

PANEVROPSKI UNIVERZITET APEIRON
FAKULTET INFORMACIONIH TEHNOLOGIJA
BANJA LUKA

Seminarski rad
SIMULACIJA ŠALTERSKOG SLUŽBENIKA

Mentor:
Prof. dr Zoran Avramović

Student:
Siniša Božić
192/20-RITP

Banja Luka, 2022

SADRŽAJ

1. UVOD.....	1
2. MODELIRANJE I SIMULACIJE	2
2.1. Odnosi sistema i modela	3
2.2 Primjena	4
2.3 Rješavanje problema korištenjem modelovanja i simulacija	5
3 FORMULISANJE PROBLEMA I MODELA	6
4 KREIRANJE MODELA SISTEMA.....	8
5 SIMULACIONI EKSPERIMENTI.....	10
6 TUMAČENJE REZULTATA	11
7 DOKUMENTOVANJE.....	12
8 IMPLEMENTACIJA REZULTATA.....	13
9 VALIDACIJA MODELA	14
10. GPSS SIMULACIONI JEZIK.....	16
11. SIMULACIJA ŠALTERSKOG SLUŽBENIKA.....	19
12. ZAKLJUČAK	25
LITERATURA	26
Popis priloga:	27

1 UVOD

Simulacije realnog svijeta predstavljaju vjerovatno najkompleksniji spoj prirodnih i društvenih nauka i tiču se proučavanja objekata ili procesa u realnom svijetu na veoma visokom nivou razumijevanja. Tendencija oponašanja prirodnih pojava u čovječanstvu je duga vjekovima, a tokom vremena, sa pronalaskom digitalnih računara konačno je postao dostupan takav aparatus kojim je moguće simulirati kako prirodne tako i društvene pojave. U poveljima računarstva, ove simulacije su bile jednostavne, u skladu sa vrlo ograničenim hardverskim sposobnostima, međutim naglim napretkom tehnologije stvaraju se sve složeniji modeli koji se simuliraju na sve složenije načine.

Proučavanje modela i simulacija se odnosi na multidisciplinarni pristup u kome osoba ili tim stručnjaka mora da ima višestruka znanja, prije svega o predmetu ili pojavi, odnosno sistemu koji se proučava i želi modelirati, a potom i o cijelom nizu drugih, direktno ili indirektno povezanih znanja kao što su matematika, fizika, ekonomija, informatičke nauke i tome slično, u skladu sa prirodom posmatranog sistema.

Cilj ovog rada je, u skladu sa njegovim obimom, pružiti osnovne teorijske postulate na osnovu kojih funkcioniše modelovanje i simulacija, dati pregled osnovnih procesa kojima se dolazi do modela, naglasiti bitne korake u njegovom nastajanju, kao i pružiti osnovna znanja o simulacijama. U praktičnom dijelu rada predstavljen je problem rada šalterskog službenika, koji bi mogao da bude shvaćen kao optimizaciona analiza rada jedne osobe zaposlene u nekoj, za ovaj rad nebitnoj djelatnosti. Kratkim modelom i simuliranim vremenom je u GPSS jeziku predstavljen jednostavan način sagledavanja ovog realnog problema i njegovog modelovanja.

Simulacioni jezici kakav je GPSS su usko specijalizovani programski jezici koji su podešeni za analiziranje kompleksnih problema koji se teško ili nikako ne mogu rješavati analitičkim putem, pa smatram da je korisno da student informacionih tehnologija bude upoznat sa osnovnim tehnikama kako rezonovanja u smislu kreiranja modela tako i u osnovnoj primjeni nekog od simulacionih jezika.

2 MODELIRANJE I SIMULACIJE

Da bi se napravio uzorak realnog sistema potrebno je apstrakcijom doći do njegovih bitnih elemenata koji se uključuju u model. Cilj generisanja vještačkog modela je simuliranje okolnosti u kojima model funkcioniše što je resursno zahtjevan posao, naročito po pitanju vremena, računarskih i intelektualnih kapaciteta. Ovakvom tehnikom se rješavaju problemi kompleksnih sistema gdje nije moguće doći do lakših rješenja. Modelovanje i simulacija omogućavaju izvođenje eksperimenata koji ne bi bili mogući kad bi se koristile uobičajene tehnike kao što je npr. analitičko donošenje zaključaka rješavanjem niza matematičkih jednačina. Na primjer, računarske simulacije su se koristile kako bi se došlo do mnogih napredaka u naučnim i poljima ljudskog djelovanja kao što su astronomija, biologija, fizika, meteorologija, vojna taktika i sl. Bez simulacije, razumijevanje pojava iz ovih oblasti i njihovo predviđanje ili dalje istraživanje ne bi bilo moguće bez dostupnosti realnim sistemima u dužem vremenu. Simulacija modela prevazilazi ove probleme što je dalo ‘vjetar u leđa’ mnogim uspješnim eksperimentima koji su doveli do novih naučno-tehničko-organizacionih dostignuća u današnjem kompleksnom svijetu koji se sve više usložnjava.

Iako su modelovanje i simulacije pogodne tehnike za rješavanje mnogih problema, njihova neodgovarajuća primjena može da vodi do pogrešnih zaključaka. Model koji ima loše ulaze imaće zasigurno i loše izlaze bez obzira na to šta se ‘unutra’ dešava. Međutim tu nema fiksnih pravila jer bez obzira na sav matematičko-statističko-logistički aparat, ova oblast je ipak rukovođena ljudskim sudom koji nije nužno i uvijek objektivan.

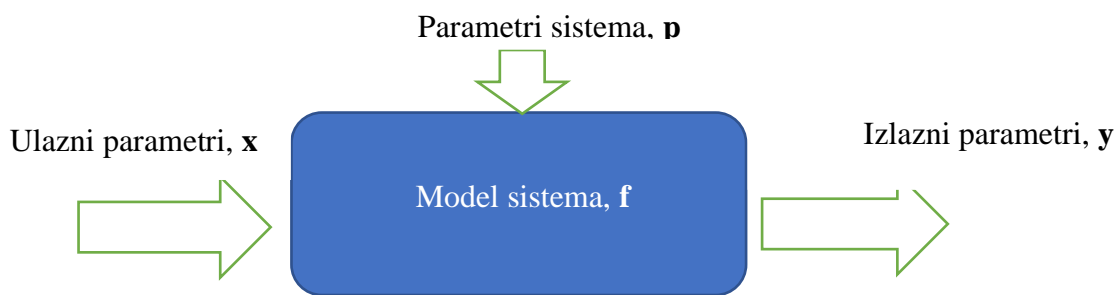
Istorija modelovanja je duga i zavisno od procjena iznosi od više stotina godina pa čak seže i do antičkih vremena. Ipak, za ovu diskusiju je relevantna novija istorija koja podrazumijeva upotrebu digitalnih računarskih naprava kakve su se počele pojavljivati tokom 2. Svjetskog rata. Stručnjaci i pioniri nauke kao što su Von Neumann, Grace Hopper, Alan Turing, Claude Shannon¹ su sredinom 20. vijeka udarili temelje osnovnim postulatima računarske nauke koja će vrlo brzo da dobije praktičnu primjenu upravo u modelovanju objekata i procesa iz stvarnog svijeta i njihovom simuliranju. Može se tvrditi da je naglim razvojem tehnike moguće izvoditi sve kompleksnije eksperimente i modelovati stvarne procese na sve bolji način što je dovelo do značajnih tehničko-tehnoloških otkrića.

