovnáním existujících metod převodu procesních diagramů vytvořených podle metodiky DEMO (Design & Engineering Methodology for Organizations) do notace BPMN (Business Process Model and Notation). V práci se nachází popis obou těchto modelovacích technik.

Dále jsou zde rozebrány existující metody převodu. Jde o metody, které vznikly v diplomových pracích Možnosti využití metodiky DEMO pro tvorbu BPMN modelů, jejímž autorem je Štěpán Heller a Možnosti využití principů metodiky DEMO pro zvýšení kvality BPMN modelů od Ondřeje Mráze. Součástí práce je ověření funkčnosti a použitelnosti obou metod. Způsob použití obou metod je předveden na dvou příkladech, a to jak ve variantě vyjádření standardního transakčního vzoru, tak v případě metody O. Mráze i ve variantě kompletního transakčního vzoru.

Vzniklé diagramy jsou porovnány na základě velikosti výsledných modelů a podle tohoto srovnání jsou vyhodnoceny silné a slabé stránky jednotlivých metod převodu. V závěru lze nalézt zhodnocení použitelnosti obou metod pro praktické použití.

Klíčová slova DEMO, BPMN, Procesní řízení, Procesní model

Abstract

This thesis deals with comparison of existing transformation ways of business process diagrams created according to the DEMO (Design & Engineering Methodology for Organizations) into BPMN (Business Process Model and Notation). Both of these techniques are described here.

Existing transformation methods are discussed here. They were developed in master theses *Možnosti využití metodiky DEMO pro tvorbu BPMN modelů* by Štěpán Heller and *Možnosti využití principů metodiky DEMO pro zvýšení kvality BPMN modelů* by Ondřej Mráz. Part of the thesis is to verify the functionality and usability of them. Usage of both methods is demonstrated on two use cases. Both in the variant of the standard transaction pattern and in the case of O. Mráz method in the complete transaction pattern variant.

The resulting diagrams are compared and the strengths and weaknesses of them are evaluated. In conclusion we can find an assessment of the suitability of both methods for practical use.

Keywords DEMO, BPMN, Business process management, Business process model

Obsah

Úvod	1
1 Definice základních pojmů	3
1.1 Podnikový proces	3
1.2 Procesní řízení	4
1.3 Modelování procesů	4
2 Notace BPMN	7
2.1 Úvod do BPMN	7
2.2 Základní elementy	7
2.3 Diagramy	12
3 Metodika DEMO	15
3.1 Úvod	15
3.2 Motivace a podniková ontologie	15
3.3 Teorie PSI	16
3.4 Modely v DEMO	23
4 Možnosti převodu DEMO modelů do notace BPMN	25
4.1 Úvod	25
4.2 Existující možnosti převodů	26
4.3 Metoda Štěpána Hellera	26
4.4 Metoda Ondřeje Mráze	29
5 Příklady převodů	37
5.1 Úvod	37
5.2 Příklad 1 – Ocenění nemovitosti podle Š. Hellera	37
5.3 Příklad 1 – Ocenění nemovitosti podle O.Mráze	45
5.4 Příklad 2 – Proces nalezení a opravy chyby v aplikaci podle Š. Hellera	52

5.5	Příklad 2 – Proces nalezení a opravy chyby v aplikaci podle O. Mráze	60
6	Vyhodnocení metod převodů	67
6.1	Porovnání metod	67
6.2	Porovnání převedených diagramů s původním BPMN diagramem	69
6.3	Návrh pro vylepšení metody	70
6.4	Použitelnost metod v praxi	71
	Závěr	75
	Literatura	77
	A Seznam použitých zkratk	79
	B Obsah příloženého CD	81

Seznam obrázků

1.1	Ukázka vývojového diagramu [7]	5
1.2	Ukázka UML diagramu aktivit [9]	6
2.1	Symboly aktivit – Task a Subprocess.	8
2.2	Symboly událostí	9
2.3	Symboly bran	10
2.4	Spojovací objekty	11
2.5	Bazén s plaveckými dráhami	12
2.6	Datový objekt a uložisko	12
2.7	Artefakty	13
2.8	Diagram spolupráce [12]	14
3.1	Znázornění operačního axiomu [15]	17
3.2	Základní transakční vzor [15]	18
3.3	Standardní transakční vzor [15]	19
3.4	Kompletní transakční vzor [16]	20
3.5	Znázornění kompozičního axiomu	21
3.6	Distinkční axiom [15]	22
3.7	Organizační teorém [15]	22
3.8	Struktura DEMO modelů [15]	23
4.1	Standardní transakční vzor pomocí Úloh a Signálů podle [11]	28
4.2	Spouštění podřízené transakce [17]	30
4.3	Čekání na dokončení child transakce [17]	30
4.4	Příklad vyjádření Revoke [17]	31
4.5	Kompletní transakční vzor podle Mráze [17]	32
4.6	Podprocesy revoků podle Mráze [17]	34
4.7	Spojení symbolů pro C-acty a C-facty [17]	35
4.8	Symbol pro spouštění child transakcí [17]	35
4.9	Symboly pro revoky [17]	35

5.1	Pořadí a závislosti transakcí	40
5.2	Actor Transaction Diagram	40
5.3	Process Structure Diagram	41
5.4	Výsledný BPMN model ocenění nemovitosti – 1. část podle Hellera	42
5.5	Výsledný BPMN model ocenění nemovitosti – 2. část podle Hellera	43
5.6	Výsledný BPMN model ocenění nemovitosti – 3. část podle Hellera	44
5.7	Výsledný BPMN model ocenění nemovitosti – 1. část podle Mráze	46
5.8	Výsledný BPMN model ocenění nemovitosti – 2. část podle Mráze	47
5.9	Výsledný BPMN model ocenění nemovitosti – 3. část podle Mráze	48
5.10	Kompletní transakční vzor ocenění nemovitosti – 1. část	49
5.11	Kompletní transakční vzor ocenění nemovitosti – 2. část	50
5.12	Kompletní transakční vzor ocenění nemovitosti – 3. část	51
5.13	BPMN model procesu opravy bez použití DEMO	54
5.14	Pořadí a závislosti transakcí	55
5.15	Actor Transaction Diagram	55
5.16	Process Structure Diagram	56
5.17	BPMN model opravy chyby – 1. část podle Hellera	57
5.18	BPMN model opravy chyby – 2. část podle Hellera	58
5.19	BPMN model opravy chyby – 3. část podle Hellera	59
5.20	BPMN model opravy chyby – 1. část podle Mráze	61
5.21	BPMN model opravy chyby – 2. část podle Mráze	62
5.22	BPMN model opravy chyby – 3. část podle Mráze	63
5.23	Kompletní transakční vzor opravy chyby – 1. část	64
5.24	Kompletní transakční vzor opravy chyby – 2. část	65
5.25	Kompletní transakční vzor opravy chyby – 3. část	66
6.1	Element kombinující úlohu a signál	70
6.2	Zkrácený základní transakční vzor	70

Seznam tabulek

5.1	Transaction Result Table procesu ocenění nemovitosti	39
5.2	Transaction Result Table procesu opravy chyby	53
6.1	Základní porovnání velikosti modelů	68
6.2	Porovnání velikosti modelů bez speciálních symbolů	68
6.3	Porovnání velikosti modelů vytvořených podle DEMO s BPMN modelem	69

Úvod

Podniky při svém rozšiřování narážejí na nutnost zvyšování efektivity fungování vnitřních procesů. Jedním ze způsobů jak tomuto zlepšení fungování napomoci je zavádění metod procesního řízení. Procesní řízení pro zvýšení kvality procesů potřebuje tyto procesy nějakým způsobem zaznamenat. Znárodnění procesů by mělo být co nejpřesnější a zároveň co nejsrozumitelnější pro všechny zúčastněné osoby.

Jednou z možností, jak vytvářet modely procesů, je využití metodiky DEMO. Modely vytvořené podle ní jsou kompletní – popisují všechny události, které mohou při průchodu procesem nastat. Problémem je nízká srozumitelnost výsledných modelů. Další z možností je modelování pomocí notace BPMN. Výsledné modely jsou srozumitelné, ale nejsou kompletní.

Tato práce se zabývá existujícími možnostmi převodu procesních modelů vytvořených podle metodiky DEMO do notace BPMN. Tyto metody převodu vznikly s cílem zkombinovat silné stránky DEMO a BPMN, tedy vytvářet modely, které budou kompletní a srozumitelné zároveň.

Motivací ke vzniku této práce je neexistence práce, která by se těmito metodami zabývala z hlediska ověření jejich funkčnosti a použitelnosti pro praktické využití při modelování podnikových procesů.

V první části práce jsou definovány základní pojmy používané v oblasti procesního řízení. Následuje kapitola věnovaná notaci BPMN, jednotlivým elementům ze kterých se modely skládají a typům vytvářených diagramů. Ve třetí kapitole je popsána metodika DEMO, a to zejména teoretický základ, na kterém je postavena. Čtvrtá kapitola popisuje existující metody převodu z DEMO do BPMN a je následována kapitolou věnovanou příkladům převodů podle těchto metod. Na závěr jsou výstupy obou metod porovnány a zhodnoceny.

Cílem je seznámení se s metodami převodu modelů z DEMO do BPMN, které vznikly v diplomových pracích Š. Hellera a O. Mráze, a na příkladech ověřit jejich funkčnost. Dalším cílem je na základě vzorových diagramů metody porovnat, zhodnotit jejich použitelnost a navrhnout možná zlepšení.

Definice základních pojmů

V první kapitole práce jsou definovány základní pojmy, které jsou nezbytné pro porozumění zbytku práce.

1.1 Podnikový proces

S pojmem proces, případně podnikový proces se setkáváme často. Intuitivně je chápán jako činnost vedoucí k nějakému výsledku. Tomuto intuitivnímu pochopení významu odpovídá například definice

„Proces je soubor činností, který vyžaduje jeden nebo více druhů vstupů a tvoří výstup, který má pro zákazníka hodnotu.“ [1]

Nejde ale o jedinou definici tohoto pojmu, ani o tu nejpřesnější. Například v [2] je pojem proces definován podrobněji:

„Proces je organizovaná skupina vzájemně souvisejících činností a/nebo subprocesů, které procházejí jedním nebo více organizačními útvary či jednou (podnikový proces) nebo více spolupracujícími organizacemi (mezipodnikový proces), které spotřebovávají materiální, lidské, finanční a informační vstupy a jejichž výstupem je produkt, který má hodnotu pro externího nebo interního zákazníka.“

Podnikový proces popisuje [3]:

- Které aktivity jsou v jeho průběhu vykonány.
- Které organizační jednotky se na provádění procesu podílejí.
- Jaká jsou použita vstupní a výstupní data.
- Jaké události a rizika mohou během provádění procesu nastat.

1.2 Procesní řízení

Procesní řízení (Business Process Management) můžeme definovat například následovně:

„Procesní řízení (management) představuje systémy, postupy, metody a nástroje trvalého zajištění maximální výkonnosti a neustálého zlepšování podnikových procesů i mezipodnikových procesů, které vycházejí z jasně definované strategie organizace a jejichž cílem je naplnit stanovené strategické cíle.“ [2]

Cílem procesního řízení tedy není jen řízení procesů, ale neustálá práce na zlepšování efektivity procesů pro dosažení strategických cílů organizace. Jednotlivé cíle, kterých lze dosahovat pomocí procesního řízení, jsou vyjmenovány v [3]:

- Sladění aspektů organizace s požadavky zákazníků
- Zvýšení transparentnosti procesů
- Zvýšení výkonnosti
- Maximální využití potenciálu zaměstnanců
- Zvýšení kvality produktů
- Snížení cen

Procesní řízení se tedy silně orientuje na zvyšování kvality procesů. Abychom ale byli schopni proces zlepšit, potřebujeme ho nějakým způsobem popsat. Tím, jakým způsobem lze procesy popisovat, se zabývá následující část – modelování procesů.

1.3 Modelování procesů

Při práci s procesy narážíme na potřebu jejich zachycení, pokud možno vizuální formou, tak aby byly srozumitelné širokému okruhu uživatelů. Stejně tak pokud chceme proces upravovat, nebo optimalizovat, potřebujeme ho mít nejdříve nějakým způsobem vyjádřený. Touto oblastí se zabývá modelování procesů – *Business Process Modeling* (BPM). Procesní model můžeme definovat například takto:

Procesní model popisuje, typicky grafickou formou, aktivity, události a jejich propojení, které tvoří podnikový proces.[4]

Proces můžeme popsat buď textově, nebo graficky. Grafické znázornění procesů je používanější, protože v obrázcích se uživatelé lépe orientují. Z existujících metod grafického vyjádření procesů zde jsou stručně popsány dvě z nich – vývojové diagramy a notace UML. Další dvě metody, BPMN a DEMO jsou pro celou práci zásadní, a proto jsou jim věnovány samostatné kapitoly.

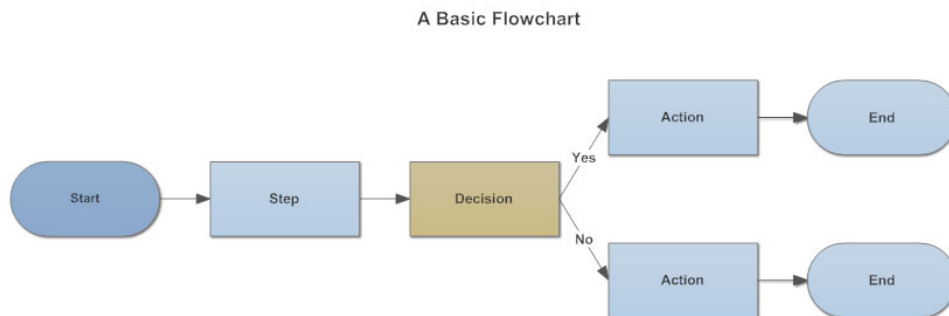
1.3.1 Vývojové diagramy

Vývojové diagramy patří mezi nejjednodušší způsoby zachycení procesů. Velmi často se používají v IT pro zápis algoritmů. Oblíbené jsou hlavně pro svou jednoduchost a s tím související srozumitelnost i pro uživatele, kteří s nimi nemají velkou zkušenost. Nevýhody vývojových diagramů jsou shrnuty v [5]: Nehodí se pro modelování komplexních procesů, výsledkem by byl rozsáhlý a nepřehledný model. Další nevýhodou je přílišné zjednodušování některých částí procesů a problematické úpravy už hotových modelů.

Základní elementy vývojových diagramů podle [6]:

- Počáteční a koncové symboly
- Šipky – Určení pořadí provádění činností.
- Kroky procesu – Znázornění jednotlivých činností.
- Rozhodnutí – Větvení procesu podle podmínky.
- Vstupy a výstupy

Příklad vývojového diagramu můžeme vidět na obrázku 1.1.



Obrázek 1.1: Ukázka vývojového diagramu [7]

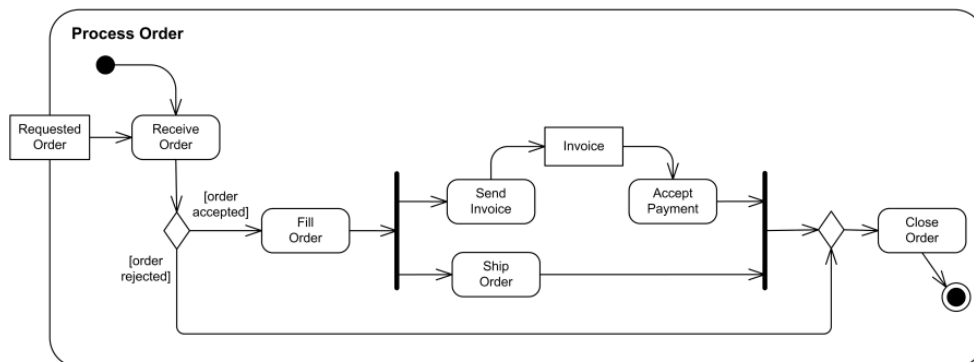
1.3.2 UML

Notace UML je dalším velmi rozšířeným nástrojem pro tvorbu procesních modelů. Stejně jako vývojové diagramy je používána především v oblasti IT. Popis v této kapitole vychází ze specifikace [8] vytvořené organizací OMG, která za UML stojí.

UML bylo ze svého základu postupně rozšiřováno, aby odpovídalo požadavkům svých uživatelů. Dnes, ve verzi 2.0 obsahuje třináct různých diagramů, rozdělených do oblastí strukturálních diagramů, diagramů chování a diagramů interakce. Nevýhodou UML pro použití k modelování podnikových procesů

1. DEFINICE ZÁKLADNÍCH POJMŮ

je neexistence standardu, který by tuto oblast pokrýval. Pro potřeby zaznamenání podnikových procesů je vhodný jeden z diagramů chování – diagram aktivit. Svou strukturou je podobný modelům v BPMN, proces vyjadřuje jako posloupnost aktivit. Příklad UML diagramu aktivit je na obrázku 1.2.



Obrázek 1.2: Ukázka UML diagramu aktivit [9]

Notace BPMN

2.1 Úvod do BPMN

BPMN (*Business Process Model and Notation*) je grafická notace pro popis podnikových procesů, kterou vyvíjí organizace OMG (*Object Management Group*). Motivací k jejímu vzniku byla snaha vytvořit takovou notaci, která bude srozumitelná pro všechny od manažerů, přes analytiky a vývojáře až po uživatele:

„The primary goal of BPMN is to provide a notation that is readily understandable by all business users, from the business analysts that create the initial drafts of the processes, to the technical developers responsible for implementing the technology that will perform those processes, and finally, to the business people who will manage and monitor those processes.“ [10]

BPMN je dnes pravděpodobně nejpoužívanější notací pro vytváření modelů podnikových procesů. Mezi jeho hlavní výhody patří to, že je standardizované a umožňuje automatizaci modelů. Dále existuje velké množství nástrojů, které tvorbu BPMN diagramů umožňují. Čtení diagramů je poměrně jednoduché, vzhled se podobá vývojovým diagramům, se kterými jsou business uživatelé obvykle seznámeni.

Nevýhodou BPMN je to, že je pouze notací popisující jednotlivé elementy a pravidla jejich použití. Nemá k sobě navázanou metodologii, která by specifikovala, jakým postupem modely vytvářet tak, aby byly kvalitní – validní, kompletní a jednoznačné. Jak říká [11], *„Špatné BPMN je normou častěji než výjimkou.“*

2.2 Základní elementy

BPMN obsahuje mnoho elementů, v této kapitole jsou představeny pouze ty, které jsou významné pro další části práce. Elementy jsou společné pro všechny tři druhy modelů, které lze v BPMN vytvářet (tyto jsou popsány v další části

2. NOTACE BPMN

práce), i když se v nich jejich použití mírně liší. Popis vychází ze specifikace [10]. Ve verzi BPMN 2.0 došlo k rozšíření symbolů o přesnější specifikaci jejich zaměření, tedy k přidání dalších podtypů jednotlivých elementů. V práci jsou použity obecné elementy, detailní se používají pro usnadnění převodu do spustitelného kódu.

2.2.1 Aktivity

Aktivita (*Activity*) symbolizuje jednotku práce. Může být dvojího druhu – Task nebo Subprocess. Task je atomická aktivita, která se nedá dále nijak dělit. Aktivita typu subprocess znázorňuje samostatný proces, který je kvůli úspoře místa a větší přehlednosti diagramu sbalen do jednoho elementu. Subproces má v sobě vlastní proces, který je ohraničen počáteční a koncovou událostí a obsahuje další aktivity. Symboly reprezentující aktivity jsou znázorněny na obrázku 2.1.



Obrázek 2.1: Symboly aktivit – Task a Subprocess.

2.2.2 Události

Události (*Event*) znázorňují jevy, které nastaly v průběhu procesu, a mohou ho různými způsoby ovlivňovat. Události se dále dělí na tři základní typy:

- Počáteční události – Označují začátek procesu. Procesy nemusí mít žádnou počáteční událost, ale mohou jich mít i více. Podmínkou je, že pokud má proces koncovou událost, musí mít alespoň jednu počáteční. Existuje sedm druhů počátečních událostí, pro tuto práci jsou důležité jen některé. Základním druhem je prázdná počáteční událost, která nemá definovanou žádnou událost, která by vedla k jejímu spuštění.
- Průběžné události – Jevy, které nastanou v průběhu procesu. Dále se dají dělit na události typu *throwing*, které vyvolávají nějakou akci a

události typu *catching*, které naopak na nějakou akci čekají. Akcí, na které události čekají/vyvolávají je existuje více druhů, my se budeme zabývat zprávou a signálem. Průběžná událost typu zpráva slouží buď k odeslání zprávy, nebo k jejímu příjmu. Lze takto komunikovat pouze mezi různými bazény (popis bazénu se nachází v sekci Plavecké dráhy). Průběžná událost typu signál je podobná předchozí události, s tím rozdílem, že signál nemá přesně určeného příjemce.

