

## Koncept simulacionog časovnika

- Postoje dva osnovna pristupa upravljanju internim vremenom simulacije, odnosno simulacionim časovnikom:

- ✓ napredovanje simulacionog vremena u **fiksnim intervalima**
- ✓ napredovanje simulacionog vremena u **intervalima promenljive dužine** (intervali između dva događaja)

- Prvi pristup podrazumeva da vreme napreduje za unapred definisanu dužinu intervala (1 sekund, 5 petominutnih intervala,...), nezavisno od toga da li je u intervalu bilo nekih događaja ili nije.

- Drugi pristup, međutim, podrazumeva da simulaciono vreme napreduje od trenutka pojave jednog događaja, do trenutka pojave narednog.

- Pristup koji podrazumeva napredovanje po intervalima između dva događaja može dati mnogo preciznije informacije o sistemu i njegovim performansama, ali je sa druge strane ovakav pristup nepogodan za vizuelizaciju, jer bi konstantna brzina kadrova potpuno poremetila stvarnu sliku toka vremena simulacije.

FIKSNI INTERVAL DUŽINE 0.25 h	DOGADJAJ	FIKSNI INTERVAL DUŽINE 0.50 h	DOGADJAJ
0 – 0.25	D1, D2	0 – 0.5	D1, D2, O1
0.25 – 0.50	O1	0.5 – 1.0	O2, D3
0.50 – 0.75	O2	1.0 – 1.5	D4, O3
0.75 – 1.00	D3	1.5 – 2.0	-
1.00 – 1.25	D4	2.0 – 2.5	O4
1.25 – 1.50	O3		
1.50 – 1.75	-		
1.75 – 2.00	-		
2.00 – 2.25	O4		

- Isto tako, nekada simulaciju i nije moguće vršiti na drugi način, već korišćenjem fiksnih intervala. Ako se, naprimer, raspolaže snimkom broja vozila koja dolaze na front pretovara u nekom intervalu, tada se taj interval mora uzeti kao jedinica vremena simulacije i onda nije moguće utvrditi tačna vremena dolazaka vozila, niti primeniti koncept napredovanja časovnika po intervalima između događaja.
- Važno je istaći da oba pristupa obezbeđuju simulaciju paralelizma procesa.
- Koncepti ova dva pristupa u formi pseudo koda

**SIMULACIJA U FIKSNIM INTERVALIMA**

```
while simulirano vreme not end  
  povećaj simulirano vreme za definisanu jedinicu  
  if dogadjaj se realizuje tokom simulirano vreme  
    simuliraj dogadjaj  
  end if  
end while
```

**SIMULACIJA U PROMENLJIVIM INTERVALIMA**

```
while lista dogadjaja not empty and simulirano vreme not end  
  uzmi dogadjaj sa najranijim početkom  
  pomeri simulacioni časovnik na vreme početka dogadjaja  
  simuliraj dogadjaj  
end while
```

## Mehanizmi simulacije

- Za opisivanje logike, odnosno mehanizma funkcionisanja, u modelima diskretne simulacije, primenjuju se tri osnovna pristupa (Ball, 1996; Ingalls, 2002; Banks et al. 2005).
- Medjutim, iako se objektno orijentisana simulacija najčešće označava kao oblik procesno interakcijskog pristupa, zbog niza specifičnosti ovaj koncept pogodnije je razmatrati posebno, pa se onda umesto tri osnovna prilaza mogu izdvojiti četiri:
  - ✓ rasporedjivanja (i skeniranja) događaja
  - ✓ skeniranja aktivnosti
  - ✓ procesno interakcijski pristup
  - ✓ objektno orijentisana simulacija
- **Pristup baziran na kreiranju i simulaciji liste događaja**, opisan na primeru i predstavljen tabelom 2, svakako je dominantan i najčešće korišćen mehanizam diskretne simulacije.

*Taj koncept u najkraćem se može sažeti rečenicom da se radi o pristupu koji događaj opisuje kao trenutnu promenu, aktivnost kao trajanje, a proces kao niz događaja i aktivnosti zajedno.*

*Čak i kada konkretan programski paket te pristupe ne podržava, prikaz osnovne ideje i koncepta simulacionog modela za slučaj **skeniranja aktivnosti**, odnosno **procesno interakcijskog pristupa** važan je za razumevanje mogućih načina kreiranja simulacionog modela.*

