

VERIFIKACIJA SIMULACIONIH MODELA

Proces verifikacije je ispitivanje da li je i u kojoj meri, konceptualni model na odgovarajući način predstavljen računarskim kodom. U postupku verifikacije nema standardnog recepta. Zato je potrebno izvršiti više različitih provera:

- **Ručna verifikacija logičke ispravnosti:** model se izvesno vreme propušta na računaru ručno, a potom se porede dobijeni rezultati.
- **Modularno testiranje:** pojedinačno testiranje svakog modula kako bi se ustanovilo da li daje razumne izlaze za sve moguće ulaze.

VERIFIKACIJA SIMULACIONIH MODELA

- **Provera u odnosu na poznata rešenja:** podesimo model tako da predstavlja sistem čija su rešenja poznata i upoređujemo ih sa rezultatima modela.
- **Testiranje osetljivosti:** variramo jedan parametar, dok ostali ostaju nepromenjeni i proveravamo da li je ponašanje modela osetljivo na promenu tog parametra.
- **Testiranje na poremećaje:** postavimo parametre modela na neprirodne vrednosti i proveravamo da li se model ponaša na neshvatljiv način. Tako se mogu otkriti greške u programu koje se vrlo teško mogu uočiti na drugi način.

SIMULACIJA DISKRETNIH DOGAĐAJA

Simulacija diskretnih događaja je metoda simulacionog modelovanja sistema kod kojih se diskretne promene stanja u sistemu ili njegovom okruženju događaju diskontinualno u vremenu. Koristi se uglavnom za analizu dinamičkih sistema sa stohastičkim karakteristikama.

Događaj je diskretna promena stanja entiteta sistema i nastupa u određenom trenutku vremena. Sistemi sa diskretnim događajima su: banke, auto servisi, pošte, samoposluge jer se stohastički karakter ogleda u slučajnoj prirodi veličina (događaja) za opis ovakvog sistema.

SIMULACIJA DISKRETNIH DOGAĐAJA

Prema Šenonu, opis diskretnih događaja je sledeći:

Entiteti, koji se opisuju atributima i uzajamno deluju u aktivnostima, pod određenim uslovima stvaraju događaje koji menjaju stanje sistema.

FORMALNI OPIS SISTEMA SA DISKRETNIM DOGAĐAJIMA

Formalni opis sistema diskretnih događaja može se predstaviti uređenom šestorkom M_d :

$M_d = \{U, S, Y, \delta, \lambda, \tau\}$, gde su:

U – skup eksternih događaja,

S – skup sekvencijalnih stanja diskretnih događaja,

Y – skup izlaza,

δ – kvazi prenosna funkcija koja se zadaje sa funkcijama:

δ_ϕ koja preslikava $S \rightarrow S$; pokazuje u koje će stanje preći sistem iz stanja s , **ukoliko ne nastupi ni jedan** eksterni događaj.

δ_{ex} koja preslikava $U \times S \times T \rightarrow S$; pokazuje u koje će stanje preći sistem iz stanja s **kada nastupi eksterni događaj U u vremenskom trenutku t ,**

FORMALNI OPIS SISTEMA SA DISKRETNIM DOGAĐAJIMA

λ – izlazna funkcija koja preslikava $S \rightarrow Y$

τ – funkcija nastupanja vremena koja preslikava

$S \rightarrow R^{+0,\infty}$;

pokazuje koliko će dugo sistem ostati u stanju **s**, **pre** nego što nastupi naredna promena stanja, pod pretpostavkom da neće nastupiti ni jedan eksterni događaj.

DOGAĐAJ, AKTIVNOST I PROCES

Kod modela sa diskretnim događajima, pored koncepata koji opisuju strukturu kao što su objekti, relacije između njih i njihovi atributi, uvedeni su i koncepti za opis dinamike: događaj, aktivnost i proces.

Događaj predstavlja diskretnu promenu stanja entiteta u sistemu ili njegovom okruženju. Između dva uzastopna događaja stanje sistema se ne menja. Događaj može nastupiti zbog ulaska/izlaska privremenog entiteta u/iz sistema (dolazak/odlazak klijenta) – **eksterni**; ili zbog promene atributa pojedinih objekata sistema (opslužilac u sistemu postaje slobodan ili zauzet) - **interni**.

DOGAĐAJ, AKTIVNOST I PROCES

Uslovni događaji mogu nastupiti kada je ispunjen uslov njihovog nastupanja. Obično su povezani sa zauzimanjem nekog resursa. *Bezuslovni događaji* su oni čiji je jedini uslov da tekuće vreme simulacije bude jednako vremenu njegovog nastupanja. Obično su povezani sa oslobađanjem nekog resursa odnosno planiranim završetkom neke aktivnosti.

Aktivnost je skup događaja koji menjaju stanje jednog ili više entiteta. Trajanje aktivnosti može biti unapred definisano (determinističko) ili da zavisi od ispunjenja uslova pa je vreme završetka takve aktivnosti nepoznato (stohastičko).

