

Univerza v Mariboru
Fakulteta za organizacijske vede

Optimiranje in simulacija
sistemov

doc. dr. Andrej Škraba

Laboratorij za kibernetiko in sisteme za
podporo odločanju

Vsebina

- Osnove simulacij
 - Koncept sistema
 - Modeliranje sistemov
 - Definicija dogodkovne simulacije
 - Vrednotenje simulacije
- Simulacijski jeziki
- Statistične osnove simulacije
 - Stohastična spremenljivka
 - Generiranje naključnih števil
- Modeli strežbe in vrst
 - Modeli strežbe
 - Distribucija časov med prihodi
- Simulacija diskretnih dogodkov
 - Diskretni dogodki
- GPSS
 - Osnove, bloki, funkcije, primeri
- Planiranje eksperimenta
 - Optimiranje sistemov

Literatura

- M. Kljajić, I. Bernik, A. Škraba: Skripta: Dogodkovna simulacija sistemov, 1999
- M. Kljajić: Teorija sistemov, Moderna organizacija, Kranj, 1994
- G. Gordon: System Simulation, Prentice-Hall, Inc. Englewood Cliffs, New Jersey, 1969
- A. M. Law in D. W. Kelton: Simulation Modeling and Analysis, McGraw-Hill, Boston MA, 2000
- J. Banks (ur.): Handbook of Simulation, Wiley, New York NY, 1998
- F. L. Severance: System Modeling and Simulation, Wiley, New York NY, 2001

Sistem

- Je množica elementov in povezav:

$$S = \{E, R\}$$

- Ob tem mora obstajati cilj oz. namen sistema.
- Sistem je skupina po določenih zakonih, ki so odvisni od namena, povezanih, soodvisnih enot, ki sestavljajo zaključeno celoto.
- Vsakemu elementu pripadajo določene lastnosti oz. atributi in dejavnosti oz. aktivnosti.

Stanje sistema

- Spremenljivke stanj so množica med seboj neodvisnih vrednosti potrebnih za opis dinamike sistema
- Stanje sistema je vrednost spremenljivk stanj v določenem času

Dogodek

- Dogodek je sprememba stanja sistema
- Pri diskretnem sistemu, opisanem z diskretnimi dogodki, se spremenljivke stanj spremenijo v trenutku v časovno ločenih točkah
- Npr. "banka" – število strank v banki se spremeni le ob vstopu ali izstopu nove stranke

Zvezni sistem

- Pri zveznem sistem se spremenljivke stanj spreminjajo zvezno glede na čas
- Npr. pozicija letala predstavlja zvezni sistem; spremenljivke stanj pozicije in hitrosti se spreminjajo zvezno
- Le malo sistemov iz prakse je v celoti zveznih ali v celoti diskretnih
- Običajno je sisteme moč klasificirati glede na dominantni karakter sistema (zvezni ali dogodkovni)

Simulacija

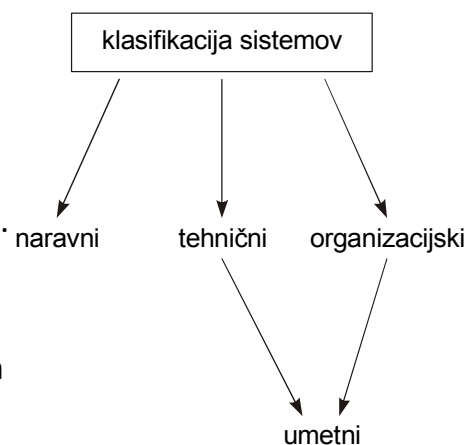
- Simulacija je proces razvoja računalniškega modela obravnavanega sistema in izvedba eksperimentov s pomočjo modela z namenom razumevanja delovanja sistema ali evalvacije različnih strategij upravljanja oz. kontrolnih strategij, ki jih izvedemo na sistemu.

Simulacija (nad.)

- Simulacija je dinamična ponazoritev odziva sistema z namenom:
 - Opisa sistema (delni ali v celoti)
 - Razlage odziva sistema v preteklosti
 - Predvidevanja odziva sistema
 - Razumevanja zakonitosti sistema

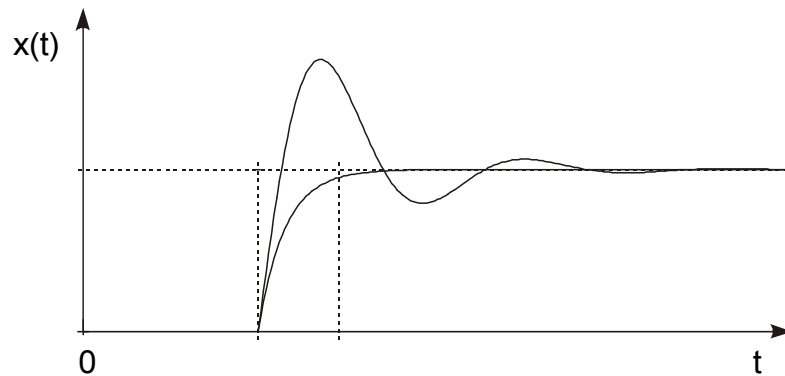
Klasifikacija sistemov

- **Stanje sistema** je vrednost spremenljivk stanja ob določenem času. Stanje sistema opišemo z elementi sistema; njihovimi lastnostmi in aktivnostmi.
- **Proces** je sprememba stanja sistema pod vplivom vhodnih spremenljivk ali notranjih dogodkov v sistemu



Odziv sistema

- Odziv (reakcija) sistema na vhodne signale (dražljaje) je opredeljen z dinamiko sistema



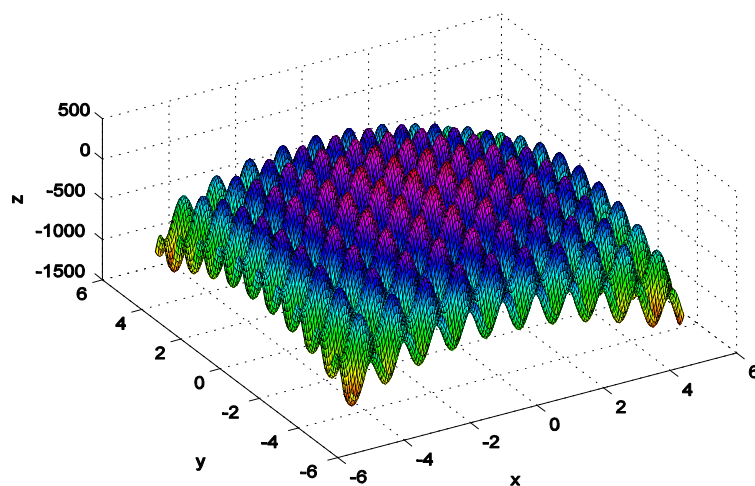
Model

- Model sistema je poenostavljena ponazoritev (koncept) realnega sistema
- Model je približek (abstrakcija) realnosti
- Pri modelu izločimo lastnosti realnosti ki nam v danem trenutku ne koristijo in se osredotočimo na lastnosti, ki so za nas pomembne

Optimiranje

- Izbor najboljše rešitve iz nabora možnih rešitev
- Preprost primer – minimum, maksimum
- Uporaba simulacijskega modela – analiza kaj-če
- Določitev kriterijske funkcije
- Postopek določitve simulacijskih scenarijev
- Izbor najboljše rešitve

Primer funkcije



$$z = 10 - 10 * 2 * ((x + y)^2 - 10 * \cos(2 * \pi * (x + y))) + ((x - y)^2 - 10 * \cos(2 * \pi * (x - y)))$$

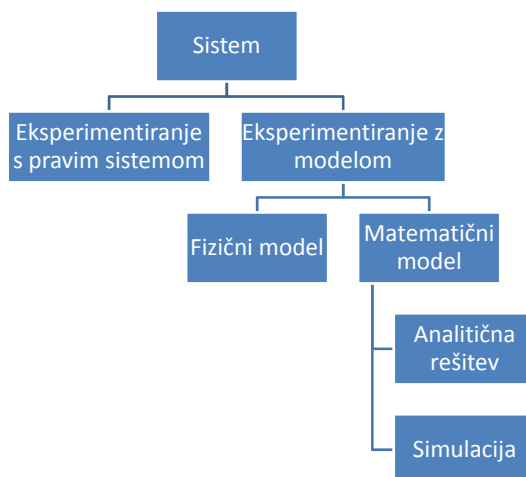
Uporaba simulacije

- V enostavnih primerih je moč razviti matematični model in pridobiti analitično rešitev
- Večina problemov iz realnega sveta je preveč kompleksna, da bi bilo možno pridobiti analitične rešitve
- V primeru kompleksnosti obravnavanih sistemov uporabimo postopke simulacije
- Izvedemo evalvacijo modela s pomočjo numeričnih metod

Uporaba – primer

- Proizvodno podjetje želi povečati proizvodne kapacitete
- Negotovost glede povečanja učinkovitosti in stroškov izgradnje
- Možnost (postaviti; če ni v redu odstraniti?)
- Dobro zasnovana simulacijska študija lahko osvetli vprašanja, ki se zastavljajo ob strateških akcijah
- Primerjava obstoječega stanja in novega, predvidenega stanja
- Podrobna analiza produkcijskih in stroškovnih učinkov

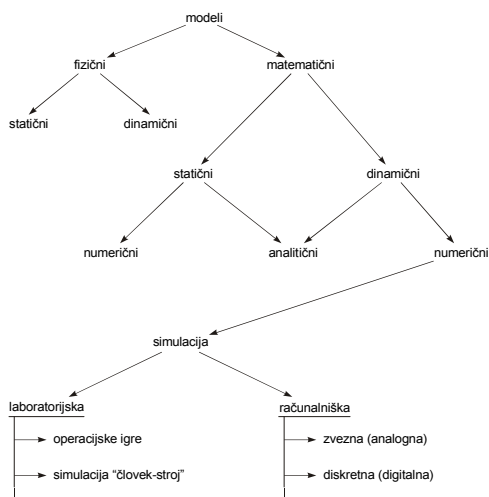
Načini analize in obravnavanja sistemov



Področja uporabe

- Ocena učinkovitosti in zasnova storitvenih organizacij kot npr. bolnišnice, krizni centri, pošte itd.
- Upravljanje s človeškimi viri s pomočjo simulacijskih modelov
- Oblikovanje in analiza proizvodnih sistemov
- Ocenjevanje logističnih sistemov
- Opredeljevanje zahtev po strojni opremi ali protokolih ter komunikacijskih mrežah
- Razvoj in upravljanje transportnih sistemov kot npr. letališč, avtocest, pristanišč in železnic
- Reorganizacija poslovnih in proizvodnih procesov
- Določanje optimalnih dobavnih politik ter kontrola skladišč
- Analiza finančnih in ekonomskih sistemov
- Opredelitev učinkovitosti vojaških organizacij in sistemov oborožitve

Klasifikacija modelov

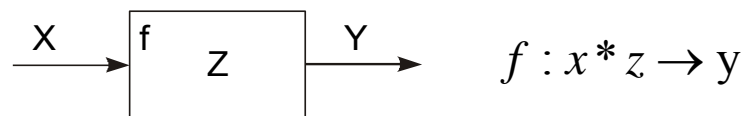


Primeri

SISTEM	ELEMENTI	LASTNOST	AKTIVNOSTI
javni promet	vozila	hitrost, razdalja	vožnja, zaviranje
veleblagovnica	kupci	nakup	plačevanje
banka	poslovni partnerji	stanje	polog, dvig
komunikacijski sistem	sporočila	dolžina, prioriteta	prenos

Opis sistema

- Na splošno opisujejo sistem:
 - vhodne spremenljivke $X = \{X_i\}; i = 1, 2, 3, \dots, m$
 - stanja sistema $Z = \{Z_j\}; j = 1, 2, 3, \dots, n$
 - izhodi iz sistema $Y = \{Y_r\}; r = 1, 2, 3, \dots, l$

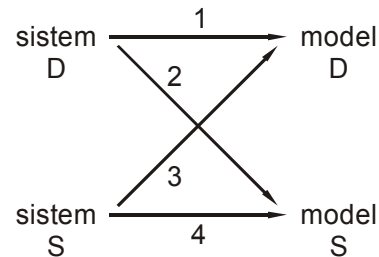


Potek modeliranja pri dogodkovni simulaciji

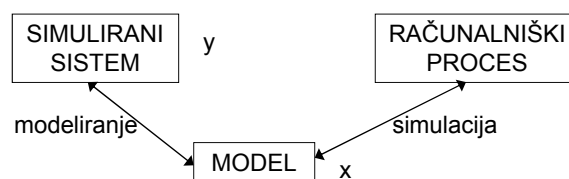
- Definicija problema
- Določitev ciljev
- Osnutek študije
- Formiranje matematičnega modela
- Zapis računalniškega programa
- Validacija modela
- Priprava eksperimenta (simulacijskih scenarijev)
- Simulacija in analiza rezultatov

