

Transformace BORM-Petriho sít' s využitím supervize

Ing. Josef Moravec¹, Ing. Martin Papík, PhD.^{1,2}

¹Katedra informačního inženýrství, PEF, Česká zemědělská univerzita v Praze, Kamýcká 129,
165 21 Praha 6 - Suchbátka
jmoravec@pef.czu.cz

²UTIA AVČR v.v.i, Pod vodárenskou věží 4,
182 08 Praha 8
papik@utia.cas.cz

Abstrakt. Pro efektivní fungování podnikových procesů je nebytné jejich pečlivé zdokumentování a případná optimalizace. Způsobů jak proces popsat je značné množství. Jako relativně jednoduchý a přitom účinný přístup se jeví použití metodiky BORM, který je implementován v nástroji CraftCase. Bohužel tento nástroj nenabízí tak komplexní možnosti simulace procesu, jakých je často zapotřebí. Možností jak tento nedostatek eliminovat je použití Petriho sítí pro simulaci procesů. Nejprve je nutné proces zachycený v BORMu transformovat na Petriho sít' . Pravidly transformace, omezeními jak BORMu tak i Petriho sítí a v neposlední řadě konstrukcí supervizoru Petriho sítě se zabývá tento příspěvek.

Klíčová slova: BORM, CraftCase, Petriho sít', Supervizor, transformace

1 Úvod

Důvodem, proč stále více společností investuje nemalé prostředky do analýzy, reengineeringu nebo auditu procesů ve firmě probíhajících, je především zvyšování efektivity práce a snižování provozních nákladů. Dobře navržené a udržované procesy jsou základním předpokladem pro bezproblémové fungování podniku.

Procesy jsou také předmětem normalizace. Normy upravují a definují kvalitu procesů. Aplikace norem přináší řadu výhod. Podmínkou pro splnění norem je detailní dokumentace a popis průběhu jednotlivých procesů v organizaci. Takto zdokumentované procesy se dobře udržují a jednoduše provádějí. Dobře normalizované procesy snižují jak časovou tak i finanční náročnost na provoz. Jednodušší procesy jsou lépe proveditelné a často spolehlivější.

Abychom mohli na proces aplikovat zvolenou normu, je nutné nejprve proces zachytit. Důležitá je proto volba vhodného přístupu a použitá metodika.

2 Cíle příspěvku

Hlavním cílem příspěvku navržení supervizoru pro Petriho sít' tak, abychom mohli simulovat proces, jehož průběh a chování nelze detailně simulovat pomocí nástroje

CraftCASE. Proces byl zachycen metodu BORM a následně transformován na Petriho síť.

Dalším cílem příspěvku je definování pravidel transformace základních struktur z procesního diagramu do Petriho sítě a vytipování takových struktur, které nelze jednoduše realizovat pomocí základních konstrukcí Petriho sítě.

Dílními cíly příspěvku je poukázat na výhody a nevýhody Petriho sítí a BORMu jako nástrojů pro modelování procesů a přinést argumenty pro použití metody BORM v kombinaci s Petriho sítěmi jako komplexního nástroje pro modelování a simulaci procesů.

3 BORM a Petriho síť

Existuje řada metodik zaměřených na modelování procesů. Velmi používanou je ARIS (Architecture of Integrated Information Systems) nebo UML (Unified Modeling Language). ARIS je vhodný pro velké projekty, protože poskytuje širokou paletu nástrojů pro modelování. Bohužel je velmi nákladný. Použití metodiky ULM je oproti tomu zadarmo, ale neposkytuje takové možnosti jako ARIS. Právě proto byla zvolena metoda BORM respektive nástroj CraftCASE, který je relativně méně nákladný na pořízení, ale disponuje bohatou sadou nástrojů jak pro návrh tak i pro simulaci průběhu procesu.

BORM je procesně orientovaný právě jako ARIS. Využívá notaci UML, která je jednoduchá a široce dobře přijímána. Hlavními výhodami metody BORM pro modelování procesů jsou:

- je velmi dobře pochopitelná i neodborníky
- srozumitelnost procesních modelů a komplexní pohled na modelovaný proces
- velmi málo omezení při návrhu procesů
- přehledný i při modelování rozsáhlých procesů

BORM má také své nevýhody. V souvislosti s modelováním a reengineeringem procesů jsou to především tyto:

- BORM je neformální metoda – dovoluje návrh i nelogických konstrukcí, což je nevýhoda při transformaci procesního modelu na jiné modely. Nutno dodat, že při dodržení zásad rozumného návrhu toto omezení odpadá.
- Neumožňuje detailní pohled na průběh procesu
- Nástroj CraftCASE neumožňuje stanovit složitější podmínky pro řízení průběhu procesu nebo řídit průběh jednoho procesu na základě stavu procesu jiného

Nevýhody se plně projeví při návrhu složitých nebo provázaných procesů. Obtížně řešitelné je modelování procesu se samoregulací nebo s velkým množstvím na sebe navázaných podmínek pro větvení nebo řízení procesu. Jelikož při simulaci musí splnění jednotlivých podmínek kontrolovat člověk, je obtížné složité procesy uřídit a odsimulovat jejich správnou funkci.