- **Koncové události** – Označují konec procesu. Stejně jako u počáteční události může diagram obsahovat i více koncových událostí, nebo žádnou. Těchto událostí existuje devět druhů. Nejdůležitější pro tuto práci je prázdná koncová událost, kdy proces skončí bez vyvolání nějaké další činnosti, a koncová událost typu zpráva, kdy končící proces pošle zprávu jinému procesu.

Symbyly základních typů událostí jsou na obrázku 2.2.



Obrázek 2.2: Symbyly událostí

2.2.3 Brány

Brána (*Gateway*) slouží k ovlivnění toku procesu. Je to jediný element umožňující rozvětvení nebo naopak sloučení toku procesu na základě různých podmínek. Symbyly pro jednotlivé druhy bran jsou na obrázku 2.3. Používají se tyto druhy bran:

- **Exkluzivní (*Exclusive*)** – Rozdělení toku procesu na více alternativních cest, podle podmínky se rozhodne, kterou cestou bude tok pokračovat. Může být vždy vybrána pouze jedna cesta.
- **Inkluzivní (*Inclusive*)** – Podobně jako exkluzivní brána dělí tok procesu na více alternativních cest, na rozdíl od ní může být vybráno více cest, kterými bude tok pokračovat.

2. NOTACE BPMN

- Paralelní (*Parallel*) – Paralelní brána slouží k vytváření paralelních toků a jejich synchronizaci. Vstupní tok se rozdělí bez jakékoli podmínky do více výstupních. Pokud je brána použita pro sloučení více toků dohromady, čeká se na příchod všech příchozích toků před pokračováním v procesu.
- Komplexní (*Complex*) – Komplexní brána umožňuje dělení a slučování toků podle složitých podmínek, které není možné vytvořit pomocí ostatních druhů bran.
- Založená na události (*Event Based*) – Výběr cesty se provede podle události, která v průběhu procesu nastala. Obvykle se používá v kombinaci s průběžnou událostí typu zpráva.

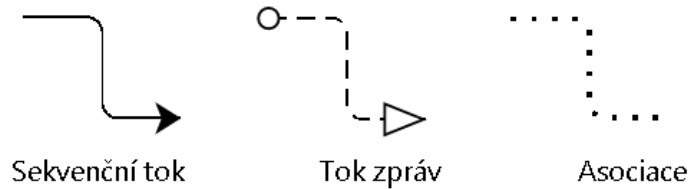


Obrázek 2.3: Symboly bran

2.2.4 Spojovací objekty

Spojovací objekty slouží k propojení ostatních objektů, jako jsou aktivity nebo brány. Existují tři druhy, znázorněné na obrázku 2.4:

- Sekvenční tok (*Sequence Flow*) – Určuje pořadí, ve kterém se procesem prochází. Zdroj i cíl musí být aktivita, událost nebo brána. Sekvenční tok z jednoho bazénu nesmí zasahovat do jiného bazénu, stejně tak nesmí překračovat hranice subprocessu.
- Tok zpráv (*Message Flow*) – Reprezentuje komunikaci mezi odesílatelem a příjemcem zprávy. Tok zpráv nesmí být použit pro komunikaci mezi entitami v rámci jednoho bazénu.
- Asociace (*Association*) – Slouží k propojení artefaktů (skupiny, anotace) s ostatními elementy v procesu.



Obrázek 2.4: Spojovací objekty

2.2.5 Plavecké dráhy

Plavecké dráhy slouží k oddělení účastníků procesů a rozdělení činností do celků. Mezi plavecké dráhy patří bazén a dráhy, jejich znázornění je na obrázku 2.5.

- **Bazén (*Pool*)** – Bazén určuje hranici procesu a reprezentuje jeho účastníka. Hranice bazénu mohou překročit pouze toky zpráv, nikoli sekvenční toky. Bazén může obsahovat model procesu (white box), nebo může být použit jako tzv. black box, ve kterém nejsou detaily zobrazeny. Bazén obsahuje jednu nebo více drah.
- **Dráhy (*Lanes*)** – Použití drah není v BPMN přesně specifikováno. Obvykle se používají pro oddělení rolí v bazénu a přehlednějšímu uspořádání aktivit. Mohou například reprezentovat jednotlivá oddělení v rámci organizace.

2.2.6 Data

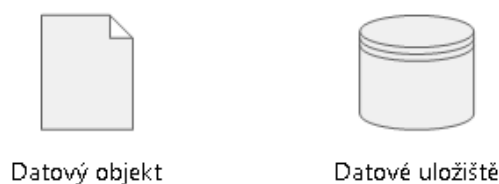
Pro grafickou reprezentaci dat potřebných k vykonání aktivity se používají dva elementy zobrazené na obrázku 2.6, datový objekt (*Data Object*) a datové úložiště (*Data Store*). Datový objekt symbolizuje data, která existují dočasně v rámci konkrétní instance procesu. Má dva speciální případy, datový vstup a datový výstup (*Data Input/Output*), které symbolizují vstupní, resp. výstupní data. Datové úložiště na rozdíl od objektu reprezentuje data, která jsou dostupná trvale, například v databázi. K ostatním elementům se datové objekty připojují pomocí datových asociací (*Data Association*).

2.2.7 Artefakty

Artefakty slouží k doplnění dalších informací do modelu procesu. BPMN poskytuje dva artefakty, skupinu (*Group*) a anotaci (*Annotation*). Někdy se mezi



Obrázek 2.5: Bazén s plaveckými dráhami



Obrázek 2.6: Datový objekt a uložení

artefakty zařazuje i asociace, která je v této práci zařazena mezi spojovací objekty a datový objekt, který zde má vlastní sekci. Skupiny umožňují vizuální rozdělení elementů do skupin. Nemají žádný vliv na tok procesu a nejsou omezeny hranicemi bazénů. Anotace se používají k přidání textových poznámek do modelu. Symboly pro oba artefakty jsou na obrázku 2.7.

2.3 Diagramy

V BPMN lze vytvářet tři druhy procesních diagramů. Nejvíce používaný je diagram spolupráce, který byl v BPMN obsažen už od první verze. Diagramy choreografie a konverzace byly přidány až ve verzi 2.0. a nejsou příliš rozší-



Obrázek 2.7: Artefakty

řené. Diagram spolupráce je někdy rozdělován na dva samostatné diagramy – spolupráce a procesu. Pro účely této práce bylo po diskusi s vedoucím práce rozhodnuto, že tento diagram na dva rozdělovat nebudeme.

2.3.1 Diagram spolupráce

Diagram spolupráce (*Collaboration diagram*), někdy nazývaný jako diagram procesu, je určen k zachycení toku procesu. Obvykle se skládá ze dvou a více bazénů, které znázorňují účastníky procesu a komunikace mezi nimi je zachycena pomocí toku zpráv. Znázorněné procesy můžeme dělit na soukromé a veřejné. Soukromé procesy se odehrávají v rámci jednoho bazénu a jedině, co může hranice bazénu překročit, je tok zpráv. Veřejné procesy znázorňují komunikaci mezi soukromým procesem a jiným účastníkem, či procesem. Jsou zde zobrazeny jen ty aktivity, které slouží ke komunikaci. Příklad diagramu spolupráce můžeme vidět na obrázku 2.8.

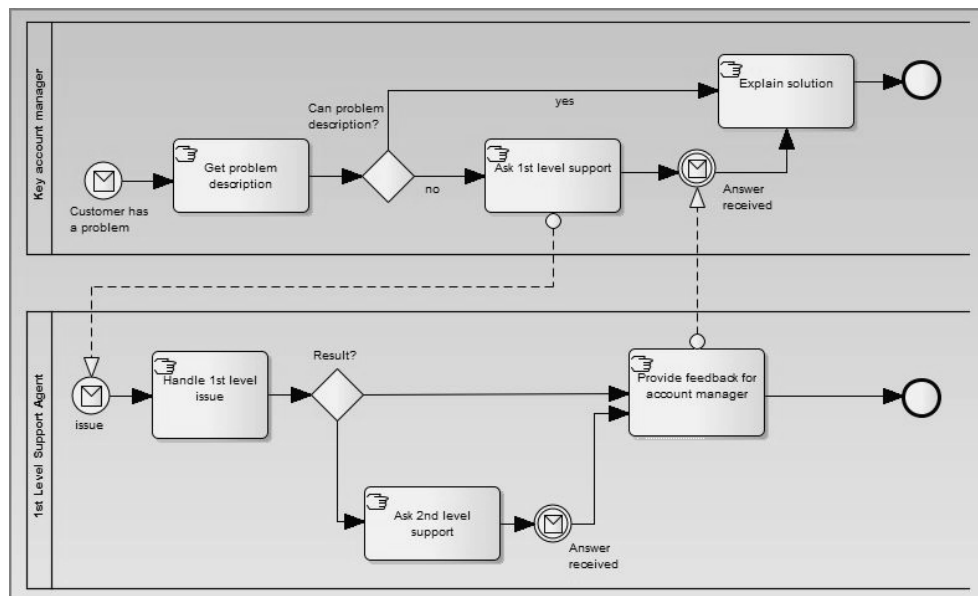
2.3.2 Diagram choreografie

Diagram choreografie (*Choreography diagram*) je zaměřen na interakci mezi jednotlivými účastníky procesu. Může být použit k analýze jejich komunikace pro zlepšení vzájemné koordinace. Protože je tento diagram určen k popisu vzájemného působení mezi účastníky procesu, nemůže choreografie existovat jen v rámci jednoho bazénu. Každá aktivita v diagramu reprezentuje komunikaci mezi dvěma a více účastníky procesu.

2.3.3 Diagram konverzace

Diagram konverzace (*Conversation diagram*) je zjednodušenou verzí diagramu spolupráce. Znázorňuje, kdy spolu kteří účastníci procesu komunikují. Jsou zde použity grafické elementy, které se v ostatních modelech nepoužívají – konverzační uzel (*Conversation Node*) a konverzační spoj (*Conversation Link*).

2. NOTACE BPMN



Obrázek 2.8: Diagram spolupráce [12]

Metodika DEMO

3.1 Úvod

Tato kapitola popisuje metodiku DEMO (*Design and Engineering Methodology for Organizations*), motivaci k jejímu vzniku, teoretický základ na kterém stojí, popis modelů které umožňuje vytvářet a postup při jejich tvorbě.

3.2 Motivace a podniková ontologie

DEMO je metodika určená pro modelování a analýzu podnikových procesů, kterou vytvořil prof. Jan Dietz. Motivací k jejímu vzniku byla podle [13] zvětšující se složitost podniků a nedostatečnost současně používaných nástrojů pro jejich modelování. Komunikace a interakce mezi jednotlivými účastníky procesů je velmi důležitá, ale žádné jiné modelovací techniky její zachycení nepodporovaly na požadované úrovni. Jak je uvedeno v [14], DEMO umožňuje pohled na podstatu organizace oproštěnou od realizace a implementace, tedy pohled podnikové ontologie – *Enterprise Ontology*.

Cílem podnikové ontologie je vytváření modelů organizací poskytujících ucelený pohled na jejich vnitřní fungování bez ohledu na implementaci. Takové modely by podle [15] měly splňovat pět kritérií, souhrnně nazývaných jako C4E:

- Jednoznačnost – Při modelování stejného procesu dojdeme vždy ke stejnému výslednému modelu.
- Kompletnost – Výsledný model je úplný, obsahuje všechny podstatné části.
- Konzistence – Je zřejmé, jak na sebe jednotlivé kroky navazují.
- Stručnost – Model je kompaktní a neobsahuje nadbytečné části.

- Esencialita – Model zobrazuje pouze podstatu podniku, nikoli implementaci.

3.3 Teorie PSI

Teorie Ψ je podle [13] teoretickým základem, na kterém DEMO stojí. Je to teorie, která se skládá ze čtyř axiomů a jednoho teorému. Cílem teorie je zobrazení podstaty organizace oprostěné od vnitřní implementace. V tom se tedy cíle DEMO a teorie PSI shodují. Bez pochopení základů teorie PSI nelze správně používat DEMO. V následujících podkapitolách jsou představeny všechny čtyři axiomy a organizační teorém. Popis vychází z [15].

3.3.1 Operační axiom

První, operační axiom Ψ teorie říká, že chod organizace se skládá z aktivit vykonávaných aktory.

Aktor je jediný aktivní element v organizaci. Není spouštěn žádnou událostí, ale jedná autonomně v rámci své agendy. V rámci teorie je vnímán jako kombinace zodpovědnosti, autority a kompetence k nějaké činnosti. Například lékař je *kompetentní* k vykonávání praxe, protože má odpovídající lékařské vzdělání. Pro to, aby mohl operovat v nemocnici, potřebuje mít *autoritativní* základ – být jejím zaměstnancem. A *zodpovědností* v tomto příkladu je to, že je vázán lékařskými normami a pravidly, která musí dodržovat.

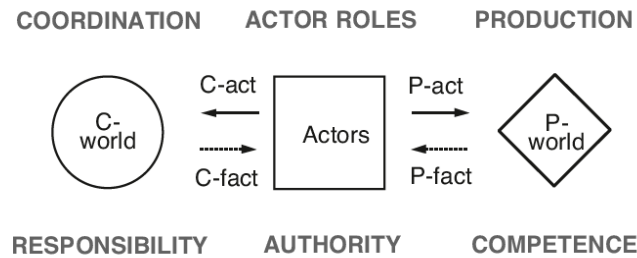
Aktoři svou činností provádějí koordinační a produkční akty, které vedou k vytvoření koordinačních a produkčních faktů.

Koordinační akt (*Coordination act, C-act*) má za následek vznik koordinačního faktu (*Coordination fact, C-fact*). Jde o akci prováděnou jedním aktorem (*Performer*) a adresovanou druhému aktorovi (*Addressee*). Pomocí těchto aktů aktoři komunikují o výsledném vzniku produkčního faktu. Množinu všech C-faktů označujeme jako *C-world*.

Produkční akty (*Production acts, P-acts*) potom vedou k vzniku produkčních faktů (*Production facts, P-facts*) jejichž množina je označována jako *P-world*. Produkční akty a fakty vytvářejí nové, hmotné nebo i nehmotné, produkty. Důležité je, že P-fact začíná existovat až v okamžiku provedení C-actů state a accept (bližší popis těchto C-actů je v následující části – Transakční axiom). Znázornění operačního axiomu můžeme vidět na obrázku 3.1.

3.3.2 Transakční axiom

První axiom říká, že chod organizace se skládá z aktivit, které vykonávají aktoři. Tím, jak jsou tyto aktivity propojené, a v jakých vzorech probíhají, se zabývá transakční axiom. Tento axiom říká, že všechny koordinační akty se odehrávají v univerzálním vzoru, nazývaném *transakce*. Každá transakce



Obrázek 3.1: Znárodnění operačního axiomu [15]

se skládá ze dvou konverzací, kde jako konverzací vnímáme sekvenci C-actů probíhající mezi dvěma aktory s cílem vytvoření P-factu.

Transakce probíhá mezi prvním aktorem, který požaduje vznik nějakého výsledku, a druhým aktorem, který tento výsledek pro prvního vytvoří. První aktor se nazývá iniciátor (*Initiator*) a druhý vykonavatel (*Executor*). Celou transakci můžeme rozdělit do tří fází:

- *Order* fáze – Aktoři se domlouvají na tom, co a kdy má být vytvořeno.
- *Execution* fáze – Výsledek je vytvořen.
- *Result* fáze – Aktoři se domlouvají, zda výsledek odpovídá požadavkům a bude akceptován. Až po dokončení této fáze začne výsledek podle PSI teorie existovat.

Jak už bylo výše uvedeno, transakce probíhají v univerzálním vzoru. Tento vzor se nazývá základní transakční vzor, jeho rozšířením je potom standardní transakční vzor a nejkompexnější je kompletní transakční vzor. V další části budou popsány blíže.

3.3.2.1 Základní transakční vzor

Pro základní popis transakce se používá základní transakční vzor. Počítá pouze s ideálním průběhem transakce, kdy není žádný akt zamítnut. Průběh transakce bude popsán na příkladu ostřihání v kadeřnictví: Zákazník přijde do kadeřnictví a řekne, jaký účes by chtěl vytvořit. Kadeřnice souhlasí a zákazníka ostříhá. Ten následně spokojeně odchází.

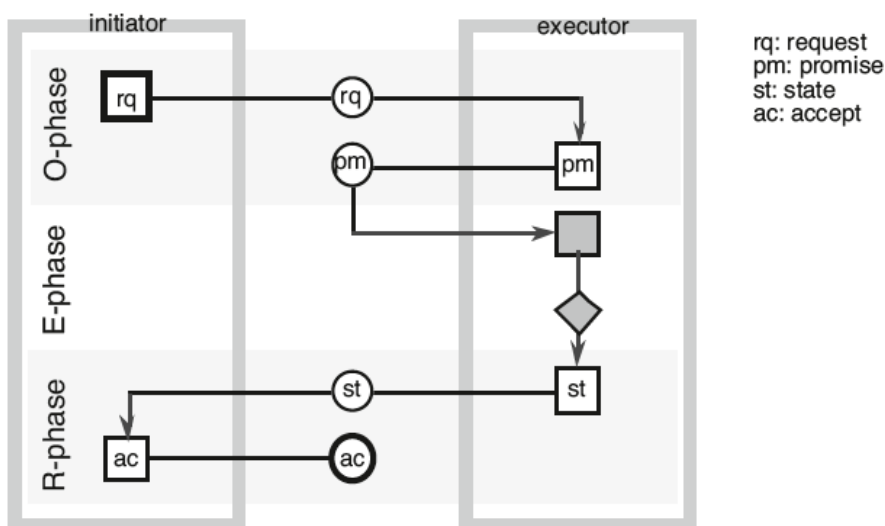
Transakční vzor popisuje následující posloupnost C-(f)actů vedoucí ke vzniku P-(f)actu:

- *Request* – Iniciátor požaduje vytvoření P-factu. V našem příkladu to je zákazníkova žádost o konkrétní účes.
- *Promise* – Executor slíbí vytvoření P-factu. Kadeřnice souhlasí.

3. METODIKA DEMO

- *Execute* – Provedení P-actu. Samotné ostříhání.
- *State* – Executor oznámí vytvoření výsledku. Kadeřnice řekne zákazníkovi, že je účes hotov.
- *Accept* – Iniciátor akceptuje výsledek. Zákazník je spokojen s výsledkem.

Průběh základního transakčního vzoru vidíme na obrázku 3.2. Bílé čtverce symbolizují jednotlivé C-acty, kruhy C-facty. Šedý čtverec reprezentuje P-act, kosočtverec potom P-fact. V obrázku jsou rozlišeni oba aktóři a zároveň je znázorněno, které části transakce spadají do order fáze, které do execution a které do result.



Obrázek 3.2: Základní transakční vzor [15]

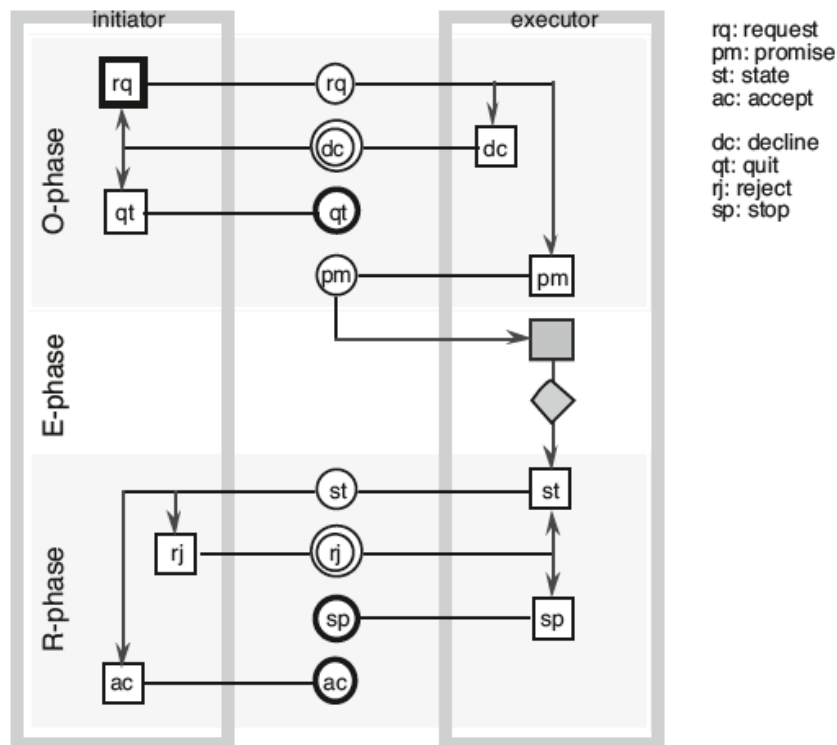
3.3.2.2 Standardní transakční vzor

Je zřejmé, že základní transakční vzor nepokrývá všechny situace, které mohou nastat. Co kdyby například kadeřnice neuměla vybraný účes vytvořit a zákazníka kvůli tomu odmítla? S těmito situacemi počítá standardní transakční vzor, který umožňuje odmítnutí requestu executorem a odmítnutí stavu state iniciátorem.