Relacije med sistemom in modelom

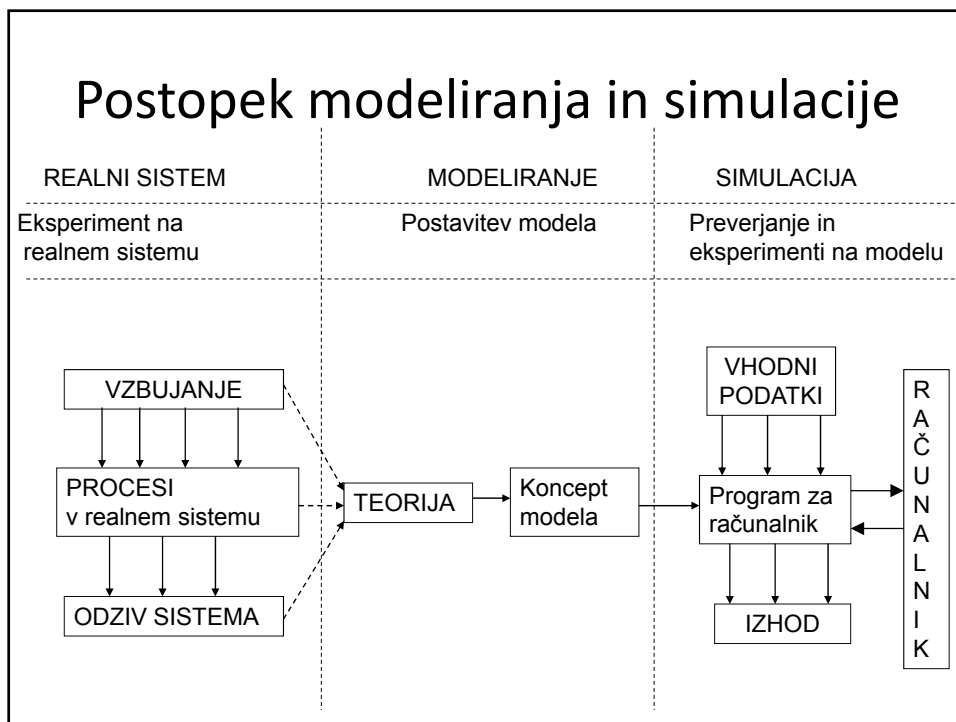
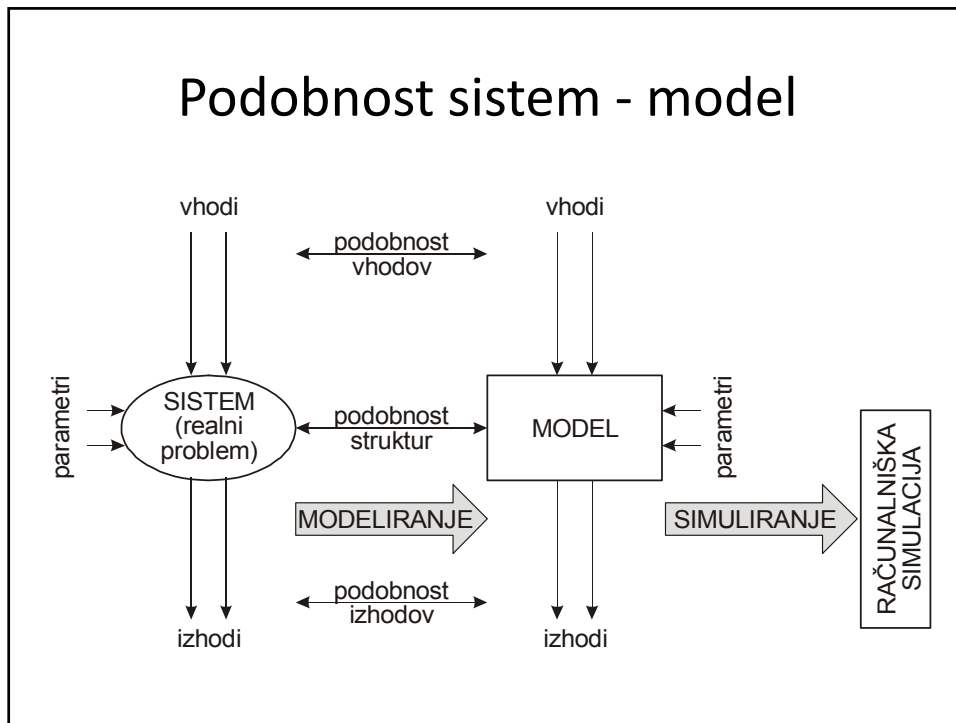
1. Sistem determinističen model determinističen. Preprosti mehanski modeli. $\ddot{x} + B\dot{x} + Cx = U(t)$
2. Sistem determinističen model stohastičen. Način poenostavitve zapletenih funkcij (metoda Monte Carlo).
3. Sistem stohastičen model determinističen. Kongruenčni generatorji naključnih števil.
4. Sistem stohastičen, model stohastičen. Kompleksni organizacijski sistemi – reševanje s pomočjo systemske simulacije.



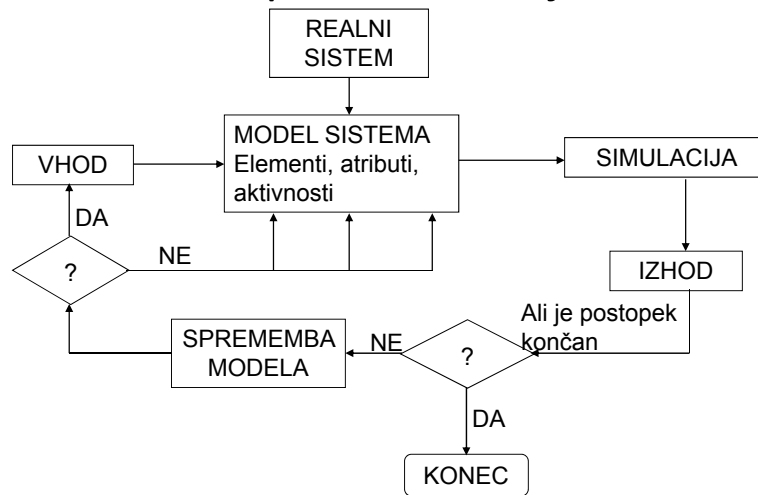
Relacija med modelom in simulacijo



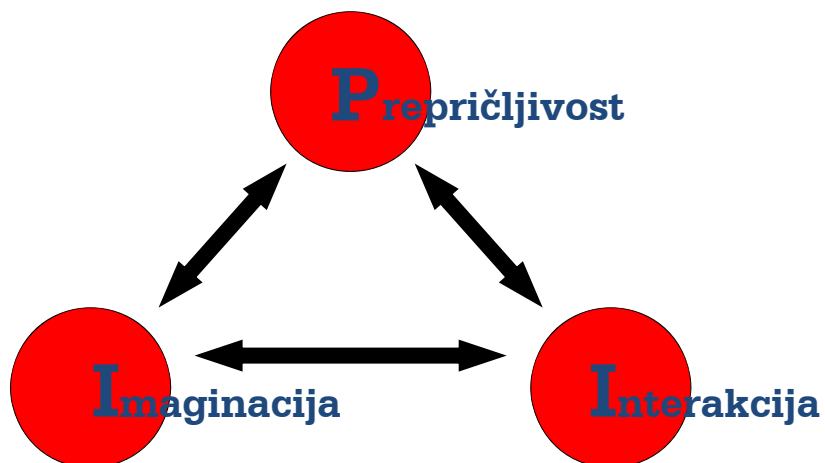
- Modeliranje predstavlja relacijo med simuliranim sistemom in modelom, simulacija pa relacijo med modelom in računalniškim procesom.
- Za x rečemo, da simulira y , če a.) x in y sta sistema b.) y predstavlja sistem katerega simuliramo, c.) x predstavlja poenostavitev simuliranega sistema (model) in d.) veljavnost x v odnosu na y ni nujno popolna.
- Proces generiranja obnašanja modela imenujemo simulacija.
- Simulacija je eksperimentiranje na modelu.

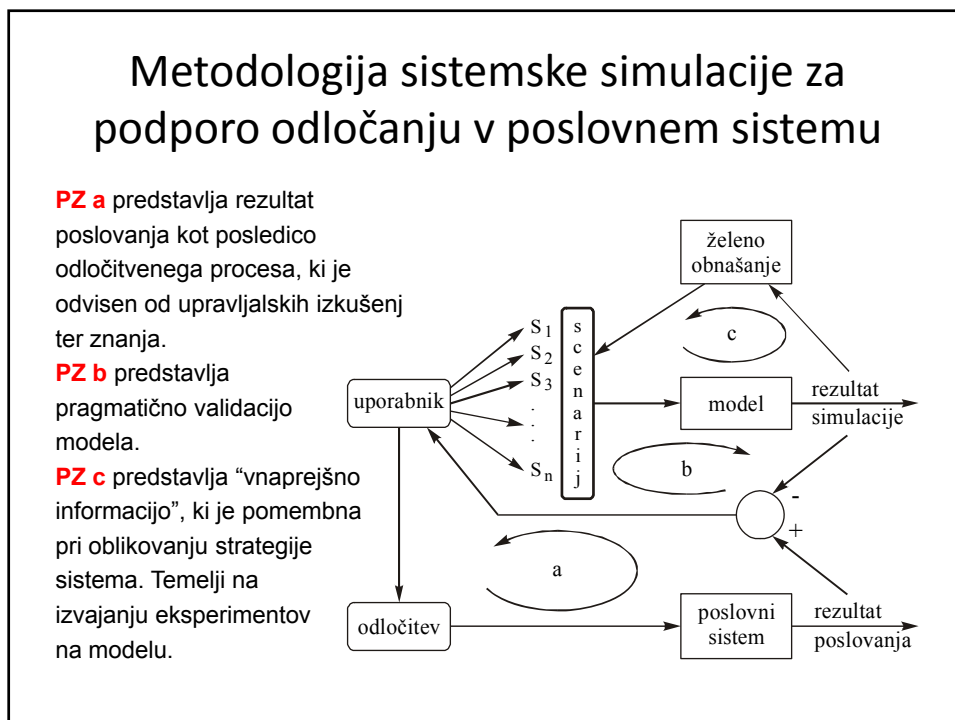
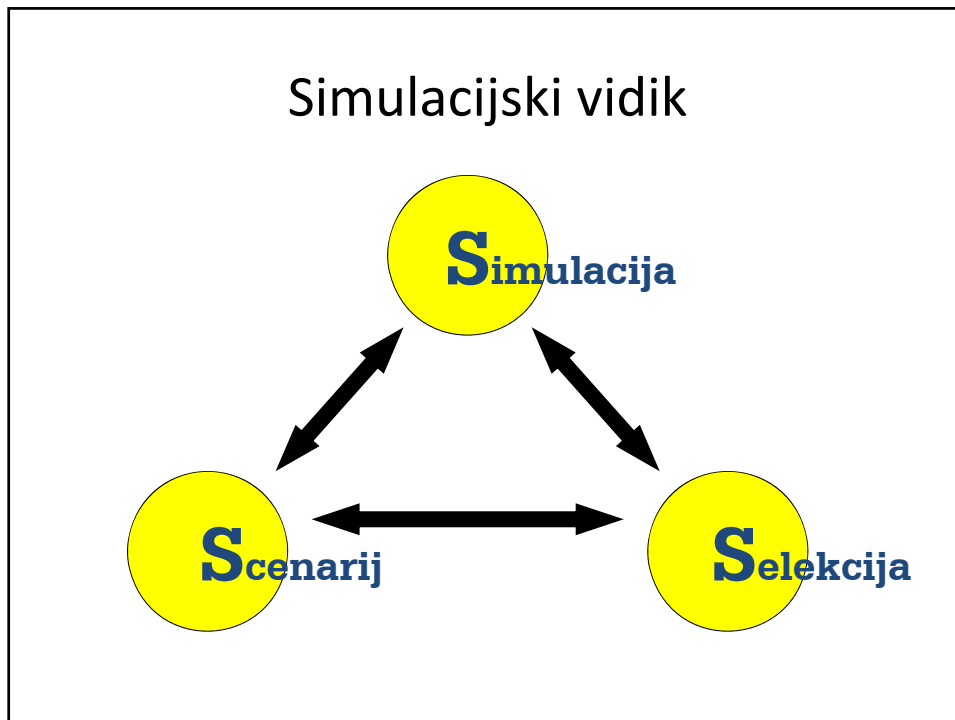


Postopek računalniškega simuliranja in eksperimentiranja

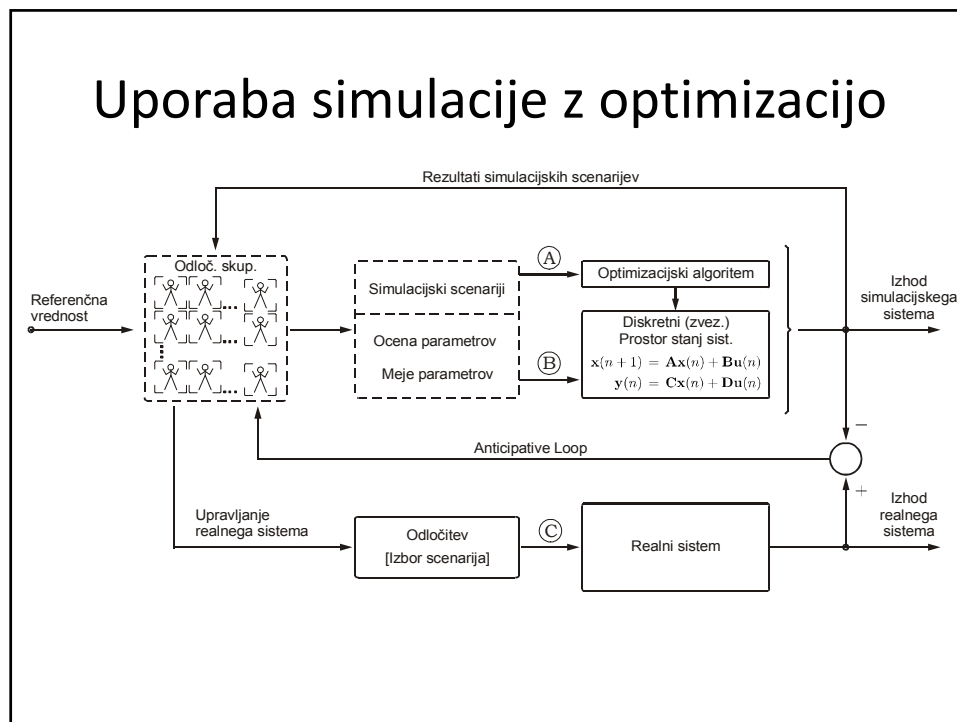


Vidik prepričljivosti "vživeti se v model"





Uporaba simulacije z optimizacijo



Simulacijski scenarij

- Množica scenarijev (struktura)

$$SC = \{SC_1, SC_2, \dots, SC_z\}$$

- Podan kot nabor vrednosti parametrov

$$SC_i = \{u_1, u_2, \dots, u_n\} \quad u_n \in \mathfrak{R}$$

- Rezultati scenarija – nabor vrednosti izhodnih spremenljivk

$$Y = \{y_1, y_2, \dots, y_m\} \quad y_m \in \mathfrak{R}$$

Dogodkovna simulacijska orodja

Operativne zahteve, ki jih mora izpolnjevati simulacijsko orodje:

- generiranje naključnih števil za predstavitev stohastičnih procesov
- generiranje poljubne porazdelitve (analitične, uporabniško podane)
- objektno funkcioniranje
- statistična obdelava rezultatov simulacije, ki omogoča primerjavo, validacijo in izboljšavo sistema
- kontrola simulacijskega časa

Dodatne zahteve

- vizualno modeliranje
- interaktivnost:
 - interakcija z uporabnikom
 - prikaza in obdelave rezultatov
- uporaba grafičnih objektov
- sodoben uporabniško prijazen vmesnik
- vgrajeni simulacijski koncepti, npr. logika razvrščanja, vgrajene funkcije ipd.

Dodatne zahteve (nad.)

- hiter razvoj modelov
- upoštevanje dimenzij objektov 1:1
- združevanje opravil in sintakse, ki se ponavlja

Dodatne zahteve (nad.)

