

SUPER-DUÁLNÍ PETRIHO SÍŤ A JEJICH UPLATNĚNÍ PŘI MODELOVÁNÍ DISTRIBUOVANÝCH PROGRAMOVÝCH SYSTÉMŮ

Ivo Martiník

Ekonomická fakulta VŠB-TU Ostrava, Sokolská třída 33, 701 21 Ostrava 1, ČR
ivo.martinik@vsb.cz

ABSTRAKT:

Super-duální Petriho síť přináší oproti klasickým P/T Petriho sítím některé nové a originální koncepty, které lze s úspěchem využít při návrhu, modelování a verifikaci distribuovaných programových systémů. V příspěvku jsou prezentovány základní vlastnosti super-duálních Petriho sítí a zejména pak nové přístupy jejich nasazení při návrhu distribuovaných objektově-orientovaných programových systémů, jejichž hlavní myšlenkou je reprezentace aktivních objektů prostřednictvím značek míst (token) a pasivních objektů formou značek přechodů (poken) příslušné super-duální Petriho sítě.

KLÍČOVÁ SLOVA:

Super-duální Petriho síť, distribuované systémy, modelování, Java servlet, životní cyklus

1. ÚVOD

Formalismus Petriho sítí (viz. [1], [3]) byl zaveden před více než 40 lety a za tu dobu prodělal bouřlivý vývoj - od klasických "černobílých" P/T (*Place/Transition*) Petriho sítí přes C-E (*Condition-Event*) sítě, stochastické sítě, vysokourovňové (*High-Level*) sítě, fuzzy sítě, barevné (*Coloured*) sítě až po objektově-orientované (*Object-Oriented*) Petriho sítě, jež jsou zaměřeny především na modelování a verifikaci objektově-orientovaných programových systémů. V současné době jsou Petriho sítě spojovány hlavně s aplikacemi při návrhu, analýze a modelování paralelních a distribuovaných systémů, popisu komunikačních protokolů, distribuovaných databázových systémů, počítačových sítí, paralelního programování a překladačů.

P/T Petriho síť představují populární formalismus, který spojuje výhody grafického znázornění modelovaného systému s možnostmi jeho simulace a formální analyzovatelnosti. Systém je popsán bipartitním grafem obsahujícím:

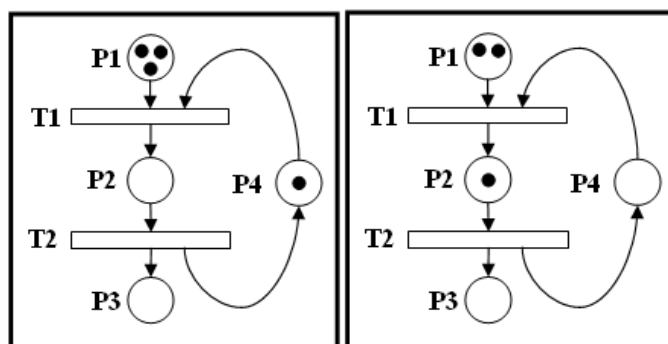
- *Místa (places)* sloužící k vyjádření stavů modelovaného systému (pro jejich znázornění obvykle používáme kružnice),
- *Přechody (transitions)* popisující změnu v systému (zakresluje je obvykle ve tvaru úseček nebo obdélníků),
- *Hrany (arcs)* jsou zásadně orientované a spojují místo s přechodem nebo přechod s místem, hrana nesmí spojit dvě místa nebo dva přechody a každá uspořádaná dvojice

(*místo, přechod*) nebo (*přechod, místo*) smí být spojena nejvýše jednou hranou (hrany zakreslujeme většinou jako úsečky se šipkami),

- *Značení sítě (marking)* - vyjadřuje aktuální stav modelovaného systému pomocí tzv. *značek (tokens)*, které považujeme za vzájemně nerozlišitelné a znázorňujeme je v podobě malých kružnic v jednotlivých místech sítě. Každému místu sítě lze tedy přiřadit celé kladné číslo vyjadřující počet značek modelovaného systému v tomto místě při popisu jeho jistého stavu. Je-li modelovaný systém v počátečním stavu, hovoříme o jeho *počátečním značení*.