¹ Pored naučnika iz zapadne hemisfere, potrebno je navesti i stručnjake koji su ostavili trag a potiču iz drugih zemalja kao što je npr. Rusija: Boris Babaian, Nikolay Brusentsov, Victor Glushkov i mnogi drugi.

2.1. Odnosi sistema i modela

Svrha modeliranja i izvođenja simulacija nad njima je proučavanje sistema. Sistem se može definisati kao skup objekata koji čine jednu funkcionalnu cjelinu ili jedan proces. Da bi se moglo eksperimentisati sa sistemom, stvara se njegova približna, apstrahirana slika zvana model. Fizički modeli prikazuju sisteme iz realnog života koji imaju svoju pojavnu, materijalnu stranu, dok matematički modeli reprezentuju sisteme kao skup računskih ili logičkih poveznica. Modeli mogu biti statički, time predstavljajući sistem u datom vremenskom momentu, ili dinamički, prateći promjene na sistemu u funkciji vremena. Druga važna podjela je na diskretne i kontinualne modele, prema stanju i broju promjena varijabli: ukoliko se varijable konstantno mijenjaju, odnosno neograničeno puno, riječ je o kontinualnim modelima, i obrnuto. Ako je prisutna barem jedna nasumična varijabla, model je stohastičan, u suprotnom govorimo o stohastičkim modelima. Nekada je sistem dovoljno jednostavan da bi se mogao u potpunosti opisati matematički i riješiti analitički, kada to nije slučaj, do rješenja se dolazi simuliranjem ulaznih varijabli i dobijanjem adekvatnih numeričkih rezultata koji nisu potpuno tačni već ih prihvatamo sa određenim stepenom vjerovatnoće.

Generalno uzevši, sistemi se modelišu na način da se uzimaju u obzir ulazne varijable x_1, x_2, \dots, x_r , te izlazne varijable y_1, y_2, \dots, y_s , te parametri sistema, p_1, p_2, \dots, p_t . Iako je svaki sistem jedinstven, ulazni parametri su često nepredvidivi, a parametri sistema su obično proizvod želje za onim izlaznim rezultatima kakve želimo da dobijemo. Stoga, ulazne varijable su obično modelovane kao stohastički, nasumični procesi, a sistemski parametri kao uslovi koje možemo podešavati. Na primjer, kod modela prijema poštanskih pošiljki, ulazni parametri bi mogli uključivati vrijeme dostave pošiljke i veličinu svake pošiljke; sistemski parametri bi mogli biti protokoli skladištenja pošiljki, broj dostavnih kanala i broj prijemnih stanica. Izlaz ovakvog modela bi mogao da obuhvata vremena kašnjenja u dostavi određene vrste poštanskih pošiljki. Možemo reći da je sistem često definisan kao funkcija f koja proizvodi izlaze y iz ulaza x pri definisanim parametrima sistema p , odnosno važi $y=f(x, p)$.



Prilog 1: Funkcionalni pogled na model sistema $y=f(x, p)$

2.2. Primjena

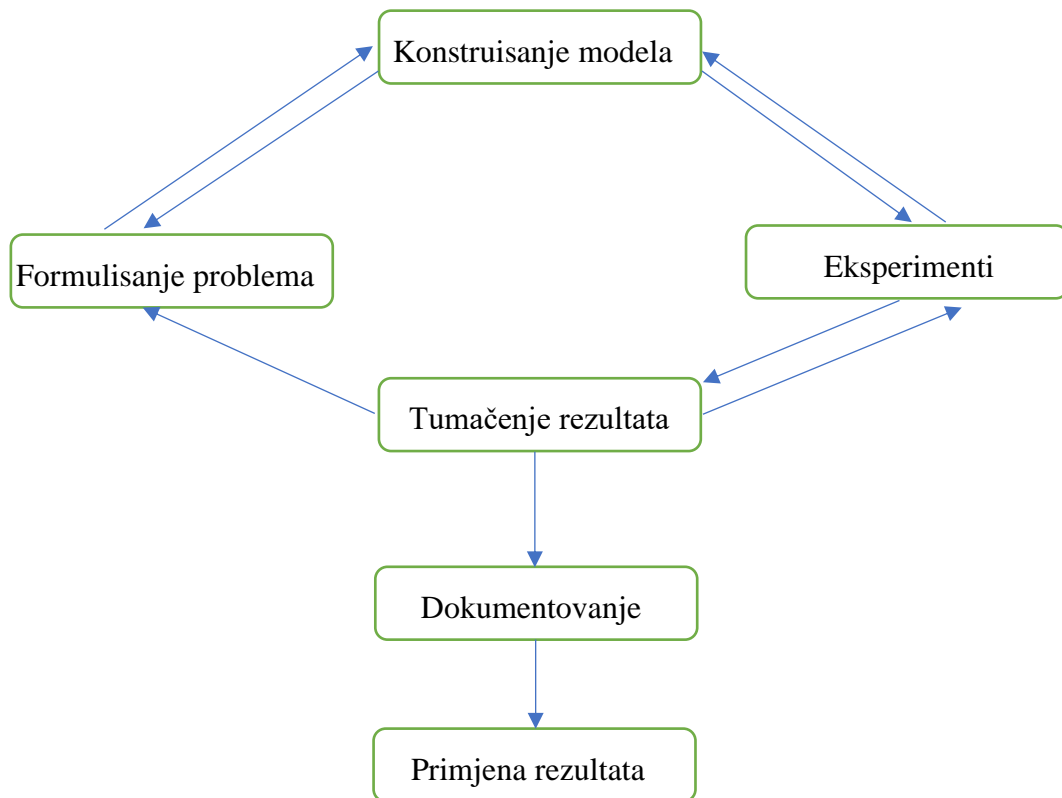
Modelovanje i simulacije se mogu primjenjivati u četiri aplikativne kategorije: predviđanje uspješnosti, modifikacije, poređenja i optimizacija. Predviđanja uspješnosti se odnose na mogućnost da se uvidi kako će se u budućnosti odvijati dizajnirani sistem. Na primjer, prije otvaranja nekog prodajnog salona ili lanca prodaje, u skladu sa određenim ulaznim podacima moguće je predviđati tražnju kupaca i njihove reakcije u skladu sa prodajnim poslovnim politikama. Predviđanje je smisleno i svrsishodno iz primarnog razloga što štedi resurse, odnosno moguće je donositi zaključke na osnovu simuliranog modela što je znatno jeftinije nego empirijskim istraživanjem na stvarnom modelu. Pored toga, simulirani model omogućava unošenje takvih graničnih ulaznih podataka koji ne bi bili mogući na stvarnom modelu.

