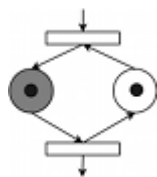


Možnosti prekladu modelov BPMN do Petriho sietí

Savka Andrej · Informačné technológie

06.06.2011



Modelovanie workflow je v súčasnosti témou číslo jeden nielen pri modelovaní biznis procesov, ale i pri vyvíjaní IS, automatizácii a podobne. Súčasné modelovacie nástroje však stále nemajú dostatočné možnosti analýzy hotových modelov. Je to i kvôli tomu, že prvoradým cieľom každého úspešného modelovacieho štandardu je ponúknuť používateľský komfort, prehľadnosť a jednoduchosť.

1. Úvod

To, ale znemožňuje použitie jednoduchej, formalizovanej syntaxe a sémantiky, ktorá by ponúkala možnosti pre matematické formulácie a algoritmy na otestovanie základných funkčných vlastností modelov. Práve preto je z pohľadu praxe veľmi zaujímavá myšlienka, preložiť modely do formátu Petriho sietí (PS), ktoré takéto analýzy vedia ponúknuť. Konkrétne sú v tomto článku rozobraté základné princípy a možnosti prekladu modelov vo formáte BPMN do formátu PS. Najprv je však potrebné oboznámiť sa s PS a štandardom BPMN.

2. Súčasný stav

Na rozdiel od mnohých modelovacích štandardov, pre PS existujú algoritmy na rôzne analýzy, ktoré sú dôležité najmä pri modelovaní veľkých WF systémov. Napríklad: Je daný model v danom bode spustiteľný? Obsahuje proces deadlocky? To sú otázky, na ktoré nám súčasné modelovacie nástroje odpoveď nedajú. Firmy však napriek tomu modelujú svoje biznis procesy v iných štandardoch, IT analytici modelujú WF systémy zväčša v UML 2.0. Tieto štandardy sú totiž oveľa flexibilnejšie a poskytujú rôzne rozšírenia. Pre každého sú však zaujímavé analýzy a transformácie, ktoré PS vedia ponúknuť.

Preto je vhodné zamyslieť sa a navrhnúť prekladacie algoritmy z najpoužívanejších štandardov ako sú BPMN alebo EPC do PS. Po preložení by totiž modely namodelované v moderných CASE nástrojoch mohli byť analyzované na úrovni PS, čo by vyriešilo mnoho krát nezmyselné projekty, na konci ktorých sú nepoužiteľné modely, ale i aplikácie. Vo diplomovom projekte [1] boli navrhnuté možnosti prekladu zo štandardov BPMN a EPC do PS. Takisto bola naprogramovaná jednoduchú Java aplikáciu umožňujúcu túto transformáciu.

Štandardy BPMN a EPC sú momentálne najpoužívanejšie v modelovaní procesov a

používajú ich všetky väčšie modelovacie softvéry. BPMN je obľúbené najmä v Severnej Amerike, EPC zasa v Európe. BPMN ponúka vďaka chybovému toku, bohatšej syntaxe a toku správ lepšie popísané modely. EPC sa svojou syntaxou a sémantikou viac podobá PS. V článku je predstavená základná syntax PS a modelov BPMN. Ďalej je popísané vlastné riešenie pre preklad.

3. Petriho siete

Vytvoril ich Carl Adam Petri v Nemecku v roku 1939 ako 13 ročný na popis chemických procesov. Petriho sieť je diskretný dvoj paritný graf. Skladá sa z nasledovných elementov:

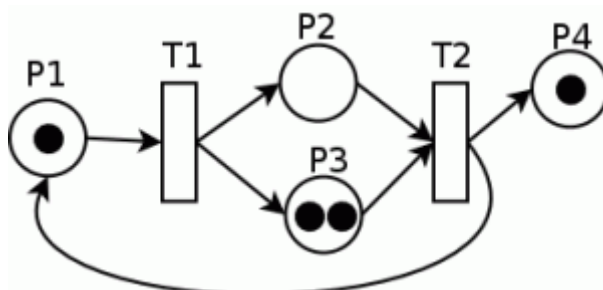
- Miesto (= Place)
- Prechod (= Transition)
- Orientovaná hrana (= Arc)

Hrany sú vždy medzi miestom a prechodom, nikdy nie medzi dvoma miestami alebo medzi dvoma prechodmi. Každé miesto môže obsahovať nenulový kladný počet tokenov. Pod pojmom **značkovanie** sa rozumie rozmiestnenie tokenov v sieti.

