**SIMPROCESS:**

**1.) Clone aktivnost**

Clone (gradnik, ki je na voljo le preko create menija) iz določene entite naredi več istih entit (klonira). Zato lahko porabimo določene resource in določen duration.

**2.) Split/Join aktivnosti v Simprocessu - kaj delata, kako delujeta**

Split – razdeli entiteto na original(parent) in child, lahko se jim da neko oznako da se bojo v nadaljevanju spet združili (z join). Ima 1 vhod in 2 izhoda. Zgornji izhod je »clone«(child), spodnji »original«. Uporabno npr za začetek paralelnih procesov.
Join – Združi »družino« entitet narejeno s split. Čaka vse parente in childe da pridejo. Ima vhod in 2 izhod. Zgornji izhod je »join« kjer izhajajo entitete joinane, spodnji pa »NoMatch« kjer izhajajo entitete ki niso iz navedene družine. Če entiteta ni iz družine gre skozi »NoMatch«.

**3.) Resursno orientirane aktivnosti v Simprocessu**

ikone – aktivnost: entitetno usmerjene, resursno usmerjene (get, free, replanish)

**4.) Entiteta vs. Entitetni tip (razlika med entitetnim tipom ter entiteto)**

Entitni tip je objekt, ki defeniran določeno entito (atribute, expressions, osnovne podatke). Entite pa so instance določenega entitnega tipa, ki potujejo skozi model simulacije. Generate je gradnik, ki generira instance določenega entitnega tipa.

**5.) Kako debuggiramo v SimProcessu**

Js sm mu napisal da simulacijo debugiramo tako da poskušamo doseči RO < 1. Pol sm mu opisal da to najlažje dosežemo če z bisekcijo spreminjamo št. entitet (OAPS1 snov :) ). Pa da že med simulacijo lahko spremljamo zasedenost sistema s pomočjo tistega grafa, ki nam izrisuje št. entitet v sistemu ter št. vstopajočih entitet - kot graf pri Litlovem teoremu.

Naštel grafe zasedenosti resursov entitet + tega da imamo opcije pri simulacji da spremljamo vse entitete kako potujejo + to da izpisujemo atribute pa ni bil lih navdusen, ker ni blo nekih sistemskih klicev za debugganje, kar pa se vedno nism ugotovu kaj je to.

sistemski atributi: globalno stanje simulacije (št. Entitt v simulaciji, izpisi tipov entitet)

**6.) Dobre lastnosti Simprocessa!**

Da lahko modeliramo praktično vse primere iz realnega življenja. Simulacije se izvršijo v nekaj minutah, urah, simuliramo pa lahko tudi v obdobju nekaj let. Po končani simulaciji imamo možnost vpogleda v poročilo (zasedenost resursov itd...). Lahko uporabimo programsko kodo za programiranje exspressionov.

**7.) slabe lastnosti simprocessa**

Počasno izvajanje, simuliranje, plačljiv.

**8.) Gate**

Akumulira entitete dokler se ne zgodi en »event«. Event je lahko npr: določena entiteta ko pride v gate, lahko je expression, itd...ko pride do »eventa« se entitete spustijo

**9.) entitetni atributi, entitetni expression-i**

Poznamo globalne in lokalne atribute, expression-e. Globalna atribut je defeniran za vsako entito, medtem ko je lokalni atribut določen le pri določenemu tipu entite, naprimer pri entiti tipa Avto dodamo atribut težo. Atributi nam omogočajo, da kreiramo/dodamo določene lastnosti entitam, ki jim Simprocess v osnovi ne podpira. Da so atribut entite koristni potrebujemo expressione, ki nam omogočajo da lahko te atribute naslavljamo/spreminjamo. Expresion je v defenirana rutina, ki vsebuje določene klice oz. instrukcije. Simproces preverja/izvaja te expresione na različnih mestih in večkrat v teku simulacije.

**10.) implicitno zasedanje**

Zasedemo vedno posebej.
Ekstremne vrednosti v povprečju.

**Implicitno** ( ***Poštena do vseh*** )

→ D1 → D2 → D3 →

R1 R1 R1 *STRATEGIJA ODELJEVANJA*

**11.) eksplicitno zasedanje**

Zasedemo na začetku in sprostimo na koncu.
Ekstremizira best, worst case. KRIVIČNO

**Eksplicitno** ( ***Favorizira prvoprispele*** )

→ G → D1 → D2 → D3 → F →

R1

**12.) dualnost petrijeve mreže**

Dualna petrijeva mreža je mreža kjer so akcije in pogoji zamenjani

*13.) naštet še ene 5 sistemskih klicev*

**POISSONOV PROCES:**

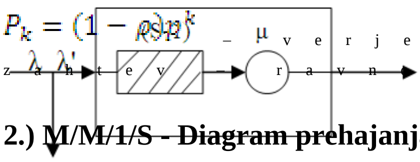
**1.) Kaj pomeni M/M/1?**

M/M/1 je poissonov proces z 1 strežnikom. Neskončna kapaciteta sistema, neskončna populacija, FIFO strežna disciplina.