Jelikož jsou hrany P/T Petriho sítě orientované, lze u každého místa nebo přechodu sítě hovořit o množině *vstupních hran*, tj. množině hran, jež jsou zakončeny v daném místě nebo přechodu, a množině *výstupních hran*, tj. hran, jež mají naopak svůj počátek v daném místě nebo přechodu. Obdobně jako o vstupních a výstupních hranách lze hovořit o *vstupních a výstupních elementech* studovaného místa nebo přechodu. Např. množina vstupních míst daného přechodu jsou všechna místa asociována se vstupními hranami přechodu a množina jeho výstupních míst je tvořena všemi místy asociovanými s výstupními hranami přechodu. Obdobně vstupní a výstupní přechody daného místa.

Jak již bylo řečeno, pomocí P/T Petriho sítí lze nejenom zachytit aktuální stav modelovaného systému, ale rovněž dynamiku přechodů mezi jeho jednotlivými stavy. Na schématu P/T Petriho sítě se tato dynamika znázorňuje *prováděním přechodů sítě*. Přechod smí být proveden (je proveditelný), je-li v každém vstupním místě přechodu alespoň jedna značka. Samotné jeho provedení spočívá v odebrání jedné značky ze všech vstupních míst přechodu a přidání jedné značky do všech výstupních míst prováděného přechodu.



Obr. 1a, b. P/T Petriho síť

Na obr. 1a je znázorněna P/T Petriho síť modelující situaci známou z oblasti tvorby vícevláknových programových systémů, v níž tři programová vlákna reprezentovaná značkami v místě *P1* sítě požadují vstup do kritické sekce programu (místo *P2* sítě) chráněné programovým zámekem (značka v místě *P4* sítě). Vstup do kritické sekce (provedení přechodu *T1* sítě) je tak umožněn pouze jednomu z programových vláken (viz. obr. 1b).

2. SUPER-DUÁLNÍ PETRIHO SÍŤ

Super-duální Petriho síť (viz. např. [2]) přináší k základnímu konceptu P/T Petriho sítí některá významná rozšíření, která lze shrnout následovně:

- Pro popis aktuálního stavu modelovaného systému jsou k dispozici **dva typy značek**. První typ značek je shodný se značkami (*token*) P/T Petriho sítí a je asociován s jednotlivými místy sítě. O těchto značkách budeme v dalším hovořit jako o *t-značkách* a stejně jako v případě P/T Petriho sítí je budeme znázorňovat v podobě malých kružnic v jednotlivých místech sítě. Druhý typ značek (*poken*) je asociován s přechody sítě, značky tohoto typu budeme proto označovat jako *p-značky* a znázorňovat prostřednictvím malých čtverců ve vybraných přechodech sítě.
- Super-duální Petriho sítě disponují rovněž **dvěma typy hran**. První typ hran, který budeme označovat jako *t-hrany*, je opět shodný s hranami P/T Petriho sítě. V grafickém znázornění sítě je tento typ hran zakreslen plnými čarami a determinuje provádění přechodů sítě, které je realizováno *t-značkami* sítě. Druhý typ hran, který je specifický pro super-duální Petriho sítě, budeme označovat jako *p-hrany* a v grafickém znázornění sítě zakreslovat čárkovanými čarami. Tento typ hran naopak determinuje **provádění míst sítě s využitím p-značek** sítě.