Vstupné miesta prechodu - všetky miesta, z ktorých do prechodu ide hrana.

Výstupné miesta prechodu - všetky miesta, do ktorých z prechodu ide hrana.

Prechod je spustiteľný ak je aspoň jeden token vo všetkých vstupných miestach prechodu. Pri spustení sa tieto tokeny presunú do výstupných miest prechodu. PS sú vhodné i na modelovanie paralelných, konkurenčných a distribuovaných systémov. PS poskytujú veľmi dobrý pomer medzi modelovaním a analýzou.



Obrázok 1 - príklad PS

Vysvetlivky k obrázku : Px - miesta, Tx - prechody, Šípky - hrany, Body - tokeny.

3.1. Matematické formalizmy Petriho sietí

Definícia 1 - Sieť

Sieť je trojica $(P; T; F)$, kde P je konečná množina miest, T je konečná množina prechodov spĺňajúca $P \cap T = \emptyset$ a $F \subseteq (P \times T) \cup (T \times P)$ je relácia spojenia. (Miesta a prechody sa taktiež nazývajú elementy siete). Nech trojica (P, T, F) je sieť, a $x \in P \cup T$ je jej element, potom pre-set *x je množina $\{y \in P \cup T \mid (y, x) \in F\}$ a post-set x^* je množina $\{y \in P \cup T \mid (x, y) \in F\}$. Potom množina $X \subseteq P \cup T$ túto notáciu rozširuje nasledovne:

$${}^*X = \bigcup_{x \in X} {}^*x \quad \text{a} \quad X^* = \bigcup_{x \in X} x^*$$

Z technického hľadiska nás zaujímajú len siete, v ktorých každý prechod má neprázdny a konečný pre-set a post-set.

Definícia 2 - P/T sieť (= Place/Transition net)

P/T sieť je štvorica (P,T,F,W) , kde (P,T,F) je sieť a $W : F \rightarrow N^+$ je váhová funkcia.

Definícia 3 - Značkovanie

Značkovanie siete $N = (P,T,F,W)$ je funkcia $m : P \rightarrow N$. Graficky je značkovanie reprezentované čiernymi bodmi - tokenmi v každom mieste.

Definícia 4 - Značkovaná P/T sieť

Značkovaná sieť je dvojica (N, m_0) , kde N je p/t sieť a m_0 je značkovanie, nazývané tiež počiatočné značkovanie siete N .

3.2. Analýzy nad PS

PS vďaka svojej matematickej formalizácii ponúkajú množstvo analýz, ktoré ostatné modelovacie štandardy nemajú. Preloženie WF modelov z BPMN, EPC alebo BPEL do PS otvára možnosti pre ich analýzu. Uvádzam základné analýzy, ktoré PS vedú matematicky dokázať.

Dosiahnuteľnosť značkovania (reachability)

Zisťujeme či je nejaké značkovanie m dosiahnuteľné z počiatočného značkovania m_0 . Značkovanie m sa dá dosiahnuť z počiatočného značkovania m_0 spúšťaním prechodov, preto nás zaujíma, či existuje nejaká postupnosť spúšťania prechodov, ktorá by viedla k danému značkovaniu.

Graf dosiahnuteľnosti

Množinu dosiahnuteľných značkování graficky vyjadruje graf dosiahnuteľnosti. Ten vyjadruje do akého značkovania sa dostaneme po spustení konkrétneho prechodu. Graf by mal pokrývať všetky dosiahnuteľné značkovania z počiatočného značkovania.