První případ, odmítnutí requestu, vede do stavu nepřijmuto – *declined*. V tomto stavu se iniciátor rozhoduje, zda celou transakci ukončí (*quit*) nebo vytvoří nový request. V našem příkladu by tedy kadeřnice zákazníkovi oznámila, že vybraný účes neumí, tedy že jeho požadavek nepřijme. Zákazník v tu chvíli může buď odejít, a tím transakci ukončit, nebo si vybrat jiný účes a vytvořit nový požadavek.

Odmítnutí stavu *state* vede do stavu *rejected*. Executor se rozhodne, zda je odmítnutí oprávněné, a podle toho buď transakci ukončí (*stop*) nebo se vrátí zpět do stavu *state*. Na příkladu kadeřnictví by *reject* nastal tehdy, pokud by zákazník nebyl po ostříhání spokojen s výsledkem. Kadeřnice posoudí, zda je nespokojenost oprávněná, a podle toho transakci ukončí nebo ne.

Standardní transakční vzor můžeme vidět na obrázku 3.3, popis je stejný jako v předchozím případě.



Obrázek 3.3: Standardní transakční vzor [15]

3.3.2.3 Kompletní transakční vzor

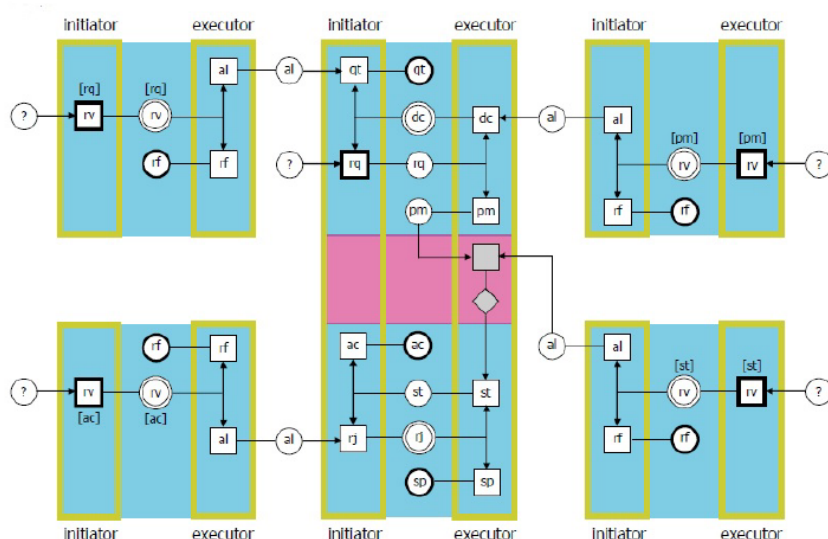
Poslední variantou je kompletní transakční vzor. Do standardního transakčního vzoru přidává možnost odvolání jakéhokoli již provedeného C-factu. Toto odvolání se provádí podle tzv. odvolávacích vzorů, revoků. Odvolání může nastat pouze tehdy, když je schváleno druhým aktorem. Na příkladu kadeřnictví lze odvolávací vzory ukázat například takto:

- **Revoke Request** – Zákazník oznámil, který účes by chtěl, pak svůj výběr ale změnil.

3. METODIKA DEMO

- Revoke Promise – Kadeřnice souhlasila, ale potom si vybraný účes lépe prohlédla a zjistila, že ho vytvořit neumí.
- Revoke State – Kadeřnice oznámila, že je účes hotov, ale pak si na něm všimla chyby.
- Revoke Accept – Zákazník si doma výsledný účes lépe prohlédl, našel nedostatek, a rozhodl se vrátit s žádostí o opravu.

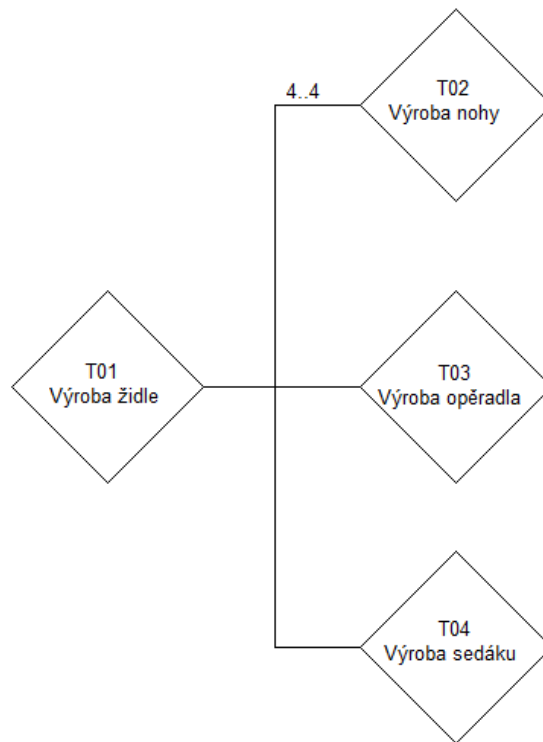
Kompletní transakční vzor je na obrázku 3.4



Obrázek 3.4: Kompletní transakční vzor [16]

3.3.3 Kompoziční axiom

Kompoziční axiom vyjadřuje, jak jsou jednotlivé transakce vzájemně propojené a závislé. Říká, že každá transakce je součástí jiné transakce, je transakcí zákazníka, nebo je self-activated. Jako příklad závislosti transakcí můžeme uvést například výrobu židle. Pro výrobu jedné židle je potřeba vyrobit jedno opěradlo, jeden sedák, a čtyři nohy. Pro dokončení transakce $T01$ – *Výroba židle* je tedy potřeba nejdříve čtyřikrát provést transakci $T02$ – *Výroba nohy*, jednou $T03$ – *Výroba opěradla* a jednou $T04$ – *Výroba sedáku*. $T01$ je dokončena, až když jsou dokončeny všechny transakce $T02$ – $T04$. Grafické vyjádření tohoto příkladu je na obrázku 3.5.



Obrázek 3.5: Znázornění kompozičního axiomu

3.3.4 Distinkční axiom

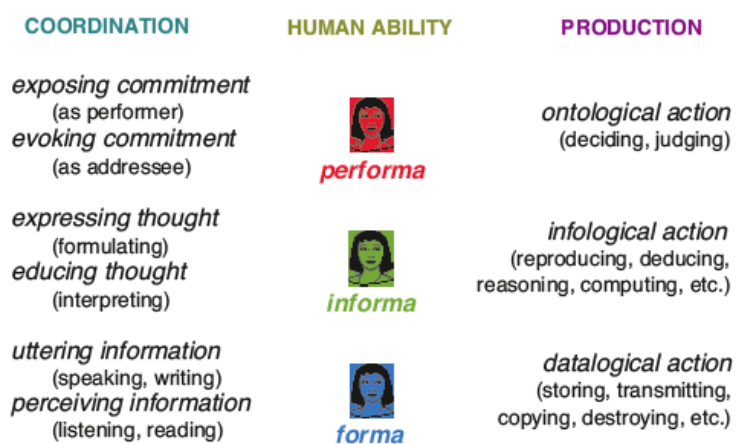
Distinkční axiom vychází z myšlenky, že činnost aktorů zahrnuje tři lidské schopnosti, nazývané *performa*, *informa* a *forma*. Nejvyšší úrovní jsou *performa* aktivity, jejichž cílem je vytváření nových věcí. K provedení *performa* aktivity je obvykle potřeba vykonat několik *informa* aktivit, které zahrnují to, jaký je obsah dané informace. *Informa* aktivity stojí na *forma* aktivitách, ve kterých jde o způsob předávání a uchovávání informací. Toto rozdělení je velmi důležité, protože v procesních modelech budou zahrnuty pouze *performa* aktivity. Na obrázku 3.6 vidíme ilustrační shrnutí distinkčního axiomu. Barevné rozdělení – *performa* červeně, *informa* zeleně a *forma* modře je použito v celém zbytku práce.

3.3.5 Organizační teorém

Organizační teorém říká, že organizace je heterogenní systém skládající se ze tří homogenních systémů:

- B–organizace (*Business*)
- I–organizace (*Intellect*)

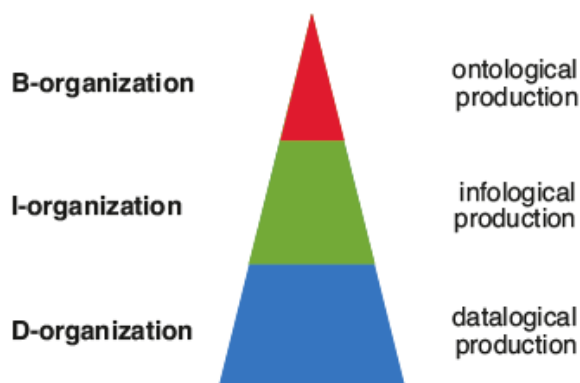
3. METODIKA DEMO



Obrázek 3.6: Distinkční axiom [15]

- D–organizace (*Document*)

Tyto tři systémy na sobě závisejí: D–organizace podporuje I–organizaci a ta B–organizaci. Reprezentace vzájemného propojení organizací je na obrázku 3.7. Pro modelování v DEMO je nejdůležitější B–organizace, kterou vidíme na vrcholu pyramidy. Tato úroveň je nejvyšší a poskytuje kompletní znalost podstaty podniku, oprostěnou od realizace a implementace.

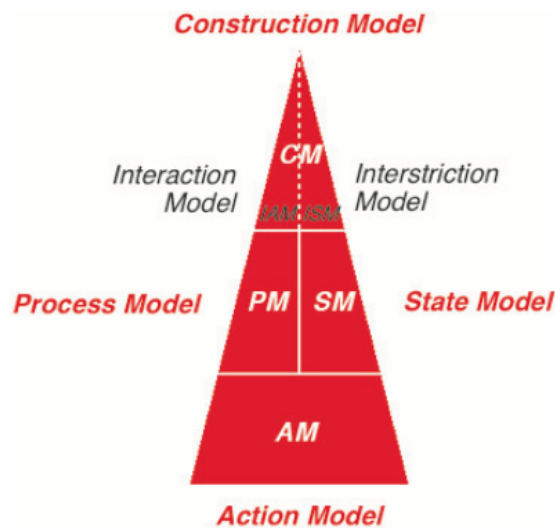


Obrázek 3.7: Organizační teorém [15]

3.4 Modely v DEMO

Tato kapitola si klade za cíl stručně představit jednotlivé modely, ze kterých je DEMO složeno a seznámit tak čtenáře se základy potřebnými pro následné transformace DEMO modelů do BPMN. Podrobnější popis jednotlivých modelů a elementů, ze kterých se modely skládají, lze nalézt například v [15]. Z této publikace a z [14] tato kapitola čerpá.

Základem DEMO jsou čtyři modely, které se dále dělí na další podmodely. Každý model poskytuje jiný pohled na organizaci, od nejdetailnějšího Action Modelu po nejabstraktnější Construction Model. Toto odstupňování úrovně abstrakce lze názorně vyjádřit jako pyramidu, kterou můžeme vidět na obrázku 3.8. Při modelování je základním elementem ontologická transakce, tedy transakce vytvářející nové, hmotné či nehmotné věci, a každá tato transakce se odehrává mezi dvěma aktory, z nichž je jeden iniciator a druhý executor.



Obrázek 3.8: Struktura DEMO modelů [15]

3.4.1 Construction Model (CM)

Jak již bylo uvedeno, Construction model poskytuje nejabstraktnější pohled na konstrukci organizace. Popisuje jednotlivé transakce, aktory, a informační spoje mezi nimi. Skládá se z dalších dvou podmodelů:

- *Interaction Model* (IAM) – Model skládající se ze dvou částí. První z nich je *Transaction Result Table* (TRT), tabulka popisující jednotlivé typy transakcí a jejich výsledky. Druhou částí potom je *Actor Transaction Diagram* (ATD) popisující vztahy mezi aktory a transakcemi.

- *Interstriction Model* (ISM) – Model popisující závislosti mezi aktory. Skládá se ze dvou diagramů a jedné tabulky.

3.4.2 Process Model (PM)

Process Model je na nižší úrovni abstrakce než CM a přináší pohled na průběh jednotlivých transakcí. Transakce probíhají podle transakčního vzoru a zároveň jsou zde popsány vzájemné závislosti mezi jednotlivými transakcemi. Stále je ovšem popis oproštěn od vnitřní implementace. I tento model se skládá z více částí, konkrétně z jednoho diagramu a jedné tabulky:

- *Process Structure Diagram* (PSD) – Diagram zobrazuje strukturu procesů, popisuje vztahy mezi transakcemi a jejich kroky.
- *Information Use Table* (IUT) – Tabulka je úzce provázaná se State Modelem, specifikuje ve kterých krocích PM se používají instance ze SM.

3.4.3 State Model (SM)

Model specifikuje stavový prostor P–worldu – objektové třídy, typy výsledků a ontologická pravidla. Model je přímo spojený s Action Modelem, specifikuje prvky, které jsou v něm obsaženy. Stejně jako PM se skládá ze dvou částí – jednoho diagramu a z tabulky.

- *Object Fact Diagram* (OFD) – Popis vztahu mezi objektovými třídami a typy výsledků.
- *Object Property List* (OPL) – Popis objektových tříd.

3.4.4 Action Model (AM)

Na nejnižší úrovni abstrakce se nachází Action Model. Popisuje akční pravidla, kterými se řídí aktoři při svých činnostech. Tento model neobsahuje žádné diagramy ani tabulky, akční pravidla jsou popsána pseudoalgoritmickým jazykem a specifikují co se děje ve stavech requested, promised, stated a accepted.

Možnosti převodu DEMO modelů do notace BPMN

4.1 Úvod

Cílem této kapitoly je popsat existující možnosti převodu DEMO modelů do BPMN. Práce se věnuje zejména dvěma metodám, které byly navrženy v diplomových pracích [11] a [17]. Proč ale vůbec modely převádět? Na tuto otázku odpovídá následující kapitola o motivaci k převodům.

4.1.1 Motivace k převodům

V předchozích dvou kapitolách jsou popsány dva různé způsoby tvorby modelů podnikových procesů, DEMO a BPMN. Zde budou stručně popsány jejich silné a slabé stránky, popis vychází z [11].

Silnou stránkou metodiky DEMO je, že poskytuje přesný postup tvorby modelů. To je dáno silným teoretickým základem, na kterém stojí. Tento teoretický základ je ovšem pro uživatele velmi složitý, naučit se vytvářet správné DEMO modely není jednoduché. Problém pak je i s čitelností těchto modelů. Když budeme model vytvořený podle zásad DEMO prezentovat uživatelům, kteří tuto metodologii neznají, nebudou mu rozumět. DEMO není příliš rozšířeno, a tak neexistuje mnoho nástrojů, ve kterých by bylo možné modely vytvářet. Stejně tak existuje pouze malé množství literatury, která se DEMO zabývá.

Notace BPMN je velmi oblíbená a rozšířená, existuje velké množství nástrojů, které tvorbu modelů podle ní umožňují. Literatura o BPMN je dobře dostupná. I pro nezasvěcené uživatele jsou modely čitelné. Pro vytvoření BPMN modelu není potřeba dlouhé studium specifikace. Slabinou je absence teoretického základu, který by popisoval, jak při tvorbě modelů postupovat. Vznikají pak modely, které jsou nepřesné a neúplné. Pokud budou dva lidé modelovat stejný proces, pravděpodobně dojdou k odlišnému výsledku.

Z popisu můžeme vidět, že se silné a slabé stránky obou způsobů doplňují. Jejich vzájemnou kombinací můžeme získat postup tvorby modelu stojící na silném teoretickém základu, ale zároveň budou výsledné diagramy pro uživatele srozumitelné. Cílem je vytváření BPMN modelů, které budou jednoznačné, kompletní a konzistentní.

4.2 Existující možnosti převodů

4.2.1 Enhancing the Formal Foundations of BPMN by Enterprise Ontology

Tato práce [18] přináší teoretický rozbor možností kombinace DEMO a BPMN. První z možností je analýza už existujícího BPMN modelu a jeho doplnění o chybějící elementy podle zásad DEMO. Druhou možností je vytvoření nového BPMN modelu podle pravidel vycházejících z teoretického základu DEMO. Vytváření modelu začíná sběrem informací o organizaci. Následuje aplikace distinkčního axiomu. Následně jsou identifikováni aktori a ontologické transakce. Na závěr je vytvořen ATD popisující vztahy mezi aktory a transakcemi.

Výstupem práce je tedy teoretický postup při vytváření BPMN modelu podle zásad DEMO. Na tomto postupu staví práce [11] a dále ho rozšiřuje.

4.3 Metoda Štěpána Hellera

V této kapitole je popsána metoda převodu DEMO modelu do notace BPMN, která vznikla v rámci diplomové práce *Možnosti využití metodiky DEMO pro tvorbu BPMN modelů* [11]. Práce neobsahuje pouze metodu převodu, ale kompletní postup vytváření BPMN modelů s pomocí DEMO. Tento postup je zde popsán, a celý je následně předveden v kapitole věnované příkladům převodů.

4.3.1 Postup při vytváření BPMN modelu

Autor při tvorbě postupu vychází z návodu pro vytváření DEMO modelů uvedeného v [13], teoretického postupu z [18], a upravuje ho na následující postup: [11]

1. **Získání textového popisu procesu**
2. **Aplikace distinkčního axiomu** – Aplikace Performa – Informa – Forma analýzy na popis procesu.
3. **Aplikace operačního axiomu** – Rozdělení performa aktivit z předchozího kroku na C-acty/facty, P-acty/facty a identifikace příslušných aktorů.

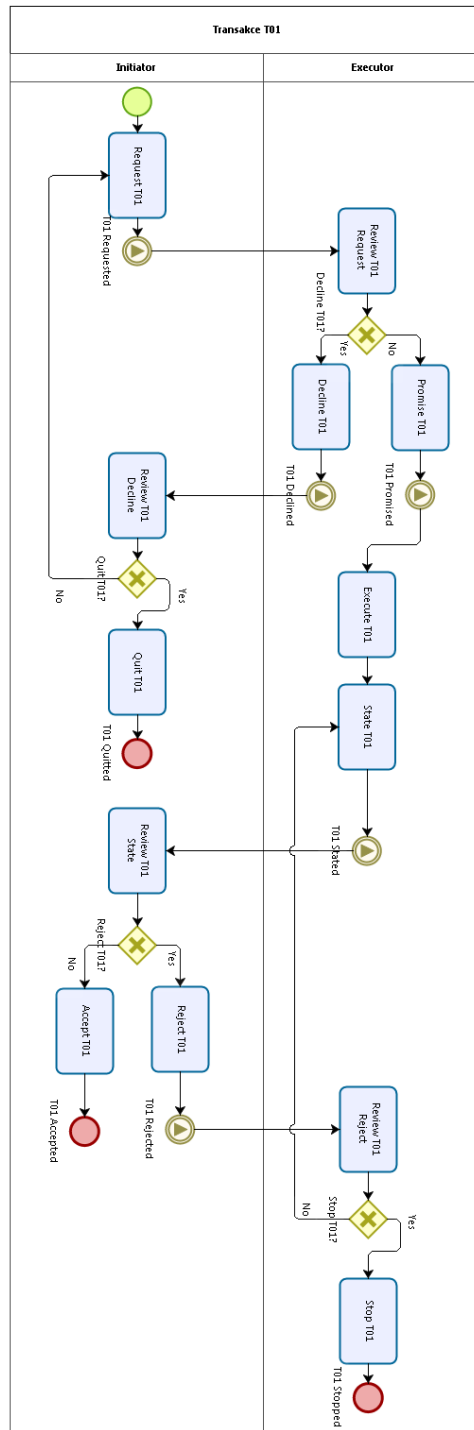
4. **Zápis nalezených transakcí a jejich parametrů** – Zapsání aktivit, označených v předchozím kroku, podle transakčního axiomu do tabulky.
5. **Aplikace kompozičního axiomu** – Identifikace vzájemných vazeb mezi transakcemi a jejich zakreslení do jednoduchého diagramu.
6. **Vytvoření DEMO modelů** – Vytvoření Actor–Transaction Diagramu (ATD) a Process Structure Diagramu (PSD).
7. **Vytvoření BPMN modelu** – Jednotlivé části PSD se mapují na BPMN elementy podle metody popsané v následující sekci.

4.3.2 Převod do BPMN

V práci je představeno několik možností, jak převést základní a standardní transakční vzor do BPMN. Bohužel zde není uvažován komplexní transakční vzor. Mezi zavržené možnosti vyjádření transakčního vzoru patří vyjádření pomocí úloh a sekvenčních toků, které neumožňuje zobrazení komunikace o vzniku C–factu, a vyjádření pomocí úloh a zpráv, které se v případě standardního transakčního vzoru příliš odchyluje od transakčního axiomu.

