

# Flexsim Process Flow

---

**Kocijan, Laura**

**Master's thesis / Diplomski rad**

**2022**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Rijeka, Faculty of Maritime Studies, Rijeka / Sveučilište u Rijeci, Pomorski fakultet**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:187:298631>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2024-07-11**



**Sveučilište u Rijeci, Pomorski fakultet**  
University of Rijeka, Faculty of Maritime Studies

*Repository / Repozitorij:*

[Repository of the University of Rijeka, Faculty of Maritime Studies - FMSRI Repository](#)



**uniri** DIGITALNA  
KNJIŽNICA



**SVEUČILIŠTE U RIJECI  
POMORSKI FAKULTET U RIJECI**

**LAURA KOCIJAN**

**FLEXSIM PROCESS FLOW**

**DIPLOMSKI RAD**

Rijeka, 2022.

**SVEUČILIŠTE U RIJECI**  
**POMORSKI FAKULTET U RIJECI**

**FLEXSIM PROCESS FLOW**  
**FLEXSIM PROCESS FLOW**

**DIPLOMSKI RAD**

Kolegij: Modeliranje i simulacije

Mentor: doc. dr. sc. Dario Ogrizović

Komentorica: Valentina Šutalo, mag. ing. logist.

Studentica: Laura Kocijan

Studijski smjer: Logistika i menadžment u pomorstvu i prometu

JMBAG: 0112066424

Rijeka, rujan 2022.

Studentica: Laura Kocijan

Studijski program: Logistika i menadžment u pomorstvu i prometu

JMBAG: 0112066424

## IZJAVA O SAMOSTALNOJ IZRADI DIPLOMSKOG RADA

Kojom izjavljujem da sam diplomski rad s naslovom

FLEXSIM PROCESS FLOW

*(naslov diplomskog rada)*

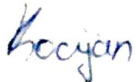
izradio/la samostalno pod mentorstvom doc. dr. sc. Daria Ogrizovića

te komentorstvom Valentine Šutalo, mag. ing. logist

stručnjaka/stručnjakinje iz tvrtke \_\_\_\_\_  
*(naziv tvrtke).*

U radu sam primijenio/la metodologiju izrade stručnog/znanstvenog rada i koristio/la literaturu koja je navedena na kraju diplomskog rada. Tuđe spoznaje, stavove, zaključke, teorije i zakonitosti koje sam izravno ili parafrazirajući naveo/la u diplomskom radu na uobičajen, standardan način citirao/la sam i povezao/la s fusnotama i korištenim bibliografskim jedinicama, te nijedan dio rada ne krši bilo čija autorska prava. Rad je pisan u duhu hrvatskoga jezika.

Student/studentica



\_\_\_\_\_  
*(potpis)*

Laura Kocijan

Studentica: Laura Kocijan

Studijski program: Logistika i menadžment u pomorstvu i prometu

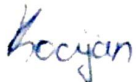
JMBAG: 0112066424

IZJAVA STUDENTA – AUTORA  
O JAVNOJ OBJAVI OBRANJENOG DIPLOMSKOG RADA

Izjavljujem da kao student – autor diplomskog rada dozvoljavam Pomorskom fakultetu Sveučilišta u Rijeci da ga trajno javno objavi i besplatno učini dostupnim javnosti u cjelovitom tekstu u mrežnom digitalnom repozitoriju Pomorskog fakulteta.

U svrhu podržavanja otvorenog pristupa diplomskim radovima trajno objavljenim u javno dostupnom digitalnom repozitoriju Pomorskog fakulteta, ovom izjavom dajem neisključivo imovinsko pravo iskorištavanja bez sadržajnog, vremenskog i prostornog ograničenja mog diplomskog rada kao autorskog djela pod uvjetima *Creative Commons* licencije CC BY Imenovanje, prema opisu dostupnom na <http://creativecommons.org/licenses/>

Student/studentica – autor



---

(potpis)

## **SAŽETAK**

Simulacijsko modeliranje jedan je od vodećih metoda modeliranja uz pomoć računala. Ona omogućuje opis, razumijevanje i kvantitativnu analizu složenih dinamičkih sustava u različitim područjima industrije i znanosti. Nakon što se razvije simulacijski model, taj model treba prevesti u računalni model tj. simulacijski program. Za izradu simulacijskog modela potrebni su odgovarajuće računalne aplikacije i pripadajući alati. U radu je korištena računalna aplikacija za simulaciju diskretnih događaja FlexSim. Aplikacija ima objektno orijentirano okruženje za simulaciju diskretnih događaja koje se lako može koristiti za razvoj simulacijskog modela za vizualizaciju i praćenje aktivnosti dinamičkog toka procesa sustava. Mogućnost izrade simulacije prikazana je kroz dva pristupa, pri čemu se prvi temelji na standardnom 3D modelu, dok se drugi temelji na dodatnom FlexSim alatu koji je integriran unutar same aplikacije, kao što je Process Flow koji je posebno koristan pri simulaciji složenih operacija. Prednost drugog pristupa je u bržem i preciznijem procesu izrade simulacijskog modela.

Ključne riječi: Simulacije, simulacijsko modeliranje, FlexSim, FlexSim Process Flow

## **SUMMARY**

Simulation modeling is one of the leading methods of computer modeling. It enables users to describe, understand and quantitatively analyse complex dynamic systems in different areas of industry and science. After the simulation model is developed, that model should be translated into a computer model, i.e. a simulation program. To create a simulation model, appropriate computer applications and associated tools are required. The computer application for the simulation of discrete events FlexSim was used in this work. The application has an object-oriented discrete-event simulation environment that can easily be used to develop a simulation model to visualize and monitor dynamic system process flow activities. The possibility of building a simulation is presented through two approaches, where the first is based on the standard 3D model, while the second is based on an additional FlexSim tool that is integrated within the application itself, such as Process Flow that is particularly useful during the simulation of complex operations. An advantage of the second concept was

demonstrated by presenting a faster and more accurate construction process of the simulation model.

Keywords: simulation, simulation modelling, FlexSim, FlexSim Process Flow

# SADRŽAJ

<b>SAŽETAK .....</b>	<b>I</b>
<b>SUMMARY .....</b>	<b>I</b>
<b>SADRŽAJ .....</b>	<b>III</b>
<b>1. UVOD .....</b>	<b>5</b>
1.1. PROBLEM, PREDMET I OBJEKTI ISTRAŽIVANJA.....	5
1.2. RADNA HIPOTEZA .....	5
1.3. SVRHA i CILJEVI ISTRAŽIVANJA .....	5
1.4. ZNANSTVENE METODE .....	5
1.5. STRUKTURA RADA.....	6
<b>2. MODELIRANJE I SIMULACIJE.....</b>	<b>7</b>
2.1. Vrste sustava .....	7
2.2. Vrste modela .....	8
2.3. Simulacije .....	10
2.3.1. <i>Simulacijsko modeliranje.....</i>	<i>11</i>
2.3.2. <i>Razvoj procesa i resursa.....</i>	<i>12</i>
2.3.3. <i>Simulacije diskretnih događaja.....</i>	<i>13</i>
2.4. Prednosti i nedostaci simulacije.....	15
<b>3. SIMULACIJSKE APLIKACIJE .....</b>	<b>16</b>
3.1. ExtendSim.....	16
3.2. Rockwell Automation Arena .....	17
3.3. AnyLogic .....	18
<b>4. FLEXSIM.....</b>	<b>19</b>
4.1. FlexSim radno okruženje .....	20
4.1.1. <i>FlexSim knjižnica objekata i alatna traka .....</i>	<i>22</i>
4.1.2. <i>FlexSim properties .....</i>	<i>27</i>
4.1.3. <i>Postupak izrade 3D modela u FlexSim-u .....</i>	<i>30</i>
4.1.4. <i>Primjer 3D modela u FlexSim-u .....</i>	<i>31</i>
<b>5. FLEXSIM PROCESS FLOW .....</b>	<b>34</b>
5.1. Process Flow grafičko sučelje.....	34
5.2. Process Flow komponente .....	37



5.2.1.	Process Flow aktivnosti .....	37
5.2.2.	<i>Process Flow instance</i> .....	38
5.2.3.	<i>Process Flow varijable</i> .....	40
5.2.4.	<i>Process Flow aktivnosti koordinacije</i> .....	40
5.3.	Primjer Process Flow modela .....	42
5.4.	Prednosti i nedostaci FlexSim Process Flow-a .....	51
<b>6.</b>	<b>ZAKLJUČAK</b> .....	<b>52</b>
	<b>LITERATURA</b> .....	<b>53</b>
	<b>POPIS SLIKA</b> .....	<b>55</b>

# 1. UVOD

## 1.1. PROBLEM, PREDMET I OBJEKTI ISTRAŽIVANJA

Na osnovi relevantnih činjenica o problematici znanstvenoga istraživanja može se definirati problem istraživanja: Izrada računalne simulacije je prilično kompleksna i vremenski zahtjevna. U vezi s problemom istraživanja definiran je predmet istraživanja. Istražiti će se izrada simulacije u simulacijskom softveru FlexSim kroz dva pristupa, pri čemu se prvi temelji na standardnom 3D modelu, dok se drugi temelji na dodatnom FlexSim alatu, koji je integriran unutar same aplikacije, Process Flow-u. Problem i predmet istraživanja odnose se na dva međusobno povezana objekta istraživanja, modeliranje i simulacije.

