

# XSLT ŠABLONA PRO VYTVOŘENÍ GRAFU OBJEKTOVÉ PETRIHO SÍTĚ Z FORMÁTU OPNML

P. Jedlička

Došlo: 18. prosince 2007

## Abstract

JEDLIČKA, P.: *XSLT template for generating of diagram of object-oriented Petri net from the OPNML format*. Acta univ. agric. et silvic. Mendel. Brun., 2008, LVI, No. 3, pp. 251–256

OPNML language is an open and portable format based on XML which is determined to the description of object-oriented Petri net. It is capable to express not only the structure of the analyzed system, but also its behavior and diagram. To display Petri net as a graph, there was created the template using XSLT language which defines the rules of transformation from OPNML language to SVG vector graphic format. SVG diagrams are then viewable with any web browser which supports SVG format, optionally relevant plug-in can be installed. The template also supports modular Petri nets as it is capable of creating several diagrams from one OPNML document. Individual diagrams can be interconnected via XLink links, so there is possible to move among them by mouse.

Petri net, object orientation, OPNML, XML, XSLT, SVG

Jazyk OPNML (Object Petri Net Markup Language) byl vytvořen jako otevřený přenositelný formát na bázi XML určený ke komplexnímu popisu objektově orientovaných Petriho sítí (Object Oriented Petri Nets – dále jen OOPN) (Jedlička, 2007). Jako výchozí formát pro definici OPNML byl použit jazyk PNML, vytvořený v Institutu informatiky Humboldtovy univerzity v Berlíně (Weber, 2006). Jazykem PNML lze však popsat pouze strukturované a modulární Petriho sítě. Objektové Petriho sítě jsou značně složitější, proto i definice jazyka OPNML je oproti PNML mnohem rozsáhlejší. Jako výchozí pojetí OOPN byla pro konstrukci jazyka OPNML použita koncepce Martiníka (1999).

Jazyk OPNML splňuje zásady jazyků XML, je tedy nezávislý na platformě, přenositelný a díky textové podobě rovněž snadno čitelný pro člověka. Čitelnost v tomto případě však rozhodně neznamená názornost. Orientace v dokumentu OPNML je pro člověka obtížná, vytvoření názorné představy o struktuře popsané sítě pak téměř nemožné. Jednou z výhod Petriho sítí je ale možnost jejich zobrazení v podobě grafu. Grafické podoby je proto vhodné využít i v případě OOPN zapsaných v jazyce OPNML, současně by však bylo výhodné zůstat ve sféře jazyků XML. Existuje zde mnoho různých nástrojů a technologií pro manipulaci s XML dokumenty a nebylo by tudíž potřeba vytvářet vlastní. V tomto příspěvku

je tedy řešen problém, jak pomocí technologie XML vytvořit z OPNML dokumentu graf OOPN.

## MATERIÁL A METODY

Výše uvedené důvody vedou k využití formátu SVG jako optimální platformy pro vizualizaci grafu OOPN. Jazyk **SVG (Scalable Vector Graphic)** byl vytvořen (podobně jako OPNML) aplikací jazyka XML. Je určen k prezentaci dvoudimenzionální vektorové grafiky a případně i na ní založených animací. Zatím poslední verze jazyka nese označení 1.2 (W3C, 2005). Kromě specializovaných editorů a prohlížečů lze dokument v jazyce SVG v grafické podobě prezentovat prostřednictvím klasických webových prohlížečů. Některé prohlížeče umějí SVG grafiku zobrazovat přímo, do jiných je třeba nainstalovat plugin (zásuvný modul), např. Adobe SVG Viewer.