Třetí možností, autorem zvolenou jako nejvhodnější, je vyjádření pomocí úloh a signálů. Vznik C–factu je znázorněn BPMN elementem signál, který na rozdíl od zpráv umožňuje komunikaci v rámci jednoho procesu. Pouze C–fauty, ve kterých transakce končí, jsou vyjádřeny elementem konečná událost, nikoli signálem. Předávání agendy mezi aktory a pořadí C–actů je znázorněno pomocí sekvenčních toků. Celá transakce se odehrává uvnitř bazénu, jednotliví aktoři jsou zobrazeni pomocí plaveckých drah. Příklad standardního transakčního vzoru převedeného do BPMN touto metodou můžeme vidět na obrázku 4.1.

4. MOŽNOSTI PŘEVODU DEMO MODELŮ DO NOTACE BPMN



Obrázek 4.1: Standardní transakční vzor pomocí Úloh a Signálů podle [11]

4.4 Metoda Ondřeje Mráze

Tato kapitola popisuje metodu převodu z diplomové práce *Možnosti využití principů metodiky DEMO pro zvýšení kvality BPMN modelů* [17], jejímž autorem je Ondřej Mráz. Celá kapitola z této práce vychází.

4.4.1 Metoda převodu

Jde o první metodu převodu kompletního transakčního vzoru z DEMO do BPMN. Důraz je kladen na to, aby výsledný model dodržoval všechny axiomy teorie PSI a aby splňoval požadavky na model podle [18].

Na rozdíl od předchozí metody se zde počítá s tím, že už jsou některé DEMO konstrukty vytvořeny, nezačíná se od začátku ziskem popisu procesu. Je potřeba mít vytvořeno následující:

- Actor Transaction Diagram
- Process Structure Diagram
- Actor–function matrix – nepovinně
- TPT tabulka – nepovinně

4.4.1.1 Vyjádření transakcí a aktorů

Každá transakce je vyjádřena pomocí jednoho bazénu se dvěma plaveckými drahami reprezentujícími aktory. Pokud se tedy proces skládá z více transakcí, pak jejich počet odpovídá počtu bazénů. Kořenová transakce je spouštěna uživatelem, všechny další transakce jsou spouštěny z rodičovské transakce. Actoři jsou vyjádřeni pomocí plaveckých drah. Podle DEMO má být aktor pojmenován podle své role v procesu a každý může provádět pouze jednu transakci. BPMN naopak umožňuje pojmenovávat plavecké dráhy libovolně a neomezuje počet vykonaných procesů. Propojení těchto dvou přístupů napomáhá nepovinná Actor–function matrix, která vyjadřuje mapování rolí v procesu na reálné osoby v organizaci.

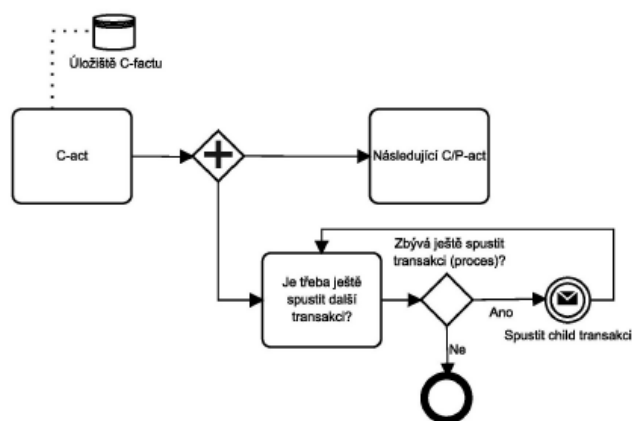
4.4.1.2 Vyjádření C–actů, C–factů a P–actů, P–factů

C–acty se, stejně jako v práci Š.Hellera vyjadřují pomocí úloh, rozdíl je ovšem ve vyjádření C–factů. Mráz upozorňuje na problém s použitím signálu – v modelu chybí událost která by signál zachycovala – a přichází s vlastním řešením v podobě použití elementu datové úložiště. Ve chvíli, kdy je C–act dokončen, je do datového úložiště vložen stav transakce a zároveň je jím reprezentováno vytvoření C–factu. Grafická reprezentace C–factu, i se zkrácenou verzí, je na obrázku 4.7

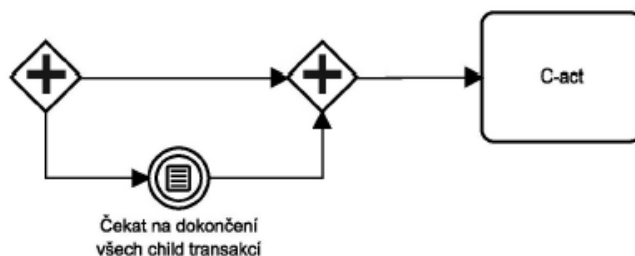
P–facty nejsou v modelu explicitně vyjádřeny, znázorňuje se jen P–act po jehož dokončení P–fact vzniká. Kvůli znázornění kompletního transakčního vzoru je nutné umožnit odvolání P–factu. K tomu byl využit element kompenzace, jehož catching událost je připojena k aktivitě vyjadřující P–act. Pokud dojde k Revoke State, pak je vyvolána throwing událost typu kompenzace, ta je zachycena catching událostí a provede se zrušení P–factu.

4.4.1.3 Vyjádření kompozičního axiomu

Spouštění podřízených transakcí z rodičovské je řešeno pomocí zpráv. V rodičovské transakci je proveden throwing události typu zpráva, tolikrát, kolikrát má být potomek spuštěn. Podřízená transakce potom začíná startovací událostí typu zpráva. Tento postup je na obrázku 4.2. K zablokování transakce rodiče, dokud není podřízená transakce dokončena, slouží událost typu podmínka. Transakce čeká, dokud nejsou všechny child transakce dokončeny a až potom pokračuje dál. Příklad je na obrázku 4.3.



Obrázek 4.2: Spouštění podřízené transakce [17]

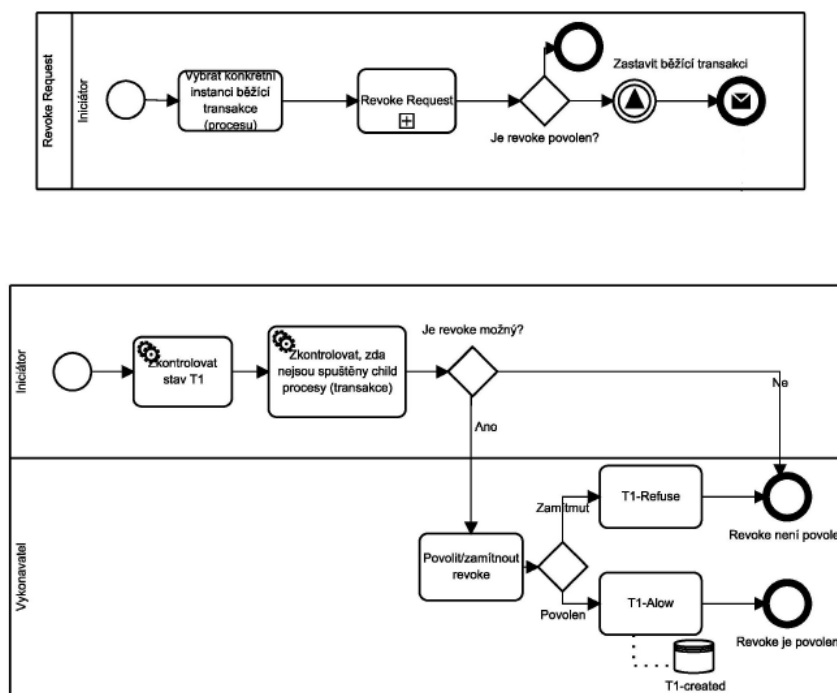


Obrázek 4.3: Čekání na dokončení child transakce [17]

4.4.1.4 Vyjádření odvolávacích vzorů

Ve chvíli, kdy jeden aktor revoke vyvolá, je potřeba ověřit splnění podmínek k jeho provedení, následně je potřeba potvrzení druhým aktorem, a až poté je skutečně proveden. Mezi podmínky pro provedení patří ověření, že revoke odvolá pouze jednu instanci procesu, že se proces aktuálně nachází ve stavu, kdy lze revoke provést a že jsou odvolány všechny případné child procesy.

Provedení revoku probíhá tak, že hned po startovacím symbolu procesu se tok rozdělí na dvě cesty. První je standardní, druhá obsahuje catching událost typu signál a je následována koncovou událostí. Revoke při aktivaci provede throwing signálu, ten je zachycen catching událostí a následně je proces zastaven. Potom revoke vyšle zprávu, která znovu proces spustí, ale z jiného místa. Toto je obecný popis provedení revoku, u některých typů odvolání lze některé kroky vynechat. Činnost navíc je naopak potřeba provést u Revoke State, kdy je vyvolávána událost typu kompenzace. Příklad revoku je na obrázku 4.4.



Obrázek 4.4: Příklad vyjádření Revoke [17]

4.4.1.5 Příklad modelu

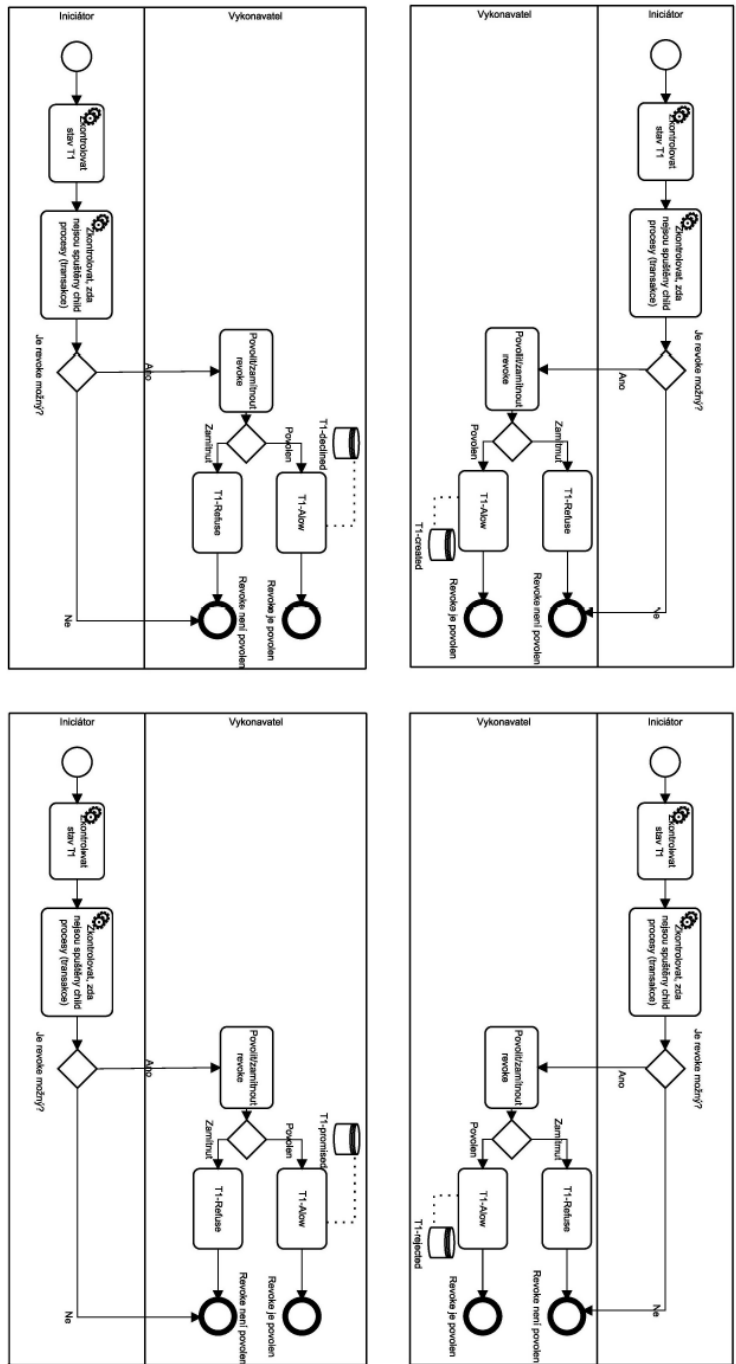
Příklad transakce převedené z DEMO do BPMN podle výše popsané metody můžeme vidět na obrázku 4.5 a podprocesy revoků na obrázku 4.6. Jde o kom-

4.4.2 Nové elementy pro zjednodušení modelů

V práci bylo navrženo několik nových elementů, které zkracují a zjednodušují výsledné BPMN modely. V této podkapitole jsou představeny a následně využity v kapitole s příklady převodů. Použití těchto symbolů znemožňuje následnou simulaci procesu, protože nejsou součástí specifikace BPMN. Pro jejich použití je nutné je ručně přidat do modelovacího nástroje. Došlo ke zkrácení tří následujících konstruktů:

1. **Vyjádření C-actů a C-factů** – Jak už bylo popsáno v předcházející části, metoda využívá pro vyjádření C-actů symbol úlohy a pro vyjádření C-factů symbol datového úložiště. Pro zkrácení byl navržen nový symbol spojující tyto dva prvky do jednoho. Jde o symbol úlohy se symbolem datového úložiště v její pravé horní části. Výsledek je na obrázku 4.7.
2. **Spouštění child transakcí** – Metoda potřebuje pro spuštění jedné child transakce celkem čtyři symboly. Byl proto navržen symbol, který všechny čtyři slučuje do jednoho. Uvnitř symbolu jsou uvedena dvě čísla, první značí minimální počet vyvolaných transakcí, druhé pak jejich maximální počet. Původní, nezkrácené, volání child transakce a nově navržený symbol vidíme na obrázku 4.8.
3. **Vyjádření odvolávacích vzorů** – Nejrozsáhlejší částí, u které došlo ke zkrácení, je vyjádření odvolávacích vzorů. Původně byly vyjádřeny jako čtyři bazény, ve kterých probíhají další procesy. Toto vyjádření výrazně zvyšovalo rozsáhlost výsledného modelu. V rámci zkrácení byly navrženy čtyři nové symboly, pro každý revoke jeden. Symbol reprezentuje celý proces odvolání a je znázorněn pomocí startovacího symbolu s popisem *Rv-R* pro Revoke Request, *Rv-P* pro Revoke Promise, *Rv-S* pro Revoke State nebo *Rv-A* pro Revoke Accept. Nové symboly jsou na obrázku 4.9.

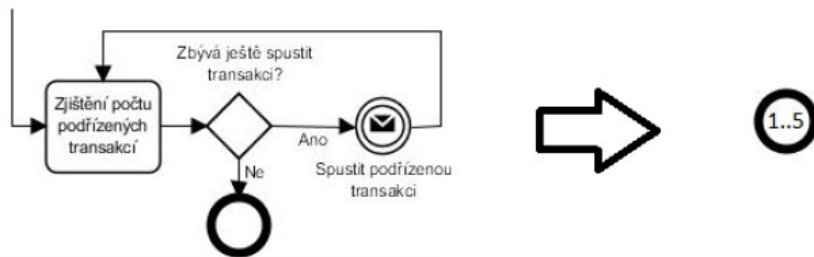
4. MOŽNOSTI PŘEVODU DEMO MODELŮ DO NOTACE BPMN



Obrázek 4.6: Podprocesy revoků podle Mráze [17]



Obrázek 4.7: Spojení symbolů pro C-acty a C-facty [17]



Obrázek 4.8: Symbol pro spouštění child transakcí [17]



Obrázek 4.9: Symboly pro revoky [17]

Příklady převodů

5.1 Úvod

V této kapitole jsou na dvou příkladech demonstrovány výše popsané metody převodu DEMO modelů do notace BPMN. Jako první je v obou případech uveden převod podle metody Š. Hellera, převod podle O. Mráze je uveden jako druhý. Na prvním příkladu je ukázán celý postup tvorby BPMN modelu, který je uveden v [11]. Samotné příklady nejsou odnikud převzaty, jde o vlastní tvorbu.

5.2 Příklad 1 – Ocenění nemovitosti podle Š. Hellera

První příklad popisuje proces ocenění nemovitosti. Jde o administrativně náročný proces, proto do něj vstupuje více aktorů. Klient komunikuje přímo jen s poradcem, který práci předává dále vystaviteli, ten vytvoří zakázku a předá ji znalci k finálnímu vytvoření odhadu.

5.2.1 Získání textového popisu procesu

Pokud se klient rozhodne nechat ocenit svou nemovitost u společnosti Odhady s.r.o, tak musí kontaktovat některého z poradců, jejichž seznam spolu s kontakty je na webu společnosti, a podat u něj žádost o odhad. Poradce s klientem projedná details, a následně kontaktuje některého z vystavitelů s požadavkem na vytvoření zakázky. Vystavitel v aplikaci vystaví novou zakázku, kterou odešle znalci ke zpracování. Znalec zakázku převezme a začne se věnovat tvorbě odhadu ceny. Může se stát, že v zadání zakázky chybí nějaké důležité informace, bez kterých nelze odhad dokončit. V tom případě vrací znalec zakázku vystaviteli k dopracování. Pokud je vše v pořádku, znalec vytvoří odhad ceny, který nahraje jako přílohu zakázky v aplikaci a odešle vystaviteli. Vystavitel

odhad zkontroluje, a pokud s ním souhlasí, tak ho akceptuje, zakázku zkompletuje a předá ji poradci. Jestliže nesouhlasí, například z důvodu chybějících povinných částí odhadu, tak může ocenění vrátit k dopracování znalci. Klientovi je potom poradcem předán výsledný odhad.

5.2.2 Aplikace distinkčního axiomu

Na popis z předchozího bodu je provedena Performa–Informa–Forma analýza. Stejně jako v [15] jsou Performa aktivity označeny červeně, Informa zeleně a Forma modře.

Pokud se klient **rozhodne** nechat **ocenit** svou nemovitost u společnosti Odhady s.r.o, tak musí **kontaktovat** některého z poradců, jejichž **seznam** spolu s kontakty je na webu společnosti, a **podat** u něj **žádost** o odhad. Poradce s klientem **projedná** detaily, a následně **kontaktuje** některého z vystavitelů s požadavkem na **vytvoření zakázky**. Vystavitel v aplikaci **vystaví** novou **zakázku**, kterou **odešle** znalci ke **zpracování**. Znalec zakázku převezme a začne se věnovat **tvorbě odhadu** ceny. Může se stát, že v zadání zakázky chybí nějaké důležité informace, bez kterých nelze odhad dokončit. V tom případě **vrací** znalec **zakázku** vystaviteli k **dopracování**. Pokud je vše v pořádku, znalec **vytvoří** odhad ceny, který **nahraje** jako **přílohu zakázky** v aplikaci a **odešle** vystaviteli. Vystavitel odhad zkontroluje, a pokud s ním **souhlasí**, tak ho **akceptuje**, **zakázku zkompletuje** a **předá** ji poradci. Jestliže **nesouhlasí**, například z důvodu chybějících povinných částí odhadu, tak může ocenění **vrátit** k **dopracování** znalci. Klientovi je potom poradcem **předán** výsledný odhad.

5.2.3 Aplikace operačního axiomu

Performa aktivity byly rozděleny na C–acty/facty (kulaté závorky) a P–acty/facty (špičaté závorky). Aktory označují hranaté závorky.

Pokud se [klient] (**rozhodne** nechat **ocenit**) svou nemovitost u společnosti Odhady s.r.o, tak musí **kontaktovat** některého z [poradců], jejichž **seznam** spolu s kontakty je na webu společnosti, a (**podat** u něj **žádost**) o odhad. [Poradce] s [klientem] (**projedná**) detaily, a následně (**kontaktuje**) některého z [vystavitelů] s požadavkem na <**vytvoření**> **zakázky**. [Vystavitel] v aplikaci <**vystaví**> novou **zakázku**, kterou **odešle** [znalci] ke (**zpracování**). [Znalec] **zakázku** převezme a začne se věnovat <**tvorbě odhadu**> ceny. Může se stát, že v zadání zakázky chybí nějaké důležité informace, bez kterých nelze odhad dokončit. V tom případě **vrací** [znalec] **zakázku** [vystaviteli] k (**dopracování**). Pokud je vše v pořádku, [znalec] <**vytvoří**> odhad ceny, který **nahraje** jako **přílohu zakázky** v aplikaci a (**odešle**) [vystaviteli]. [Vystavitel] odhad zkontroluje, a pokud s ním **souhlasí**, tak ho (**akceptuje**), **zakázku zkompletuje** a (**předá**) ji [poradci]. Jestliže **nesouhlasí**, například z důvodu chybějících povinných částí odhadu, tak může ocenění **vrátit** k (**dopracování**) [znalci]. [Klientovi] je potom [poradcem] (**předán**) výsledný odhad.