$\lambda_k = \lambda, \mu_k = \mu$ Neodvisnost obeh intenzivnost od trenutnega števila zahtev v sistemu – neodvisnost intenzivnosti od stanja sistema (stanje sistema ponazarja število zahtev v njem)
Prihodni in strežni časi so eksponentno porazdeljeni.

$P_0 = 1 - \rho$ – verjetnost, da je sistem prazen

$P_k = (1 - \rho)\rho^k$ – verjetnost, da je v sistemu k zahtev ravnotežna enačba




**2.) M/M/1/S - Diagram prehajanja stanj**

λ*$P_b$(verjetnost zasedenosti sistema $P_b$=$P_s$)
λ'≤λ

*S -> 1 streznik*
*S-1 mest v vrsti*
*- ko je v sistemu S zahtev začne sistem zahteve odklanjati(»izguba!«)*

**3.) markovska veriga za M/M/1**

$$P_k = \frac{(1-\rho) * \rho^k}{1-\rho^{s+1}}$$

s – končna kapaciteta sistema: v sistemu se lahko nahaja največ (s-1) zahtev v vrsti in 1 zahteva v edinem strežniku (s-1)+1=s. Diagram prehajanja stanj je omejen na desni strani.

Če je v sistemu s zahtev, se novoprispele zahteve izgubljajo: $\lambda = \lambda' + \lambda * p_b$

$p_b$ – verjetnost polnega sistema in s tem izgube novoprispele zahteve

$\lambda'$ - intenzivnost zahtev, ki so vstopile v nezaseden sistem

$\lambda' = \lambda * (1 - p_b)$

**4.) M/M/m**

*imamo paralelno vezanih* ***m****, funkcionalno ekvivalentnih strežnikov. Vsaka zahteva obdela natanko en strežnik. SHEMA ROJEVANJA IN SMRT.*

**5.) poissonov proces**

Gre za proces, v katerem so časi porazdeljeni eksponentno (M). Procesi M/M so poissonovi procesi. Eksponenta gostota verjetnosti: $P_n(t) = \frac{(\lambda * t)^n}{n!} * e^{-\lambda t}$.

Verjetnost porajanja k točk na intervalu (0,t): $P(X(t) = k)$. Povprečno število pojavitev točk v časovnem intervalu $\lambda * t$

Lastnosti:

- Superpozicija
- Eksponenta porazdelitev časov <-> PP
- Verjetnost ne porajanja zahteve na (0,t): $P_0(t) = 1 - e^{-\lambda t}$
- Čas porajanja nove zahteve je neodvisen od časa porajanje predhodne zahteve

6.) diskretna časovna markovska veriga

**PETRIJEVE MREŽE:**

**1.) Formalna definicija Petrijevih mrež.**

Graf PM je bipartitni (dvodelni) usmerjeni graf.

**2.) Petrijeva mreža (kaj jo sestavlja? prehodi, stanja itd.)**

Petrijevo mrežo sestavljajo:

- Pogoji
- Akcije
- Povezave
- Žeto

3.) Nedeterminizem Petrijevih mrež - kaj je to, ter podaj preprost primer (nariši)

**4.) Konzervativnost Petrijevih mrež**
Striktna konservativnost. Št. žetonov v PM je skozi dinamiko konstantno (skozi čas). Konservativnost -Št. žetonov skozi čas gledano skozi vse pogoje se ne spreminja. Striktna (gledano po vseh pogojih) nestriktna (samo na določen del pogojev) zagotovljena v primeru, ko za vse akcije velja, da je število vhodnih povezav enaka številu izhodnih povezav

**5.) Kakšen je označen Petrijev graf (z žetoni)?**
Označevanje = porazdeljevanje žetonov po pogojih.

Trenutno stanje PM = Trenutna označitev PM
Št. Žetonov v pogoju = kratnost izpolnjenega pogoja
Označitev v času t v PM: $o(t) = (o_1, o_2, \dots, o_n)$, n – moč množice pogojev (število pogojev)
**Definicija: Označitev PM je funkcija, ki množico pogojev P preslika v vektor celih nenegativnih števil!.**
Vsaka mreža ima potencialno neskončno možnih označitev.