Modifikacije se primjenjuju na postojećim sistemima kako bi se podesile performanse sistema pod određenim uslovima ili kod promjene načina funkcionisanja sistema. Na primjer, kod vojnih sistema je potrebno procijeniti njihove performanse kod pojave novih tehnoloških dostignuća koji mogu biti prijetnja, i u skladu sa tim modifikovati sistem, odnosno podesiti ga tako da odgovori na nove prijetnje na najbolji mogući način. Analogno prvoj kategoriji, modifikovanje sistema je puno svrsishodnije uraditi na simuliranom nego na realnom sistemu.

Poređenja se odnose na sisteme koji međusobno konkurišu jedni drugima. Na primjer, dva različita sistema za upravljanje rizicima u osiguranju mogu se uporediti u smislu njihovih performansi u uslovima simuliranih uslova poslovanja. Moguće je porediti i sisteme koji u realnom svijetu ne bi bili direktno uporedivi, na primjer poređenje poslovnih rezultata ulaganja u različite branše, proizvodnju i trgovinu.

Konačno, optimizacija se koristi da se utvrde najbolji uslovi za funkcionisanje datog sistema. Na primjer, kod proizvodnih sistema se optimizacionim simulacijama može doći do takvog funkcionisanja sistema koji proizvodi najmanje škarta sa najvećom efikasnosti. Iako je optimizacija širok pojam i obuhvata neke druge naučne discipline, značaj simuliranja je taj da se na relativno jednostavan način simulirani model može promijeniti u smislu manipulacije ulaznim podacima i na osnovu toga donositi zaključke, što sa analitičkim sistemima nije moguće.

2.3. Rješavanje problema korištenjem modelovanja i simulacija



Prilog 2: Proces rješavanja problema korištenjem modelovanja i simulacija

Kao što se vidi iz Priloga 2, rješavanje problema korištenjem modelovanja i simulacija se sastoji iz sljedećih koraka:

1. Formulisanje problema i modela. Piše se opis problema, parametri modela i vrši se odabir tehnike za rješavanje (u ovom slučaju, modelovanje i simulacija).
2. Kreiranje modela sistema. Potom, kreiranje logičkog i kompjuterizovanog modela i ulaznih podataka.
3. Simulacioni eksperimenti. Kreiranje eksperimenata i prikupljanje podataka.
4. Tumačenje rezultata. Statistička analiza izlaznih podataka i procjena njihove relevantnosti sa ciljevima istraživanja.
5. Dokumentovanje. Pisanje opisa analize i svih glavnih koraka u procesu rješavanja problema.
6. Primjena rezultata. Primjena zaključaka do kojih se došlo tokom istraživanja.
7. Validacija modela. Tokom svakog koraka, procijeniti ispravnost i tačnost analize. Validacija znači podešavanje procesa istraživanja kako bi se dobili podaci sa visokom tačnošću i pouzdanošću.

3 FORMULISANJE PROBLEMA I MODELA

Formulisanje problema i modela je korak u rješavanju problema kojem se često ne posvećuje dovoljna pažnja. Razlog tome može biti žurba da se problem riješi na što brži način, pa ako preskakanje analize postane pravilo, moguće je da stručnjaci koji se bave ovom tematikom tokom vremena izgube važne vještine kao što su rangiranje problema, uproštavanje njihovih dijelova i prepoznavanje značaja parametara modela.

Kako bi se problem formulisao, uobičajeno je početi sa pisanjem skice problema. Ova skica opisuje sistem koji je predmet posmatranja ili izučavanja, te navođenje ulaznih, sistemskih i izlaznih parametara. Skiciranje problema počinje sa analizom sistema što može biti zahtjevan posao budući da složeni sistemi sadrže mnoštvo isprepletenih sadržaja. Jedan od načina sa kojim se može boriti sa složenošću je stvaranje podsistema koji su hijerarhijski povezani. Tako jedan sistem može da bude razložen na manje podsisteme rangirajući ih od najvišeg do najnižeg nivoa. Svaki tako opisan podsistem sadrži svoje ulazne i izlazne podatke te opisno izraženo njihovu obradu, kao i načine povezanosti sa drugim podsistemima. Da bi se došlo do ovakve kategorizacije sistema koji se analizira, potrebno je izučavati dokumentaciju sistema, kroz razgovor sa drugim stručnjacima doći do relevantnih podataka, posmatrati sistem u dužem vremenu, koristeći složenu opremu (hardver i/ili softver). Skica problema također može da uključi listu ciljeva studije, zahtjeve ili neke pretpostavke. Na primjer, lista ciljeva kod studije odvijanja saobraćaja na nekoj putnoj dionici² može uključivati optimizaciju saobraćaja u smislu najmanje provedenog vremena, najmanjeg broja osoblja koje održava dionicu i tome slično. Ciljevi istraživanja trebaju biti istaknuti što je jasnije moguće, jer nejasna pitanja zamagljuje rezultate i onemogućuju njihovo smisleno tumačenje. Zahtjevi postavljeni prije i tokom istraživanja obično uključuju stavke vezano za efikasnost i načine funkcionisanja sistema. Pretpostavke koje se tiču funkcionisanja sistema su obično u funkciji pojednostavljivanja modelovanja sistema i one se donose u skladu sa prethodno obavljenom analizom sistema.

Kada je u pitanju modelovanje, kako je već ranije navedeno, to je konstruisanje apstraktne reprezentacije nekog sistema. Tako je prvo potrebno koji su elementi sistema bitni i koji se moraju uključiti u model. To je vrlo bitno jer ako se ovakvi podaci izostave model neće reprezentovati sistem na zadovoljavajući način, a ako pored ključnih parametara dodaju i drugi, koji nisu bitni za funkcionisanje modela, dobiće se složen model koji će trošiti više resursa nego što je potrebno.

² Saobraćajni problemi su vrlo često rješavani modeliranjem i simulacijom.

Budući da je svaki sistem jedinstven, nije moguće dati tačno uputstvo kako modelovati ali je moguće dati neke smjernice. Prije svega, osoba koja modeluje sistem treba da se upozna sa osnovnim postulatima modelovanja putem adekvatne literature. Potom, potrebno je korištenje kako logičkih pretpostavki, tako i iracionalnog, imaginativnog razmišljanja. Također, preporuka je početno kreiranje jednostavnijeg modela i po potrebi njegovo usložnjavanje dodavanjem podsistema tokom kasnijeg istraživanja.

4 KREIRANJE MODELA SISTEMA

Nakon početnih odluka o tipu problema, sadržaju sistema i njegovoj analizi, fokus istraživanja se prebacuje na stvaranje modela sistema koji se istražuje. Poželjno je krenuti od stvaranja konceptualnog modela, koji predstavlja strukturisani opis sistema i daje odgovore na pitanja kako sistem funkcioniše i kako je organizovan. Konceptualni model se može razvijati korištenjem bilo kojih raspoloživih tehnika, kao što su dijagrami toka, opisni tekst, pseudokod, tabele odlučivanja, ili bilo koje druge raspoložive tehnike.

Kompjuterizovani model se stvara prema prihvaćenim pravilima softverskog inženjeringa. To podrazumijeva primjenu idućih koncepata:

- efikasnost,
- fleksibilnost,
- integritet,
- ispravnost,
- konzistentnost,
- mogućnost testiranja.
- održivost,
- ponovna iskoristivost,
- portabilnost,
- pouzdanost,
- proširivost (skalabilnost),
- robusnost,
- sigurnost.