Efektivním způsobem jak eliminovat nevýhody spojené s použitím metody BORM je využití možností Petriho sítě. Jedním z prvních projektů, který využívá Petriho síť pro modelování procesů je již zmiňovaný ARIS.

Petriho síť jsou často zmiňovány v souvislosti s paralelními a distribuovanými systémy. Další oblastí je automatizace průmyslu nebo návrh hardwaru a softwaru. Petriho síť, jak ukazuje právě ARIS, je možno použít také pro modelování procesů.

Petriho síť umožňují při modelování procesů zejména:

- detailní pohled dílčí stavy procesu
- stanovit i složité podmínky pro přechody do jednotlivých stavů procesů pomocí soustav rovnic, proces lze mnohem přesněji řídit
- detailní pohled na simulaci procesu

Nevýhodami jsou především:

- nepřehlednost při modelování složitých procesů
- jedná se o formální nástroj. Při tvorbě modelu je třeba dodržovat pevně daná pravidla
- obtížná pochopitelnost pro neoborníky
- neposkytuje komplexní pohled na modelovaný proces. Proces je rozdělen do dílčích stavů.

Proč volit tak komplikovaný postup pro modelování a ověření správnosti modelu procesu? Odpověď není složitá. Spojením „neformálního“ nástroje podporujícího BORM a „formálních“ Petriho sítí získáme nástroj, který:

- je dobře pochopitelný a srozumitelný i pro neoborníky
- poskytuje komplexní pohled i na velmi složitý modelovaný proces
- velmi málo omezuje při modelování procesu
- umožňuje detailní pohled dílčí stavy procesu
- dovoluje stanovit i složité podmínky pro přechody do jednotlivých stavů procesů pomocí soustav rovnic
- při použití Petriho sítí k simulaci minimalizuje možnost výskytu nelogických struktur

Verbální popis procesu lze lépe zachytit pomocí procesních diagramů metody BORM. Naproti tomu Petriho síť nabízejí silný nástroj pro formální ověření navrženého procesu. To je důvod, proč používat tyto nástroje společně.

4 Stavba procesu

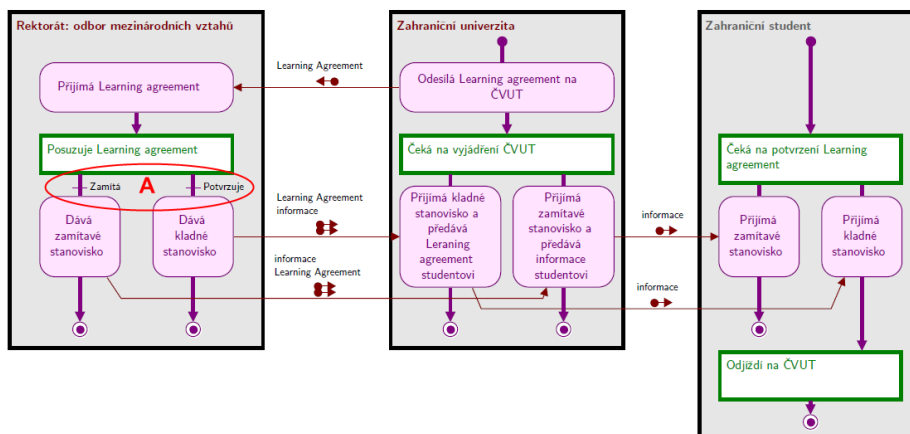
Jednoduše řečeno je proces posloupnost vzájemně propojených činností a stavů mající nějaký výsledek nebo konečný stav. Když se zaměříme na prvky, ze kterých se může proces skládat a jejich možné kombinace, zjistíme že máme k dispozici jen několik málo základních konstrukcí. Proces se může skládat v zásadě ze čtyř různých stavebních kamenů, kterými jsou:

- *posloupnost* je charakteristická tím, že jednotlivé činnosti na sebe lineárně navazují. Tedy posloupnost Činnost 1 poté Činnost 2 a tak dále.
- *paralelní činnost* pro niž je charakteristické rozvětvení procesu na 2 a více vláken, která probíhají paralelně a nemusejí být na sobě vzájemně závislé. Na jednu činnost tedy navazují 2 a více činností, které se vykonávají souběžně.
- *větvení* je podobné paralelní činnosti s tím rozdílem, že u větvení zvolíme pouze některou větev procesu respektive větve procesu, které se budou vykonávat.
- *opakování neboli iterace* je sled činností, který probíhá opakovaně, dokud není splněna podmínka. Při splnění podmínky je opakování ukončeno.

