

Kapitola 5: Využití metodologie DEMO pro vytváření BPMN modelů

Bc. Štěpán Heller

27. prosince 2015

1 Motivace

Tato kapitola se zabývá přístupy, jak využít silných stránek metodologie DEMO pro vytváření BPMN modelů. Cílem těchto přístupů je vytvořit modely podnikových procesů v BPMN, které budou splňovat stejné požadavky, jaké klade metodologie DEMO. Tedy, že modely musí být ucelené, konzistentní, úplné, výstižné a obsahují jen nezbytné množství informací. Jak píše [?], BPMN začíná tam, kde DEMO končí, tedy by se mohlo zdát, že je vhodné jejich silné stránky spojit.

1.1 Slabé stránky DEMO

Jak již bylo v průběhu této práce několikrát zmíněno, na metodologii DEMO je nejcennější, že říká tvůrcům procesních modelů „jak“ by měli při vytváření modelů podnikových procesů postupovat a nedává jim tedy pouze notaci, pomocí které je možné to realizovat. 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 zkoušení 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 3 měsíčního kurzu měli jeho účastníci problém se základními technikami metodologie DEMO, jakou je například korektní aplikace distinction axiomu performan-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 množství nástrojů, které DEMO podporují, kterých je minimum a žádný z nich dosud nepokrývá všechny aspect

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

Cílem této práce je umožnit takovým 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.

Problematické 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 [?] 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 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 ontologiích a Ψ -theory, který by mohl nahradit tyto pseudo-metodologie.

Zajímavá zjištění uvádí [?], kde 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 takový model kompletní, neboli jestli všechny kroky transakce dle transakčního axiomu DEMO se dají namapovat na aktivity BPMN modelu. Po důkladné analýze těchto procesů došli k následujícím zjištěním:

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

- Chybělo 50 % promise C-actů. Requesty byly implicitně potvrzovány bez jakékoliv jasné dohody mezi aktory.
- Chybělo 25 % state C-actů. Výsledky transakce tak nebyly jasně komunikovány s jejím iniciátorem a tedy není jasné, zda dohlídnutí 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 nepociťují [?]) a jeho vytvoření by zvýšilo kvalitu vytvářených modelů podnikových procesů.

2 Benefity kombinace DEMO a BPMN

Hlavním přínosem této práce je vytvoření rámce, který vychází z metodologie DEMO a slouží ke stanovení základů metodologie, která 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. Abstrahujme zde od toho, zda je to vůbec možné, to musí být potvrzeno dalším výzkumem. Pokud bychom se k takovému přístupu rozhodli, nedosáhli bychom vyřešení problémů uvedených v předchozí 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 s sebou nést dlouhou křivku učení, kterou obsahuje DEMO. Co je tedy míněno „kvalitnějšími“ BPMN modely?

- Modely takto vytvořené budou *kompletní* dle transakčního axiomu, tedy nebude chybět žádný krok dle transakčního axiomu.
- Modely takto vytvořené budou *jednoznačné*, neboli při modelování toho samého procesu by vždy měl vzniknout ten stejný model
- Modely budou oproštěné od implementačních detailů a budou tedy zachycovat jen skutečné jádro organizace

3 Existující možnosti společného využití obou přístupů

Vhodnost spojení obou přístupů nešla pozornosti jiných autorů, například [?] nebo . Ve stručnosti si tak tyto přístupy nyní popíšeme, protože i na jejich zjištěních autor této práce svůj návrh metodologie staví.

Práce [?] 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 Actora (Actor-Initiator nebo Actor-Executor)

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

- Existují identifikovatelní actoři plnící role, kde se actor vztahuje k roli a nikoli nutně k jedinci, který tu 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 aktor vyjma kořenového actor-initiator má jeden a pouze jeden vztah actor-executor k jednomu a pouze jednomu P-factu, ale může mít $0...n$ actor-initiator vztahů k jiným (child) P-factům.
- Každý P-fact může být definován jako množina $0...n$ jeho potomků
- U každého rodičovského P-factu musí nejdříve dojít k dokončení (state and accept) child P-factů dříve než může začít exekuce rodičovského P-factu.
- Actor, který je exekutorem nějakého P-factu je zároveň iniciátorem každého jeho child P-factu.
- 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 pouze exekutorem tohoto P-factu.
- Existuje množina osmi atributů (Request, Promise, etc.), které se unikátně vztahují ke konkrétnímu C-factu, která popisuje aktuální stav komunikace ohledně úkonu (vytvoření P-factu).