DOGAĐAJ, AKTIVNOST I PROCES

Proces je niz uzastopnih, logički povezanih događaja kroz koje prolazi neki privremeni objekat. To je hronološki uređena sekvenca događaja koja opisuje jednu pojavu od nastajanja do gašenja (terminacije). On može da obuhvati deo ili celokupan život privremenog entiteta.

RAZVOJ SIMULACIJE DISKRETNIH DOGAĐAJA

Ključni elementi razvoja simulacije diskretnih događaja su:

- 1.mehanizam pomaka vremena,
- 2.Pristup generisanju događaja i
- 3.osnovne strategije simulacije.

RAZVOJ SIMULACIJE DISKRETNIH DOGAĐAJA

Mehanizam pomaka vremena

U simulaciji diskretnih događaja koriste se dva osnovna mehanizma pomaka vremena: 1.pomak vremena za konstantni priraštaj i 2.pomak vremena na naredni događaj.

Pomak vremena za konstantni priraštaj – vreme u simulacionom modelu se menja tako da se uvek dodaje konstantan priraštaj Δt . Nakon svakog pomaka vremena, odnosno ažuriranja vrednosti simulacionog sata, ispituje se da li je u prethodnom intervalu vremena trebalo da dođe do nastupanja nekih događaja.

RAZVOJ SIMULACIJE DISKRETNIH DOGAĐAJA

Ukoliko jeste, tada se ti događaji planiraju za kraj intervala. Nedostatak je u tome što se pomeranjem događaja na kraj vremenskog intervala uvodi greška u simulaciji. Događaji koji nisu istovremeni u ovom se pristupu prikazuju kao istovremeni, a potom se određuje redosled njihovog izvođenja (koji se može razlikovati od stvarnog redosleda).

Smanjenjem vremenskog prirasta te se greške smanjuju, ali se povećava vreme koje se troši na izvođenje simulacije, a takođe se povećava i broj vremenskih intervala u kojima nema događaja.

RAZVOJ SIMULACIJE DISKRETNIH DOGAĐAJA

Pomak vremena na naredni događaj – simulacioni sat se pomera na vreme u kojem će nastupiti prvi naredni događaj (ili više njih). Simulacija se završava kada nema više događaja ili kada je zadovoljen neki unapred definisan uslov završetka simulacije. Na ovaj način se izbegava greška u vremenu izvođenja događaja a ujedno se preskaču intervali u kojima nema događaja. Ovaj princip je složeniji ali i efikasniji pa svi ključni simulacioni jezici koriste ovaj mehanizam.

RAZVOJ SIMULACIJE DISKRETNIH DOGAĐAJA

Generisanje događaja

Događaj se opisuje sa više atributa, koji formiraju **slog** događaja. S obzirom na promenljiv broj događaja u vremenu, slogovi događaja se memorišu u **listama događaja**.

Postoje dva pristupa generisanja događaja:

1. **Definisanje događaja unapred** – svi događaji su unapred poznati i definisani, a lista događaja sadrži slogove svih događaja.
2. **Definisanje narednog događaja** – poznat je jedino prvi naredni događaj, a lista događaja sadrži samo jedan slog, slog poznatog događaja. Pri izvršavanju događaja, planira se i ubacuje u listu njegov naslednik.

RAZVOJ SIMULACIJE DISKRETNIH DOGAĐAJA

Događaje možemo svrstati u dve kategorije u odnosu na mesto nastajanja (generisanja):

- 1. *Eksterni*** događaji su oni događaji koji ne zavise od modela i predstavljaju uticaj okoline na sistem (dolazak kupaca u samoposlugu, vozača u auto servis);
- 2. *Interni*** događaji zavise od modela i u njemu se generišu (dolazak kupaca u red na kasu).

RAZVOJ SIMULACIJE DISKRETNIH DOGAĐAJA

Strategija izvođenja simulacije

Te strategije su:

- 1.raspoređivanje događaja;**
- 2.skaniranje aktivnosti i**
- 3.interakcija procesa.**

Raspoređivanje događaja podrazumeva da se događaji planiraju unapred i drže u listi budućih događaja, najčešće sortirani po vremenu nastupanja i prioritetu.

RAZVOJ SIMULACIJE DISKRETNIH DOGAĐAJA

Procedura planiranja događaja: generiše se slog događaja; dodele se vrednosti njegovih atributa (vreme nastajanja, prioritet); događaj se stavlja u listu budućih događaja koja je uređena po vremenu nastupanja događaja i prioritetu. *Funkcionisanje simulatora:* sa liste budućih događaja uzima se prvi; ažurira se simulacioni sat na vreme njegovog nastupanja; na osnovu tipa izabranog događaja poziva se odgovarajuća procedura koja izvršava sva ažuriranja u modelu i simulatoru; kada se izvrše svi događaji koji imaju isto vreme nastupanja, simulacioni sat se ažurira na vreme sledećeg događaja iz liste budućih događaja.

