

Technická univerzita v Liberci
Fakulta přírodovědně-humanitní a pedagogická

VYUŽITÍ PETRIHO SÍTÍ VE VÝUCE

Závěrečná práce programu DVPP

Liberec 2012

Autor

Vedoucí závěrečné práce

Mgr. Jan Berki

Ing. Jindra Drábková, Ph.D.

Katedra: Katedra aplikované matematiky

Studijní program: Další vzdělávání pedagogických pracovníků

Studijní obor: Rozšiřující studium informatiky střední školy

VYUŽITÍ PETRIHO SÍTÍ VE VÝUCE USING PETRI NETS IN TEACHING

Závěrečná práce: 11–FP–KAP– 0038

Autor:

Mgr. Jan BERKI

Podpis:

Vedoucí práce: Ing. Jindra Drábková, Ph.D.

Počet

stran	grafů	obrázků	tabulek	pramenů	příloh
43	0	41	1	12	0

Liberec, 5. 9. 2012

TECHNICKÁ UNIVERZITA V LIBERCI

FAKULTA PŘÍRODOVĚDNĚ-HUMANITNÍ A PEDAGOGICKÁ

Oddělení dalšího vzdělávání

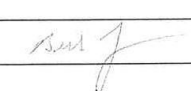
ZADÁNÍ ZÁVĚREČNÉ PRÁCE

(pro program Dalšího vzdělávání pedagogických pracovníků)

Kandidát:	Mgr. Jan Berki
Vzdělávací program:	Rozšiřující studium informatiky pro střední školy
Název ZP:	Využití Petriho sítí ve výuce
Název ZP v angličtině:	Using Petri Nets in Teaching
Vedoucí práce:	Ing. Jindra Drábková, Ph.D.
Termín odevzdání:	srpen 2011
Kód ZP:	11-FP-KAP-0038

V Liberci dne 10. 3. 2011

		
děkan		garant kurzu

Převzal (kandidát):	JAN BERKI		
Datum:	9. 5. 2011	Podpis:	

Název ZP:	Využití Petriho sítí ve výuce
-----------	-------------------------------

Vedoucí práce:	Ing. Jindra Drábková, Ph.D.
----------------	-----------------------------

Cíl:	Autor provede zhodnocení možností implementace Petriho sítí do vyučovacího procesu a připraví vzorové příklady, na nichž by se dala vyučovat algoritmizace pomocí Petriho sítí.
------	---

Požadavky:	<ol style="list-style-type: none"> 1. Provést rešerši k tématu Petriho sítě. 2. Navrhnout možnosti využití Petriho sítí v rámci výuky jak na základní, tak střední škole. 3. Vytvořit alespoň tři příklady pro výuku algoritmizace pomocí Petriho sítí.
------------	--

Literatura:	<p>Schubert, S., Schwill, A. <i>Didaktik der Informatik</i>. 1. vydání. Mnichov: Elsevier, 2004. ISBN 3-8274-1382-6.</p> <p>Kochaničková, M. <i>Petriho síť</i>. [online]. Olomouc, 2008. Dostupné z http://phoenix.inf.upol.cz/esf/ucebni/petriho_site.pdf.</p> <p>Balda, P. <i>Informační a řídicí systémy I. – Úvod do Petriho sítí</i>. [online]. Plzeň. Dostupné z http://vendulka.zcu.cz/Download/Free/IRS1/IRS1-01_Petriho_site.pdf.</p>
-------------	---

Prohlášení

Byl jsem seznámen s tím, že na mou závěrečnou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé závěrečné práce pro vnitřní potřebu TUL.

Užiji-li závěrečnou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Závěrečnou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím závěrečné práce a konzultantem.

Datum

Podpis

Děkuji tímto dr. Drábkové za podněty pro zlepšení úrovně mé závěrečné práce a za čas, který mi věnovala. Dále děkuji doc. Vaníčkovi za to, že mě přivedl k přečtení knihy Didaktik der Informatik. V neposlední řadě musím poděkovat rodině a spolupracovníkům, kteří mi umožnili svým pochopením a tolerancí práci napsat.

Abstrakt

Práce se nejprve zaměřuje na vymezení pojmu modelování v rámci oboru informatika. Popisuje různá pojetí modelování a jejich nástroje, zvláště pak Petriho sítě. Autor nastiňuje možnosti využití Petriho sítí ve výuce algoritmizace na základní či střední škole. Pro názornost jsou v práci umístěny tři rozsáhlejší příklady včetně didakticko-metodických komentářů a základních (atomických) úloh.

Klíčová slova

modelování, algoritmizace, Petriho sítě, informatika

Abstract

First, thesis focuses on the definition of modelling in computer science. It describes various concepts and modelling tools, especially Petri nets. Author outlines the possibilities of using of Petri nets in teaching of algorithms at secondary schools. There are located three extensive examples in the thesis for illustration, including didactic-methodical comments and basic (atomic) tasks.

Keywords

modelling, algorithms, Petri nets, informatics

OBSAH

1	Úvod	10
2	Vymezení pojmů	11
2.1	Model	11
2.1.1	Grafy a stromy	13
2.1.2	Automaty	15
2.2	Modelování	16
2.3	Specifikace problému	18
3	Petriho síť	20
3.1	Varianty Petriho sítí	22
3.2	Příklady modelů	23
3.3	Nástroje pro modelování Petriho sítí	26
4	Výuka algoritmizace a modelování	28
4.1	Ukotvení v kurikulárních dokumentech	28
4.2	Inspirace využití Petriho sítí ve výuce	30
5	Příklady	33
5.1	Domácí rodinný úklid	34
5.2	Tvorba informačních tabulí	36
5.3	Železniční stanice	38
6	Závěr	41
	Použitá literatura	43

SEZNAM OBRÁZKŮ

Obrázek 1 – Apostelův model tvorby modelu	12
Obrázek 2 – Ikonický model sítě německých dálnic	12
Obrázek 3 – Struktura molekuly znázorněná grafem	13
Obrázek 4 – Model pokrytí televizním signálem	14
Obrázek 5 – Rovinné grafy typu K_4 , K_5 a $K_{3,3}$ (zleva)	14
Obrázek 6 – Morseova abeceda jako binární strom	15
Obrázek 7 – Model části automatu na nápoje.....	15
Obrázek 8 – Jazyky Petriho sítí a Chomského hierarchie	20
Obrázek 9 – Základní sada ikon Petriho sítí.....	21
Obrázek 10 – Základní sekvence ikon.....	21
Obrázek 11 – Model silničního úseku	23
Obrázek 12 – Model světelné křižovatky	23
Obrázek 13 – Model letiště.....	24
Obrázek 14 – Část provozu továrny	24
Obrázek 15 – Situace v továrně znázorněna pomocí Petriho sítě.....	25
Obrázek 16 – Počáteční značení.....	25
Obrázek 17 – Situace po první změně stavů.....	25
Obrázek 18 – Situace po druhé změně stavů.....	25
Obrázek 19 – Situace po třetí změně stavů.....	25
Obrázek 20 – „Koncová“ situace	26
Obrázek 21 – Výběr aplikací pro modelování Petriho sítí	27
Obrázek 22 – Model výpočtu výrazu $(a - b) : (c + d)$	30
Obrázek 23 – Model logických operací.....	31
Obrázek 24 – Model souběhu procesů	31
Obrázek 25 – Model asynchronního zasílání zpráv.....	32
Obrázek 26 – Model vypracování domácího úkolu.....	32
Obrázek 27 – Model domácího úkolu v Petriho sítích	32
Obrázek 28 – Model jedné domácí činnosti	34
Obrázek 29 – Model sekvence Janových úkolů	34
Obrázek 30 – Model činností Jana i Lenky	34
Obrázek 31 – Model navázání činností	35
Obrázek 32 – Finální model domácího úklidu	35

Obrázek 33 – Základní model jedné činnosti	36
Obrázek 34 – Model smyčky fotografování série	36
Obrázek 35 – Model počátku větvení a paralelních dějů	37
Obrázek 36 – Neodladěný návrh výsledného modelu	37
Obrázek 37 – Model nádraží	38
Obrázek 38 – Model jednokolejného nádraží v jednom směru	38
Obrázek 39 – Model jednokolejného obousměrného nádraží	39
Obrázek 40 – Model jednokolejné trati se signálem „volno“	39
Obrázek 41 – Model jednokolejné trati se „signálem“ volno v obou směrech	40
Obrázek 42 – Model jednokolejné trati v obou směrech bezpečný	40

1 ÚVOD

Jedním z úkolů učitele je pomáhat žákovi poznávat svět kolem sebe. Aby byl žák připraven na život v něm, musí porozumět jeho zákonitostem. Nezřídka k tomuto poznávání používá společně s učitelem různé modely. Každá vědní disciplína upřednostňuje své nástroje, jak je vytvořit. Matematika či fyzika dávají přednost číslům a vzorcům, chemie kádinkám a zkumavkám, historie psanému slovu či dřevěným replikám, anatomie plastovým orgánům, biologie pokusům se zvířaty atd. Informatika si obraz světa obvykle naprogramuje.

Množství aplikací, nástrojů a jazyků, které lze k modelování použít, je mnoho. Běžně vidíme ve školách děti kreslit různé stavby v grafickém editoru. Na některých školách se potkáme s programy, pomocí kterých si děti vytvoří svou hru anebo rozpohybují svou postavu. Součástí vzdělávací oblasti Informatika (resp. ICT) je také tematická oblast algoritmizace. Na středních školách můžeme tedy narazit už na modelování pomocí programovacích jazyků – ať již strukturované, objektově-orientované nebo dokonce neprocedurální. Liší se svou přívětivostí běžnému uživateli, syntaxí a také tím, co umí vymodelovat – tedy výstupem. „Klasické“ programovací jazyky se díky množství vyhrazených slov a syntaxi stávají pro žáky obtížnějším učivem samy o sobě, natož pak jejich využití pro znázornění reálného světa, byť zjednodušeného.

Mezi všemi nástroji si vybíráme jednak podle toho, k čemu daný model potřebujeme, a pak podle toho, jak obtížné je pro nás daný nástroj prakticky používat. Musíme ovšem mít z čeho vybírat. Cílem této práce je poskytnout inspiraci pro využití jednoho z nástrojů modelování – Petriho sítí. Impulzem pro tento výběr byla publikace profesora Schuberta a profesorky Schwill, kteří se zabývají didaktikou informatiky.

Pomocí dalších tuzemských i zahraničních zdrojů se nejprve společně seznámíme se základy teorie Petriho sítí. Teorie Petriho sítí je díky jejich dalšímu rozvoji poměrně obsáhlá. Bude tedy nutné vybrat výsek vhodný pro implementaci do výuky. Výsledkem práce budou návodné příklady využívající modelování pomocí Petriho sítí vhodné k výuce na základních či středních školách – od nejjednodušších (atomárních) po rozsáhlejší. Každý z příkladů je zasazen do didaktického kontextu informatiky. Jednotlivé úlohy využívají jako motivační prvky propojení jednak s jinými předměty (resp. jejich učivem) a jednak s reálnými situacemi. Ostatně pro modelování reálných situací mají Petriho sítě mimo jiné sloužit.

2 VYMEZENÍ POJMŮ

Snaha člověka vytvářet si modely reálného světa je známa již od pravěku. Nástěnné malby v jeskyních zachycovaly průběh lovu. Hliněné Venuše zase připomínaly přednosti správné ženy a matky. Malířství a sochařství dávalo vzpomenout na hrdinské činy v různých dobách. Snem většiny kluků (i když už vyrostou) je velký a realistický model kolejiště s vláčky, budovami a lidmi. Nebo si představí funkční modely letadel. Své modely používají všechny obory, všechna odvětví; každý člověk se s nimi setkává. Co to ale model je? A jak ho vnímá informatika?

Úvahy nad modelováním v informatice se rozvíjejí v souvislosti se snahou znázornit reálný svět do kybernetického. Tyto snahy jsou vedeny například potřebou simulace některých přírodních jevů pro vědecké pokusy a podobně, neboť jejich reálné pozorování je nemožné či životu nebezpečné

2.1 Model

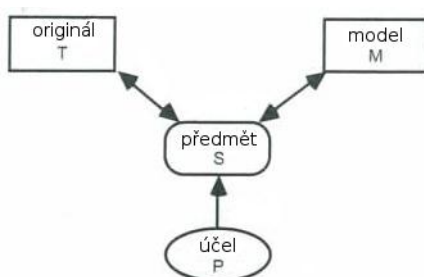
Tento pojem může mít několik lexikálních významů, a to i v samotné informatice. Jak bylo naznačeno již v úvodní části práce i kapitoly, modelování je vedeno snahou znázornit věci kolem nás. Ve vědních oborech je tato snaha motivována studiem a zkoumáním předmětů reálného prostředí. Nemusí se jednat pouze o předměty již existující. Modely se používají i pro studium předmětů, o jejichž existenci se uvažuje (Křivý a Kindler 2001, str. 10).

