

Procesy, vlákna

Modelovanie a simulácia procesov

pomocou Petriho sietí

Zoltán Balogh

Vlákna

Procesy a vlákna

Program

- Súbor definovaného formátu obsahujúce inštrukcie, dáta a ďalšie informácie potrebné na vykonanie danej úlohy

Proces

- Systémový objekt charakterizovaný svojím pamäťovým priestorom a kontextom (pamäť aj niektoré ďalšie zdroje sú pridelované procesom)

Vlákno, tiež "sled"

- Objekt, ktorý vzniká v rámci procesu, je viditeľný iba vo vnútri procesu a je charakterizovaný svojím stavom (CPU sa prideľujú vláknám)

Model - len procesy (nie vlákna)

- Proces: jednotka plánovania činnosti aj jednotka vlastníci prostriedky

Model - procesy a vlákna

- Proces: jednotka vlastníaca zdroje
- Vlákno: jednotka plánovanie činnosti



Každé vlákno si udržiava svoj vlastný

- zásobník
- PC (program counter)
- registre
- TCB (Thread CB)

Vlákno môže pristupovať k pamäti a ostatným zdrojom svojho procesu

- Zdroje procesu zdieľa všetky vlákna jedného procesu
- Akonáhle jedno vlákno zmení obsah (nelokálne - mimo zásobník) bunky, všetky ostatné vlákna (toho istého procesu) to vidí
- Súbor otvorený jedným vláknom majú k dispozícii všetky ostatné vlákna (toho istého procesu)

Procesy a vlákna

Prečo využiť vlákna

- Využitie multiprocesorových strojov (vlákna jedného procesu môžu bežať na rôznych CPU)
- Jednoduchšie programovanie
- Typický príklad: jedno vlákno vykonáva používateľom požadovanú úlohu a druhé vlákno prekresľuje obrazovku

1: 1 UNIX Systém V, (MS-DOS)

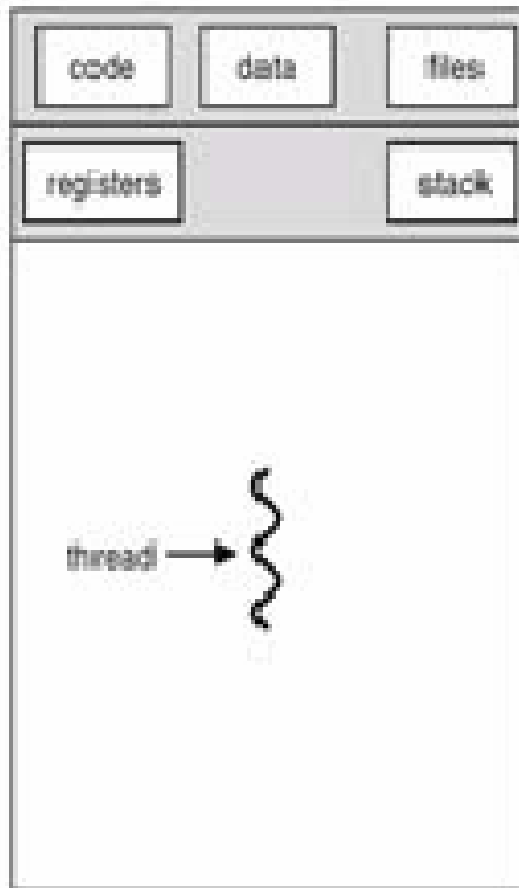
- Pojem vlákno neznámy, každé "vlákno" je procesom s vlastným adresovým priestorom a s vlastnými prostriedkami

1: M OS / 2, Windows XP, Mach, ...

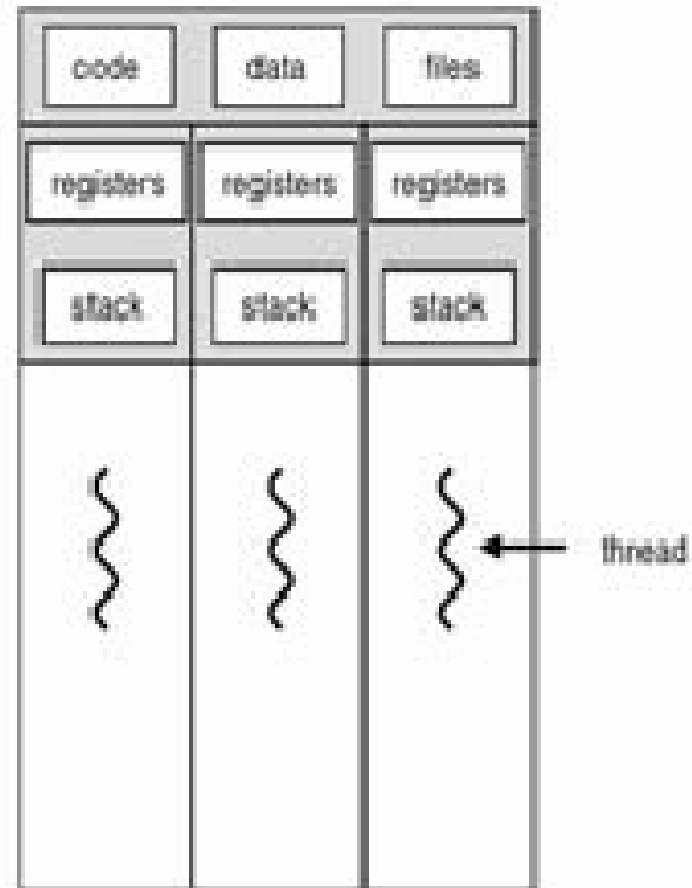
- V rámci 1 procesu možno vytvoriť viac vlákien, proces je vlastníkom zdrojov (vlákna zdieľajú zdroje procesu)



Procesy vs. vlákna



single-threaded



multithreaded

Jednovláknový OS:

- Nepodporuje koncept vlákien (nepozná pojem vlákno)
- MS-DOS: 1 proces, 1 vlákno
- UNIX: n procesov, 1 vlákno / 1 proces

Multivláknový OS:

- Podporuje koncept viacero vlákien v rámci procesov
- Windows 7, Solaris, ...

Výhody použitia vláken

Výhody

- Vlákno sa vytvorí rýchlejšie ako proces
- Vlákno sa ukončí rýchlejšie ako proces
- Medzi vláknami sa rýchlejšie prepína než medzi procesmi
- Jednoduchšie programovanie (jednoduchšia štruktúra programu)
- U multiprocessorových systémov môže na rôznych procesoroch bežať viac vlákien jedného procesu súčasne

Príklady

- sieťový súborový (alebo aj iný) server
 - Musí riešiť rad požiadaviek klientov
 - Pre vybavenie každej požiadavky vytvára samostatné vlákno (efektívnejší ako samostatný proces)
- 1 vlákno zobrazuje menu a číta vstup od používateľa a súčasne 1 vlákno vykonáva príkazy používateľa
- prekresľovanie obrazovky súbežne so spracovaním dát

Vlákná (Threads)

- programy je možné rozdeliť na vlákna, ktoré sa budú vykonávať paralelne (ale stále v rámci jedného programu)
- každý proces má 1 základné vlákno vytvorené automaticky
- rozdelenie na vlákna musí nadefinovať autor programu



Vlákná (thready)

- vytvorenie procesu je pomerne časovo náročná operácia - operačný systém musí prečítať do pamäte nový exe súbor, prideliť mu pamäť a vykonať veľa administratívy.
- Procesy môžu zdieľať dáta len pomocou súborov alebo pomocou posielania správ.



Kedy je vhodné vlákna využiť

- ak aplikácia musí reagovať na stlačenie klávesov/klikanie a súčasne niečo počítať (jedno vlákno čaká na vstupy a druhé robí výpočty)
- ak má serverovská aplikácia komunikovať po sieti s viacerými klientmi (pre každého klienta vytvorí nové vlákno)



Petriho siete

ÚVOD DO MODELOVANIA

Model - vytvorený systém, ktorý je určitým (niekedy i podstatným spôsobom) zjednodušením originálu modelovaného systému (Peringer, 2006)

- Modelovanie v zmysle výskumnej techniky je náhrada skúmaného systému jeho modelom, jeho cieľom je získať pomocou pokusu s modelom informácie o pôvodnom skúmanom systéme (Křivý et al., 2001)

Klasifikácia modelov (Strogatz, 2007)

- Konceptuálne modely
- Deklaratívne modely
- Funkcionálne modely
- Modely popísané rovnicami
- Priestorové modely
- Multimodely



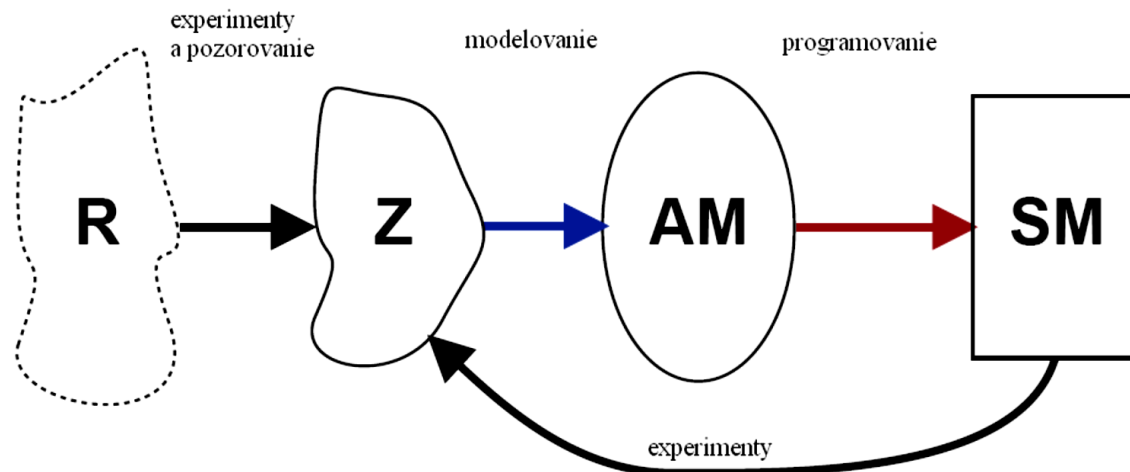
ÚVOD DO MODELOVANIA

Simulácia - metóda získavania nových poznatkov o systéme prostredníctvom experimentovania s jeho modelom (Chung, 2004)