## 1.2. RADNA HIPOTEZA

Sukladno bitnim odrednicama problema, predmeta i objekta istraživanja postavljena je radna hipoteza: U odnosu na standardni FlexSim 3D model FlexSim Process Flow ima brži i precizniji proces izrade simulacijskog modela.

## 1.3. SVRHA I CILJEVI ISTRAŽIVANJA

Svrha ovog istraživanja je utvrditi kako i na koji se način se mogu izraditi simulacijski modeli u računalnoj aplikaciji FlexSim. Cilj ovog rada je istražiti izradu simulacije u simulacijskom softveru FlexSim kroz dva pristupa, pri čemu se prvi temelji na standardnom FlexSim 3D modelu, a drugi pristup na FlexSim Process Flow-u. Čitatelji će dobiti odgovore na sljedeća pitanja:

1. Što je to simulacijsko modeliranje?
2. Koja je svrha simulacije?
3. Kako izraditi simulaciju koristeći FlexSim 3D model?
4. Kako izraditi simulaciju koristeći FlexSim Process Flow?
5. Koje su prednosti, a koji nedostaci pojedinog pristupa?

## 1.4. ZNANSTVENE METODE

Znanstvene metode koje se koriste u ovom radu su metoda indukcije i dedukcije, metoda deskripcije, metoda komparacije, statistička metoda, metoda dokazivanja i opovrgavanja,

metoda kompilacije, metoda modeliranja te grafičko prikazivanje podataka.

## **1.5. STRUKTURA RADA**

U prvom dijelu, Uvodu, navedeni su problem, predmet i objekt istraživanja, radna hipoteza i pomoćne hipoteze, svrha i ciljevi istraživanja, znanstvene metode i obrazložena je struktura rada. Drugi dio rada ima naslov MODELIRANJE I SIMULACIJE gdje je definiran sustav modeliranja kao i njegove komponente. U ovom poglavlju su elaborirani tipovi simulacijskih modela, prema vrsti varijabli u modelu i prema načinu na koji se stanje modela mijenja u vremenu, kao i tri vrste simulacijskog modeliranja. Poglavlje završava prednostima i nedostacima simulacija. Treće poglavlje predstavlja SIMULACIJSKE APLIKACIJE sa pojedinačnim opisima simulacijskih računalnih. U četvrtom poglavlju FLEXSIM je opisana računalna aplikacija FlexSim za 3D simulaciju koja modelira, simulira, predviđa i vizualizira poslovne sustave u raznim industrijama. Peti dio rada FLEXSIM PROCESS FLOW temelji se na dodatnom FlexSim alatu koji je integriran unutar same aplikacije, kao što je Process Flow koji je posebno koristan pri simulaciji složenih operacija.

U posljednjem dijelu, Zaključku, dana je sinteza rezultata istraživanja kojima je dokazivana postavljena radna hipoteza.

## 2. MODELIRANJE I SIMULACIJE

Model je pojednostavnjeni prikaz realnoga sustava napravljen zato da bi poslužio boljem razumijevanju i/ili daljnjem proučavanju tog sustava i eksperimentiranju s njim.

Realni sustav je više ili manje izdvojeni dio stvarnoga svijeta koji čini funkcionalnu cjelinu i koji se promatra, a definira se kao skup određenih dijelova koji međusobnim djelovanjem vode prema ostvarenju nekog logičkog cilja. Realnim sustavom smatra se i nešto što u stvarnosti ne postoji, nego je zamišljeno (planirano) da se napravi u budućnosti.<sup>1</sup>

### 2.1. Vrste sustava

Simulacijski se modeli dalje dijele prema dvama kriterijima: prema vrsti varijabli u modelu i prema načinu na koji se mijenja stanje modela u vremenu.

Prema izvjesnosti/neizvjesnosti rezultata modeli su: deterministički i stohastički.

Prema načinu na koji se mijenja stanje modela razlikuju se: modeli diskretnih događaja i modeli kontinuiranog stanja.

Nema strogih pravila (recepta) kako napraviti dobar model, nego veliku važnost imaju: razumijevanje problema, sposobnost apstrakcije, sistematičnost, sposobnost razlučivanja bitnog od nebitnog i za kraj samo iskustvo.

Postoje tri osnovna tipa simulacija, koji se razlikuju po pristupu modeliranja, vrsti problema te tehnikama modeliranja i simulacije koje su za njih razvijene, a to su:

- Kontinuirana simulacija
- Simulacija diskretnih događaja
- Miješana kontinuirano-diskretna simulacija.

---

<sup>1</sup> Schmidt, J.W., Taylor, R.E.: *Simulation and Analysis of Industrial Systems*, Richard D. Irwin, Inc., Homewood, USA, 1970.

## 2.2. Vrste modela

Postoje dva osnovna tipa podjele simulacijskih modela, jedan prema vrsti varijable u modelu (deterministički i stohastički), a drugi prema načinu na koji se stanjem modela mijenja u vremenu (diskretni i kontinuirani).

Deterministički modeli su oni čije je ponašanje potpuno predvidivo, odnosno u kojima je novo stanje sustava u potpunosti određeno prethodnim stanjem.

Stohastički modeli su oni čije se ponašanje ne može unaprijed predvidjeti, ali se mogu odrediti vjerojatnosti promjena stanja sustava. Stohastičke modele karakterizira slučajno ponašanje, odnosno postojanje slučajnih varijabli u sustavu. Tako neki stroj može nakon izvođenja operacije ostati ili u ispravno stanju ili se pokvariti, pri čemu svako od tih stanja ima neku vjerojatnost (zavisno o vrsti i starosti stroja). Drugi je izvor slučajnosti, aktivnost sa slučajnim trajanjem, tj. trajanjem koje se određuje iz uniformne razdiobe vjerojatnosti. Primjer te aktivnosti je točenje goriva na benzinskoj crpki, gdje se mjerenjem određenog uzroka točenja goriva vozilima dobivaju podaci iz kojih se može odrediti razdioba vjerojatnosti trajanja te aktivnosti.

U diskretnim modelima stanje sustava mijenja se samo u nekim vremenskim točkama. Takve promjene se još nazivaju i događaji. Tako model samoposluge može sadržavati varijable koje opisuju broj posjetitelja u redovima pred blagajnama, a taj broj može se mijenjati samo u trenutku dolaska posjetitelja u red i u trenutku početka posluživanja na blagajni. Isto tako, cijelo posluživanje na blagajni karakterizira se samo trenutkom početka i završetka posluživanja.

Kod kontinuiranih modela, varijable stanja mijenjaju se kontinuirano u vremenu. Primjer kontinuirane promjene je let aviona čiji se položaj i brzina mijenjaju kontinuirano u vremenu. Treba imati na umu da se na digitalnim računalima ne mogu izvoditi kontinuirane promjene veličina, već se one moraju aproksimirati brojevima od konačno mnogo znamenki. Već i zbog ograničenja u ukupnom vremenu izvođenja simulacije mora se zamijeniti kontinuirani tok vremena sa pomakom vremena u malim odsječcima. Unatoč diskretnim i kontinuiranim modelima, mogući su i miješani kontinuirano diskretni modeli koji sadrže i kontinuirane i diskretne varijable.<sup>2</sup>

Najvažniji konceptualni modeli u diskretnoj simulaciji jesu: dijagrami ciklusa aktivnosti i Petrijeve mreže (engl. Petri nets) kojih ima više vrsta.

---

<sup>2</sup> Čerić, V.: *Simulacijsko modeliranje*, Školska knjiga, Zagreb, 1993.

Konceptualne prikaze modela systemske dinamike čine: dijagrami uzročnih petlji (engl. causal loop diagrams) i dijagrami toka (engl. flow charts).

Prema rastućoj složenosti sustava, odnosno prema problemu koji se modelira, postoje sljedeće metode modeliranja i rješavanja problema: numeričke i simulacijske.

Simulacijski modeli općenito primjenjuju radi ovih triju ciljeva<sup>3</sup>:

1. ISTRAŽIVANJE – provjera (verifikacija/falsifikacija) hipoteza o strukturi i ponašanju sustava. Realni sustav koji istražujemo često nam je nedovoljno ili nepotpuno poznat te ćemo na temelju onoga što znamo ili samo pretpostavljamo o tom sustavu napraviti njegov model koji pak treba poslužiti daljnjemu proučavanju tog sustava i eksperimentiranju s njim. Ideja je da nam taj model posluži za reprodukciju ponašanja sustava kako bismo na temelju analize rezultata simulacije verificirali ili falsificirali hipotezu o njegovoj strukturi ili pak ocijenili pojedine kvantitativne parametre.

2. PROJEKTIRANJE – projekcija ponašanja sustava u situacijama s kojima nemamo iskustva (npr. u budućnosti za koju se pripremamo). Druga situacija u kojoj rješavanju stvarnih problema pristupamo s pomoću simulacijskog modeliranja jest ona kad trebamo predvidjeti ponašanje realnog sustava u situacijama s kojima nemamo iskustva (npr. u budućnosti za koju se pripremamo). Tada ćemo ponašanje promatranog sustava simulirati s pomoću računala da bismo ga temeljito analizirali ispitivanjem različitih, potencijalno mogućim uvjetima u kojima bi se mogao naći sustav i odabirom onih rješenja koja smatramo optimalnima. Najčešće se to radi kako bi se sam realni sustav promijenio (npr. reorganizirao) pa rezultati simulacije čine temelj za donošenje odluka o promjeni realnoga sustava.