Kada je u pitanju izbor programskog jezika, tradicionalni programski jezici koji se najčešće koriste u svrhe modeliranja i simulacija su C++³ a u novije doba i neki drugi kao što su Python i Java. U pitanju su provjereni programski jezici, sa obimnom dokumentacijom, međutim njihovo korištenje može da bude teško i komplikovano. Stoga postoje specifični simulacioni jezici koji su prilagodljiviji i podesniji za ove namjene od opštih programskih jezika. Simulacione jezike možemo podijeliti u dva tipa: opšti simulacioni jezici (GPSS, SIMSCRIPT, SLAM), te specifični simulacioni jezici (AutoMod, SIMFACTORY). Simulacioni jezici su tako podešeni da isporučuju unaprijed podešene funkcionalnosti kao što

³ Vjerovatno bi trebalo navesti i FORTRAN i MATLAB.

su generisanje nasumičnih brojeva, statistički alati i slično. Neki simulacioni jezici također omogućavaju animaciju podataka. Njihovi ugrađeni alati mogu smanjiti vrijeme razvoja, poboljšati kvalitet softvera i povisiti stepen pouzdanosti modela. Također, jedan od izbora softvera za modeliranje mogu biti i proračunske tabele.

Prilikom modelovanja ulaznih podataka treba voditi računa o stohastičkim, nasumičnim parametrima sistema, te o njihovoj obradi u skladu sa specifikacijama modela. Tako, prema ulaznim podacima, simulacioni modeli mogu da budu kategorisani prema unosu podataka koji je istovjetan sa funkcionisanjem stvarnog sistema, te na podatke koji su dobijeni posmatranjem statističkih distribucija u slučaju da nije moguće kvantifikovati ulazne podatke. Distribucije koje se često koriste u ovu svrhu su normalna distribucija te normalne distribucije. Proces korištenja statističkih distribucija u modelovanju se sastoji od izabira tipa distribucije, procjene vrijednosti za sve parametre modela i ispitivanje takve distribucije u smislu verifikovanja izlaznih rezultata.

Jedan od ključnih zadataka prilikom generisanja modela je generisanje nasumičnih, slučajnih brojeva. Sam termin slučajni broj se obično odnosi na slučajnu varijablu uniformno distribuisanu na intervalu od 0,1. Teško je definisati šta je to slučajnost, a na ovom mjestu će se pomenuti dva načina generisanja ovakvih brojeva: linearni kongruentalni generatori ili višestruke rekurzije. Linearni kongruentalni generatori imaju sljedeću formu:

$$x_n = (\alpha x_{n-1} + \beta) \pmod{m}$$

Formula 1: Uopšteni oblik linearnog kongruentalnog modela

X_n je $(n+1)$ ti pseudoslučajni broj u nizu, x_0 je prvi član niza, α i β su također konstante i x_n vrijednosti dijeljene sa m koje generišu vrijednosti u intervalu $(0,1)$. Kod višestrukih rekurzivnih generatora su izostavljena sabiranja stavljanjem da je $\beta=0$. Dužina ovog ciklusa se naziva period generatora. Kada je period jednak modulu m , generator ima puni period. Puni period je jednak najvećem cijelom broju koji je moguće prikazati, koji je obično 2^b , gdje je b broj bita korišten da se smjesti ili predstavi numerički broj (obično 31, čime se maksimalni period dobija od preko 2,1 milijarde).

Dobar pseudogenerator brojeva je računarski brz, zahtjeva malo memorije i ima veliki period. Dobri generatori bi trebali generisati takve brojeve koji prolaze određene testove kada su upoređeni sa uniformnim distribucijama. Također, ne bi trebali da budu u korelaciji jedno sa drugima.

5 SIMULACIONI EKSPERIMENTI

Kada je kompjuterizovani model kreiran, potrebno je izvršiti eksperimente u cilju skupljanja podataka koji su u skladu sa ciljem istraživanja, odnosno rješavanja postavljenog problema. Svrha eksperimenata je proučavanje generisanog modela i njegovog funkcionisanja te analiziranje dobijenih rezultata u smislu usklađenosti sa ciljevima istraživanja. Ovo je bitno uraditi prije donošenja zaključaka o modelu i implementacije njegovih rezultata. U ove svrhe, pored drugih tehnika, upotrebljava se i analiza osjetljivosti i odnosi se na generisanje sljedećih odnosa nakon višestrukog izvršavanja simulacije prilikom korištenja različitih parametara: $\Delta\hat{y}/\Delta p$ i $\Delta\hat{y}/\Delta\theta$, gdje \hat{y} predstavlja aritmetičku sredinu ili varijansu izlazne varijable, a p i θ predstavljaju parametre sistema i distribucije. Veličina ovih odnosa predstavlja pokazatelj osjetljivosti izlaznih podataka.

6 TUMAČENJE REZULTATA

Kada je generisani model kreiran sa nasumičnim varijablama, njegov izlaz takođe ima slučajni karakter. Izlazni podaci bi stoga trebali da budu procenjeni sa stanovišta aritmetičke sredine i varijanse. Pri tome, neki sistemi imaju određena vremena početka i kraja. Na primjer, fabrika koja se otvara u 07:00 časova i zatvara u 17:00 časova bi bila modelirana kao jedan takav, vremenski ograničeni sistem. Analitičar skuplja podatke u ovom intervalu i dalje ih analizira. Drugi sistemi nemaju vremena početka i kraja i zbog njihove kontinualne prirode, izlazni podaci uzimaju jedan određen set vrijednosti koji zavisi od fluktuacija u stanju takvog sistema. Prilikom tumačenja rezultata u bilo kojem tipu sistema, treba imati u vidu centralnu graničnu teoremu koja kaže da normirana i centrirana suma velikog broja nezavisnih i identično raspoređenih slučajnih promjenljivih teži normalnom rasporedu vjerovatnoće. Budući da je nasumični karakter ulaznih podataka veoma često pravilo (izuzev kod potpuno determinističkih sistema), izlazni podaci će takođe biti nasumične varijable, pa prilikom višestrukog ponavljanja simulacija će se dobiti distribucije izlaznih podataka koje će pratiti normalnu statističku distribuciju. Što je veći broj uzorkovanja, odnosno ponavljanja simulacije, dobiće se finiji izlazni set podataka iz koga će biti očiglednije utvrditi međusobne zavisnosti i raspored distribucije, te zaključivati na osnovu tako utvrđenih zakonitosti.

7 DOKUMENTOVANJE

Dokumentovanje bi trebao biti obavezan korak u modelovanju i simulaciji. Ako se sam proces od početka do kraja ne dokumentuje, protokom vremena će sigurno biti poteškoća u ponovnom korištenju, tumačenju rezultata, proširivanju i modifikovanju modela. U slučaju promjene osoblja, hardvera ili softvera, korištenje modela u dužem vremenskom periodu će postati vrlo problematično ako nije dostupna detaljna dokumentacija iz koje se svaki bitan korak može razumjeti i ponovno kreirati. Samo kreiranje dokumentacije je ozbiljan i vremenski zahtjevan proces koji se obavlja tokom svih faza modelovanja i to je vrlo često postupak koga izbjegavaju modeling stručnjaci što kasnije dovodi do raznih problema. Dobra dokumentacija bi trebala da sadrži sljedeće:

- opis ciljeva istraživanja, pretpostavki, zahtjeva i ograničenja,
- opis konceptualnog modela,
- analizu u vezi odabira i tipa ulaznih podataka,
- analizu eksperimenata kompjuterizovanog modela,
- analizu izlaznih rezultata,
- zaključke,
- preporuke.