Deadlock

Deadlock je značkovanie, v ktorom nie je spustiteľný žiadny prechod, a teda sieť uviazla v mŕtvom bode.

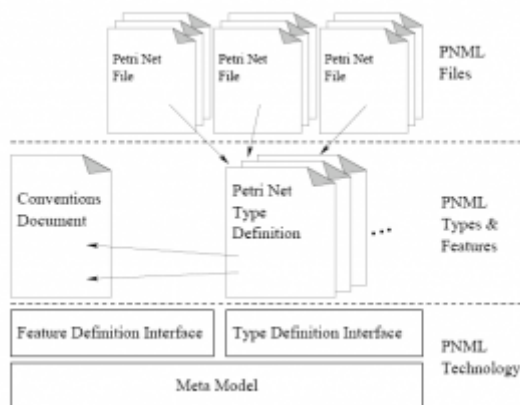
Ohraničenosť

Vo WF systémoch je dôležité pre daný systém určiť všetky stavy, resp. či má konečnú množinu dosiahnuteľných stavov. Táto vlastnosť sa nazýva ohraničenosť PS.

3.3. PNML - Petri Net Markup Language

PNML je výmenný formát (štandard) pre PS založený na XML. Ide o štandard, ktorý umožňuje výmenu namodelovaných PS nezávisle na aplikácii a platforme. PNML ako

každý štandard musí byť flexibilný a kompatibilný. Pod flexibilitou sa rozumie to, že PNML musí podporovať akúkoľvek PS s jej špecifickými vlastnosťami a rozšíreniami. Preto PNML berie PS ako graf s návěstami, kde všetky dodatočné informácie môžu byť uložené v návěstiach. Návěst môže byť priradená ku sieti, k uzlu alebo k hrane. PNML umožňuje definovať typy - Petri net type definition (PNTD), ktoré presne definujú prípustné návěsti pre daný typ PS.



Obrázok 2 - Architektúra PNML

PS sa uloží nakoniec do XML súboru, ktorý je možné otvoriť v akomkoľvek modelovacom nástroji podporujúcom PNML formát. Najdôležitejšie PNML značky sú:

- `<PNML></PNML>` - V tele tohto uzla je celá sieť
- `<TRANSITION></TRANSITION>` - Uzol pre prechody
- `<PLACE></PLACE>` - Uzol pre miesta
- `<ARC></ARC>` - Uzol pre hrany

Každý element PS obsahuje aj informácie o svojej polohe (súradnice X a Y), popis a ostatné relevantné údaje. Informácie o polohe sú v uzle `<graphics>` napríklad: `<graphics><position x="165" y="202"/> </graphics>`.

4. BPMN

BPMN je skratka pre "Business Process Modeling Notation". Je to štandard vydaný v roku 2004 organizáciou Business Process Management Initiative, ktorej členmi sú všetci väčší výrobcovia CASE nástrojov, respektíve nástrojov na modelovanie biznis procesov. Je teda veľmi pravdepodobné, že BPMN sa stane akýmsi priemyslovým štandardom pre modelovanie biznis procesov. V súčasnosti projekt BPMN prevzala organizácia OMG - Open Management Group. Formát BPMN používa i spoločnosť IBM vo svojom Websphere Modeleri (jeden s najpoužívanejších modelovacích nástrojov). Jazyk BPMN ponúka:

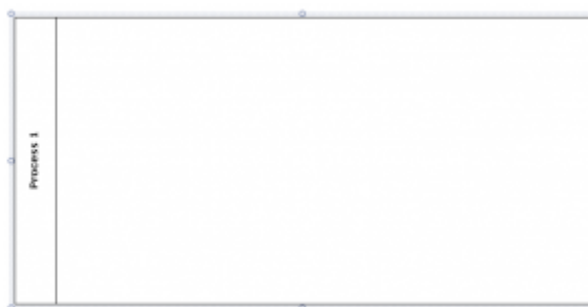
- modelovanie biznis procesov
- modelovanie orchestrácie webových služieb
- možnosť definovať pravidlá, podmienky a výnimky
- možnosť definovať artefakty (vstupy / výstupy)