3. EDUKACIJA – upoznavanje ponašanja sustava stjecanjem umjetnog (posrednog) iskustva (engl. vicarious experience). Mnogi su modeli razvijeni kao edukacijski (nastavni) materijali koji onomu tko uči omogućuju da u interakciji s računalom postavi svoju vlastitu strategiju rješavanja problema i ispita je simulacijom na računalu te tako stekne posredno iskustvo kako o ponašanju promatranog sustava i mogućnostima za njegovo mijenjanje. Mnogi modeli koji su prvotno bili razvijeni radi istraživanja ili projiciranja ponašanja sustava nakon što su provjereni i pokazali da vjerno označuju i oponašaju sustav prerasli u modele za edukaciju.

---

<sup>3</sup> Božikov, J.: *Modeliranje i simulacija*, poglavlje u knjizi Medicinskoinformatičke metode, Medicinski fakultet Sveučilišta u Zagrebu i Medicinska Naklada, Zagreb, 2007., 85-108.

### 2.3. Simulacije

Izraz simulacija dolazi od latinske riječi „simulare“ što znači „oponašati“, koji je glavni cilj izvođenja simulacije odnosno oponašanje realnih stvari, stanja ili procesa. Simulacijom se želi odrediti utjecaj promjene određenih ulaznih varijabli na ponašanje odabranog fizičkog ili apstraktnog sustava. Simulacija se koristi u mnogim prilikama, uključujući modeliranje prirodnih i ljudskih sustava. U tehnici se simulacije koriste za poboljšanje karakteristika, povećanja sigurnosti te samo testiranje sustava, a vrlo često i za obuku i obrazovanje ljudi. Simulacije se koriste i za prikaz stvarnih učinaka koje neki sustav ima na okoliš. Simulacije daju odgovor na pitanje „što će se dogoditi ako“, zato poboljšanje poslovanja bilo kojeg sustava je nezamislivo bez simulacija. Simulacijski proces je struktura rješavanja stvarnih problema s pomoću simulacijskog modeliranja. On se može opisati u obliku niza koraka koji čine pojedine faze rješavanja problema ovom metodom i koji slijede jedan nakon drugog, iako ne strogo sekvencijalno, jer je moguć povratak na prethodne korake procesa, ovisno o rezultatima dobivenima u pojedinim fazama toga procesa.

Osnovna priroda simulacije se očitava kroz primjenu računala za oponašanje operacija, procesa ili objekata stvarnog svijeta koji su usklađeni s adekvatno razvijenim pretpostavkama koje imaju oblik logičkih, statističkih ili matematičkih odnosa koji su oblikovani modelom. Krajnji rezultat se može izmjenjivati mijenjanjem raznih skupova ulaznih parametara uz pomoć kojih bi analitičar mogao bolje razumjeti dinamiku simuliranog sustava. Model se najčešće procjenjuje brojčano tijekom simuliranog vremenskog perioda. Podaci se prikupljaju kako bi se putem karakteristika sustava u stvarnim svijetu dala procjena.<sup>4</sup>

Problem korištenja simulacije vidi se u tome što ne postoji oko nje realni svijet, jer ga ona pokušava kroz prototip prikazati, ali je to dosta financijski teško iznijeti. Sustav je fizički nemoguće napraviti u svrhu samo jedne simulacije. Realni sustav kao i kod svakog eksperimenta ili istraživanja postoji, ali je u određenim situacijama jako težak za izvođenje ili održavanje. Potrebno je analizirati sustav tijekom jako dugog razdoblja zbog daljnjih utjecaja ili promjena, jedna od njih može biti povezana s porastom broja stanovništva.

Simulacijski modeli su modeli dinamičkih sustava, odnosno sustava koji se mijenjaju u vremenu. Njihove tipične primjene su u područjima inženjerstva i ekonomije. To su na primjer, modeli redova čekanja, proizvodnih procesa, skladištenja i prometa. Simulacijski modeli

---

<sup>4</sup> Robinson, S.: Simulation: The Practice of Model Development and Use, John Wiley & Sons, New York, 2004.

moraju omogućiti ispravan prikaz i efikasno izvođenje pomaka vremena. Također je bitno omogućavanje istovremenog odvijanja aktivnosti te opisivanje procesa koji konkuriraju za iste resurse (strojevi, blagajne ili financijska sredstva). Ti su zahtjevi znatan problem za modeliranje, zbog čega su se simulacijski modeli razvili u posebnu kategoriju modela. Razvijeni su i specifični alati za konceptualno modeliranje i specifični programski jezici kao adekvatna sredstva za posebne zahtjeve simulacijskog modeliranja.

### **2.3.1. Simulacijsko modeliranje**

Simulacijsko modeliranje i analiza je stvaranje i eksperimentiranja s automatiziranim i računalnim matematičkim modelom fizičkog sustava. Simulacija računalom koristi se za umanjivanje rizika koji su povezani uz kreiranje novih sustava ili promjenu onih koji već postoje. Simulacija koristi model za kreiranje zaključaka koji dozvoljavaju pristup u ponašanje elemenata stvarnog svijeta koji se istražuje i analizira. Računalna simulacija koristi jednak koncept, ali traži da se model generira uporabom računalnog programiranja.<sup>5</sup>

Simulacijsko modeliranje te analiza raznih tipova sustava primjenjuje se sljedećim svrhama:

1. Dobivanje uvida u rad sustava
2. Razvijanje operativnih ili resursa za poboljšanje performansi sustava
3. Testiranje novih koncepata ili sustava prije implementacije
4. Dobivanje informacija bez ometanja stvarnog sustava

Neki su sustavi toliko složeni da je teško razumjeti rad i interakcije unutar sustava bez dinamičkog modela. Nije moguće proučiti neki sustav ako se zaustavi ili ako se obraća pozornost na izolirane pojedinačne komponente.

Kod primjene simulacijskog modeliranja ne može se dobiti rješenje u analitičkom obliku, u kojem su zavisne varijable funkcije nezavisnih varijabli, već se rješenje problema dobiva eksperimentiranjem modelom sustava. Prema tome, simulacijski eksperimenti daju kao rezultat skup točaka, tj. vrijednosti zavisnih varijabli za pojedine vrijednosti nezavisnih varijabli. Zbog slučajnog karaktera varijabli modela dobiva se čak i više različitih vrijednosti zavisnih varijabli za istu vrijednost nezavisnih varijabli, odnosno eksperimenti daju određeni uzorak vrijednosti zavisnih varijabli. Pri tome planiranje i analiza simulacijskih eksperimenata zahtijevaju statistički pristup.

---

<sup>5</sup> Čerić, V.: *Simulacijsko modeliranje*, Školska knjiga, Zagreb, 1993.



### 2.3.2. Razvoj procesa i resursa

Razne promjene u poslovnim politikama sastoje se od raznovrsnih prioriteta raspoređivanja poslova. Promjene u politikama resursa mogu uključivati i raspored rada. Ako sustav još ne postoji, simulacijski model može pomoći odrediti koliko će dobro predloženi sustav raditi. Trošak modeliranja novog sustava može biti vrlo mali u usporedbi s kapitalnim investicijama uključenim u instalaciju bilo kojeg značajnog proizvodnog procesa. Učinci različitih razina i troškova opreme mogu se procijeniti. Uporaba simulacijskog modela prije implementacije može poboljšati konfiguraciju odabrane opreme. Dobivanje informacija bez ometanja stvarnog sustava. Omogućuje eksperimentiranje bez prekida postojećih sustava. U sustavima koji već postoje, testiranje novih ideja može biti teško, skupo ili nemoguće. Najprije se mogu napraviti željene modifikacije na modelu, i odrediti utjecaj sustava, a potom se može donijeti odluka o primjeni promjena u stvarnom sustavu.

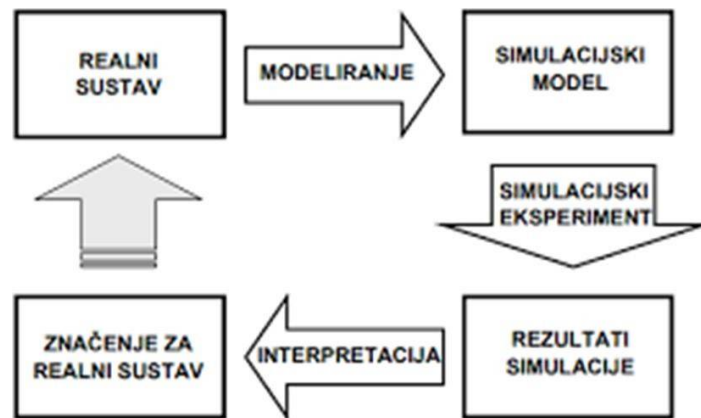
Koraci simulacijskog procesa su:<sup>6</sup>

1. Definicija cilja simulacijske studije
2. Identifikacija sustava
3. Prikupljanje i analiza podataka o sustavu
4. Izgradnja simulacijskog modela i simulacijskog programa
5. Vrednovanje simulacijskog modela
6. Planiranje i izvođenje simulacijskih eksperimenata
7. Analiza rezultata eksperimenata
8. Zaključci i preporuke

Na slici 1. prikazan je postupak simulacijskog procesa. Prvi korak u tom procesu je eksperimentiranje realnog sustava. Nakon izrade simulacijskog modela potrebno je mijenjati ulazne podatke i pratiti promjene koje se dogode na modelu. Dobivene promjene je potrebno interpretirati na realnom sustavu, s ciljem njegovog poboljšanja.

---

<sup>6</sup> Law, A.M., Kelton, W.D.: *Simulation Modelling and Analysis, 2nd edition*, McGraw-Hill, Inc., Singapore, 2000.