RAZVOJ SIMULACIJE DISKRETNIH DOGAĐAJA

Skaniiranje aktivnosti podrazumeva da se događaji implicitno raspoređuju tako da se promena stanja izvršava preko funkcija koje se nazivaju aktivnosti. Svaka aktivnost ima *uslov i akciju*.

Za svaki vremenski korak Δt , aktivnosti se skaniraju i traži se prva aktivnost koja ima zadovoljen uslov. Tada se izvršava odgovarajući programski segment koji specificira akciju za zadatu aktivnost.

RAZVOJ SIMULACIJE DISKRETNIH DOGAĐAJA

Proces skaniranja traje sve dok sve aktivnosti ne budu blokirane. Tada i samo tada se simulacioni sat ažurira na sledeći vremenski korak. Skaniranje aktivnosti je bolje od raspoređivanja događaja kada je broj aktivnosti u modelu mali, a broj događaja u okviru aktivnosti veliki, jer se štedi na računarskim operacijama koje se odnose na listu i njegovo skidanje sa liste događaja.

RAZVOJ SIMULACIJE DISKRETNIH DOGAĐAJA

Interakcija procesa predstavlja tehniku simulacije koja je nastala kombinacijom raspoređivanja događaja i skaniranja aktivnosti. Proces možemo posmatrati kao skup isključivih aktivnosti, povezanih tako da terminiranje jedne aktivnosti dozvoljava inicijalizaciju neke druge aktivnosti iz skupa aktivnosti procesa.

Glavni problem je sinhronizacija procesa jer je model sistema skup paralelnih procesa od kojih neki mogu biti uzajamno isključivi. Da do ovoga ne bi došlo uvode se dve naredbe WAIT i DELAY i to i u uslovnom i u bezuslovnom kontekstu.

RAZVOJ SIMULACIJE DISKRETNIH DOGAĐAJA

WAIT naredba u uslovnom kontekstu ima oblik: WAIT UNTIL <uslov> – ako je uslov zadovoljen, proces koji čeka na taj uslov nastavlja svoj tok, a u suprotnom biće blokiran. Tada se ispituje koji od ostalih procesa može da se aktivira i on se izvršava.

DELAY naredbom se odlaže nastavak procesa za određeni broj vremenskih jedinica. U tom slučaju, ispituje se koji od ostalih procesa može da ostvari svoj tok. U bezuslovnom kontekstu naredba DELAY označava se i sa ADVANCE <broj vremenskih jedinica>.

RAZVOJ SIMULACIJE DISKRETNIH DOGAĐAJA

U okviru procesora održavaju se dve liste događaja:

1. Lista tekućih događaja (LTD) koja sadrži uslovne događaje i sortirana je po prioritetu.
2. Lista budućih događaja (LBD) koja sadrži raspoređene događaje koji tek treba da nastupe i sortirana je po vremenu nastupanja.

Pri svakom premeštanju događaja iz LBD u LTD planira se njegov naslednik koji se stavlja u LBD gde čeka svoje vreme nastupanja. Pre početka simulacije, za svaki proces generiše se prvi događaj i stavlja se u LBD a zatim se vreme simulacije ažurira na vreme događaja sa vrha LBD.

RAZVOJ SIMULACIJE DISKRETNIH DOGAĐAJA

Rad procesora odvija se u dve faze:

1. Faza ažuriranja vremena simulacije,

simulacioni sat se postavlja na vrednost vremena nastupanja prvog događaja iz LBD, a zatim se prebace svi događaji sa tim vremenom nastupanja u LTD. Pri tome se za svaki premešten događaj planira naslednik i stvalja se u LBD.

RAZVOJ SIMULACIJE DISKRETNIH DOGAĐAJA

2. Faza skaniranja liste tekućih događaja,

skanira se LTD i ukoliko ima neblokiranih događaja oni se izvršavaju. Kada nema događaja u LTD ili su svi blokirani (WAIT naredbom), tada se prenosi prvi događaj iz LBD i ažurira se simulacioni sat na njegovo vreme.

Simulacija se odvija sve dok se ne zadovolji kriterijum za završetak simulacije, kada se vrši završna obrada rezultata simulacije i procesor završava rad.

Kontrolna pitanja

45. Koje je provere potrebno izvršiti u procesu verifikacije?
46. Kako se vrši testiranje osetljivosti?
47. Kako se vrši testiranje na poremećaje?
48. Šta je simulacija diskretnih događaja?
49. Šta je diskretni događaj?
50. Šta je aktivnost?
51. Šta je process?
52. Koji su ključni elementi razvoja simulacije diskretnih događaja?
53. Opišite mehanizam pomaka vremena za konstantan priraštaj.
54. Opišite mehanizam pomaka vremena na naredni događaj.

Kontrolna pitanja

55. Koji pristupi generisanja događaja postoje?
56. Šta je interakcija procesa?