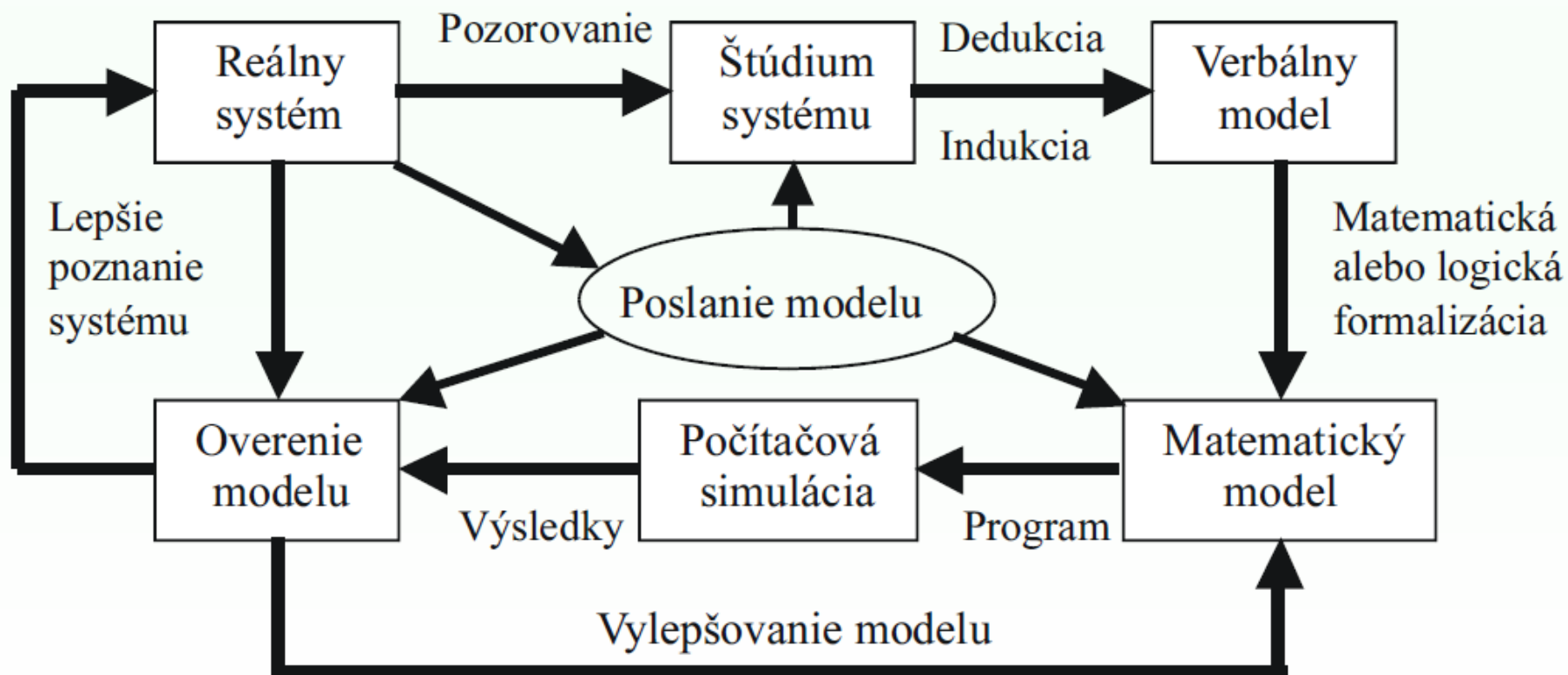
- Precízne vybudovaný model so správne nadefinovanou štruktúrou je základnou podmienkou korektne realizovanej simulácie (Dlouhý, 2005)

Proces konštrukcie modelu (Adamuščíková, 2007)

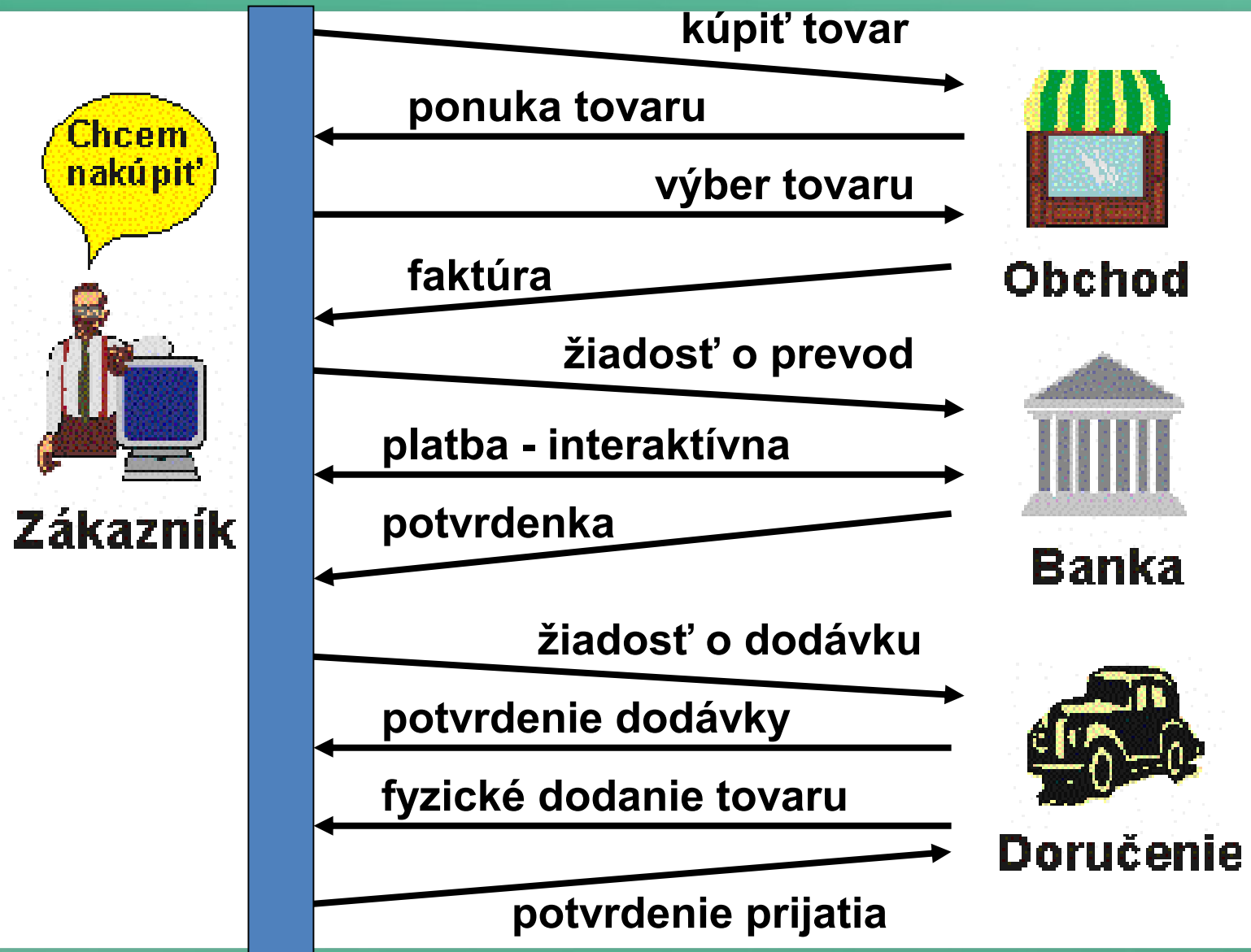
Realita → Znalosti → Abstraktný Model → Simulačný Model



ZOSTAVOVANIE MODELU



Príklad aplikácie: E - commerce



Vlastnosti uvedeného systému z hľadiska modelovania

paralelný

súbežný (concurrent)

dynamický

distribúovaný

závislý od spoľahlivosti komunikačného protokolu



Ktorý nástroj je schopný
modelovať a následne simulovať
systém vykazujúci uvedené
vlastnosti?



Petriho siete

vznikli v 60.-70. rokoch (Carl Adam PETRI, Nemecko)

sú grafickým a matematickým nástrojom vhodným na opis a analýzu systémov, ktoré sú:

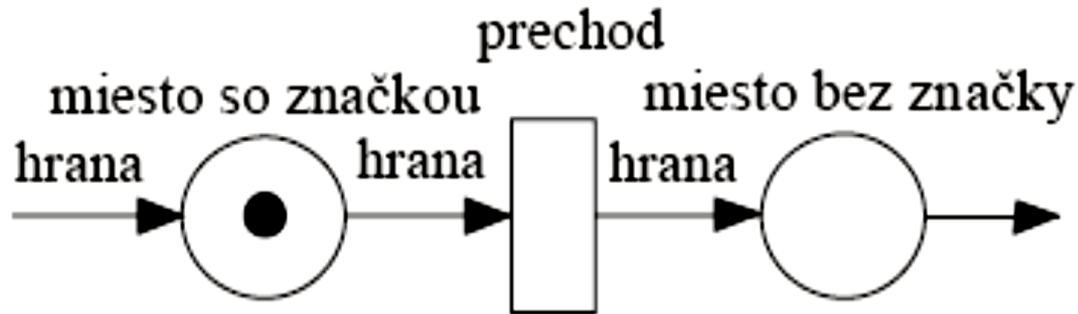
- súbežné (concurrent)
- asynchrónne
- distribované
- paralelné
- nedeterministické (stochastické)

PS je orientovaný, bipartitný graf obsahujúci uzly dvoch typov: miesta (places) a prechody (transitions)



ÚVOD DO PROBLEMATIKY PETRIHO SIETÍ

Petriho siete (PN) je orientovaný, bipartitný graf obsahujúci uzly dvoch typov **miesta** (places) a **prechody** (transitions)



Petriho siete (PN) sú grafickým a matematickým nástrojom vhodným na opis a analýzu systémov, ktoré sú

- súbežné (concurrent)
- asynchrónne
- distribuované
- paralelné
- nedeterministické (stochastické)