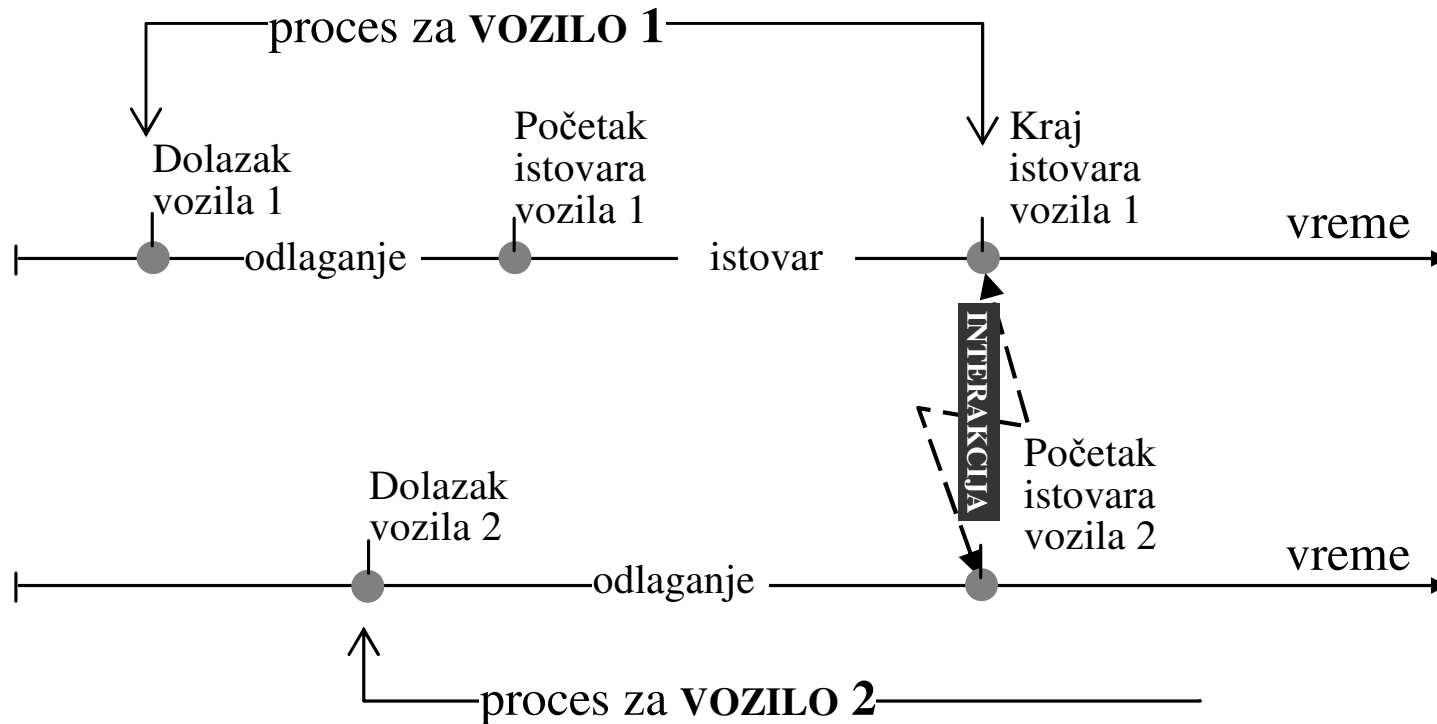
- ***Pristup zasnovan na skeniranju aktivnosti*** nešto je redji, mada je jednostavan za razumevanje. Naime, ovaj pristup podrazumeva napredovanje simulacionog časovnika u fiksnim intervalima i korišćenje odgovarajućih pravila na bazi kojoj se odlučuje o započinjanju aktivnosti u nekom intervalu, **pri čemu su događaji takodje predstavljeni aktivnostima čije je trajanje jednako nuli.**

U principu, skeniranje aktivnosti realizuje se u veoma kratkim vremenskim intervalima što znači da se u svakom ispituje ispunjenost uslova za njihovo otpočinjanje. S obzirom da skraćivanje intervala znači u isto vreme i veoma veliki broj intervala koji se simuliraju, trajanje simulacije primenom ovog koncepta najčešće je veoma dugo, **pri čemu se, logično, skenira i veliki broj intervala u kojima se ne realizuje nikakva aktivnost.**

U primeni su i određene modifikacije ovog pristupa kojima se obezbedjuje veća brzina simulacije. Podelom aktivnosti u dve grupe kod kojih jedna sadrži samo “**bezuslovne**” događaje i aktivnosti sa poznatim, fiksnim vremenom nastanka, odnosno realizacije, a druga jedino “**uslovne**” koji se realizuju nakon ispunjenja određenih uslova povezanih sa drugim aktivnostima, omogućuje se da se **bezuslovne aktivnosti simuliraju po principu skeniranja događaja**, a druga grupa “**uslovnih**”, prema konceptu simulacije na osnovu skeniranja aktivnosti.