Původním významem slova model byla předloha. Dnes se používá pro předmět v reálném světě spíše pojem originál (Křivý a Kindler 2001, str. 13–14; Schubert a Schwill 2011, str. 135). Při vytváření modelu je důležité uvědomovat si aspekty předmětu. Ten existuje vždy v nějakém prostředí. Je ovlivňován dalšími předměty či vlivy tohoto prostředí. Obvykle je velmi obtížné zahrnout všechny tyto činitele do plánovaného modelu, proto dochází k abstrakci, při níž se situace zjednodušuje. Některé prvky prostředí jsou zanedbány. Je ale třeba mít na zřeteli účel, pro který je model vytvářen. Nemůžeme pominout aspekty, které by mohly ovlivnit validitu interpretace výsledků pozorování.

Tuto abstrakci nazývají Křivý a Kindler (2001, str. 10) systémem. Dále rozlišují, zda zanedbávanou složkou reality je čas. Pokud ano, pak označují systém jako statický, pokud ne, pak se jedná o systém dynamický. Model je podle Křivého a Kindlera (2001, str. 13) analogií mezi dvěma systémy. Podle Voráčkové, Pěničky a Veselého (2008, str. 10) je model nástrojem poznávání reality, přičemž mezi originálem a modelem není dokonalá

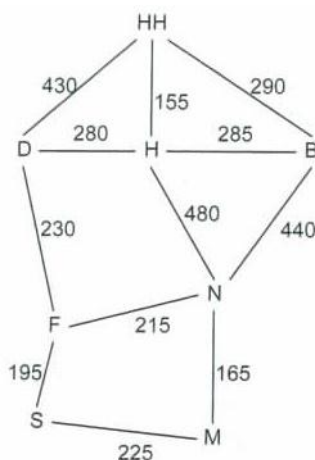
shoda. Systém, ve kterém se předmět nachází, vymezují jako fyzicky či myšlenkově ohraničenou část prostředí.

Vztahy mezi jednotlivými pojmy popisuje Apostelův relační model (Schubert a Schwill 2011, str. 135) na obrázku 1.



Obrázek 1 – Apostelův model tvorby modelu

Podle důrazů na fyzickou stránku modelu a jeho podobu rozlišujeme dva základní typy. Ikonické modely využívají obrázků, které znázorňují samotné předměty i vztahy mezi nimi. Příkladem z informatiky jsou různé stromy, strukturogramy, vývojové diagramy a grafy. Oproti tomu symbolické modely popisují předmět pomocí jazyka (přírozeného, resp. programovacího). Příkladem z informatiky jsou různé programy či gramatiky (Schubert a Schwill 2011, str. 139). Jak je patrné, ikonické modely upřednostňují fyzickou podobu modelu. Do této kategorie by patřily i zmiňované modely vláčků (Voráčková, Pěnička a Veselý 2008, str. 10). Názorný rozdíl mezi ikonickým a symbolickým modelem si ukážeme na modelu části sítě německých dálnic (Schubert a Schwill 2011, str. 139–140), kde písmena nahrazují jména měst a čísla udávají vzdálenost mezi nimi právě po dálnici. Čára nahrazuje existující silniční spojení mezi městy.



Obrázek 2 – Ikonický model sítě německých dálnic

Tutéž reálnou situaci můžeme vyjádřit také symboly. Písmena a čísla mají shodný význam. Množina X obsahuje města našeho modelu. Množina Y obsahuje cesty mezi sousedními městy, přičemž každé město sousedí i samo se sebou. Funkce d přiřazuje cestě její délku.

$$X = \{HH, D, H, B, F, N, S, M\},$$

$$Y = \left\{ \{HH, D\}, \{D\}, \{H\}, \{B\}, \{F\}, \{N\}, \{S\}, \{M\}, \{HH, D\}, \{HH, H\}, \{HH, B\}, \right. \\ \left. \{H, D\}, \{H, N\}, \{H, B\}, \{D, F\}, \{B, N\}, \{F, N\}, \{F, S\}, \{N, M\}, \{S, M\} \right\}, Y \subseteq 2^X$$

$$d: Y \rightarrow \mathbb{N}_0,$$

$$d(\{HH, D\}) = 430, d(\{HH, H\}) = 155, d(\{HH, B\}) = 290, d(\{H, D\}) = 280,$$

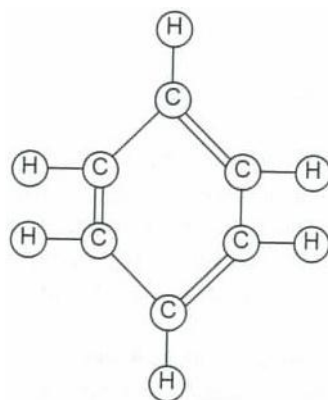
$$d(\{H, N\}) = 480, d(\{H, B\}) = 430, d(\{D, F\}) = 230, d(\{B, N\}) = 440,$$

$$d(\{F, N\}) = 215, d(\{F, S\}) = 195, d(\{N, M\}) = 165, d(\{S, M\}) = 225,$$

$$d(z) = 0, \forall \text{ ostatní } z \in Y.$$

2.1.1 Grafy a stromy

Jedním z typických příkladů ikonického modelu jsou grafy. Nejedná se však o grafy, s jakými se žáci setkávají v hodinách matematiky či při výuce zpracování dat v tabulkovém editoru. Základní charakteristikou grafů v informatice jsou uzly a hrany, které je spojují. Přesto, že se bavíme o grafech používaných v informatice, setkáme se stejným typem také v jiných vědních oborech. V chemii mají grafy podobu struktury molekul. Všimněme si, že na obrázku 3 (Schubert a Schwill 2011, str. 253) mohou být některé hrany násobné. S tímto jevem se setkáme později i u Petriho sítí. Na rozdíl od Petriho sítí jsou hrany v tomto grafu neorientované.

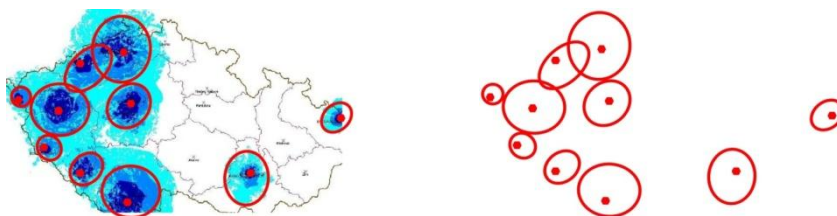


Obrázek 3 – Struktura molekuly znázorněná grafem

Tvorba grafů s žáky nemusí být nijak náročná. Na následujícím příkladu si názorně ukážeme, jak je možné využít běžný grafický či textový editor, pro tvorbu jiného typu grafu např. v zeměpise. Na internetu si najdeme mapu pokrytí televizním signálem*

* V našem případě je mapa stažena z <http://satcom.cz/image.php?file=475.jpg>.

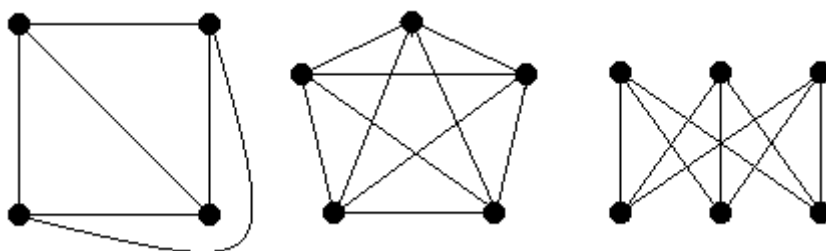
a vložíme ji jako obrázek do editoru, který umožňuje vrstvit obrázky na sebe. Potom zvolíme vhodnou ikonu pro vysílač a elipsu pro dosah vysílače. Nakonec odstraníme mapu z podkladu a zůstane nám pouze model. Zajímavé na tomto příkladu je, že originálem pro náš model je původně také model – mapa. Z obrázku 4 je také patrné, jak dochází ke zkreslení při modelování.



Obrázek 4 – Model pokrytí televizním signálem

Za poznámku stojí ještě jedna zvláštnost tohoto modelu. Tento graf nemá hrany, které by spojovaly jednotlivé uzly.

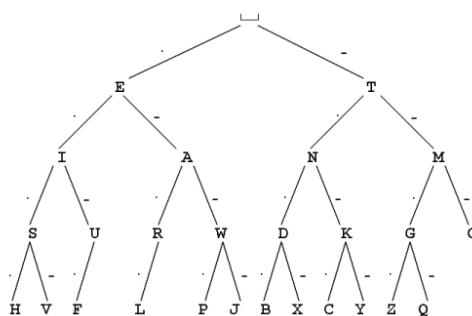
Typickým příkladem grafů v informatice jsou rovinné grafy. Jako příklad nám poslouží grafy typu K_4 , K_5 a $K_{3,3}$, které se používají pro výuku rovinných grafů* (viz obrázek 5). Všimněme si také, že ani na tomto grafu nejsou hrany orientované.



Obrázek 5 – Rovinné grafy typu K_4 , K_5 a $K_{3,3}$ (zleva)

Ani v jednom z uvedených příkladů není mezi jednotlivými uzly hierarchie, obdobně tomu je také v Petriho sítích. Někdy ale potřebujeme modelovat strukturu, v níž jeden prvek je nadřazen druhému. Dalo by se říci, že takovým grafem, který nám toto rozlišení umožňuje, je strom. Pěkným příkladem binárního (se dvěma větvemi) stromu, který uvádí i Schubert a Schwill (2011, str. 257), je Morseova abeceda. K šifrování písmen anglické abecedy používá různé kombinace teček a čárek. Uzly stromu (viz obrázek 6) jsou uspořádány tak, že levá větev vždy představuje tečku a pravá čárku.

* Rovinný graf je takový graf, který se dá v jedné rovině nakreslit tak, že se žádné jeho hrany nekříží. K_4 je úplný graf o 4 uzlech, které jsou vzájemně propojeny hranami. K_4 je rovinný graf. K_5 je úplný graf o 5 uzlech a není rovinný. $K_{3,3}$ je bipartitní graf a také není rovinný.

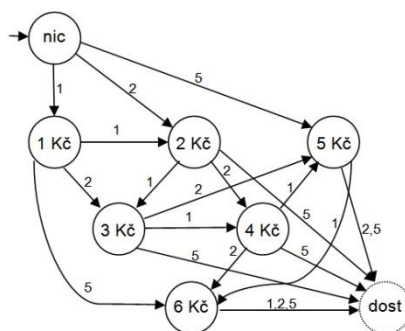


Obrázek 6 – Morseova abeceda jako binární strom*

Na zvoleném obrázku je patrné, že není nutné dodržovat jako ikonu uzlu kružnici. Doporučujeme ale, především s mladšími žáky, při osvojování kompetencí vytváření modelů pomocí grafů tuto konvenci dodržovat, alespoň na začátku.

2.1.2 Automaty

Modelem pro děje či postupy jsou v informatice také automaty. Konečné automaty jsou definovány pomocí abecedy (vstupní symboly), množiny stavů (uzlů), z nichž některé jsou počáteční (vstupy) a některé koncové (výstupy). Pravidla pro přechody mezi stavy definuje tzv. přechodová funkce. Určuje odkud, za jakých podmínek a kam se můžeme v automatu pohybovat. Jejich ikonická znázornění vycházejí de facto z grafů. Oproti příkladům uváděným výše se však liší v hranách. Z povahy přechodové funkce potřebujeme určit směr přechodu, graf tedy musí být orientovaný. Dále je určena podmínka přechodu znakem abecedy, ten se píše nad hranu. Ve většině škol se žáci setkávají s automaty na nápoje. Aby automat vydal pití, musí se do něj vházet mince v dostatečné hodnotě. Pro zjednodušení modelu vezměme automat, který přijímá pouze mince s nominální hodnotou 1 Kč, 2 Kč a 5 Kč. Nápoj stojí 7 Kč. Na obrázku 7 je model kontroly sumární hodnoty pomocí konečného automatu.



Obrázek 7 – Model části automatu na nápoje

* Obrázek je stažen z http://www.fd.cvut.cz/personal/xfabera/Y2PJ/doplneni/morse_strom.png.

