

Kapitola 5: Využití metodologie DEMO pro vytváření BPMN modelů

Bc. Štěpán Heller

10. ledna 2016

1 Motivace

V předchozích kapitolách jsme se zabývali notací BPMN a metodologií BPMN odděleně a popsali jsme jejich silné i slabší stránky. Motivací pro napsání této práce však je silné stránky obou přístupů zkombinovat a na takovém základě vytvořit metodu, které umožní vytvářet BPMN modely podnikových procesů, které budou *kompletní* a *konzistentní*. Že je spojení těchto přístupů vhodné potvrzuje i [32], když píše, že BPMN začíná tam, kde DEMO končí.

1.1 Slabé stránky DEMO

Jak již bylo v průběhu této práce několikrát zmíněno, na metodologii DEMO je nejcennější to, že říká tvůrcům procesních modelů „jak“ by měli při vytváření modelů podnikových procesů postupovat a nedává jim tedy pouze notaci, pomocí které je možné modely „kreslit“. Nutno však zároveň říct, že teoretický základ metodologie DEMO (tedy kombinace Enterprise ontology a Ψ -theory) je pro běžného uživatele velmi nepřístupný a obtížný na pochopení, protože jeho základy sahají až k fundamentálním filozofickým konceptům, jakými právě ontologie jsou.

Křivka učení je tedy v tomto případě velmi pozvolná a k plnému pochopení metodologie DEMO je potřeba hodně času a učení metodou pokus-omyl. Autor této práce má zkušenosti z účasti na kurzu MI-MEP (Modelování ekonomických procesů) na Fakultě informačních technologií ČVUT, kde po dobu jednoho semestru bylo certifikovanými DEMO profesionály vyučováno DEMO po teoretické i praktické stránce. Z autorových zkušeností i po absolvování celého tříměsíčního kurzu měli jeho účastníci problém se základními technikami metodologie DEMO, jakou je například korektní aplikace distinkčního axiomu Performa-Infoma-Forma analýzou, ale i s dalšími fundamentálními věcmi.

Nebudeme tedy daleko od pravdy, když konstatujeme, že v případě běžných uživatelů (návrhářů, analytiků, vývojářů) v rámci podniků bude problém podobný. Dalším nedostatkem DEMO je malé množství nástrojů, které DEMO podporují a žádný z existujících dosud nepokrývá všechny aspect modely,

kteřé DEMO obsahuje. Nutno však říct, že komunita kolem DEMO je velmi aktivní a na vývoji těchto nástrojů se intenzivně pracuje.

Jedním z cílů této práce je umožnit uživatelům nadále využívat nástroje, které znají (BPMN) a umí používat, ale v rámci metodologie DEMO identifikovat „nezbytné minimum“ z teoretického základu, které zvýší kvalitu výsledných procesních BPMN modelů a tím umožní i jejich lepší analýzu a optimalizaci.

1.2 Slabé stránky BPMN

U BPMN nacházíme slabiny tam, kde jsme u DEMO indentifikovali silné stránky a naopak. Jak již bylo naznačeno v kapitole 2, velkým problémem BPMN je absence pevného teoretického základu, který dává návrhářům pevný rámec, podle kterého se mohou řídit a vytvářet modely, podle pevně daných pravidel a postupů. Absence takového pevného rámce pak ústí v to, že modely jsou neúplné a chybí v nich podstatné informace nebo v to, že v momentě, kdy různí uživatelé modelují jeden a ten samý proces, tak skončí s různými modely i když jsou všechny dle specifikace BPMN korektní.

Problematická je i absence seznamu pravidel, co který element v BPMN přesně vyjadřuje, kdy ho používat a kdy ne. Popis jednotlivých elementů je rozprostřen na 500 stranách v celé specifikaci BPMN [29] a uživatelé jsou tak odkázáni na interpretaci této specifikace, která pochází většinou od tvůrců modelovacího nástroje, který tito uživatelé aktuálně používají. To pak dále tříští způsob, jakým jsou elementy v praxi používány.

V praxi v organizacích často vidíme, že mezi procesními analytiky a vývojáři přirozeně vzniká jakási pseudo-metodologie, právě z důvodu toho, aby bylo možné uvnitř organizace konzistentně vytvářet modely podnikových procesů stejným způsobem i v týmu více lidí a aby byla zajištěna kontinuita i v případě fluktuace členů tohoto týmu. Jedním z přínosů této práce, by mělo být vytvoření pevného rámce založeného na Enterprise ontology a Ψ -theory, který by mohl nahradit tyto pseudo-metodologie.

Zajímavá zjištění uvádí [9], kde autoři při svém výzkumu aplikovali zásady, na kterých stojí metodologie DEMO, na dva klíčové procesy ve velké organizaci (více než 2000 zaměstnanců). Tyto procesy obsahovaly dohromady přes 500 aktivit a více než 60 aktorů. Analytici prověřovali, zda je procesní model *konzistentní*, tedy zda pořadí kroků v tomto procesu odpovídá transakčnímu axiomu. Dále se zajímali, zda je model *kompletní*, neboli jestli všechny kroky transakce dle transakčního axiomu DEMO se dají namapovat na aktivity uvedené v BPMN modelu. Po důkladné analýze těchto procesů došli k následujícím zjištěním:

- V původních procesech chybělo 25 % P-factů neboli tyto procesy obsahovaly aktivity, které nebylo možné vztáhnout k vytváření nějakého výsledku (hmotného či nehmotného).
- Chybělo rovněž 25 % *request* C-actů. Tedy výsledky byly produkovány bez explicitního požadavku na jejich vytvoření a nebylo tak možné jasně identifikovat iniciátora celé transakce.

- Chybělo 50 % *promise* C-actů. Requesty byly implicitně potvrzovány bez jakékoliv jasné dohody mezi actory.
- Chybělo 25 % *state* C-actů. Výsledky transakce tak nebyly jasně komunikovány s jejím iniciátorem a tedy není jasné, zda dohlédnutí na výsledek transakce leží na iniciátorovi nebo exekutorovi.
- Chybělo 40 % *accept* C-actů. Nebylo tedy možné identifikovat, jak jsou výsledky transakce akceptovány jejím iniciátorem a jestli takové výsledky splňují požadavky, které na ně byly kladeny při requestu.

Je tedy zřejmé, že absence pevného teoretického je reálným problémem (ač ho uživatelé sami nepocítují [32]) a jeho vytvoření by zvýšilo kvalitu vytvářených modelů podnikových procesů.

1.3 Benefity kombinace DEMO a BPMN

Hlavním přínosem této práce je vytvoření základů metody, která vychází z Enterprise ontology, Ψ -theory a metodologie DEMO. Tato metoda v budoucnu umožní vytváření kvalitnějších BPMN modelů. Slovo „kvalitnější“ si na tomto místě zaslouží hlubší rozebrání.

Cílem této práce není vytvoření metodologie, která bude dokonale kopírovat DEMO a v podstatě jen umožní vytvoření DEMO modelů za použití notace BPMN (abstrahujeme zde od toho, zda je to vůbec možné, to musí být případně potvrzeno dalším výzkumem). Pokud bychom se k takovému přístupu rozhodli, nedosáhli bychom vyřešení problémů uvedených v sekci 1. Cílem této práce je nalézt takový *teoretický rámec*, který bude stále pochopitelný pro uživatele a umožní jim vytvářet kvalitnější BPMN modely, ale nebude mít tak dlouhou křivku učení, jakou pozorujeme u DEMO. Co je zde tedy míněno „kvalitnějšími“ BPMN modely?

- Modely takto vytvořené budou *kompletní*, tedy nebude chybět žádný krok dle transakčního axiomu a bude jasně specifikováno, který actor je zodpovědný za provedení konkrétního transakčního kroku.
- Modely takto vytvořené budou *konzistentní*, neboli pořadí provádění transakčních kroků bude odpovídat transakčnímu axiomu.
- Modely takto vytvořené budou *jednoznačné*, neboli při modelování toho samého procesu by vždy měl vzniknout stejný výsledný model.
- Modely budou oproštěné od implementačních detailů a budou zachycovat jen skutečné jádro činnosti organizace.

2 Existující možnosti společného využití DEMO a BPMN

Vhodnost spojení obou přístupů neušla pozornosti jiných autorů, například [32], [10] nebo [9]. Ve stručnosti budou na tomto místě tyto přístupy popsány, protože i na jejich zjištění autor této práce svůj návrh metody staví.