Slika 1 – Simulacijski proces

Izvor: Božikov, J.: *Modeliranje i simulacija*, poglavlje u knjizi Medicinskoinformatičke metode, Medicinski fakultet Sveučilišta u Zagrebu i Medicinska Naklada, Zagreb, 2007., 85-108.

### 2.3.3. Simulacije diskretnih događaja

Diskretna simulacija (engl. Discrete event simulation (DES)) je proces kodiranja ponašanja složenih sustava kao slijeda čvrsto definiranih događaja. Diskretna simulacija opisuje promjene stanja koje se odvijaju diskontinuirano u vremenu. Promjene stanja su posljedica međudjelovanja među objektima sustava (npr. ako stroj napravi obradu na materijalu, obrada materijala je završena i njegovo stanje se mijenja). Namijenjena je razvoju modela koji detaljno opisuju strukturu sustava i njegove elemente, tj. oponašaju stvarne sustave i procese, te objekte iz stvarnog svijeta i njihovo međudjelovanje.<sup>7</sup>

Najčešće se upotrebljava za modeliranje i analizu sustava s redovima čekanja na resurse (resursi mogu biti: blagajne, autobusi, računala, šalteri i dr.), pri čemu su glavni elementi opisa:<sup>8</sup>

1. Tijek procesa - obično se prikazuje strelicama koje povezuju pojedine elemente i odlukama koje određuju smjer tijeka procesa
2. Kapaciteti resursa - opisuju vrijeme u kojem će npr. resurs raditi 1 poluproizvod za 3 minute
3. Ograničenja - opisuju raspoložive resurse, npr. 3 raspoloživa stroja ili 4 djelatnika ili sl.
4. Objekti - komponente sustava koje mogu obraditi pojedini entitet koji prolazi kroz sustav.

<sup>7</sup> Banks, J., Carson, J.S.II, Nelson, B.L., Nicol, D.M.: *Discrete-Event System Simulation, 5th edition*, Pearson Education, Inc., Upper Saddle River, New Jersey, USA, 2010.

<sup>8</sup> Diskretne-simulacije, <http://www.efos.unios.hr/poslovne-simulacije/wp-content/uploads/sites/180/2013/04/Predavanje-4-Diskretne-simulacije.pdf> (11.07.2022.)

5. Stalni entiteti - ostaju u modelu tijekom trajanja simulacije (npr. šalteri).
6. Privremeni entiteti - prolaze kroz sustav (npr. klijenti, autobusi itd.) Svaki entitet može imati više atributa.
7. Atributi - opisuju svojstva objekata i entiteta (npr. odredište putnika, brzina obrade stroja, itd.)

Kod diskretne simulacije, nezavisne (ulazne varijable) čije vrijednosti korisnik upisuje u sustav su:

1. Vrijeme između dva uzastopna dolaska entiteta (engl. Time Between Arrivals)
2. Vrijeme odvijanja pojedinog procesa (npr. vrijeme opsluživanja na šalteru, ili vrijeme obrade na stroju ili dr.)
3. Broj raspoloživih resursa (npr. broj šaltera, ili djelatnika, ili strojeva)
4. Organizacija čekanja (FIFO, LIFO, po prioritetu i sl.)

Zavisne ili izlazne varijable su varijable čije vrijednosti se dobiju kao rezultat simulacije:

1. broj entiteta u redu čekanja,
2. vrijeme čekanja,
3. iskorištenje resursa,
4. propusnost sustava i slično.

Konceptualni model diskretne simulacije služi da se izdvoje najvažnija svojstva sustava, opišu njegovi bitni elementi i njihovo međudjelovanje. Izradom konceptualnog modela omogućuje se strukturiranje problema i njegovo bolje razumijevanje. Za njegovu izradu koriste se grafičke metode modeliranja, (npr. dijagrami ciklusa aktivnosti (DCA)).<sup>9</sup>

Da bi se stvorio primjeren simulacijski model sustava ili njegovih važnijih podsustava, potrebno je utvrditi:

1. Predmet simulacijskog modeliranja
2. cilj modela simulacije
3. subjekte, resurse i kritičke procese kojima je potrebno posvetiti posebnu pozornost
4. „što-ako“ scenarij

---

<sup>9</sup> Mourtzis, D.: *Simulation in the design and operation of manufacturing systems: state of the art and new trends*, International Journal of Production Research, Vol. 58, No. 3, 1–23, 2019. doi:10.1080/00207543.2019.1636321

5. potrebne ulazne podatke
6. output za statističku analizu

#### **2.4. Prednosti i nedostaci simulacije**

Prednosti:<sup>10</sup>

- a) Može se doći do zaključaka bez remećenja rada stvarnog sustava
- b) Ne trebaju se kupovati dodatni stvarni resursi
- c) Mogu se testirati bilo kakve hipoteze o sustavu
- d) Vrijeme se može sažeti i rastegnuto ovisno o potrebi
- e) Mogu se analizirati kritične točke sustava prije nego one dođu do izražaja u stvarnom sustavu
- f) Simulacija često pomaže da se shvati kako sustav funkcionira

Nedostaci:

- a) Modeliranje sustava zahtijeva specifično znanje i trening
- b) Ponekad je teško protumačiti rezultate simulacija
- c) Modeliranje i analiza mogu jako dugo trajati i biti vrlo skupi

---

<sup>10</sup> Informacijske mreže - Simulacija diskretnih događaja, [https://www.fer.unizg.hr/\\_download/repository/INFMRE-2019\\_05.pdf](https://www.fer.unizg.hr/_download/repository/INFMRE-2019_05.pdf) (10.07.2022.)

### **3. SIMULACIJSKE APLIKACIJE**

Simulacijski računalne aplikacije koriste se za provedbu simulacija na računalu. Pomoću njih se pokušava optimizirati realni sustav, kako bi se smanjili troškovi i povećala efikasnost. Neki od tih računalnih aplikacija su: ExtendSim, Arena Rockwell Automation, Flexsim, AnyLogic. Najviše pažnje usmjerit će se na program Flexsim u slijedećim poglavljima, dok će se ostali ukratko opisati u nastavku.

#### **3.1. ExtendSim**

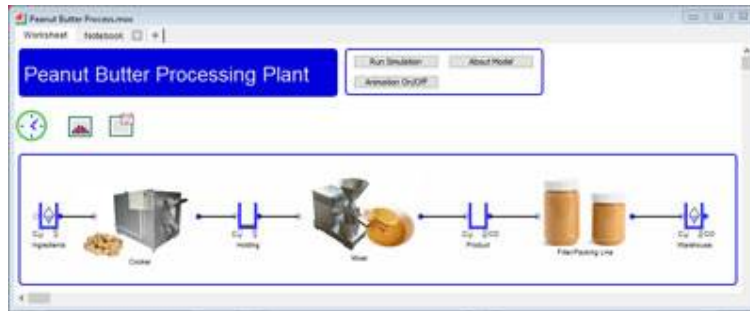
ExtendSim<sup>11</sup> je aplikacija za izradu simulacijskih modela i simuliranje operacijskih procesa. Jednostavna za korištenje, ali iznimno efikasna za razumijevanje složenih sustava te daje brzo pouzdane rezultate simuliranja, slika 2.

S ExtendSim može se:

- predvidjeti tijek i rezultate pojedinih akcija
- pretpostaviti i potaknuti kreativno razmišljanje
- vizualizirati procese logičnom ili u virtualnom okruženju
- identificirati problematična područja prije provedbe
- istražite potencijalne učinke izmjena
- potvrdi da li su sve varijable poznate
- optimizirati poslovanje
- procijeniti ideje i identificirati neučinkovitosti
- razumjeti zašto promatrani čin odvija
- predvidjeti u cjelovitosti provedivost svojih planova

---

<sup>11</sup> ExtendSim. <https://extendsim.com/products/line/overview> (15.07.2022.)

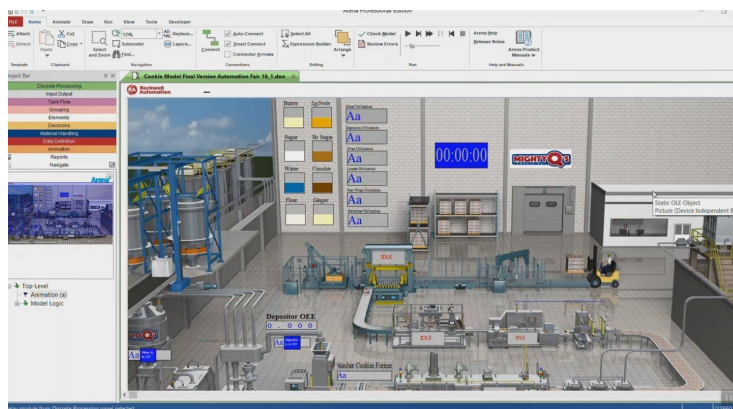


**Slika 2 – ExtendSim Pro**

Izvor: <https://extendsim.com/products/line/extendsimpro> (15.07.2022.)

### 3.2. Rockwell Automation Arena

U Areni<sup>12</sup>, slika 3, korisnik gradi model eksperimenta postavljanjem modula koji predstavljaju procese. Linije konektora koriste se za spajanje modula zajedno i za specificiranje tijeka entiteta. Dok moduli imaju posebne radnje u odnosu na entitete, tijek i vrijeme, precizan prikaz svakog modula i entiteta u odnosu na objekte iz stvarnog života podložan je promjenama. Isto tako bilježe se statistički podaci, kao što je vrijeme ciklusa i WIP (engl. work in process) razine, mogu se i ispisati kao izvješća. Koristi se u različitim sferama: industrija zabave, infrastruktura, prehrambena industrija, rudarstvo, automatizacija itd. Podržava uvoz Microsoft Visio dijagrama toka, kao i čitanje ili slanje izlaza u Excel proračunske tablice i Access baze podataka te Visual Basic kroz ActiveX kontrole.