S žáky je nejlepší postupovat tak, aby určili vždy postupně všechny situace vhozením akceptovaných mincí, a tím postupně budovali stavy. Tedy odpovídáme na otázku: Jaká bude sumární hodnota, když na začátku hodíme korunovou minci? Jaká, když hodíme dvoukorunu, a jaká, když pětikorunu? Potom postoupíme do stavu, který odpovídá hodnotě 1 Kč, a postup zopakujeme. Žáci postupně přicházejí na to, že některé stavy už v grafu mají.

Na podobném principu fungují i Petriho sítě. Ty vznikly rozšířením modelovací schopnosti právě konečných automatů (Češka 1994, str. 9). Podrobněji se jejich analogii věnuje kapitola 3.

2.2 Modelování

Dalo by se tedy jednoduše říci, že modelování je vytváření modelu. *Podstatou modelování ve smyslu výzkumné techniky je náhrada zkoumaného systému jeho modelem (přesněji: systémem, který jej modeluje), jejímž cílem je získat pomocí pokusu s modelem informaci o původním zkoumaném systému* (Křivý a Kindler 2001, str. 15). Modelování dynamického systému je označováno jako simulace (Křivý a Kindler 2001, str. 17).

Modelování využívá také např. fyzika, ale používá odlišný postup. Ukažme si to na příkladu volného pádu. Fyzika si situaci definuje pomocí formulí (vzorců) a kámen, který padá, redukuje na jakousi konstantu. Informatika na rozdíl od fyziky kámen neopomíjí, pro ni je to jeden ze základních prvků (Schubert a Schwill 2011, str. 138) a definuje jeho podrobné vlastnosti.

fyzika

- rychlost $v(t) = at$
- pozice $s(t) = \frac{1}{2}at^2$
- energie $E(t) = \frac{1}{2}m(v(t))^2$
- ...

informatika

kámen = object
 leží na ...;
 je šedý;
 může se s ním házet;
 ...
 end.

Model prostředí lze také vytvořit jako program. Pro jeho tvorbu je zapotřebí znalost některého z programovacích jazyků. Pro žáky se nabízí analogie s cizincem. Abychom se společně domluvili, musíme znát společný jazyk. Mezinárodním jazykem je např. angličtina. A pro dorozumění s počítačem při vytváření modelu ve formě programu používáme některý z programovacích jazyků, např. C. Programovací jazyk je také mezinárodní, protože mu budou rozumět počítače bez ohledu na zemi, ve které se nacházejí. Stejně jako angličtina, má i C svoje (vyhrazená) slova (příkazy) a svá pravidla psaní (syntax).

Schubert a Schwill (2011, str. 148) definují tři základní programovací styly či paradigmaty – imperativní, funkcionální a predikativní – a ty pak dále dělí na dvě kategorie – procedurální a deklarativní. Objektově-orientované programování vyčleňují jako čtvrté paradigma*. U některých prostředí je možné nalézt objektově-orientovanou nadstavbu původně strukturovaného programovacího jazyka – např. Lazarus a Pascal.

Všechny výše popsané programovací jazyky seskupují vzájemně podobné předměty podle určitých pravidel do struktur, kterým se říká datové typy (Schubert a Schwill 2011, str. 144). Jedná se víceméně o standardizovaný model určitých předmětů pro symbolické modelování. Datový typ je tedy dán množinou přípustných znaků/symbolů a operací, které lze na této množině realizovat. Rozlišujeme základní datové typy a strukturované. Strukturované se skládají ze základních. V programech, nezávisle na syntaxi jazyka, se obvykle pracuje např. s následujícími datovými typy:

- znak $char = \{a, b, c, \dots\}$
- text (posloupnost znaků) $string = (char^*, \{\circ\})$
- čísla $integer = (\mathbb{Z}, \{+, -, \dots\})$
- logika $boolean = (\{0,1\}, \{and, or, not\})$

Jak si ukážeme dále, základní (P/T) Petriho síť jednotlivé datové typy nerozlišují, vyšší (tzv. barevné Petriho síť) je ale rozlišovat mohou.

Pokud by v rámci modelování existovaly pouze standardizované datové typy, byli bychom značně omezeni při řešení některých problémů. Z tohoto důvodu se kromě datových typů ustálily i tzv. konstruktory, které umožňují definovat další typy. Mezi nejznámější konstruktory patří na příklad definování typu výčtem jeho prvků nebo omezením existujících množin čísel danou vlastností.

* Zajímavé srovnání lze nalézt na wikipedii pod hesly programovací jazyk a programovací paradigmaty.

V souvislosti se vzděláváním nesmíme zapomenout na programovací jazyky označované jako „dětské“. Jedná se např. o jazyk Karel, (Imagine) Logo, Baltík. Ty byly vyvinuty právě pro účely výuky algoritmizace a programování. Programovací jazyk Pascal sice nemá prostředí a syntax úplně zjednodušené, ale i on vznikl se stejným cílem.

2.3 Specifikace problému

Jak je uvedeno již v předchozích kapitolách, modely se využívají pro zkoumání daných objektů. Každé řešení problému začíná (nejen) v informatice jeho definováním. Tento proces označujeme jako specifikaci a je prováděn s ohledem na účel modelování (Schubert a Schwill 2011, str. 141–142). Specifikace, někdy označovaná jako funkcionální, vychází z matematického jazyka a její struktura vypadá následovně:

spec $f: X \rightarrow Y, f(x) = y$, kde	definice funkce f , kde X jsou možné vstupy a Y příslušné výstupy
pre $P(x)$	předběžná podmínka
post $Q(x, y)$.	následná podmínka

Z tohoto důvodu se ve specifikacích problémů, i pro modelování v Petriho sítích, můžeme často setkat s běžnými matematickými prvky jako jsou množiny – $\mathbb{N}, \mathbb{Z}, \mathbb{R}$, konstanty – $7, \pi$, proměnné – $a \in \{0,1\}$, kvantifikátory – $\forall x \geq 3, \exists y < 12$, operace – $a + b, A \cup T$ apod. S žáky na základní škole je lépe zůstat u specifikace slovní. Jednak matematický zápis vyžaduje větší abstrakci, pak také žáci v nižších ročnících nemají dostatečné matematické znalosti. Přesto je třeba dodržovat jistou míru formalismu. K využívání programu a následnému vyhodnocování nemusíme vždy vědět, jak program funguje uvnitř. Je ale nutné znát specifikaci problému, který daný program řeší. Po vyhotovení programu je možné vhodně zvoleným příkladem a formální metodou verifikace dokázat, zda program vyhovuje zadané specifikaci (Schubert a Schwill 2011, str. 144).

Vraťme se k modelu části sítě německých dálnic. Ukažme si na problému hledání nejkratší cesty jeho možnou formální specifikaci (Schubert a Schwill 2011, str. 143).

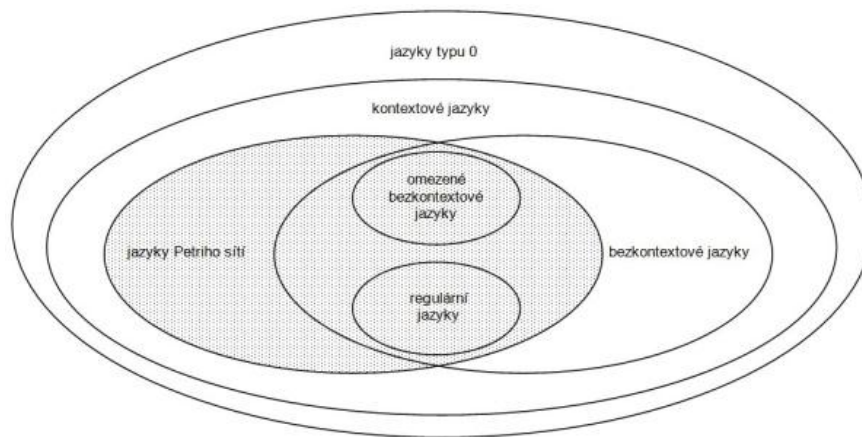
$$\begin{aligned} &\text{spec } c_n: 2^X \rightarrow X^* \times \mathbb{N}_0, c_n(M) = (c, d), \text{ kde} \\ &\text{pre } 1 \leq |M| \leq 2 \\ &\text{post } M = (s, s') \rightarrow c = [s = x_1, \dots, x_n = s'], n \geq 1, \\ &d = \sum_{i=1}^{n-1} d(\{x_i, x_{i+1}\}), \forall w' \in X^*, w' = [s = y_1, \dots, y_m = s'], d \leq \sum_{i=1}^{m-1} d(\{y_i, y_{i+1}\}) \end{aligned}$$

Přeloženo do „přirozeného“ jazyka můžeme specifikaci číst následovně:

Zavedeme funkci nejkratší cesta (c_n) a určíme, že vybíráme dvě města z našeho seznamu (2^X). Výsledkem bude popis cesty pomocí měst, kterými projíždíme (X^*, c), a její délkou (N_0, d). Předpokladem (*pre*) je, že vybíráme cestu pro jedno či dvě města. Následnou podmínkou (*post*) je, že cesta (c) je posloupnost cest mezi projížděnými městy a že délka výsledné cesty (d) je součtem délek projížděných cest a také že jakákoli jiná cesta (w') je delší.

3 PETRIHO SÍŤ

Autorem Petriho sítí je Carl Adam Petri, který je definoval v rámci své disertační práce v roce 1962. Jsou analogií konečných automatů. Jak se tedy liší? Co přináší Petriho síť nového? Je to především uchopení podmínky a značení místa (Češka 1994, str. 9). Stejně jako mají vztah Petriho sítě a automaty, můžeme srovnat jazyky Petriho sítí s Chomského hierarchií jazyků. Jak vidíme na obrázku 8 (Češka 1994, str. 69), Petriho sítě mají stejnou modelovací schopnost jako některé kontextové i jako některé bezkontextové jazyky.



Obrázek 8 – Jazyky Petriho sítí a Chomského hierarchie

Formálně je Petriho síť definována (Schubert a Schwill 2011, str. 269) jako čtveřice $P = (S, T, A, E)$, kde

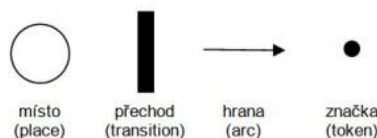
- S je neprázdná konečná množina míst/stavů (places),
- $T \cap S = \emptyset$ je neprázdná konečná množina přechodů (transitions),
- $A \subseteq S \times T$ je konečná množina hran směřujících ze stavů do přechodů (arcs),
- $E \subseteq T \times S$ je konečná množina hran směřujících z přechodů do stavů (arcs).

Balda (2007, str. 8) definuje Petriho síť jako pětici, přičemž zahrnuje do definice také váhovou funkci a označování. Češka (1994, str. 15) dokonce vymezuje šestici, neboť do definice zahrnuje i kapacitu místa. Tento zdánlivý rozpor vzniká pouze tím, že jednotliví autoři vycházejí pro definici z jiné úrovně Petriho sítí. Místu se říká vstupní (pre) podmínka, pokud od něj vede šipka k události. U výstupní (post) podmínky vede šipka od události k tomu místu.

Za výchozí považujeme první uvedenou definici. Postupně budeme budovat vyšší úrovně Petriho sítí přidáváním vlastností tak, aby se zvětšovala jejich modelovací schopnost. Ne

všechny uvedené úrovně jsou ale vhodné pro využití v rámci výuky na základní či střední škole, proto se některým budeme věnovat více a jiné jen zmíníme.

Pro znázorňování Petriho sítí budeme používat jejich ikonický model. Výhodou tohoto modelu je intuitivní uchopitelnost použitých symbolů. Stejně jako existují základní a vyšší úrovně Petriho sítí, existuje také základní a rozšířená sada ikon. Výhodou základního modelu pro žáky je, že obsahuje pouze čtyři prvky (viz obrázek 9). Z definice známe první tři prvky. Značka (token) nám umožňuje modelovat aktuální umístění v systému.

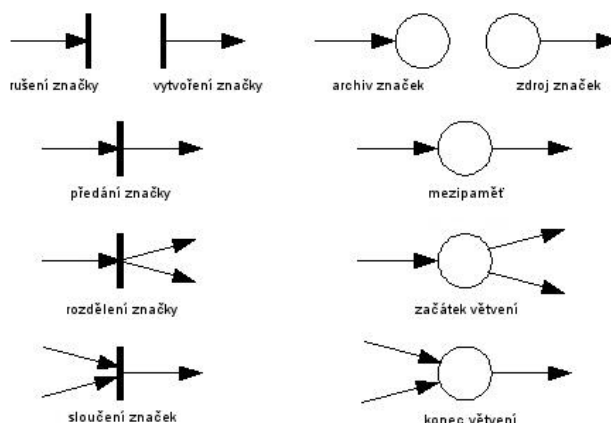


Obrázek 9 – Základní sada ikon Petriho sítí

Není ovšem možné ikony skládat libovolně. Při znázorňování Petriho sítí platí následující pravidla (Schubert a Schwill 2011, str. 268–269; Balda 2007, str. 9):

- Hrany spojují pouze místa s přechody, ne dvě místa či dva přechody.
- Místa mohou obsahovat libovolné množství značek (tokens).
- Ke změně místa (přes přechod) může dojít v libovolném čase, tedy i nikdy.
- V jeden okamžik se může pohybovat více značek.
- Při větvení může dojít k pohybu značky libovolným (možným) směrem.
- K přechodu může dojít jen v případě, že v každém z jeho vstupních míst je alespoň jedna značka.