- procesni koncept
- definicija objektov kot C objekta oz. podatkovnega objekta
- definicija in delo nad množico objektov
- paralelno procesiranje
- razumljiva sintaksa s podporo dokumentaciji
- generiranje poročil, specifikacija uporabnika

Zgodovina

- Razvoj sistema GPSS datira v leto 1960 (Gordonov simulator)
- IBM
- Razširjenost sistema z IBM računalniki
- Enostavnost uporabe tudi za netehnične študije
- Primeren za simulacijo strežnih sistemov (sistemi z vrstami)

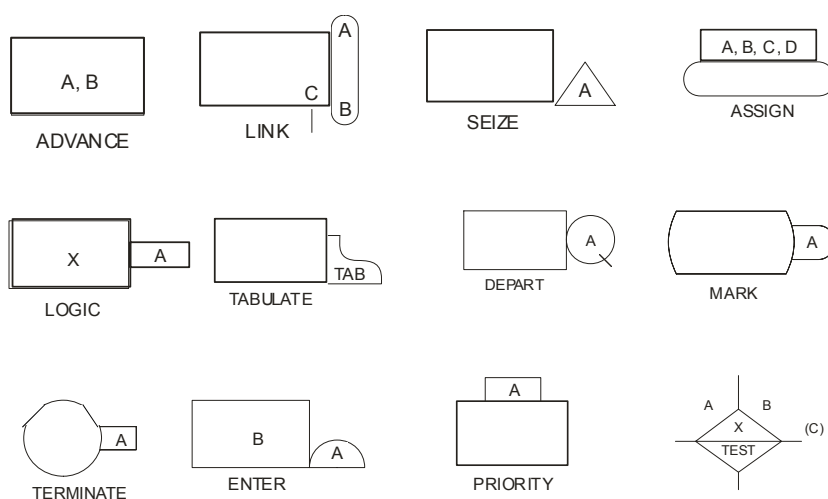
Zgodovina (nad.)

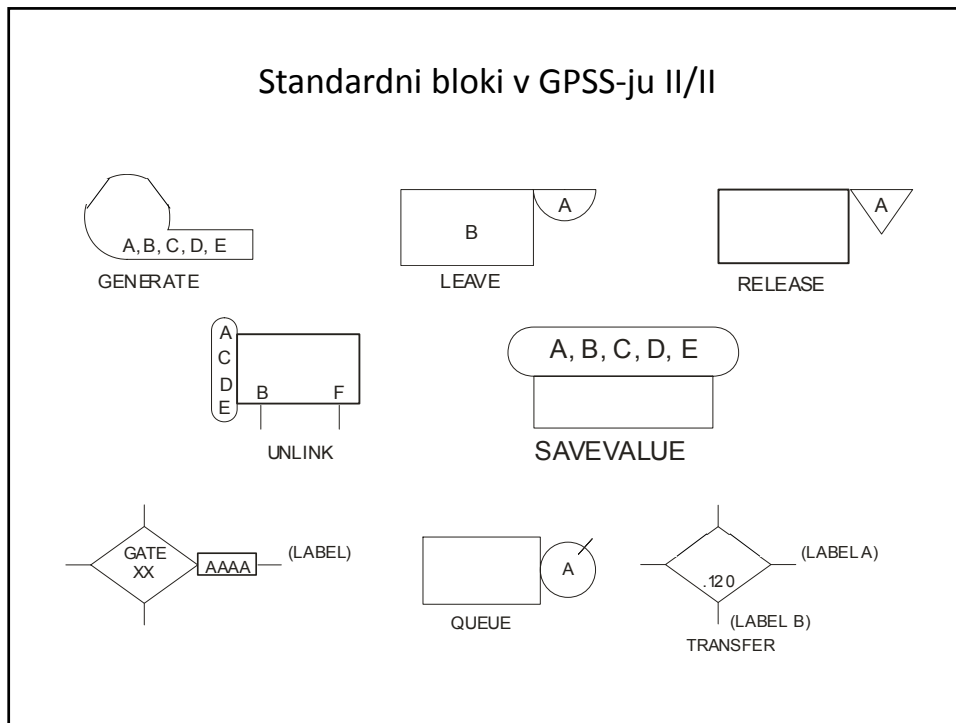
- Problemska domena orodja ~ komunikacijski sistemi
- Potreba po grafičnem izhodu, ki bi služil interaktivni validaciji je bila izražena že v letu 1965
- Priklop prikazovalnika (terminala), ki je bil interaktivno povezan z GPSS (1965) ~ Boeing

Zgodovina (nad.)

- Več verzij orodja GPSS
- 1977 – James O. Henriksen
- Verzija GPSS/H (H ~ Henriksen)
- Wolverine Software Corporation
- Prehod iz platforme IBM na platformo PC
- Najbolj razširjena verzija orodja

Standardni bloki v GPSS-ju I/II





Tip blokov v GPSS-ju

Operation	A	B	C	D	E	F
ADVANCE	Mean	Modifier				
ASSIGN	Param. No. (+)	Source				
DEPART	Queue No.	(Units)				
ENTER	Storage No.	(Units)				
GATE	Item No.	(Next block B)				
GENERATE	Mean	Modifier	(Offset)	(Count)	(Priority)	(Params.)
LEAVE	Storage No.	(Units)				
LINK	Chain No.	Order	(Next block B)			
LOGIC	$\left. \begin{matrix} R \\ S \\ I \end{matrix} \right\}$	Switch No.				
MARK	(Param. No.)					
PRIORITY	Priority					
QUEUE	Queue No.	(Units)				
RELEASE	Facility No.					
SAVEVALUE	S.V. No. (+)	SNA				
SEIZE	Facility No.					
TABULATE	Table No.	(Units)				
TERMINATE	(Units)					
TEST	Arg. 1	Arg. 2	(Next block B)			
TRANSFER	Select. Factor	Next block A	Next block B			
UNLINK	Chain No.	Next block A	Counter	(param. No.)	(Arg.)	(Next block B)

Standardni numerični atributi SNA

C1- trenutna vrednost urnega časa

M1- prehodni čas transakcije

CH n – štev. trans. in verigi n

F n - trenutno stanje *Facility* n

FN n - vrednost funkcije n

K n - *integer* n

N n - celotno št. tran., ki so vstopile v blok n

P n - param. št. n transakcije

Q n - dolžina repa n

Lista SNA v GPSS II/II

R n - prostor, ki je v *STORAGE* n

RN n - naključni generator $n=1,2,\dots$

S n - tekoča zasedenost *storage* n

V n - vrednost sprem. št. n

W n - štev. trans. v bloku n (tekoča)

X n - shranjena vrednost na lokaciji n

SNA primer

Tip SNA označimo z eno ali dvema črkama in s številom.

PR: TABLE M1

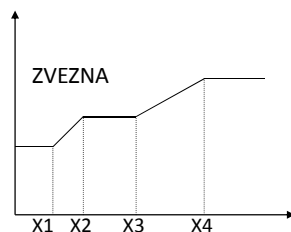
Poljuben SNA uporabljamo s kratico TABULATE za zbiranje statistike.

Aritmetične operacije +, -, /, *, @, ...

PRIMER:

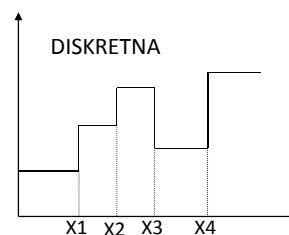
5 VARIABLE S6+5*(Q12+Q17)

Funkcije



n FUNCTION RN1, C12

X1,Y1/X2,Y2/X3,Y3/... X12,Y12



m FUNCTION RN7,D5

X1,Y1/X2,Y2/... X5,Y5

Podatkovni zapis v stolpcih 1 do 71.

Prenosni (Transfer) način v GPSS

NAČIN	Polje A	Polje B	Polje C
nepogojni	,	nasl. blok MN	
naključni	.xy	nasl. blok MN	nasl. blok BM
pogojni	BOTH	nasl. blok. A	nasl. bl. A
parametrski	P	param. št	
funkcijski	FN	funk. št.	param. št.

PRIMER: ASSIGN 1,FN3

TRANSFER P,1

GPSS/H lokacija, delovanje

- Lokacija:
`ftp://kib1.fov.uni-mb.si/GPSSH/`
- GPSS/H je splošen simulacijski sistem, ki teče v okolju MS-DOS.
- Delovanje:
simulacijski model, ki je zapisan v ASCII datoteki s končnico .GPS (npr.: Model1.gps) izvede, rezultate pa zapiše v datoteko s končnico .LIS (npr.: Model1.lis).



Univerza v Mariboru
Fakulteta za organizacijske vede



Laboratorij za kibernetiko in sisteme
za podporo odločanju

GPSS/H, namestitev, zagon

- Po prenosu datoteke gpssh.zip preko omrežja uporabite program za stiskanje podatkov WinZip.
- Vse prenešene datoteke kopiramo v isto mapo.
- Program poženete z ukazom gpssh (v DOS-ovski vrstici oz. Command prompt - u, v mapi, kjer so GPSS/H datoteke).
- Kot test vpišite v dos-ovski vrstici: "gpssh Model1" (brez narekovajev), ki vam bo izpisala rezultate v datoteko Model1.lis

GPSS/H - delovanje

1. Model (*.gps)

```

Model1.gps - Notepad
File Edit Search Help
SIMULATE
*
* ONE-LINE, SINGLE-SERVER QUEUEING MODEL
*
* GENERATE      18,6  ARRIVALS EVERY 18 +- 6 MINUTES
* ADVANCE      0.5   HANG UP COAT
* SEIZE        JOE   CAPTURE THE BARBER
* ADVANCE     15,3  HAIRCUT TAKES 15 +- 3 MINUTES
* RELEASE     JOE   FREE THE BARBER
* TERMINATE   1     EXIT THE SHOP
*
* START       100
* END
  
```

2. Simulacijski tek

3. Rezultati (*.lis)

```

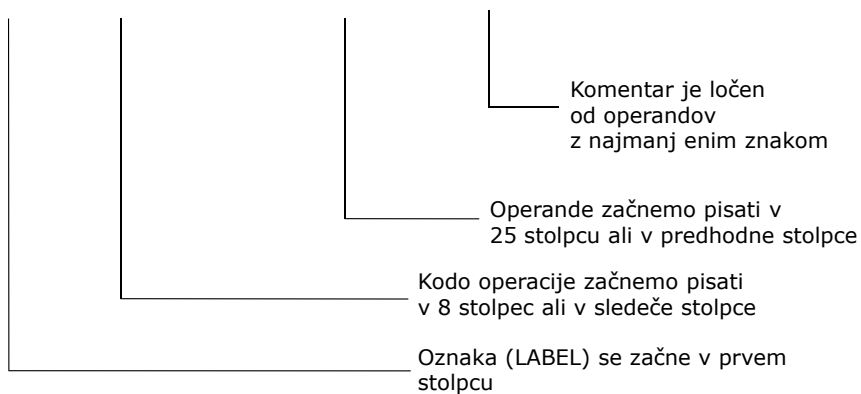
Model1.lis - Notepad
File Edit Search Help
STUDENT GPSS/H RELEASE 3.0c-C10 (EP195) 22 Jan 2010 11
LINE# STMT# IF DO BLOCK# *LOC OPERATION A,B,C,D,
1 1 SIMULATE
2 2 *
3 3 * ONE-LINE, SINGLE-SERVER
4 4 *
5 5 1 GENERATE 18,6
6 6 2 ADVANCE 0.5
7 7 3 SEIZE JOE
8 8 4 ADVANCE 15,3
9 9 5 RELEASE JOE
  
```

GPSS/H, zapis modela

- Model napišemo v poljubnem urejevalniku teksta (npr. Notepad ali Beležnica).
- Bloke in kontrolne stavke zapišemo z velikimi črkami.
- Datoteko z modelom shranimo s končnico .GPS, pri tem mora biti dolžina imena 8.3 (pri Beležnici shranimo datoteko tako, da ime napišemo med dvema narekovajema, npr: File/Save As/File Name: "Model1.gps")

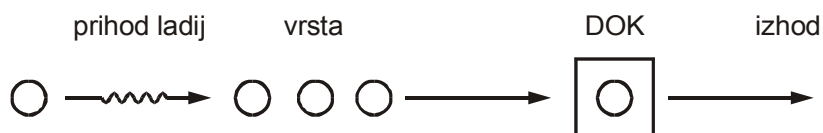
GPSS/H, vhodni format

```
12345678901234567890123456789012345678901234567890123
OZNAKA OPERACIJA          OPERANDI  KOMENTAR
PRIPR  ADVANCE            3.75,1   Pripravimo sveženj
```



Primer – pristanišče - Luka

- Opis sistema: tovorne ladje prispejo na razkladanje v pristanišče vsakih 16 ± 5 ur. Pristanišče ima en razkladalni prostor (dok).
V primeru, da je dok prost, ladja odpluje na razkladanje, v nasprotnem primeru pa čaka v sidrišču. Čas razkladanja : 15 ± 10 ur.
- Situacijska shema:



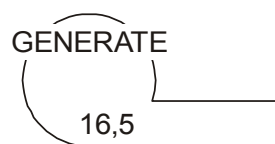
Luka – Problem, naloga in kriteriji

- Problem: Zaradi dolgega časa čakanja na razkladanje in prevelikega števila ladij na sidrišču smo opazili, da gre določeno število ladij k konkurenci.
- Naloga: določi čas čakanja ladij v pristanišču, čas čakanja v sidrišču in število ladij, ki čakajo na razkladanje. Razvijte sistem, ki bo imel krajše čase čakanja.
- Kriteriji:
 - maksimalno število ladij v sidrišču: $5 + 10\%$
 - maksimalen čas čakanja: $6 \text{ dni} + 10\%$

Pregled uporabljenih blokov

- **GENERATE**
- **QUEUE**
- **SEIZE**
- **DEPART**
- **ADVANCE**
- **RELEASE**
- **TERMINATE**

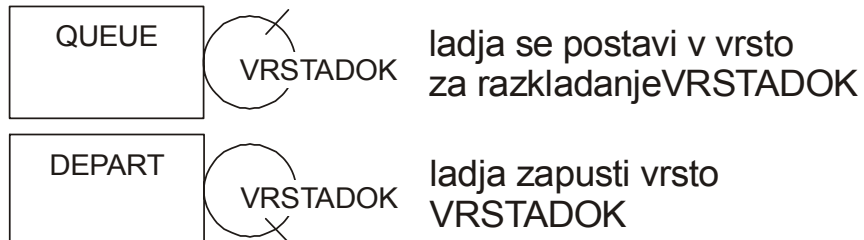
Blok GENERATE



prihodi ladij v sidrišče

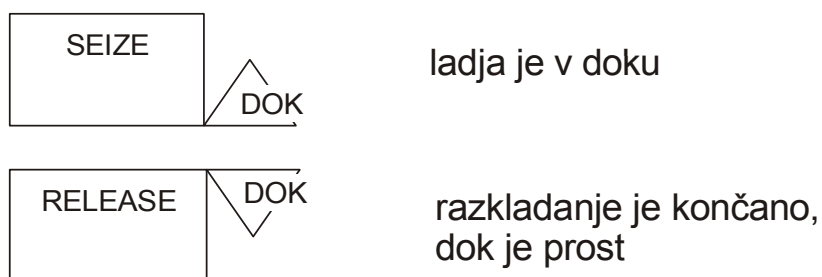
Operand	Pomen	Prednastavljena vrednost ali rezultat
A	Povprečen čas med prihodi	0.0 (nič)
B	Polovico obsega naključne spremenljivke podane z enakomerno porazdelitvijo časa med prihodi	0.0 (nič)
C	Offset interval (čas prihoda prve transakcije)	Brez prednastavljenega offset-a
D	Limitni števec (maksimalno število prihodov)	Brez prednastavljenega limitnega števca
E	Prioriteta transakcije	0 (nič)