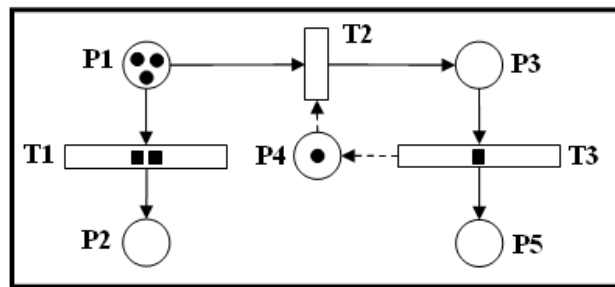
Obdobně jako v případě P/T Petriho sítí lze i u super-duálních Petriho sítí zavést pojmy *vstupních a výstupních t-hran*, *vstupních a výstupních p-hran*, *vstupních a výstupních t-elementů* a *vstupních a výstupních p-elementů* vybraného místa nebo přechodu sítě.

Významnou změnou oproti P/T Petriho sítím je i mechanismus provádění přechodů a míst super-duální Petriho sítě. Přechod smí být proveden, obsahuje-li alespoň jednu *p-značku* a v každém jeho vstupním *t-místě* (tedy vstupním místě přechodu, které je s ním spojeno *t-hranou*) je obsažena alespoň jedna *t-značka*. Samotné jeho provedení spočívá v odebrání jedné *t-značky* ze všech vstupních *t-míst* přechodu a přidání jedné *t-značky* do všech výstupních *t-míst* prováděného přechodu. Obsahuje-li prováděný přechod n ($n > 1$) *p-značek*, udává tento počet značek maximální paralelismus v provádění přechodu, kdy je možno z každého vstupního *t-místa* prováděného přechodu odebrat stejný počet až n *t-značek* a tento počet *t-značek* pak přidat do všech výstupních *t-míst* prováděného přechodu.

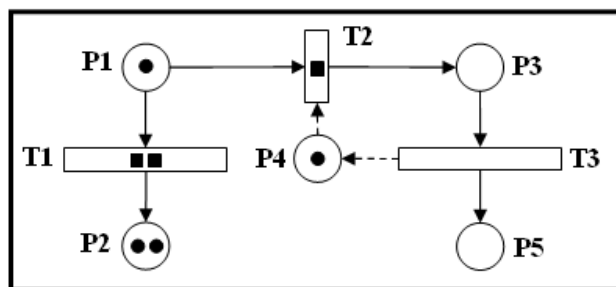
Obdobným způsobem je zavedeno i provádění míst super-duální Petriho sítě. Místo smí být provedeno, obsahuje-li alespoň jednu *t-značku* a v každém jeho vstupním *p-přechodu* (tedy vstupním přechodu místa, které je s ním spojeno *p-hranou*) je obsažena alespoň jedna *p-značka*. Samotné jeho provedení spočívá v odebrání jedné *p-značky* ze všech vstupních *p-přechodů* místa a přidání jedné *p-značky* do všech výstupních *p-přechodů* prováděného místa. Obsahuje-li prováděné místo m ($m > 1$) *t-značek*, udává tento počet značek maximální paralelismus v provádění místa, kdy je možno z každého vstupního *p-přechodu* prováděného místa odebrat stejný počet až m *p-značek* a tento počet *p-značek* pak přidat do všech výstupních *p-přechodů* prováděného místa.

Příklad super-duální Petriho sítě je znázorněn na obr. 2. Místa *P1* a *P4* obsahují *t-značky*, přechody *T1* a *T3* jsou asociovány s *p-značkami*. Přechod *T3* je spojen s místem *P4* *p-hranou*, obdobně jako místo *P4* s přechodem *T2*. Ostatní hrany sítě jsou *t-hranami*. V daném značení sítě smí být proveden přechod *T1*, jehož jediné vstupní *t-místo* *P1* obsahuje tři *t-značky* a samotný přechod *T1* je asociován se dvěma *p-značkami*. Přechod *T2* nemůže být v daném značení sítě proveden, neboť i když jeho vstupní *t-místo* *P1* obsahuje dostatečný počet značek pro jeho provedení, přechod *T2* neobsahuje žádnou *p-značku*. Přechod *T3* rovněž nemůže být v daném značení proveden, neboť jeho jediné vstupní *t-místo* *P3* neobsahuje žádnou *t-značku*. Provedeno ale smí být místo *P4*, neboť jeho vstupní *p-přechod* *T3* obsahuje potřebnou *p-*

značku a s místem $P4$ je asociována t -značka. Značení sítě po provedení přechodu $T1$ a místa $P4$ je znázorněno na obr. 3.