Z těchto vlastností plyne podstatná schopnost Petriho sítí. Umožňují modelovat paralelní jevy (Balda 2007, str. 4). Také z těchto pravidel vyplývá, že sekvence ikon není náhodná. Některé sekvence mají svůj specifický význam (viz obrázek 10).



Obrázek 10 – Základní sekvence ikon

3.1 Varianty Petriho sítí

Nejjednodušší variantou jsou tzv. C/E (Condition/Event) Petriho sítě. Místa znázorňují logickou proměnnou. To znamená, že v každém místě může být nejvýše jedna značka. Nebo jinak řečeno, kapacita každého místa se rovná jedné (Češka 1994, str. 10). Tímto se zavádí pojem kapacita, který udává maximální počet značek na jednom místě. Ke změně stavu může tedy v C/E Petriho sítích dojít pouze, pokud jsou všechny vstupní podmínky přechodu splněny (označeny) a zároveň všechny jeho výstupní nesplněny (Voráčová, Pěnička a Veselý 2008, str. 14).

Základní variantou jsou tzv. P/T (Place/Transition) Petriho sítě. Pro tuto variantu jsou také zpracovávány úlohy v praktické části. V P/T Petriho sítích je kapacita místa neomezená. Často kapacitu modelovat potřebujeme, ale omezení kapacity na jednu značku nestačí. Symbolizuje-li značka např. počet aut na parkovišti (bez rozlišení konkrétních parkovacích míst), musí být kapacita vyšší. Omezení kapacity je první rozšířenou variantou P/T Petriho sítí. V případě, že se může vyskytovat více značek na jednom místě ve stejný okamžik, může se také stát, že v jeden okamžik více značek místo opouští. Místo, abychom kreslili dvě souběžné hrany, umístí se nad hranu číslo, které určuje, kolik značek se z místa bude odebírat. Toto číslo se nazývá ohodnocením hrany a nezvyšuje modelovací schopnost Petriho sítí, neboť se jedná pouze o jiný zápis násobných hran (Voráčová, Pěnička a Veselý 2008, str. 17). K dalším zjednodušení znázornění modelu je možné využít testovací hrany. Ty se znázorňují čárkovanou šipkou. Značí situaci, kdy přechod na svém výstupu rozděluje značku jak do výstupní podmínky, tak zpět do vstupní (Voráčová, Pěnička a Veselý 2008, str. 34).

Mezi další podtřídy P/T varianty patří stavové stroje, značené grafy, Petriho sítě s volným výběrem (Češka 1994, str. 70–73).

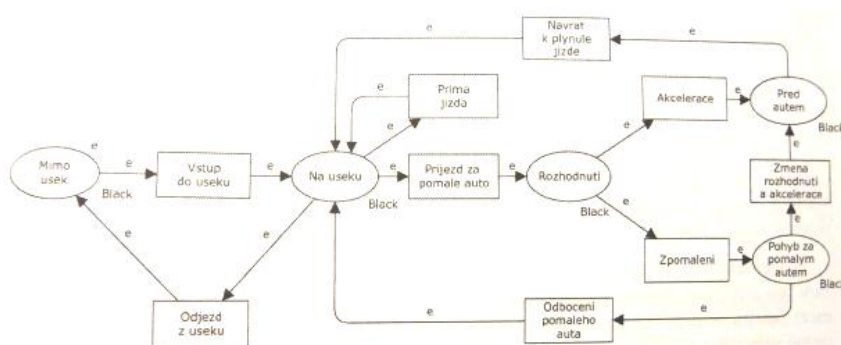
Rozšířením P/T Petriho sítí jsou tzv. inhibitory. Jedná se v podstatě o negativní pojetí testovacích hran, z čehož plyne, že inhibitory sice ovlivňují proveditelnost přechodu, ale nijak nemění značení uvnitř systému. Síla P/T Petriho sítí s inhibitorem odpovídá Turingovu stroji (Voráčová, Pěnička a Veselý 2008, str. 35; Češka 1994, str. 75). Potřebujeme-li ovlivňovat pořadí, ve kterém se provádějí jednotlivé přechody, přiřadíme jim nezáporné celé číslo. Tato hodnota se nazývá priorita. Přechod tedy může být proveden, jednak jsou-li splněny všechny vstupní podmínky, a jednak nemá-li žádný

z proveditelných přechodů vyšší prioritu. Modelovací síla je srovnatelná s P/T Petriho sítěmi s inhibitory (Voráčová, Pěnička a Veselý 2008, str. 38).

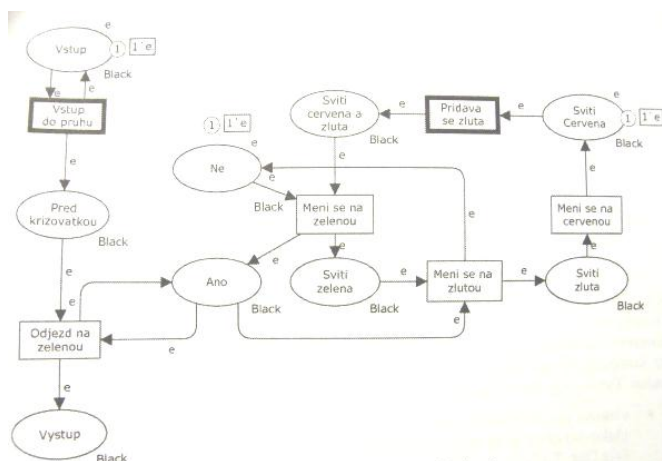
V některých situacích je potřeba v rámci modelu rozlišovat typy značek, např. jako datové typy. Z tohoto důvodu se odlišují značky barvami, a proto se tato nazývá varianta jako barevné Petriho sítě. Ty jsou společně s časovanými Petriho sítěmi řazeny do skupiny vyšších úrovní (Voráčová, Pěnička a Veselý 2008, str. 43–59; Češka 1994, str. 76–77). Zvláštní zmínku je třeba věnovat frontovým Petriho sítím, neboť k takové variantě bude směřovat úloha Tvorba informačních tabulí v kapitole 4. Ty umožňují nahradit místo frontou.

3.2 Příklady modelů

Následující kapitola je kompilací příkladů modelů využívajících Petriho sítě, na kterých je názorně vidět propojení s reálným světem. Obvykle jsou využívány různé varianty v dopravních modelech, jak vidíme na obrázcích 11, 12 a 13 (Voráčová, Pěnička a Veselý 2008, str. 99, 109 a 112). Tyto dopravní modely posloužily jako inspirace pro příklad Železniční stanice.

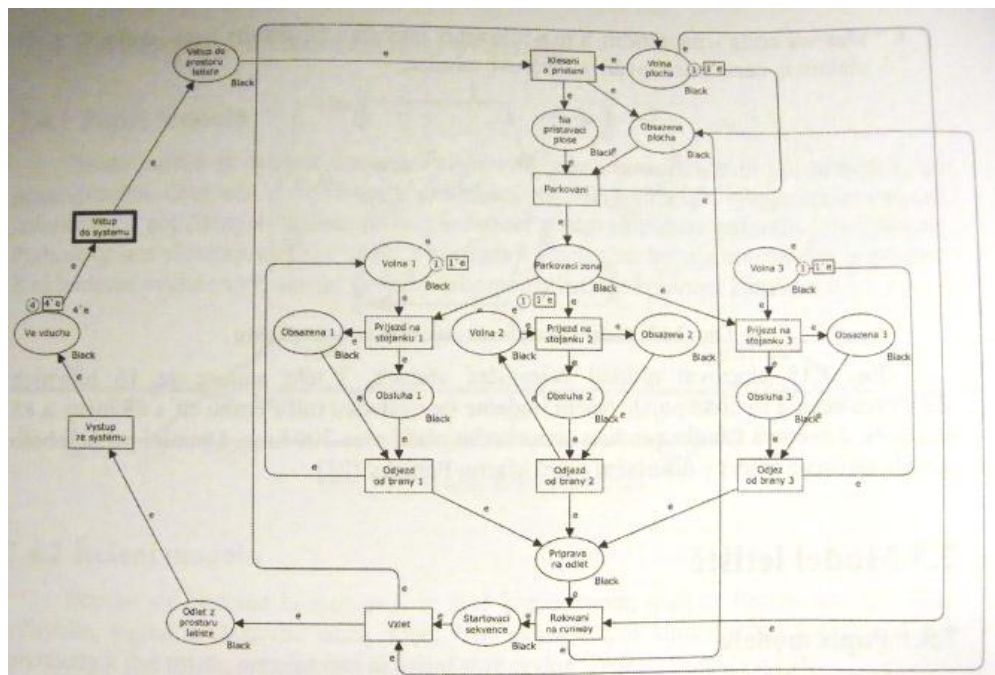


Obrázek 11 – Model silničního úseku



Obrázek 12 – Model světelné křižovatky

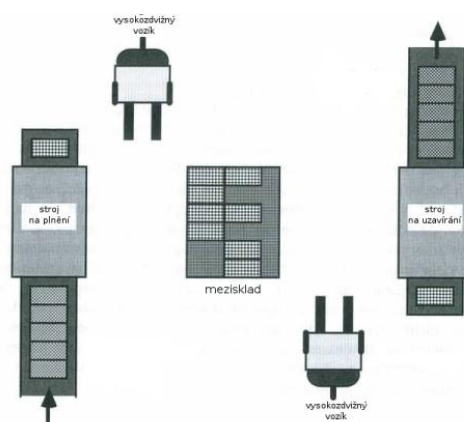
Na těchto modelech jsou popisy umístovány do míst a přechodů. My se přidržíme konvence umístování popisů pod či nad jednotlivé ikony. U rozsáhlejších modelů je model při dodržování konvence méně přehledný.



Obrázek 13 – Model letiště

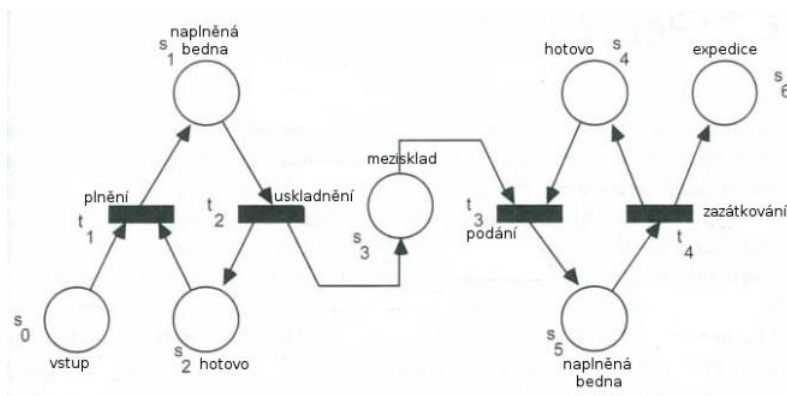
Model letiště byl vybrán pro demonstraci možné složitosti výsledných modelů. Přesto modely tvořené pomocí Petriho sítí zůstávají poměrně čitelné. Takové příklady je ovšem dobré žákům ukazovat pouze jako ilustrativní.

Příklad vhodný k modelování žáky střední školy najdeme v Schubert a Schwill (2011, str. 267–270). Fungování části provozu v továrně na výrobu nápojů si dokážou žáci představit (viz obrázek 14). Paralelnost dějů na linkách je také zjevná.



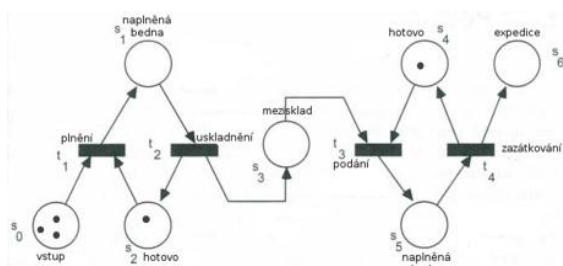
Obrázek 14 – Část provozu továrny

Na levé lince se lahve plní tekutinou, poté jsou vysokozdvížným vozíkem přemístěny do mezikladu. Následně jsou umístěny na linku, kde jsou zazátkovány a na konec expedovány. Při modelování pomocí Petriho sítí je nejprve nutné určit místa, přechody a hrany mezi nimi. Na následujícím modelu (viz obrázek 15) se nachází místo, které je pojmenováno jako hotovo. Jedná se o signál, že lahev byla naplněna a do stroje může další.