8 IMPLEMENTACIJA REZULTATA

Implementacija rezultata podrazumijeva uspješno modelovanje i studiju simulacije. Ako se rezultati ne mogu implementirati to je pokazatelj da je došlo do ozbiljne greške u prethodnom procesu i poruka istraživaču da je potrebno da se vrati unatrag, detektuje i ispravi grešku. Sama implementacija rezultata se ne tiče samog procesa modelovanja i simulacije već zavisi od eksternih faktora, načina funkcionisanja stvarnog sistema i odgovornih lica za njega. Naravno, znatne razlike između predviđenih, odnosno dobijenih simulacionih rezultata i stvarnih podataka sistema su znak da su dijelovi realnog sistema ili pogrešno modelovani ili izostavljeni iz modela, pa shodno okolnostima, istraživač može da se vrati i revidira neke od ranijih koraka koje je preduzeo u stvaranju modela.

9 VALIDACIJA MODELA

Modelovanje i simulacija uključuju imitaciju postojećeg ili predloženog sistema koji koristi model koji je uprošćena verzija originalnog sistema te stoga problem tačnosti je uvijek prisutan. Validacija obuhvata proces davanja odgovora na pitanja koliko daleko se konceptijski nalazi model od stvarnog, realnog sistema i odnosi se na količinu tačnosti u modelovanju i analizi podataka. Verifikacija označava težnju da se osigura ispravnost kompjuterizovanog modela i njihove implementacije. Akreditacija u smislu prihvatanja rezultata modelovanja znači stepen izvjesnosti u kome se mogu prihvatiti rezultati simulacije kreiranog modela. S tim u vezi, model je moguće analizirati sa stanovišta kreiranja aksioma, empirijskog istraživanja i pozitivne ekonomije. Kreiranje aksioma je racionalni način analiziranja modela, jer su aksiomi nepobitni načini na koje jedan model funkcioniše i iz kojih je logičkim pristupima kao što su dedukcija, analiza i predviđanje moguće izvlačiti zaključke o modelu. Empirijsko, praktično istraživanje koristi praktične podatke do kojih se dolazi simulacijama, a pozitivna ekonomija je fokusirana na uzroke i posljedice ponašanja modela.

Iako ne postoji neko čvrsto definisano pravilo, ispravna validacija modela može da obuhvati sve tri navedene tehnike. Postavlja se pitanje ko treba da izvrši validaciju modela, da li isti stručnjak ili tim stručnjaka odgovornih za model ili neko drugi. Ako validaciju obavljaju iste osobe koje su i modelovale, oni će sigurno kroz ovaj dodatni proces steći nova znanja o modelu i njegovom funkcionisanju. Ako se angažuju nezavisne osobe, to će sigurno biti skuplja varijanta jer će osobe koje nisu kreirale model utrošiti više resursa da ga razumiju i potom pristupe validaciji dobijenih rezultata.

Tehnike validacije obično obuhvataju statističke ili subjektivne. Statističke tehnike se koriste u studijama postojećih sistema iz kojih je moguće kvantifikovati odnosno dobiti numeričke podatke. To su analize varijanse, testovi hipoteza, regresivna analiza, analiza osjetljivosti, upotreba vremenskih serija i tome slično. Subjektivne tehnike su validacija hipoteza, prediktivna validacija, testiranje podsistema modela.

Proces verifikovanja ispravnosti kompjuterski generisanog modela i njegove implementacije ima dvije glavne komponente: testiranje softvera i tehnički pregledi. Tehnički pregledi obuhvataju ispitivanja softverskih zahtjeva, dizajna, programskog koda. Softverski zahtjevi se procjenjuju u skladu sa njihovom mogućnosti da podrže ciljeve modelovanja i simulacije. Softverski dizajn se ispituje preliminarno, u smislu implementacije softverskog rješenja u skladu sa zahtjevima i specifikacijama, te ispitivanja ispravnosti algoritama i

eventualnih grešaka u softveru. Detaljan pregled programskog koda provjerava detalje kao što su manipulacija greškama i ispituje tehničke i formalne detalje napisanog koda.

Softversko testiranje ima za cilj pronalaženje do tada nepoznatih grešaka u softveru i uključuje testiranje u tzv. programskim kutijama. Postoji testiranje putem bijelih i crnih kutija. Testovi putem bijelih kutija provjeravaju da su sve linije programskog koda ispitane i da su sva moguća stanja modela testirana. Crne kutije pokazuju funkcionalnost komponenti softvera putem dizajniranja scenarija koji testiraju veze između ulaznih i izlaznih parametara. Softver za testiranje softvera obično koristi vrlo korisne alate kao što su automatizovano pokretanje testova, interaktivno dibagovanje, grafički i animacioni prikaz podataka itd. Cilj softverskog testiranja u ovom smislu je pronalaženje i izbjegavanje tri tipa grešaka:

- Greške koje dovode do odbacivanja rezultata studije iako su korektno ispitane i prihvatljive.
- Greške koje dovode do prihvatanja rezultata studije koje nisu dovoljno ispitane i prihvatljive.
- Greške do kojih dolazi zbog izostavljanja modelovanja značajnih dijelova realnog sistema.

10 GPSS SIMULACIONI JEZIK

GPSS programski simulacioni jezik (eng. General Purpose Simulation System) je simulacioni jezik za simuliranje diskretnih sistema sa stohastičkom komponentom. Razvijen je u američkom IBM 1960ih godina. Za razliku od opštih programskih jezika, ovo je prilagođeni jezik koji putem jednostavnih i jezički razumljivih naredbi definiše strukturu modela i izvršava simulaciju. Kada se simulacija izvrši, ispisuje se rezultat simulacije sa raznim statističkim pokazateljima.

GPSS je jezik koji je fokusiran na procese. Za predstavljanje modela u jeziku GPSS se koristi dijagram toka koji ima više desetina različitih blokova koji imaju svoje atribute kao što su ime, simbol i namjenu. Proces programiranja se sastoji od biranja blokova i njihovog povezivanja. Blokovi su statički elementi i obuhvataju skladišta, logičke prekidače, redove, itd. Entiteti se dijele na četiri vrste:

- dinamički entiteti,
- statički entiteti,
- statistički entiteti,
- entiteti operacija.

Dinamički entiteti se sastoje iz transakcija koje imaju svoje parametre i spadaju u aktivne objekte jer prolaze kroz blokove po potrebi. Transakcije su u modelima predstavljene kao klijenti u radnjama, vozila na putevima, pisma u poštanskom saobraćaju itd.

Statički entiteti su pasivni objekti kao što su skladišta, uređaji, prekidači itd. Skladište može da posluži više transakcija, logički prekidači mogu biti postavljeni u više stanja, uređaji mogu posluživati samo jednu transakciju u vremenskoj jedinici.