Transformace modelu z jazyka OPNML do jazyka SVG je v obecné rovině transformací dat z jednoho XML jazyka do jiného. K těmto účelům je určen jazyk **XSLT (eXtensible Stylesheet Language Transformation)**, který slouží k vytváření šablon, na jejichž základě jsou prováděny transformace XML dokumentů. XSLT je součástí širší specifikace – jazyka XSL (*eXtensible Stylesheet Language*). Druhou část XSL – podstatně složitější – tvoří jazyk XSL-FO (*XSL Formatting Objects*), který slouží k definici formátování

výstupu transformace. Transformace dokumentů z jednoho jazyka XML do jiného na základě XSLT šablony jsou prováděny speciálními programy, tzv. *XSLT procesory*. Mezi nejznámější patří Xalan Java, Xalan C++, Saxon, Microsoft XML Parser a XT. Pro přístup k jednotlivým částem dokumentu využívá XSLT jazyka *XPath*.

Stejně jako SVG je rovněž XSL výsledkem práce konsorcia W3C. Jazyk XSL vznikl odvozením od jazyka *DSSSL (Document Style Semantics and Specification Language)*. První verze jazyka XSLT byla v podobě doporučení konsorcia W3C zveřejněna 16. listopadu 1999. Následně byl uvolněn pracovní návrh verze 1.1, ale než byl dopracován do podoby doporučení, začaly práce na verzi 2.0 a vývoj verze 1.1 byl zastaven. Verze 2.0 byla ve formě doporučení zveřejněna 23. ledna 2007 (W3C, 2007).

Myšlenka použít jazyky XSLT a SVG k vytvoření grafu Petriho sítě není zcela nová. Na katedře informatiky Univerzity Carla von Ossietzky v německém Oldenburgu bylo ve spolupráci s dalšími vědeckými pracovišti v Německu vytvořeno několik nástrojů pro modelování, transformace, simulace a verifikace Petriho sítí. Celá sada dostala název *PEP – Programming Environment based on Petri Nets* (Stehno, 2004). Mimo jiné zde byla vytvořena i transformační šablona v jazyce XSLT pro konverzi Petriho sítí z jazyka PNML do formátu SVG. Tato šablona sice není přímo použitelná pro vytvoření grafu z dokumentu OPNML, ale lze z ní částečně vycházet. Nyní následuje popis této šablony, tedy charakteristika toho, jak je zde řešeno vytvoření grafu Petriho sítě a jeho komponent. Šablona je příliš rozsáhlá na to, abychom ji popisovali podrobně. Soustředíme se proto pouze na významné prvky a principy využití jazyka SVG.

### Výstup do více souborů

Šablona předpokládá dokument v jazyce PNML ve verzi *basic* nebo *structured* (viz Weber, 2006). Druhá ze jmenovaných verzí umožňuje hierarchizovat Petriho síť do více stránek. Šablona grafy těchto stránek zobrazuje volitelně buď do jednoho dokumentu, nebo ukládá každý graf do samostatného souboru. OOPN bude prakticky vždy hierarchická a její zobrazení do jediného grafu by bylo velmi nepřehledné, ne-li nemožné. Možnost ukládat grafy jednotlivých částí OOPN do samostatných souborů bude tedy zcela určitě využita.

### Místo

*Místo* je v grafu Petriho sítě znázorněno pomocí kružnice, příp. elipsy. Jak PNML, tak i OPNML připouští i použití obrázku. Pro kružnici a elipsu je v PEP z jazyka SVG použit element *ellipse*, pro obrázek element *image*. Pomocí atributů je pak nastaveno umístění, rozměry, parametry obrysové čáry a u kružnice a elipsy rovněž barva výplně. Kromě souřadnic umístění mají všechny parametry implicitní hodnoty.

Ukázka elementu *ellipse*:

```
<ellipse cy="0" cx="0" rx="10" ry="10" style="fill:white;stroke: black;"/>
```

### Přechod

*Přechody* bývají v grafech Petriho sítí zobrazeny jako obdélníky. Stejně jako v případě *místa* také zde PNML i OPNML připouští i použití obrázku. Použije se buď SVG element *rect* (pro obdélník) nebo již známý *image*. Volitelné parametry i jejich implicitní hodnoty se neliší od předchozího případu *místa*.