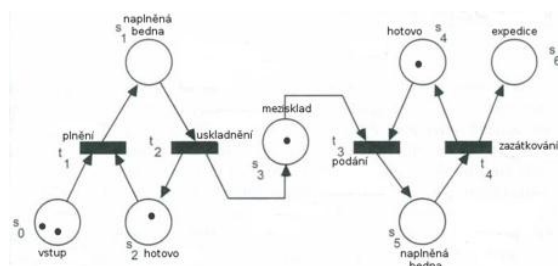


Obrázek 15 – Situace v továrně znázorněna pomocí Petriho sítě

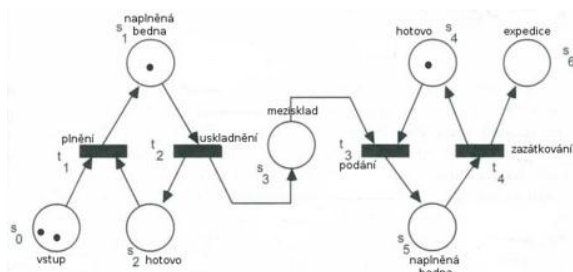
Simulaci běhu děje můžeme provádět například animací anebo sekvencí obrazů téhož modelu se změnou značení po provedení změny stavů. Na následující sekvenci (obrázek 16–20) je zajímavé, že značka symbolizuje dva různé objekty – lahev a signál. V tomto případě však není nutné využít barevné Petriho sítě.



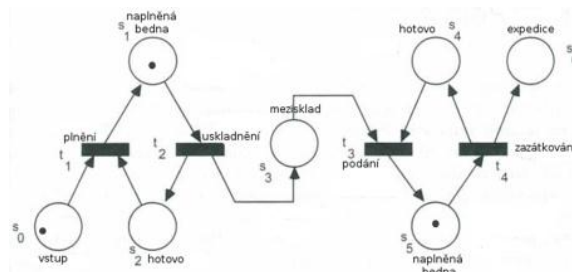
Obrázek 16 – Počáteční značení



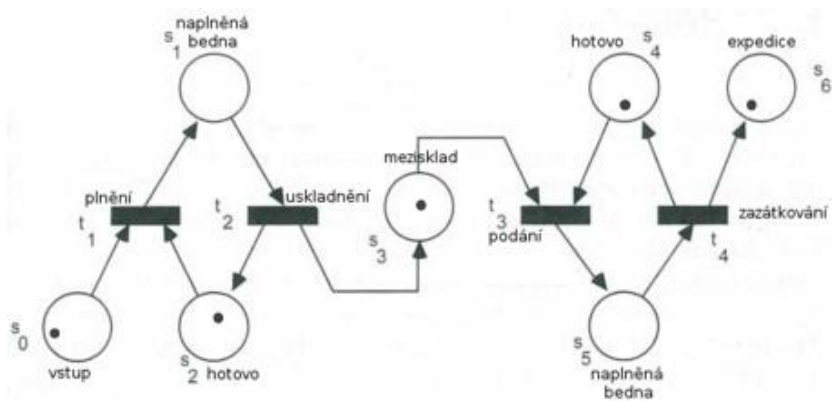
Obrázek 18 – Situace po druhé změně stavů



Obrázek 17 – Situace po první změně stavů



Obrázek 19 – Situace po třetí změně stavů



Obrázek 20 – „Koncová“ situace

3.3 Nástroje pro modelování Petriho sítí

Pro tvorbu modelů v Petriho sítích z povahy ikon postačují obyčejné kancelářské programy. Je například možno využít kreslení defaultně nainstalované v operačním systému. Vhodné jsou také freewarové aplikace jako Inkscape nebo Gimp. Dostačující a čitelné výsledky také poskytuje textový editor pomocí nástroje vkládání tvarů. Výhodné pro výuku je, pokud používaný nástroj umožňuje práce ve vrstvách. Každé značení po změně stavů je umístěno do vlastní vrstvy a jednoduchým přepínáním pohledů mezi vrstvami (skrývání a zobrazování) se ukazují sekvence situací. Při výuce je možné využít barevného rozlišení značek v jednotlivých vrstvách. Cílem není rozlišit typy značek, jako u barvených Petriho sítí, ale zlepšit čitelnost změn stavů, pokud použijeme více vrstev najednou.

Potřebujeme-li simulovat běh děje v Petriho sítích, můžeme opět použít součást kancelářské balíku (tzv. Office). V příloze na CD k této práci naleznete některé příklady zpracované v programu určeném pro tvorbu prezentací (MS PowerPoint) pomocí nástroje animace objektů. Aktivita je vhodná do výuky informatiky (resp. ICT). Ukazuje jednak netradiční využití prezentace a lze na ní procvičit algoritmizaci. Značky je nutné rozpohybovat, proto je potřeba rozplánovat simulaci na jednotlivé kroky a jejich sekvenci. Následně můžeme ukázat optimalizaci vytvořeného „programu“. Studenti porovnají mezi sebou počet využitých snímků, tedy v případě, že průběh byl nasimulován správně (po verifikaci).

Takové nástroje postačují pro školní prostředí, neboť nevyžadují ani nákup nové aplikace, ani její instalaci. Pro složitější modely nebo pro využití v odborné praxi však vhodné nejsou. Existují specializované programy zaměřené na modelování a analýzu Petriho sítí.

Na trhu jsou jak komerční řešení, tak freewarové aplikace (Češka 1994, str. 79; Voráčová, Pěnička a Veselý 2001, str. 113; uni-hamburg.de). Ve výběru níže uvádíme zástupce pro různá prostředí a různé varianty Petriho sítí.

název aplikace	prostředí	poznámky
ARP Tool	MS DOS	zdarma, Turbo Pascal, bez GUI ^{**} , Federal University of Santa Catarina
Artifex	MS Windows (NT–XP) Linux, SunOS	komerční, komplexní nástroj pro většinu typů PS ^{***} a jejich analýzu
CPN Tools	MS Windows (XP) Linux	zdarma, Aarhus university, především pro barevné PS
F-net	MS Windows OS/2	komerční, vhodné pro stochastické a časované PS
HPSim	MS Windows (95–XP)	zdarma, MS Visual Studio C++, německý editor, bez možnosti analýzy PS
JFern	Java	zdarma, simulace a jednoduché analýzy základních i vyšších PS
PESIM	MS Windows (3.1)	C++, VUT v Brně
Snoopy	MS Windows Linux	zdarma, Technická univerzita v Cottbuse, dobrý pro hierarchické PS
Visual Object Net++	MS Windows	zdarma, navržen pro hybridní PS, Technická univerzita v Ilmenau

Obrázek 21 – Výběr aplikací pro modelování Petriho sítí

* MS = Microsoft

** GUI = grafické rozhraní

*** PS = Petriho síť

4 VÝUKA ALGORITMIZACE A MODELOVÁNÍ

Některé popisy v předchozích kapitolách jsou opatřeny didakticko-metodickým komentářem pro případnou implementaci do výuky. Proč ale vůbec zařazovat témata jako jsou programování, modelování či Petriho sítě do výuky?

4.1 Ukotvení v kurikulárních dokumentech

Vycházejme z rámcových vzdělávacích programů jako strategických dokumentů ovlivňujících obsah vzdělávacích oblastí, a tedy i výuky na školách. Základní vzdělávání v oblasti Informační a komunikační technologie je zaměřeno především na získání ICT gramotnosti. Dovednosti při práci s aplikacemi a informacemi nejsou samy o sobě cílem. Cílem je naučit žáky získané kompetence kreativně aplikovat v ostatních vzdělávacích oblastech a při řešení problémů (ve smyslu úkolů, situací) v běžném životě. Z deklarovaného cílové zaměření oblasti v Rámcovém vzdělávacím programu pro základní vzdělávání citujeme následující (Jeřábek a Tupý 2007, str. 32):

- *pochopení funkce výpočetní techniky jako prostředku simulace a modelování přírodních i sociálních jevů a procesů*
- *schopnosti formulovat svůj požadavek a využívat při interakci s počítačem algoritmické myšlení,*
- *využívání výpočetní techniky, aplikačního i výukového software ke zvýšení efektivnosti své učební činnosti a racionálnější organizaci práce,*
- *tvořivému využívání softwarových a hardwarových prostředků při prezentaci výsledků své práce.*

Petriho sítě pomáhají přímo naplňovat první uvedený cíl. Jak je popsáno výše, modelování pomocí Petriho sítí pomáhá rozvíjet i algoritmické myšlení, tím naplňuje i druhý z cílů. Na zbylé dva cíle se lze zaměřit při výběru jevů, které budeme modelovat (viz příklad Domácí rodinný úklid).

Očekávané výstupy týkající se modelování či algoritmizace zatím v Rámcovém programu pro základní vzdělávání nenajdeme, ale je možné zabývat se informačními technologiemi a „nedotknout“ se těchto témat? Vaníček (2004, str. 1) používá pěkné přirovnání k matematice. Algoritmizaci využíváme při řešení úloh a trénujeme tím mozek. Jako příklad uveďme řešení konstrukční úlohy. Konstrukci musíme rozdělit na jednotlivé kroky, které nelze provádět v úplně libovolném pořadí. O řešení je nutné diskutovat, zda je

správné a zda je jediné možné. V neposlední řadě se žáci postupně učí algoritmus zaznamenávat matematickými symboly, tedy standardizovaným jazykem s vyhrazenými ikonami. Algoritmizace je tedy i matematická disciplína. Konstrukční úloha se dá zpracovat jak na papír, tak pomocí počítače, tím se vracíme zpět k citovaným cílům vzdělávací oblasti Informační a komunikační technologie a aplikaci získaných dovedností do dalších vzdělávacích oblastí, v tomto případě Matematika a její aplikace.

Pro ověření některých hypotéz či řešení úloh potřebují žáci mít na čem je vyzkoušet. Z tohoto důvodu využívají učitelé s žáky při edukačním procesu různé modely. Zmíněná matematika často pracuje s tabulkovými editory/procesory. Jejich uživatelská přívětivost spočívá v předpřipraveném prostředí. Tabulku nemusí žák vytvářet a ani ho nezajímá, jak funguje spreadsheet uvnitř. Do buněk dosazuje konkrétní hodnoty, využívá předdefinovaných funkcí a dalších nástrojů. Není náhodou, že výuka např. v programovacím jazyku Pascal začíná taktéž matematickými příklady. Počítače ostatně, jak název napovídá, vznikly jako výkonné stroje pro obtížnější výpočty. Běžné programovací jazyky jsou však pro mladší žáky náročné. Cílem výukových programovacích prostředí je snížit nutnou úroveň abstrakce. Základní ideou těchto prostředí je obvykle personifikovaná postava (robot, čaroděj, želva), která je pomocí příkazů ovládána (Vaníček 2004, str. 2). Nezapomínejme také na úlohu robotických stavebnic. Poskytují žákům přirozenou motivaci k osvojování programátorských kompetencí.

Situace na středních školách (ne informatického zaměření) je obdobná. Vzdělávací oblast Informatika a ICT v Rámcovém vzdělávacím programu pro gymnázia kladé větší důraz na informatiku jako vědní obor, což se projevuje také ve vymezení cílového zaměření vzdělávací oblasti (Jeřábek, Krčková a Hučínová 2007, str. 63), z něhož citujeme:

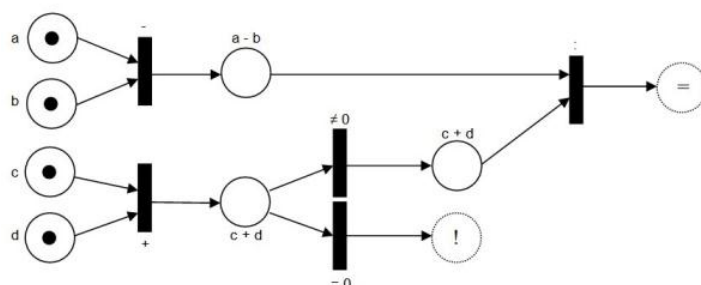
- *využívání prostředků ICT k modelování a simulaci přírodních, technických a společenských procesů a k jejich implementaci v různých oborech;*
- *uplatňování algoritmického způsobu myšlení při řešení problémových úloh;*
- *porozumění základním pojmům a metodám informatiky jako vědního oboru a k jeho uplatnění v ostatních vědních oborech a profesích;*
- *porozumění zásadám ovládání a věcným souvislostem jednotlivých skupin aplikačního programového vybavení a k vhodnému uplatňování jejich nástrojů, metod a vazeb k efektivnímu řešení úloh.*

Bohužel cílům stěžejním v kontextu tématu modelování a algoritmizace odpovídá pouze jediný očekávaný výstup „žák aplikuje algoritmický přístup k řešení problémů“, přestože se jedná o zásadní informatická témata.