4.1. Syntax BPMN

I. Plávacie dráhy (Swimlanes)

Bazén (Pool)

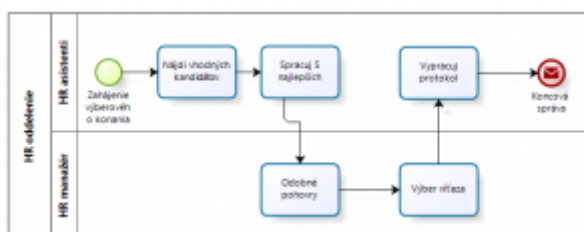
Definuje nadradený proces, ktorý je v rámci BPMN rozkreslený. Medzi bazénmi je komunikácia znázornená formou posielania správ. Odporúča sa, aby bazén predstavoval jeden kľúčový proces. Bazén reprezentuje účastníkov v rámci procesu, napríklad HR oddelenie (v bazéne sú aktivity procesu ktoré vykonáva HR oddelenie). Procesné časti v rôznych bazénoch spolu komunikujú pomocou toku správ.



Obrázok 3 - Bazén pre Proces 1

Dráha (Lane)

Dráhy rozčleňujú bazén na menšie časti. Napríklad HR oddelenie môžeme v rámci procesu rozdeliť na časti ktoré vykonáva HR manažér a HR asistenti.



Obrázok 4 - Dráhy vrámci bazénu

II. Objekty toku (Flow Objects)

Sú modelovacie objekty, ktoré sa pripájajú k sekvenčnému toku. Patria k nim udalosti, úlohy a brány.

Udalosť (Event)

Reprezentuje udalosť, ktorá priamo ovplyvňuje chod procesu. Je to veľmi dôležitý objekt toku.

Odchyťavanie udalosti (Catching Events) - špecifická udalosť sa môže spustiť, po splnení určitých podmienok alebo situácií (catch = odchytenie týchto udalostí).

Vysielanie udalostí (Throwing Events) - udalosť môže „vyhodiť“, vyslať spúšťač inej udalosti. Tá je aktivovaná na základe tohto spúšťača (napr. správa, časovač a podobne). Rozlišujeme 3 základné typy udalostí:

Začiatočná - zahajuje reťazec úloh v rámci procesu, ktorý je znázorňovaný na procesnej mape. Každý proces má aspoň jednu začiatočnú udalosť. Začiatočná udalosť nemá žiadne vstupy sekvenčného toku a nemôže byť výstupom toku správ. Poznáme

päť druhova začiatočných udalostí podľa spúšťača udalosti:

- **Žiadna (None)** - používa sa ako začiatok podprocesu alebo v prípade keď začiatok nie je identifikovateľný.
- **Správa (Message)** - spúšťačom udalosti je správa.
- **Časovač (Timer)** - udalosť sa spustí v určitý deň alebo cyklus (napr. desiaty deň v mesiaci a podobne).
- **Podmienka (Conditional)** - spúšťačom udalosti je splnená podmienka (napríklad počet zákazníkov > 10 a podobne).
- **Signál (Signal)** - udalosť spúšťa signál vyslaný z iného procesu, niekedy viac procesov môže byť spustených tým istým signálom.
- **Násobná (Multiple)** - spúšťačom procesu sú dva alebo viac spúšťačov. Stačí jeden na spustenie procesu.