Statistički entiteti obuhvataju redove i tabele. Redovi su grupisanja transakcija zbog nemogućnosti opsluživanja, na primjer kada klijenti čekaju u redu u trgovini. Tabela sadrži numeričke vrijednosti nekog parametra poredane po frekvenciji pojavljivanja. Simulacija podrazumijeva dobijanje informacija o statističkim entitetima kao npr. vrijeme provedeno u redu.

Entiteti operacija daju instrukcije transakcijama o daljem nastupanju i kretanju i nazivaju se blokovi. Blokovi su definisani tipom operacije i međusobno povezani. Blok dijagram je model sistema i unutar njega mogu da postoje četiri tipa događaja:

- kreiranje ili uništavanje transakcija,
- promjena numeričkih atributa entiteta,
- vremensko kašnjenje transakcija,
- promjena toka transakcije.

U GPSS jeziku transakcije se kreiraju naredbom GENERATE. Transakcije se kreću kroz blokove sve dok ne naiđu na takve blokove koji ne mogu da je prime ili dok ne naiđe na naredbu TERMINATE koja poništava transakciju. Transakcije se čuvaju u listama tekućih događaja i listi budućih događaja. Liste tekućih događaja se odnose na one transakcije koje se kreću kroz blok dijagram a transakcije iz liste budućih događaja čekaju na uslov koji im onda omogućuje kretanje i tada se prebacuju u listu tekućih događaja. Simulacija ima svoje tačno određeno interno vrijeme i svaka transakcija se kreće sve dok se kretanje ne prekine uništavanjem, odbijanjem ili zadržavanjem transakcije. Sve promjene transakcija i njihovog prioriteta uzrokuju promjene njihovog statusa u dvjema, naprijed navedenim listama.

Postoje tri tipa naredbi u jeziku GPSS. To su deklaracione naredbe, blok naredbe i kontrolne naredbe. Deklaracione naredbe definišu attribute skladišta, uređaja, tabela i drugih stalnih entiteta. Razlikuju se polje lokacije, gdje se definiše određeno npr. skladište, a polje varijable se koristi za upotrebu raznih argumenata. Blok naredbe definišu blokove programa koje mogu da sadrže argumente. Kontrolne naredbe kontrolišu tok simulacije.

Neke od osnovnih naredbi jezika GPSS su sljedeće:

Opis naredbe	Sintaksa i primjer	Opis sintakse
Generisanje transakcija	GENERATE A,[B],[C],[D],[E] GENERATE 10, 8	A – srednje vrijeme između transakcija B – modifikator C – vrijeme generisanja prve transakcije D – ukupan broj transakcija E – prioritet transakcije
Vremensko zadržavanje transakcije	ADVANCE A, [B] ADVANCE 5	A – srednje vrijeme zadržavanja B – modifikator srednje vrijednosti zadržavanja
Statističko grananje transakcije	TRANSFER .A, [B],[C] TRANSFER .1,,ODB	A – vjerovatnoća skoka na blok B, C – blokovi u koje odlazi transakcija poslije grananja
Bezuslovni skok	TRANSFER, B	B – blok na koji transakcija odlazi bezuslovno

Uklanjanje transakcije	TERMINATE [A]	A – vrijednost za koju se umanjuje tajmer kada transakcija dođe na ovaj blok
Start simulacije	START A,[B]	A – početna vrijednost terminacionog brojača B – koristi se za ukidanje štampanja rezultata na kraju simulacije
Označavanje početka i kraja programa	SIMULATE END	-
Zaposjedanje / oslobađanje uređaja	SEIZE A RELEASE A	A – broj/ime uređaja
Ulazak/napuštanje skladišta	ENTER A LEAVE A	-
Ulazak/izlazak iz reda	QUEUE A DEPART A	-

Prilog 3: Tabelarni pregled osnovnih GPSS naredbi

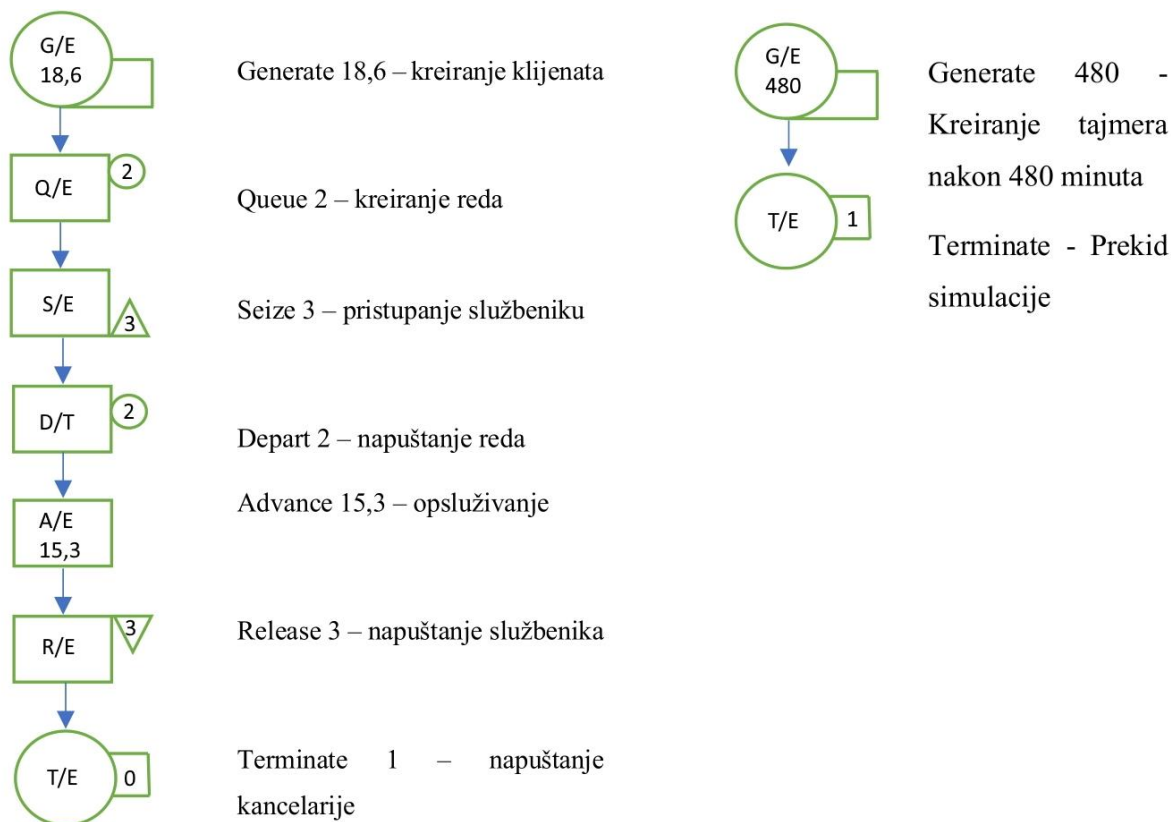
11 SIMULACIJA ŠALTERSKOG SLUŽBENIKA

Da bi se primjerom pokazala praktična upotrebu modelovanja u GPSS programskom jeziku, biće predstavljena simulacija rada šalterskog službenika pod sljedećim pretpostavkama:

- kancelarija sadrži jednog službenika sa osmosatnim radnim vremenom,
- klijenti dolaze na prosječno svakih 18 minuta, dok vrijeme dolaska varira između 12 i 24 minute,
- ako je službenik zauzet, klijenti čekaju u redu,
- kada službenik završi sa opsluživanjem trenutnog klijenta, idući dolazi na red,
- opsluživanje traje između 12 i 18 minuta, sa prosjekom od 15 minuta,
- kada je opsluživanje završeno, klijent napušta kancelariju.