4.2 Inspirace využití Petriho sítí ve výuce

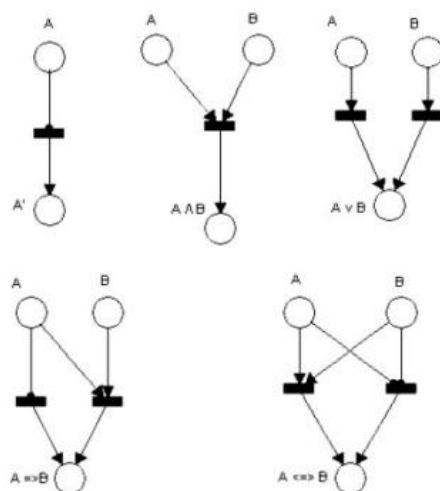
Petriho sítě poskytují výuce informatiky mezistupeň výuky programování/modelování mezi popisováním algoritmů jazykem běžným (česky) a programovacím (Pascal, C, ...). Petriho sítě, jako nástroj ikonického modelování, přináší úsporu v záznamu (nahrazuje některá slova obrázkem), a přesto mohou zůstat pro žáky srozumitelné. Žáci se učí používat vyhrazené symboly (příkazy). Petriho sítě vyžadují určitá pravidla pro skládání ikon, tím se žáci učí dodržovat syntax. Ani ikon ani pravidel není velké množství. Jsou zapamatovatelná i pro mladší žáky. Petriho sítě umožňují pomocí simulace názorně ukázat posloupnost vykonávání jednotlivých kroků algoritmu, a to i u paralelního jevu. V neposlední řadě je možné modely vytvářet s běžným aplikačním vybavením počítače. Nejsou však na počítačové prostředí vázané. Modely je možno vytvářet kreslením na papír. Pro mladší žáky by bylo možné připravit jakousi sadu Petriho sítí, stejně jako existuje pro žáky prvních tříd abeceda, kterou lze rozstříhat. Následně je možné z jednotlivých písmen skládat slova. Pomůcka pro modelování v Petriho sítích by byla založena na stejném principu, jen místo písmen by obsahovala jednotlivé ikony základní sady P/T Petriho sítí.

Podívejme se na příklady využití modelů Petriho sítí u jiných autorů. Zajímavou ukázkou didaktického využití Petriho sítí nalezneme u Baldy (2007, str. 16). Výpočet lomeného výrazu totiž obsahuje pro žáky skrytě paralelní jev. Výpočty hodnot čitatele a jmenovatele na sobě nejsou nijak závislé a mohou probíhat současně. Na modelu (viz obrázek 22) vidíme počáteční značení.



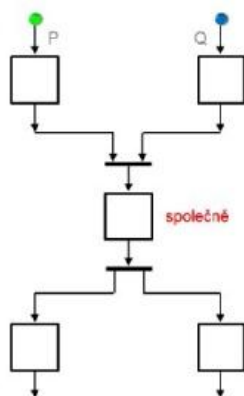
Obrázek 22 – Model výpočtu výrazu $(a - b) : (c + d)$

Příklad na pomezí matematiky a informatiky, z oblasti booleovské algebry, najdeme u Kochaníčkové (2008, str. 25). V souvislosti s výukou výrokové logiky na střední škole a modelováním logických operandů pomocí tabulek se o totéž můžeme pokusit pomocí Petriho sítí.



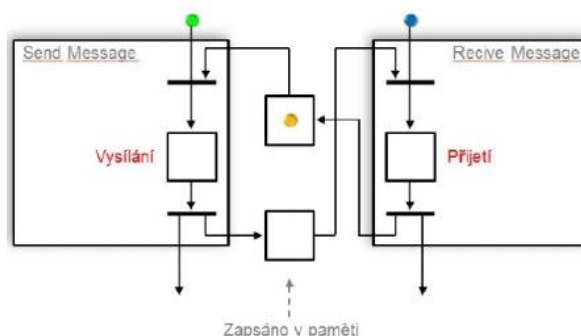
Obrázek 23 – Model logických operací

Další sadu příkladů využitelných ve výuce zaměřené na principy operačních systému (např. souběh procesů nebo asynchronní zasílání zpráv) včetně animací najdeme u Smrčky (2001, str. 2–4).



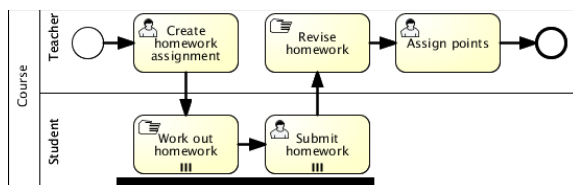
Obrázek 24 – Model souběhu procesů

Na modelech zmiňovaných příkladů na obrázcích 24 a 25 je zajímavé to, že autor pro místo používá ikonu čtverce, nikoli kružnice. S žáky je žádoucí spíše dodržovat jednu konvenci. Byť se jedná o jednoduché přeznačení, může to ze začátku zhoršovat orientaci v modelu. Naopak se může jednat o jednoduchý (primitivní) test porozumění konceptu Petriho sítí.



Obrázek 25 – Model asynchronního zasílání zpráv

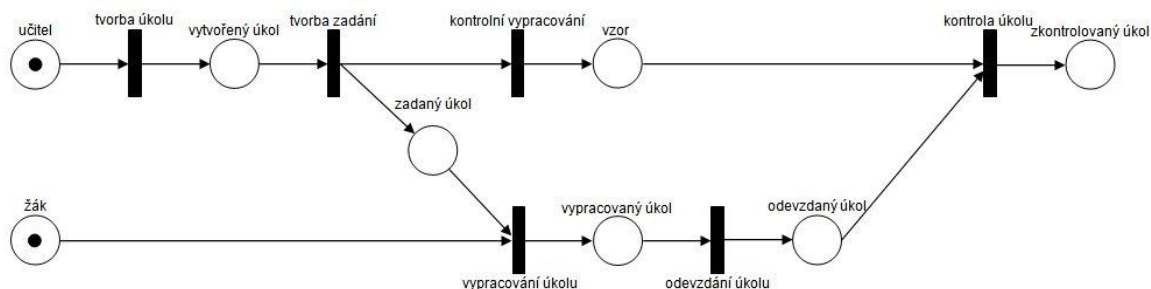
Pro žáky zajímavým tématem může být modelování výukových procesů. Pro procesy na vysoké škole modely zpracoval ve své diplomové práci Jiří Novák (2010, str. 42–47). K modelování sice nevyužívá notaci Petriho sítí, ale úlohu pro starší žáky můžeme koncipovat jako převedení jednoho modelu do jiného.



Obrázek 26 – Model vypracování domácího úkolu

Zajímavé na modelech je, že všechny činnosti na sebe lineárně navazují, nejedná se tedy o paralelní jevy. Je to mu tak ve skutečnosti vždy?

Najdeme s žáky výukové procesy/činnosti, které mohou probíhat souběžně? Při modelování stejné situace v Petriho sítích začleníme paralelnost jevů (viz obrázek 27).



Obrázek 27 – Model domácího úkolu v Petriho sítích

Model není didakticky zcela správně, neboť ve skutečnosti pedagog vypracovává vzorové řešení ještě před zadáním úkolu. Otázka původně položená o několik řádků výše tedy zůstává.

5 PŘÍKLADY

Všechny následující příklady patří k těm s ne úplně jednoduchým a jednoznačně správným řešením. Jednodušší příklady nalezneme snadno v různých publikacích nebo na internetu. Některé z nich jsou uvedeny v teoretické části.

V textu budou naše modely statické. Zachytíme pouze strukturu sítě. V reálné výuce samozřejmě předpokládáme vizualizaci pohybu pomocí některých z nástrojů uvedených v kapitole 3.3. Řešení příkladů je rozkrokováno tak, aby žáky jednotlivé kroky přivedly k důležitým pojmům, resp. k jejich vnímání. Pedagog může při implementaci příkladů do výuky vytvořit menší kroky či vytvořit seskupení mezikroků jinak. Na úvod každého kroku je vložena motivační otázka (značena kurzivou), kterou pokládá učitel, s odpovědí, ke které by měli žáci dojít. Nejde o doslovné znění, formulace nejsou v podobě žákovského pojmového aparátu, ale o myšlenku/princip. Metoda vychází z heuristického rozhovoru. Cílem je dovést žáky k řešení, nikoli jim ho naservírovat, aby nezůstali pouze pasivními příjemci informací. Následující příklady se nezabývají úvodem do Petriho sítí a předpokládají již znalost tohoto modelu, resp. jeho základní sady ikon a pravidel. Informace o modelu a jeho nástrojích jsou uvedeny v kapitole 3.

V následujících příkladech pracujeme s podobou modelu, která je nazývána počáteční značení. Demonstruje výchozí postavení značek modelovaného systému. U příkladu Domácí rodinný úklid je použito počáteční značení přesné. Místa „Jan“ a „Lenka“ nemohou obsahovat více značek, protože se jedná vždy o jednu konkrétní osobu. V příkladu Tvorba informačních tabulí obsahuje zdroj „nápad“ pouze jednu značku. Jedná se ale o zjednodušené znázornění, neboť se předpokládá, že nápadů je více. Tedy i toto místo by mělo obsahovat v počátečním značení více značek. Místo „hotovo“ musí značku obsahovat již v počátečním značení. Přejít může být využit pouze, pokud jsou splněny všechny vstupní podmínky. Pokud by místa „hotovo“ na začátku neobsahovala značku, proces by vůbec nezačal. Obdobně tomu je se signálem „volno“ v příkladu Železniční stanice.

5.1 Domácí rodinný úklid

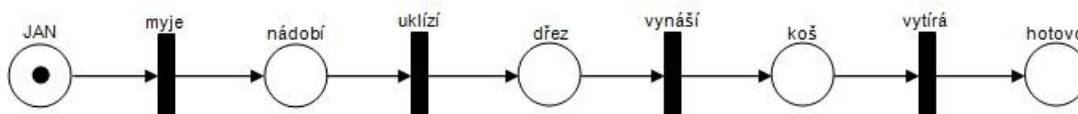
Jan má sestru Lenku. Dostali od maminky na sobotní odpoledne seznam úkolů domácích prací. Až je udělají, dostanou kakao a zákusek, který maminka upekla. Sourozenci vyhodnotili náročnost úkolů a rozdělili si je. Jan umyje a utře nádobí, potom po sobě uklidí dřež, a pak vynese koš. Lenka mezitím utře prach. Až půjde Jan s košem, vyluxuje. Nakonec Jan vytře.

Nejprve si vytvoříme model prvního Janova úkolu. Jak na to? Co bude místo a co přechod? Místa budou popisovat, co je už uděláno, přechody jednotlivé činnosti. A první místo popíšeme Janovým jménem.



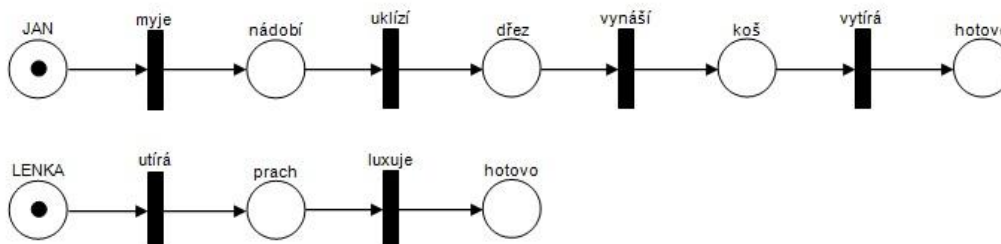
Obrázek 28 – Model jedné domácí činnosti

Jak ukážeme v modelu, že na sebe činnosti navazují? Umístíme je lineárně, tj. za sebe.