Bloka QUEUE in DEPART



Operand	pomen	Prednastavljena vrednost ali rezultat
A	Ime vrste v katero transakcija na bloku <code>QUEUE</code> ali izstopa na bloku <code>DEPART</code>	Napaka (error) pri prevajanju

Bloka SEIZE in RELEASE



Operand	Pomen	Prednastavljena vrednost ali rezultat
A	Ime strežnega mesta tipa <code>FACILITY</code> ki ga zasedemo na bloku <code>SEIZE</code> ter sprostimo na bloku <code>RELEASE</code>	Prednastavljena vrednost ni podana

Blok ADVANCE

ADVANCE

15,10

razkladanje ladje

Operand	Pomen	Prednastavljena vrednost ali rezultat
A	Povprečen čas zadrževanja transakcije	0.0 (nič)
B	Polovico obsega naključne spremenljivke podane z enakomerno porazdelitvijo časa zadrževanja	0.0 (nič)

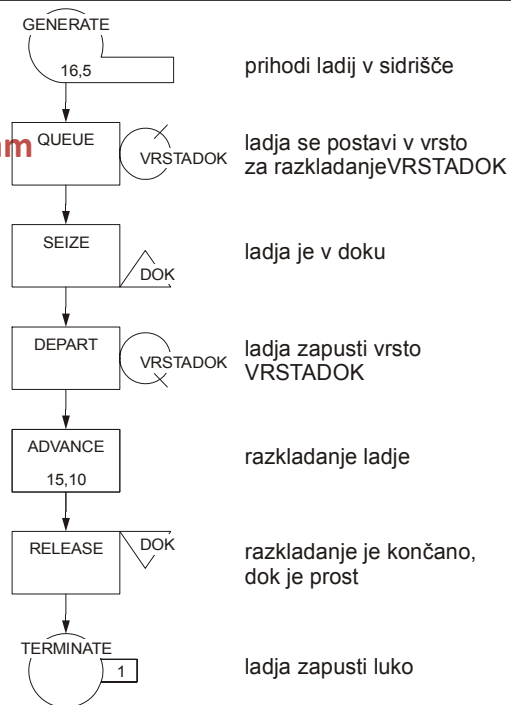
Blok TERMINATE

Operand	Pomen	Prednastavljena vrednost ali rezultat
A	Odštevanje števca terminacij (TC – Termination Counter) v modelu	0 (nič)



ladja zapusti luko

Luka – blokovni diagram



GPSS/H Kontrolni stavki

- **SIMULATE**
- **START**
- **END**



Luka - model

```

SIMULATE
*
*   Simulacija luke
*
*
*   GPSS/H blokovni del
*
*
*   GENERATE   16,5   Prihod ladij v luko 15 +- 5 ur
*   QUEUE     VRSTADOK   Ladja se postavi v vrsto
*   SEIZE     DOK       Ladja je v doku
*   DEPART    VRSTADOK   Ladja zapusti vrsto
*   ADVANCE   15,10    Čas razkladanja
*   RELEASE   DOK       Ladja zapusti dok - dok je prost
*   TERMINATE 1        Ladja zapusti luko
*
*   GPSS/H kontrolni stavki
*
*   START     100      TC (Termination counter) postavimo na 100
*   END

```

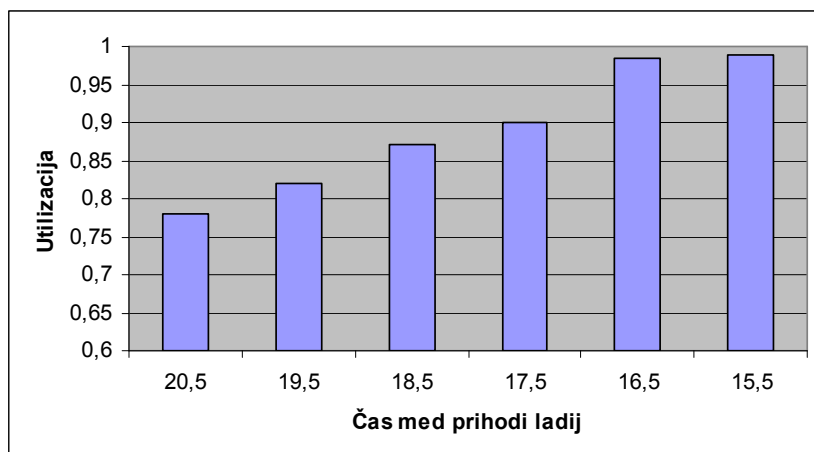


Sprememba parametrov na bloku

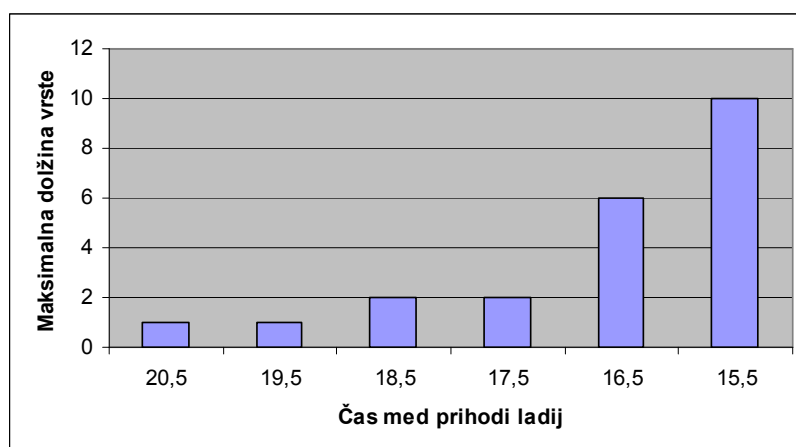
GENERATE

Prihodi	Utilizacija	Max Q	AVG time Q
20,5	0,779	1	1,353
19,5	0,82	1	2,095
18,5	0,872	2	3,809
17,5	0,901	2	4,504
16,5	0,984	6	34,567
15,5	0,988	10	67,327

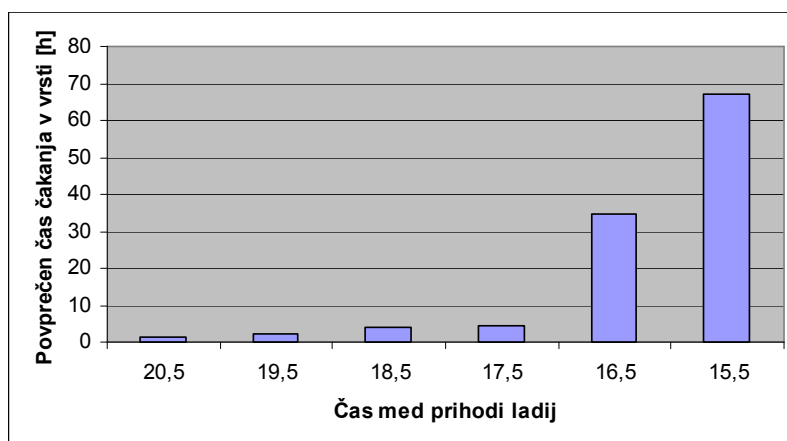
Zasedenost strežnega mesta



Vrsta



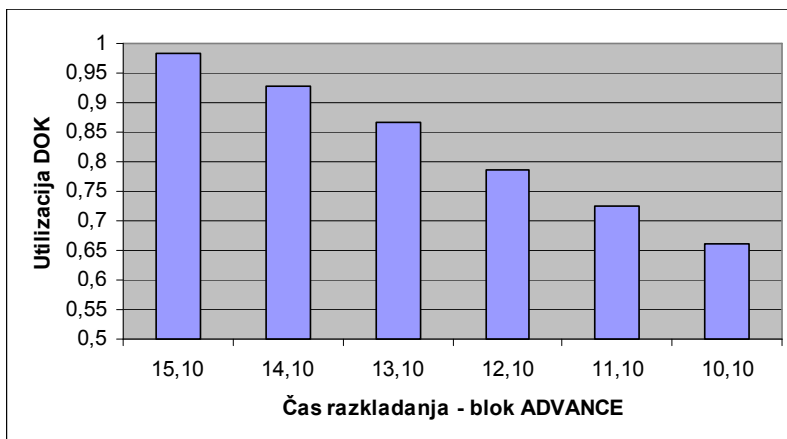
Čas čakanja



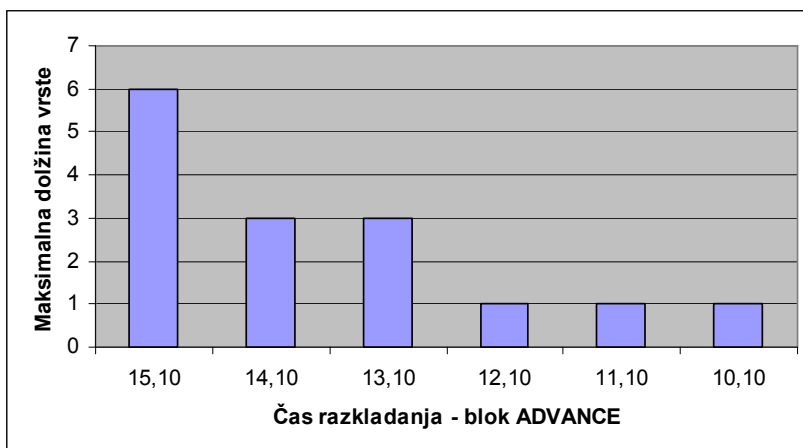
Sprememba parametrov na bloku ADVANCE

Čas razk.	Utilizacija	Max Q	AVG time Q
15,10	0,984	6	34,567
14,10	0,929	3	10,812
13,10	0,866	3	4,76
12,10	0,787	1	2,095
11,10	0,725	1	1,353
10,10	0,662	1	0,91

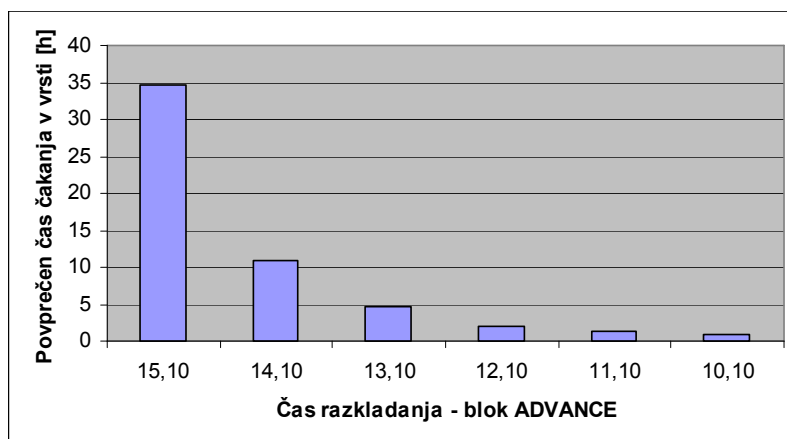
Zasedenost strežnega mesta



Vrsta



Čas čakanja



Kontrola izvajanja simulacijskega teka

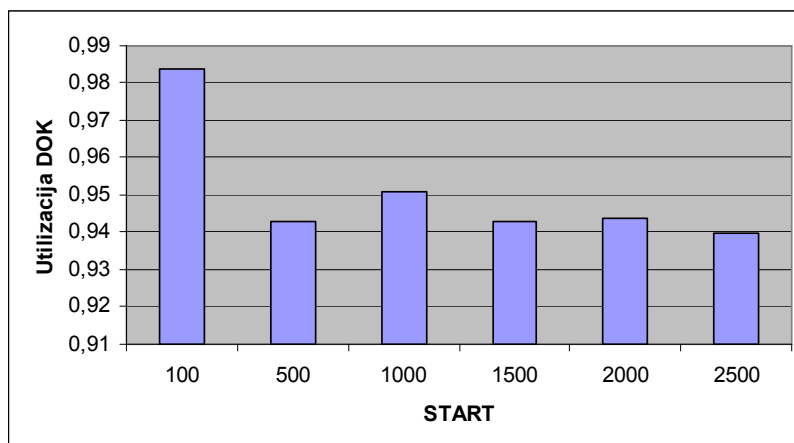
- Na bloku **TERMINATE** s parametrom **A** transakcija zapusti model
- Parameter **A** je celo število in določa vrednost, ki jo odštejemo števcu terminacij (TC – Termination Counter)
- Vrednost števca TC določimo s kontrolnim stavkom **START** (npr. **START 100**)
- Ko je vrednost števca TC enaka 0 (ali manjša) se simulacijski tek ustavi



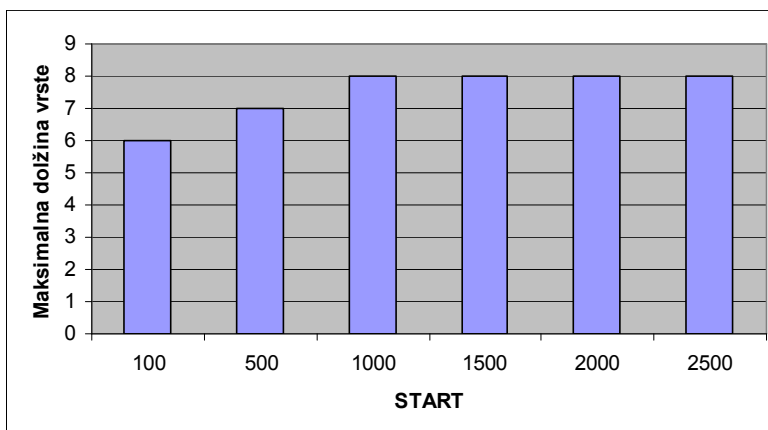
Sprememba parametrov na kontrolnem stavku START

START	Utilizacija	Max Q	AVG time	Clock
100	0,984	6	34,567	1617
500	0,943	7	23,141	8006
1000	0,951	8	26,536	16031
1500	0,943	8	22,428	24050
2000	0,944	8	20,958	32036
2500	0,94	8	18,889	40068