Ukázka elementu *rect*:

```
<rect x="80" y="12" width="160" height="20" fill="white" stroke="black"/>
```

### Hrana

Šablona z projektu PEP podporuje pouze zobrazení obyčejných hran reprezentovaných jednosměrnou šipkou. Testovací a inhibiční hrany Petriho sítí nejsou podporovány. Hrana je modelována SVG elementem *path*. Jeho atributy umožňují zvolit obvyklé parametry čáry, jako jsou souřadnice bodů, kterými má procházet, dále barva, šířka, styl (plná, tečkovaná, čárkovaná, čerchovaná) apod. Zajímavá je v SVG možnost vytvoření vlastního stylu čerchování, kdy lze dvojicemi čísel nastavit délky čárek a mezer mezi nimi. Je také možné zvolit mezi rovnou, resp. lomenou úsečkou a Bézierovou křivkou.

Počátek každé hrany je stanoven na střed jejího počátečního uzlu. Hrany jsou vykresleny dříve než uzly (to je dáno pořadím zápisu elementů do dokumentu SVG), proto je vždy část hrany od středu k okraji značky uzlu překryta samotnou značkou uzlu a hrana se zobrazuje až od okraje značky. Druhý konec čáry hrany je však opatřen šipkou a ta ji musí končit přesně na okraji koncového uzlu. Souřadnice tohoto bodu jsou v šabloně vypočítány na základě znalosti souřadnic středu uzlu, jeho rozměrů a tvaru (elipsa/obdélník).

Za pozornost stojí rovněž způsob vytvoření šipky na konci čáry hrany. Tvar, který má být vykreslen na konci čáry (u klasické plné šipky reprezentující obyčejnou hranu je to trojúhelník), se popíše některým z elementů jazyka SVG v rámci pojmenovaného elementu *marker*. URL tohoto elementu je pak zadáno do atributu *marker-end* elementu *path*, jenž modeluje čáru hrany.

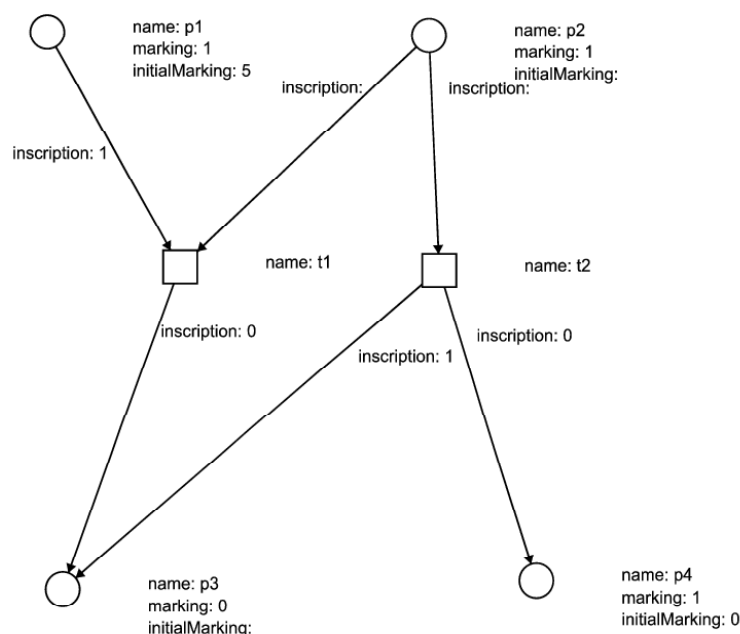
Ukázka elementu *path*:

```
<defs>
  <marker id="mEndArc" markerWidth="5" markerHeight="6" orient="auto" refX="5" refY="3">
    <path d="M 0 0, 5 3, 0 6 Z" style="fill:black; stroke-width:1"/>
  </marker>
</defs>

<path d="M5,5 C5,45 45,45 45,5" stroke="black" fill="red"
  marker-end="url(#mEndArc)"/>
```

### Dílčí shrnutí

Představená XSLT šablona byla použitelnou výchozí definicí pro vytvoření šablony transformace dokumentů z jazyka OPNML do jazyka SVG. Na obrázku 1 je ukázka výstupu transformace.