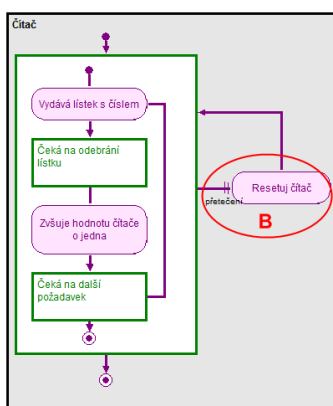
5 Reprezentace procesu v BORM

Proces je v BORMu reprezentován procesním diagramem. Procesní diagram lze definovat jako množinu participantů, stavů, činností a datových toků. Každý diagram obsahuje alespoň jednoho participanta. Každý participant obsahuje alespoň jednu činnost. Většinou participant obsahuje více stavů a činností. Existují zde pravidla pro vzájemnou komunikaci prvků procesu. Činnost může být spojena pouze se stavem v rámci jednoho participanta. Naopak při vzájemné komunikaci dvou účastníků procesu zpravidla dochází k propojení dvou činností. Nelze propojit dva stavy navzájem. Ilustrativní procesní diagram na Obr. 1. ukazuje základní prvky procesního diagramu. Základní prvky procesního diagramu jsou:

- *Participant* – reprezentuje účastníka procesu a je zobrazen jako černý rámeček
- *Stav* – každý participant může v čase měnit svůj stav na základě vykonaných činností. Stav je reprezentován zeleným obdélníkem
- *Činnost nebo aktivita* – představuje činnosti vykonávané participantem. Slouží pro přechod z jednoho stavu do jiného. Pro aktivity je použit fialový obdélník se zaoblenými rohy.
- *Komunikace* – propojují stavy a činnosti v rámci jednoho participanta nebo vyjadřují vzájemnou komunikaci více participantů. Komunikace jsou znázorněny šipkou. Součástí komunikace mohou být data, která jsou znázorněna menší šipkou spolu s identifikací dat.



Obr. 1. Procesní diagram – Příprava příjezdu zahraničního studenta v rámci programu Erasmus



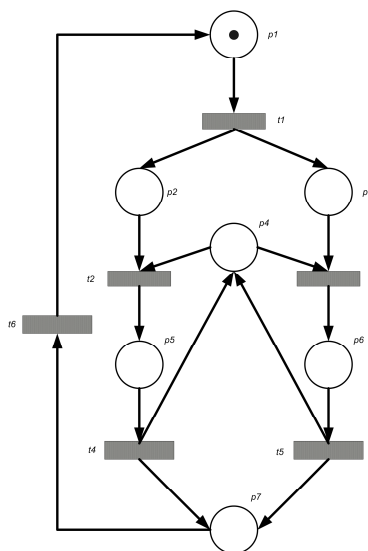
Obr. 2. Ukázka použití výjimky u procesu v CraftCase

Na obrázcích Obr. 1. a Obr. 2. jsou vyznačena místa související se řízením průběhu procesu. Místo označené písmenem A znázorňuje podmínku. Podmínkou se stanoví pravidlo při jehož splnění bude komunikace provedena a proces přejde do jiného stavu. Místo označené písmenem B je výjimka. Výjimky slouží podobně jako podmínky pro přechod účastníka do jiného stavu. Výjimku je možné spustit ve kterémkoli kroku procesu na který je výjimka aplikovaná. Po jejím spuštění proces přejde do předem zvoleného stavu.

6 Re prezentace procesu pomocí Petriho sítě

Petriho síť je orientovaný ohodnocený bipartitní graf, který se skládá z míst, přechodů a hran. Při tvorbě Petriho sítě narážíme na omezení. Místo může být spojeno pouze s

přechodem a naopak přechod může být spojen pouze s místem. Nelze propojit přímo dvě místa ani dva přechody. Ukázka procesu realizovaného Petriho sítí je na obrázku Obr. 3.



Obr. 3. Ukázka reprezentace procesu Petriho sítí

7 Jednoznačná transformace

Hlavním přínosem tohoto příspěvku je právě definování pravidel transformace základních struktur z procesního diagramu do Petriho sítě a vytipování takových struktur, které nelze jednoduše realizovat pomocí základních konstrukcí.