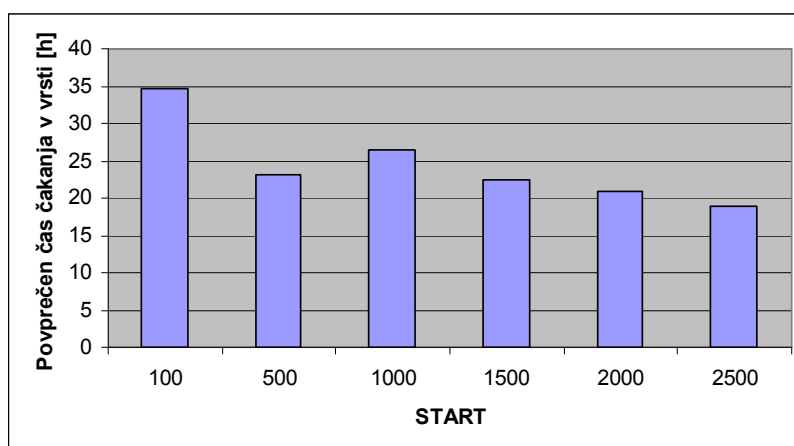
Zasedenost strežnega mesta



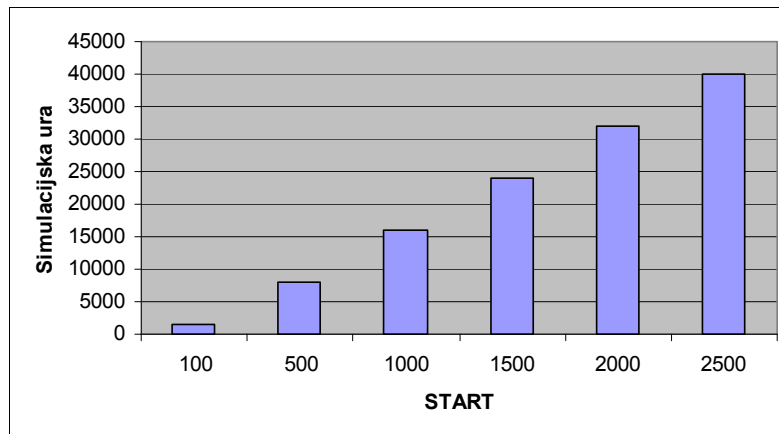
Vrsta



Čas čakanja

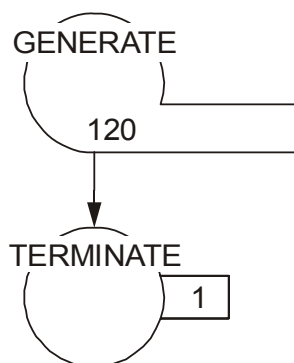


Čas izvajanja simulacijskega teka



Kontrola izvajanja simulacijskega teka glede na simulacijski čas

- Primer modela z dvema blokoma



kontrolna transakcija je uvedena ob času 120 časovnih enot

števec TC (termination counter) zmanjšamo za 1

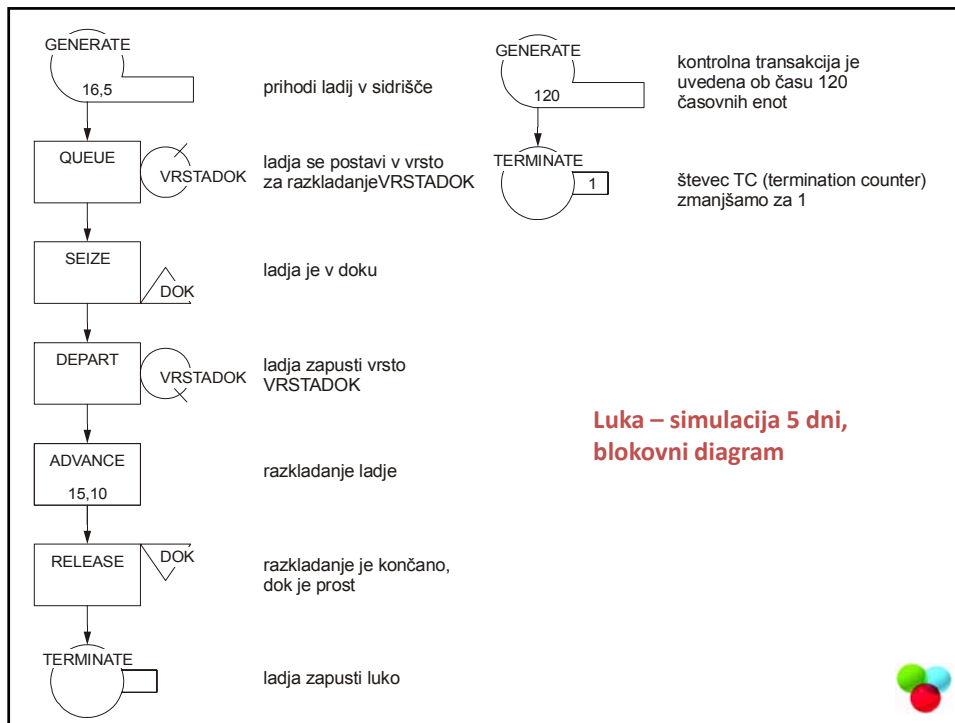


Model - ura

```
SIMULATE
*
*   Model z dvema blokoma
*
* GPSS/H blokovni del
*
*   GENERATE   120   Uvedba transakcije v model
*   TERMINATE  1     Transakcija zapusti model
*
*   GPSS/H kontrolni stavki
*
*   START      1     TC postavimo na 1
*   END
```

Luka – simulacija 5 dni

- Koliko ladij bo raztovorjenih v petih dnevih?
- Uvedemo kontrolo simulacijskega teka glede na simulacijski čas.



Luka – simulacija 5 dni

```

SIMULATE
*
*   Simulacija luke
*
* GPSS/H blokovni del
*
GENERATE 16,5      Prihod ladij v luko 15 +- 5 ur
QUEUE    VRSTADOK Ladja se postavi v vrsto
SEIZE    DOK       Ladja je v doku
DEPART   VRSTADOK Ladja zapusti vrsto
ADVANCE  15,10    Čas razkladanja
RELEASE  DOK       Ladja zapusti dok - dok je prost
TERMINATE        Ladja zapusti luko
*
*   Kontrola simulacijskega teka - ura
*
GENERATE 120      Uvedba transakcije v model
TERMINATE 1       Transakcija zapusti model
*
*   GPSS/H kontrolni stavki
*
START    1        TC (Termination counter) postavimo na 1
END
  
```

Kontrolni stavek CLEAR

- Pomoč pri izvajanju več zaporednih simulacijskih tekov.
- Odstrani vse transakcije iz modela.
- Absolutno in relativno uro postavi na nič.
- Naključni generator ni ponovno nastavljen, generiranje naključnih števil se nadaljuje iz predhodnega teka.



Luka, kontrolni stavek CLEAR

```

SIMULATE
*
*   Simulacija luke - uporaba kontrolnega stavka CLEAR
*
*
*   GPSS/H blokovni del
*
*
*   GENERATE   16,5   Prihod ladij v luko 15 +- 5 ur
*   QUEUE     VRSTADOK   Ladja se postavi v vrsto
*   SEIZE      DOK       Ladja je v doku
*   DEPART     VRSTADOK   Ladja zapusti vrsto
*   ADVANCE    15,10     Čas razkladanja
*   RELEASE    DOK       Ladja zapusti dok - dok je prost
*   TERMINATE  1         Ladja zapusti luko
*
*   GPSS/H kontrolni stavki
*
*   START      100      TC (Termination counter) postavimo na 100
*   CLEAR      CLEAR    CLEAR za drugo ponovitev
*   START      100      TC (Termination counter) postavimo na 12
*   CLEAR      CLEAR    CLEAR za tretjo ponovitev
*   START      100      TC (Termination counter) postavimo na 12
*   END

```

Luka, kontrolni stavek CLEAR, rezultati

Tek	Utilizacija	AVG time	Max Q	AVG time	Clock
1.	0,984	15,906	6	34,567	1617
2.	0,961	15,609	3	15,986	1625
3.	0,867	14,197	4	7,793	1637
Povprečje:	0,94	15,24	4,33	19,45	1626,33

Kontrolni stavek RESET

- Pomoč pri izvajanju statističnih eksperimentov v **GPSS/H**.
- Postavi vso statistiko in vrednost relativne ure na 0.
- Trenutnih transakcij ne odstrani iz modela.
- Vrednost absolutne ure se ne spremeni.
- Aktivira se relativna ura.
- Skupaj s stavkom START se uporablja za ugotavljanje delovanja modela v pogojih stabilnega delovanja



Luka, model, kontrolni stavek, RESET

```

SIMULATE
*
*   Simulacija luke - kontrolni stavek RESET
*
*   GPSS/H blokovni del
*
*   GENERATE 16,5      Prihod ladij v luko 15 +- 5 ur
*   QUEUE   VRSTADOK  Ladja se postavi v vrsto
*   SEIZE    DOK       Ladja je v doku
*   DEPART  VRSTADOK  Ladja zapusti vrsto
*   ADVANCE 15,10     Čas razkladanja
*   RELEASE DOK       Ladja zapusti dok - dok je prost
*   TERMINATE 1       Ladja zapusti luko
*
*   GPSS/H kontrolni stavki
*
*   START      1000    TC (Termination counter) postavimo na 1000
*   RESET      RESET  za drugo ponovitev
*   START      1000    TC (Termination counter) postavimo na 1000
*   RESET      RESET  za tretjo ponovitev
*   START      100     TC (Termination counter) postavimo na 12
*   END

```

Luka, kontrolni stavek RESET , rezultati

Tek	Utilizacija	AVG time	Max Q	AVG time	R. Clock	A. Clock
1.	0,951	15,249	8	26,536	16031	16031
2.	0,937	15,002	5	15,379	16005	32036
3.	0,916	14,484	3	8,55	1581	33617
Povprečje:	0,93	14,91	5,33	16,82		

- Uporaba parametra **NP** – (no print)

```

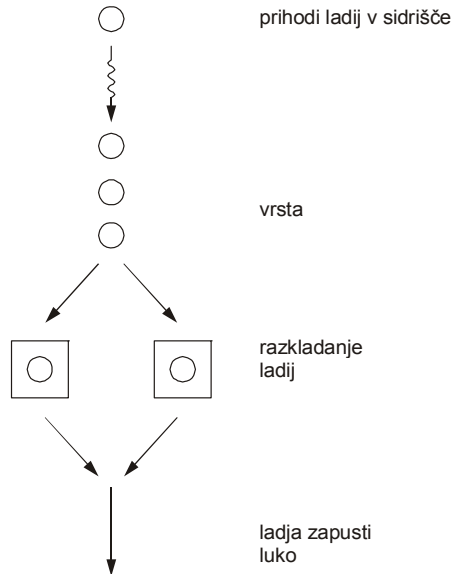
START      1000,NP    TC = 1000, NP - ne izpišemo rezultatov
RESET      RESET  za drugo ponovitev
START      1000,NP    TC = 1000, NP - ne izpišemo rezultatov
RESET      RESET  za tretjo ponovitev
START      100       TC = 100
END

```

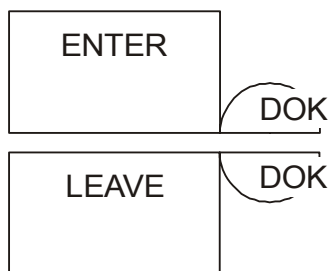


Luka – dva doka, situacijska shema

- V pristanišču odpremo še en dok z enako kapaciteto. Kakšna bo zasedenost doka in čas čakanja v vrsti?



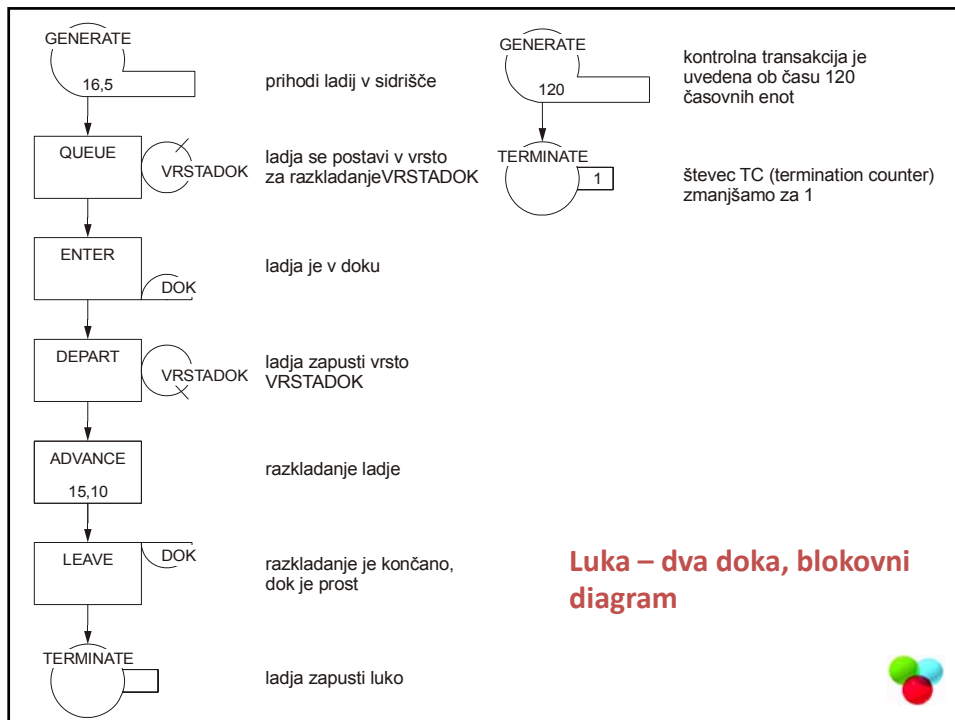
Luka – dva doka, bloka ENTER in LEAVE



ladja je v doku

razkladanje je končano, dok je prost

Operand	Pomen	Prednastavljena vrednost ali rezultat
A	Ime strežnega mesta tipa STORAGE (katerega eno ali več strežnih mest je zasedenih na bloku ENTER ter sproščenih na bloku LEAVE)	Napaka (error) pri prevajanju
B	Število strežnih mest, ki se zasedejo na bloku ENTER ter sprostijo na bloku LEAVE	1



Luka – dva doka, model

```

SIMULATE
*
*   Simulacija luke
*
*   Deklaracija mehanizma strežbe (STORAGE)
*
*   STORAGE      S(DOK),2
*
*   GPSS/H blokovni del
*
*   GENERATE     16,5      Prihod ladij v luko 15 +- 5 ur
*   QUEUE        VRSTADOK  Ladja se postavi v vrsto
*   ENTER        DOK       Ladja je v enem izmed dokov
*   DEPART       VRSTADOK  Ladja zapusti vrsto
*   ADVANCE      15,10     Čas razkladanja
*   LEAVE        DOK       Ladja zapusti dok - dok je prost
*   TERMINATE    LADJA     Ladja zapusti luko
*
*   Kontrola simulacijskega teka - ura
*
*   GENERATE     120      Uvedba transakcije v model
*   TERMINATE    1        Transakcija zapusti model
*
*   GPSS/H kontrolni stavki
*
*   START       1         TC (Termination counter) postavimo na 12
*   END