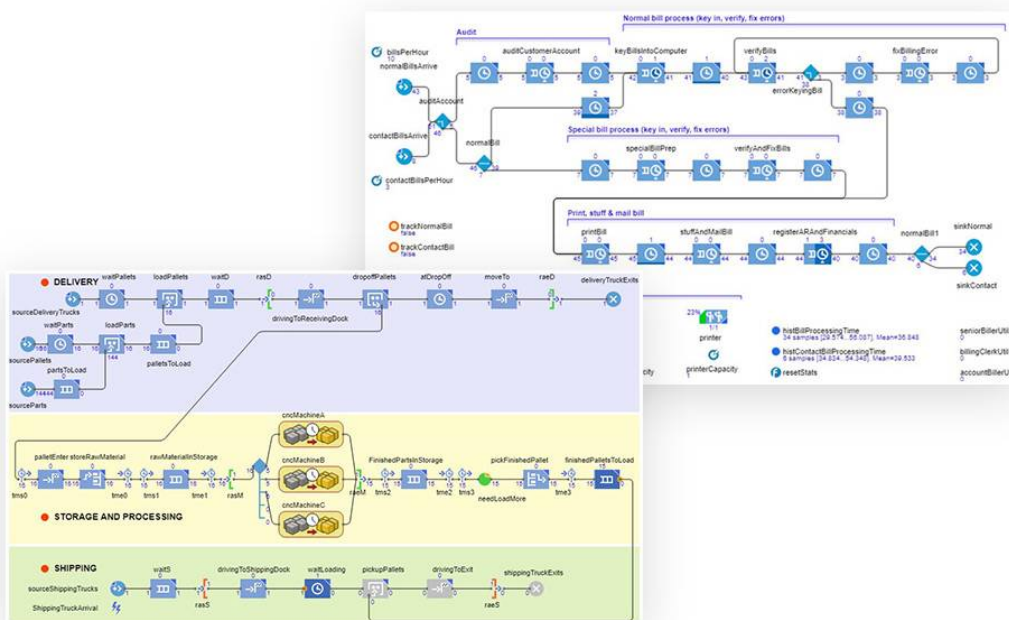
**Slika 3 - Rockwell Automation Arena**

Izvor: <https://www.rockwellautomation.com/en-us/products/software/arena-simulation/discrete-event-modeling/food-beverage.html> (15.07.2022.)

<sup>12</sup> Rockwell Automation Arena, <https://www.rockwellautomation.com/en-us/products/software/arena-simulation.html> (15.07.2022.)

### 3.3. AnyLogic

AnyLogic<sup>13</sup>, slika 4, podržava metodologije simulacije diskretnih događaja, simulacije zasnovane na agentima i sistemsku dinamiku. Koristi se za simulaciju: tržišta i konkurencije, zdravstvene zaštite, proizvodnje, lanaca opskrbe i logistike, maloprodaje, poslovnih procesa, upravljanje projektima i imovinom, dinamika pješaka i cestovni promet itd.



Slika 4 – AnyLogic

Izvor: <https://www.anylogic.com/features/libraries/process-modeling-library/> (15.07.2022.)

<sup>13</sup> AnyLogic, <https://www.anylogic.com/> (15.07.2022.)

## 4. FLEXSIM

FlexSim<sup>14</sup> je softver za 3D simulaciju koji modelira, simulira, predviđa i vizualizira poslovne sustave u raznim industrijama: proizvodnji, rukovanju materijalima, zdravstvu, skladištenju, rudarstvu, logistici i još mnogo toga. FlexSim aplikacija može pomoći tvrtki u donošenju informiranijih odluka.<sup>15</sup>

FlexSim može koristiti za:

- Uslužne problemi – Ubrzanje udovoljenja potreba kupaca te brzi odgovor na njihove zahtjevi koji dovodi najveći nivo satisfakcije, koristeći najmanje moguće uloženi sredstva
- Probleme proizvodnje – Izrada najboljeg mogućeg proizvoda uz najmanje uloženi finansijski sredstva u određenom vremenskom periodu
- Logističke probleme – Proces transporta u određenom vremenskom roku uz minimalno finansijsko ulaganje
- Vizualizaciju - vidljive rezultate predloženih promjena radi optimizacije protoka proizvoda, osoblja, korištenja resursa, oblikovanja tlocrta i gotovo bilo kojeg drugog aspekta sustava
- Optimizaciju - prije nego što se primijene promjene u stvarnom sustavu,
- Proučavanje alternativnih ideja za ulaganja i planove za smanjenje troškova

Mnoge tvrtke koriste FlexSim za:

- Iskoristivost opreme
- Kraće vrijeme čekanja i kraći redovi čekanja
- Poboljšana alokacija resursa
- Eliminacija ne imanja zaliha
- Manji učinak kvara na sustav
- Manji utjecaj otpada na sustav
- Alternativne investicijske ideje
- Uvid u jeftinije planove izrade sustava

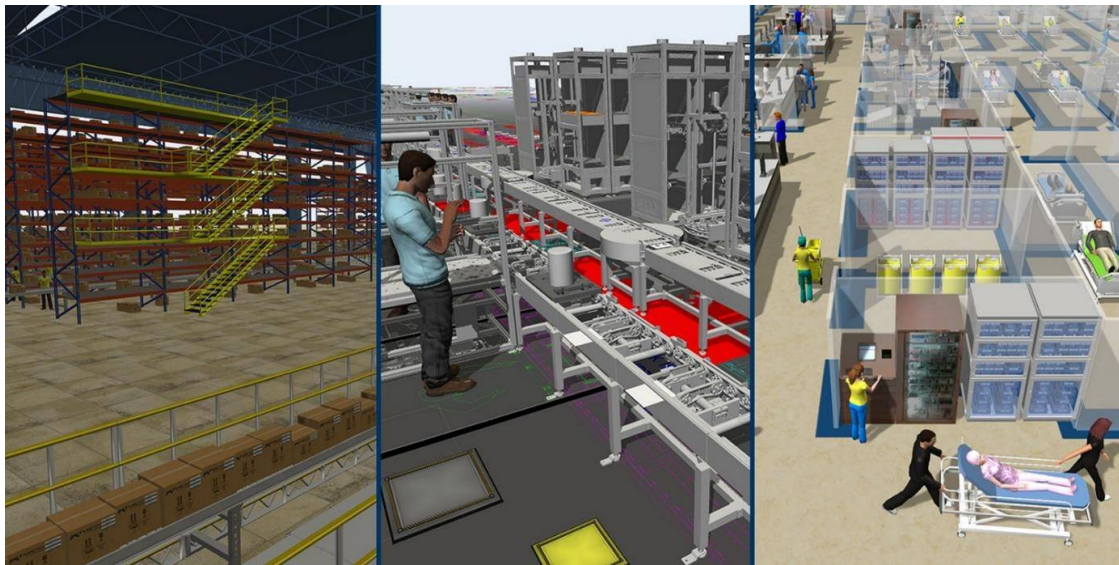
---

<sup>14</sup> FlexSim, <https://www.flexsim.com/flexsim/> (17.07.2022.)

<sup>15</sup> Nordgren, W. B.: *Flexsim simulation environment*. Proceedings of Winter Simulation Conference, Phoenix, Arizona, USA. IEEE, 197-200 Vol. 1. 2003.



- Poboljšano rukovanje materijalima
- Poboljšanje pripreme alata
- Optimalnije određivanje prioriteta te brže otpremanje dobara i usluga
- Obuka radnika u vezi ponašanja u sustavu čime se dolazi do boljih performansi
- Mogućnost primjene novih dizajna poslovnog prostora
- Lakše donošenje odluka



**Slika 5 – FlexSim**

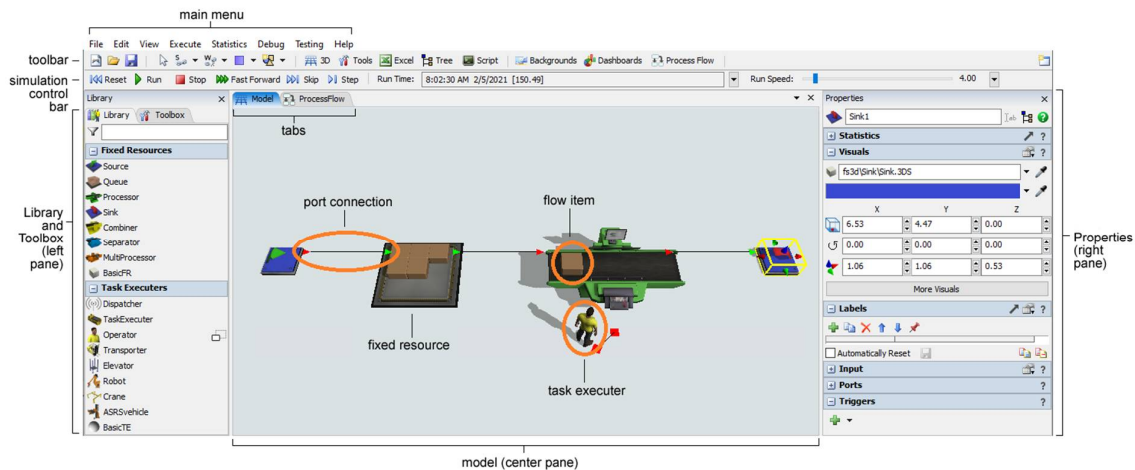
Izvor: <https://www.flexsim.com/flexsim/> (17.07.2022.)