Vzhledem k podobnosti procesních diagramů a Petriho sítí se nemusí jejich vzájemná transformace jevit jako obtížný úkol. Problém je právě ve volnosti, která je poskytnuta při návrhu procesu metodou BORM. Jelikož se zde mohou vyskytnout nelogické konstrukce, definujeme nejprve ekvivalentní struktury k základním konstrukcím Petriho sítí. Následně vytipujeme konstrukce v procesním diagramu BORMu, které nelze jednoduše realizovat základními konstrukcemi Petriho sítí. Pro vytipované konstrukce navrhneme možná řešení.

Procesní diagram v BORMu má oproti Petriho sítí následující odlišnosti:

- BORM definuje účastníky procesu neboli participanty. Petriho sítě nedisponují žádnou ekvivalentní konstrukcí. Petriho síť představuje pouze jeden komplexní model. Každý participant tedy představuje část tohoto modelu. Z hlediska Petriho sítě je každý participant subprocess.

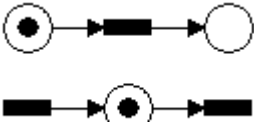
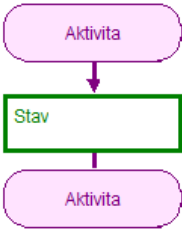
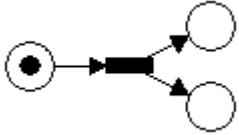
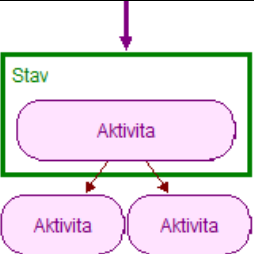
- BORM má vlastní symboly pro počátek a konec procesu. Petriho síť definují pouze počáteční značení, tedy nějaký zvolený stav modelu. Konec procesu, tedy konečný stav může být opět libovolný zvolený stav modelu.

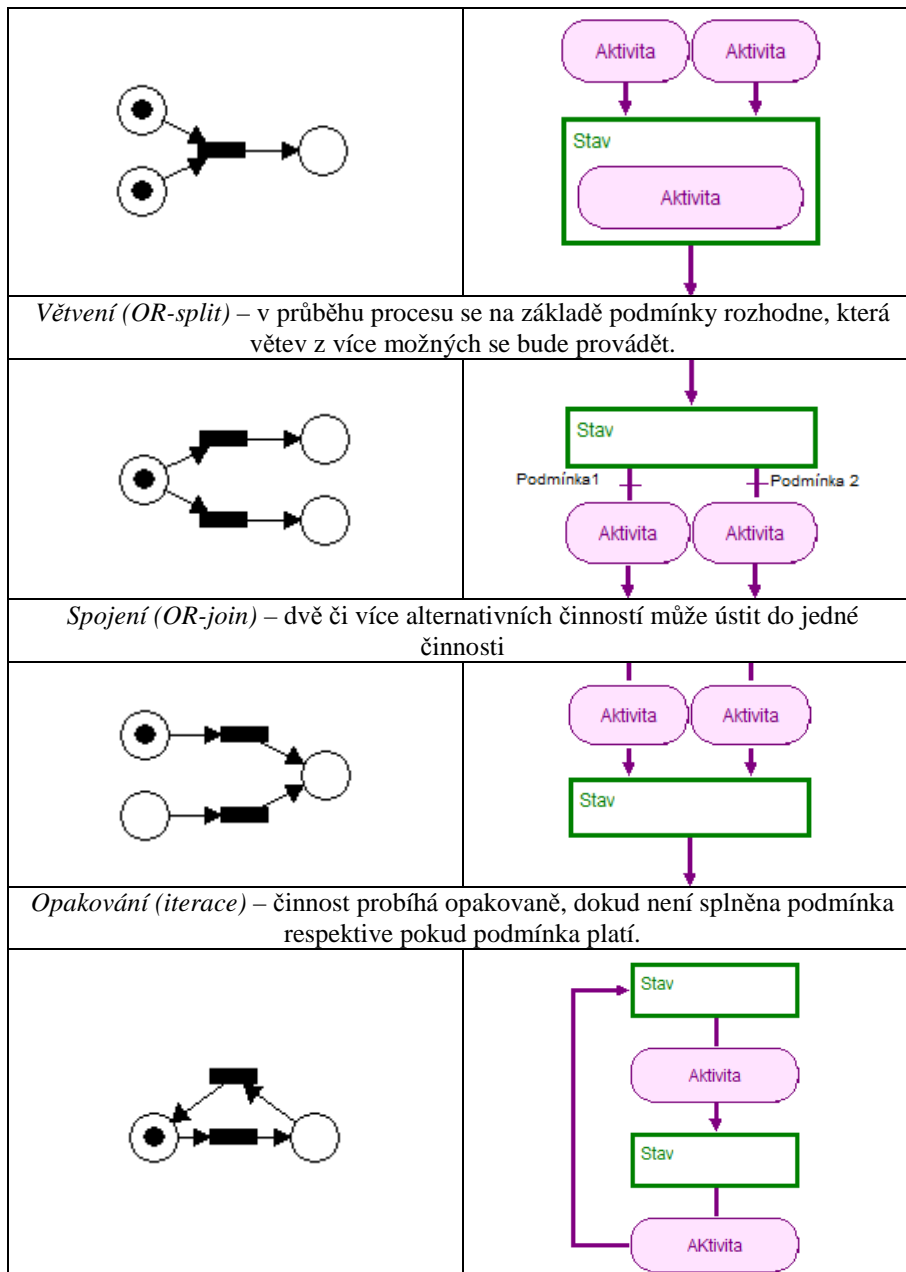
Petriho síť může vytvářet tyto základní konstrukce:

- *Sekvence (posloupnost)*, kde jednotlivé činnosti na sebe lineárně navazují.
- *Paralelní směřování (AND-split)*, na jednu činnost navazují dvě či více činností, které pak probíhají současně.
- *Synchronizace (AND-join)*, dvě či více paralelně probíhajících činností ústí do jedné činnosti.
- *Větvení (OR-split)*, v průběhu procesu se na základě podmínky rozhodne, která větev z více možných se bude provádět.
- *Spojení (OR-join)*, dvě či více alternativních činností může ústít do jedné činnosti.
- *Opakování (iterace)*, činnost probíhá opakovaně, dokud není splněna podmínka respektive pokud podmínka platí.

Převědeme-li tyto základní konstrukce do procesního diagramu podle metodiky BORM získáme následující ekvivalentní, taktéž základní konstrukce:

Tabulka 1. Transformace základních konstrukcí Petriho sítě do BORMu

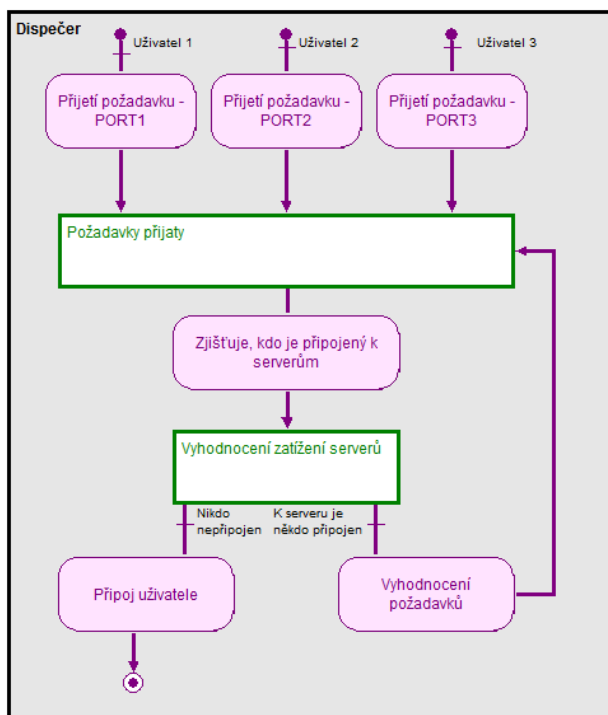
<i>Sekvence (posloupnost)</i> – jednotlivé činnosti na sebe lineárně navazují	
	
<i>Paralelní směřování (AND-split)</i> – na jednu činnost navazují dvě či více činností, které pak probíhají současně.	
	
<i>Synchronizace (AND-join)</i> – dvě či více paralelně probíhajících činností ústí do jedné činnosti.	



Problém nastane, pokud potřebujeme realizovat složitější nebo nestandardní konstrukci. Problém není v samotné konstrukci, ale hlavně v simulaci procesu. Vemněme jednoduchý příklad, kdy je dán systém zabezpečující rovnoměrné zatížení serverů. Pro zjednodušení budeme předpokládat, že máme k dispozici dva servery

a tři uživatele, kteří se snaží připojit k aplikaci, která je na serverech umístěna. Omezující podmínkou je, že v jednu chvíli může být k jednomu serveru připojen pouze jeden uživatel. Ve skutečnosti by byl počet uživatelů mnohonásobně vyšší a taktéž i dovolené vytížení každého serveru by bylo vyšší. Popisovaný systém je zachycen na obrázku Obr. 4., což je procesní digram metodiky BORM. Pro modelování a simulaci byl použit nástroj CraftCase.