Obrázok 5 - Začiatočné udalosti: žiadna, správa, časovač, podmienka, signál, násobná

Priebežná - najčastejšie sa používa na prepojenie viacerých procesných máp. Priebežná udalosť neštartuje ani neukončuje proces. Používa sa často na vyhodenie alebo chytenie udalosti ako príjem alebo odoslanie správy a podobne. Reprezentuje určitý časový interval počas behu procesu, kedy sa niečo udeje (napríklad príde správa). Poznáme osem druhov priebežných udalostí:

- **Žiadna (None)** - používa sa v prípade, že typ udalosti nie je definovaný. Takáto udalosť nikdy nerieši tok výnimiek a znamená len, že v priebehu procesu sa nachádza určitá udalosť, kedy dochádza k určitým zmenám.
- **Správa (Message)** - Používa sa v prípade odoslania alebo prijímania správy. Udalosť je spustená prijatím správy alebo správou odosiela.
- **Časovač (Timer)** - udalosť sa spustí v určitý deň alebo cyklus (napr. desiaty deň v mesiaci a podobne). Často sa používa na zastavenie procesu na určitý čas.
- **Chyba (Error)** - vždy sa umiestňuje na hranu aktivity. Pri vopred definovaných podmienkach spustí tok výnimky.
- **Zrušenie (Cancel)** - používa sa na zrušenie procesu.
- **Kompenzácia (Compensation)** - vráti proces späť do stavu pred vykonaním príslušnej aktivity. Musí byť vždy umiestnený na hranu aktivity.
- **Podmienka (Conditional)** - v prípade, že je splnená podmienka udalosť ovplyvní proces.
- **Linka (Link)** - používa sa na prepojenie dvoch častí procesu. Obe aktivity prepojenia majú priebežnú udalosť typu Linka, ktoré navzájom korešponujú.
- **Signál (Signal)** - udalosť sa používa na posielanie a prijímanie signálov počas behu procesu.
- **Násobná (Multiple)** - používa sa v prípade, keď existuje viacero spúšťačov pre danú udalosť.



Obrázok 6 - Priebežné udalosti: žiadna, správa, časovač, kompenzácia, podmienka, linka, signál, násobná udalosť

Koncová - každý proces musí byť ukončený aspoň jednou koncovou udalosťou. Koncová udalosť reprezentuje miesto, kde proces končí. Po spustení koncovej udalosti v podprocese sa ďalej pokračuje v chode bežiaceho hlavného procesu. Rozlišujeme osem druhov koncových udalostí.

- **Žiadna (None)** - používa sa v prípade, že sa nehodí žiadny iný typ, respektíve keď modeler nepotrebuje bližšie špecifikovať udalosť.
- **Správa (Message)** - na konci procesu sa účastníkom posiela správa.
- **Chyba (Error)** - táto udalosť ukončuje proces určitou chybou. Táto chyba bola vyhodnená priebežnou udalosťou s tým istým kódom chyby. Pokiaľ neexistuje priebežná udalosť, ktorá vyvolala ukončenie procesu chybou, tak táto koncová udalosť sama o sebe nemá význam.
- **Zrušenie (Cancel)** - používa sa pri podprocesoch, v prípade vyvolania chyby sa vykonávanie podprocesu zruší.
- **Kompenzácia (Compensation)** - aby bola aktivita obnovená, resp. aby sa proces vrátil pred jej vykonanie, musí byť k jej okraju pripojená priebežná udalosť kompenzácie.
- **Signál (Signal)** - udalosť sa používa na posielanie signálov pri ukončení procesu.
- **Ukončenie (Terminate)** - používa sa na okamžité ukončenie procesu, okamžite sa ukončia všetky inštancie procesu. Pri takomto ukončení ani výnimky alebo správy nie sú dokončené.
- **Násobná (Multiple)** - používa sa v prípade keď existuje viacero dôsledkov ukončenia procesu, odosiela sa viacero správ, signálov a podobne.



Obrázok 7 - Koncové udalosti - žiadna, správa, chyba, zrušenie, obnovenie, signál, ukončenie, násobná

Činnosť (Task)

Reprezentuje predpísanú činnosť, ktorá má byť vykonaná. Poznáme dva základné typy

- **Jednoduchá (Úloha)** - reprezentuje jednoduchú úlohu, ktorá sa vykonáva v rámci procesu.
- **Zložená (Podproces)** - vložený podproces s náhľadom priamo v diagrame alebo nezávisle uložený v inom.