Ciljevi simulacije su dobijanje odgovora na iduća pitanja:

- koliko je šalterski službenik zauzet tokom radnog dana,
- koliko traje red koji se eventualno stvara,
- koliko prosječno klijenti čekaju na uslužnje.



Prilog 4: Blok dijagram modela

Programski kôd u GPSS izgleda ovako:

SIMULATE			
GENERATE	18,6		KREIRANJE KLIJENATA
QUEUE	2		KREIRANJE REDA AKO JE POTREBNO
SEIZE	3		PRISTUPANJE SLUŽBENIKU
DEPART	2		KLIJENT NAPUŠTA RED
ADVANCE	15,3		KLIJENT SE OPSLUŽUJE
RELEASE	3		KRAJ OPSLUŽIVANJA
TERMINATE	0		NAPUŠTANJE KANCELARIJE
GENERATE	480		GENERISANJE TAJMERA OD 480 MINUTA SIMULIRANOG VREMENA
TERMINATE	1		ISKLUČIVANJE SIMULACIJE
START	1		POČETAK SIMULACIJE
END			POVRATAK KONTROLE OPERATIVNOM SISTEMU

Tumačenje kôda je sljedeće:

Prva linija, `SIMULATE`, označava početak programa.

Linija `GENERATE 18,6` generiše transakciju – klijenta – svakih 18 minuta \pm 6 minuta.

`QUEUE 2` definiše red čekanja sa ID 2, označavajući red gdje će klijenti da čekaju na opsluženje kod službenika.

`SEIZE 3` definiše skladište sa ID 3. Skladište je šalterski službenik i ova linija znači da ako je službenik slobodan, idući klijent ga zauzima sve do završetka opsluživanja.

`DEPART 2` znači da klijent napušta red kada je opsluživan od službenika.

Linija `ADVANCE 15,3` znači da transakcije u ovom stanju će se pomjerati naprijed nakon 15 minuta \pm 3 minute što je vrijeme potrebno za opsluživanje. Nakon toga, `RELEASE 3` označava da klijent više ne zauzima vrijeme kod službenika, odnosno da više nije opsluživan i linija `TERMINATE 0` završava transakciju, sa značenjem da je klijent napustio kancelariju.

Ovakav programski kôd bi se izvršavao neograničeno dugo, odnosno ne bi se nikad zaustavio, pa zato dodajemo novu transakciju linijom `GENERATE 480`, koja određuje simulirano vrijeme izvršavanja od 480 minuta odnosno 8 sati, a linija nakon nje `TERMINATE 1` zaustavlja simulaciju nakon kreiranja ove transakcije. Zadnje dvije linije kôda, `START 1` i `END` otpočinju i završavaju simulaciju.

Kada se ovaj programski kôd unese u razvojno okruženje GPSS World Student Version 5.2.2., izgled programa je sljedeći:

```

SIMULATE
GENERATE 18,6 ;KREIRAJ KLIJENTE
QUEUE 2 ;KREIRANJE REDA AKO JE POTREBNO
SEIZE 3 ;PRISTUPANJE SLUŽBENIKU
DEPART 2 ;KLIJENT NAPUŠTA RED
ADVANCE 15,3 ;KLIJENT SE OPSLUŽUJE
RELEASE 3 ;KRAJ OPSLUŽIVANJA
TERMINATE 0 ;NAPUŠTANJE KANCELARIJE
GENERATE 480 ;SIMULIRANO VRIJEME OD 480 MINUTA
TERMINATE 1 ;ISKLJUČIVANJE SIMULACIJE
START 1 ;POČETAK SIMULACIJE
END ;KRAJ SIMULACIJE
    
```

Prilog 5: Programski kôd modela

GPSS ispisuje sljedeće rezultate po završetku simulacije:

GPSS World Simulation Report - Model seminarski.12.1

Friday, July 08, 2022 13:27:38

START TIME		END TIME		BLOCKS	FACILITIES	STORAGES			
0.000		480.000		9	1	0			
LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT	COUNT	RETRY			
	1	GENERATE	26		0	0			
	2	QUEUE	26		0	0			
	3	SEIZE	26		0	0			
	4	DEPART	26		0	0			
	5	ADVANCE	26		1	0			
	6	RELEASE	25		0	0			
	7	TERMINATE	25		0	0			
	8	GENERATE	1		0	0			
	9	TERMINATE	1		0	0			
FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
3	26	0.810	14.950	1	27	0	0	0	0
QUEUE	MAX CONT.	ENTRY	ENTRY (0)	AVE. CONT.	AVE. TIME	AVE. (-0)		RETRY	
2	1	0	26	19	0.038	0.710		2.636	0
FEC	XN	PRI	BDT	ASSEM	CURRENT	NEXT	PARAMETER	VALUE	
27	0		481.836	27	5	6			
28	0		482.707	28	0	1			
29	0		960.000	29	0	8			

GPSS World - [Model seminarski.12.1 - REPORT]

File Edit Search View Command Window Help

GPSS World Simulation Report - Model seminarski.12.1

Friday, July 08, 2022 13:27:38

START TIME	END TIME	BLOCKS	FACILITIES	STORAGES
0.000	480.000	9	1	0

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT COUNT	RETRY
1		GENERATE	26	0	0
2		QUEUE	26	0	0
3		SEIZE	26	0	0
4		DEPART	26	0	0
5		ADVANCE	26	1	0
6		RELEASE	25	0	0
7		TERMINATE	25	0	0
8		GENERATE	1	0	0
9		TERMINATE	1	0	0

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
3	26	0.810	14.950	1	27	0	0	0	0

QUEUE	MAX CONT.	ENTRY	ENTRY(0)	AVE.CONT.	AVE.TIME	AVE. (-0)	RETRY
2	1	0	26	19	0.038	0.710	2.636 0

FEC	XN	PRI	BDT	ASSEM	CURRENT	NEXT	PARAMETER	VALUE
27	0		481.836	27	5	6		
28	0		482.707	28	0	1		
29	0		960.000	29	0	8		

Prilog 6: Ispis rezultata simulacije u GPSS

Programski kôd predstavljenog modela se sastoji iz samih komandi te komentara koji su naznačeni sa znakom ;. GPSS funkcionira putem vremenskih jedinica a ne putem apsolutnog vremena; programi se čitaju od vrha do dna i sa lijeve ka desnoj strani; simulator pokreće virtuelni časovnik i shodno njemu izvršava naredbe onako kako su definisane u tom vremenskom momentu.

Linija `SIMULATE` označava početak programa.