Tabulka 5.1: Transaction Result Table procesu ocenění nemovitosti

Transakce	T01	T02	T03
Název transakce	Vytvoření ocenění nemovitosti	Vytvoření zakázky	Vypracování odhadu
Výsledek transakce	Ocenění předáno klientovi	Zakázka zpracována	Odhad nahrán k zakázce
Initiator	Klient	Poradce	Vystavitel
Executor	Poradce	Vystavitel	Znalec
Request	Podána žádost o odhad	Požadavek na vytvoření zakázky	Zakázka odeslána znalci
Promise	Príslib vytvoření odhadu	Príslib vytvoření zakázky	Převzetí zakázky znalcem
State	Odhad předán klientovi	Předání zakázky	Nahrání odhadu k zakázce
Accept	Odhad převzat	Zakázka převzata	Akceptace vystavitelem
Decline	Odmítnutí žádosti	Odmítnutí požadavku	Vrácení k dopracování vystaviteli
Reject	Převzetí odmítnuto	Zakázka odmítnuta	Vrácení k dopracování znalci

5.2.4 Zápis nalezených transakcí a jejich parametrů

V textovém popisu procesu byly nalezeny následující transakce:

- Vytvoření ocenění nemovitosti
- Vytvoření zakázky
- Vypracování odhadu

Parametry transakcí byly následně zapsány do tabulky 5.1

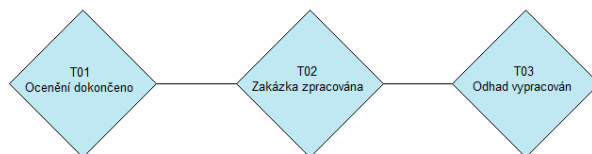
5.2.5 Aplikace kompozičního axiomu

Dalším krokem je aplikování kompozičního axiomu, tedy určení pořadí provádění transakcí a jejich vzájemné závislosti. Výsledek pro proces ocenění nemovitosti je znázorněn na obrázku 5.1.

5.2.6 Vytvoření ATD a PSD diagramů

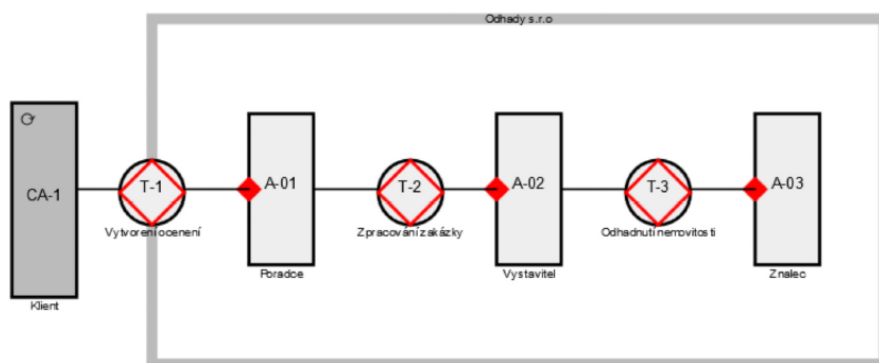
V rámci předposledního kroku jsou vytvořeny Actor Transaction Diagram a Process Structure Diagram. První jmenovaný můžeme vidět na obrázku 5.2. Mimo organizaci Odhady s.r.o se nachází aktor reprezentující klienta, v diagramu označen jako CA-1. Ostatní aktoři jsou uvnitř organizace, A01 je

5. PŘÍKLADY PŘEVODŮ



Obrázek 5.1: Pořadí a závislosti transakcí

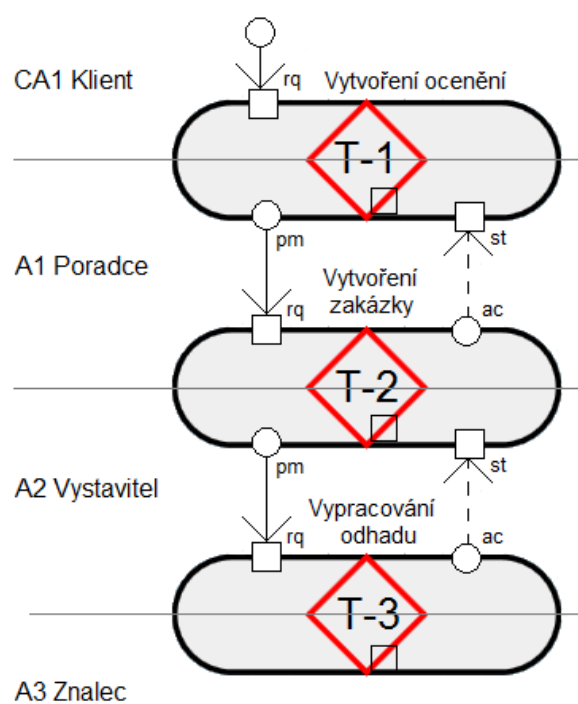
poradce, A02 vystavitel a A03 znalec. Přesnější pohled na provázanost jednotlivých transakcí potom poskytuje PSD na obrázku 5.3.



Obrázek 5.2: Actor Transaction Diagram

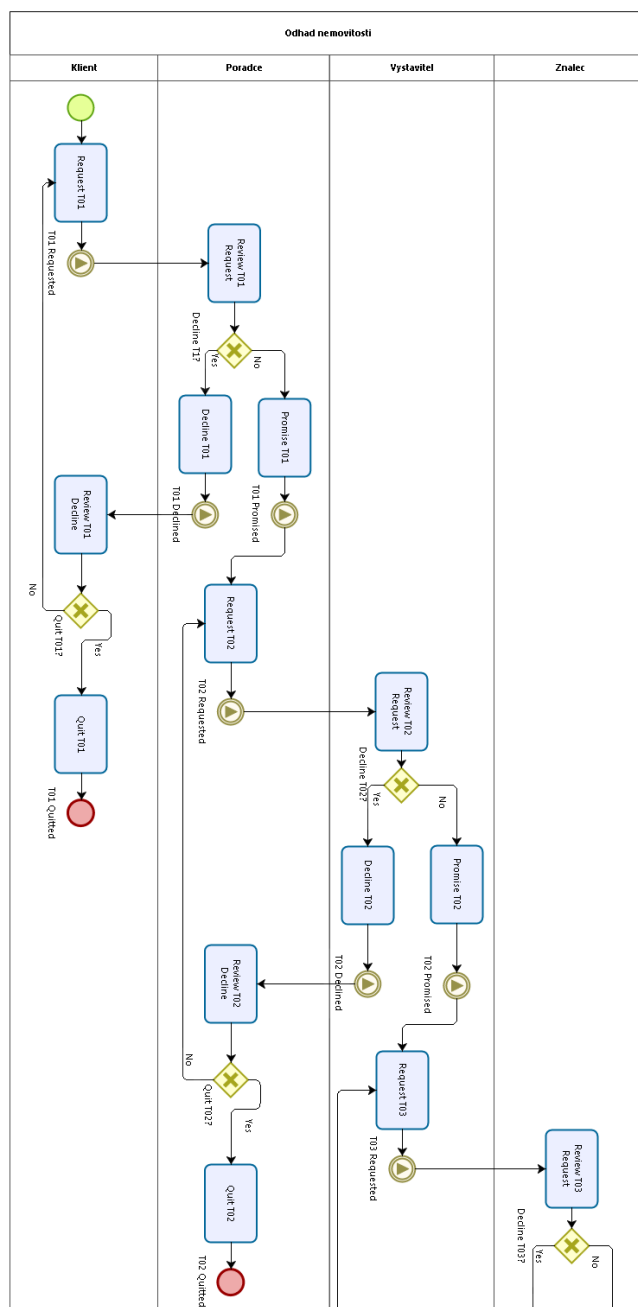
5.2.7 Vytvoření BPMN modelu

Posledním krokem je vytvoření samotného BPMN modelu. Na obrázcích 5.4, 5.5 a 5.6 je vidět výsledný diagram standardního transakčního vzoru v BPMN. Pro větší přehlednost nejsou v modelu popsány jednotlivé události textově, ale zástupnými jmény, jako Request T01 apod.



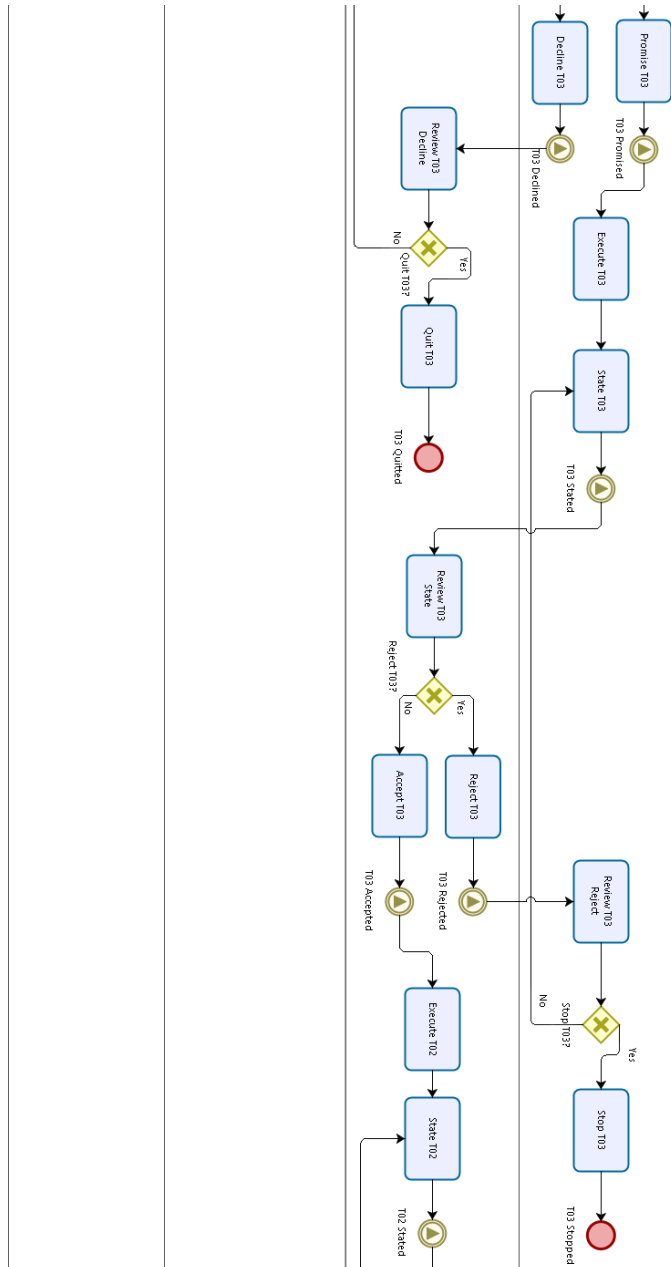
Obrázek 5.3: Process Structure Diagram

5. PŘÍKLADY PŘEVODŮ



Obrázek 5.4: Výsledný BPMN model ocenění nemovitosti – 1. část podle Hellera

5.2. Příklad 1 – Ocenění nemovitosti podle Š. Hellera



Obrázek 5.5: Výsledný BPMN model ocenění nemovitosti – 2. část podle Hellera

5.3 Příklad 1 – Ocenění nemovitosti podle O.Mráze

V této podkapitole je předchozí příklad vymodelován podle metody O. Mráze. Vychází zejména z diagramů ATD 5.2 a PSD 5.3. Nejprve je uveden model standardního transakčního vzoru, který můžeme srovnávat s modelem Š. Hellera a následně model kompletního transakčního vzoru.

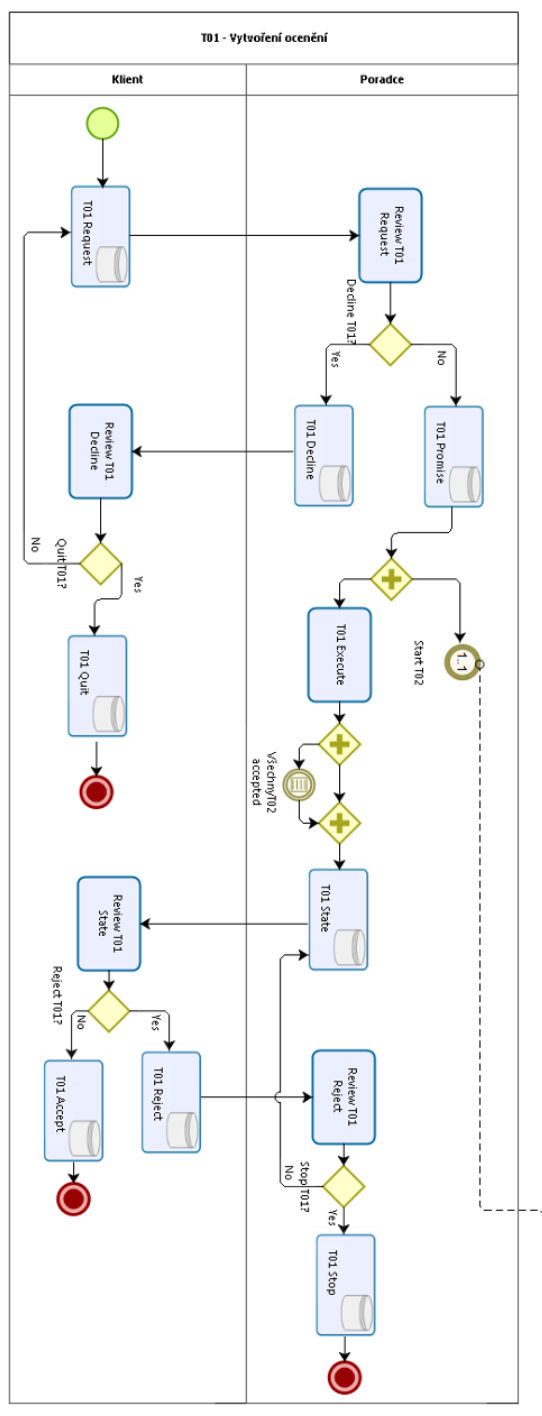
5.3.1 Standardní transakční vzor

Model obsahuje speciální symboly, které O. Mráz navrhl, aby nebyl výsledný model příliš rozsáhlý. Výsledek můžeme vidět na obrázcích 5.7, 5.8 a 5.9.

5.3.2 Kompletní transakční vzor

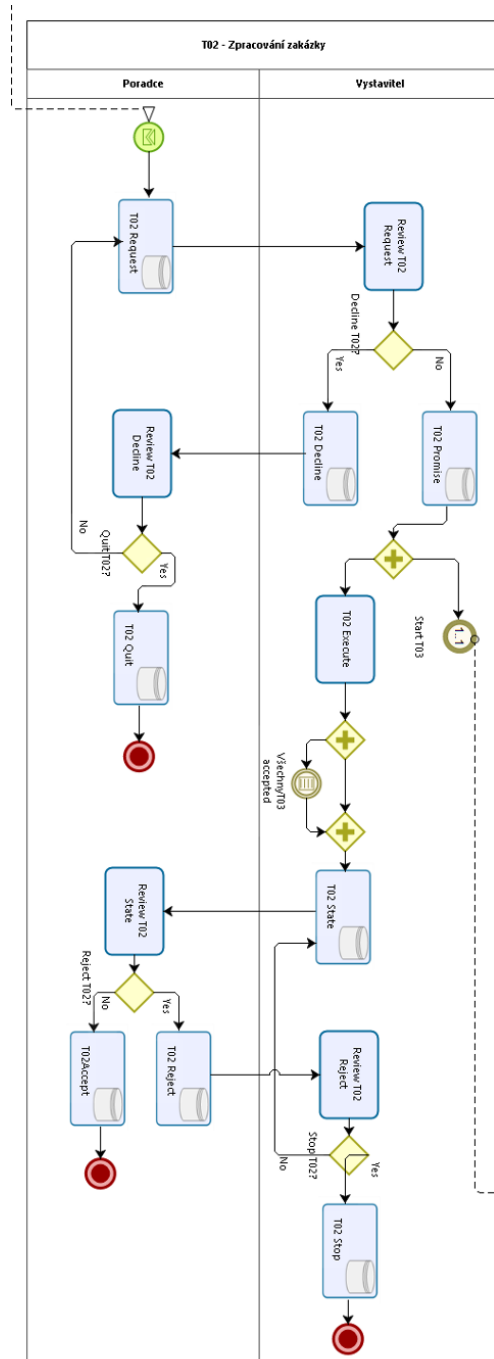
Posledními modely prvního příkladu jsou modely podle kompletního transakčního vzoru. Stejně jako u standardního jsou použity speciální symboly pro zkrácení výsledného modelu. První transakci zobrazuje obrázek 5.10, druhou 5.11 a třetí obrázek 5.12.

5. PŘÍKLADY PŘEVODŮ



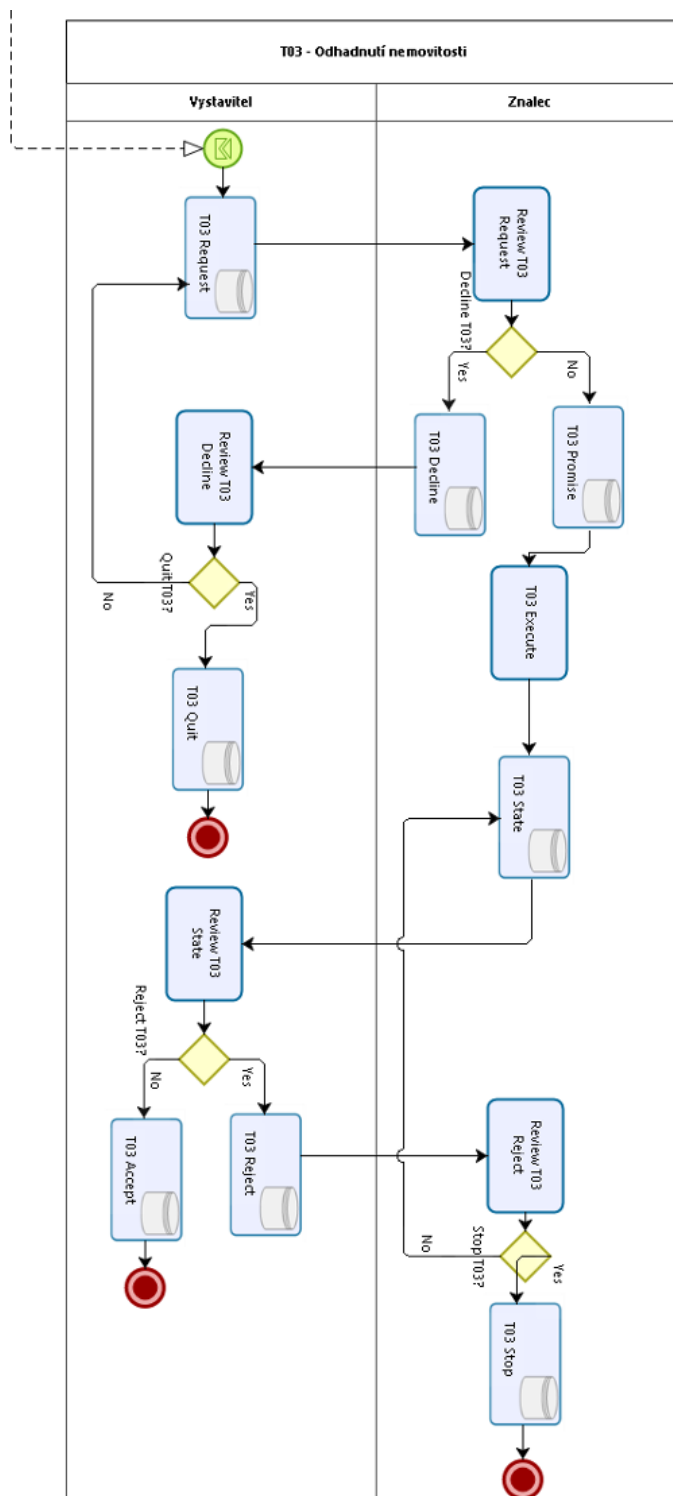
Obrázek 5.7: Výsledný BPMN model ocenění nemovitosti – 1. část podle Mráze