Nástroj CraftCASE bohužel neumožňuje podrobnou specifikaci podmínek pro takové případy. Simulace probíhá tak, že musíme sami hlídat, aby nebylo připojeno k jednomu serveru více uživatelů zároveň. Simulace takto relativně triviálních podmínek je tedy značně obtížná. Řešením je použít pro simulaci právě Petriho síť, která umožňuje velmi podrobně stanovit podmínky přechodu z jednoho stavu do druhého.



Obr. 4. Systém zabezpečující rovnoměrné vytížení serverů

Jakým způsobem tedy procesní diagram na obrázku Obr. 4. převést na Petriho síť? Je zřejmé, že nevystačíme pouze se základními konstrukcemi. Jednou z možností je vytvoření Supervizoru, který bude proces sledovat a bude řídit jeho průběh tak aby byla splněna omezující podmínka.

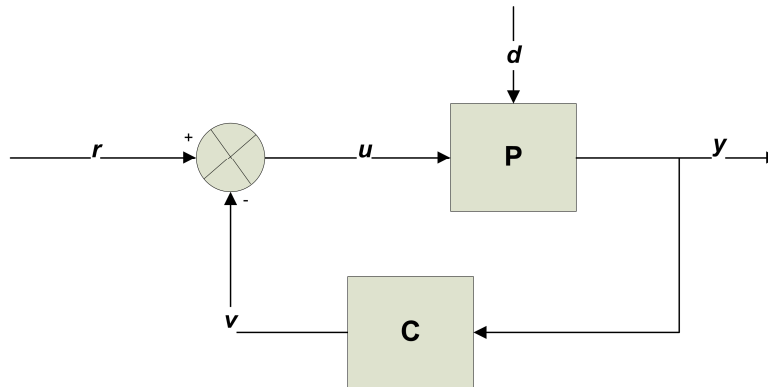
8 Supervizor a Petriho síť

Nežádoucí vlastnosti systému lze měnit řízením, tak aby systém s řízením žádoucím vlastnostem vyhovoval. Základním principem řízení je zde zpětnovazební řízení. Jde zde o zpětné zavedení výstupu na vstup systému. To umožňuje kompenzovat vliv poruch a odchylek výstupu v případě negativní zpětné vazby. Zpětná vazba je v praxi základním principem používaným zejména v regulační technice.

Základní strukturu systému se zpětnou vazbou tvoří dvě komponenty (části):

- *Řízená soustava*, kterou označujeme P . Vstupem do soustavy je řízení u a výstupem soustavy je signál y . Dále u technických aplikací uvažujeme vnější signál poruchy d .
- *Regulátor (řídící člen)*, který označujeme C . Vstupem do regulátoru je výstup ze systému y a výstupem je signál v .

Výše uvedený popis základního systému se zpětnou vazbou je znázorněn na obrázku Obr. 5. Vstup do systému P je $u = r - v$, kde r je řídicí signál. Základní úlohou je zde návrh regulátoru C tak, aby byly splněny požadavky kladené na zpětnovazební systém. V případě dynamických systémů jsou to požadavky na stabilitu, nulovou odchylku $r - y$ pro čas $t \rightarrow \infty$ a dynamické požadavky na chování vstup – výstup.



Obr. 5. Princip systému se zpětnou vazbou

Supervizor vychází z původní myšlenky regulátoru, avšak v poněkud modifikované podobě. Supervizor pro systém popsáný Petriho sítí monitoruje a řídí tuto síť dle zadaných požadavků. Supervizor je v tomto případě navržen jako další Petriho síť, které je spojená s Petriho sítí daného systému. Rozdíl mezi supervizorem a regulátorem ve výše uvedeném klasickém smyslu je v tom, že regulátor určuje vstupní signály do systému zatímco supervizor pouze omezuje množinu vstupních signálů do systému. Obecně je tato množina omezena dynamicky v závislosti na pozorování systému. Supervizor by měl zakázat pouze vstupy nevyhnutelně vedoucí k porušení zadané specifikace. Tato specifikace se může týkat dosažitelného značení

nebo přípustných posloupností přechodů. Supervizor zakomponovaný do původní Petriho sítě je rovněž Petriho sítí a celý systém se nazývá zpětnovazebním systémem.

Supervizor je řídicí člen, který ponechává volnost vývoje řízené soustavy a je aktivován pouze k zabránění nežádoucího chování řízené soustavy. Řídicí členy navržené pro řízení invariant jsou supervizory. Supervizor Petriho sítě je implementován ve formě řídicích míst připojených k řízené soustavě (Petriho sítí). Návrh supervizoru založeného na invariantách Petriho sítě s plně říditelnými a pozorovatelnými přechody vede na bezproblémové vztahy procesu návrhu supervizoru.