Tako listu događaja sačinjavaju samo bezuslovne aktivnosti izmedju kojih se simulacioni časovnik pomera po principu varijabilnih intervala, a potom se u fiksnim intervalima izmedju dva događaja skeniraju uslovne aktivnosti i analizira zadovoljenje pretpostavki za njihovu realizaciju.

- **Procesno interakcijski pristup** podrazumeva da se model koncipira kao skup procesa koji su u medjusobnoj interakciji. Svaki proces uključuje operacije koje se realizuju tokom “životnog veka” entiteta i obuhvata sve događaje i aktivnosti koje su sa time povezane. Ukoliko se posmatra primer jednokanalnog pretovarnog sistema, u kome je entitet vozilo koje dolazi na istovar, procesni pristup može se predstaviti interakcijom dva entiteta (vozila) koja dolaze na front pretovara.



Lista događaja obuhvata hronološki uređene sekvence događaja i aktivnosti svih procesa, sa indikacijom pripadnosti određenom procesu. Proces se može naći u jednom od sledećih stanja:

- AKTIVAN (active) – proces se izvršava u tekućem vremenu (u sistemu se uvek nalazi samo jedan takav proces)
- SPREMAN (ready) – proces je na listi događaja čekajući aktiviranje
- NEAKTIVAN (idle) – proces nije na listi događaja, ali može biti reaktiviran od strane nekog entiteta (označava se i kao BLOKIRAN - blocked)
- ZAVRŠEN (terminated) – okončana je sekvenca aktivnosti u procesu te nije na listi događaja i ne može biti reaktiviran

U slučaju ovog pristupa simulacioni časovnik se pomera između događaja, dakle u intervalim promenljive dužine, a simulacija bazirana na ovom konceptu često se primenjuje s obzirom da je intuitivno najbliža poimanju procesa u realnom sistemu.

Pored toga, ovaj pristup najlakše je predstaviti blok dijagramom, pri čemu se interakcija procesa, (interakcija blokova u dijagramu) najčešće realizuje automatski. ***S obzirom da je mehanizam simulatora koji obezbedjuje interakciju procesa najčešće skriven od korisnika***, to se on može posvetiti opisivanju sistema preko odgovarajućih blok dijagrama ili mreža, ne vodeći mnogo računa o često veoma složenoj strukturi međusobnih uticaja pojedinih procesa.

Dakle, **osnovna komponenta procesnog pristupa je proces**, pri čemu se sekvenca događaja sastoji u povezivanju parcijalnih sekvenci u svim procesima.

Listu budućih događaja čini **sekvenca listi**, a simulator izvršava sledeće zadatke:

- ✓ "postavlja" proces u određeni vremenski trenutak na listi događaja,
- ✓ briše proces sa liste,
- ✓ aktivira proces koji odgovara narednom događaju sa liste i
- ✓ preraspoređuje procese sa liste.

➤ Poseban tip procesne simulacije, a slobodno se može reći i najrazvijeniji, odnosno najnapredniji, je i **objektno orijentisana simulacija**. Ovaj pristup koji počiva na konceptu objektnog programiranja, baziran je na premisi da stvarnost može biti opisana preko skupa objekata koji su u međusobnoj interakciji i na tri osnovne pretpostavke (Cunningham H.C, 2000):

1. Predstavljanje kompleksnih sistema kao objekata u međusobnoj interakciji čini razumevanje tih sistema jednostavnijim nego kada se to čini nekim drugim pristupom (ovo pojednostavljuje analizu i čini je pouzdanijom)
2. Ponašanje objekata u realnim sistemima ima tendenciju da bude stabilno u vremenu (novi tipovi objekata nastaju veoma sporo, a postojeće vrste sporo nestaju). Posledica toga je mogućnost ponovnog korišćenja koda, razvoj prototipova i postepena poboljšanja.
3. Promene se dešavaju kod malog broja objekata u sistemu.

➤ Osnovu strukture objektnog modela čine:

- **OBJEKTI** (objects) – predstavljaju abstraktne strukture podataka
- **KLASE** (classes) – abstraktni tipovi podataka
- **NASLEDJIVANJE** (inheritance) – hijerarhijske relacije medju klasama
- **POLIMORFIZAM** (polymorphism) – mnoštvo oblika