4 Návrh metodologie

4.1 Výchozí předpoklady

Důkladným studiem notace BPMN i metodologie DEMO bylo vypořádání několik výchozích předpokladů, které navazují na pozorování dle [?], která jsme uvedli v sekci 3.

- 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.
-

4.2 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 konstrukty z metodologie DEMO dle konceptu metodologie 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.

4.2.1 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 Actora
- 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 [?]. 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 *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 actoři mezi sebou koordinovat kroky (C-acty, P-acty) pomocí zasílání zpráv. A právě zde tkví hlavní úskalí tohoto přístupu. V odborné literatuře (například [?]) 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.

4.2.2 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 tato metodologie používá BPMN element *Úlohy (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.

Second, C-acts may be performed tacitly. This means that there is no act at all that counts as performing the C-act. Tacitly performing a C- act, however, is still performing that C-act.

4.2.3 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 explicitně v modelu není 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 první způsob, ve kterém je to předpokládáno implicitně, nezaručuje.
3. C-fact je vyjádřen pomocí *Zprávy*, kterou zašle Actor, který příslušný C-act vykonal druhému Actorovi.

This C-fact is drawn between the two actor roles to show that it is a fact in their social or intersubjective world ([32]): both actors are allowed to know the fact.

4.2.4 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). Metodologie, 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.

4.2.5 Vyjádření Agendy

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

V metodologii DEMO nejednají Actoři náhodně, ale každý jejich krok je striktně definován pomocí *action rules* v *Action modelu*. Neboli je jasné řečeno například za jakých okolností se vytvoření P-factu přislíbí (*Promise*) a kdy je nutné Request odmítnout. Agenda dle DEMO je čtveřice 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 metodologie, 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 Action Modelu DEMO nebo přímo DEMO Action Model. Dalším výzkumem je nutné ověřit, zda je vytvoření takové vrstvy nutné a případně takovou vrstvu vytvořit tak, aby 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. Neboli díky Plavecké dráze nebo Bazénu, který obsahuje elementy, jež spadají do kompetence a odpovědnosti Actora, 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.

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. [?]

5 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 metodologie, kterým je přispět k vytváření BPMN modelů, které budou kompletní a jednoznačné. Jak bylo popsáno v sekci 4.2, vyjádření jednotlivých komponent transakčního vzoru je možné provést pomocí různých elementů a v této sekci jsou tyto různé přístupy rozebrány do většího detailu.

5.1 Vyjádření pomocí Úloh

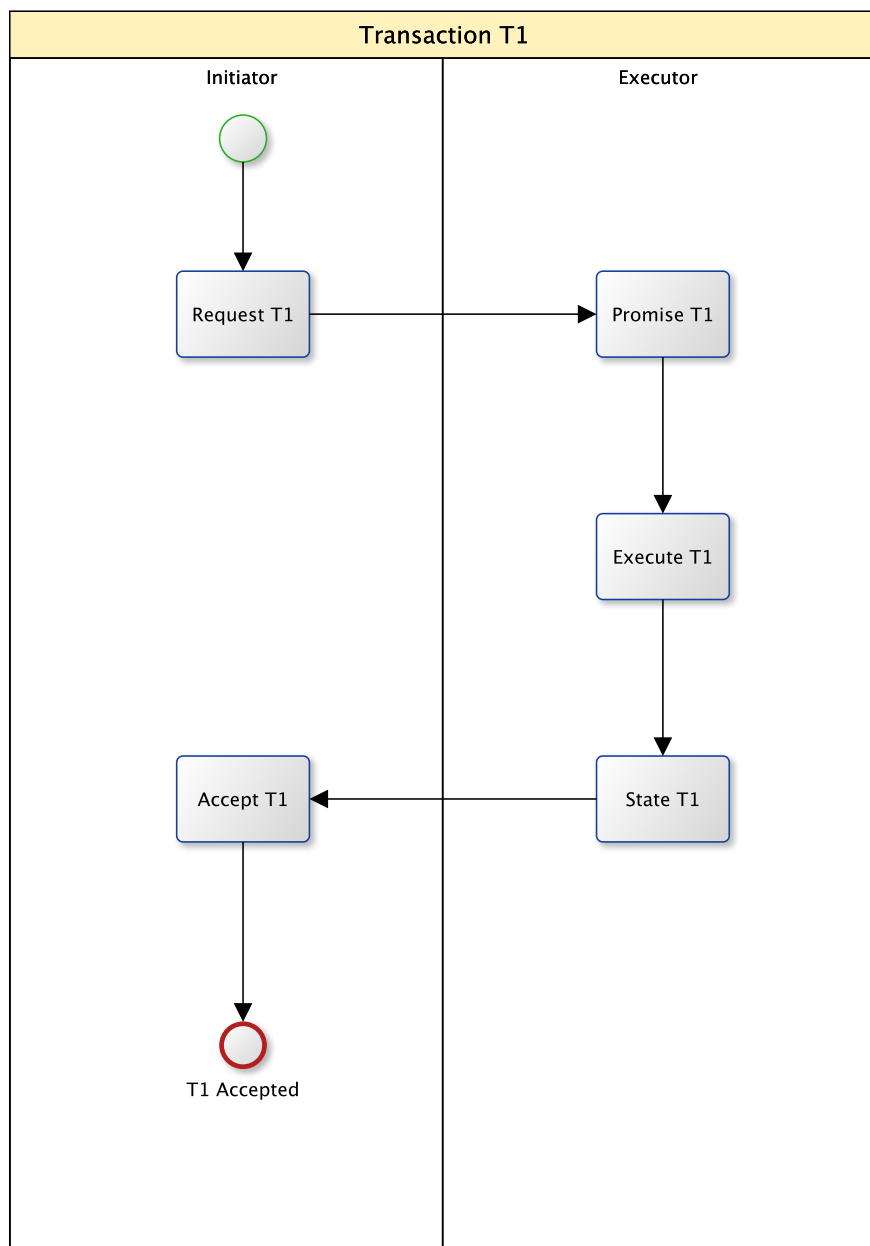
Prvním a nejjednodušším přístupem je vyjádřit transakční vzor čistě pomocí *Úloh* a *Sekvenčních toků*. Úlohy jsou použity pro vyjádření *C-actů* a *P-actů*. Sekvenční tok je použit ve více významech:

- jednak pro určení pořadí provádění jednotlivých *C-actů*,
- jednak pro zadávání *C-factů* do Agendy druhému Actorovi,
- jednak Sekvenční tok vycházející z každé Úlohy reprezentující *C-act* reprezentuje úspěšný vznik korespondujícího *C-factu*.

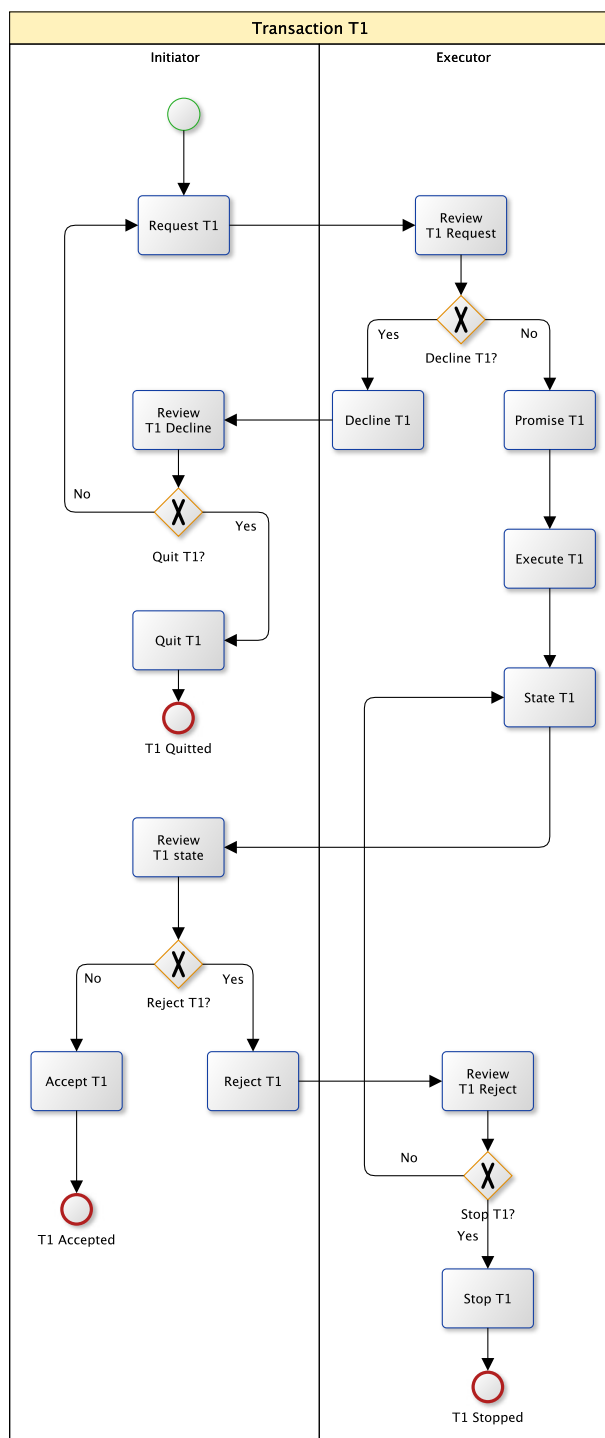
Na obrázku 1 je zobrazen *základní transakční vzor* pomocí Úloh. *Bazén* slouží pro zastřešení celé transakce a *Plavecké dráhy* oddělují Agendy obou zúčastněných Actorů (Iniciátora a Exekutora). Základní transakční vzor bere v potaz pouze „šťastné scénáře“, takže transakce vždy skončí ve stavu *T1 Accepted*. Všechny *C-facty*, ze kterých transakce již nikam dál nepokračuje jsou vyjádřeny BPMN elementem *Konečná událost (End event)*.