1: Ukázka grafu P/T Petriho sítě vytvořeného v jazyce SVG transformační šablonou projektu PEP

## VÝSLEDKY

V této části bude prezentována XSLT šablona pro transformaci OOPN z jazyka OPNML do grafu sítě ve formátu SVG. Šablona umožňuje převedení všech údajů popisujících objektovou Petriho síť v jazyce OPNML do grafického vyjádření. Proto byly pro transformaci zvoleny pouze *stránky tříd* a *stránky instancí tříd* sítě, nikoliv jejich instance a značení těchto instancí. Zpracování *instancí stránek* by přineslo stejné grafy jako jsou grafy *stránek*, rozdílné by bylo pouze aktuální umístění značek. Kompletní šablona je dostupná na internetu na webové adrese <https://akela.mendelu.cz/~petrj/opnml/>. V následujícím popisu jsou uvedena jen významná rozšíření oproti výše představené šabloně z projektu PEP.

### Výstup do více souborů

Výstup do více souborů, který je v šabloně PEP volitelný a nastává, jen když je zpracovávána síť hierarchická, je v případě objektové Petriho sítě aplikován vždy. Každá hlavní podstránka a každá podstránka metody každé třídy nebo její instance je zobrazena samostatným grafem uloženým do samostatného souboru. Názvy souborů jsou tvořeny podle následujícího schématu:

- hlavní stránka objektové Petriho sítě: `mainPage.svg`
- hlavní podstránka třídy: `class.název_třídy.svg`
- hlavní podstránka instance třídy: `object.název_třídy.svg`
- podstránka konstrukturu: `constructor.název_třídy(datové_typy_parametrů).svg`
- podstránka destrukturu: `destructor.název_třídy(datové_typy_parametrů).svg`

- podstránka metody: `method.název_třídy.typ_vlastníka.název_metody(datové_typy_parametrů).svg`
- hierarchická podstránka: `page.název_podstránky.svg`

### Místo

Byly zavedeny značky pro *vstupní místo* a *výstupní místo* podstránky metody, konstrukturu a destrukturu s velkým písmenem „I“ (input), resp. „O“ (output) uprostřed značky místa (kružnice). V šabloně byl zaveden parametr *displayParams*. Nastavení jeho hodnoty na 1 způsobí automatické zobrazování parametrů vstupních a výstupních míst, aniž by v dokumentu modelu bylo třeba používat popisný element *inscription* převzatý z jazyka PNML. Stejně použití atributu *displayIDs* způsobí automatické zobrazování identifikátorů uzlů (míst a přechodů) v grafu sítě.

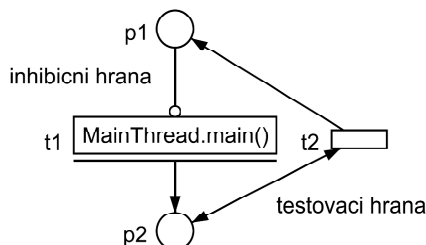
### Přechod

Bylo třeba zavést nové značky pro synchronní a asynchronní hierarchické přechody. Uvnitř těchto značek se zobrazuje název podstránky, kterou přechod reprezentuje. Je-li touto podstránkou „prostá“ hierarchická podstránka (tedy nikoliv podstránka metody), je její název současně odkazem na její graf, který se po kliknutí na odkaz zobrazí v prohlížeči. U metod tuto vazbu nelze aplikovat, neboť z důvodu polymorfismu není předem známo, ze které třídy bude pocházet metoda reprezentovaná přechodem.

### Hrana

Definice objektové Petriho sítě (Martiník, 1999) zavádí kromě „obyčejných“ hran ještě hrany *inhibiční*

a jazyk OPNML umožňuje použít i hrany *testovací*. Inhibiční hrana je v grafu reprezentována šipkou zakončenou malou kružnicí, testovací hrana má podobu oboustranné šipky – viz obr. 2. Obě tyto hrany šablona v grafu vytváří.