ÚVOD DO PROBLEMATIKY PETRIHO SIETÍ

Formalizácia

Zovšeobecnená (generalized) Petriho sieť (PN) je päťica

$$N = (P, T, \text{preset}, \text{postset}, M_0)$$

$P = \{p_1, p_2, \dots, p_m\}$ je konečná množina miest

$T = \{t_1, t_2, \dots, t_n\}$ je konečná množina prechodov

preset: $P \times T \rightarrow N$ je násobnosť vstupných hrán

postset: $T \times P \rightarrow N$ je násobnosť výstupných hrán

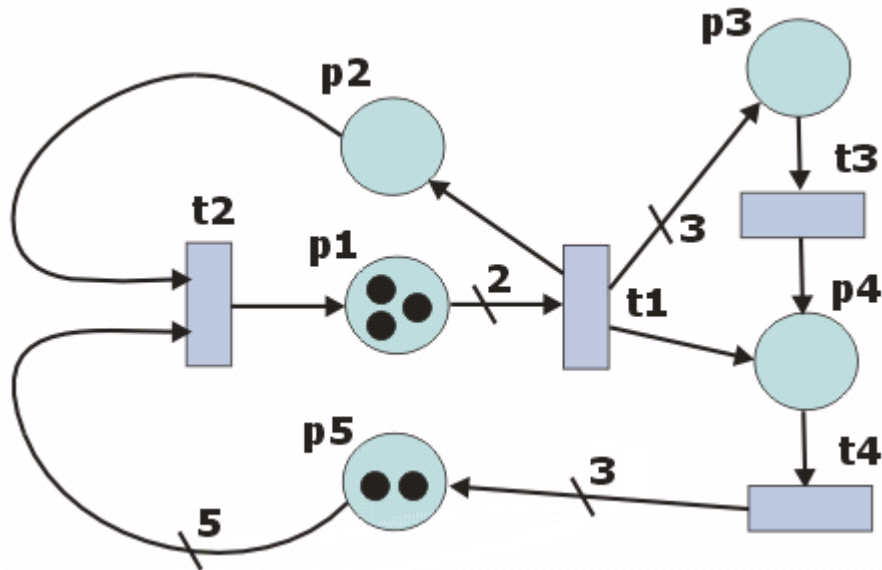
$M_0: P \rightarrow N$ je počiatočné označenie* PN

pričom platí $P \cap T = \emptyset$ a $P \cup T \neq \emptyset$

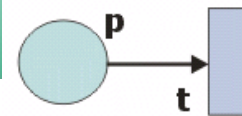
*** Označenie** (*marking*) priradí každému miestu nezáporné, celé číslo. Ak označenie priradí miestu ***p*** nezáporné celé číslo ***k***, hovoríme, že ***p*** je **označené** (*marked*) ***k*** tokenmi (*tokens*)



Príklad PS

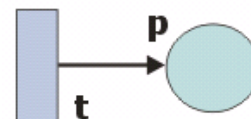


- $P = \{p_1, p_2, p_3, p_4, p_5\}$
- $T = \{t_1, t_2, t_3, t_4\}$
- $M_0 = (3, 0, 0, 0, 2)$



$$\Leftrightarrow \text{pre}(p,t) \neq 0$$

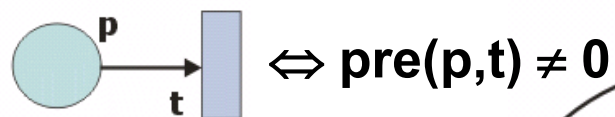
P	T	pre(p,t)
p ₁	t ₁	2
p ₂	t ₂	1
p ₅	t ₂	5
p ₃	t ₃	1
p ₄	t ₄	1
inak		0



$$\Leftrightarrow \text{post}(p,t) \neq 0$$

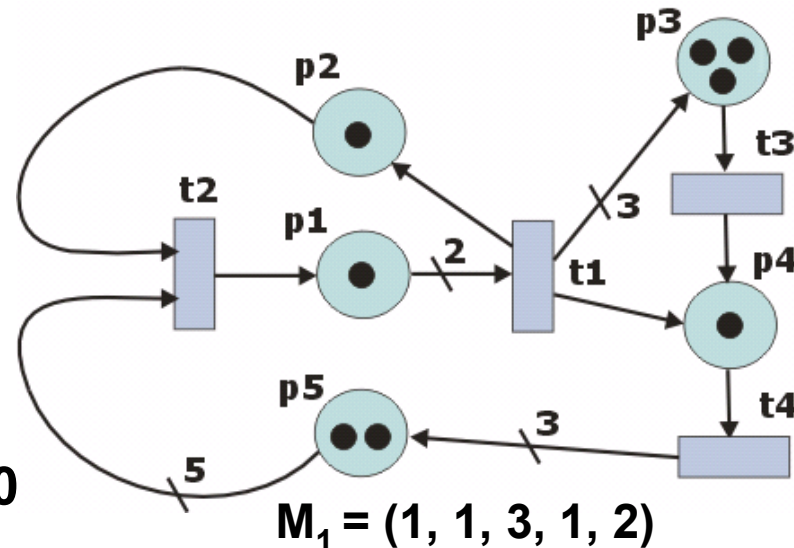
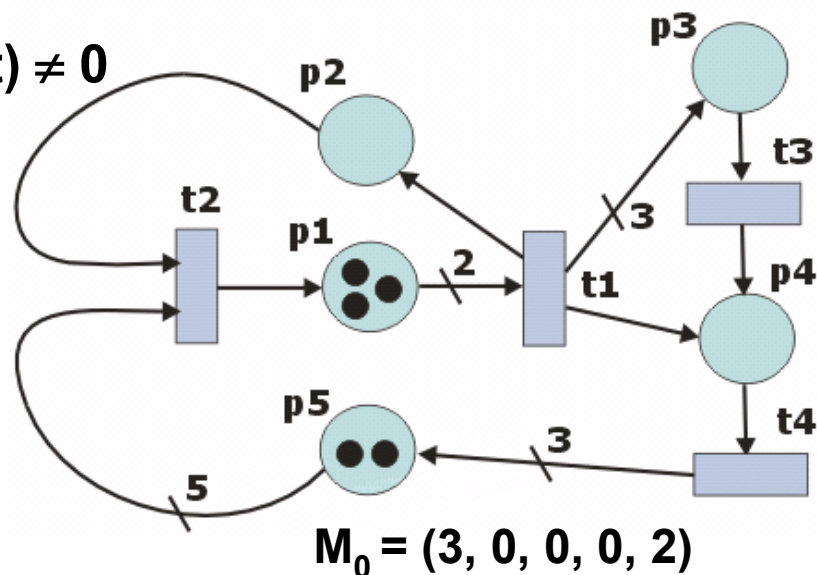
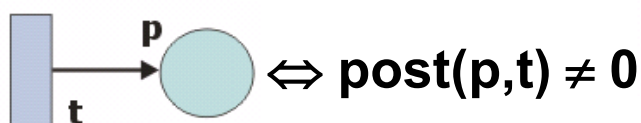
P	T	post(p,t)
p ₁	t ₂	1
p ₂	t ₁	1
p ₃	t ₁	3
p ₄	t ₁	1
p ₄	t ₃	1
p ₅	t ₄	3
inak		0

ÚVOD DO PROBLEMATIKY PETRIHO SIETÍ



- Prechod t je **umožnený (enabled)** ak každé vstupné miesto p prechodu t je označené minimálne **preset** (p,t) počtom tokenov

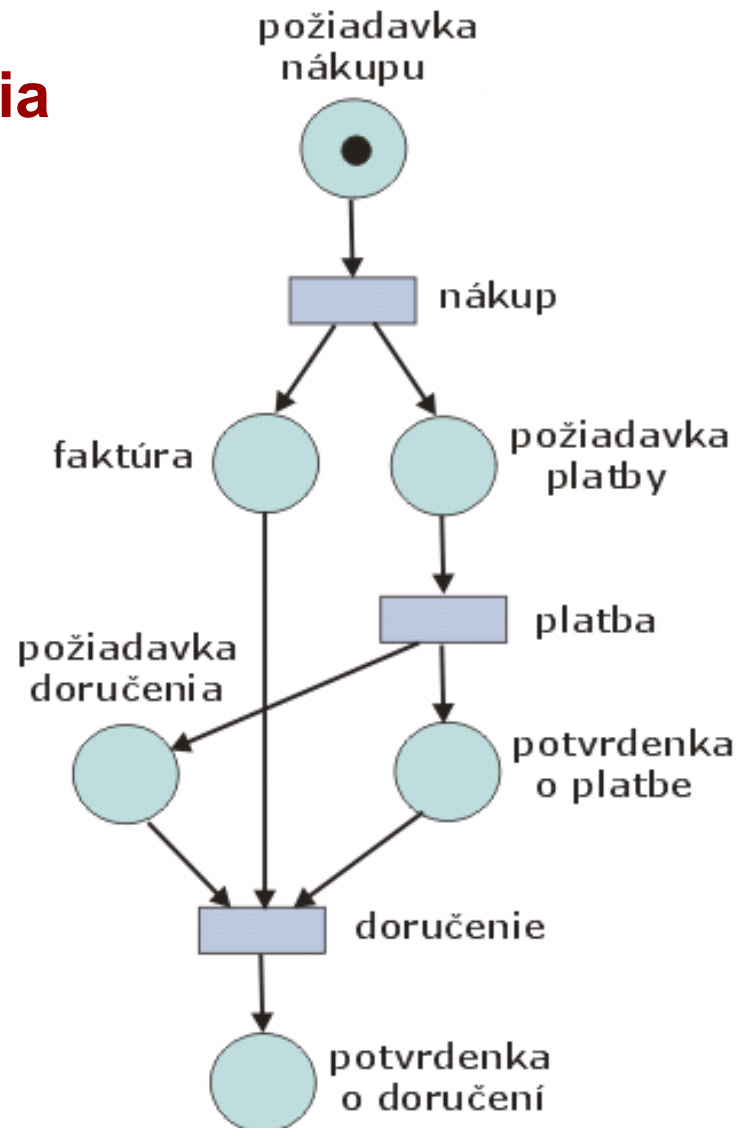
- **Uskutočnenie (firing)** umožneného prechodu t odstráni **preset**(p,t) počet tokenov z každého vstupného miesta p prechodu t a pridá **postset** (t,p') počet tokenov do každého výstupného miesta p' prechodu t



E-commerce príklad

Zjednodušená reprezentácia bez komunikačného protokolu:

- **P** - množinu miest reprezentujú dokumenty (vstupné podmienky a výstupy)
- **T** - množinu prechodov reprezentujú operácie s nimi



Úvaha...



Sú zovšeobecnené PS postačujúce na modelovanie a simuláciu reálnych systémov?