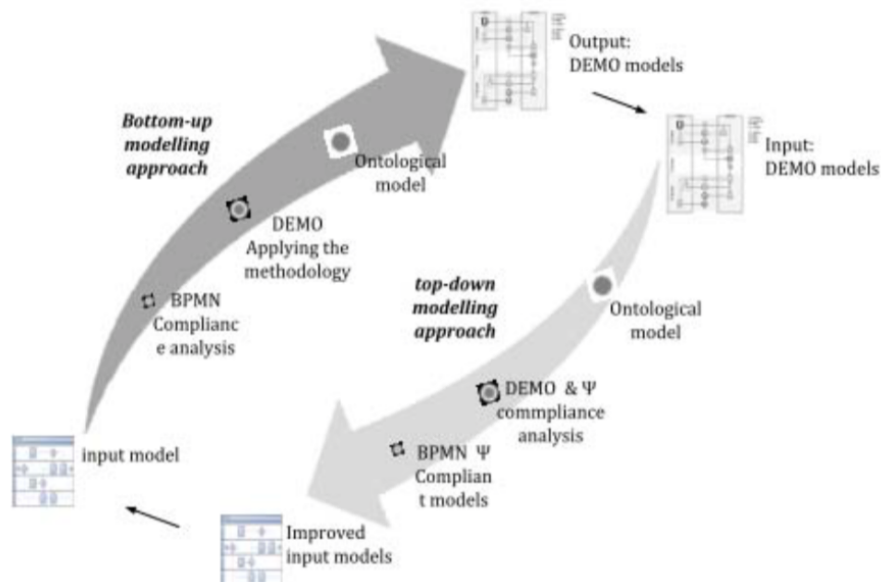
2.1 Analýza procesních modelů pomocí Ψ -theory

V [9] a [10] se autor zabývá využitím Ψ -theory pro analýzu modelů podnikových procesů s cílem zvýšit jejich *kompletnost* a *konzistentnost*. Za tímto účelem představuje autor metodu, jejímž vstupem je procesní model v notaci BPMN. Metoda se skládá z následujících čtyř kroků:

1. Provedení analýzy vstupního BPMN modelu s cílem klasifikovat jednotlivé elementy jako C-act nebo P-act a také rozlišit, zda jsou ontologické, info-logické nebo datalogické. Dále je nutné se pokusit v modelu identifikovat actory odpovědné za vykonání C-actů a P-actů.

Ontologické C-acty jsou následně klasifikovány dle jednotlivých transakčních kroků dle *standardního transakčního vzoru*.
2. V dalším kroku je vygenerován DEMO Process Structure Diagram (PSD) na základě seznamu actorů, transakcí, C-actů a P-actů získaných v předchozím kroku.
3. Ve třetím kroku je PSD diagram z předchozího kroku doplněn o C-acty a P-acty, které v něm chybí oproti transakčnímu vzoru. Dále je vytvořen DEMO Actor Transaction Diagram (ATD) pro zobrazení návaznosti transakcí dle kompozičního axiomu.
4. V následujícím kroku je provedena revize vstupního BPMN modelu a ten je na odpovídajících místech doplněn o elementy představující chybějící C-acty nebo P-acty identifikované v předchozím kroku. Takto revidovaný vstupní model je nyní *kompletní* a *konzistentní* dle Ψ -Theory.

Opakovanou aplikací této metody za účasti doménových i procesních vlastníků je možné dosáhnout procesních modelů, které uspokojí business uživatele a zároveň budou *kompletní* a *konzistentní* dle Ψ -Theory. Iterativní způsob aplikace představené metody ilustruje obrázek 1.



Obrázek 1: Iterativní aplikace metody dle [10]

2.2 Rozšíření formálních základů BPMN pomocí Enterprise Ontology

Práce [32] nazvaná *Enhancing the Formal Foundations of BPMN by Enterprise Ontology* se na teoretické úrovni zabývá možností, jak využít Enterprise ontology jako ontologický základ pro vytváření BPMN modelů a argumentuje, že fundamentální koncepty BPMN jsou „podobné“ konceptům DEMO:

- Koncept *aktivita* je „podobný“ konceptu výkonu (performance) nebo DEMO P-factu nebo určité komunikaci o tomto výkonu neboli DEMO C-factu.
- Koncept *plavecké dráhy* je „podobný“ konceptu aktora (Actor-Iniciátor nebo Actor-Executor).

Na tomto základě a následnou důkladnou analýzou axiomů Ψ -theory autoři došli k 14 pozorováním. Jelikož jsou tato pozorování jedním ze základních stavebních kamenů metody navrhované v této práci budou zde uvedeny:

- Existují identifikovatelní *actoři* plnící určité role, kde se actor vztahuje k roli a nikoli nutně k jedinci, který tuto roli fyzicky vykonává.
- Existují identifikovatelné P-facty reprezentující nějaký *úkon*.
- Existují C-facty reprezentující *komunikaci* o konkrétním úkonu, který má být vykonán.

- Každý P-fact je vztažen pouze k jednomu unikátnímu C-factu a naopak.
- Každý actor vyjma kořenového Actora-Iniciátora má jeden a pouze jeden vztah Actora-Exekutora k jednomu a pouze jednomu P-factu, ale může mít $0...n$ Actor-Iniciátor vztahů k jiným P-factům (potomkům).
- Každý P-fact může být definován jako množina $0...n$ jeho potomků.
- U každého rodičovského P-factu musí nejdříve dojít k dokončení (*state* a *accept*) P-factů, které jsou jeho potomky, dříve než může začít exekuce rodičovského P-factu.
- Actor, který je exekutorem nějakého P-factu je zároveň iniciátorem každého P-factu, který je jeho potomkem.
- Pouze iniciátor kořenového P-factu není iniciátorem žádného jiného P-factu.
- U každého modelu existuje alespoň jeden P-fact, který je ve vztahu k actorovi, který je jen a pouze exekutorem tohoto P-factu.
- Existuje množina osmi atributů (*request*, *promise*, etc.), které se unikátně vztahují ke konkrétnímu C-factu, který popisuje aktuální stav komunikace ohledně úkonu (vytvoření P-factu).

Autoři ve své práci představují nástin dvou přístupů, jak využít DEMO a Enterprise Ontology pro vytváření či vylepšení BPMN modelů. Jedním z přístupů je analýza a doplnění existujících BPMN modelů na základě čtyř základních axiomů Ψ -theory v základu podobně jak bylo popsáno v sekci 2.1.

Pro tuto práci přínosnější je nástin druhé možnosti – vytváření nových BPMN modelů na formálním základě, který z DEMO a Enterprise Ontology vychází. Vytváření takového BPMN modelu začíná sběrem informací o organizaci, jejím chodu a funkci. Dalším krokem je aplikace *distinkčního axiomu* a vyřazení všech informací, které nejsou ontologické. Třetím krokem je pak ve zbylých informacích nalézt ontologické transakce a actory, kteří vystupují v rolích Actor-Iniciátor a Actor-Exekutor ve vztahu k těmto transakcím. Závěrečným krokem je pak vytvoření DEMO Actor-Transaction Diagramu, který zobrazuje vztah mezi transakcemi a actory.

Výsledkem tohoto postupu tedy není BPMN model, ale kroky v něm nastíněné je nezbytné provést, aby bylo možné vytvořit BPMN model, který je *konzistentní* a *kompletní*. Metoda nastíněná v této práci právě na těchto krocích staví a posouvá je směrem ke skutečnému BPMN modelu.

3 Mapování základních DEMO konceptů na BPMN primitiva

3.1 Výchozí předpoklady

Důkladným studiem notace BPMN i metodologie DEMO bylo vypořádáno několik výchozích předpokladů, které navazují na pozorování dle [32], která jsme uvedli v sekci 2.

- V notaci BPMN nenajdeme žádné transakce tak, jak je definuje DEMO. Ve verzi BPMN 2.0 sice konstrukt nazvaný transakce existuje, ale v tomto případě se jedná pouze o speciální druh elementů, který vyžaduje definici kompenzačních aktivit v případě, že je nutné transakci odvolat.
- Neexistují tedy ani žádné pevně dané *transakční kroky* a výsledný procesní model může mít takřka jakoukoliv strukturu.
- Nejsou pevně specifikovány role účastníků (actorů) podnikového procesu.
- Není pevně dáno, že každý proces musí obsahovat popis „šťastných“ i „nešťastných“ scénářů a ty „nešťastné“ se v praxi často nemodelují.
- Není řečeno, které druhy aktivit se mají v modelu zachytit, a které jsou zbytečné nebo příliš detailní.

3.2 Základní pravidla používání notace BPMN

V této sekci jsou vypsány základní pravidla, jak používat některé elementy z notace BPMN tak, aby jejich použití korespondovalo s vybranými koncepty z metodologie DEMO dle základu metody představené v této práci. Z metodologie DEMO byly vybrány zejména ty koncepty, které jsou součástí transakčního axiomu, který je pro vlastní tvorbu modelů podnikových procesů nejzásadnější a vyjádření tohoto axiomu pomocí korespondujících elementů BPMN by výrazně přispělo k možnosti vytvářet kvalitnější modely v této notaci tak, jak bylo popsáno v sekci 1.3

3.2.1 Vyjádření actorů

Actor je v DEMO definován jako kombinace *odpovědnosti* a *kompetence* a není nezbytně svázán s konkrétním jedincem, který ve výsledku aktivitu či transakci fyzicky provádí. Pokud chceme v BPMN transakce zobrazovat, musí se podařit v paletě elementů, které BPMN nabízí nalézt takové, pomocí kterých je možné aktory korektně vyjádřit. Vybraný element musí splňovat tyto předpoklady:

- Musí jasně určovat, které kroky transakčního axiomu (C-acty a P-acty) patří do pole zodpovědnosti daného actoru.
- Musí umožňovat *komunikaci* s druhým actorem v transakci.

- Musí umožňovat navazování dalších actorů a transakcí v *transakčním stromu*.