```

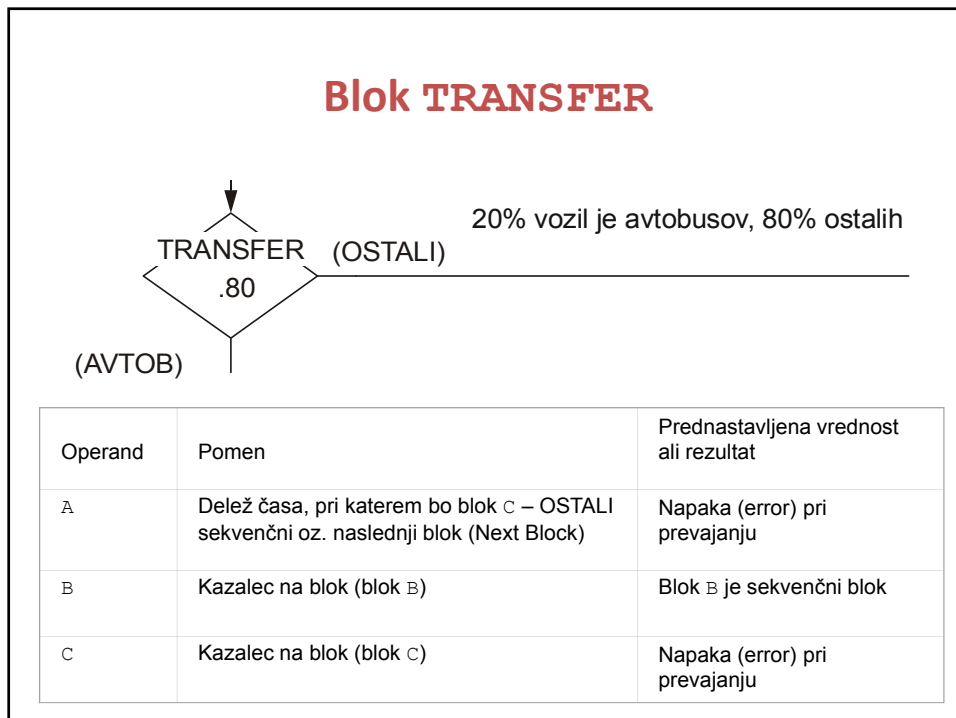
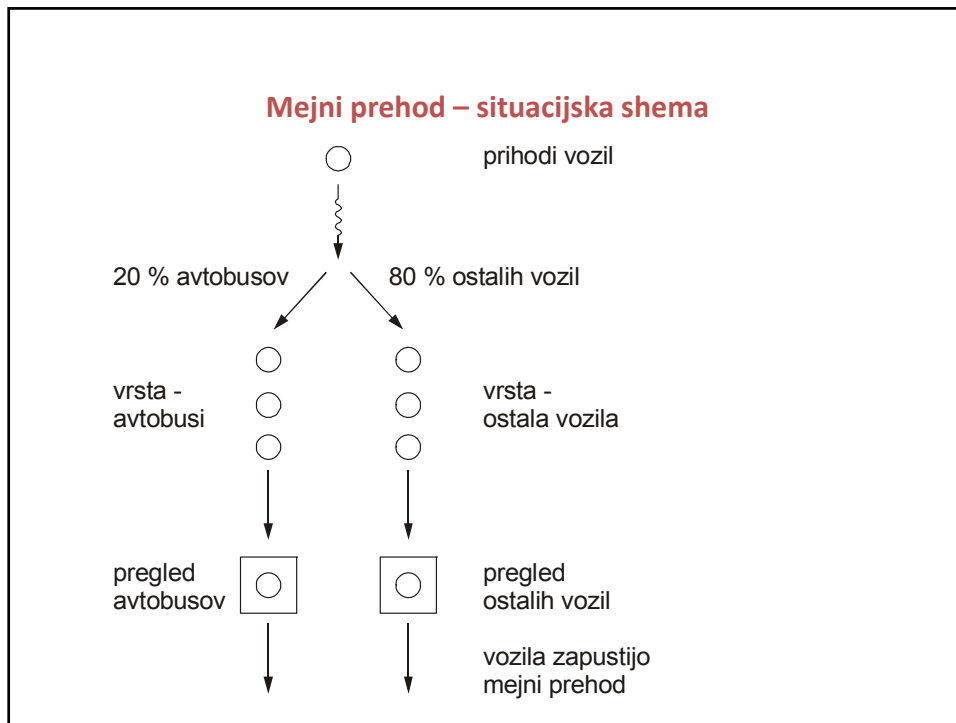
Luka – dva doka, rezultati

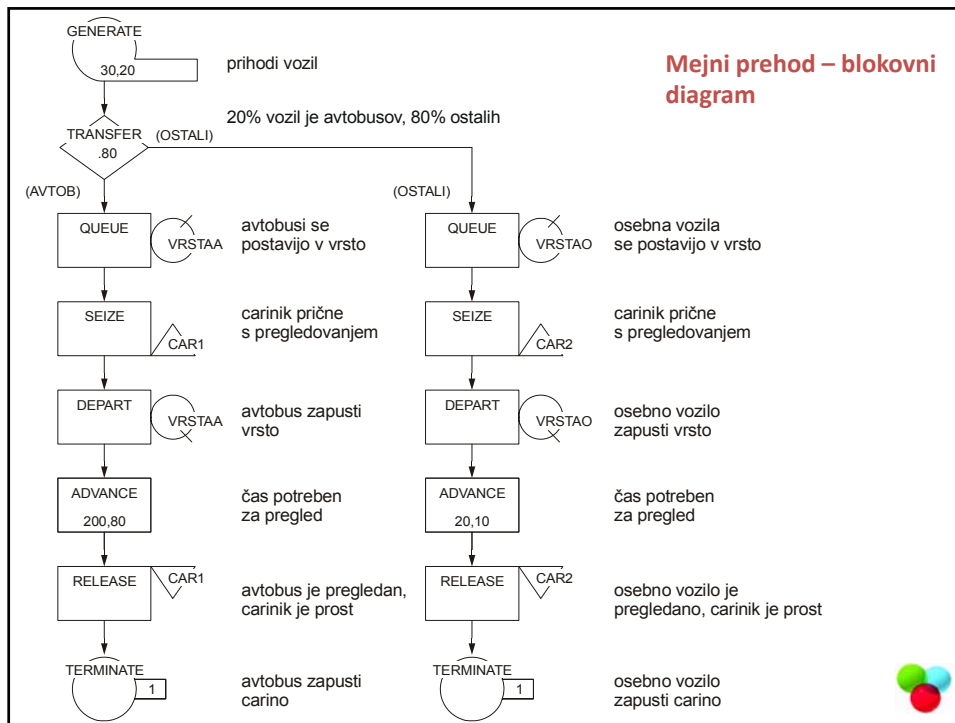
	Utilizacija	AVG time Q
1 DOK	0,78	0,77
2 DOKA	0,39	0,00

Primer – mejni prehod, uporaba bloka TRANSFER

Na mejni prehod pripelje vozilo v povprečju vsakih 30 sekund z odklonom 20 sekund, časi so enakomerno porazdeljeni. Med vozili je

20 % avtobusov, ki jih pregleduje prvi od dveh carinikov, povprečni čas pregleda je 200 s z odklonom +/- 80 s. Ostala vozila pregleduje drugi carinik. Čas enega pregleda je 20 s z odklonom +/- 10 s. Napišite rešitev – program v simulacijskem jeziku GPSS/H.





Mejni prehod - model

```

SIMULATE
*
* Primer - mejni prehod
*
* GPSS/H blokovni del
*
*
* Časi prihodov vozil
* 20% vozil je avtobusov, 80% ostalih
*
* Pregled avtobusov
*
AVTOB QUEUE VRSTAA           Avtobus se postavi v vrsto
      SEIZE CAR1             Zaposlimo carinika
      DEPART VRSTAA        Zapustimo vrsto
      ADVANCE 200,80       Čas pregleda
      RELEASE CAR1        Carinik je opravil pregled
      TERMINATE 1         Število pregledanih povečamo za 1
*
* Strežba ostalih
*
OSTALI QUEUE VRSTAO         Vozilo se postavi v vrsto
      SEIZE CAR2           Zaposlimo carinika
      DEPART VRSTAO       Zapustimo vrsto
      ADVANCE 20,10       Čas pregleda
      RELEASE CAR2        Carinik je opravil pregled
      TERMINATE 1         Število pregledanih povečamo za 1
*
* GPSS/H kontrolni stavki ...
*
  
```

Sedanost, prihodnost simulacijskih orodij

- zgodovina simulacijskih jezikov je precej bolj razgibana kot zgodovina programskih jezikov
- objektni koncept kot osnova za abstrakcijo realnega sveta upoštevan pri orodju "SIMULA"
- orodje razvito pred časom
- razvoj ni lovil koraka s časom

Sedanost, prihodnost (nad.)

- razvoj računalniške strojne opreme in grafike
- 3d igre
- animacija
- simulatorji
- potrebe po razvoju sodobnega orodja, ki je razvito na tehnološki platformi sedanosti

Sedanost, prihodnost (nad.)

- problem razvijalcev ~ obstoječe rešitve preveč kompleksne za uvedbo sprememb
- nezmožnost prilagajanja
- neupoštevanje objektnega pristopa
- neustrezen pristop k razvoju aplikacije
- obremenjenost razvijalcev z “zgodovino”
- na eni strani izjemna računalniška grafika, na drugi strani delo z ikonografijo ter konzolnimi aplikacijami

Flexsim

- www.flexsim.com
- Orodje za dogodkovno in zvezno simulacijo sistemov
- Omogoča analizo, vizualizacijo ter optimizacijo procesov – proizvodnja, oskrbovalna veriga, strežni sistemi, organizacijski sistemi itd.
- Flexsim je orodje, ki je zasnovano na paradigmi izgradnje 3D interaktivnih simulacijskih modelov
- 3D interaktivno delovno okolje
- Uvoz in izvoz podatkov iz in v preglednico Excel
- 3D interaktivni prikaz simulacijskih rezultatov kot. Npr. izkoriščenost strežnega mesta, povprečna dolžina vrste, čas čakanja v vrsti itd.

Simulacijska študija

- Opredelitev problema
- Izgradnja modela
- Izvedba simulacije
- Izdelava 3D predstavitev
- Izdelava datoteke s posnetkom .avi .mpg

Zasnova

- Flexsim je edino orodje, ki vključuje C++ IDE (Integrated Development Environment) s prevajalnikom v obliki grafičnega okolja za modeliranje
- C++ lahko uporabimo neposredno pri definicij logike modela
- Brez DLL-ov ter kompleksnih povezovanj z uporabniško podanimi spremenljivkami
- Zahtevni simulacijski projekti
- Napredna alternativa na področju razvoja simulacijskih orodij in izgradnje simulacijskih modelov

Modeliranje

- Uporaba visoko razvitih objektov, ki predstavljajo procesne aktivnosti ter sisteme vrst in razvrščanja
- Enostavna uporaba – povleci in klikni (drag-and-drop)
- Izredno hiter razvoj modelov
- Realizacija kompleksne logike je zajeta v objektu
- Uporabniška prilagodljivost objektov
- Vsak objekt je možno v izjemno velikem obsegu prilagoditi uporabniku
- Objekte lahko ustvarimo, uničimo ter celo vstavimo v druge objekte
- Podamo lahko unikatne lastnosti objekta in funkcionalnost ali pa določene dele dedujemo (objektna tehnologija)
- Objektna tehnologija je uporabljena za hitor in učinkovito izgradnjo modelov
- Uporaba – na področju proizvodnje ali poslovanja

Hierarhija

- Flexsim omogoča hierarhično izgradnjo modela
- Hierarhičnost izboljšana funkcionalnost modela ter sam proces modeliranja
- Dedovanje je osnovni princip izgradnje uporabniško podanih objektov
- Z objektno tehnologijo pospešimo proces razvoja ter zmanjšamo potrebo virov
- Uporaba dedovanja je enostavna
- Hierarhična struktura omogoča predstavitev kompleksnih sistemov
- Logična ureditev objektov – hierarhija
- Razumevanje modela ter istočasna delitev nalog med razvijalci
- Uporaba celotnega Microsoft Visual C++ okolja ter njegovih hierarhičnih lastnosti
- Flexsimova drevesna struktura omogoča enostaven pregled in organizacijo objektov

Uporabniško prirejanje

- Uporabniško prirejanje v orodju Flexsim je dobro podprto
- Vsak del razvite programske opreme "ima v mislih" uporabnika ter izpolnjevanje zahtev uporabnikov
- Dostopno s strani uporabnika, npr.: objekti, pogledi, GUI, meniji, seznamih opravi, parametri objektov
- Podajanje lastne logike modela – uporabniško razvite
- Priredba in dopolnitev ali izgradnja modelov "od začetka"
- Možnost ponovne uporabe objektov
- Objekti se lahko ustvarijo ter modificirajo z uporabo jezika C++, ki je srce Flexsima
- C++ upravlja z odzivom objektov
- Izgled Flexsima, uporabniški vmesnik, opravilne vrstice, meniji in GUI so kontrolirani preko "flexscript"-a – zmogljive C++ knjižnice

Izmenljivost

- Zaradi odprtosti objektov jih lahko izmenjujemo med uporabniki, knjižnicami in modeli
- Visoka stopnja zmoglosti uporabniškega prirejanja ter izmenljivost pomembno povečata hitrost izgradnje modela (proces modeliranja)
- Po tem, ko dodamo objekt v uporabniško knjižnico je vse, kar je potrebno za njegovo uporabo v nekem novem modelu le poteg miške
- Izboljšan življenjski cikel tako objektov kakor tudi modelov

Simulacijsko jedro

- Simulacijsko jedro skrbi za vizualizacijo modela, izris 3D in ortografskih predstavitev
- Simulacijsko jedro izvede preračun simulacijskih parametrov v realnem in "pospešenem" času
- Jedro skrbi za izvajanje klicev funkcij, preračunov, generiranje naključnih spremenljivk ter generiranje simulacijskih rezultatov
- Flexscript omogoča uporabo ukazov, ki jih posredujemo simulacijskemu jedru ob izvedbi simulacijskega modela
- Omogočena uporaba simulacijskega jedra s strani uporabnika ob izvedbi simulacijskega jedra – aplikativna učinkovitost

Dodatne možnosti

- Orodje za izvajanje scenarijev "kaj-če"
- Avtomatska izvedba scenarijev in shranjevanje simulacijskih rezultatov
- Analiza učinkovitosti sistema
- Standardni izhodni podatki
- Možnost definiranja lastnih izhodnih vrednosti
- Izvoz v standardne okenske aplikacije (preglednica ipd.)
- ODBC
- DDE

Vizualizacija

- Orodje Flexsim je zgrajeno s ciljem popolne 3D interaktivnosti modelov
- 1slika = 1000besed
- 1posnetek = 1000^3 besed
- Uporabljena je trenutno najsodobnejša tehnologija na področju računalniške grafike: OpenGL
- Možnost uvoza formatov 3D modelov: (.3ds (3D Studio MAX), .wrl (VRML), .dxf (AutoCAD), in .stl)
- Možnost dodajanja izvorov svetlobe, megle in drugih vizualnih efektov

Vizualizacija (nad.)