- ... vieme pomocou nich modelovať napr. to, že nejaká operácia trvá určitý čas?
- ... postačujú na označenie miest prirodzené čísla?
- ... postačuje jeden dátový typ tokenu?
- ... dá sa vyjadriť pomocou zovšeobecnených PS hierarchia systému?
- ... atď

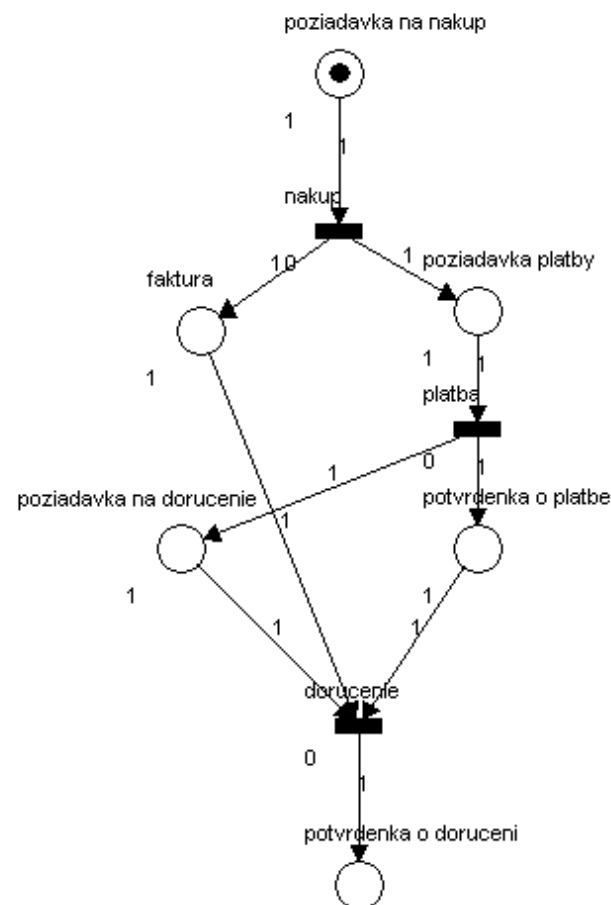
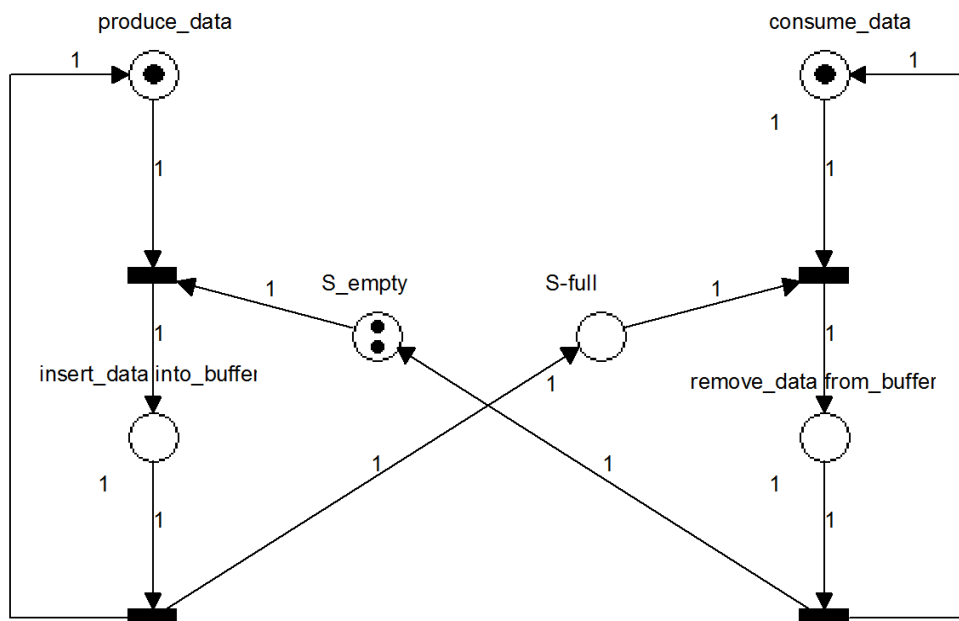


MODELOVANIE A SIMULÁCIA PROCESOV V OPERAČNÝCH SYSTÉMOCH PC

Problematika synchronizácie

- synchronizáciu dvoch alebo viacerých procesov navzájom
- odovzdávanie údajov medzi procesmi

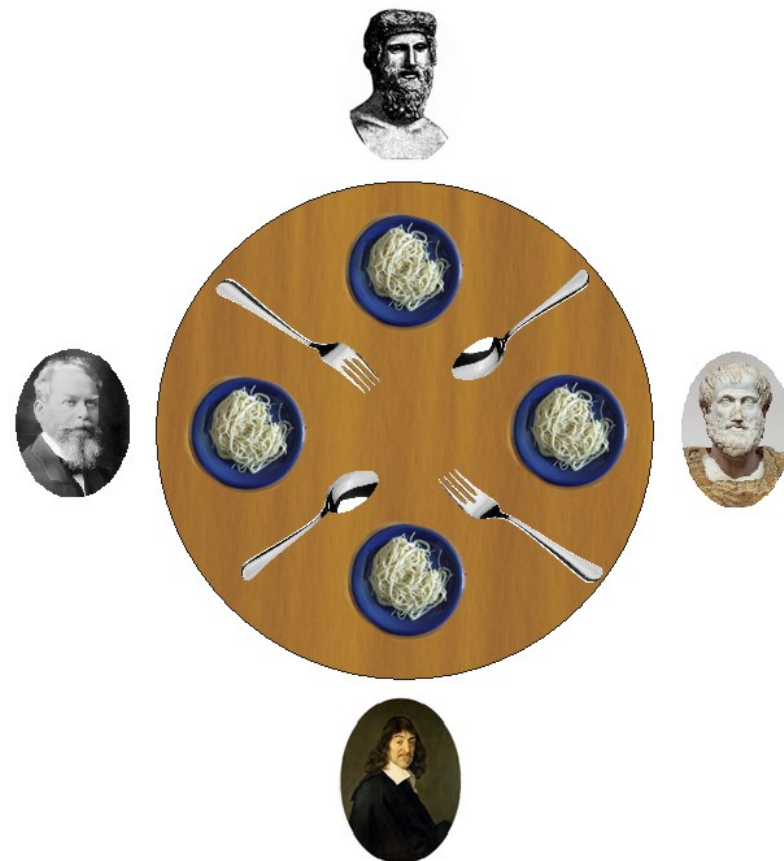
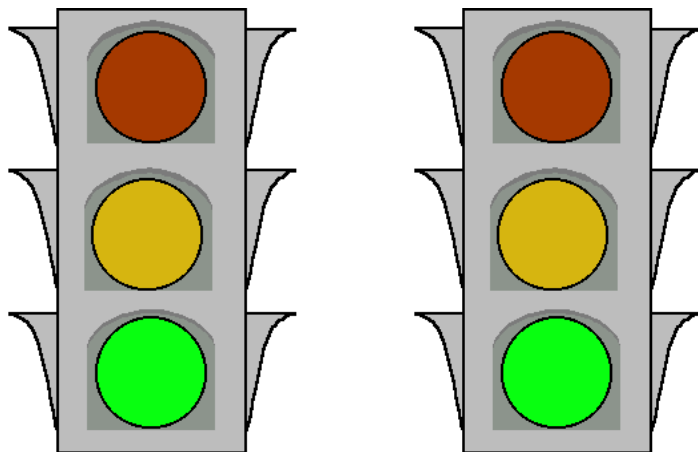
Problematika paralelizmu



MODELOVANIE A SIMULÁCIA PROCESOV V OPERAČNÝCH SYSTÉMOCH

Príklady vizualizácie problematiky synchronizácie

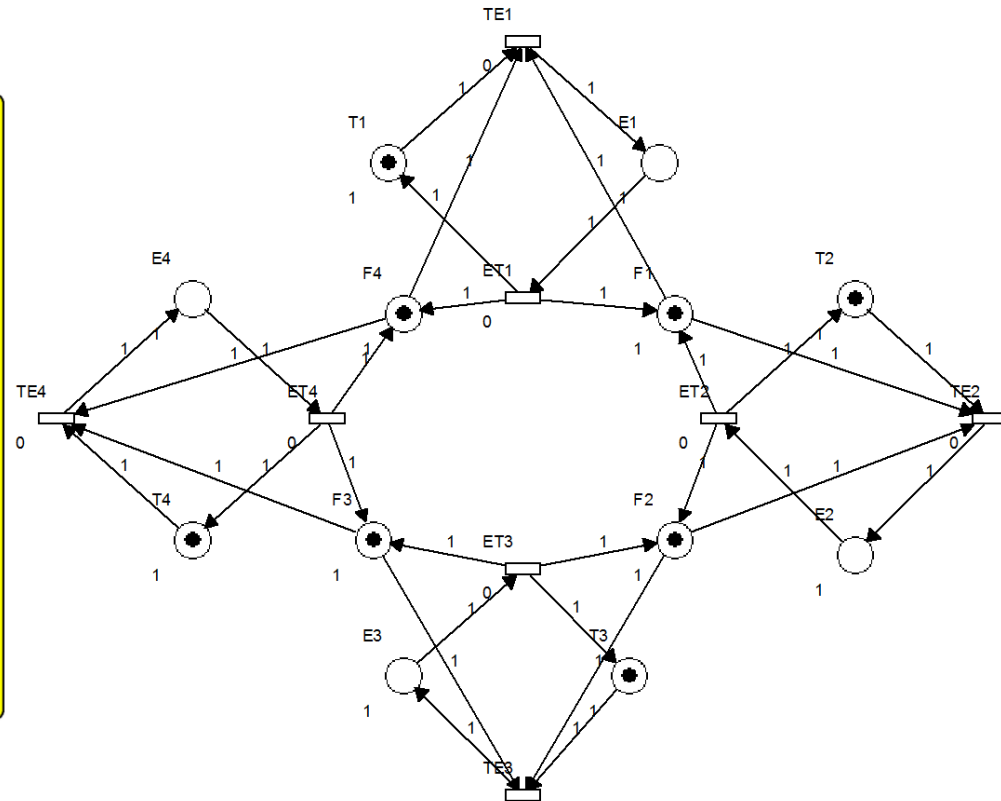
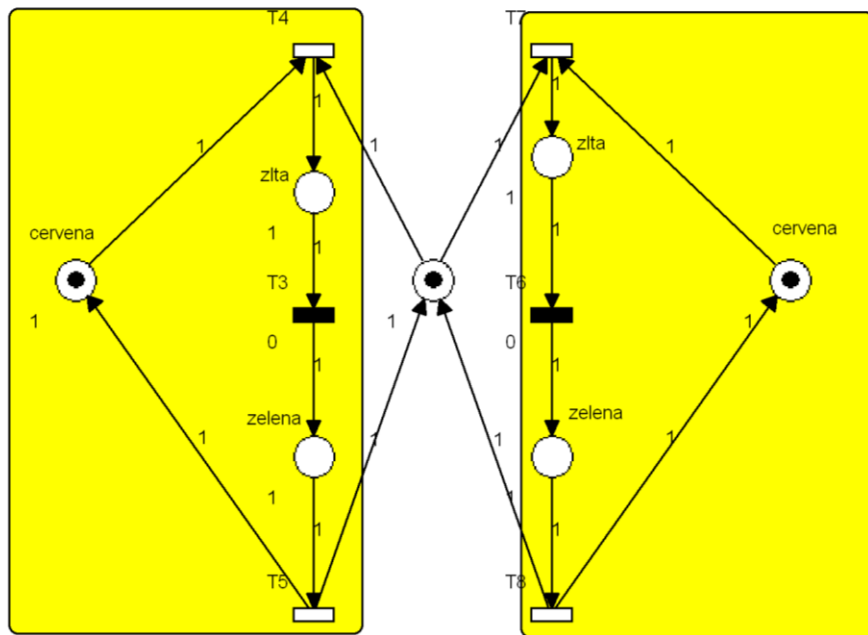
- Semaforey
- Problém hladných filozofov (The Dining Philosophers Problem) (Dijkstra, 1965)