Pro znázornění actora se nabízí využití BPMN elementu *plavecká dráha* (*swimlane*). BPMN nijak konkrétně nespecifikuje, jak plavecké dráhy využívat a nechává to tedy v rukou osoby, která model vytváří [29]. Pro znázornění dvou actorů v transakci by pak sloužily dvě sousedící plavecké dráhy. Komunikace mezi nimi by pak probíhala především pomocí *sekvenčních toků*.

Další možností, jak reprezentovat actora, je teoreticky použitím elementu *bazén*, který by tak fungoval samostatně pro každého actora a jelikož standard notace BPMN 2.0 zakazuje, aby sekvenční toky překročily hranice bazénu, museli by actori mezi sebou koordinovat kroky (C-acty, P-acty) pomocí zasílání zpráv. A právě zde tkví hlavní úskalí tohoto přístupu. V odborné literatuře (například [29]) je obecně uváděno, že není dobrým přístupem modelovat aktivity v rámci jednoho procesu uvnitř více bazénů. Vznikají pak problémy v navazování posloupnosti aktivit na sebe, ale i s korektností výsledného BPMN modelu dle zásad DEMO neboť využití bazénů nám rozdělí transakci fakticky do dvou procesů, což přináší problémy, které budou popsány v sekci 4.3.

3.2.2 Vyjádření C-actů a P-actů

C-acty a *P-acty* jsou jednotlivé kroky v transakčním vzoru, které jsou prováděny v definovaném pořadí a jsou v transakci vždy přítomny, i když jsou někdy prováděny *mlčky*. Pro vyjádření C-actů a P-actů navrhovaná metoda využívá BPMN element *úloha* (*Task*), který je dále nedělitelným typem *aktivity*. Úlohy mohou být dále rozčleněny na ty, které provádí člověk, které systém nebo nějaký stroj atp. Pro účely modelování popsané v této práci se nejlépe hodí abstraktní úlohy, které jsou bez označení.

V předchozím odstavci bylo zmíněno, že některé transakční kroky mohou být prováděny *mlčky*. To znamená, že ve skutečnosti neexistuje žádná aktivita, která by daný C-act prováděla, nicméně i provedení C-actu *mlčky* se dle metodologie DEMO počítá jako faktické provedení C-actu. [14]

3.2.3 Vyjádření C-factů

C-fact neboli *coordination fact* je výsledkem provedení C-actu. Je tedy přímo navázán na C-act, který jej „vytváří“. V BPMN lze C-fact vyjádřit třemi způsoby:

1. C-fact v modelu není explicitně vyjádřen. Jeho existence je zajištěna implicitně pomocí *sekvenčních toků* vycházejících z *úloh*, které reprezentují C-act.
2. C-fact je vyjádřen pomocí *signálu*. Důvod pro použití elementu signál je ten, že druhý actor by měl být o vzniku C-factu (respektive změně *C-worldu*) informován, což způsob představený v bodě 1 nezaručuje.
3. C-fact je vyjádřen pomocí *zprávy*, kterou zašle actor, který příslušný C-act vykonal druhému actorovi.

V [14] se uvádí, že v DEMO modelech je C-fact zakreslen mezi elementy, které reprezentují role actorů, což znamená, že C-fact, existuje v jejich společném *mezisvětě* neboli oba actoři mohou o vzniku C-factu vědět. Z toho vyplývá, že pro reprezentaci C-factu je nejvhodnější některý ze způsobů uvedených v bodech 2 nebo 3.

3.2.4 Vyjádření P-factů

Podobně jako C-fact je i *P-fact* výsledkem provedení konkrétního P-actu. V transakci se objevuje pouze jednou a zjednodušeně lze říci, že je „výsledkem“ celé transakce. P-fact může být hmotné povahy (např. upečení pizzy), ale i nehmotné (např. rozhodnutí poroty). Metoda, která je nastíněna v této práci, P-fact v modelu BPMN explicitně neuvádí. Důvody pro toto rozhodnutí jsou následující:

- Korespondující P-act (vyjádřený *úlohou*) nazvaný *Execute* a *sekvenční tok* z něj vycházející už sami o sobě vyjadřují vznik P-factu.
- Actor-Iniciátor je o vzniku P-factu informován ve chvíli, kdy je mu předložen pomocí C-actu *state*, který dle transakčního axiomu nastane vždy.
- Ve standardním i základním transakčním vzoru v metodologii DEMO je P-act i P-fact uváděn pouze v rámci agendy patřící Actorovi-Exekutorovi.

3.2.5 Vyjádření Agendy

Slovo „agenda“ vyjadřuje „co musí být uděláno“ neboli agenda je „úkol k vyřešení“. [14] ¹

V metodologii DEMO nejednají Actoři náhodně, ale každý jejich krok je striktně definován pomocí *action rules* v DEMO Action Modelu. Například je jasně řečeno za jakých okolností se vytvoření P-factu přislíbí (*promise*) a kdy je nutné *request* odmítnout. *Agenda* dle DEMO je tvořena čtveřicí parametrů:

- *C-fact*, který má být vykonán.
- *P-fact*, kterého se C-fact týká,
- *Čas* vztahující se k provedení P-factu.
- *Čas*, kdy by měl být C-fact vykonán.

Jelikož návrh metody, který tato práce popisuje, pracuje pouze s existujícími BPMN elementy, není v tuto chvíli možné kompletní Agendu pomocí nich vyjádřit. Řešením by bylo buď vytvořit vlastní doplňující model podobný DEMO Action Modelu nebo přímo DEMO Action Model. Dalším výzkumem je nutné ověřit, zda je vytvoření takové vrstvy nutné a případně takovou vrstvu vytvořit tak, aby

¹The word “agendum” is the singular form of the (plural) Latin word “agenda”. It means: what has to be done. In other words, an agendum is a to-do item. [14]

byla připravena pro implementaci v softwarovém nástroji, který by tak umožnil simulaci modelovaného procesu.

Metodologie navrhovaná v této práci si vystačí s částečným zobrazením agendy přímo v BPMN modelu, kterého je docíleno pomocí *plavecké dráhy* (případně *bazénu*), *sekvenčního toku* a *časového hlediska*. Jinými slovy díky plavecké dráze nebo bazénu, který obsahuje elementy, jež spadají do kompetence a odpovědnosti actoru, kterému je plavecká dráha nebo bazén příslušná a díky sekvenčnímu toku, který určuje pořadí provádění jednotlivých úloh daný actor v každém momentě v čase ví, kterou úlohu má provést. Pravidla pro vyhodnocení, zda přijmout *request* nebo *state* pro zjednodušení opomíjíme.

4 Vyjádření transakčního vzoru v BPMN

Pro metodologii, která je nastíněna v této práci je transakční vzor dle transakčního axiomu DEMO tím nejdůležitějším stavebním kamenem. Dává totiž řád a jasnou strukturu tomu, jak popsat podnikový proces, který je dle metodologie DEMO jen uspořádanou strukturou na sebe navázaných transakcí – *transakčním stromem*.

Vyjádření transakčního vzoru pomocí elementů z notace BPMN je tak zcela zásadní pro to, aby mohl být naplněn účel nově vznikající metody, kterým je přispět k vytváření BPMN modelů, které budou *kompletní* a *jednoznačné*. Jak bylo popsáno v sekci 3.2, vyjádření jednotlivých komponent transakčního vzoru je možné provést pomocí různých elementů a v této sekci jsou tyto různé přístupy rozebrány do většího detailu.

4.1 Vyjádření pomocí úloh

Prvním a nejjednodušším přístupem je vyjádřit transakční vzor čistě pomocí *úloh* a *sekvenčních toků*. Úlohy jsou použity pro vyjádření *C-actů* a *P-actů*. Sekvenční tok je použit ve více významech:

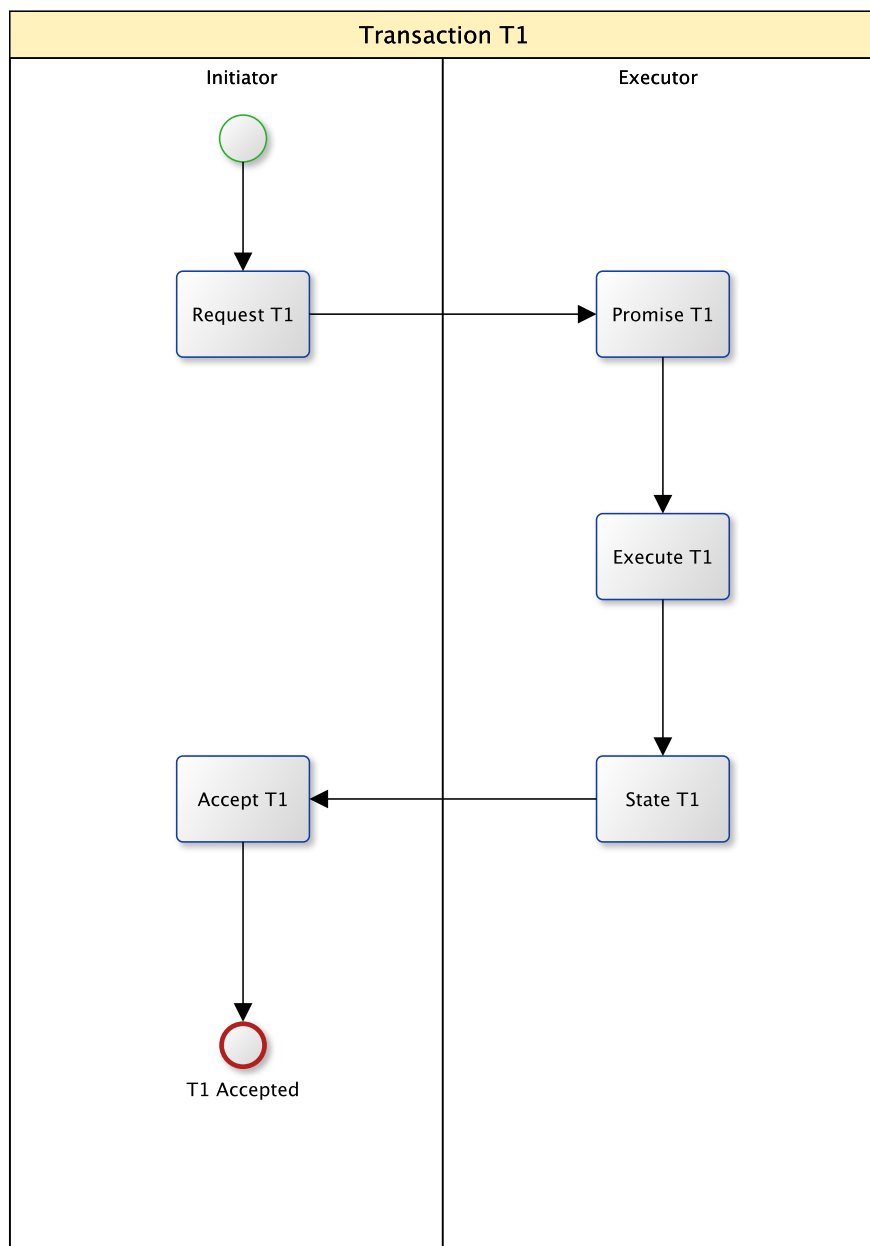
- jednak pro určení *pořadí* provádění jednotlivých C-actů,
- jednak pro zadávání C-actů do *agendy* druhému actorovi,
- jednak sekvenční tok vycházející z každé úlohy reprezentující C-act reprezentuje úspěšný *vznik korespondujícího C-factu*.

Na obrázku 2 je zobrazen *základní transakční vzor* pomocí úloh. *Bazén* slouží pro zastřešení celé transakce a *plavecké dráhy* oddělují agendy obou zúčastněných actorů (Iniciátora a Exekutora). Základní transakční vzor bere v potaz pouze „šťastné scénáře“, takže transakce vždy skončí ve stavu *T1 Accepted*. Všechny C-facty, ze kterých transakce již nikam dál nepokračuje jsou vyjádřeny BPMN elementem *konečná událost (End event)*.

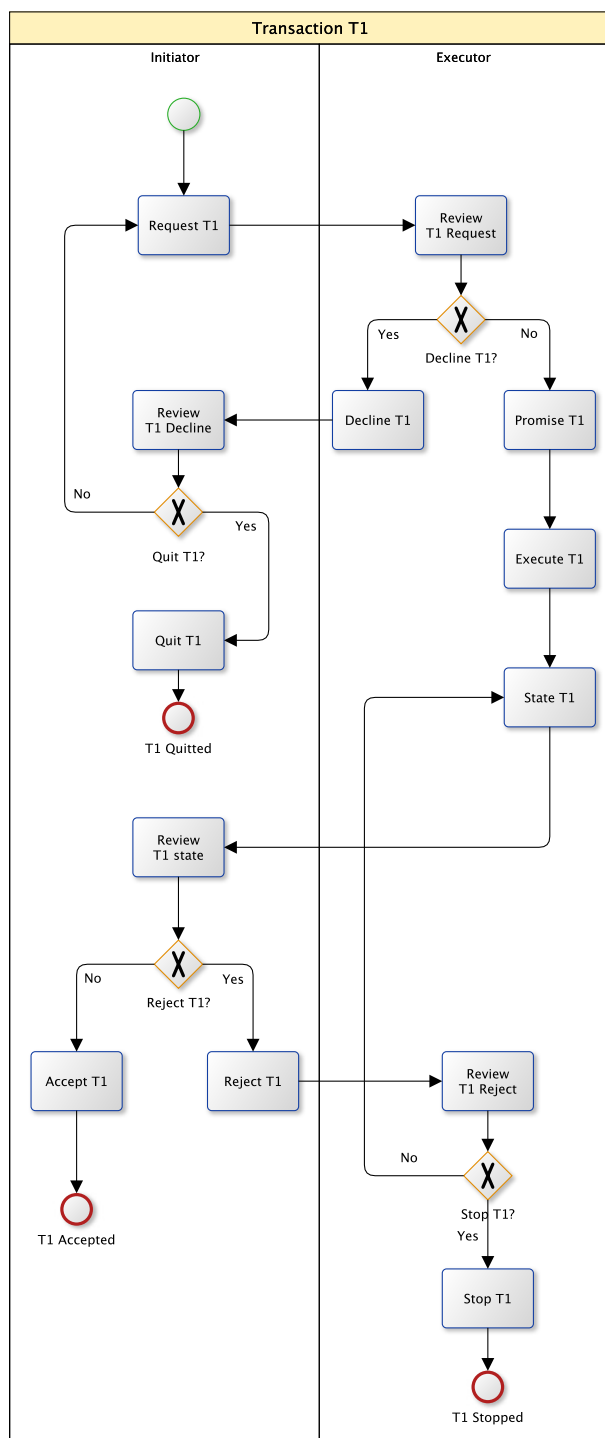
Obrázek 3 zobrazuje *standardní transakční vzor*, tedy kompletní transakci včetně všech „nešťastných scénářů (unhappy paths)“. Oproti základnímu transakčnímu

vzoru zde přibyly čtyři *brány* označující místo, kde actor musí rozhodnout, jak s agendou naloží. Rozhodnutí je exkluzivní, tudíž je použita *exkluzivní brána XOR*.

V tomto zobrazení nejsou zobrazeny žádné C-facty, což je z hlediska DEMO problematické. Ačkoliv jejich vznik vyjadřuje sekvenční tok vycházející z C-actu, C-fact není explicitně vyobrazen a druhý actor o něm nemusí vědět, protože v modelu není žádná komunikace zobrazena. Dle [14] však oba Actoři mají mít umožněno vědět o vzniku C-factu.