Obrázok 8 - Úloha a podproces

Brána (Gateway)

Sú to miesta vetvenia alebo zbiehania chodu procesu. Poznáme štyri základné typy brán.

- **Synchrónna - XOR** - beh pokračuje práve jednou vetvou.

- **Asynchrónna (Paralelná) - OR** - beh pokračuje jednou alebo všetkými vetvami.
- **Asynchrónna (Paralelná) - AND** - beh pokračuje všetkými vetvami.
- **Komplexná** - používa sa v prípade, že hore uvedené nestačia alebo chceme zlúčiť viac vetvení. Používa sa v prípade neštandardných vetvení.



Obrázok 9 - XOR, AND, OR a Komplexná brána

III. Konektory

Sekvenčný tok (Sequence Flow)

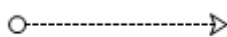
Vyjadruje následnosť procesných prvkov. Ak je výstupným prvkom brána, potom obsahuje i podmienku prechodu.



Obrázok 10 - Sekvenčný tok

Tok správy (Message Flow)

Popisuje správy, ktoré majú priamy vplyv na chod procesu.

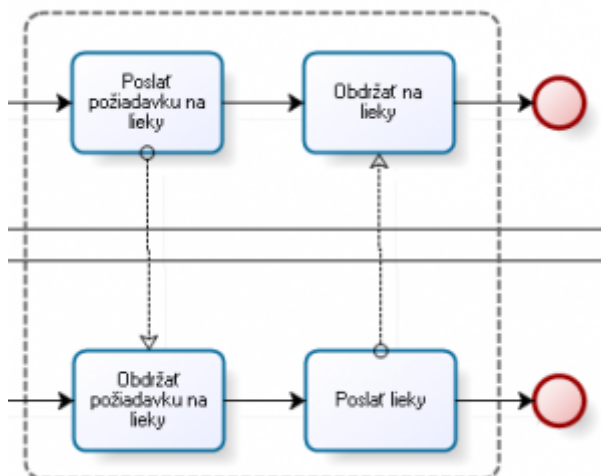


Obrázok 11 - Tok správy

IV. Artefakty

Používajú sa na zobrazenie produktov IS. Väčšinou sú to vstupné a výstupné dokumenty. Vstupné a výstupné artefakty sú prepojené asociáciou týchto artefaktov.

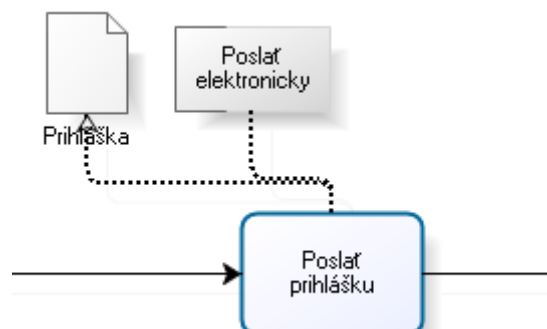
Skupina (Group) - slúžia na zvýraznenie alebo zoskupenie určitých častí diagramu bez zvýšenia nárokov na výkon (na rozdiel od podprocesu). Môže sa tiež použiť na kategorizáciu elementov pre výkazníctvo (reporting).



Obrázok 12 - Skupina

Textové anotácie (Text Annotations) - prostriedok ako vyjadriť dodatočné informácie o procese.

Dátové objekty (Data Objects) - artefakty, ktoré ukazujú, ako sú používané dáta a dokumenty v rámci procesu. Môžu definovať vstupy a výstupy aktivít. Môžu im byť priradené statusy, ktoré definujú, že dokument má byť zmenený alebo aktualizovaný.



Obrázok 13 - Dátový objekt (vľavo) a textová anotácia

Druhá časť článku bude zverejnená o týžden.

Ing. Andrej Savka, Absolvent FEI STU Bratislava, Aplikovaná Informatika