Uplatněná specifikace pro návrh řízení supervizorem mají tvar omezení lineární nerovnosti

$$Lmp \leq b,$$

kde $mp \in N^{np}$ je vektor značení Petriho sítě, $L \in N^{nc \times np}$, $b \in N^{nc}$ a nc je počet omezení.

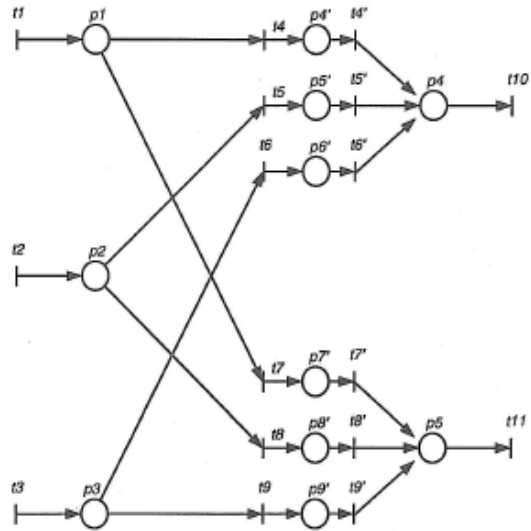
Metoda popisovaná v textu předpokládá, aby proces, který má být řízen byl modelován Petriho sítí. Řízení tohoto procesu bude navrženo pomocí Petriho sítě. Řízení je realizováno pomocí principu zpětné vazby. Omezení, která mají být procesem splněna, jsou vyjádřena pomocí logických výrazů, případně rovností a nerovností. Další informace a podrobné rozpracování této metody lze nalézt v literatuře [1][2].

8.1 Konstrukce supervizoru

Uveďme nyní ilustrativní příklad návrhu supervizoru pro Petriho sít'. Použijme již rozpracovaný příklad podle zdroje[3], kde byli stanoveny následující omezující podmínky

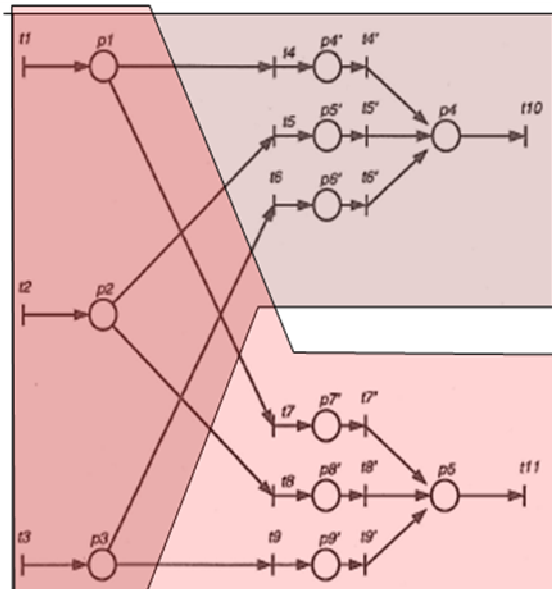
$$u4+u5+u6+u7+u8+u9 \leq 2.$$

Původní návrh, který simuluje proces probíhající v ATM přepínač (switch), konkrétně jde o simulaci jednotlivých portů a jejich vyrovnávacích pamětí (bufferů), je na následujícím obrázku Obr. 6. My jsme využili podobnosti s námi řešeným modelovým příkladem popsáním procesním digramem na obrázku Obr. 4.



Obr. 6. Petriho síť – realizace portů přepínače

Dalším krokem je návrh supervizoru. Použijeme metodu návrhu pro případ překrytí subsystémů podle [4]. Identifikujeme jednotlivé subsystémy viz. následující obrázek.



Obr. 7. Znázornění překrytí dvou systémů

Tento systém můžeme popsat následující stavovou maticí D .