5.3. Příklad 1 – Ocenění nemovitosti podle O.Mráze



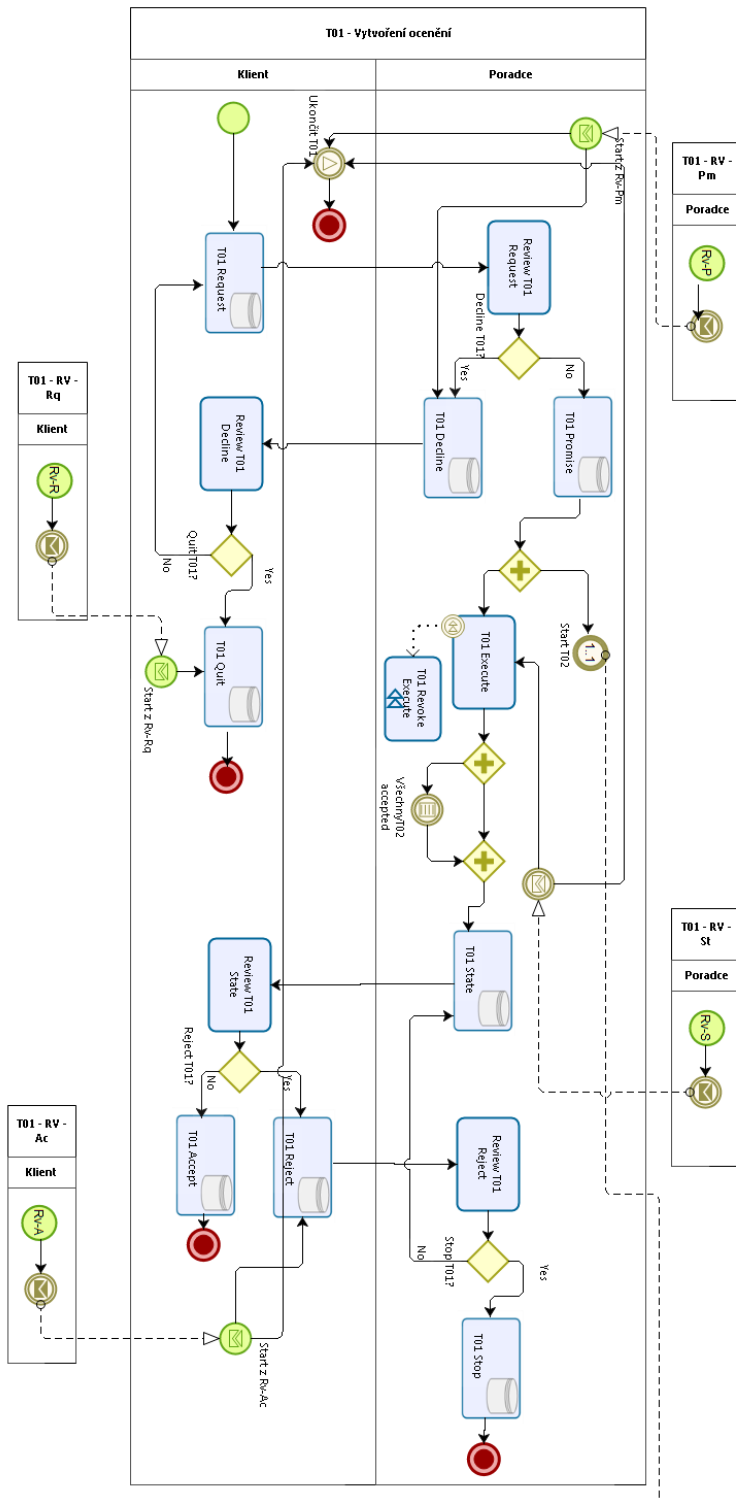
Obrázek 5.8: Výsledný BPMN model ocenění nemovitosti – 2. část podle Mráze

5. PŘÍKLADY PŘEVODŮ



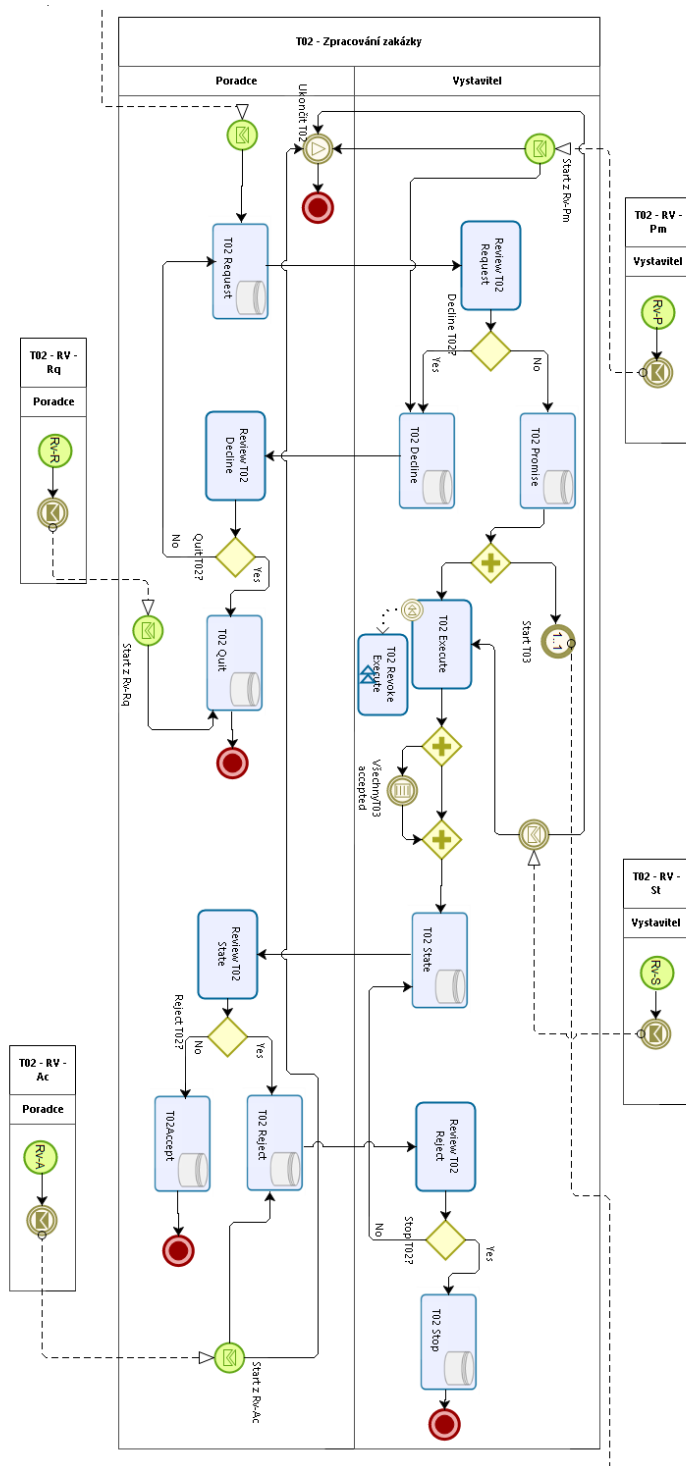
Obrázek 5.9: Výsledný BPMN model ocenění nemovitosti – 3. část podle Mráze

5.3. Příklad 1 – Ocenění nemovitosti podle O.Mráze



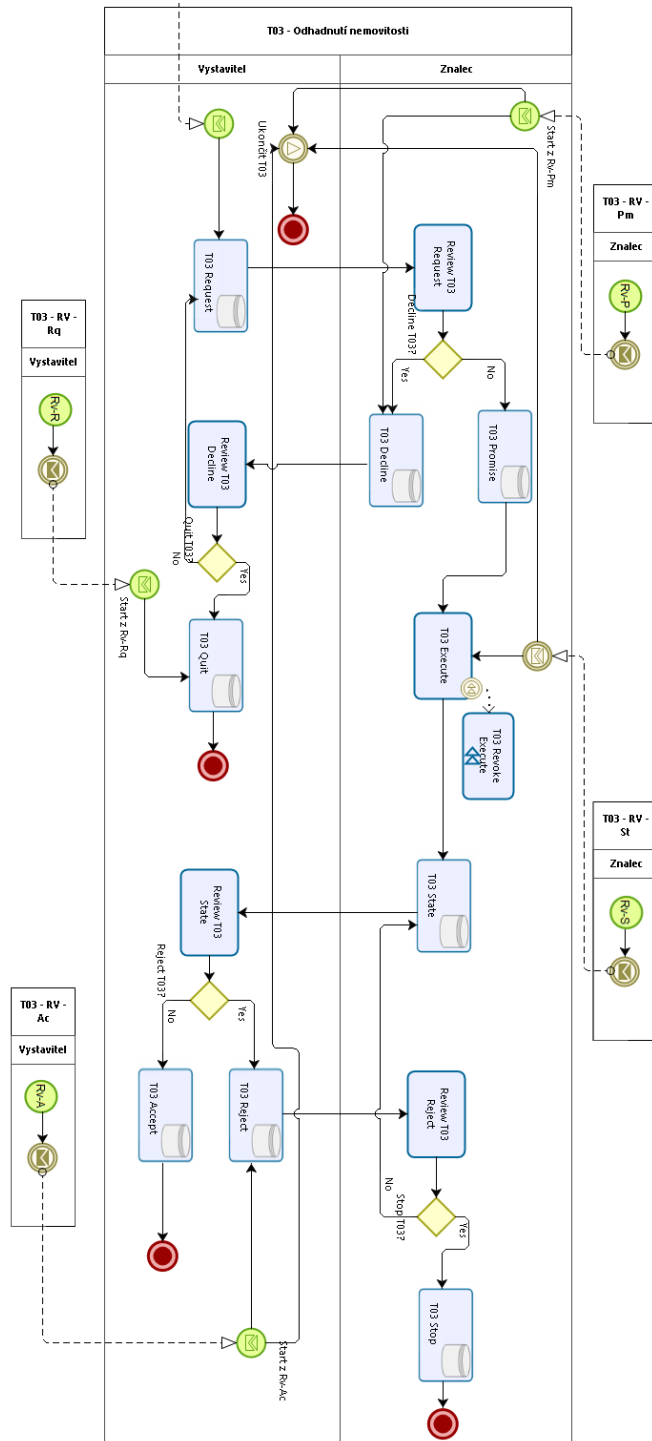
Obrázek 5.10: Kompletní transakční vzor ocenění nemovitosti – 1. část

5. PŘÍKLADY PŘEVODŮ



Obrázek 5.11: Kompletní transakční vzor ocenění nemovitosti – 2. část

5.3. Příklad 1 – Ocenění nemovitosti podle O.Mráze



Obrázek 5.12: Kompletní transakční vzor ocenění nemovitosti – 3. část

5.4 Příklad 2 – Proces nalezení a opravy chyby v aplikaci podle Š. Hellera

Druhý příklad znázorňuje životní cyklus chyby – defektu – nalezené v aplikaci. Tester, který chybu objeví, vypíše defekt a předá ho k validaci test manažerovi. Ten ho předává dál vývoji k opravě. Po opravení se defekt vrací zpět testerovi, který zařídí, aby byla oprava nasazena. V tomto příkladu už nejsou, kvůli rozsáhlosti, uvedeny všechny kroky postupu tvorby modelu. Oproti předchozímu příkladu je zde uveden i BPMN model takový, jaký by byl bez znalosti DEMO.

5.4.1 Textový popis procesu

Pokud tester objeví chybu v testované aplikaci, tak vytvoří v nástroji pro hlášení chyb nový defekt a předá ho test manažerovi. Test manažer zkontroluje, zda vypsáný defekt obsahuje všechny náležitosti a že se nejedná o duplicitu. Potom defekt přepne do stavu Open a pošle ho některému ze členů vývojového týmu. Vývojář defekt přijme tím, že ho označí statusem Solving. Následně chybu opraví a ve stavu Fixed předá zpět test manažerovi. Ten opravu zalogue do seznamu opravených chyb a pošle zpět testerovi, který chybu objevil. Tester zkontroluje, zda byla oprava úspěšná, a pokud ano, tak opravu defektu akceptuje. Pokud nebyla, tak se vrací zpět test manažerovi. Zároveň tester po opravení chyby napíše zprávu osobě odpovědné za nasazování oprav do produkce s žádostí o nasazení opravy. Tato osoba během nejbližšího nasazovacího okna opravu nasadí, a tím je celá oprava chyby dokončena.

5.4.2 BPMN model

Na obrázku 5.13 je BPMN model oproštěný od znalostí DEMO. Model je důležitý pro následné porovnání rozdílů mezi převedenými diagramy a tímto. Cílem bylo vymodelování jednoduchého průchodu procesem podle textového popisu z předchozí části.

5.4.3 Analýza popisu procesu

V textovém popisu procesu byly nalezeny následující tři transakce:

- Nahlášení chyby
- Vyvinutí opravy
- Nasazení opravy

Parametry transakcí byly následně zapsány do tabulky 5.2

Při aplikaci kompozičního axiomu bylo určeno pořadí provádění transakcí a jejich vzájemné závislosti. Výsledek je znázorněn na obrázku 5.14.

5.4. Příklad 2 – Proces nalezení a opravy chyby v aplikaci podle Š. Hellera

Tabulka 5.2: Transaction Result Table procesu opravy chyby

Transakce	T01	T02	T03
Název transakce	Nahlášení chyby	Vyvinutí opravy	Nasazení opravy
Výsledek transakce	Chyba opravena	Oprava vyvinuta	Oprava nasazena
Initiator	Tester	Test manažer	Tester
Executor	Test manažer	Vývojář	Správce nasazení
Request	Vypsán defekt	Odeslání defektu vývoji	Žádost o nasazení
Promise	Defekt přepnut do Open	Přepnutí do Solving	Žádost odsouhlasena
State	Opravený defekt vrácen	Přepnutí do Fixed	Oprava nasazena
Accept	Oprava akceptována	Zalogování opravy	Akceptace nasazení
Decline	Vrácení defektu	Odmítnutí defektu	Žádost zamítnuta
Reject	Oprava odmítnuta	Odmítnutí opravy	N/A

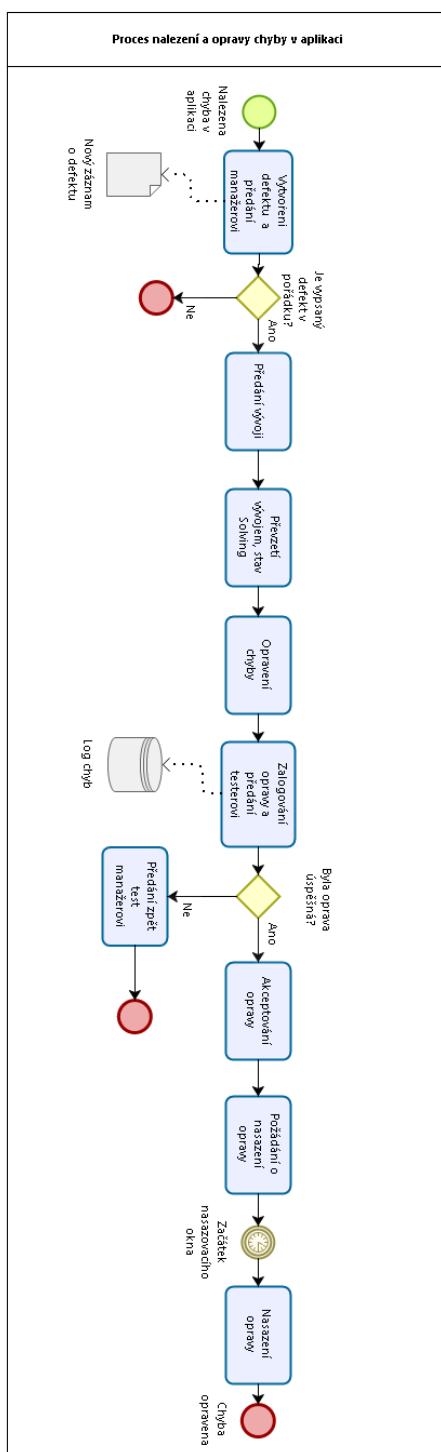
5.4.4 Vytvoření ATD a PSD diagramů

Dále jsou vytvořeny Actor Transaction Diagram a Process Structure Diagram. ATD je zobrazen na obrázku 5.15. Do procesu vstupují čtyři aktóři. Tester je označen jako CA-1, následně je zde test manažer A-1, vývojář A-2 a správce nasazení A-3. Přesnější pohled na provázanost jednotlivých transakcí potom poskytuje PSD na obrázku 5.16.

5.4.5 Vytvoření BPMN modelu

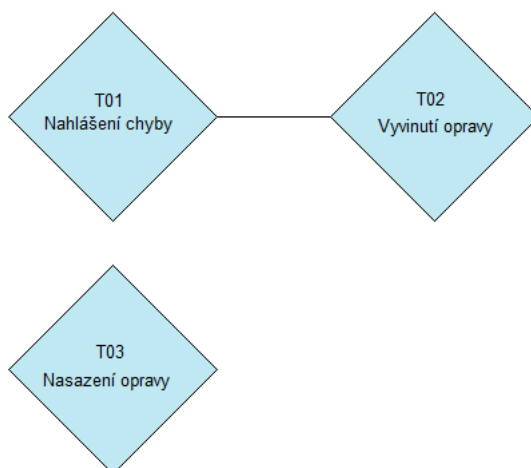
V posledním kroku je vytvořen samotný BPMN model. Na obrázcích 5.17 5.18 a 5.19 je vidět výsledný diagram standardního transakčního vzoru v BPMN. Pro větší přehlednost nejsou v modelu popsány jednotlivé události textově, ale zástupnými jmény, jako Request T01 apod.