Slika 5 pokazuje FlexSim aplikaciju koja u 3D prikazu pruža izgled skladišta, tvornice i bolnice kako bi se pokazalo funkcioniranje procesa u realnim uvjetima.

#### **4.1. FlexSim radno okruženje**

Radno okruženje<sup>16</sup>, na slici 6, se sastoji od traka padajućih izbornika i ikona, traka za simulaciju, knjižnice objekata i radne površine. Radna površina je namijenjena izradi modela, u njemu je grafički prikazan cijeli model i omogućuje prikaz iz različitih kutova. Objekti iz knjižnice se povlačenjem prenose na radu površinu.

<sup>16</sup> FlexSim priručnik, <https://docs.flexsim.com/en/22.2/FlexSimUI/OverviewUserInterface/OverviewUserInterface.html> (25.07.2022.)



**Slika 6 – Izgled radnog okruženja simulacijskog softvera FlexSim**

Izvor: <https://docs.flexsim.com/en/22.2/FlexSimUI/OverviewUserInterface/OverviewUserInterface.html>  
(25.07.2022.)

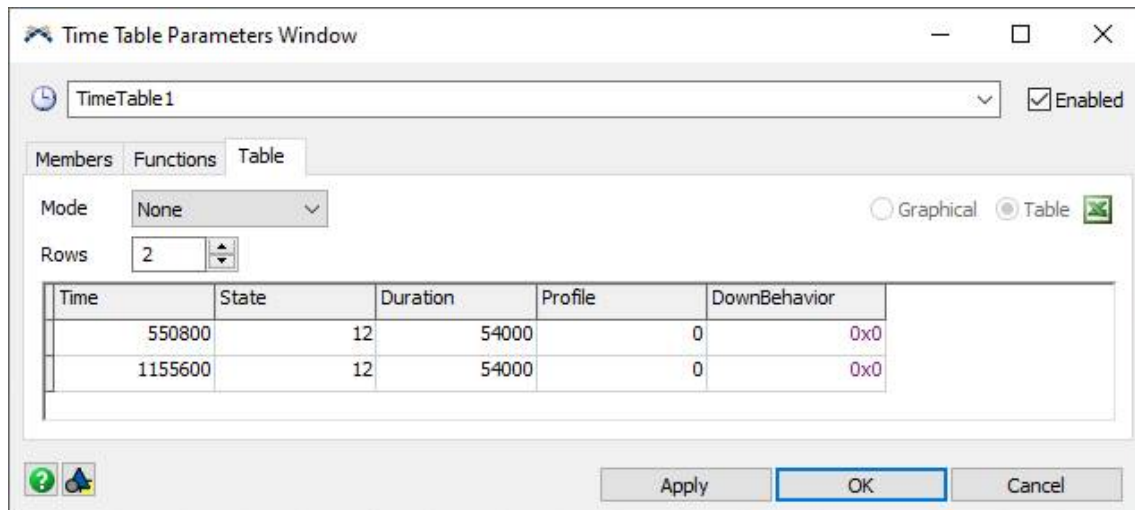
U traci padajućih izbornika i ikona se nalazi:

- a) File, datoteka, omogućuje otvaranje novog modela, pohranjivanje postojećeg modela te otvaranje postojećeg već pohranjenog modela.
- b) Edit meni, odnosno uređivanje, sadrži funkcije poput poništavanja unosa ili povratka unosa/upisa, određivanje decimalnih mjesta te zaključavanje i otključavanje postavki.
- c) View, pogled, je meni koji sadrži mogućnosti različitih pogleda na model.
- d) Execute meni omogućuje pokretanje, resetiranje i zaustavljanje modela kao i povećanje i smanjenje brzine.
- e) U meniju Statistics se nude standardne informacije o modelu ili konkretne informacije za svaki posebni objekt u modelu, dokumentacije modela, objekta ili samo dijela objekta.
- f) Debug meni je za odstranjivanje nepravilnosti.
- g) Zadnji meni je Help gdje se može pronaći korisnički priručnik za lakši početak i učenje.

#### 4.1.1. FlexSim knjižnica objekata i alatna traka

Osnova svakog pojedinog modela u FlexSim-u su objekti. Objekti se nalaze u knjižnici objekata. Objekti su implementirani pomoću objektno orijentirane metodologije, koja uključuje hijerarhiju superklasa/podklasa. Objekti podklase nasljeđuju attribute i zadano ponašanje od svoje superklase, prilagođavajući takvo ponašanje kako bi odgovaralo specifičnoj situaciji. U Flexsim-u, većina objekata u biblioteci objekata su stvoreni kao jedna od dvije glavne vrste objekata ili superklasa. Ove dvije opće vrste se nazivaju FixedResources superklasa i TaskExecuters superklasa odnosno fiksni i mobilni objekti. Fiksni objekti su okosnica simulacijskog modela, određuju tijek modela, primaju i šalju entitete. Mobilni objekti su dijeljeni resursi i mobilni resursi koji obavljaju dodijeljene zadatke.

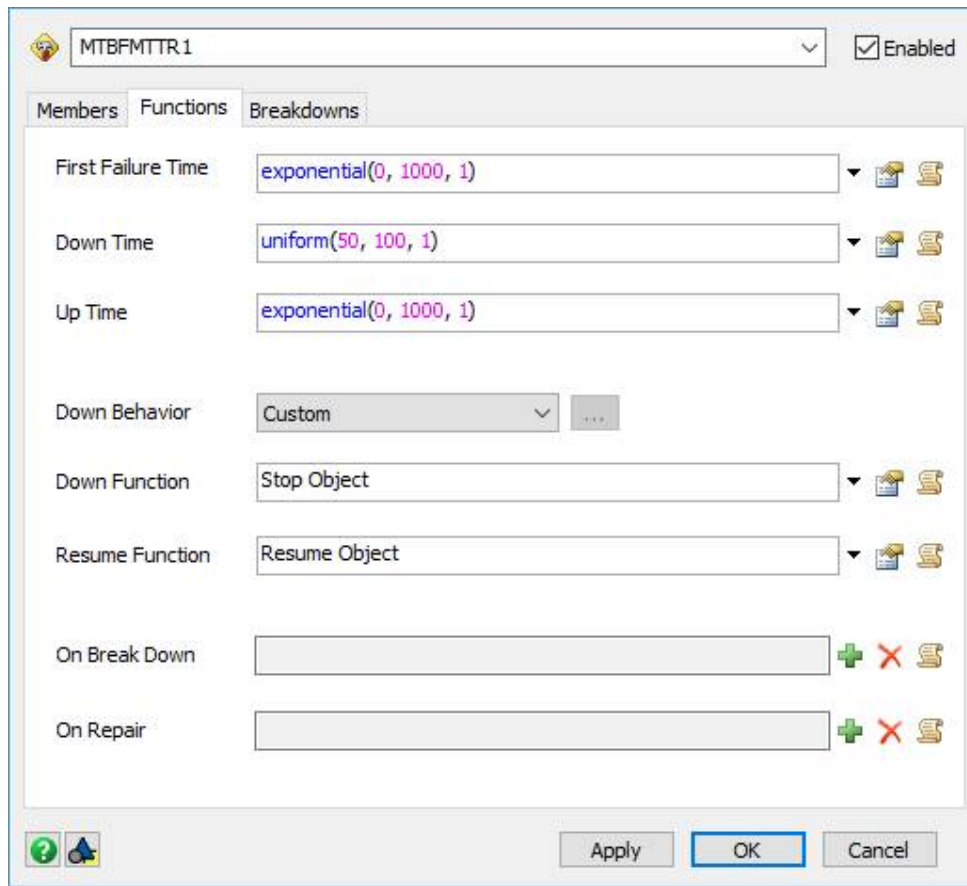
Flexsim toolbox, odnosno alatna traka prvotno se sastoji od globalne tablice (global table). To su unutarnje proračunske tablice u koje se mogu unijeti te memorirati raznovrsne informacije i podaci koji mogu biti dinamički referencirani i drugim alatima u FlexSim-u. Zatim slijedi Time Table, odnosno vremenska tablica, slika 7. Njena primarna zadaća je zakazivanje promjena stanja za specifične 3D objekte u modelu. Vremenske tablice služe za planiranje vremena prestanka rada, a tome slično su promjene stanja, za određene objekte u modelu. Svaki Time Table može kontrolirati mnogo objekata, a svakim objektom može upravljati više vremenskih tablica. Model može sadržavati bilo koji broj vremenskih tablica. Primarna namjena objekta Time Table je utvrditi planirana razdoblja zastoja za objekte u modelu. Ta takozvana razdoblja zastoja definirana su s dva broja: vrijeme zastoja rada i trajanje. Razdoblje zastoja rada počinje u navedenom vremenu zastoja rada i trajat će određeno vrijeme u jedinicama vremena modela.