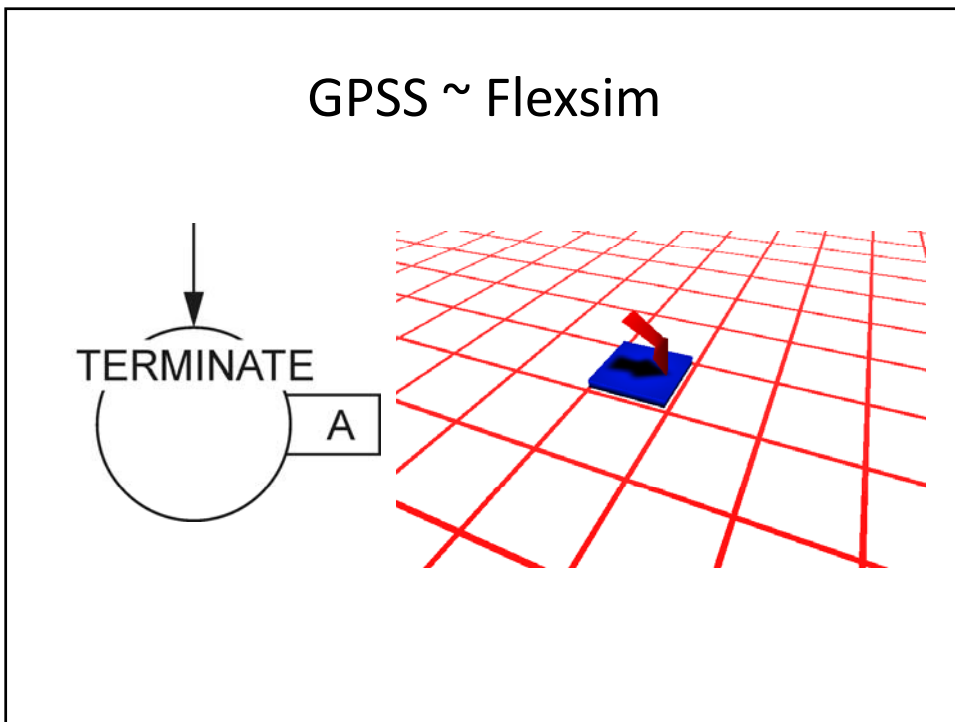
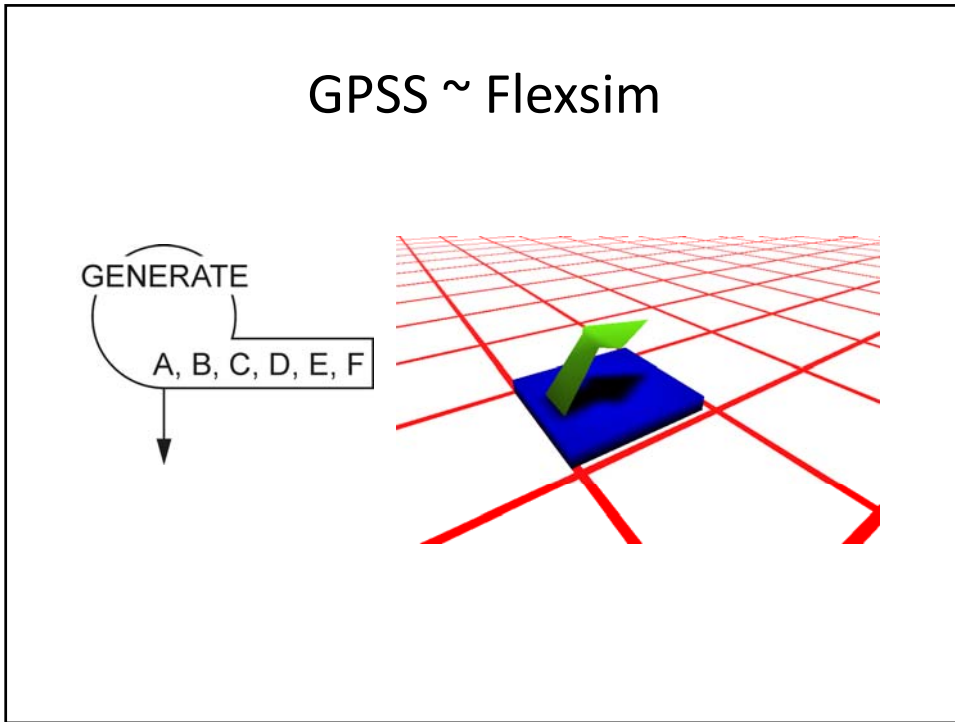
- Učinkoviti algoritmi za natančen preračun grafike
- Možnost izvedbe površin s kovinskim leskom – krom, zlago, steklo v realnem času
- Virtualna resničnost

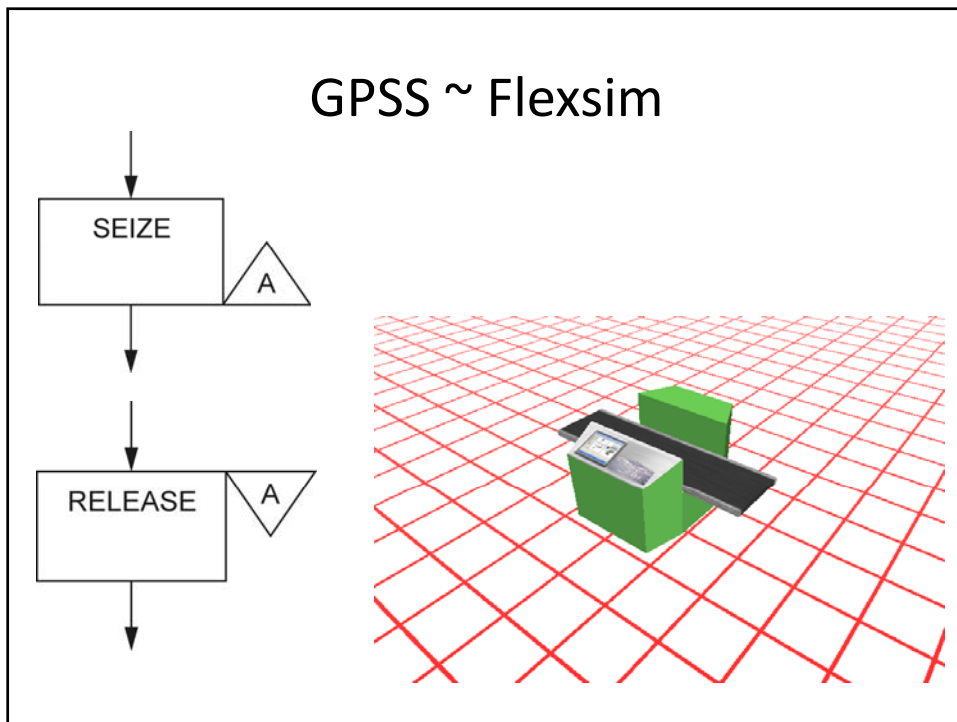
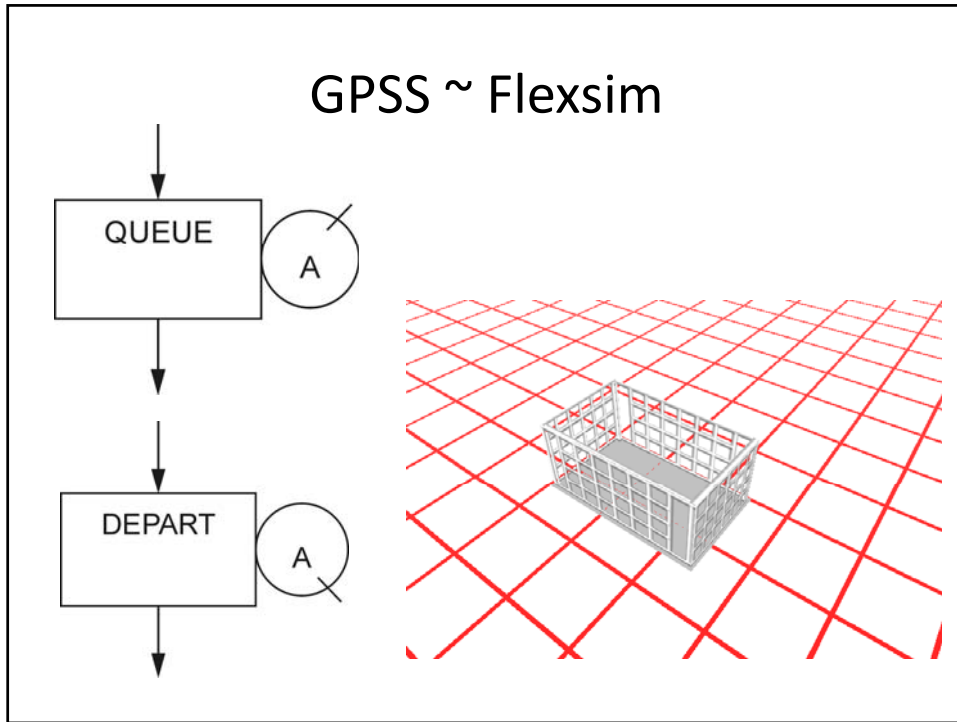
Predstavitve

- Poti preleta
- Generiranje video posnetka s predstavitvijo
- Vključena poslovna grafika – grafikoni, preglednice, histogrami itd.
- Izdelava uporabniško podanih poročil
- AC3D – izdelava poljubnih 3D objektov

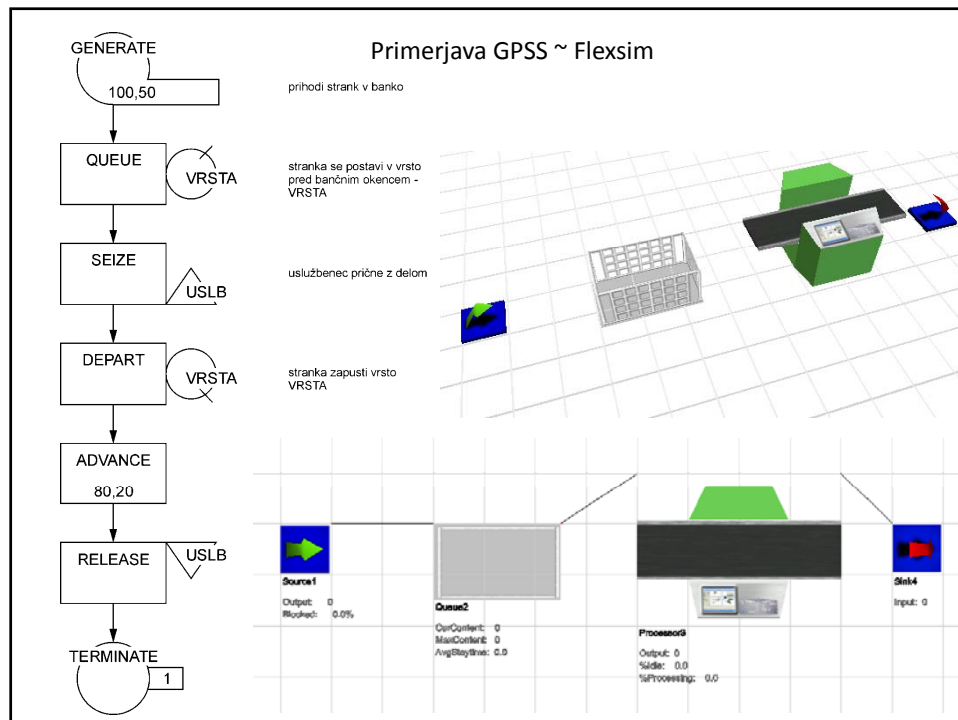
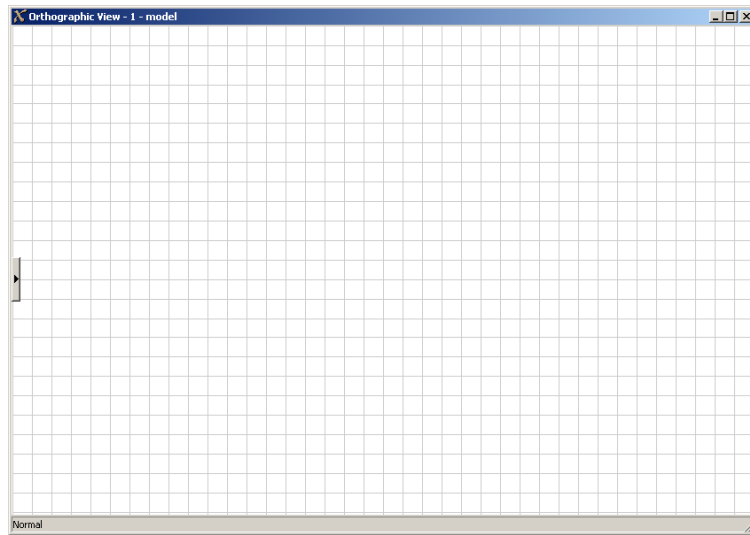
Expert Fit

- Del simulacijskega orodja Flexsim je Expert Fit
- Avtomatsko ter natančno določanje porazdelitve, ki se najbolj prilega realnim podatkom
- Pomembna metodološka točka
- Podpora pri naborih podatkov, ki niso celoviti
- Izvedba prilagajanja porazdelitev na področju proizvodnje, poslovanja, organizacije, vojske itd.
- Izvoz porazdelitev v ustrezen format
- Porazdelitve s področja aktuarstva, kemije, ekonomije, ekoloških študij, financ, gozdarstva, hidrologije, medicine, meteorologije, rudarstva, igralništva, fizike, psihologije, zanesljivosti, inženirstva in analize tveganja
- Vodilno orodje na področju prilagajanja porazdelitvenih funkcij