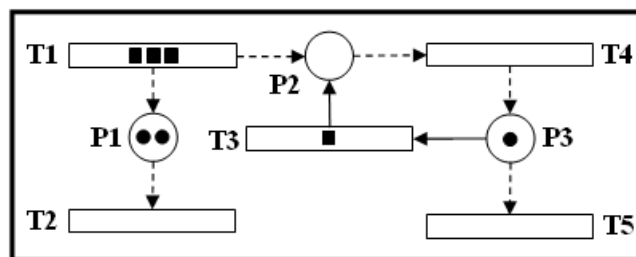


Obr. 2. Super-duální Petriho síť



Obr. 3. Provedení přechodu a místa v super-duální Petriho síti

Ke každé super-duální Petriho síti lze zkonstruovat její **duální síť**, v níž jsou všechna její místa nahrazeny přechody, všechny přechody místy, všechny p -značky t -značkami a všechny t -značky p -značkami. Rovněž je možno provést konstrukci tzv. **konjugované** sítě k super-duální Petriho síti, v níž jsou všechny její t -hrany nahrazeny p -hranami a všechny p -hrany nahrazeny t -hranami. Duální konjugovaná síť k super-duální Petriho síti na obr. 2 je znázorněna na obr. 4.



Obr. 4. Duální konjugovaná síť k super-duální Petriho síti

Ke každé super-duální Petriho síti lze rovněž stanovit její t -komponentu, tj. síť, z níž jsou odebrány všechny p -hrany původní sítě, a p -komponentu, tj. síť, z níž jsou odebrány všechny t -hrany původní sítě.

3. MODELOVÁNÍ DISTRIBUOVANÝCH PROGRAMOVÝCH SYSTÉMŮ S VYUŽITÍM SUPER-DUÁLNÍCH PETRIHO SÍTÍ

Hlavním cílem tohoto příspěvku je neformálně uvést nové přístupy nasazení aparátu super-duálních Petriho sítí při modelování distribuovaných objektově-orientovaných programových systémů. Budeme je demonstrovat na jednoduchém příkladu generování

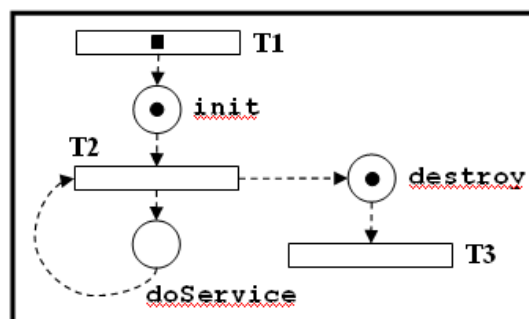
dynamické www stránky na základě požadavku strany klienta s využitím technologie Java servletů v prostředí Java EE aplikačního serveru (podrobně viz. např. [4]).

Základním problémem, který je potřeba obvykle řešit při modelování komplexních distribuovaných objektově-orientovaných programových systémů s využitím Petriho sítí, je sestavení výsledné Petriho sítě modelovaného systému na základě znalostí životních cyklů jednotlivých instancí tříd, z nichž je daný programový systém zkomponován.