2: Graf sítě s inhibiční hranou a testovací hranou

Druhé rozšíření šablony si vyžádala potřeba zpracování *výrazů hranových funkcí*. Výraz je zapsán buď v jednom textovém elementu nebo strukturovaně v jazyce MathML. Pokud je použita textová podoba, je obsah elementu přímo zobrazen na určené souřadnice. Strukturovaně zapsaný výraz je nejprve převeden do textové podoby a teprve poté umístěn do grafu.

V úvodní části souboru šablony je možné nastavit několik parametrů transformace. Některé se aplikují pouze v případě, kdy příslušný parametr chybí v OPNML dokumentu. Jsou to tyto parametry:

Jméno parametru	Popis	Aktuální hodnota
defDir	cílový adresář pro ukládání SVG dokumentů	./SVG/
defSize	implicitní šířka a výška značky místa	20
defTransWidth	implicitní šířka značky přechodu	30
defTransHeight	implicitní výška značky přechodu	10
defFillColor	implicitní barva výplně prvků grafu	white
defLineColor	implicitní barva čáry	black
defIdSize	implicitní velikost písma	12
displayIDs	přepínač automatického zobrazování identifikátorů uzlů: 0 = nezobrazovat, 1 = zobrazovat	1
displayParams	přepínač automatického zobrazování parametrů vstupních a výstupních míst podstránek: 0 = nezobrazovat, 1 = zobrazovat	1

## DISKUSE

Navržená XSLT šablona umožňuje transformaci modelu objektové Petriho sítě z jazyka OPNML do vektorového formátu SVG. Pro každou stránku sítě je vygenerován jeden soubor s jejím grafem. Grafy jsou mezi sebou provázány pomocí XLink odkazů tak, že jimi zobrazované stránky sítě

tvoří jednu hierarchickou strukturu OOPN. Šablona je ve své finální podobě zveřejněna na adrese <https://akela.mendelu.cz/~petrj/opnml/>. Tam jsou také vystaveny ukázky grafů podstránek objektové Petriho sítě, jež jsou výsledkem transformace vzorového OPNML dokumentu právě podle této šablony.

## SOUHRN

V jazyce XSLT (Extensible Stylesheet Language) byla vytvořena šablona s detailními pravidly pro transformaci modelu objektové Petriho sítě z jazyka OPNML (Object Petri Net Markup Language) do vektorového grafického formátu SVG. Místo, znázorněné v grafu Petriho sítě v podobě kružnice, je vykresleno pomocí SVG elementu *ellipse*. Přechody, které bývají v grafech Petriho sítí zobrazeny jako obdélníky, jsou modelovány elementem *rect*. Specifický vzhled mají synchronní a asynchronní hierarchické přechody. Hrana je modelována SVG elementem *path*. Na typu hrany (normální, testovací, inhibiční) závisí zakončení její šipky v grafu. Různá zakončení jsou definována pomocí elementu *marker* a poté ve formě odkazu použita v definici hrany (element *path*). Součástí popisu objektové Petriho sítě v jazyce OPNML jsou i hranové výrazy (booleovské u vstupních a přiřazovací u výstupních hran přechodu), které jsou rovněž zpracovány a následně zobrazeny v grafu. K prohlížení vytvořených SVG grafů je možno použít libovolný webový prohlížeč, který podporuje SVG formát, případně lze doinstalovat příslušný zásuvný modul. Šablona podporuje i modulární a hierarchické Petriho sítě tím, že umožňuje vytvoření více grafů z jednoho OPNML dokumentu. Každá hlavní podstránka a každá podstránka metody každé třídy nebo její instance je zobrazena samostatným grafem ulože-



ným v samostatném souboru. Jednotlivé grafy pak mohou být mezi sebou provázány pomocí XLink odkazů, aby mezi nimi bylo možno v prohlížeči pohodlně přecházet pomocí myši. Kompletní šablona pro XSLT transformaci je dostupná na webové adrese <https://akela.mendelu.cz/~petrj/opnml/>. Na této webové stránce je také zveřejněno XML schéma jazyka OPNML a ukázky grafů ve formátu SVG vytvořených z OPNML dokumentu pomocí XSLT šablony.