5. PŘÍKLADY PŘEVODŮ

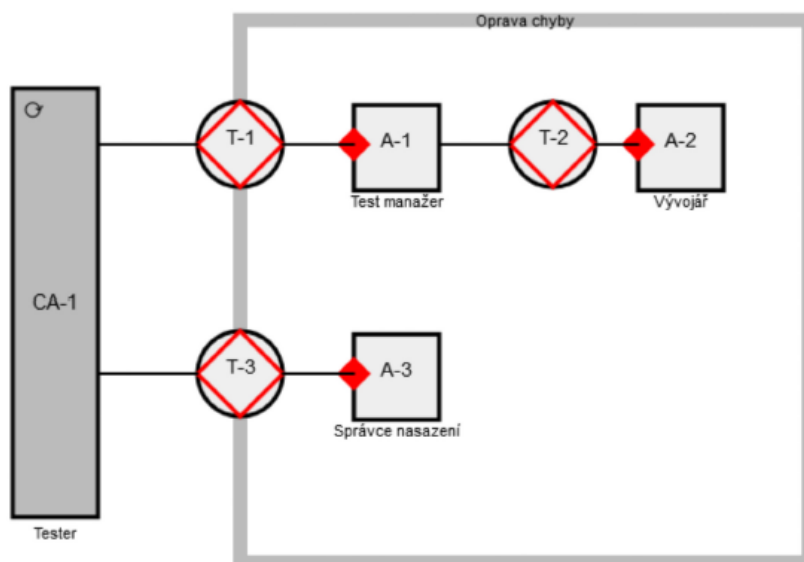


Obrázek 5.13: BPMN model procesu opravy bez použití DEMO

5.4. Příklad 2 – Proces nalezení a opravy chyby v aplikaci podle Š. Hellera

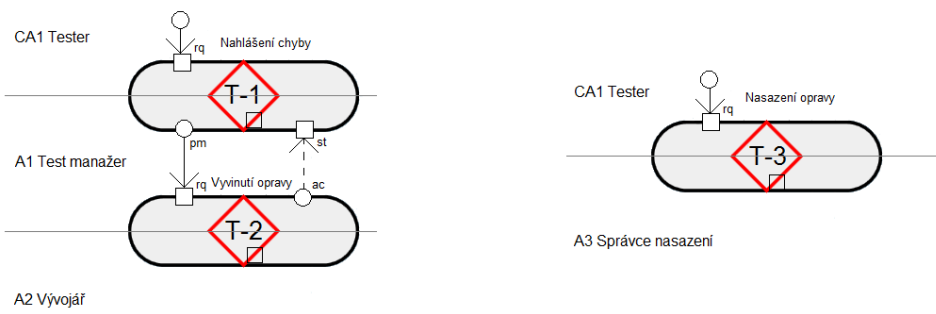


Obrázek 5.14: Pořadí a závislosti transakcí



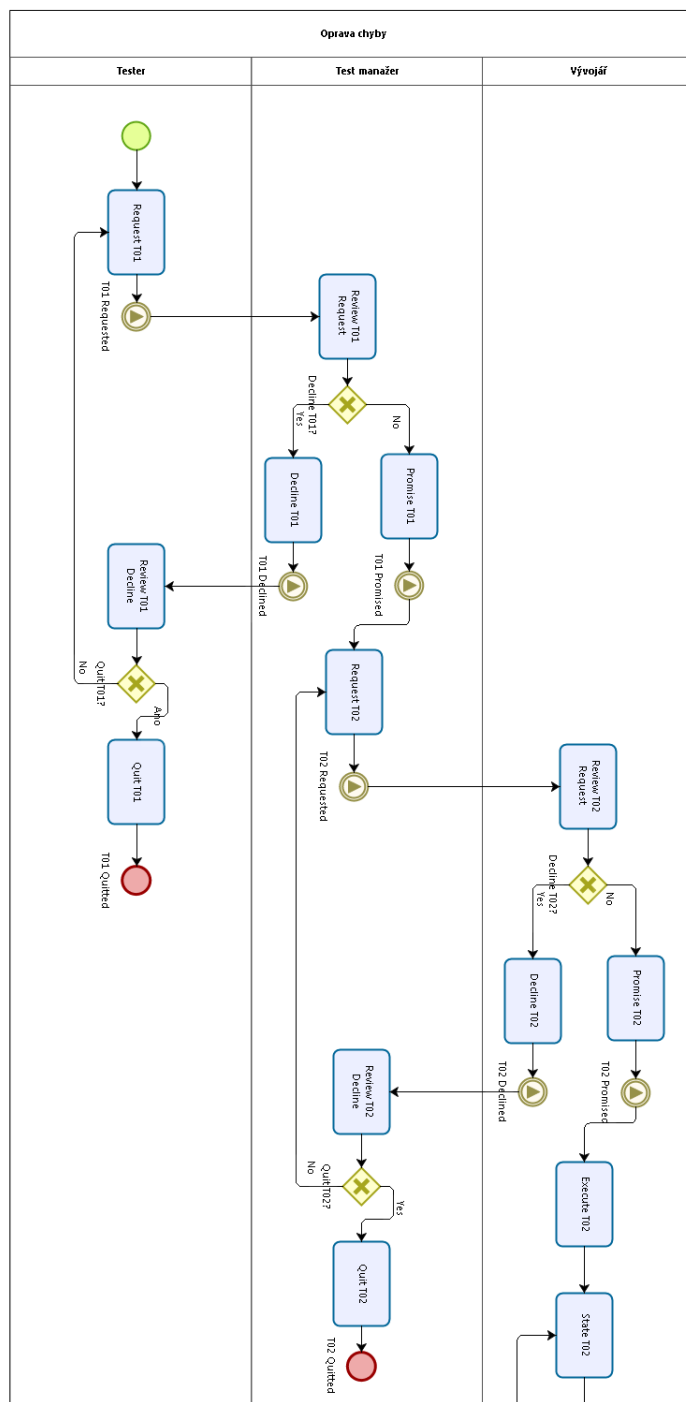
Obrázek 5.15: Actor Transaction Diagram

5. PŘÍKLADY PŘEVODŮ



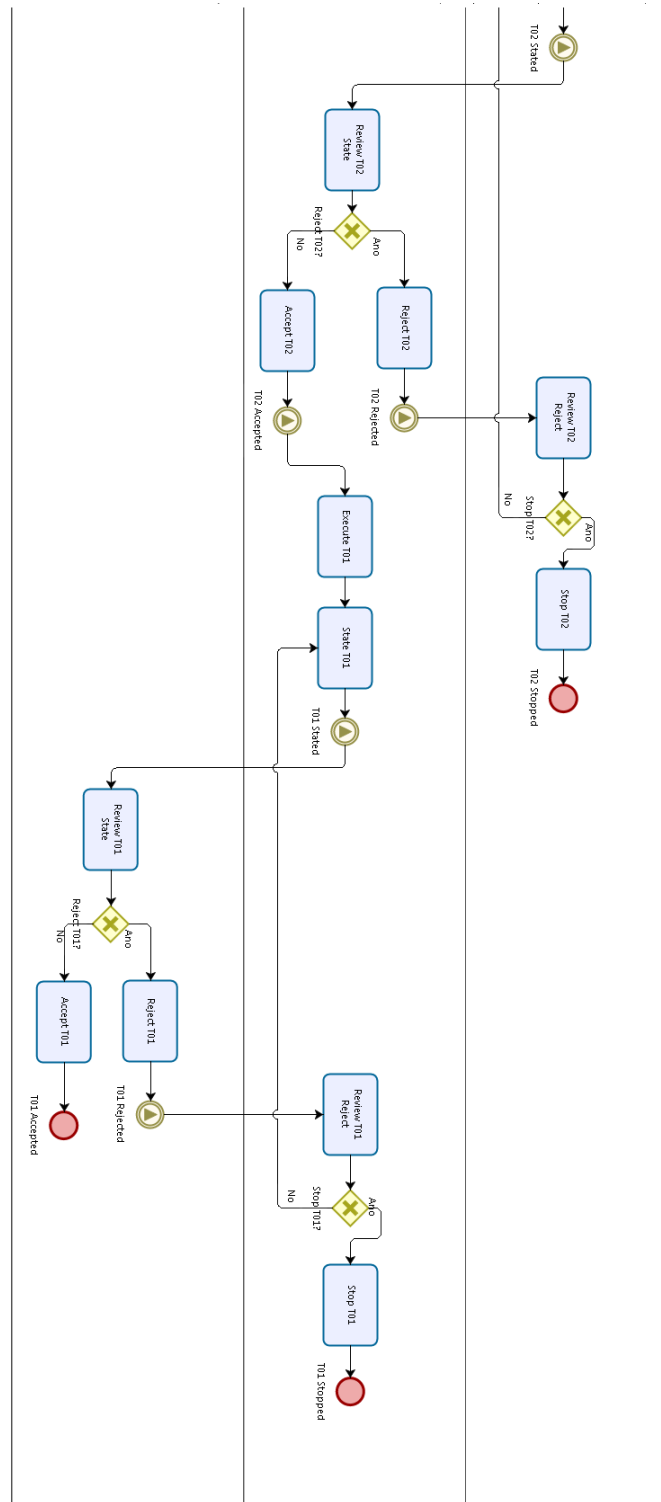
Obrázek 5.16: Process Structure Diagram

5.4. Příklad 2 – Proces nalezení a opravy chyby v aplikaci podle Š. Hellera



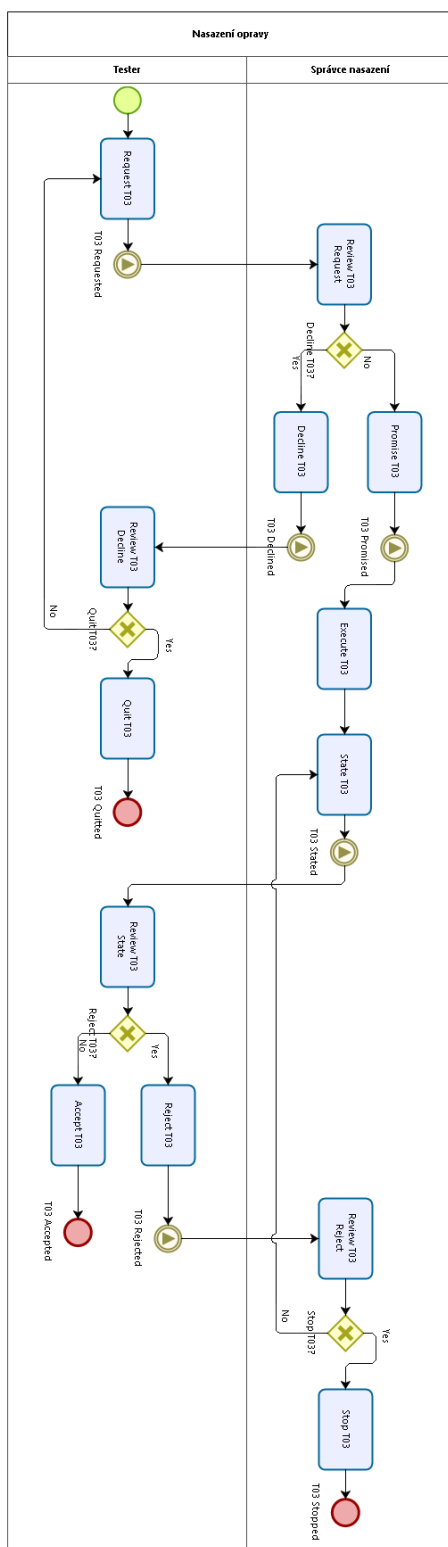
Obrázek 5.17: BPMN model opravy chyby – 1. část podle Hellera

5. PŘÍKLADY PŘEVODŮ



Obrázek 5.18: BPMN model opravy chyby – 2. část podle Hellera

5.4. Příklad 2 – Proces nalezení a opravy chyby v aplikaci podle Š. Hellera



Obrázek 5.19: BPMN model opravy chyby – 3. část podle Hellera

5.5 Příklad 2 – Proces nalezení a opravy chyby v aplikaci podle O. Mráze

Zde jsou výsledné modely procesu opravy chyby vymodelované podle metody O. Mráze. Opět je nejdříve uveden model standardního transakčního vzoru, následovaný modelem kompletního transakčního vzoru.

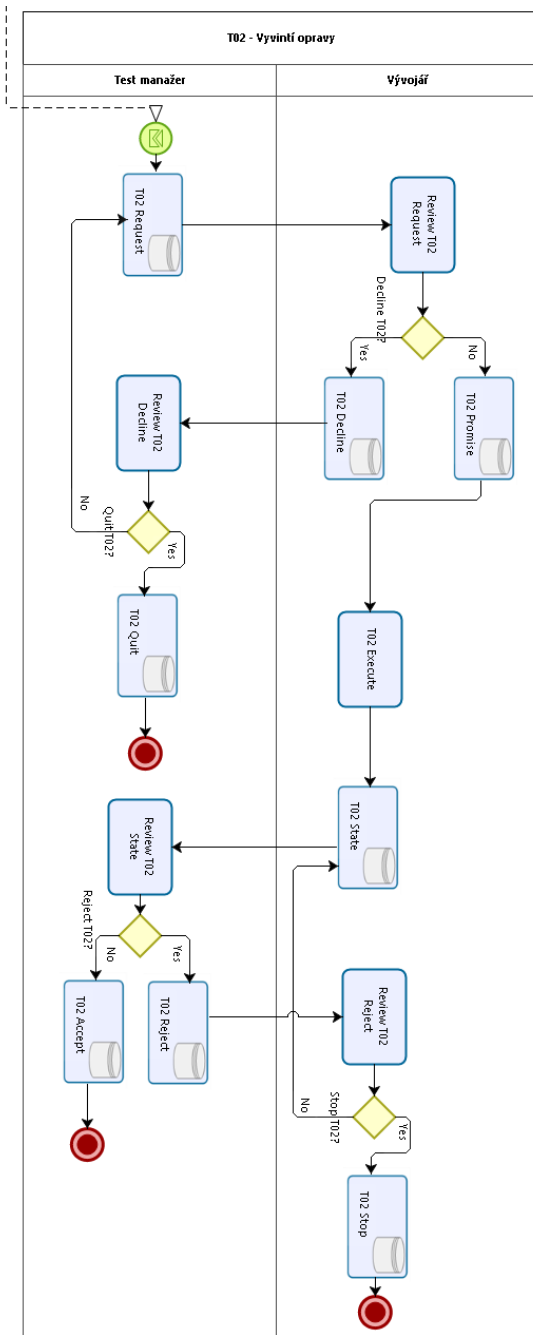
5.5.1 Standardní transakční vzor

Výsledek převodu můžeme vidět na obrázcích 5.20, 5.21 a 5.22.

5.5.2 Kompletní transakční vzor

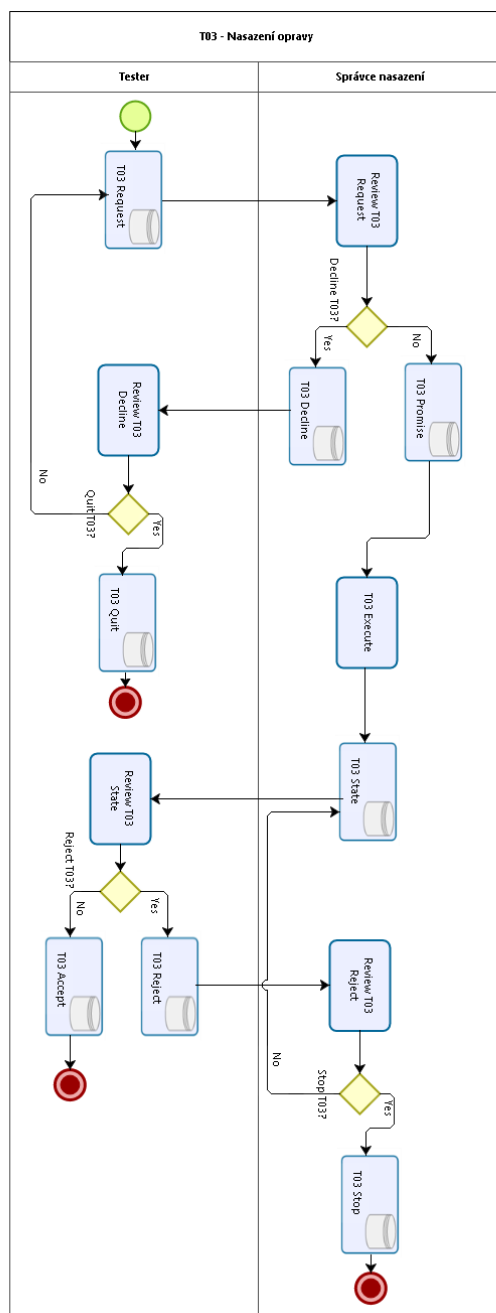
Jako poslední jsou uvedeny modely kompletního transakčního vzoru. První transakci zobrazuje obrázek 5.23, druhou 5.24 a třetí obrázek 5.25.

5. PŘÍKLADY PŘEVODŮ



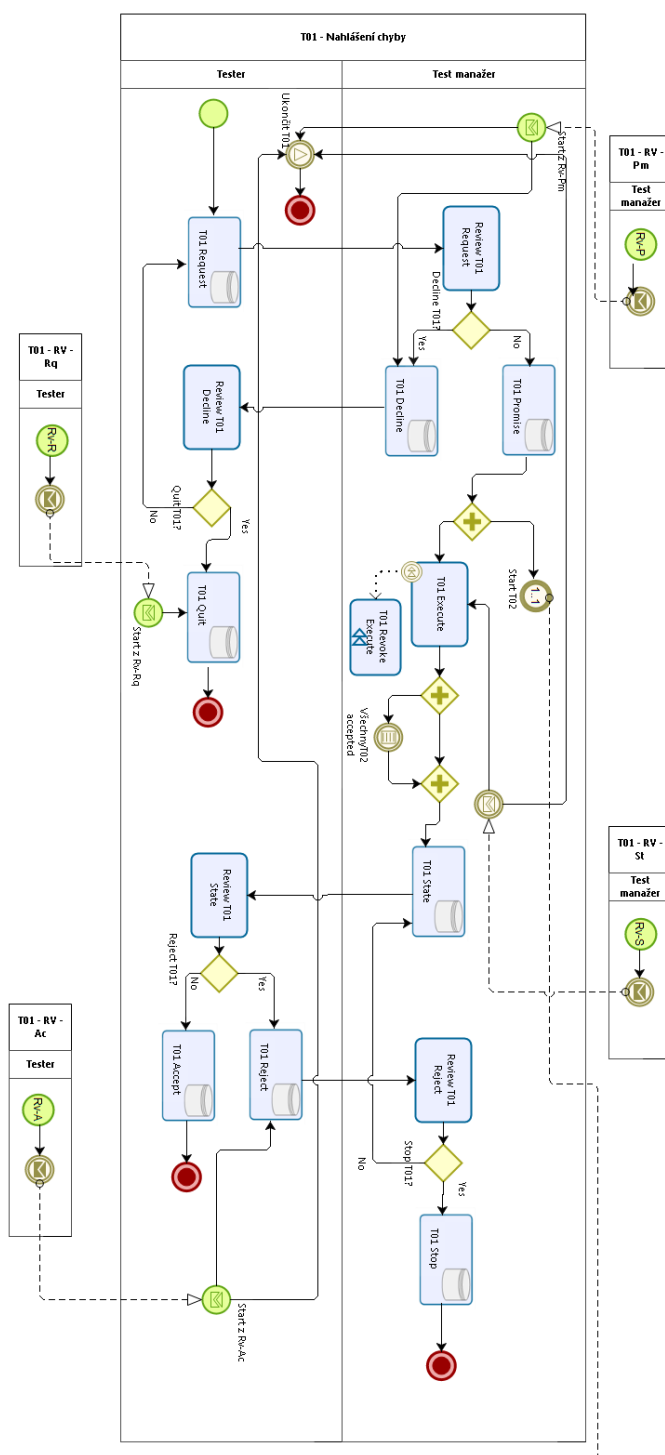
Obrázek 5.21: BPMN model opravy chyby – 2. část podle Mráze

5.5. Příklad 2 – Proces nalezení a opravy chyby v aplikaci podle O. Mráze



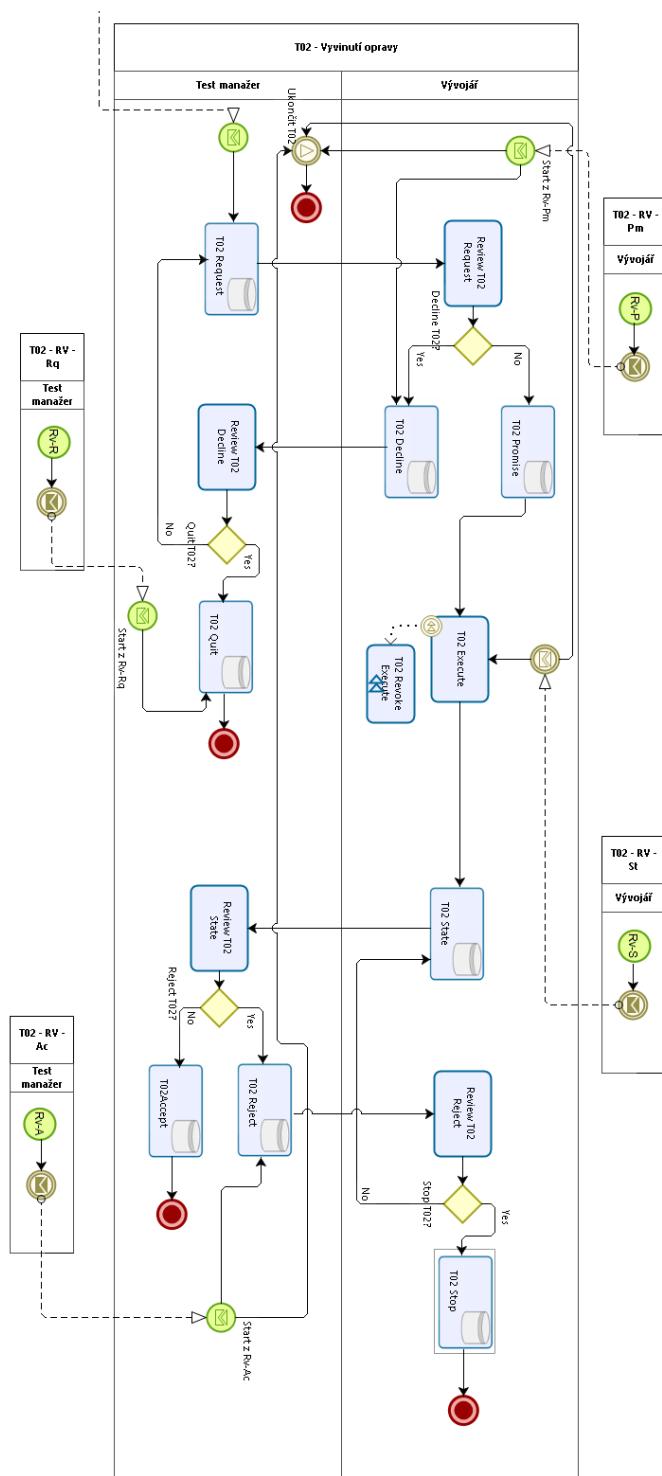
Obrázek 5.22: BPMN model opravy chyby – 3. část podle Mráze

5. PŘÍKLADY PŘEVODŮ



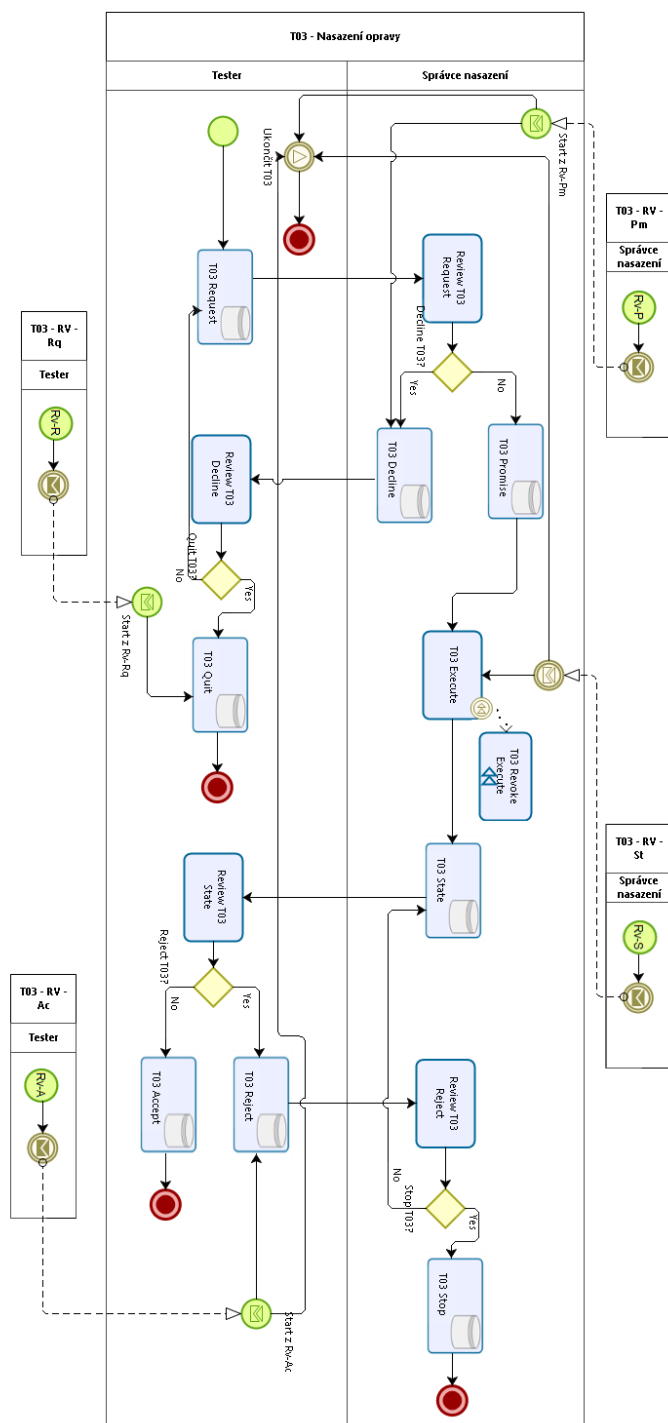
Obrázek 5.23: Kompletní transakční vzor opravy chyby – 1. část

5.5. Příklad 2 – Proces nalezení a opravy chyby v aplikaci podle O. Mráze



Obrázek 5.24: Komplettní transakční vzor opravy chyby – 2. část

5. PŘÍKLADY PŘEVODŮ



Obrázek 5.25: Kompletní transakční vzor opravy chyby – 3. část

Vyhodnocení metod převodů

V této kapitole je zhodnocena kvalita a použitelnost metod převodů z DEMO do BPMN, které byly předvedeny na příkladech v předchozí části práce.