Formalismus super-duálních Petriho sítí nabízí možnost využití následujícího přístupu:

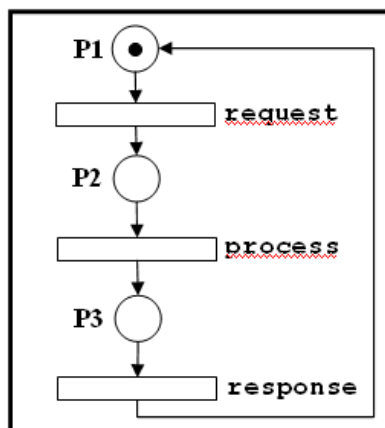
- aktivní objekty (programová vlákna) budou ve výsledné super-duální Petriho síti modelována *t-značkami*, které budou v jednotlivých značeních sítě asociovány s jejími místy, pasivní objekty *p-značkami* asociovanými s přechody sítě,
- životní cyklus aktivních objektů bude ve výsledné síti vyjádřen prostřednictvím jejich *t-komponent*, životní cyklus pasivních objektů prostřednictvím příslušných *p-komponent* sítě,
- výsledná super-duální Petriho síť vznikne (neformálně řečeno) kompozicí jednotlivých super-duálních Petriho sítí modelujících životní cykly instancí tříd programového systému, přičemž bude provedeno sloučení (navázání) míst a přechodů jednotlivých sítí, v nichž dochází k interakci instancí tříd modelovaného systému.

V našem příkladu nejprve popíšeme životní cyklus instance vybraného Java servletu jednoduchou super-duální Petriho sítí (viz. obr. 5). Instance Java servletu je pasivním objektem reprezentovaným *p-značkou* v přechodu *T1* sítě. Životní cyklus instance Java servletu je zahájen jeho inicializací, která je realizována provedením metody `init(...)` (ve schématu super-duální Petriho sítě je znázorněna místem `init` obsahujícím *t-značku* reprezentující instanci programového vlákna, které uvedenou metodu provádí). Je-li instance Java servletu korektně inicializována, smí být prováděna metoda `doService(...)` určená k vlastnímu procesu generování dynamické www stránky (místo `doService` ve schématu sítě). Vlastní provedení metody `doService(...)` realizuje vybrané programové vlákno Java EE aplikačního serveru, které je určeno k obsluze požadavků strany klienta. Ukončení životního cyklu instance Java servletu je zabezpečeno provedením metody `destroy()` (místo `destroy` ve schématu sítě).



Obr. 5. Super-duální Petriho síť modelující životní cyklus instance Java servletu

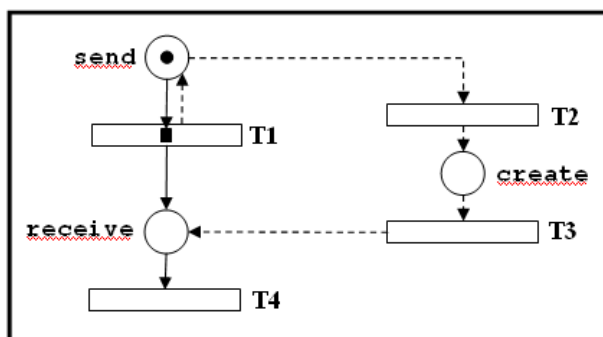
Zjednodušený životní cyklus programového vlákna Java EE serveru určeného pro obsluhu požadavků strany klienta a zabezpečení generování dynamické www stránky je znázorněn super-duální Petriho sítí na obr. 6.



Obr. 6. Super-duální Petriho síť modelující životní cyklus programového vlákna Java EE aplikačního serveru

Programové vlákno (*t-značka* v místě *P1* schématu sítě) při přijetí požadavku strany klienta určeného pro generování dynamické *www* stránky (přechod *request* ve schématu sítě) provede příslušnou metodu Java servletu určenou k vygenerování požadované *www* stránky (přechod *process* ve schématu sítě), výsledný obsah stránky pak zašle prostřednictvím *www* serveru straně klienta (přechod *response* ve schématu sítě) a je aplikačním serverem zařazeno zpět do k obsluze dalších požadavků strany klienta.

Základní funkcionality strany klienta požadujícího generování dynamické *www* stránky jsou znázorněny super-duální Petriho sítí na obr. 7.