Obrázek 2: Základní transakční vzor pomocí Úloh



Obrázek 3: Standardní transakční vzor pomocí Úloh

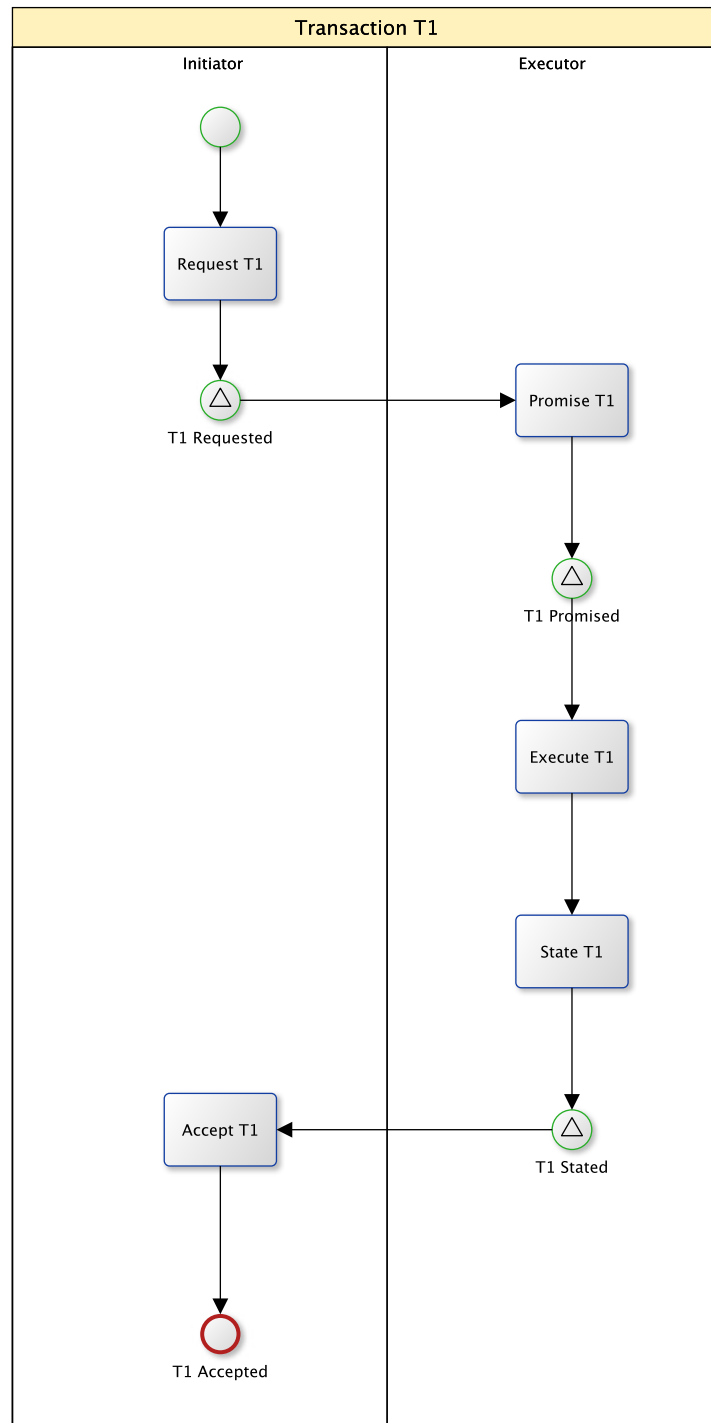
4.2 Vyjádření pomocí úloh a signálů

Další možností, jak transakční vzor zobrazit, je využití *úloh* a *signálů*. Narozdíl od způsobu popsaného v sekci 4.1 slouží *sekvenční toky* v tomto případě pouze k vyjádření pořadí, ve kterém jsou C-acty prováděny a pro předávání agendy mezi actory. O reprezentaci vzniklého C-factu se zde však stará BPMN element *signál*, což poskytuje dvě velké výhody. Jednak je díky tomu C-fact v modelu vizuálně zastoupen, takže se výsledný model více blíží transakčnímu vzoru tak, jak ho definuje DEMO a jednak BPMN element signál e dle specifikace [23] i [29] možné použít ke komunikaci uvnitř procesu, respektive k odeslání neadresné zprávy komukoliv, kdo je připraven jí naslouchat. Toto mnohem přesněji vyhovuje popisu společného *mezisvěta*, který jsme uvedli v sekci .

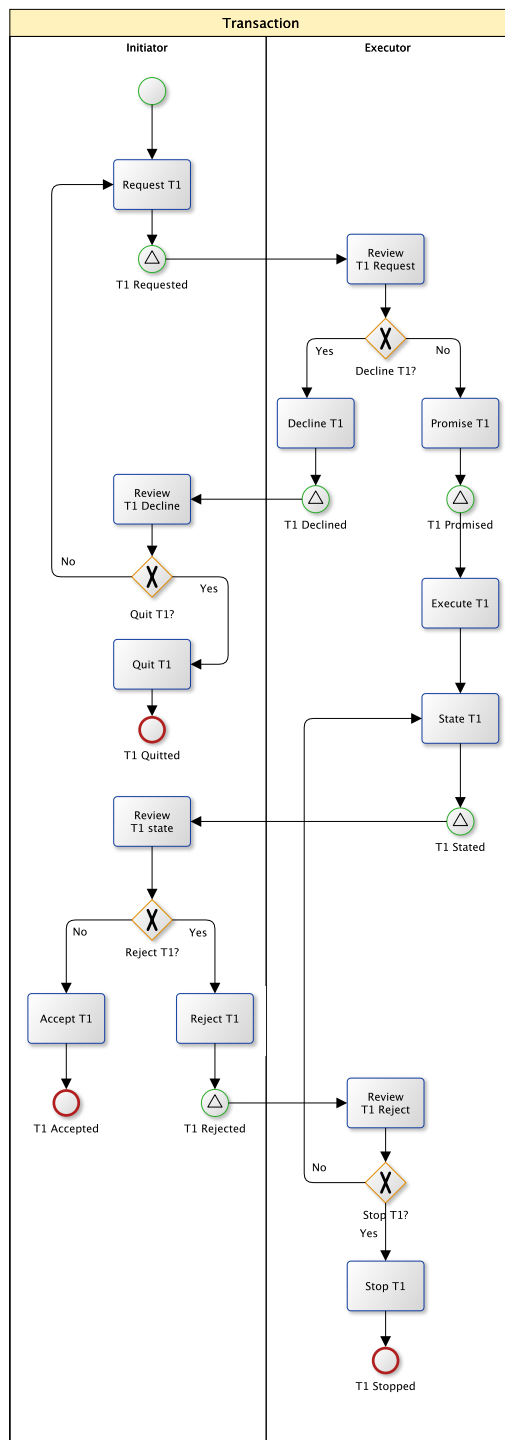
Pro reprezentaci actorů jsou v tomto případě opět použity *plavecké dráhy* uvnitř *bazénu*, který zastřešuje celou transakci. C-facty, ve kterých transakce končí jsou i v tomto případě vyjádřeny pomocí BPMN elementu *konečná událost*.

Na obrázku 4 vidíme *základní transakční vzor* pomocí úloh a signálů. Lze vidět, že po provedení C-actu *Request T1* se sekvenční tok přesouvá do signálu *T1 Requested*, který vyjadřuje vzniklý C-fact. Předpokládáme, že Actor-Executor tomuto Signálu naslouchá a může tak provést C-act *Promise T1*.

Obrázek 5 zobrazuje *standardní transakční vzor* vyjádřený úlohami a signály.



Obrázek 4: Základní transakční vzor pomocí Úloh a Signálů



Obrázek 5: Standardní transakční vzor pomocí Úloh a Signálů
16

4.3 Vyjádření pomocí Úloh a Zpráv

Jiným přístupem k vyjádření transakčního vzoru je využít ke komunikaci mezi actory BPMN element *zpráva*. Jelikož specifikace BPMN předepisuje, že zpráva může být použita pouze ke komunikaci mezi *bazény*, musí být v tomto případě pro každého actora vyhrazen zvláštní bazén. Na obrázku 6 lze vidět *základní transakční vzor* vyjádřený právě tímto způsobem.

Předávání agendy v tomto případě probíhá pomocí zaslání zprávy a *průběžné události* (*Intermediate event*). Jakmile *Actor-Iniciátor* provede C-act *Request T1*, vyšle zprávu reprezentující C-fact, kterou přijme *Actor-Exekutor* na druhé straně pomocí *počáteční události* typu *catch (Message)*, čímž vznikne nová instance procesu na straně Exekutora. O každém provedeném C-actu je pak druhý actor informován pomocí zprávy.

V případě základního transakčního vzoru se většina problematických míst tohoto přístupu neprojevuje. K diskusi je však rozdělení actorů do dvou bazénů, které v podstatě znamená rozdělení transakce na dva podprocesy, jeden na straně Iniciátora a jeden na straně Exekutora. Toto je poměrně výrazné odchýlení od transakčního axiomu tak, jak ho definuje DEMO a přináší komplikace, které se projeví v případě *standardního transakčního vzoru*.

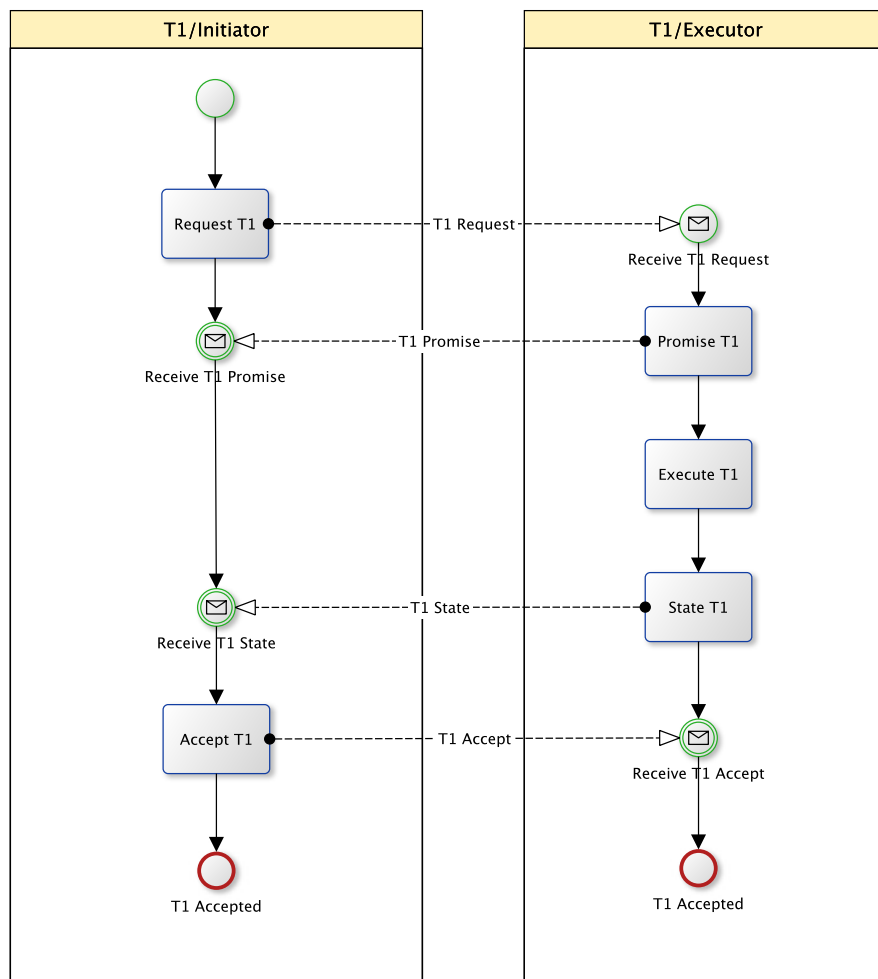
Ve standardním transakčním vzoru totiž existují místa, která při maximální snaze o dodržení standardního transakčního vzoru z DEMO mohou vyústit v *deadlock*. Jedná se o případy, kdy se actor rozhodne zamítnout výsledek C-actu *request* nebo *state*. V takovém případě totiž musí actor odpovědný za C-act rozhodnout, zda transakci zastavit nebo se pokusit o C-act znovu. Transakční vzor dle DEMO, ale nepočítá s tím, že by transakce byla rozdělena mezi dva procesy a že by měl existovat mechanismus, jak o tomto rozhodnutí informovat druhého actora. Řešením je v tomto případě přidat do transakčního vzoru mezistavové zprávy.