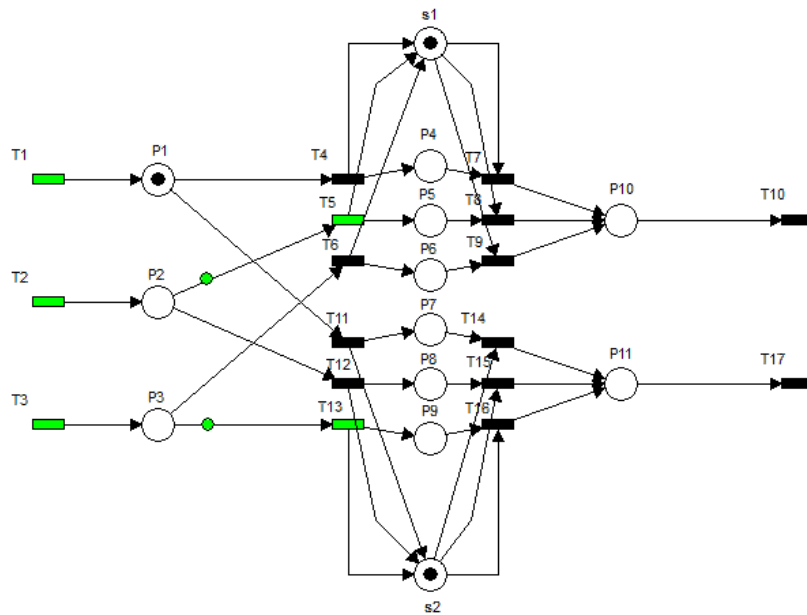
	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10
p1	1	0	0	-1						
p2	0	1	0		-1					
p3	0	0	1			-1				
p4	0	0	0	1			-1			
p5	0	0	0		1			-1		
p6	0	0	0			1			-1	
p10	0	0	0				1	1	1	-1

Obr. 8. Stavová matice systému z Obr. 7.

Stanovíme invariant míst jednoho subsystému, přičemž v tomto případě jsou invarianty míst pro oba subsystémy stejné.

$$L = (0\ 0\ 0\ 1\ 1\ 1\ 0)$$

Výsledný systém doplněný o supervizi, tedy systém, který zachovává stanovenou omezující podmínku je na následujícím obrázku. Výsledek byl ověřen simulací.



Obr. 9. Výsledná Petriho síť se supervizí

9 Závěr

Každý proces lze rozložit na několik málo základních konstrukčních prvků. Nejprve jsme definovali tyto obecné základní struktury a popsali jejich vlastnosti. Dalším krokem byla identifikace odlišností mezi procesními diagramy BORMu a Petriho sítěmi. Výsledkem bylo stanovení postupů transformace těchto odlišných struktur. Petriho síť pracuje s několika základními strukturami. Na jejich základě byly vytvořeny ekvivalentní struktury pro procesní diagramy v BORMu.

Jelikož je metodika BORM neformální nástroj, umožňuje vytvářet i nestandardní konstrukce. Právě takové konstrukce je nutné vytipovat a stanovit způsob transformace do Petriho sítě.

Hlavním cíle příspěvku bylo ukázat, jakým způsobem lze simulovat složitější procesy, zejména co se omezujících podmínek týče. Na příkladu přepínače respektive dispečeru, který byl nejprve zobrazen jako procesní diagram podle metodiky BORM byla demonstrována transformace na Petriho síť. Bez této transformace a použití Petriho sítě jako nástroje pro ověření funkčnosti, by nebylo možné jednoduše proces simulovat v nástroji CraftCase.

Příspěvek dále poukázal na silné a slabé stránky metody BORM a Petriho sítí, které se využívají pro modelování procesů. Omezení, které má nástroj CraftCASE lze účinně kompenzovat použitím Petriho sítě pro verifikaci a simulaci procesu.

Kombinací těchto dvou metod získáme komplexní nástroj pro modelování a verifikaci procesů, který je dobře pochopitelný i pro neodborníky a přitom dovoluje stanovit i složité podmínky pro řízení procesu.

Příspěvek vznikl jako součást grantu: Procesní modelování s pokročilejšími způsoby řízení – reg.č. 201011130009.

Reference

1. J.O. Moody, P.J. Antsaklis. *Supervisory Control of Discrete Event Systems using Petri Nets*. Kluwer Academic Publishers, Norwell, MA, 1998.
2. J.O. Moody, M.V. Iordache, P.J. Antsaklis. Enforcement of event-based supervisory constraints using state-based methods. *In Proceedings of the 38th IEEE Conference on Decision and Control*, pages 1743–1748, 1999.
3. Li, W.M. Wonham. Concurrent vector discrete-event systems. *IEEE Transactions on Automatic Control*, 40(4):628–638, 1995.
4. Papik M. *Teoretické nástroje procesního modelování*. Disertační práce. Praha 2010.

Abstract

For effective functioning of business processes need to processes be well documented and optimized. There are many ways to describe processes. Simple but effective approach appears to be the methodology BORM, which is implemented in the tool CraftCase. Unfortunately, this tool don't offer such a comprehensive process simulation capabilities, which are often needed. Ways to eliminate this shortcoming is using Petri nets for process simulation. First, we need capture the process with BORM methodology and then we transform it to the Petri net. This post deals with transformation rules, restrictions of BORM and Petri nets and finally we show how to construct supervisor of Petri nets.