MODELOVANIE A SIMULÁCIA PROCESOV V OPERAČNÝCH SYSTÉMOCH

Príklady vizualizácie problematiky synchronizácie

- Semaforey
- Problém hladných filozofov (The Dining Philosophers Problem) (Dijkstra, 1965)



Modifikácie Petriho sietí

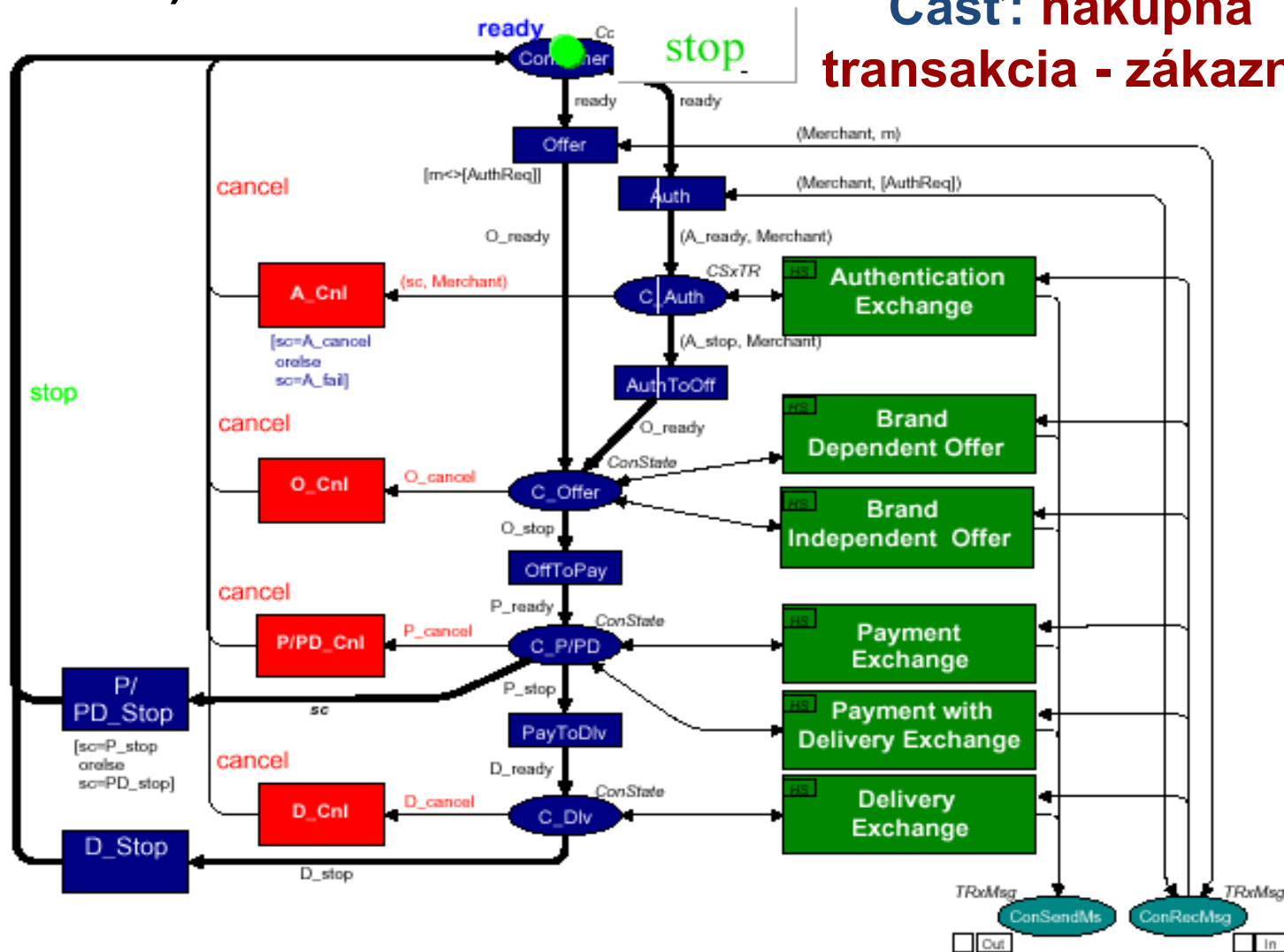
- **Deterministické časované PS**
 - deterministické časové oneskorenia prechodov a miest
- **Stochastické časované PS**
 - stochastické časové oneskorenia prechodov
- **Farebné PS**
 - tokeny s rôznymi farbami (dátové typy)
- **Spojité PS**
 - aplikácia reálnych čísel na označovanie miest
- **Hybridné PS**
 - kombinácia spojitých a diskretných miest a prechodov



E-commerce príklad

Systém je modelovaný v CSEC, South Australia. Cieľom je zefektívniť **IOTP protokol** (Internet Open Trading Protocol)

Časť: nákupná
transakcia - zákazník



Využitie Petriho sietí

Priemyselné a výrobné systémy

Sekvenčné algoritmy (nábeh linky, optimalizácia práce manipulátorov)

Komunikačné protokoly a siete (“úzke hrdlá“, neoptimálny návrh protokolu na rôznych úrovniach OSI referenčného modelu)

Modelovanie rozhodovacích procesov

Software (návrh, špecifikácia, simulácia, validácia a implementácia)

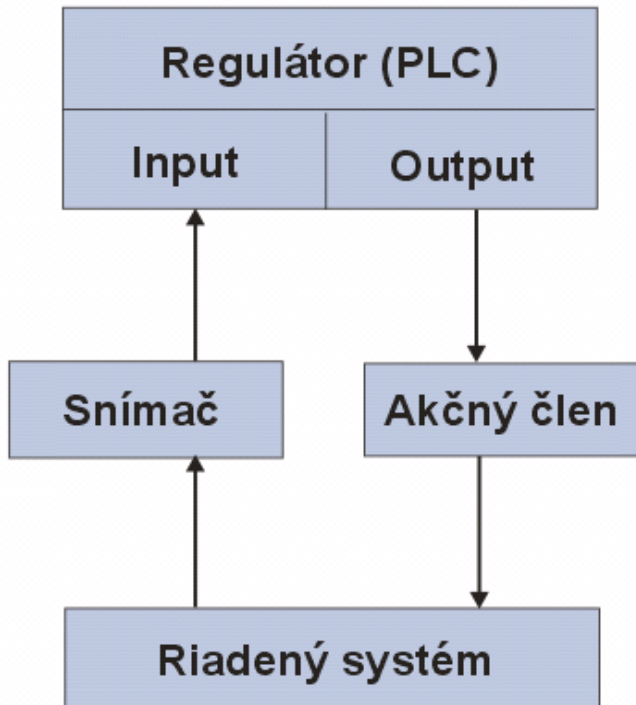
Modelovanie inteligentných regulačných obvodov so sieťovou štruktúrou (certifikácia technologických protokolov, návrh inteligentných snímačov a akčných členov)

t

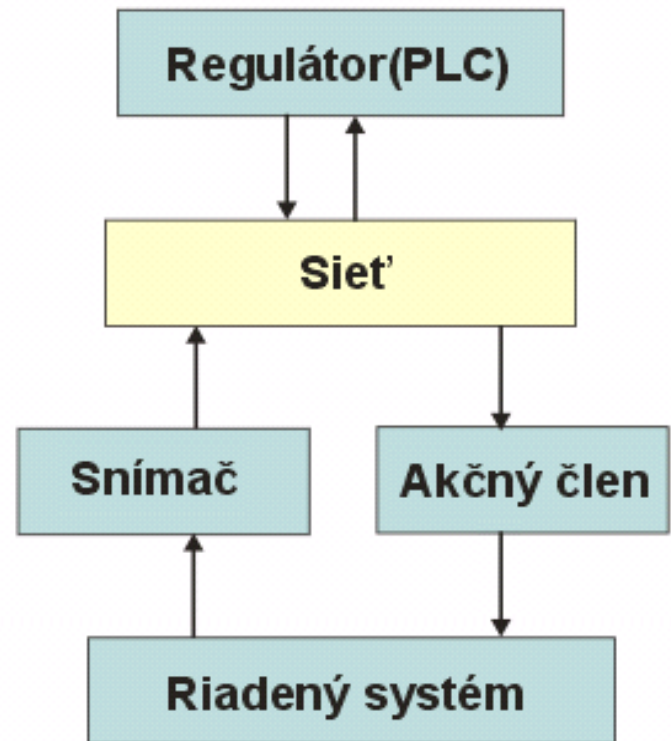


Uzavreté regulačné obvody so sieťovou štruktúrou

Klasické URO:



URO so sieťovou štruktúrou:

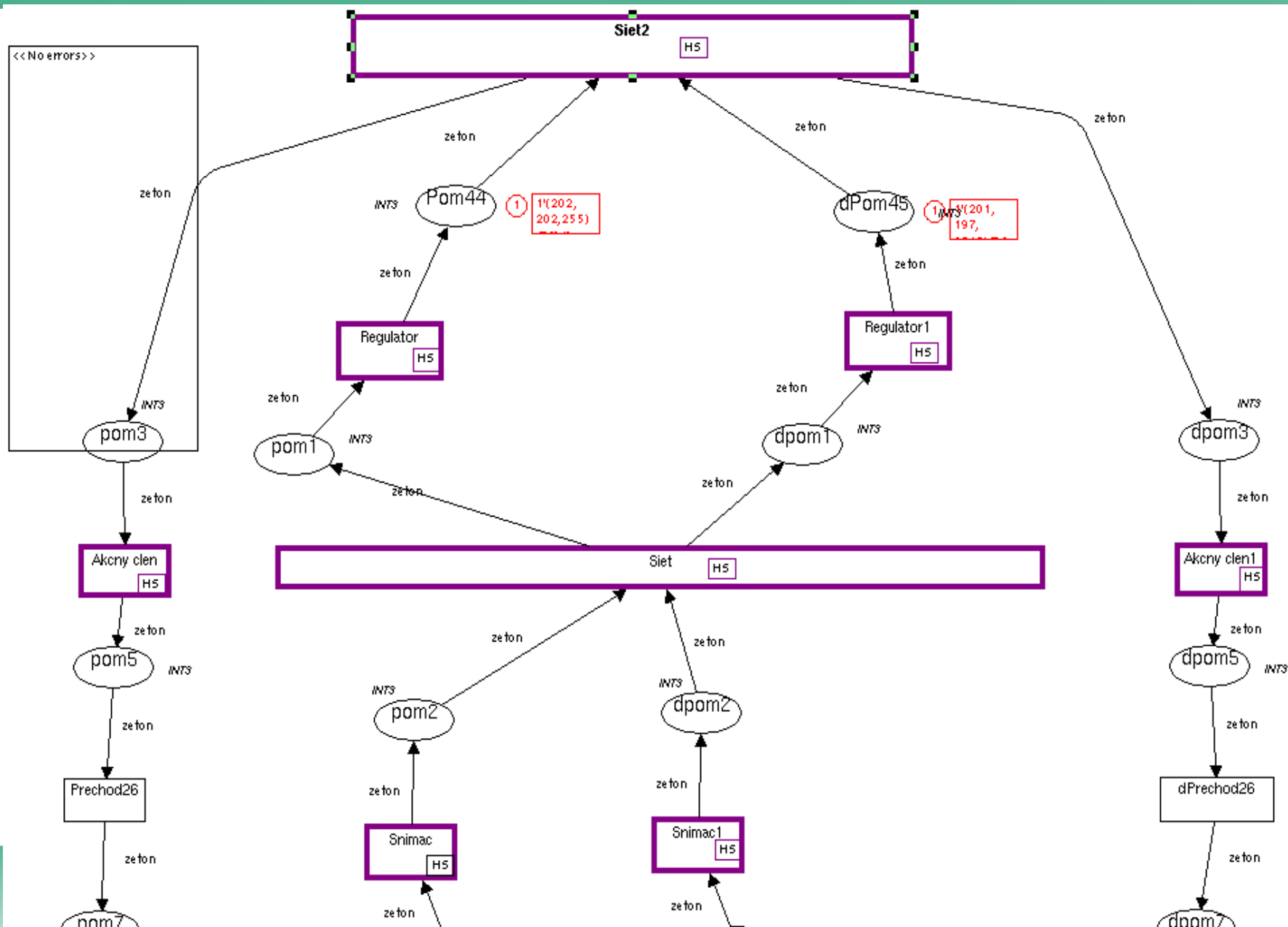


Ciele v oblasti riadenia so sieťovou štruktúrou

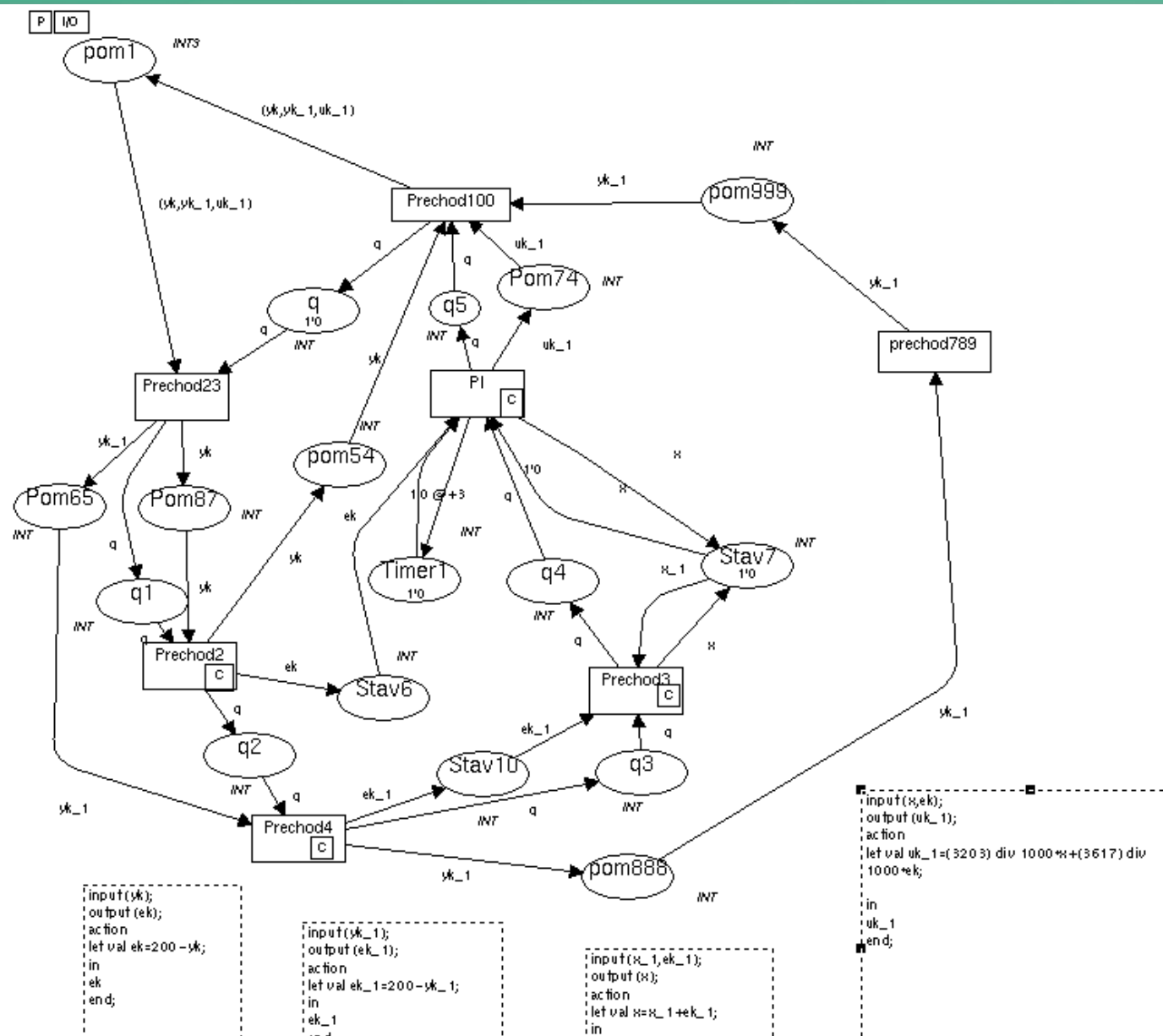
modelovanie prvkov uzavretého regulačného obvodu
simulovanie a testovanie kvality riadenia vzhľadom na možnú nestabilitu siete
návrh inteligentného protokolu, snímača, akčného člena a regulátora



Model URO v Design/CPN



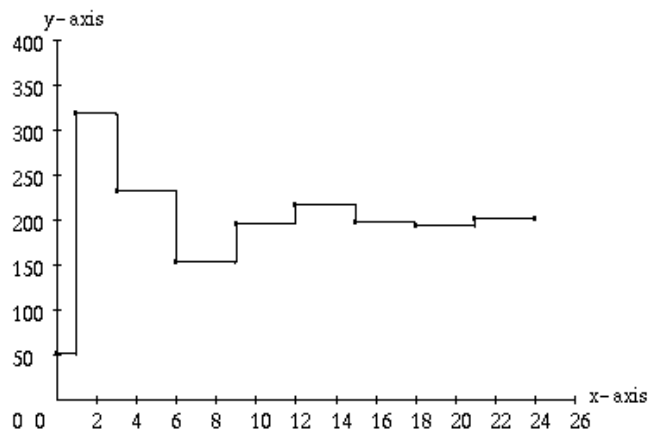
Model regulátora v Design/CPN



Grafický výstup Design/CPN programu

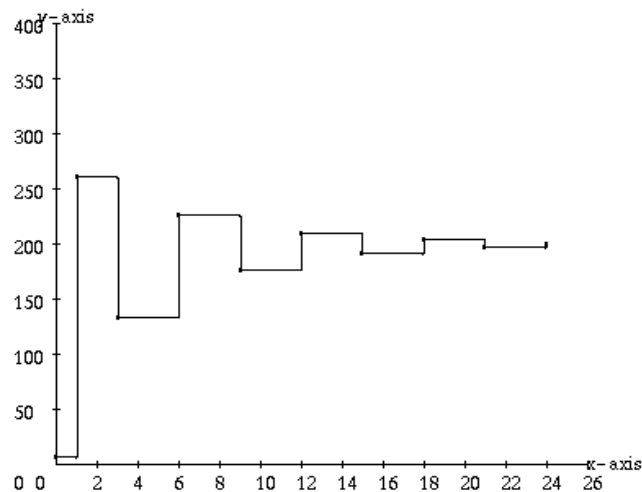
Graf P. 8

Grafický priebeh vysky hladiny v prvej nádobe



Graf P. 11

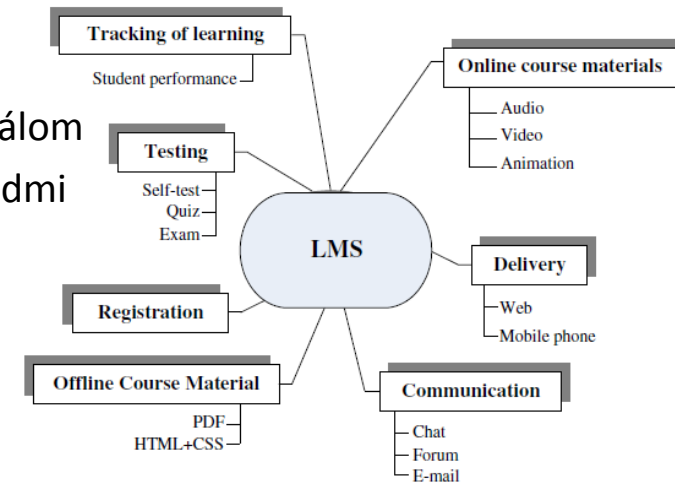
Grafický priebeh vysky hladiny v druhej nádobe