V případě, že bychom nepoužili mezistavové zprávy, nastal by problém hned v případě, že by se Actor-Exekutor rozhodl pro *Decline T1*. V takovém případě Exekutor vyšle zprávu Actoru-Iniciátorovi a ten může buď transakci zastavit použitím *Quit T1* nebo poslat *Request T1* znovu. Jenže sekvenční tok v procesu na straně Exekutora je stále v situaci *Decline T1* a bez mezistavové zprávy (neboli zprávy o rozhodnutí jak se Iniciátor rozhodl naložit s *T1 Declined*) nemůže dále pokračovat. Částečným řešením, které můžeme vidět na obrázku 7, by v tomto případě mohlo být použito na tomto místě koncový stav *T1 Declined* a instanci procesu u Exekutora ukončit. V případě, že by se Iniciátor rozhodl transakci neukončit a poslal *request* znovu, vznikla by nová instanci procesu na straně Exekutora.

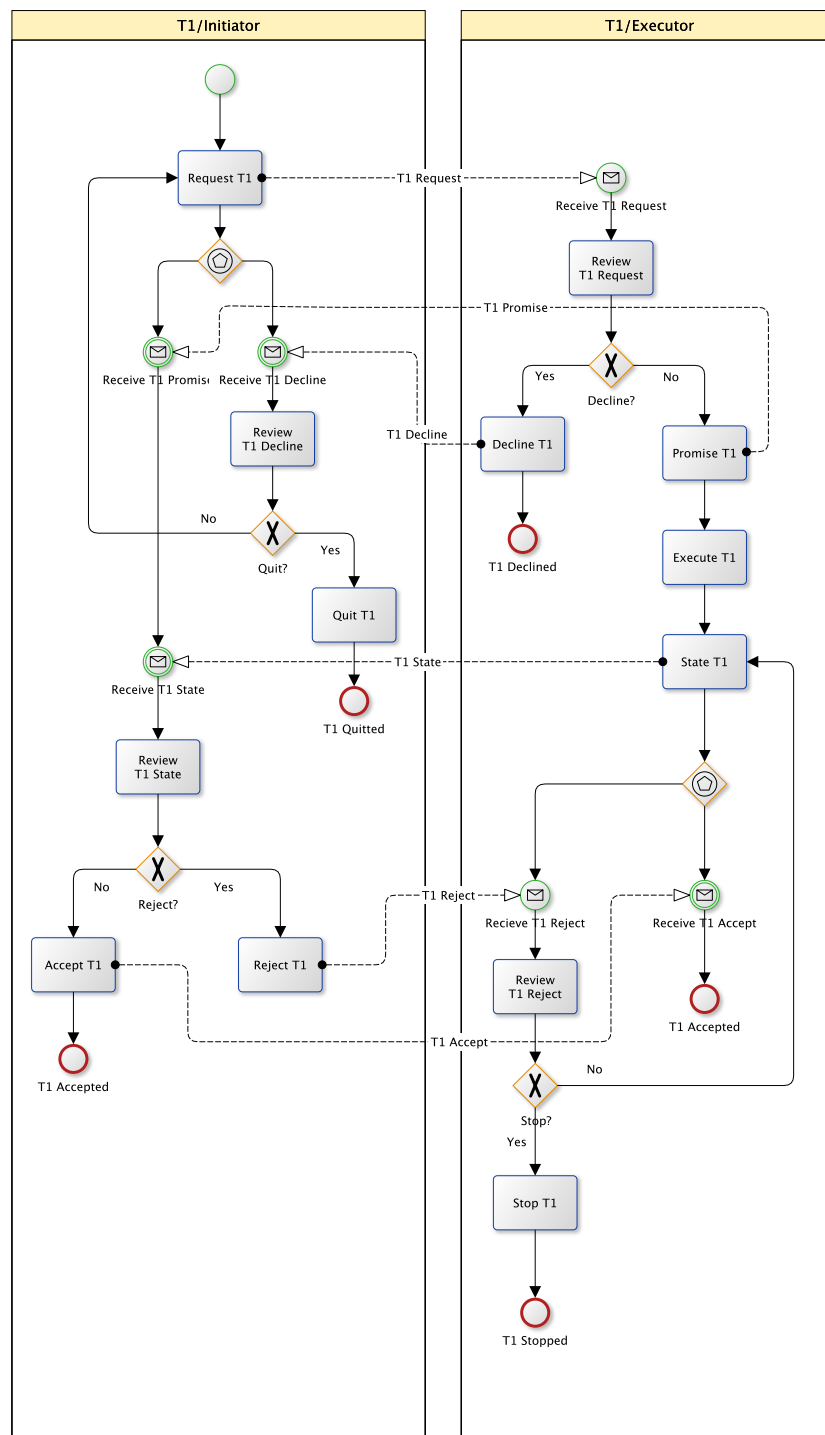
Větší problém však nastane v případě, kdy by se Iniciátor po obdržení *T1 State* rozhodl tento *state* odmítnout a provést *Reject T1*. Exekutor by následně obdržel zprávu o *rejectu* a musel by rozhodnout, zda transakci zastavit provedením *Stop T1* nebo provést *State T1* znovu. Jenže bez mezistavových zpráv se Iniciátor nemá šanci dozvědět výsledek tohoto rozhodnutí Exekutora. Řešení popsané v předchozím odstavci zde nebude fungovat, protože pokud by se Exekutor rozhodl provést znovu *State T1*, proces na straně Iniciátora by již byl ukončen.

Jediným řešením jsou tedy mezistavové zprávy, které informují druhého actora o rozhodnutích v případě, kdy by mohlo dojít k deadlocku. Toto řešení lze vidět na obrázku 8. Jakmile nastane *Decline T1* proces na straně Exekutora se přesune do *průběžné události (Multiple)* a čeká na příchod mezistavové zprávy od Iniciátora, jestli se rozhodl transakci ukončit nebo pošle *request* znovu. Na základě toho buď proces ukončí nebo ho přesune na začátek a bude čekat na *request* od Iniciátora. Analogicky je pak mezistavových zpráv využito v případě *state*.

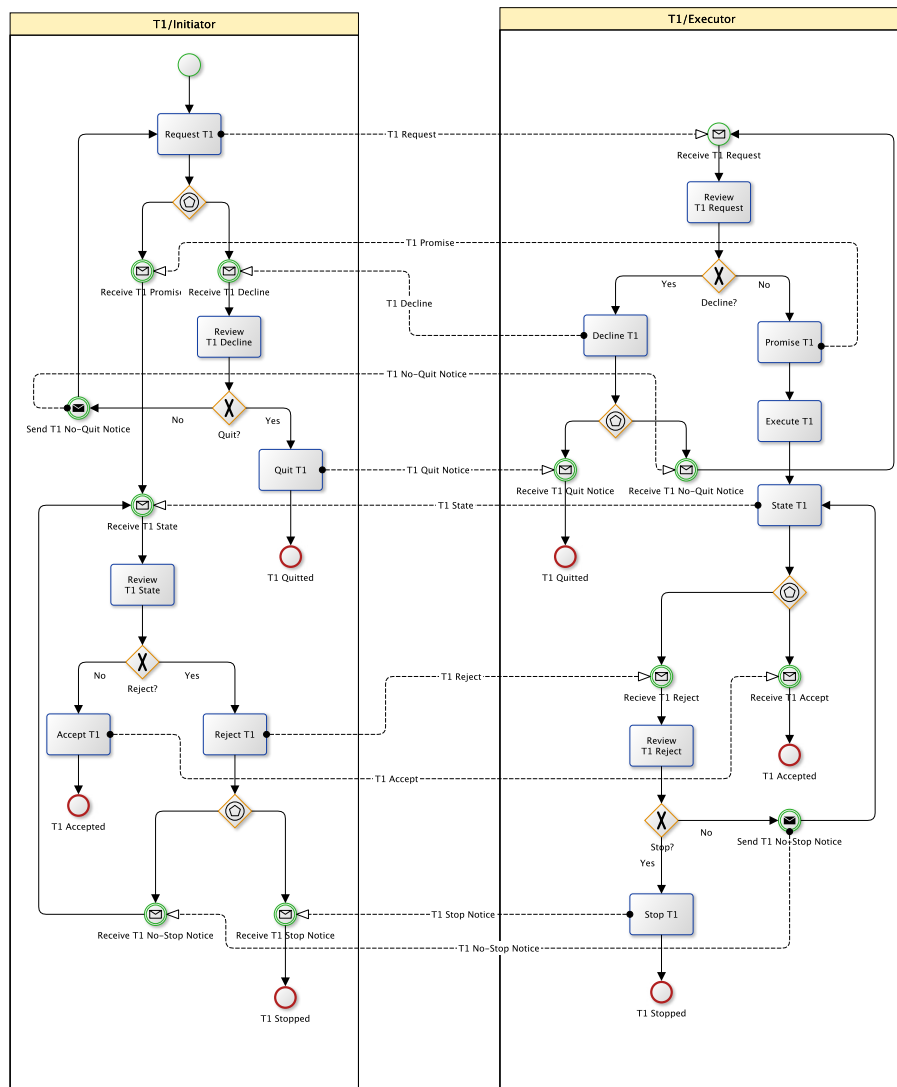
Využití mezistavových zpráv je však výrazným odchýlením od transakčního axiomu, protože tyto zprávy fakticky přidávají další komunikaci, se kterou transakční axiom nepočítá. Z tohoto důvodu je tento způsob pro vyjádření transakčního axiomu nevhodný.