6.1 Porovnání metod

Pro porovnání metod můžeme použít buď modely základního, nebo standardního transakčního vzoru. Zvolen byl standardní transakční vzor, protože je složitější a umožňuje tak přesnější porovnání. Pro srovnání tedy vycházíme z modelů uvedených v předchozí kapitole.

6.1.1 Porovnání velikosti modelu

Pro porovnání je použita metrika velikosti modelu popsána v [19]. Tato metrika porovnává celkový počet elementů, ze kterých se modely skládají. Počítají se všechny elementy – bazény, dráhy, aktivity, události, rozhodovací bloky, datové objekty, artefakty a propojovací objekty. Předpoklad je takový, že čím více elementů model obsahuje, tím je složitější. Počty elementů jsou v tabulce 6.1.

V případě metody O. Mráze se počítá s použitím speciálních symbolů pro zkrácení výsledného modelu. Vyjádření C-actu a C-factu se počítá jako jedna aktivita a spouštění child transakcí jako jedna událost. Modely příkladu ocenění nemovitosti jsou označeny jako P1, modely z příkladu testování softwaru potom jako P2.

Z tabulky vidíme, že velikost výsledných modelů je velmi podobná. Musíme ovšem počítat s tím, že v případě metody O. Mráze jsou použity elementy pro zkrácení výsledného modelu. Výsledek porovnání bez použití speciálních elementů pro zkrácení ukazuje tabulka 6.2. Zde je vidět, že model převedený podle metody O. Mráze je bez zkrácení větší o cca 25 % než model podle Hellera.

Tabulka 6.1: Základní porovnání velikosti modelů

Příklady	P1 – Heller	P1 – Mráz	P2 – Heller	P2 – Mráz
Počet bazénů	1	3	2	3
Počet drah	4	6	5	6
Počet aktivit	39	39	39	39
Počet událostí	25	16	26	14
Počet rozhodnutí	12	18	12	15
Počet datových objektů	0	0	0	0
Počet artefaktů	0	0	0	0
Počet propojovacích objektů	76	79	80	73
Celkem	157	161	164	150

Tabulka 6.2: Porovnání velikosti modelů bez speciálních symbolů

Příklady	P1 – Heller	P1 – Mráz	P2 – Heller	P2 – Mráz
Počet bazénů	1	3	2	3
Počet drah	4	6	5	6
Počet aktivit	39	41	39	40
Počet událostí	25	20	26	15
Počet rozhodnutí	12	20	12	16
Počet datových objektů	0	24	0	24
Počet artefaktů	0	0	0	0
Počet propojovacích objektů	76	109	80	100
Celkem	157	223	164	204

6.1.2 Splnění vlastností C4E

V kapitole 4.2 je popsáno pět kritérií, které by měly DEMO modely splňovat. Těmito kritérii jsou jednoznačnost, kompletnost, konzistence, stručnost a esencialita. Vzhledem k tomu, že cílem propojení DEMO a BPMN je vytváření modelů, které budou správné podle DEMO, je nutné ověřit, zda jsou tato kritéria u obou postupů splněna.

- **Jednoznačnost** – Při modelování stejného procesu dojdeme vždy ke stejnému výsledku. Díky dodržení transakčního axiomu budou modely stejného procesu vždy stejné.
- **Kompletnost** – Modely vytvořené podle obou metod obsahují všechny kroky transakčního axiomu a díky použití plaveckých drah je jasné, který aktor je za který krok zodpovědný.

6.2. Porovnání převedených diagramů s původním BPMN diagramem

Tabulka 6.3: Porovnání velikosti modelů vytvořených podle DEMO s BPMN modelem

Způsob modelování	P2 – BPMN	P2 – Heller	P2 – Mráz
Počet bazénů	1	2	3
Počet drah	1	5	6
Počet aktivit	9	39	39
Počet událostí	5	26	14
Počet rozhodnutí	2	12	15
Počet datových objektů	2	0	0
Počet artefaktů	0	0	0
Počet propojovacích objektů	17	80	73
Celkem	37	164	150

- Konzistence – U výsledných modelů je jasné, jaké je pořadí provádění jednotlivých kroků procesu, tyto kroky na sebe navazují podle transakčního axiomu.
- Stručnost – Žádný z modelů neobsahuje prvky, které by byly zbytečné nebo navíc.
- Esencialita – Ani jeden z výsledných modelů zjevně neobsahuje implementační detaily, popisují pouze jádro činnosti.

U obou postupů jsou kritéria splněna. Výsledné modely splňují pravidla, která jsou na ně kladena.

6.2 Porovnání převedených diagramů s původním BPMN diagramem

V této části je porovnán BPMN model procesu opravy chyby 5.13, vytvořený bez ohledu na metodiku DEMO, s modely vytvořenými převodem z DEMO do BPMN. Následující tabulka 6.3 pro porovnání uvádí velikost jednotlivých modelů. Model bez použití DEMO je označen jako P2 – BPMN, model vytvořený dle metody Š. Hellera jako P2 – Heller a model podle metody O. Mráze jako P2 – Mráz.

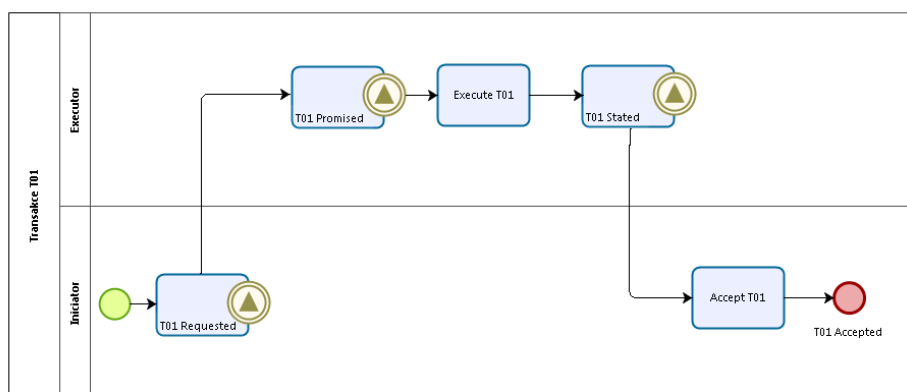
Z tabulky vidíme, že počet symbolů je při použití metod převodu přibližně čtyřnásobný. Tento rozdíl, i u jednoduchého příkladu poměrně markantní, může být zásadní pro použitelnost převodů. Na druhou stranu model bez použití DEMO obsahuje jen malou část z cest, kterými může proces projít.

6.3 Návrh pro vylepšení metody

Další částí práce je zamyšlení se nad možnostmi, jak by šlo metody dále vylepšit. Při pozorování výsledného diagramu převedeného podle metody Š. Hellera si lze všimnout, že se v něm několikrát opakuje symbol aktivity následovaný signálem – tedy vyjádření C-actu a C-factu. Nabízí se spojení těchto dvou symbolů dohromady, podobně jako spojil dohromady vyjádření C-actu a C-factu ve své práci O. Mráz. Je zde proto navržen nový symbol, který tyto dva spojuje dohromady. Nový element má vzhled úlohy se symbolem signálu v pravé části a můžeme ho vidět na obrázku 6.1. Vzor použití tohoto elementu na základním transakčním vzoru vidíme na obrázku 6.2.



Obrázek 6.1: Element kombinující úlohu a signál



Obrázek 6.2: Zkrácený základní transakční vzor

6.4 Použitelnost metod v praxi

V této části jsou shrnuty výhody, nevýhody, silné a slabé stránky obou metod převodu, s důrazem na jejich použitelnost v praxi.

6.4.1 Použitelnost metody Š. Hellera

V práci [11] je kromě mapování DEMO modelů do BPMN popsán i postup tvorby modelu už od zisku textového popisu procesu. Tento postup je sice aplikovatelný v obou metodách, ale popsán je pouze zde. U modelů malých procesů, o maximálně dvou vnořených transakcích, může být výhodou vyjádření všech transakcí v rámci jednoho bazénu. Výsledkem bude jeden diagram, ze kterého bude na první pohled zřejmé, jak jsou jednotlivé transakce propojeny.

Hlavním problémem, na který metoda naráží, je ignorování odvolávacích vzorů a tedy nemožnost vymodelování kompletního transakčního vzoru. Metodu lze použít jen v případě, že nám stačí základní nebo standardní transakční vzor. U modelů složitějších procesů, které by vyžadovaly několik úrovní zanoření transakcí, bude problémem vyjádření celého stromu transakcí v jednom bazénu. Výsledný model bude příliš velký například pro tisk, nepůjde ho rozumně dělit na menší části.

Metoda neřeší jak znázornit opakované volání transakce potomka. Stejně tak, pokud je provedení vnořené transakce volitelné, není popsán způsob zakreslení.

Posledním problémem je samotný způsob vyjádření vzniku C-factu pomocí signálu. V modelu se nenacházejí catching události typu signál, které by tyto signály zachytily. Druhý aktor se tedy o vzniku C-factu nedozví. Heller nicméně ve své práci předpokládá, že aktor signálu naslouchat bude. Je otázkou pro další diskuzi, zda je nutné zachycení signálu explicitně uvádět.

6.4.2 Použitelnost metody O. Mráze

Nejsilnější stránkou této metody je umožnění modelování kompletního transakčního vzoru. Výsledný model je ovšem poměrně složitý. Tento problém řeší nové elementy, které autor metody navrhl. Pro jejich snadné používání by bylo potřeba, aby byly zahrnuty do standardu BPMN. Bez zahrnutí do standardu je nutné jejich ruční přidání do modelovacího nástroje a nelze modely, ve kterých jsou použity, simulovat.

Metoda popisuje, jak zakreslit opakované i volitelné volání transakce potomka, a ani v jiných oblastech nebyly nalezeny nejasnosti. Díky tomu by mohla být metoda vhodná i pro případnou automatizaci převodu.

6.4.3 Závěrečné shrnutí

Cílem práce bylo ověření praktické použitelnosti obou metod převodu z DEMO do BPMN. Z příkladů použití vyplývá, že pomocí obou lze, s menšími či většími výhradami, dojít k BPMN modelu, který je vytvořený podle zásad DEMO. Můžeme tedy prohlásit obě metody za funkční.

Další otázkou je, která metoda je lepší pro praktické použití. Pokud chceme vytvořit model kompletního transakčního vzoru, je výběr metody jednoduchý – jiná možnost než metoda O. Mráze není. Při modelování ostatních transakčních vzorů si vybrat můžeme. Z porovnání velikosti výsledných diagramů (tabulka 6.1) plyne, že i když jsou vytvořeny různými metodami, jsou prakticky stejně velké (při použití symbolů pro zkrácení). Rozdíl je ve struktuře, kdy proces podle Hellera bude v jednom bazénu, zatímco podle Mráze má každá transakce svůj bazén. Záleží na konkrétním případě použití, která varianta bude preferována.

Více nevýhod a nejasností bylo nalezeno v Hellerově metodě. Tyto problémy se projevují zejména při modelování rozsáhlejších procesů.

Za výsledek porovnání tedy můžeme prohlásit, že pro modelování kompletního transakčního vzoru nebo pro složitější procesy je vhodnější volbou metoda O. Mráze. Při modelování základního nebo standardního transakčního vzoru jednoduchého procesu se rozdílily mezi metodami stírají, a tak volba závisí na osobních preferencích.

Z výsledku porovnání velikosti modelů vzniklých dle metod převodů a modelu vytvořeného jen dle BPMN 6.3 je zřejmé, že nárůst počtu prvků je velký. Ačkoli je původní modelovaný proces jednoduchý, výsledné diagramy dle metod převodu jsou velké a komplikované. Při jejich tvorbě jsem musela postupovat pomalu a často jsem se vracela k předcházejícím krokům z důvodu přehlednutí některých detailů. Způsob, jakým na sebe jednotlivé C-(f)acty a P-(f)acty navazují je sice na pohled logický a jednoduchý, ale při samotném modelování je velmi snadné některé prvky vynechat. Z těchto důvodů je ruční modelování kompletních modelů složitých procesů prakticky nemožné. Jako řešení tohoto problému se nabízí možnost automatizace převodu. V případě použití automatizovaného převodu by nebyl problém převést i složité procesy podle zásad DEMO do BPMN. Následně by se dalo využít toho, že existuje velké množství nástrojů umožňujících BPMN modely spustit a simulovat, pro důkladnou analýzu daných procesů.

Pro použitelnost v praxi je v neposlední řadě důležité množství času potřebné k osvojení si daných metod. DEMO stojí na silném teoretickém základu Enterprise Ontology, jejíž základy je nutné pochopit pro další práci. Pro modelování podle metod převodů, popsaných v této práci, stačí porozumění základů DEMO na úrovni kapitoly 3 – Metodika DEMO. Není nutné umět vytvářet všechny modely, stačí znalost ATD a PSD. Je ovšem nutné umět aplikovat všechny axiomy teorie PSI. Správně identifikovat ontologické transakce, tedy správně aplikovat distinkční axiom vyžaduje delší studium a vyzkoušení více

příkladů. Osobně jsem se s metodikou DEMO setkala až při tvorbě této práce a její pochopení mi trvalo řádově několik týdnů.

Notace BPMN je pro pochopení mnohem jednodušší, nestojí na teoretickém základu, který by bylo nutné nastudovat. Naučení se základů BPMN na úrovni této práce mi trvalo jednotky dnů.

Z tohoto srovnání vyplývá, že DEMO vyžaduje mnohem delší studium než BPMN, a to i jen pro aplikaci metod převodů. Na druhou stranu jsou tyto metody pochopitelné i pro osobu, která se s ničím podobným dříve nesetkala. Pro použití v praxi záleží na konkrétním případě použití, zda hlavní výhoda výsledného modelu – jeho kompletnost – převáží nevýhody, kterými jsou delší doba pro osvojení si potřebných teoretických i praktických dovedností a nečitelnost modelu pro neproškolené osoby.

Závěr

Cílem práce bylo seznámení se s metodologií DEMO a notací BPMN, seznámení se s existujícími metodami převodu z DEMO do BPMN, jejich ověření a vytvoření vzorových modelů podle těchto metod s následným zhodnocením jejich použitelnosti.

V práci byly popsány dvě metody převodu, které vznikly v rámci diplomových prací Možnosti využití metodiky DEMO pro tvorbu BPMN modelů Š. Hellera a Možnosti využití principů metodiky DEMO pro zvýšení kvality BPMN modelů O. Mráze. Obě metody byly ověřeny při vymodelování dvou příkladů procesů. U prvně jmenované metody šlo o vůbec první příklad použití k vymodelování standardního transakčního vzoru, protože práce, která metodu popisuje, obsahuje jen příklad základního transakčního vzoru. Podle obou metod byly vytvořeny modely standardních transakčních vzorů, podle metody O. Mráze byl vytvořen i model kompletního transakčního vzoru. Výsledné diagramy standardních transakčních vzorů byly porovnány na základě velikosti výsledných modelů.

Dále byly v práci zhodnoceny silné a slabé stránky obou metod a bylo upozorněno na některé problémy a nejasnosti, které se v průběhu modelování vyskytly. Jako možné vylepšení metody Š. Hellera byl navržen nový element pro zkrácení výsledného diagramu. Cíle práce se tedy podařilo splnit.

Do budoucna by bylo zajímavé prozkoumat možnosti automatizace převodu. Analýza metody O. Mráze ukazuje, že by pro ni mohla být vhodná. Vzhledem k nízkému počtu existujících metod převodu je zde prostor pro vznik dalších. Rozhodně jde o zajímavé téma s potenciálem zvýšení kvality vznikajících procesních modelů a celkově posunutí oblasti procesního řízení.

Literatura

- [1] Michael Hammer, J. C.: *Reengineering - radikální proměna firmy*. Praha: Management Press, 2000, ISBN 80-7261-028-7.
- [2] Šmída, F.: *Zavádění a rozvoj procesního řízení ve firmě*. Praha: Grada Publishing, 2007, ISBN 978-80-247-1679-4.
- [3] ARIS Community: Business process management (BPM) [online]. [cit. 2017-05-03]. Dostupné z: <http://www.ariscommunity.com/business-process-management>.
- [4] Recker J., Rosemann M., Indulska M., etc.: Business process modeling: a comparative analysis. *Journal of the Association for Information Systems*. 2009, volume 10, no.4.
- [5] EdrawSoft: Limitations of Flowcharts [online]. [cit. 2017-05-02]. Dostupné z: <https://www.edrawsoft.com/flowchart-limitations.php>.
- [6] Concept Draw: Basic Flowchart Symbols and Meaning [online]. [cit. 2017-05-02]. Dostupné z: <http://www.conceptdraw.com/How-To-Guide/flowchart-symbols>.
- [7] Smart Draw: Flowchart [online]. [cit. 2017-05-02]. Dostupné z: <https://www.smartdraw.com/flowchart/>.
- [8] OMG: Introduction to OMG's unified modeling language [online]. [cit. 2017-05-02]. Dostupné z: <http://www.uml.org/what-is-uml.htm>.
- [9] Uml-diagrams: Process Shopping Order. *UML Activity Diagram Example* [online]. [cit. 2017-05-02]. Dostupné z: <http://www.uml-diagrams.org/shopping-process-order-uml-activity-diagram-example.html>.
- [10] OMG: Business Process Model and Notation Version 2.0.2 [online]. [cit. 2017-04-21]. Dostupné z: <http://www.omg.org/spec/BPMN/2.0.2/PDF/>.

- [11] Štěpán Heller: *Možnosti využití metodiky DEMO pro tvorbu BPMN modelů*. Diplomová práce, České vysoké učení technické v Praze, Fakulta informačních technologií, 2016.
- [12] Sparx Systems: Enterprise Architect user guide. *BPMN 2.0 Collaboration Toolbox Page* [online]. [cit. 2017-04-20]. Dostupné z: http://sparxsystems.com/enterprise_architect_user_guide/13.0/model_domains/bpmn_2_0_collaboration_toolbox.html/.
- [13] Vejražková, Z.: *Business Process Modeling and Simulation: DEMO, BORM and BPMN*. Diplomová práce, České vysoké učení technické v Praze, Fakulta informačních technologií, 2013.
- [14] van der Horst, P.: *From business transactions to business process work flows*. Diplomová práce, Delft University of Technology, 2010.
- [15] Dietz, J. L. G.: *Enterprise Ontology: Theory and methodology*. New York: Springer, 2006, ISBN 978-3-540-29169-5.
- [16] Dietz, J. L. G.: Demo Bachelor course. [cit. 2017-05-01]. Dostupné z: https://edux.fit.cvut.cz/courses/MI-MEP/_media/lectures/prednasky.pdf.
- [17] Mráz, O.: *Možnosti využití principů metodiky DEMO pro zvýšení kvality BPMN modelů*. Diplomová práce, České vysoké učení technické v Praze, Fakulta informačních technologií, 2016.
- [18] Van Nuffel D., Mulder H., Van Kervel S.: Enhancing the Formal Foundations of BPMN by Enterprise Ontology. doi:10.1007/978-3-642-01915-9_9, [cit. 2017-04-21]. Dostupné z: http://link.springer.com/10.1007/978-3-642-01915-9_9.
- [19] Hronza R., Pavlíček J., Náplava P.: Míry kvality procesních modelů vytvořených v notaci BPMN. *Acta Informatica Pragensia*. 2015, doi:10.18267/j.aip.66, dostupné z: <https://aip.vse.cz/index.php/aip/article/view/113>.

Seznam použitých zkratk

ATD Actor Transaction Diagram

BPMN Business Process Model and Notation

DEMO Design and Engineering Methodology for Organizations

OMG Object Management Group

PSD Process Structure Diagram

PSI Performance in Social Interaction

UML Unified Modeling Language

Obsah přiloženého CD

	readme.txt.....	stručný popis obsahu CD
	src	
	thesis	zdrojová forma práce ve formátu L ^A T _E X
	models	modely vytvořené v rámci práce
	text	text práce
	thesis.pdf	text práce ve formátu PDF