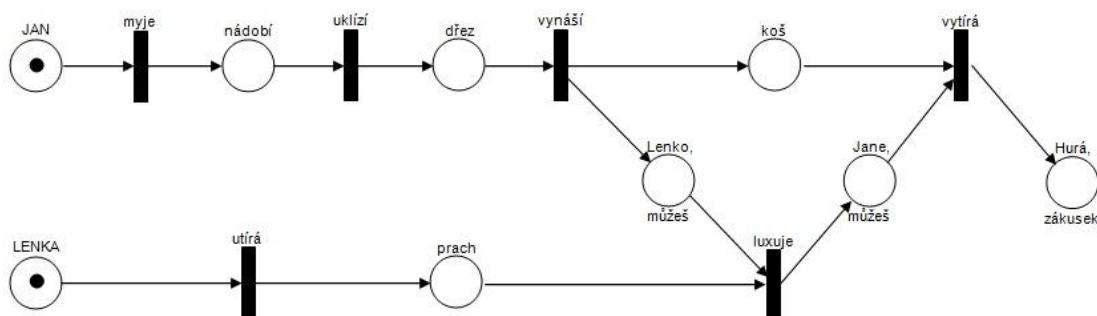
Obrázek 29 – Model sekvence Janových úkolů

Lenka dělá svoje úkoly současně s Janem. Jak to vymodelujeme? Umístíme je paralelně, tedy vedle (pod) sebe.



Obrázek 30 – Model činností Jana i Lenky

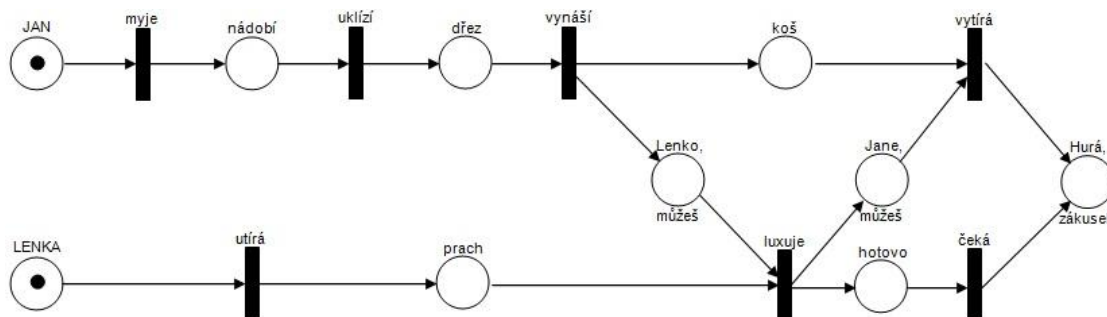
Už v úvodu této úlohy bylo naznačeno, že Lenka luxuje až v průběhu toho, co Jan vynáší koš. Jinak by jí totiž překážel. A Jan s vytíráním počká, až je vyluxováno. Jak se to projeví v našem modelu? Obě linie propojíme. Musí si nějak vzájemně říct, že mají hotový příslušný úkol.



Obrázek 31 – Model navázání činností

U tohoto modelu je třeba dodržovat pravidla Petriho sítí. Žáci budou mít v první chvíli tendenci kreslit šipku z místa „dřez“. Taková sekvence ale znamená větvení programu (viz obrázek 10). Značka by pak mohla pokračovat libovolným směrem. V tomto případě je potřeba značku rozdělit, neboť Jan pokračuje ve své činnosti dál a Lenka se zároveň dozví, že Jan již vynáší koš. Proto je nutné vést šipku až z přechodu „vynáší“, tím právě dojde k rozdělení značky. Dále je potřeba zdůraznit, že délka šipky neznamena konkrétní časovou hodnotu.

Je náš model správně? Ano, až na jednu maličkost. Ztratí se nám Lenka. Začínáme se dvěma značkami a končíme jen s jednou.



Obrázek 32 – Finální model domácího úklidu

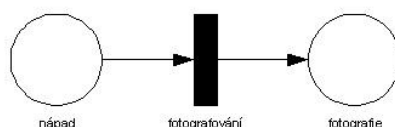
V modelu je nutné přidat přechod odpovídající situaci, ve které Lenka čeká, až Jan dokončí svůj poslední úkol. Existence třetí značky v určité okamžiky (signály „můžeš“) může být v argumentaci k popisované úpravě poněkud problematická. Uvádíme-li jako důvod špatný počet značek. Příklad vyžaduje od žáka abstrakci v tom smyslu, že značka v tomto příkladu nepředstavuje vždy stejný předmět. Signál vznikne v průběhu simulace a také v průběhu simulace zanikne. Předměty „Jan“ a „Lenka“ ale volně zanikat a vznikat nemohou.

5.2 Tvorba informačních tabulí

Na některých školách se v poslední době realizovaly projekty na téma naučná stezka. Ta je založena na informačních tabulích, které jsou umístěny na vytipovaných místech např. v areálu školy, blízkém okolí apod. Pokusíme se tedy namodelovat hrubé schéma postupu při tvorbě těchto cedulí. Tím rozumějme, že při vytváření modelu není třeba rozepisovat akce do nejmenších podrobností. Zachytíme v něm jen nejdůležitější okamžiky tvorby.

Máme již vytipovaná místa pro naše cedule. Co chceme na informační ceduli mít? Bude na ní umístěna fotografie místa a krátký text se základními a zajímavými daty o místě či události k místu se vztahující.

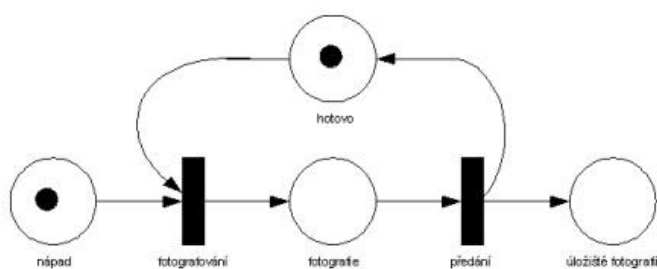
Jak začneme? Musíme mít nápad a ten sdělit fotografovi. Ten pak udělá fotografii.



Obrázek 33 – Základní model jedné činnosti

Kdy je dobré udělat další fotografii? Aby nevznikl zmatek, je dobré začít novou sérii až po té, co předáme graficky hotové fotografie předchozího místa. Výhodné je také nečekat na nafocení všech míst proto, aby se mohly cedule vytvářet průběžně.

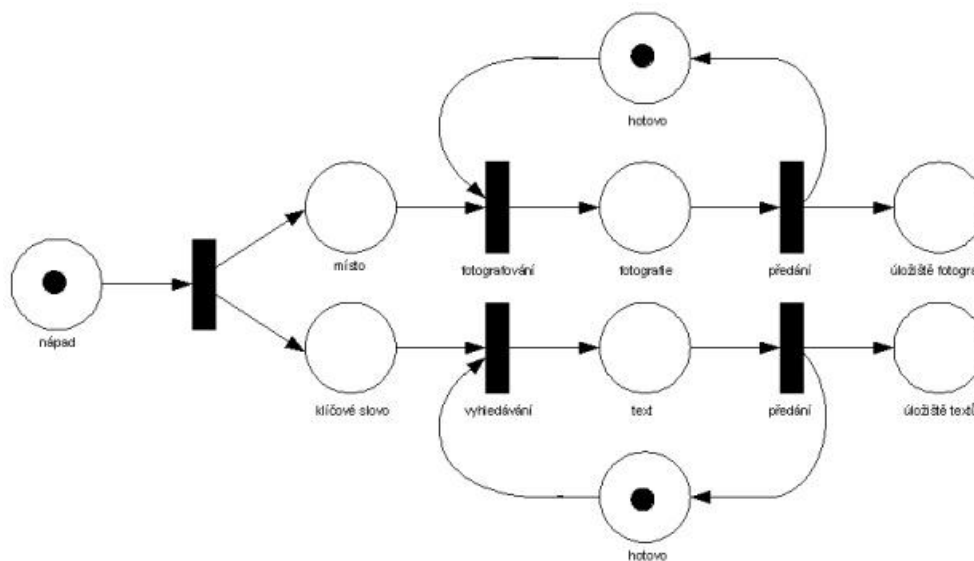
To znamená, že musíme vědět i v našem modelu, kdy je možné začít fotografovat novou sérii. Jak to poznáme? Předá se signál „hotovo“.



Obrázek 34 – Model smyčky fotografování série

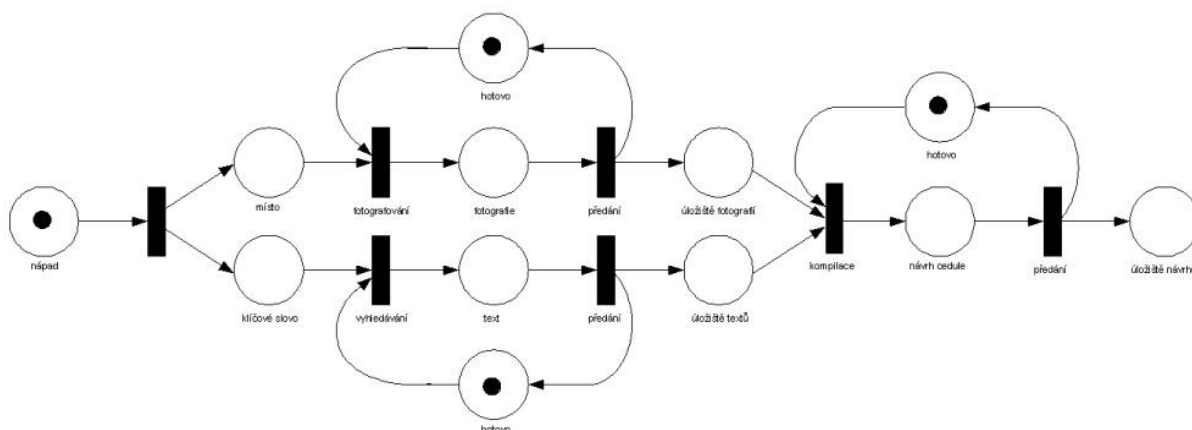
Fotografie tedy řešíme, pojďme k textům. Musí se s vytvářením textu čekat na fotografii? Nemusí, pokud vzešly ze stejného nápadu. Tyto dva procesy mohou běžet současně (paralelně).

Jak bude model tvorby textů vypadat? V podstatě velmi podobně jako u fotografií. Jako zdroj značek bude totéž místo jako u fotografií.



Obrázek 35 – Model počátku větvení a paralelních dějů

Kdy začneme dávat text a fotografie dohromady? Pokud jsou obě části téže cedule hotové (dochází k ukončení větvení) a pokud ten, kdo má kompilaci na starosti, má „volné ruce“. A proces kompilace fotografií a textů tedy vypadá opět obdobně jako proces tvorby fotografie či textů.



Obrázek 36 – Neodladěný návrh výsledného modelu

Následně, po schválení návrhů, se mohou nechat cedule začít vyrábět a osazovat na potřebná místa.

Celý model je tedy možné dále rozšiřovat přidáváním dalších souvisejících činností. Výhodou zvoleného příkladu je opakované použití stejného jednoduchého principu pro „tvorbu věci“ se signálem hotovo.

Bude tento proces za každých okolností produkovat správné cedule? Nebude, protože není zaručeno spárování správné fotografie s příslušným textem, pokud bude fotograf pracovat rychleji než tvůrce textů. Kapacita míst, jak víme, je v P/T Petriho sítích neomezená.

Jak můžeme dosáhnout toho, aby se správně párovaly? Dvěma způsoby – můžeme předávání signálu „hotovo“ v první fázi posunout až do distribuce nápadu nebo bychom potřebovali, aby si úložiště fotografií a textů pamatovalo pořadí, v jakém na ně dorazily. První však vede ke ztrátě možnosti dělat fotografie a vyhledávat texty nezávisle na sobě, ztrátě výhody paralelního děje. Druhý vede k pojmu „fronta“. Přirozeně se tak dostáváme do situace, kdy bychom potřebovali variantu P/T Petriho sítí rozšířit. Takové příklady je lepší modelovat pomocí tzv. frontových Petriho sítí (viz kapitola 3.1).

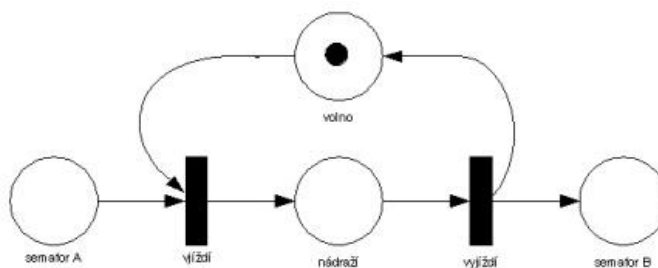
5.3 Železniční stanice

Pro inspiraci nám může posloužit například návštěva výstavy vlakových modelů nebo řídicího stanoviště pracoviště přímo na vlakovém nádraží. V našem případě budeme modelovat jen jednodušší situace. Ty se dají stále zesložitovat přidáváním různých prvků z „reálné“ železnice*.