**6.) inverzna petrijeva mreža (smer puščice se spremeni)**
V inverzni petrijevi mreži se obrne smer puščic.

7.) Pretvorba avtomata v petrijevo

**8.) Varnost petrijevih mrež - Kako spremenimo ne varno PM v varno PM**
Za vprasanje sem napisal kdaj so varne in kdaj niso varne ter da jih spremenimo z dodajanjem pogojev, kjer pogoj ni varen. Samo je hotu se tisti 2 slikci k smo jih narisal in pa tiste pogoje k so zraven.

PM varna takrat oz. pogoj je varen takrat, ko je v njem natanko en žeton. Če so vsi pogoji varni pol je tudi PM varna. Če obstaja en pogoj, ki ni varen vpeljemo nov pogoj p_i'

**9.) Petrijeve mreže: drevo označitev**
zakaj rabimo, itd... Da ugotovimo, po kolikih n-korakih lahko iz začetne označitve pridemo, do neke druge označitve. V tem primeru lahko pridemo do deadlocka (konec), zanke ali pa nekončnega zaporedja označitev.Vozlišča v drevesu predstavljajo označitve (število žetonov v vsakem pogoju).
Drevo označitev: Končna označitev, Cikličnost označitev, Neskončna veja

10.) stražar v PM

**11.) omejenost PM**
k-omejenost: maximalno št. žetonov v posameznem pogoju skozi čas (v vseh pogojih)

**OSNOVE TEORIJE STREŽBE in OSTALO:**
**1.) Stacionarne značilnosti sistemov! (kaj vse vpliva na i?)**
Stacionarna stanja so takrat, kadar v sistemu obstajajo stacionarne limite stanj.
Da imamo stacionarne limite stanj mora diskretni časovni Markovski proces imeti:

- aperiodičnost

- nereducibilnost
- časovno homogenost

Če to imamo, ima sistem verjetnost definirano kot:
Reducibilnost: Sistem je reducibilen, če vsebuje več kot 1 izolirano podmnožico stanj.
Aperiodičnost: je nasprotje periodičnosti. Markovski sistem je periodičen s periodo t, če se po vsakih nt (n = 1,2, ...) korakih vrača v isto stanje
Časovna homogenost: če se $P_{ij}$ v času ne spreminja.

2.) Kapaciteta sistema

**3.) Strežna enota, strežna mreža. Kaj je eno, kaj je drugo.**
Strežna enota ima eno čakalno vrsta, N paralelno vezanih strežnikov in ponuja 1 strežbo.
Strežna mreža je poljubna vezava m strežnih enot (m > 1).

**4.) Rojstno smrtni sistem**
Rojstno smrtni proces je posebna oblika Markovske verige, kjer so prehodi možni le v »sosednja stanja« Smrt in rojstvo se ne moreta zgoditi istočasno.

**5.) Stohastični procesi (lastnosti)**
Stohastični proces je proces, v katerem je dinamika vsaj deloma odvisna od statistično pogojenih spremenljivk.
SP determiniran z:

- Prostorom stanj: X(t), zaloga stanj končna
- Časovnim parametrom
    - Diskretni čas – stohastično zaporedje ($X_n$)
    - Zvezni čas – $X(t)$

| | Diskretni prostor | Zvezni prostor |
|---|---|---|
| Diskretni čas | Diskretno časovne stohatstične verige | Diskretno časovne stohatstični proces |
| Zvezni čas | Zvezno časovne stohatstične verige | Zvezno časovni stohatstični proces |

**6.) Littlov teorem**

$$N = T * \lambda$$

$$N = \frac{N(t)}{t}$$

**10.) stacionarnost, reducibilnost, Peroidičnost, Povrnljivost Markovske verige**
Stacionarnost: obstajajo limitne vrednosti verjetnosti stanj, ki niso odvisne od začetne porazdelitve verjetnosti.
reducibilnost: je nereducibilna, če je vsako stanje dosegljivo iz vseh ostalih stanj v končnem št. korakov. Je reducibilna, če vsebuje več kot eno izolirano podmnožico stanj.
Peroidičnost: sistem M.V. je periodičen s periodo t, če se po t*n korakih vrača v isto stanje M.V.
Povrnljivost (rekurenčnost): povrnljivo stanje j je tiso, pri katerem je verjetnost nahajanja v stanju j pri zelo velikem številu korakov (lim→neskončno) večja od 0. $\lim_{k\to\infty} \Pi_j^{(k)} > 0$