Obrázek 2 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*.

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 [?] však oba Actoři mají mít umožněno vědět o vzniku *C-factu*.



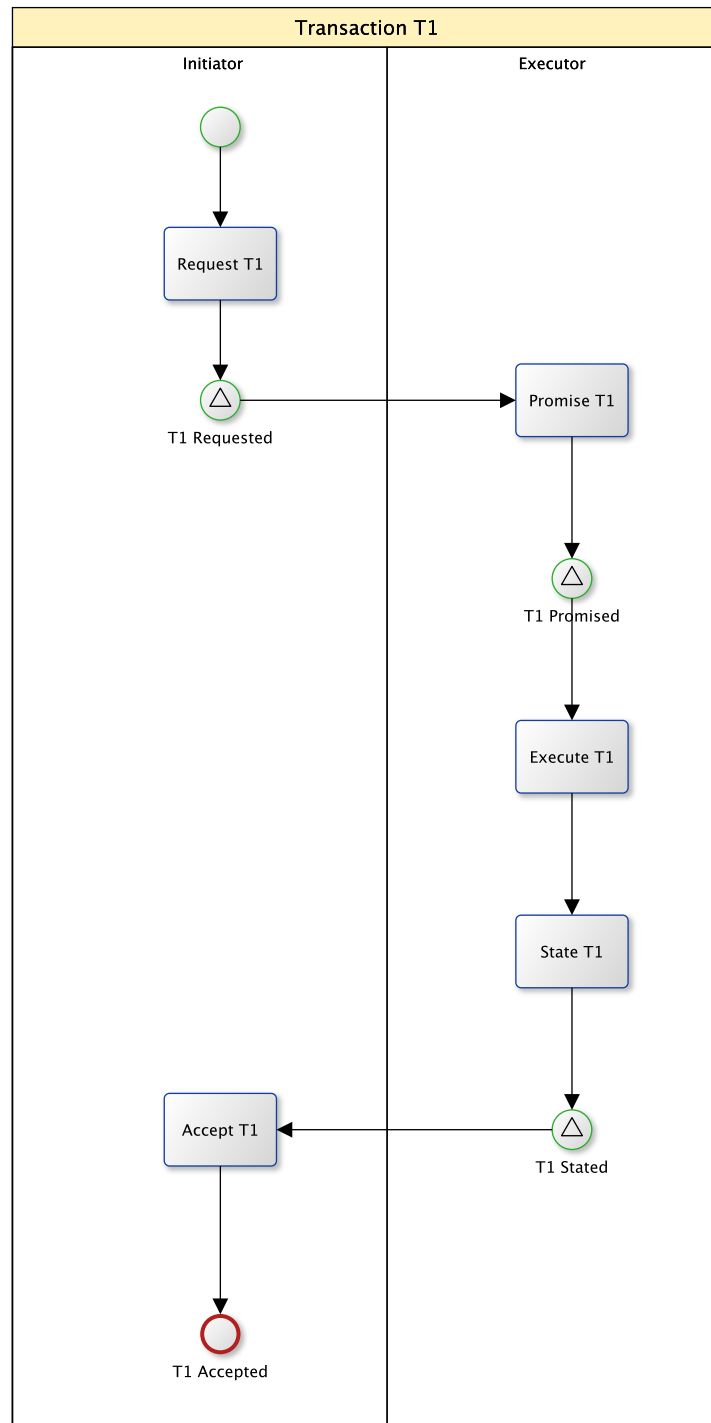
Obrázek 1: Základní transakční vzor pomocí Úloh