Obrázek 6: Základní transakční vzor pomocí úloh a zpráv



Obrázek 7: Standardní transakční vzor pomocí úloh a zpráv



Obrázek 8: Standardní transakční vzor pomocí *úloh* a *zpráv* (včetně mezistavových zpráv)

5 Návrh metody pro vytváření BPMN modelů dle zásad DEMO

V této sekci je popsána metoda pro vytváření BPMN dle zásad DEMO. Představená metoda je navržena pro vytváření zcela nových modelů, nikoli k analýze a doplnění již existujících BPMN modelů. Cílem této metody je umožnit popisovat

průběh podnikových procesů v oblíbené notaci BPMN a zároveň zajistit, aby výsledný model byl *kompletní* a *konzistentní*.

5.1 Předpoklady

Aby bylo možné implementovat tuto metodu v praxi, je nezbytné, aby odpovědní pracovníci byli vybaveni následujícími teoretickými znalostmi, které vycházejí z Ψ -theory, Enterprise ontology a metodologie DEMO.

- Znalost *operačního axiomu* a schopnost rozlišit komunikační acty od produkčních.
- Znalost *transakčního axiomu*, *základního* i *standardního transakčního vzoru* včetně všech jeho kroků (C-actů, C-factů, P-actů, P-factů).
- Znalost a schopnost vytvořit DEMO Actor-Transaction Diagram (ATD) a DEMO Process Structure Diagram (PSD).
- Znalost notace BPMN včetně elementů, které oblíbená publikace [29] označuje jako Level 2.
- Znalost a schopnost aplikovat *kompoziční axiom*, tedy umět identifikovat jak jsou na sebe transakce navázané.

5.2 Krok 1: Získání textového popisu procesu

Při vytváření modelu určitého podnikového procesu je obvyklým prvním krokem získání informací od osob, které se tohoto procesu účastní nebo mu dobře rozumí. Může se jednat o výkonné i vedoucí pracovníky a obvykle je užitečné posbírat více pohledů na jeden konkrétní proces a následně diskutovat nesrovnalosti, které mohou vzniknout.

Díky tomu, že sběr informací provádějí pracovníci se znalostí transakčního axiomu i transakčních vzorů, mohou se cíleně zeptat na jednotlivé jeho kroky. Tedy zjistit, jak probíhá *request*, jak *state*, jak *accept* a podobně. Na konci by tedy měl vzniknout textový popis procesu s maximem relevantních informací pro další postup.

5.3 Krok 2: Aplikace distinkčního axiomu

Nyní jsme schopni na vzniklý textový popis procesu, chodu organizace či její složky aplikovat tzv. *Performa-Informa-Forma analýzu*, což není nic jiného než praktická aplikace *distinkčního axiomu*, který je popsán v sekci ???. Použijeme stejný postup jako v [14] a v textovém popisu označíme Performa aktivity červeně, Informa zeleně a Forma modře. Je naprosto přirozené, že u některých aktivit v textu bude problematické určit o jaký druh se jedná. S tím, jak budeme provádět další kroky dojdeme i k většímu porozumění celému procesu a je možné rozhodnutí přehodnotit. Výsledkem této části tedy je textový popis procesu s identifikovanými a vizuálně označenými Performa, Informa a Forma aktivitami.

5.4 Krok 3: Aplikace operačního axiomu

V tomto kroku budeme pracovat pouze s aktivitami, které jsme v Kroku 2 označili jako Performa (červeně). Úkolem je rozlišit, zda se jedná o C-act, C-fact, P-act nebo P-fact a dále k nim identifikovat příslušné actory tak jak předepisuje *operační axiom* popsany v sekci ?? . Opět není důvod nedodržet postup doporučený v [14], tedy části textu, které indikují actora označit hranatými závorkami „[“ a „]“ . Části textu, které indikují C-act nebo C-fact označíme kulatými závorkami „(“ a „)“ . Nakonec P-acty a P-facty nalezené v textu označíme špičatými závorkami „<“ a „>“ . Text uzavřený v závorkách navíc podtrhneme.

5.5 Krok 4: Zápis nalezených transakcí a jejich parametrů

Ve čtvrtém kroku označené C-acty, C-facty, P-acty a P-facty roztrídíme dle jednotlivých *transakčních kroků* (*request, promise, state, accept atd.*) dle *transakčního axiomu*, jehož popis najdeme v sekci ?? . Zároveň specifikujeme výsledek transakce a všechny poznatky učiněné v předchozích krocích zapíšeme společně do tabulky, která má strukturu dle tabulky 1. Navržená struktura vychází ze struktury představené v [21].

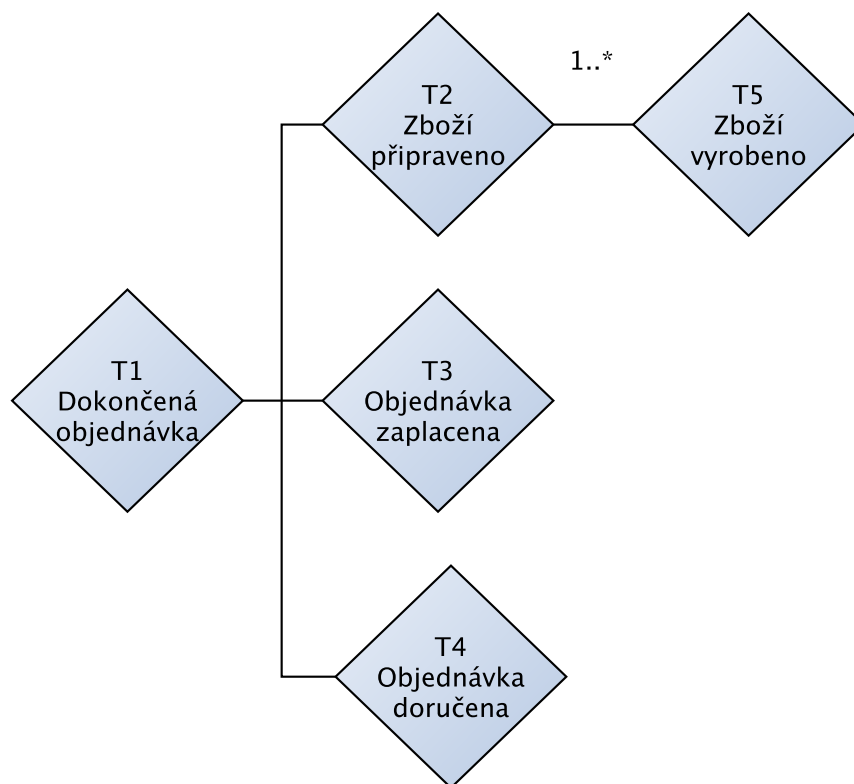
Silnou stránkou DEMO, respektive transakčního axiomu je, že dává každé transakci (a tím i podnikovému procesu) pevně danou strukturu, kterou nelze obejít nebo upravit. I když v textovém popisu procesu nenalezneme popsání všechny transakční kroky (což se stane téměř vždy), neznamená to, že neexistují. V takovém případě je třeba jít zpět do Kroku 1 a zjistit chybějící informace od doménových a procesních vlastníků a dle nových zjištění upravit i výsledky následujících kroků metody.

ID transakce	T01
Název transakce	Vyřízení objednávky <i>O</i>
Výsledek transakce	Objednávka <i>O</i> byla vyřízena
Initiator	Zákazník
Executor	Mia
Request	Zavolání s objednávkou
Promise	Objednávka potvrzena s cenou a časem vyhotovení
State	Zboží předáno zákazníkovi
Accept	Zákazník odchází s balíčkem z prodejny
Decline	Zamítnutí objednávky pokud zboží není na skladě
Reject	<i>chybí v popisu</i>
Revoke request	<i>chybí v popisu</i>
Revoke promise	<i>chybí v popisu</i>
Revoke state	<i>chybí v popisu</i>
Revoke accept	<i>chybí v popisu</i>

Tabulka 1: Tabulka s parametry transakce

5.6 Krok 5: Aplikace kompozičního axiomu

Jak již bylo popsáno v ?? podnikový proces můžeme definovat jako množinu volně propojených transakcí. Úkolem pátého kroku je tedy identifikovat exekuce transakcí, které jsou závislé na předchozím provedení exekucí jiných transakcí. Jedinou možností jak toto prakticky provést je vrátit se k textovému popisu procesu a pokusit se identifikovat fráze, které značí závislost provádění P-actů nebo P-factů na sobě. Pokud se nám podaří objevit, že vznik P-factu B byl iniciován v průběhu vytváření P-factu A a zároveň je vznik P-factu A na B závislý, poté B je součástí A. Všechny takto objevené závislosti zakreslíme do diagramu, který můžeme vidět na obrázku 9.