MODELOVANIE EDUKAČNÝCH PROCESOV V PROSTREDÍ LMS

Návrh metodiky tvorby modelu e-kurzu

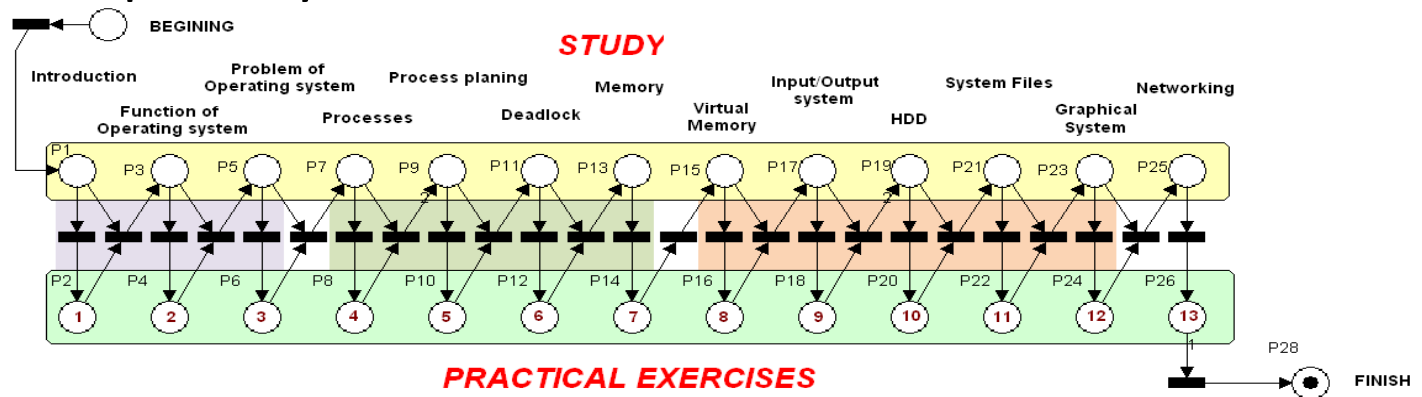
- Tvorba univerzálneho modelu e-kurzu v LMS
- Model má reflektovať potreby študentov a pedagóga
- Model má reflektovať všetky aspekty tvorenia e-kurzov
 - úvod
 - ciele štúdia
 - časový harmonogram a sprievodca študijným materiálom
 - samotný výkladový text doplnenými riešenými príkladmi priebežnými otázkami, testami a pod.
 - korešpondenčné úlohy
 - zhrnutie
 - záverečné testy
 - slovníček pojmov
 - literatúra, dôležité odkazy, prílohy a pod
- Modularita modelu
- Adaptivita e-kurzu