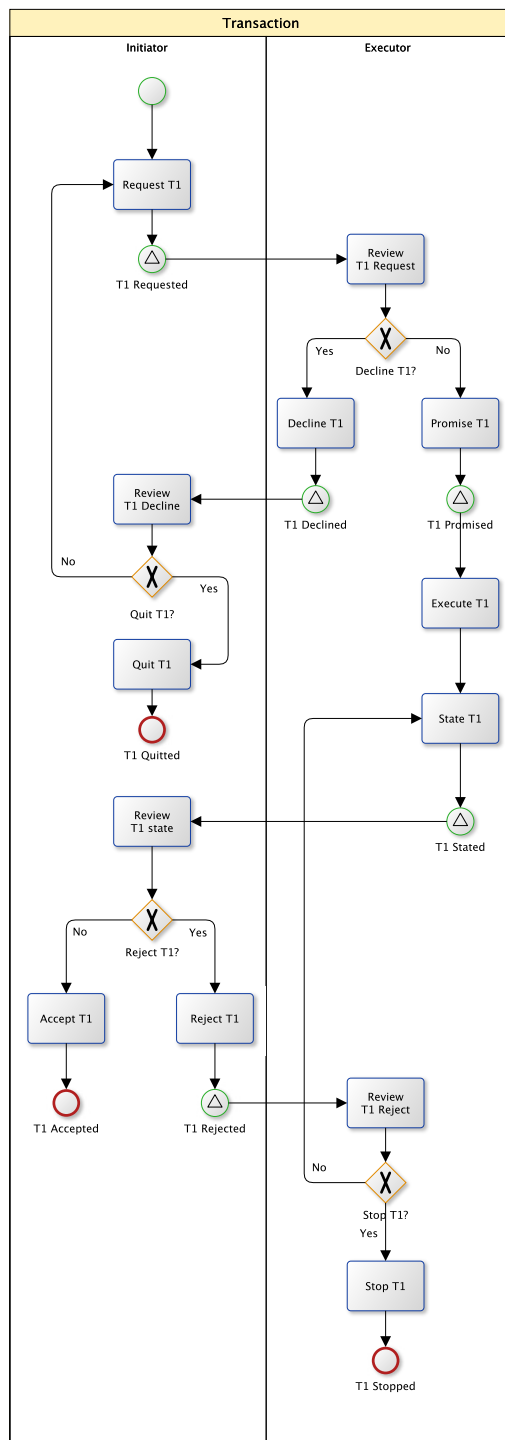
Obrázek 2: Standardní transakční vzor pomocí Úloh

5.2 Vyjádření pomocí Úloh a Signálů

Další možností, jak transakční vzor zobrazit je využití *Úloh* a *Signálů*. Narozdíl od způsobu popsaného v sekci 5.1



Obrázek 3: Základní transakční vzor pomocí Úloh a Signálů



Obrázek 4: Standardní transakční vzor pomocí Úloh a Signálů
13

5.3 Vyjádření pomocí Úloh a Zpráv

V případě, že bychom nepoužili mezistavové zprávy, dostali bychom se do problémů hned v případě, že by se Executor rozhodl pro Decline. V takovém případě Executor vyšle zprávu Initiatorovi a ten může buď transakci zastavit použitím Quit nebo poslat Request znovu. Jenže sekvenční tok v případě Executora je stále v situaci Decline T1 a bez mezistavové zprávy (neboli zprávy o rozhodnutí jak se Initiator rozhodl naložit s T1 Declined) nemůže dále pokračovat. Částečným řešením by v tomto případě mohlo být použít na tomto místě koncový stav T1 Declined a instanci procesu u Executora ukončit. V případě, že by se Initiator rozhodl transakci neukončit a poslal request znovu, což by vytvořilo novou instanci procesu na straně Executora.

Ještě větší problém by však nastal v případě, kdy by Initiator po obdržení T1 State rozhodl tento State odmítnout a provést Reject T1. Executor by následně obdržel zprávu o Rejectu a musel by rozhodnout, zda transakci zastavit provedením Stop nebo provést State znovu. Jenže bez mezistavových zpráv se Initiator nemá šanci dozvědět výsledek tohoto rozhodnutí Executora, které může být buď zastavení transakce provedením Stop nebo nové provedení State. Řešení popsané v předchozím odstavci zde nebude fungovat, protože pokud by se Executor rozhodl provést znovu State, proces na straně Initiatora by již byl ukončen. Jediným řešením jsou tedy mezistavové zprávy, které informují druhého Actora o rozhodnutích v případě, kdy by mohlo dojít k deadlocku.

6 Návrh postupu návrhu BPMN modelů dle zásad DEMO