`GENERATE 18, 6` znači generisati transakciju svakih 18 minuta plus/minus 6 minuta.

`QUEUE 2` definiše red sa oznakom ID 2, označavajući red u kojem će klijenti čekati.

SEIZE 3 definiše skladište sa oznakom ID 3. Skladište je šalterski službenik i ova linija kaže ako je službenik slobodan, iduću klijent ga zauzima sve do završetka transakcije odnosno posla koji šalterski službenik obavlja (šta tačno službenik radi je za ovaj program nebitno).

DEPART 2 određuje da klijent napušta red po završetku rada službenika.

ADVANCE 15,3 znači da transakcija u ovom stanju se pomjera nakon 15 minuta plus/minus 3 minute – što je potrebno vrijeme za obavljanje radnje radi koje je klijent došao kod šalterskog službenika. Nakon toga RELEASE 3 pokazuje da klijent ne zauzima više službenika i TERMINATE 0 završava transakciju, pokazujući da je klijent napustio kancelariju.

Ovom kôdu treba još dodati tajmer jer ako to ne učinimo program bi se izvršavao beskonačno dugo. Simuliramo vrijeme od 8 sati, odnosno 480 minuta, naredbom GENERATE 480. Nakon toga, naredba TERMINATE 1 zaustavlja simulaciju te zadnje dvije linije programskog kôda pokreću simulator i završavaju izvršavanje programa.

START TIME		END TIME	BLOCKS	FACILITIES	STORAGES				
0.000	Vrijeme	480.000	Broj 9	Broj 1	0				
			blokova	skladišta					
LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT	COUNT	RETRY			
	1	GENERATE	26		0	0			
	2	QUEUE	26		0	0			
	3	SEIZE	26		0	0			
Oznaka bloka	4	DEPART	26		0	0			
	5	ADVANCE	26	1	0	0			
	6	RELEASE	25	0	0	0			
	7	TERMINATE	25	0	0	0			
	8	GENERATE	1	0	0	0			
	9	TERMINATE	Broj 1	0	0	0			
		Tip bloka	transakcija						
FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
3	26	0.810	14.950	1	27	0	0	0	0
		Prosječno	Prosječno						
		iskorištenje	vrijeme transakcije						
QUEUE	MAX	CONT.	ENTRY	ENTRY(0)	AVE. CONT.	AVE. TIME	AVE. (-0) RETRY		
2	1	0	26	19	0.038	0.710	2.636	0	
			Ukupan broj transakcija u redu	Bez čekanja	Prosječno	Prosječno vrijeme čekanja u redu			

Prilog 7: Tumačenje rezultata

Kao što se iz Priloga 4 može vidjeti, simulacija je izvršena od nulte do 480. vremenske jedinice, pri tom je kreirano 9 blokova i jedno skladište. U bloku 1 je generisano 26 transakcija, što znači da je ukupno 26 klijenata ušlo u radnju tokom definisanog vremena, odnosno osmosatnog radnog dana. Blokovi 8 i 9 su imali jednu transakciju što je i očekivano jer su to blokovi koji počinju i završavaju simulator. Blok 5 ima jednu trenutnu transakciju dok ostali nemaju nijednu, to je zato što taj blok označava samo izvršenje transakcije kod šalterskog

službenika odnosno odnosi se na pravilo definisano na početku modela da kod službenika može biti maksimalno jedan klijent. Ukupno je u kancelariju ušlo 26 klijenata koji su prosječno bili opsluživani po 14,95 minuta, odnosno službenik je tokom dana utrošio 6,47 sati efektivno, odnosno njegovo prosječno iskorištenje je 81% (6,47/8). Simulacija daje podatke i o stanju generisanog reda. Vidi se da je u redu maksimalno jedan klijent čekao, da je ukupno 26 klijenata bilo u redu međutim da je 19 klijenata odmah ušlo i izašlo iz reda jer je službenik bio slobodan, odnosno da je 7 klijenata čekalo u redu dok je službenik opsluživao druge klijente. Prosječno vrijeme čekanja u redu je iznosilo 2,6 minuta dok je ukupno vrijeme svih klijenata koji su ušli u red (bez obzira na dalji ishod) 0,71 minuta.

Značaj ovakvog modeliranja datog problema je, u konkretnom slučaju, procjena efikasnosti rada službenika, odnosno njegovih performansi s jedne strane, te sa druge strane kreiranja okruženja u kome se simuliraju neki problemi sa kojima se ovakav službenik može susresti, odnosno u ovom slučaju stvaranje gužve odnosno reda za čekanje klijenata. Rezultati analize simulacije mogu da ukažu da početni parametri nisu adekvatni (na primjer, simulacija bi mogla da se ponovo pokrene sa kraćim vremenom zadržavanja klijenata, ili sa dodavanjem još jednog službenika, i takvi rezultati bi se mogli upoređivati i iz toga izvlačiti adekvatni zaključci).

12 ZAKLJUČAK

U predmetnom radu su date teorijske osnove i uvodna razmatranja koja se tiču modelovanja realnih sistema i simulacija tako dobijenih modela. Navedene su kategorije modela, načini na koje se proces modelovanja odvija u smislu preporuka i smjernica, za svaku od nabrojanih etapa je dato obrazloženje i objašnjenje.

Kroz praktičan primjer je kreiran model rada kancelarijskog službenika, navedeni ulazni parametri i ograničenja, te postupak njegovog rada koji uključuje stvaranje struktura kao što su redovi, te na osnovu toga dobijeni izlazni rezultati koji su potom protumačeni.

Na osnovu naprijed navedenog, može se zaključiti da su modeli i simulacije te simulacioni jezici koristan alat koji se može upotrebljavati za rješavanje problema koji se ne mogu lako ili koji se nikako ne mogu riješiti analitičkim putem. U pitanju je kompleksna ali aktuelna i korisna tematika za rješavanje kompleksnijih problema koji uključuju stohastičke, odnosno nasumične varijable.

LITERATURA

- [1] B. Radenković, M. Stanojević i A. Marković, *Računarska simulacija*, Beograd: FON, 1999.
- [2] S. Robinson, *Simulation: The Practice of Model Development and Use*, Red Globe Press, 2014.
- [3] G. Dubois, *Modeling and Simulation*, CRC Press, 2018.
- [4] M. Weisberg, *Simulation and Similarity*, Oxford University Press, 2015.
- [5] *GPSS World Tutorial Manual*, Minuteman Software, [Na mreži]: http://www.minutemansoftware.com/tutorial/tutorial_manual.htm. [Poslednji pristup 12 07 2022].
- [6] Z. Avramović, *Simulacije i simulacioni jezici*, materijal u e-formi za pripremu ispita, Banja Luka: Panevropski univerzitet „APEIRON”, 2021.

Popis priloga:

Prilog 1: Funkcionalni pogled na model sistema $y=f(x, p)$	3
Prilog 2: Proces rješavanja problema korištenjem modelovanja i simulacija	5
Prilog 3: Tabelarni pregled osnovnih GPSS naredbi	18
Prilog 4: Blok dijagram modela.....	19
Prilog 5: Programski kôd modela	21
Prilog 6: Ispis rezultata simulacije u GPSS	22
Prilog 7: Tumačenje rezultata	23