Obrázek 37 – Model nádraží

Jak by mohl vypadat model nádraží na trati, kde se jezdí pouze jedním směrem? Žáky po zkušenostech z minulého příkladu napadne, že se musí předávat signál „volno“. Za zdroj a archiv objektů pro stanici budeme považovat umístění semaforů A a B.



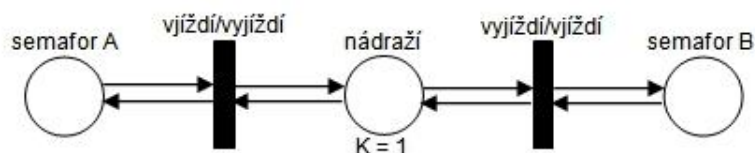
Obrázek 38 – Model jednokolejného nádraží v jednom směru

* Za dobrovolný úkol mohou žáci zjistit, které skutečné nádraží je na obrázku. Je to nádraží v Lyonu – Francii – u letiště Saint-Exupéry a je určeno pro vlaky TGV.

Je nutnému mít v modelu signál volno? Ano, protože na jednu kolej smí vjet pouze jeden vlak. Signál „volno“ bychom mohli vynechat v případě, že bychom místu „nádraží“ nastavili kapacitu rovnou jedné. Tím bychom se dostali k rozšíření P/T Petriho sítě o omezenou kapacitu místa.

Není ale potřeba stejně nastavit kapacitu místu „volno“? Není, protože v počátečním značení je v tomto místě pouze jedna značka. Další se do místa „volno“ může dostat pouze vznikem z přechodu „vyjíždí“. Ten, ale může být aktivován pouze v případě, že vlak vjede do stanice. Vlak může vjet do stanice ale pouze tehdy, pokud se značka z místa „volno“ přesune do přechodu „vjíždí“ (a tam zanikne). Nemůže tedy nastat situace, kdyby v jeden okamžik existovaly dvě značky v místě „volno“.

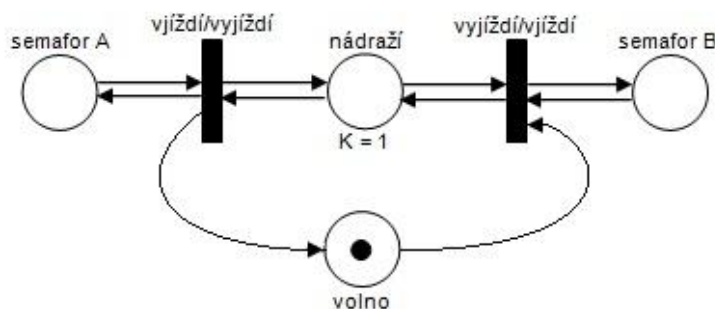
Ve skutečnosti jezdí vlaky nádražím obvykle oběma směry. Jak tuto situaci namodelujeme? Stačí omezit kapacitu nádraží a nepotřebujeme ani signál „volno“.



Obrázek 39 – Model jednokolejného obousměrného nádraží

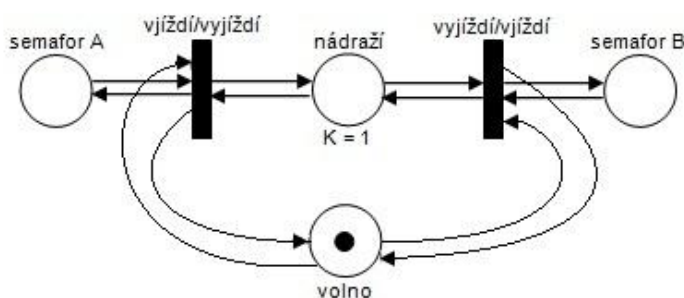
Co se stane, když vlak/značka přijede do obou míst semaforů? V místě „nádraží“ žádná značka není, takže obě mohou ve stejnou chvíli využít přechodů. To znamená, že signál „volno“ budeme přeci jen potřebovat.

Stačí signál „volno“ přidat k našemu předchozímu modelu? Zkusme to.



Obrázek 40 – Model jednokolejné trati se signálem „volno“

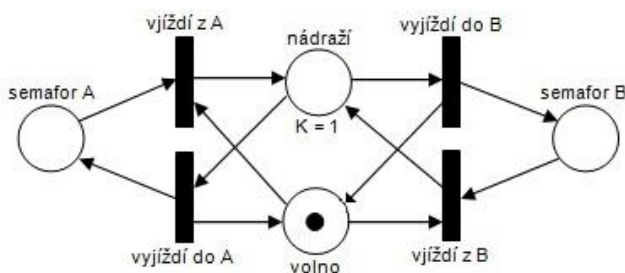
Na takovéto trati by opět mohly vlaky vjíždět pouze z jednoho směru – od semaforu B. Tak doplníme signál i do druhého směru.



Obrázek 41 – Model jednokolejné trati se „signálem“ volno v obou směrech

Skutečně nám tato úprava poskytuje bezpečný průjezd stanicí v obou směrech? Nikoli, když bude vlak přijíždět např. od semaforu B. Značky z míst „semafor B“ a „volno“ se přesunou do přechodu „vyjíždí/vjíždí“. Podle modelu se vlak rozdělí na dvě části – jedna se vrátí do bodu, odkud přijela, druhá pokračuje do nádraží. Navíc se ale vrátí na své místo signál „volno“. Takže vlak stojí na nádraží, na semaforu B a ještě nám model signalizuje, že vjezd do nádraží je povolen.

Jak to tedy vyřešíme? Využijeme rozlišení vlaku na vjíždějící a vyjíždějící. Budeme pak mít pro jedno reálnou kolej dva přechody v Petriho sítích.



Obrázek 42 – Model jednokolejné trati v obou směrech bezpečný

Příklad můžeme rozvíjet „do délky“. Z hlediska samotného nádraží nás nezajímalo, jak je řešena situace kolem semaforů. Tato místa jsme ponechali s neomezenou kapacitou. Vlaky se nám mohou na semaforech hromadit. Vypadá to tedy, že před stanicí je trať vícekolejná a na nádraží se zredukuje, což je opak obvyklé reality. Můžeme tím pádem příklad rozvíjet také „do šířky“. Aby v některých stanicích mohli lidé pohodlně přestupovat, mívají nádraží více kolejí/nástupišť.

6 ZÁVĚR

V současné době je velkou neznámou, jakým směrem se bude výuka informatiky ubírat vzhledem ke stále měnícím se okolnostem v kurikulu pro základní (revize RVP, doporučené obsahy, standardy) a střední školy (státní maturity) na státní úrovni. Na jedné straně je snaha více zaměřit vzdělávací oblast na běžné kancelářské nástroje a uživatelské dovednosti aplikované do jiných oborů. Na druhé straně stojí Informatika jako vědní disciplína a jedny z jejích stěžejních témat – algoritmizace a modelování. Toto vše se v rámci současné povinné časové dotace, podle našeho názoru, ve škole stihnout nedá.

Tato práce poskytuje inspiraci pro vyučující, kteří chtějí ve svých hodinách realizovat i aktivity zaměřené právě na modelování. V rámci vymezení základních pojmů modelování se mohl čtenář seznámit i s dalšími souvislostmi. Velkým teoretickým východiskem pro jejich výběr byla publikace německých autorů Schubert a Schwill „Didaktik der Informatik“. Obdobná publikace v našem prostředí chybí.

Protože se vycházelo z předpokladu, že většina učitelů informatiky se nikdy nesetkala s Petriho sítěmi, byla jejich teorii věnována samostatná kapitola. V ní se mohl pedagog seznámit s používanými značkami. Byly ve stručnosti popsány rozdíly mezi jednotlivými variantami Petriho sítí. Pokusili jsme se ukázat příklady dobré praxe využití Petriho sítí ve vzdělávacím procesu. Jednoduché příklady byly přejaty z odborných publikací různých autorů. Další příklady lze snadno vytvořit využitím jiného motivu pro stejný princip.

Pro využití Petriho sítí hovoří především tři následující důvody:

- Umí modelovat paralelní jevy.
- Jedná se o ikonické modelování, které je podle našeho názoru pro žáky více intuitivní, než symbolické. Přesto dobře rozvíjí abstraktní myšlení a buduje pojmy týkající se algoritmů a modelů.
- Pro vytváření Petriho sítí je možné používat běžný aplikační/kancelářský software. Dokonce je lze vytvářet i zcela bez počítače.

Pro přemýšlivé učitele i žáky byly připraveny tři obsáhlejší příklady, na kterých jsme se snažili demonstrovat zajímavé situace. V příkladu Domácí rodinný úklid objevují žáci paralelnost děje nebo rozdělení značky. Příklad Tvorba informačních tabulí zavádí signál „hotovo“, opakování stejných schémat a ukazuje potřebu rozšíření P/T Petriho sítí o fronty. U modelování železniční stanice dochází ke zjednodušení reálné situace, využití principu

zvaného „hrozny problémů“ (rozvíjení situace co do obtížnosti, komplikovanosti a složitosti), rozšíření základní varianty Petriho sítí o omezení kapacity místa, rozdělení jednoho předmětu originálu na dva předměty modelu atd.

Formálně lze považovat vytčené cíle za naplněné. Za skutečné naplnění cíle však budeme považovat, pokud se podaří v některém z učitelů vzbudit zájem – ať již konkrétně o naše či převzaté příklady, o Petriho sítě a jejich využití ve výuce, nebo o modelování jako takové. Dokonce bude úspěchem, když tato práce donutí čtenáře přemýšlet o obsahu předmětů vzdělávacích oblasti Informační a komunikační technologie a Informatika.

POUŽITÁ LITERATURA

- [1] Balda, P. *Informační a řídicí systémy I. – Úvod do Petriho sítí* [online]. Západočeská univerzita v Plzni, Fakulta aplikovaných věd, Katedra kybernetiky, 2007.
[cit. 17. 8. 2012]. Dostupné z http://vendulka.zcu.cz/Download/Free/IRS1/IRS1-01_Petriho_site.pdf.
- [2] Češka, M. *Petriho sítě – Úvod do teorie a nástrojů pro aplikaci Petriho sítí*. Brno: Akademické nakladatelství CERM, 1994. ISBN 80-85867-35-4.
- [3] Jeřábek, J., Krčková, S., Hučínová, L. a kol. *Rámcový vzdělávací program pro gymnázia*. Praha: Výzkumný ústav pedagogický, 2007. ISBN 978-80-87000-11-3.
- [4] Jeřábek, J., Tupý, J. *Rámcový vzdělávací program pro základní vzdělávání – Pomůcka na pomoc učitelům* [online]. Praha: Výzkumný ústav pedagogický, 2007. Aktuální znění k 1. 9. 2010. [cit. 23. 7. 2012] Dostupné z <http://www.vuppraha.cz/wp-content/uploads/2009/12/RVPZV-pomucka-ucitelum.pdf>.
- [5] Kochaničková, M. *Petriho sítě*. Olomouc: Univerzita Palackého, 2008.
- [6] Křivý, I., Kindler, E. *Simulace a modelování* [online]. Ostravská univerzita, 2001.
[cit. 20. 7. 2012]. Dostupné z <http://prf.osu.cz/kip/dokumenty/Msm.pdf>.
- [7] Novák, J. *Modelování výukových procesů* [online]. Brno, 2010. Diplomová práce. Masarykova univerzita, Fakulta informatiky. Vedoucí práce: Mgr. Jiří Kolář.
[cit. 1. 9. 2012]. Dostupné z: http://is.muni.cz/th/172704/fi_m/dp.pdf.
- [8] *Petri Nets Tools Database* [online]. Universität Hamburg. [cit. 23. 8. 2012]. Dostupné z <http://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/quick.html>.
- [9] Schubert, S., Schwill, A. *Didaktik der Informatik*. 2. vydání. Heidelberg: Spektrum Akademischer Verlag, 2011. ISBN 978-3-8274-2652-9.
- [10] Smrčka, F. *Využití animace ve výuce předmětu Operační systémy*. In: Alternativní metody výuky, 2011 [online]. Praha: Karlova univerzita, Přírodovědecká fakulta.
[cit. 2. 9. 2012]. Dostupné z http://everest.natur.cuni.cz/konference/2011/prispevek/smrcka_prispevek.pdf.
- [11] Vaníček, J. *Programování*. In: Přednášky z didaktiky informatiky a výpočetní techniky [online]. JČU v Českých Budějovicích, 2004. [cit. 19. 7. 2012] Dostupné z http://eamos.pf.jcu.cz/amos/kat_inf/externi/kat_inf_0548/7_programovani.pdf.
- [12] Voráčková, Š., Pěnička, M., Veselý, J. *Úvod do modelování procesů Petriho sítěmi*. 1. vydání. Praha: České vysoké učení technické v Praze, 2008. ISBN 978-80-01-03979-3.