Petriho síť, objektová orientace, OPNML, XML, XSLT, SVG

## SUMMARY

Using XSLT (Extensible Stylesheet Language), template to transform Object-Oriented Petri Net described by OPNML (Object Petri Net Markup Language) into a graph was created. The template defines rules of transformation from OPNML language to the vector graphic format SVG. *Place*, which is in Petri Net graph displayed in shape of a circle, is painted by the medium of SVG-element *ellipse*. Transitions, displayed as rectangles, are modeled via element *rect*. Synchronous and asynchronous hierarchical transitions have specific appearance. The *arc* of the Petri Net graph is created by the help of element *path*. The shape of the arc vertex in graph depends on the arc type (normal, test, inhibitory). Individual types of vertex are defined through element *marker* and then referenced from the element *arc*. Arc expressions (Boolean-type in case of input arcs of the transition and assignment-type at output arcs), which are also included with the Object-Oriented Petri Net definition, are processed and then displayed as well. Generated SVG diagrams are then viewable with any web browser which supports SVG format, optionally relevant plug-in can be installed. The template also supports modular and hierarchical Petri nets as it is capable of creating several diagrams from one OPNML document. Each main subpage and each method subpage of each class of the net is generated as a separate graph and saved in separate SVG file. Individual diagrams can be interconnected via XLink links, so there is possible to shift among them by mouse. The XSLT transformation template is accessible on web page <https://akela.mendelu.cz/~petrj/opnml/>. XML schema of OPNML language and samples of SVG graphs created from OPNML document via XSLT template are exposed on the same page.

Tento příspěvek vznikl v rámci řešení výzkumného záměru VZ MSM 6215648904/03/03/04.

## LITERATURA

- JEDLIČKA, P., 2007: *Object Petri Net Markup Language* [online]. [2006–2007], poslední aktualizace 13. 10. 2007 [cit. 2007-10-20]. URL: <<https://akela.mendelu.cz/~petrj/opnml/>>.
- JEDLIČKA, P., 2007: XML formát pro záznam objektové Petriho sítě. In *Acta Universitatis Agriculturae et Silviculturae Mendelianae Brunensis*. 2007, sv. LV, č. 3, s. 47–55. ISSN 1211-8516.
- MARTINÍK, I., 1999: *Metodologie tvorby objektově-orientovaných programových systémů s využitím teorie objektových Petriho sítí*. 1. vyd. Ostrava: VŠB-TU, 1999. 218 s. Disertační práce.
- STEHRNO, C., 2004: *PEP – Programming Environment based on Petri Nets*. [online] Poslední aktualizace 27. 10. 2004. [cit. 2007-10-17]. URL: <<http://theoretica.informatik.uni-oldenburg.de/~pep/>>.
- W3C – WORLD WIDE WEB CONSORTIUM *Scalable Vector Graphics (SVG) Full 1.2 Specification*. [online] Ver. 1.2, poslední aktualizace 13. 4. 2005. [cit. 2007-10-18]. URL: <<http://www.w3.org/TR/SVG12/>>.
- W3C – WORLD WIDE WEB CONSORTIUM *XSL Transformations (XSLT) Version 2.0* [online] Ver. 2.0, poslední aktualizace 23. 1. 2007. [cit. 2007-10-14]. URL: <<http://www.w3.org/TR/2007/REC-xslt20-20070123/>>.
- WEBER, M., 2006: *Petri Net Markup Language*. [online] poslední aktualizace 10. 1. 2006. [cit. 2007-10-11]. URL: <<http://www2.informatik.hu-berlin.de/top/pnml/about.html>>.

Adresa

Ing. Petr Jedlička, Ph.D., Ústav informatiky, Mendelova zemědělská a lesnická univerzita v Brně, Zemědělská 1, 613 00 Brno, Česká republika