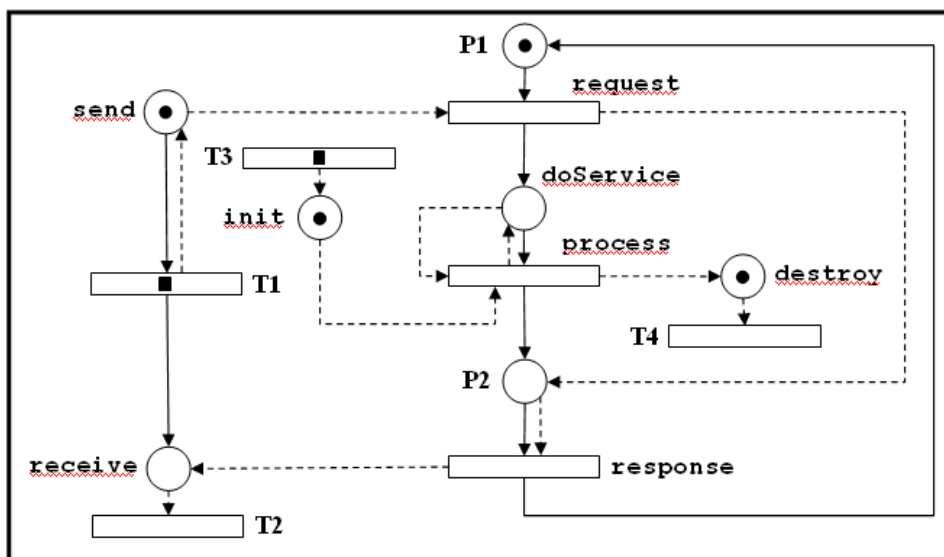


Obr. 7. Super-duální Petriho síť modelující funkcionality strany klienta

Programové vlákno strany klienta (*t-značka* v místě *send* schématu sítě) realizuje odeslání požadavku (*p-značka* v přechodu *T1* schématu sítě) straně Java EE aplikačního serveru (přechod *T2* schématu sítě), která zabezpečí vygenerování dynamické *www* stránky (místo *create* schématu sítě) a její zaslání zpět straně klienta (místo *receive* schématu sítě) v prostředí distribuovaného programového systému.

Naším úkolem je nyní na základě znalostí super-duálních Petriho sítí modelujících životní cykly jednotlivých instancí tříd distribuovaného objektově-orientovaného programového systému vytvořit jedinou síť modelující životní cyklus systému jako celku.

Výsledná super-duální Petriho síť modelující životní cyklus celého distribuovaného systému, která vznikla kompozicí super-duálních Petriho sítí na obr. 5, 6 a 7, je znázorněna na obr. 8. Při jejím vytváření došlo ke sloučení přechodu $T2$ sítě vyjadřující životní cyklus Java servletu a přechodu $process$ sítě životního cyklu programového vlákna Java EE aplikačního serveru (přechod $process$ výsledné sítě), ke sloučení místa $doService$ sítě vyjadřující životní cyklus Java servletu a místa $P2$ sítě životního cyklu programového vlákna Java EE aplikačního serveru (místo $doService$ výsledné sítě), ke sloučení přechodu $request$ sítě vyjadřující životní cyklus Java servletu a přechodu $T2$ sítě vyjadřující funkcionality strany klienta (přechod $request$ výsledné sítě), místa $P2$ sítě vyjadřující životní cyklus Java servletu a místa $create$ sítě vyjadřující funkcionality strany klienta (místo $P2$ výsledné sítě) a přechodu $response$ sítě vyjadřující životní cyklus Java servletu a přechodu $T3$ sítě vyjadřující funkcionality strany klienta (přechod $response$ výsledné sítě).