Obrázek 9: Struktura závislosti transakcí v podnikovém procesu

5.7 Krok 6: Vytvoření DEMO modelů

Nyní již máme dostatek informací pro vytvoření dvou DEMO modelů, které budou sloužit jako pevný bod pro verifikaci výsledného BPMN modelu i pro diskusi s business uživateli v případě požadavků na úpravy. Důvodem proč k této

diskusi využívat DEMO modely (zejména ATD) a ne přímo výsledný BPMN model je vysoká úroveň komplexnosti BPMN modelu, který znesnadňuje jeho porozumění business uživateli.

Tabulky s popisem transakcí, které vznikly v Kroku 4 je dostatečným podkladem pro vytvoření DEMO Actor-Transaction Diagramu (ATD) i DEMO Process Structure Diagramu (PSD). ATD model zachycuje jevy v organizaci na nejvyšší úrovni abstrakce, tedy pouze actory a mezi nimi probíhající transakce. PSD jde o úroveň níže a zobrazuje jednotlivé transakční kroky stejně jako vzájemnou provázanost transakcí, kterou jsme indetifikovali v předchozím kroku. Oba modely jsou podrobně popsány v rámci kapitoly 4.

5.8 Krok 7: Vytvoření BPMN modelu

Nyní již máme vytvořený dostatečně silný základ a můžeme přikročit k vytvoření samotného BPMN modelu, který je výsledkem celé metody. Při jeho tvorbě využijeme zejména PSD diagram vytvořený v předchozím kroku a způsob vyjádření standardního transakčního vzoru pomocí *úloh* a *signálů* popsaný v sekci 4.2.

Při vytváření BPMN modelu postupejeme tak, že následujeme PSD diagram od místa, kde dojde k provedení C-actu *request* u kořenové transakce celého podnikového procesu a postupně převádíme jednotlivé transakční kroky do BPMN primitiv (*úloh*, *signálů*, *sekvencních toků*, *bran*) popsaných v sekci 4.2. V případě, že je podnikový proces složen z více vzájemně provázaných transakcí musíme při vytváření BPMN modelu v momentě, kdy v PSD diagramu dojdeme na místo, kde je při provedení C-actu *promise* vytvořen *request* na další transakci, odklonit sekvencní tok a pokračovat v provádění transakčního vzoru transakce-potomka. Stejně tak pokračujeme i v dalších případech dokud se opět „nevynoříme“ (v případě, že všechny transakce-potomci dopadnou úspěšně) v kořenové transakci a jsme schopni provést exekuci P-actu.

6 Další výzkum

Obsah této práce je prvním pokusem o vytvoření metody pro vytváření BPMN modelů podnikových procesů za použití teoretických konstruktů z Ψ -theory , Enterprise ontology a metodologie DEMO. Jako každá první verze i tato musí být doplněna dalším výzkumem a všechna představená východiska musí být prověřena a potvrzena. Pro zdůraznění je zde vypsáno několik bodů, kde autor považuje další výzkum za nezbytný:

1. Posouzení důležitosti a případně vytvoření určité formy textového modelu na způsob DEMO Action Modelu, který umožní jednak naprosto přesně zachytit informace o agendě actorů v daném momentě, jednak bude sloužit jako podklad pro implementaci simulace modelu pomocí nějakého softwarového nástroje a jednak umožní formální verifikaci pomocí gramatiky, která bude pro takový model vytvořena.

2. Posouzení, zda je BPMN element *úloha* dostatečný pro vyjádření DEMO C-actů a P-actů.
3. Posouzení použití konkrétních názvů aktivit pro jednotlivé transakční kroky identifikované v Kroku 4 metody.
4. V DEMO verze 3 byla přidána do transakčního vzoru další vrstva umožňující prakticky kdykoliv „vzít zpět“ již vyjádřený C-act. Dalším výzkumem musí být posouzeno zda je toto možné znázornit v DEMO, jestli je to žádoucí a jaký to bude mít vliv na komplexnost výsledných BPMN modelů.
5. Návrh implementace automatického generování BPMN modelů z DEMO modelů pomocí vyjádření transakčního vzoru v BPMN dle sekce 4. Výsledkem by mělo být propojení se softwarovou aplikací, kterou připravuje firma Formetis.

Reference

- [1] BPEL – jazyk pro automatizaci procesů.
- [2] Enterprise Engineering and DEMO.
- [3] ISO 9000:2005.
- [4] Notace.
- [5] UML 2 Activity Diagramming Guidelines.
- [6] Rámce a metodiky, 2013.
- [7] Mike Bandor. Process and Procedure. 2007.
- [8] J. Barjis. Enterprise Modeling and Simulation Within Enterprise Engineering. *Journal of Enterprise Transformation*, 1(3):185–207, 2011.
- [9] Artur Caetano, Aurélio Assis, José Borbinha, and José Tribolet. An Application of the ψ -Theory to the Analysis of Business Process Models. *Enterprise Information Systems of the Future*, 2012.
- [10] Artur Caetano, Aurélio Assis, and José Tribolet. Using business transactions to analyse the consistency of business process models. *Proceedings of the Annual Hawaii International Conference on System Sciences*, pages 4277–4285, 2011.
- [11] Ondřej Černý. *BPEL*. PhD thesis, Vysoká škola ekonomická, 2010.
- [12] Jiří Chytil and Zdeněk Lehocký. Vývojové diagramy - 1. díl, 2005.
- [13] Jan L. G. Dietz. Enterprise Ontology, 2005.

- [14] Jan L G Dietz. *Enterprise ontology: Theory and methodology*. 2006.
- [15] Marlon Dumas, M La Rosa, Jan Mendling, and Hajo Reijers. *Fundamentals of Business Process Management*. Springer Berlin Heidelberg, 2013.
- [16] Hans-Erik Eriksson and Magnus Penker. *Business Modeling With UML: Business Patterns at Work*. 2000.
- [17] Hans-Erik Eriksson and Magnus (Open Training) Penker. Business Modeling with UML.
- [18] Paul Harmon. *Business Process Change: A Business Process Management Guide for Managers and Process Professionals*. Morgan Kaufmann, 2014.
- [19] Andreas Jedlitschka, Outi Salo, and Frank Bomarius. Process Management. *Journal of Software Maintenance and Evolution: Research and Practice*, (May):143–149, 2010.
- [20] Frank Leymann. BPEL vs BPMN 2.0: Should you care?, 2009.
- [21] Pavel Náplava. A7B16ISP Informační systémy a procesní řízení, 2015.
- [22] František Ochrana. *Metodologie vědy: úvod do problému*. Praha: Karolinum, 2009.
- [23] Object Management Group Omg. Business Process Model and Notation (BPMN) Version 2.0, 2011.
- [24] Theodore Panagacos. *The Ultimate Guide to Business Process Management: Everything you need to know and how to apply it to your organization*. CreateSpace Independent Publishing Platform, 2012.
- [25] Gregor Polancic. Managing business processes with BPMN – SWOT Analysis, 2014.
- [26] Jan C. Recker, Michael Rosemann, Marta Indulska, and Peter Green. Business process modeling: a comparative analysis. *Journal of the ...*, 10(4):333–363, 2009.
- [27] Michael Rowley. Why use BPMN for BPEL?, 2009.
- [28] Boris Shishkov and Jan Dietz. Deriving Use Cases from Business Processes - The advantages of DEMO. pages 249–257, 2005.
- [29] Bruce Silver. *BPMN Method and Style, 2nd Edition, with BPMN Implementer’s Guide: A structured approach for business process modeling and implementation using BPMN 2.0*. 2011.
- [30] Bruce Silver. Executable BPMN 2.0, 2011.
- [31] Bruce Silver. BPMN: The Four Aspects of Process, 2013.

- [32] Dieter Van Nuffel, Hans Mulder, and Steven Van Kervel. Enhancing the formal foundations of BPMN by enterprise ontology. *Lecture Notes in Business Information Processing*, 34 LNBIP:115–129, 2009.
- [33] Petr Vašíček. Úvod do bpmn, 2008.
- [34] Zuzana Vejražková. Design and Engineering Methodology for Organizations. 2012.
- [35] Zuzana Vejražková. *Business Process Modelling and Simulation: DEMO, BORM and BPMN*. PhD thesis, České vysoké učení technické, 2013.
- [36] Mathias Weske. *Business Process Management*. Springer, 2007.
- [37] Václav Řepa. *Podnikové Procesy. Procesní řízení a modelování*. Grada Publishing, a.s., 2007.
- [38] Václav Řepa. Řízení procesů versus procesní řízení, 2008.