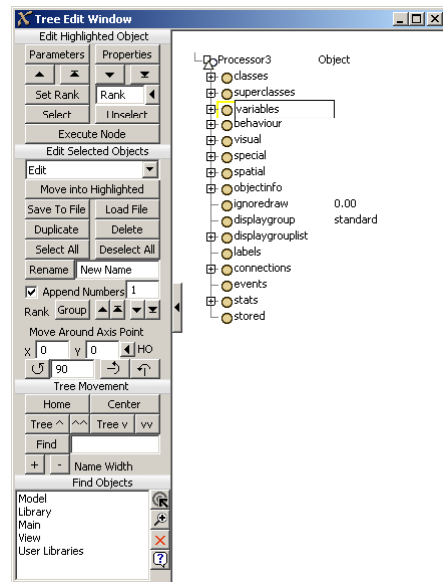


Flexsim ~ delovna površina z realnimi dimenzijami 1m x 1m



Hierarhija, objekti

- Vsak element je definiran v smislu objekta
- Drevesna struktura

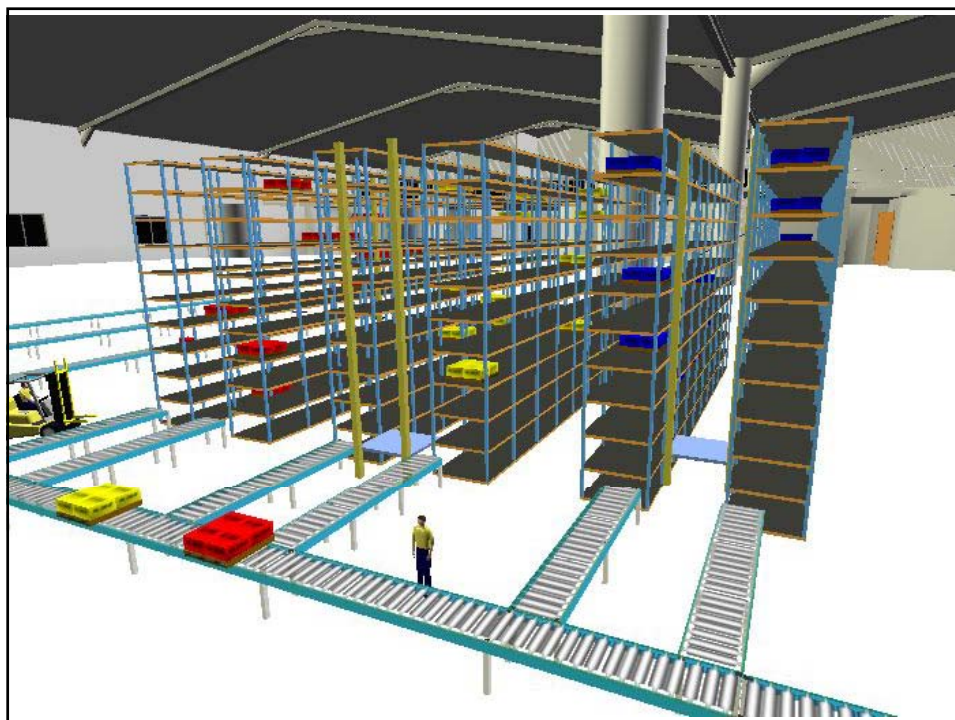


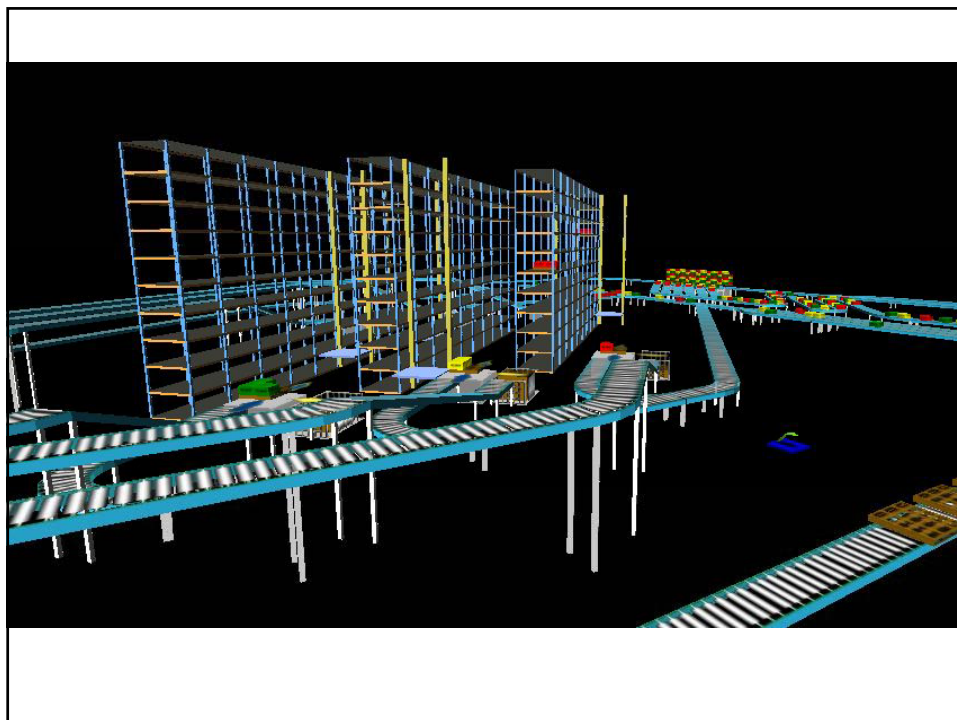
Simulacija

- Ali bo proizvodna enota izboljšala prepustnost?
- Kateri so vzroki za ozka grla v proizvodnji?
- Ali najmanjše novih delavcev doprinaša k finančni učinkovitosti sistema?
- Ali je potrebno zagotoviti višjo izkoriščenost strežnih mest?
- Kakšen je vpliv organizacije sistema na njegovo učinkovitost?
- Na zgoraj navedena vprašanja lahko odgovorimo na determinističen način.
- S pomočjo Flexsima in uporabe statističnih porazdelitev lahko upoštevamo faktorje, ki pomembno vplivajo na obravnavani sistem
- Zaupanja vredna rešitev – validacija modela
- Izvedba pravih akcij v realnem sistemu

Primer ~ embaliranje

- Vprašanja v zvezi s proizvodnjo (embaliranje)
- Število proizvodov v eni uri?
- Izvedba mešanic, tehtanja, embaliranja
- Uravnoveženost postavitve
- Organizacija dela
- Vizualizacija





Primer ~ pristanišče (luka)

- simulacija pristanišča (luke)
- natovarjanje, razkladanje ladijskih zabojnikov
- simulacija prihodov ladij
- simulacija transporta
- analiza kombinacij transporta ~ železnica, priklopniki
- razviti objekti (žerjavi ipd.)



Primer ~ skladiščenje

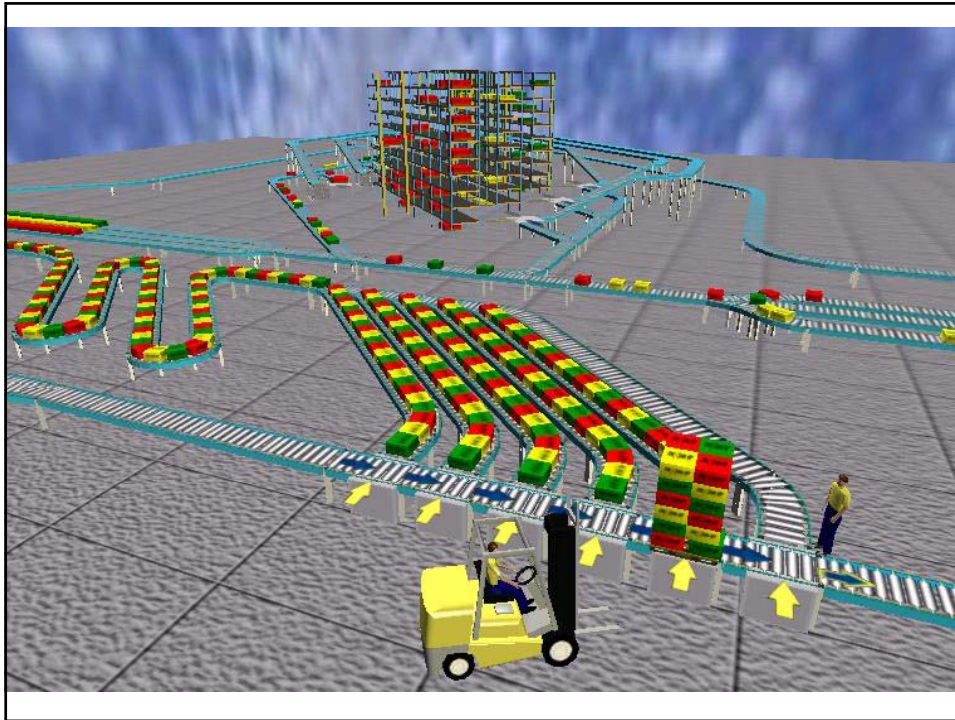
- modeliranje procesov v skladišču
- orodje je primerno za modeliranje sistemov različnih kompleksnosti
- določitev zaporedja aktivnosti
- izpolnjevanje naročil in potreb kupcev
- vizualizacija procesov v skladišču ~ 3d





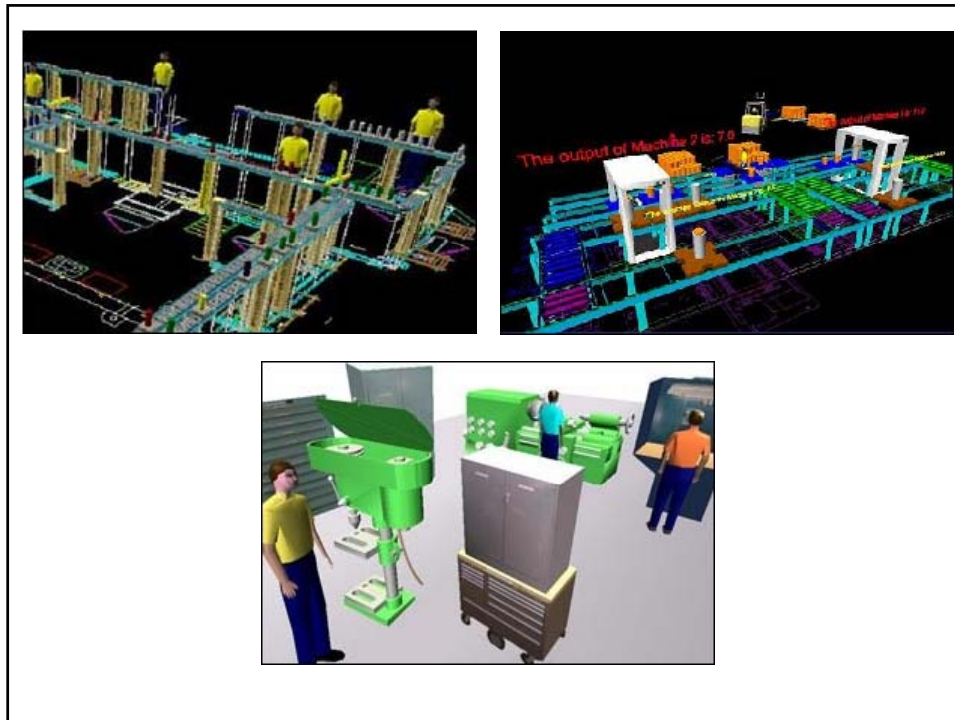
Primer ~ upravljanje z materialom

- simulacija in optimizacija sistemov za upravljanje z materialom
- možnost znatnih prihrankov
- vizualizacija delovanja tekočih trakov, viličarjev, avtomatsko vodenih vozil, žerjavov, robotov in ljudi
- določitev ustreznosti kapacitete ob konicah povpraševanja



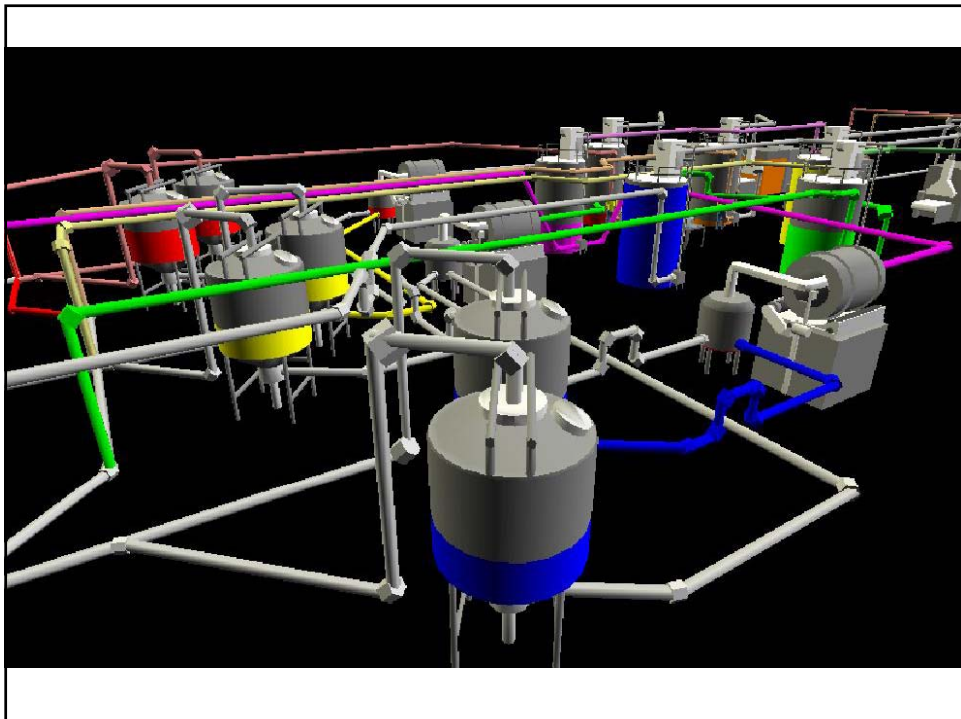
Primer ~ proizvodnja

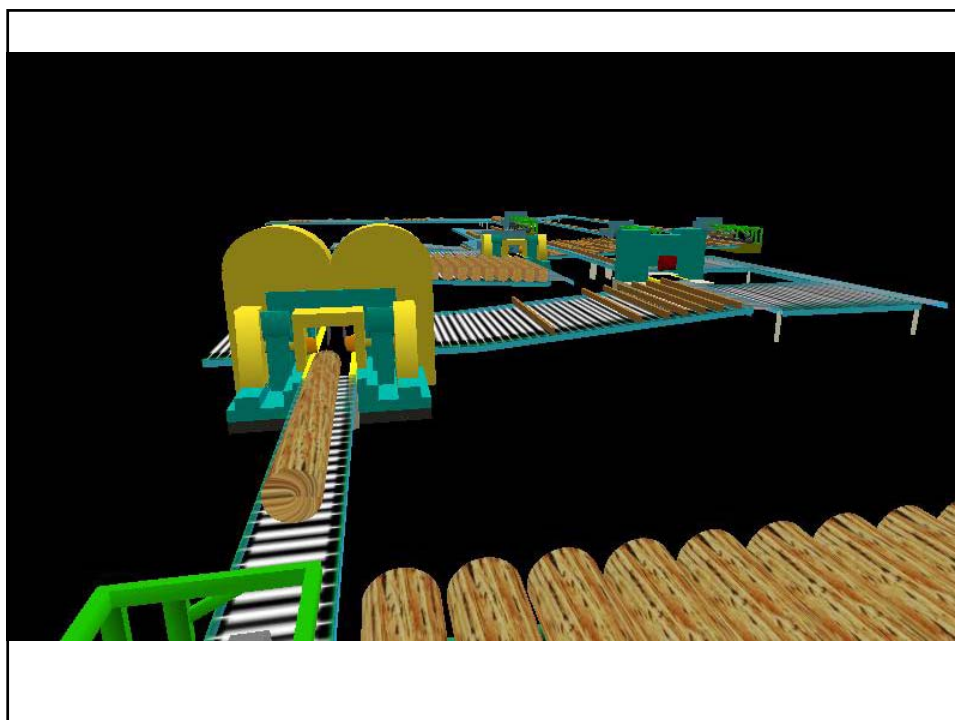
- Število proizvodov v eni uri?
- embaliranje, tehtanje
- uravnoteženost proizvodnega sistema
- analiza pretočnosti
- določitev ozkih grl
- izkoriščenost strežnih mest
- analize delovnih operacij
- analiza izvedljivosti in učinkovitosti proizvodnega plana



Primer ~ uporabniško podana domena

- reševanje problemov v domeni podani s strani uporabnika
- upravljanje v primeru večjih naravnih nesreč
- procesna proizvodnja
- žaga
- razvoj uporabniških knjižnic in objektov za ponovno uporabo





Primer ~ predstavitev

- nadomestilo za predstavitve s PowerPoint-om
- 3d predstavitve
- preleti preko objektov
- vstavljanje napisov ter interaktivni izpis rezultatov