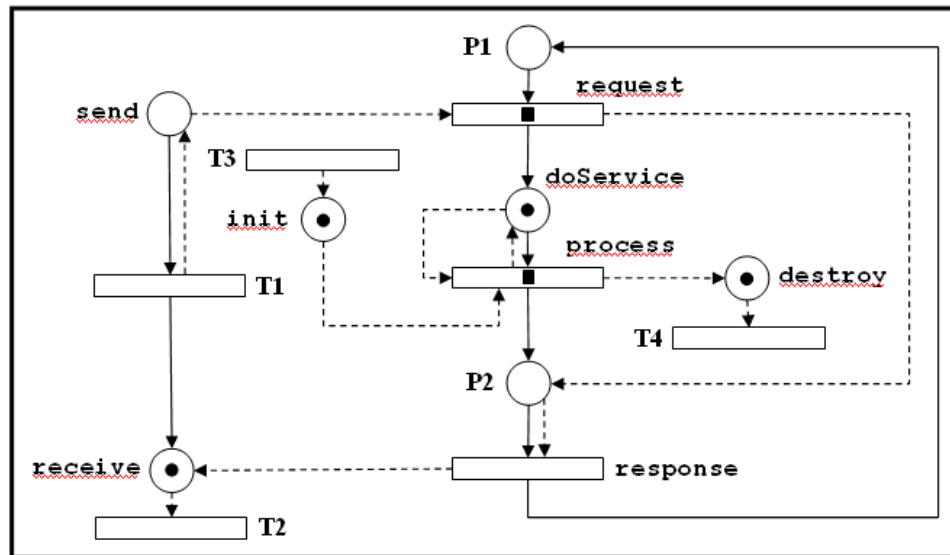
Obr. 8. Výsledná super-duální Petriho síť

Ve výsledné super-duální Petriho síti jsou tedy:

- pasivní objekty (instance Java servletu, požadavek strany klienta) reprezentovány p -značkami,
- aktivní objekty (programová vlákna Java EE aplikačního serveru a strany klienta) reprezentovány t -značkami,
- p -komponenta sítě modeluje životní cyklus Java servletu a požadavku strany klienta,
- t -komponenta sítě modeluje životní cyklus programových vláken Java EE aplikačního serveru a programového vlákna strany klienta distribuovaného programového systému,
- bylo provedeno sloučení vybraných míst a přechodů v rámci kompozice sítí.

V počátečním značení výsledné super-duální Petriho sítě na obr. 8 smí být t -značkou v místě $send$ proveden přechod $T1$, p -značkou v přechodu $T1$ provedeno místo $send$ a p -značkou v přechodu $T3$ provedeno místo $init$. Po úspěšné realizaci těchto kroků pak smí

být *p-značkou* v místě *P1* proveden přechod *request*. Tento krok převede síť do značení znázorněném na obr. 9. Uvedené značení odpovídá stavu programového systému, ve kterém je úspěšně inicializována instance Java servletu (*p-značka* asociovaná s přechodem *process*) a programové vlákno Java EE aplikačního serveru přijalo požadavek strany klienta na vytvoření dynamické *www* stránky (*t-značka* v místě *request*), která bude následně generována provedením metody *doService* Java servletu a předána straně klienta.



Obr. 9. Značení výsledné super-duální Petriho sítě

4. ZÁVĚR

Super-duální Petriho sítě reprezentují velice zajímavý formalismus, který může být využit k modelování distribuovaných objektově-orientovaných programových systémů. V příspěvku byly prezentovány nebarevné super-duální Petriho sítě, které lze zobecnit na třídu barevných a dále pak rovněž na třídu objektových super-duálních Petriho sítí a studovat možnosti a vlastnosti jejich kompozice.

LITERATURA

1. REISIG W., ROZENBERG G.: Lectures on Petri Nets I: Basic Models, Springer, 1998, ISBN 3-540-65306-6
2. KÖHLER M., RÖLKE H.: Web Services Orchestration With Super-Dual Petri Nets, In ICATPN 2007, Lecture Notes in Computer Science 4546, pp. 263–280, Springer-Verlag, 2007, ISBN 978-3-540-73093-4
3. REISIG W.: Elements Of Distributed Algorithms, Springer, 1998, ISBN 3-540-62752-9
4. JENDROCK E., BALL J., CARSON D., EVANS I., FORDIN S., HAASE K.: The Java EE 5 Tutorial, Sun Microsystems, 2008