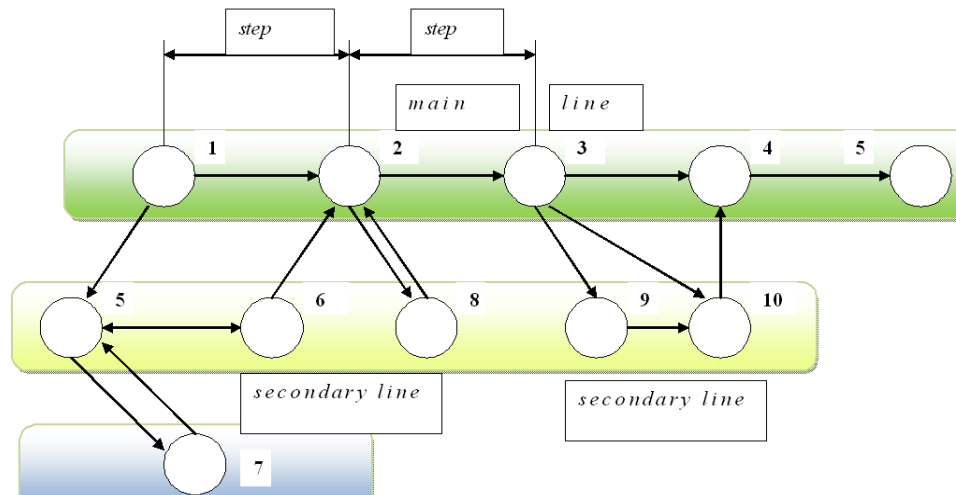
MODELOVANIE EDUKAČNÝCH PROCESOV V PROSTREDÍ LMS

Typy štruktúr modelov e-kurzu

■ Lineárne procesy

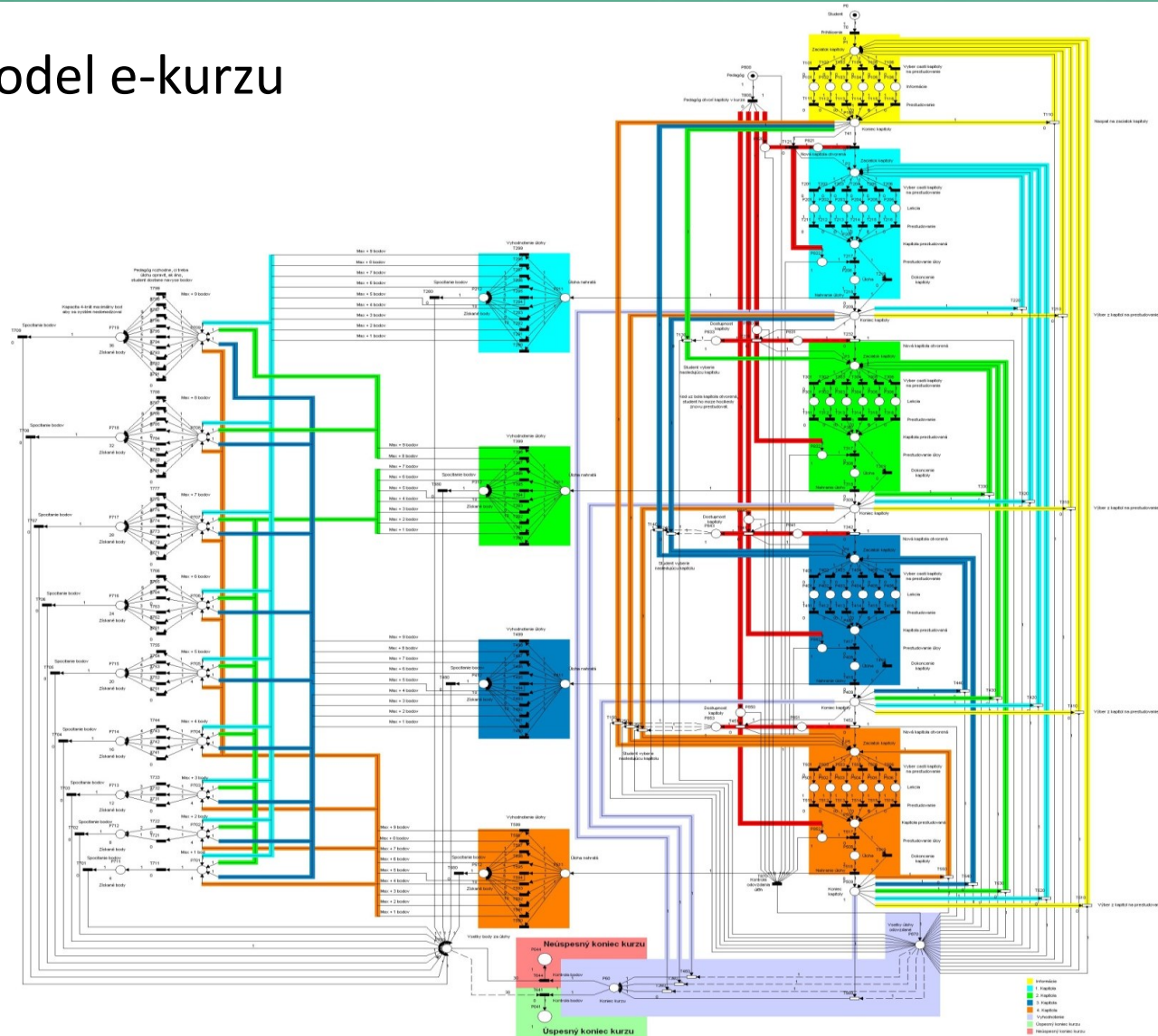


■ Vetvené procesy



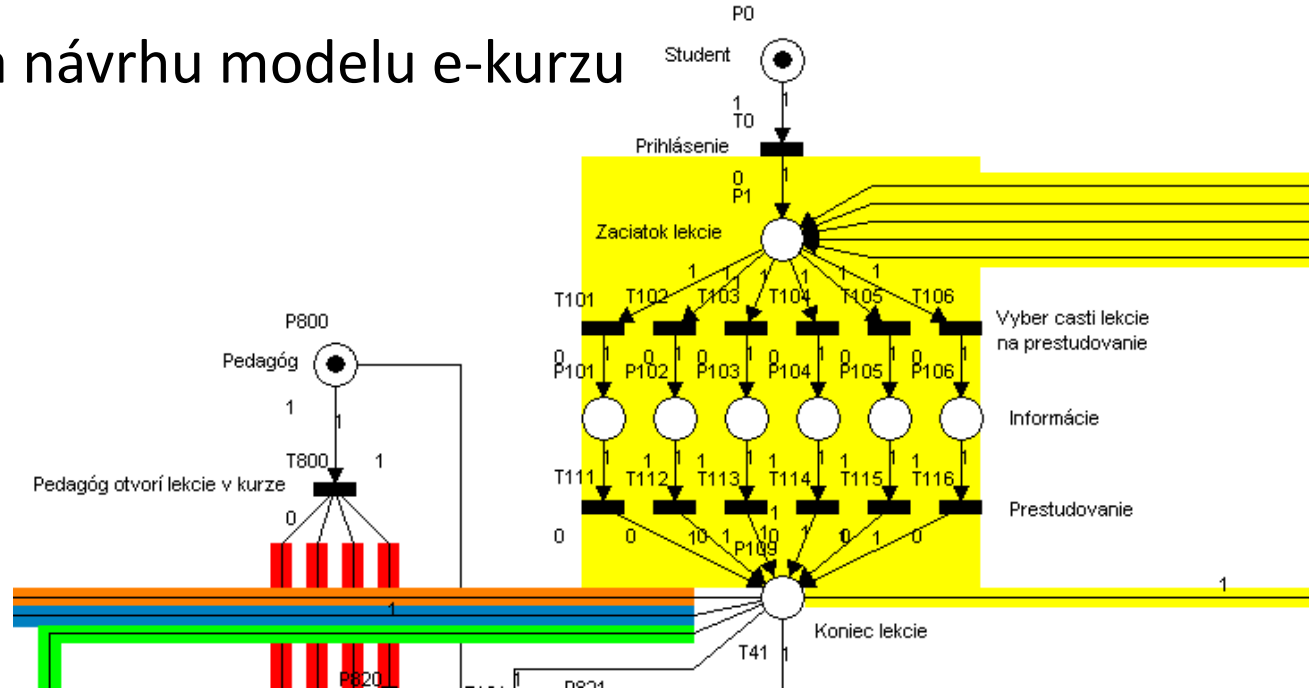
MODELOVANIE EDUKAČNÝCH PROCESOV V PROSTREDÍ LMS

Univerzálny model e-kurzu



MODELOVANIE EDUKAČNÝCH PROCESOV V PROSTREDÍ LMS

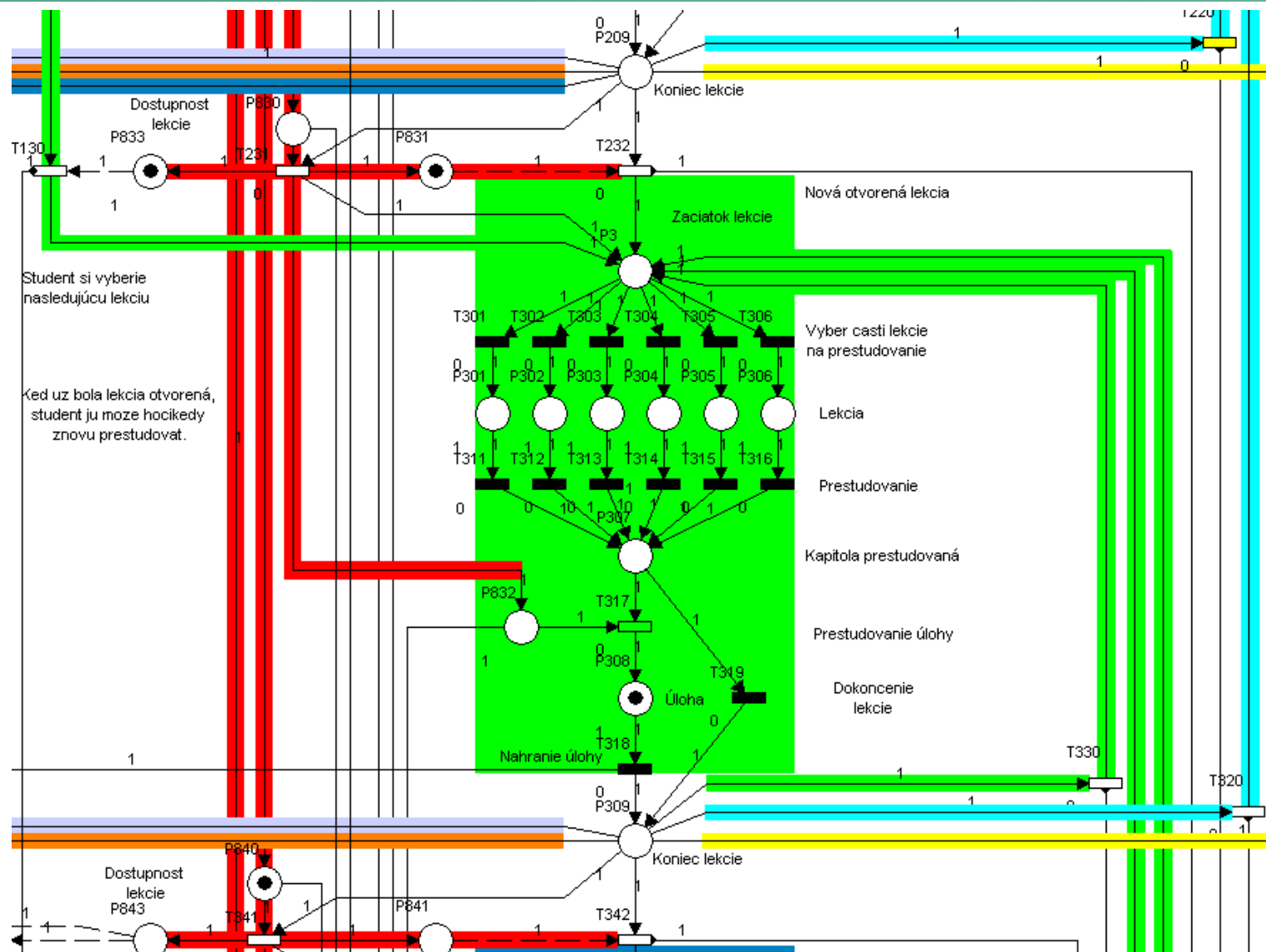
Metodika návrhu modelu e-kurzu



Počiatkový stav modelu

- Počet a obsah lekcí nie je limitovaný, PX01 –PX06, kde $N = \{1, 2, 3, \dots\}$
- Model obsahuje inhibičné a testovacie hrany, deterministické typy prechodov, miesta so stanovenou kapacitou a rôzne IF-THEN pravidlá

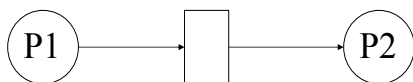
MODELOVANIE EDUKAČNÝCH PROCESOV V PROSTREDÍ LMS



MODELOVANIE EDUKAČNÝCH PROCESOV V PROSTREDÍ LMS

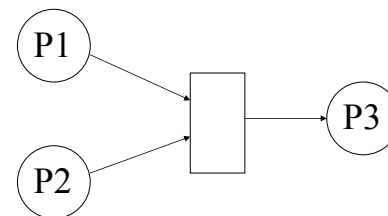
Modelovanie pomocou IF-THEN pravidiel a fuzzy Petriho sietí

IF P1 THEN P2



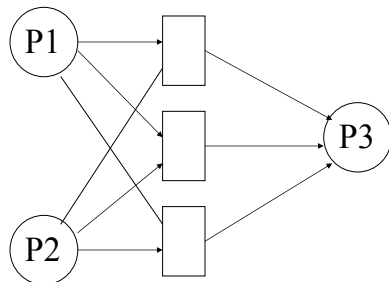
$$\alpha_2 = \lambda_t \alpha_1 \text{ ak } \alpha_1 \geq \theta_t$$

IF P1 AND P2 THEN P3



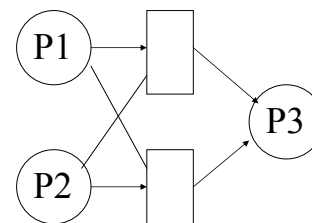
$$\alpha_3 = \lambda_t \min_{\alpha_i \geq \theta_{tAND}} \{ \alpha_1 \alpha_2 \} \text{ pre } i = 1 \wedge 2.$$

IF P1 OR P2 THEN P3



$$\alpha_3 = \lambda_{tOR} \max_{\alpha_i \geq \theta_{tOR}} \{ \alpha_1 \alpha_2 \} \text{ pre } i = 1 \vee 2.$$

IF P1 XOR P2 THEN P3



$$\alpha_3 = \lambda_{tXOR} \alpha_1 \text{ ak } \alpha_1 \geq \theta_{tXOR} \wedge \alpha_2 = 0$$

$$\alpha_3 = \lambda_{tXOR} \alpha_2 \text{ ak } \alpha_2 \geq \theta_{tXOR} \wedge \alpha_1 = 0$$

Model adaptácie výučby pomocou fuzzy pravidiel

- Pravidlá sú úlohy hľadania závislostí medzi atribútmi v tvare IF-THEN pravidla (výroky)
 - IF (príčina=true) THEN (dôsledok=true)
- Pre stanovenie konkrétneho modelu adaptácie výučby musíme definovať najprv **vstupy** modelu
 - motivácia, prostredie, pamäť, sústredenosť, životospráva, doba učenia, dĺžka učenia, čas, čítanie, prístup, teória, príklady
- Definované **výstupy** modelu
 - znalosť, pochopenie, aplikácia

MODELOVANIE EDUKAČNÝCH PROCESOV V PROSTREDÍ LMS

Modelovanie IF-THEN pravidiel

IF (prostredie = veľmi zlé AND sústredenost' = nízka) THEN znalost' = veľmi zlá

IF (prostredie = zlé AND dĺžka učenia = veľmi dlhá)
THEN znalosť = zlá

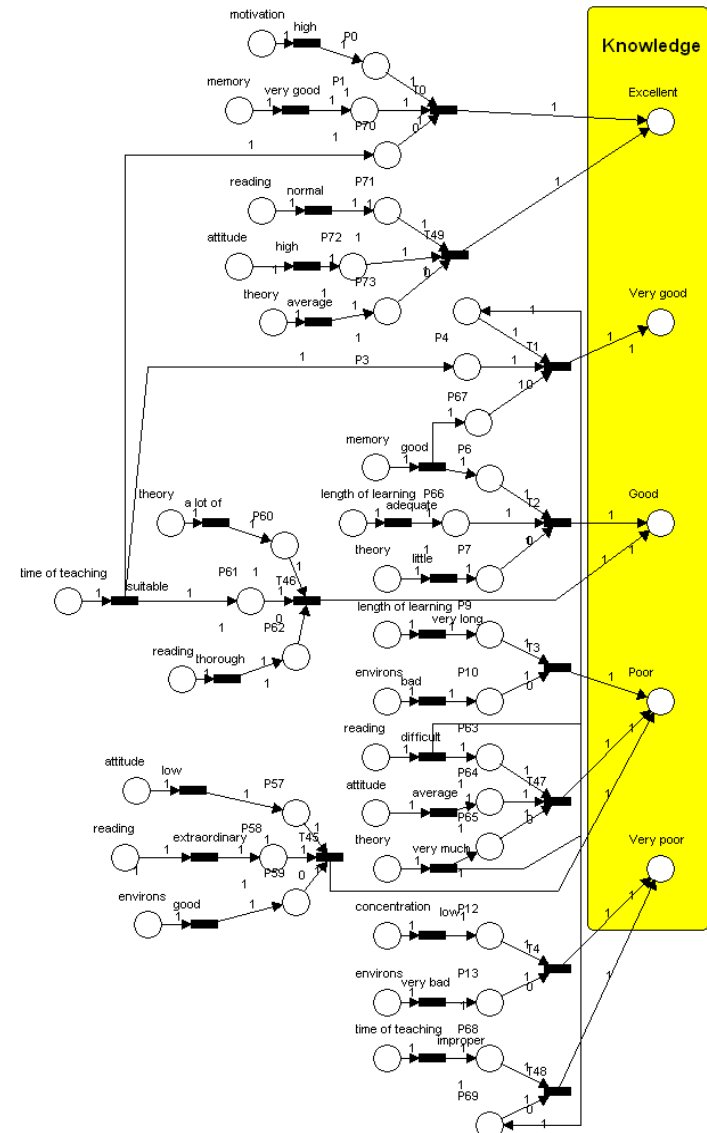
IF (prostredie = dobré AND čítanie = mimoriadne
AND prístup = nízky) THEN znalosť = zlá

IF (doba učenia = vhodná AND čítanie = dôkladné
AND teória = veľmi veľa) THEN znalosť = zlá

IF (čítanie = obtiažne AND teória= veľa AND
prístup = priemerný) THEN znalosť = zlá

IF (pamäť = dobrá AND dĺžka učenia = primeraná
AND teória = primerane) THEN znalosť =
dobrá

IF (doba učenia = vhodná AND pamäť = dobrá
AND teória = primerane) THEN znalosť =
veľmi dobrá



EVALVÁCIA VÝSLEDKOV MODELOVANÝCH EDUKAČNÝCH PROCESOV