**Slika 7 – Vremenska tablica**

Izvor: FlexSim aplikacija

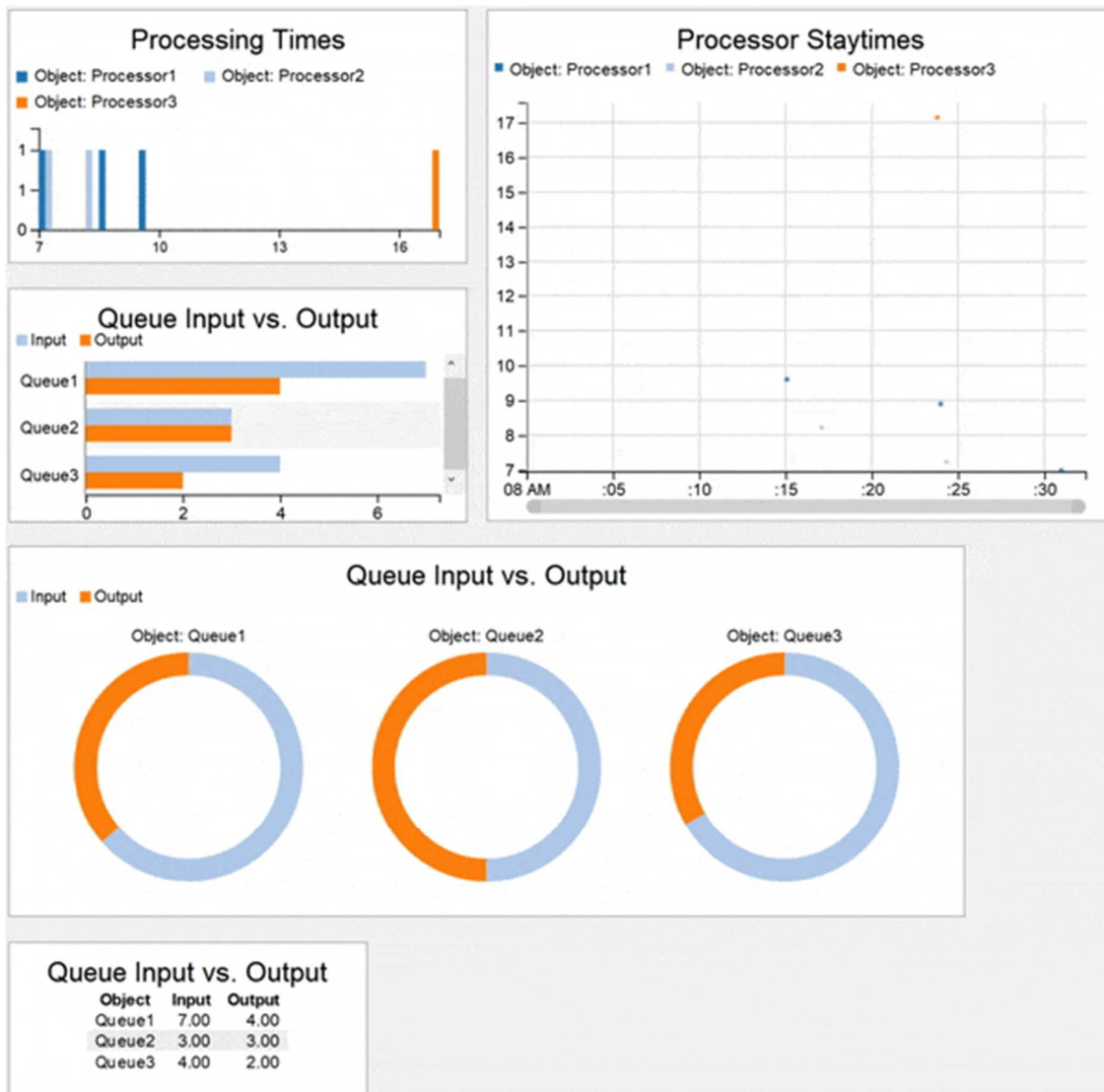
Potom na red dolazi opcija MTBF/MTTR, slika 8. Njezina glavni zadatak je postavljanje nasumičnog vremena kvara i oporavka za grupe 3D objekata u modelu. Svaka opcija MTBF/MTTR može sadržavati bilo koji broj članova objekta i svaki objekt može kontrolirati više od jedne opcije MTBF/MTTR. Opcija MTBF/MTTR također omogućuje i da se specificira drugo stanje u koje će objekti prijeći kada padne i te kakvo će njegovo ponašanje biti. Model može sadržavati bilo koji broj MTBF/MTTR opcija.



**Slika 8 – MTBF/MTTR**

Izvor: FlexSim aplikacija

Zatim slijedi Dashboard ili Nadzorna ploča, slika 9. Koristi se za prikaz statističkih informacija grafički ili tekstualno za vrijeme izvođenja simulacije u stvarnom vremenu. Nadzorne ploče su prazan prozorski okvir koji se može prilagoditi bilo čemu, gdje se treba pristupiti tijekom izvođenja simulacije. Najčešći razlog upotrebe nadzorne ploče je prikaz grafikona. Grafikoni prikazuju statistiku i vizualno će prikazati podatke iz Statistics Collector-a. Oni se zapravo ažuriraju u stvarnom vremenu dok simulacijski model radi. Osim što sadrže prikaz grafikona, nadzorne ploče također mogu uključivati i druge vrste prikaza modela kao što su datum i vrijeme ili dokumentacija modela.



**Slika 9 – Dashboard**

Izvor: FlexSim aplikacija

Onda slijedi Process Flow, što u prijevodu znači Tijek procesa. Primarna zadaća mu je izgradnja cjelokupne logike modela. Sastoji se od Process Flow View-a što je glavni prostor za izradu Process Flow-a i od Process Flow Pane-a koji je zapravo okvir koji se stvori svaki put kada se otvori novi okvir modela.

Niz nastavlja Global list, upotrebljava se kada se treba stvoriti složeniji tok između 3D objekata. Isto tako mogu koristiti i za usmjeravanje bez povezivanja, filtriranje entiteta, upravljanje zadacima i određivanje prioriteta fiksnim resursima. Osim usmjeravanja entiteta

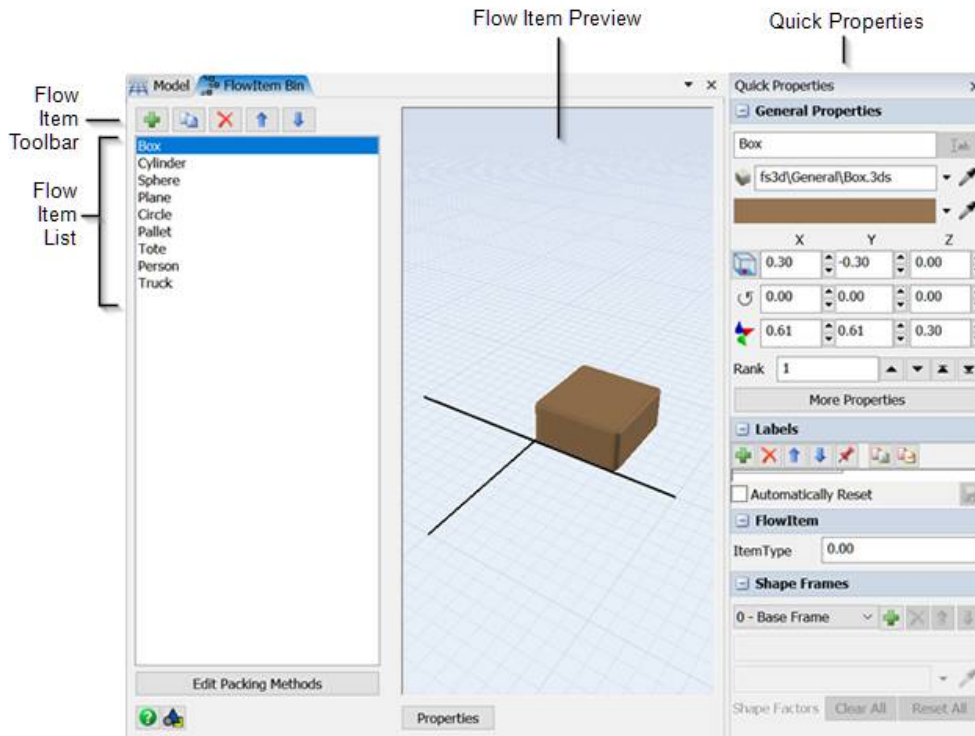
između objekata Global list ima i druge funkcije kao što su: usmjeravanje bez povezivanja, filtriranje entiteta, upravljanje podacima, određivanje prioriteta fiksnim resursima, Potom slijedi Statistics Collector (sakupljač statistike). Služi za dobivanje standardne ili prilagođene statistike iz simulacijskog modela. To je alat koji prikuplja podatke iz objekata i događaja koji se navedu tijekom izvođenja simulacije i staviti podatke u tablicu, poput one prikazane na slici 10, koja prati vrijeme u kojem artikli ulaze i izlaze iz pokretne trake:

StayTime		
ItemID	EnterTime	ExitTime
1.00	7/6/2017 8:00:05 AM	7/6/2017 8:00:22 AM
2.00	7/6/2017 8:00:14 AM	7/6/2017 8:00:31 AM
3.00	7/6/2017 8:00:20 AM	7/6/2017 8:00:38 AM
4.00	7/6/2017 8:00:20 AM	7/6/2017 8:00:45 AM
5.00	7/6/2017 8:00:30 AM	7/6/2017 8:00:52 AM
6.00	7/6/2017 8:00:31 AM	7/6/2017 8:00:59 AM
7.00	7/6/2017 8:00:39 AM	7/6/2017 8:01:06 AM
8.00	7/6/2017 8:00:51 AM	7/6/2017 8:01:13 AM
9.00	7/6/2017 8:00:51 AM	7/6/2017 8:01:27 AM
10.00	7/6/2017 8:00:54 AM	7/6/2017 8:01:55 AM
11.00	7/6/2017 8:00:58 AM	7/6/2017 8:01:20 AM
12.00	7/6/2017 8:01:00 AM	7/6/2017 8:01:41 AM
13.00	7/6/2017 8:01:01 AM	7/6/2017 8:02:37 AM
14.00	7/6/2017 8:01:04 AM	7/6/2017 8:01:34 AM
15.00	7/6/2017 8:01:18 AM	7/6/2017 8:01:48 AM
16.00	7/6/2017 8:01:38 AM	7/6/2017 8:02:02 AM
17.00	7/6/2017 8:01:41 AM	7/6/2017 8:02:16 AM
18.00	7/6/2017 8:01:50 AM	7/6/2017 8:02:09 AM
19.00	7/6/2017 8:01:53 AM	7/6/2017 8:02:30 AM
20.00	7/6/2017 8:01:55 AM	7/6/2017 8:02:44 AM
21.00	7/6/2017 8:01:58 AM	7/6/2017 8:02:23 AM
22.00	7/6/2017 8:02:27 AM	7/6/2017 8:02:51 AM
23.00	7/6/2017 8:02:35 AM	7/6/2017 8:02:58 AM
24.00	7/6/2017 8:02:43 AM	7/6/2017 8:03:05 AM

**Slika 10 – Statistics Collector**

Izvor: FlexSim aplikacija

Flow Item Bin služi za prilagodbu vizualnog izgleda entiteta u određenom simulacijskom modelu. Putem njega može se promijeniti boja, veličina ili rotacija osnovnih oblika entiteta, dodati vlastite prilagođene 3D stavke, postaviti zadane oznake ili vrste stavki, stvoriti različite okvire oblika za osnovne oblike toka tako da se njihov izgled može mijenjati u različitim fazama simulacijskog modela. Flow Item Bin je označen na slici 11.



**Slika 11 – Flow Item Bin**

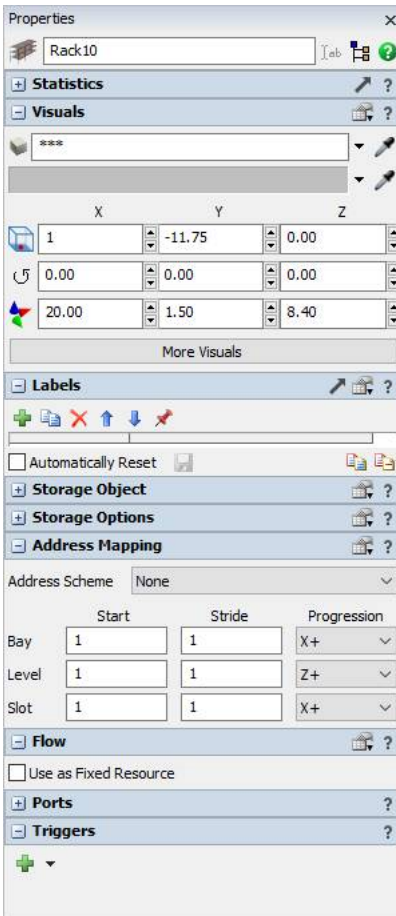
Izvor:

<https://docs.flexsim.com/en/22.2/Using3DObjects/WorkingWithFlowItems/UsingFlowItemBin/UsingFlowItemBin.html> (25.07.2022.)

#### 4.1.2. FlexSim properties

Opcija Properties omogućava editiranje svojstava za bilo koji objekt koji je trenutno odabran u modelu ili u određenom alatu, slika 12 i 13. Također se može koristiti i za pregled važnih informacija o objektima kada su odabrani tijekom izvođenja simulacije. Osjetljiva je i na kontekst. Prikazuje svojstva koja odgovaraju aktivnoj kartici na kojoj se trenutno radi, kao i objekt ili čvor na koji se kliknulo u toj kartici. Na primjer, ako je korisnički priručnik aktivna kartica, desni okvir prikazuje sadržaj za korisnički priručnik.

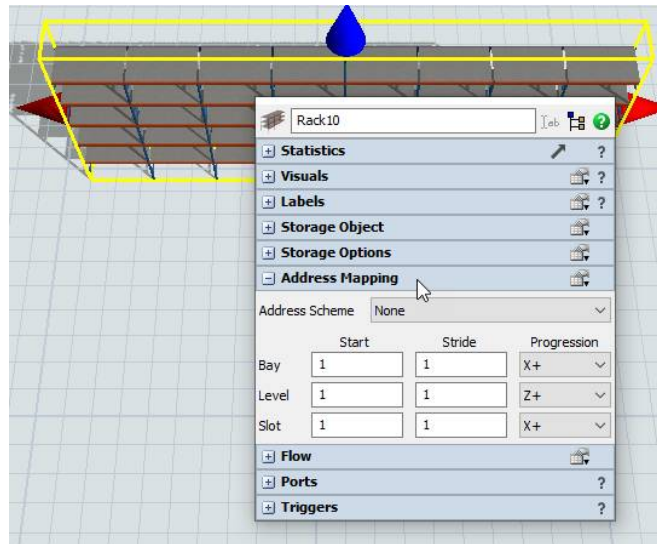




**Slika 12 – FlexSim properties**

Izvor: FlexSim aplikacija

Properties se nalazi na desnoj strani standardnog korisničkog sučelja FlexSima. Kako bi se uredila svojstva objekta, potrebno je samo kliknuti na njega, te će se sva njegova svojstva pojaviti u Properties s desne strane. Alternativno, može se i dvaput kliknuti na objekt kako bi se vidjela njegova svojstva. Time će se otvoriti Properties pored pokazivača, gdje se mogu urediti svojstva objekta, slika 13.



**Slika 13 – FlexSim properties objekta**

Izvor: FlexSim aplikacija

Tipično svojstvo u FlexSim-u najvjerojatnije će imati slične komponente. Sljedeća slika 14 prikazuje ove komponente za dva različita svojstva na procesoru:



**Slika 14 – Prikaz simulacijskog procesa**

Izvor: FlexSim aplikacija

Property name, prikazuje namjenu odnosno svojstva objekta. Na primjer, na prethodnoj slici ova dva svojstva zovu se vrijeme postavljanja i vrijeme obrade. Ti nazivi opisuju i predstavljaju njihovu funkciju. Vrijeme postavljanja imat će utjecaja na to koliko je vremena

potrebno 3D objektu prije nego što može obraditi entitet, a vrijeme obrade kontrolira i regulira koliko će dugo 3D objekt obrađivati predmet.

Zatim postoji Text box, odnosno tekstni okvir koji služi kako bi se prikazala trenutna postavka ili vrijednost svojstva. Ako se odmah znaju vrijednosti važeće za ovo svojstvo, mogu se upisati izravno u tekstni okvir. Različita svojstva imat će različite moguće postavke ili vrijednosti. Neka svojstva omogućuju korištenje fiksnog broja, neka nude mogućnost korištenje statističke distribucije za nasumično izračunavanje broja, druga dopuštaju da se a referencira drugi 3D objekt u modelu ili aktivnost/dijeljeno sredstvo u toku procesa, a neki omogućavaju referenciranje vrijednosti u oznaci ili globalnoj tablici. Na prethodnoj slici se vide tekstni okviri za vrijeme postavljanja i vrijeme obrade, koji prikazuju vrlo različite vrijednosti. Vrijeme postavljanja postavljeno je na 5, što znači da će biti potrebno 5 sekundi simulacije za postavljanje 3D objekta prije nego što obradi entitet. Nasuprot tome, vrijeme procesa postavljeno je na trokutasto (5,0, 15,0, 10,0, 0). To znači da vrijeme procesa koristi trokutastu statističku distribuciju za nasumično dodjeljivanje vremena procesa između 5 i 15 sekundi simulacije.

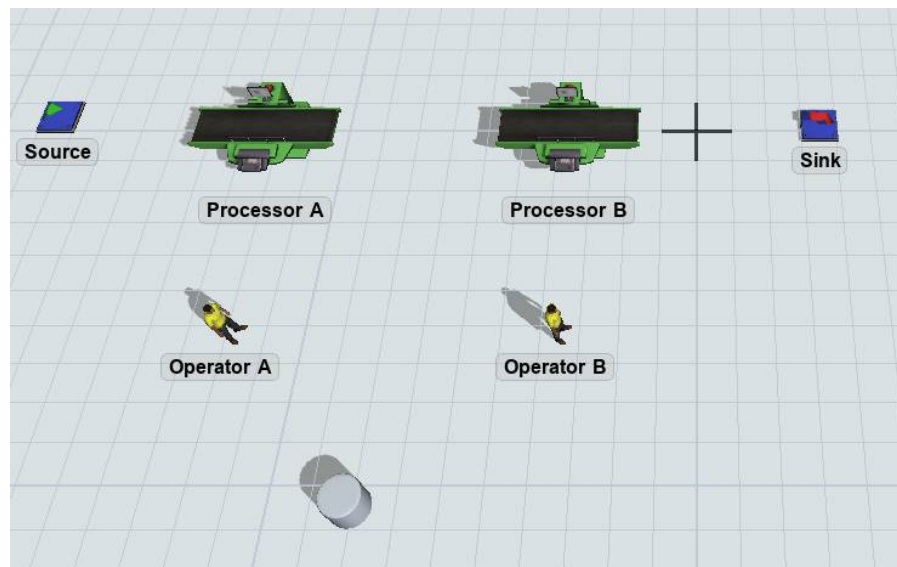
#### **4.1.3. Postupak izrade 3D modela u FlexSim-u**

Prvi korak u izgradnji 3D modela je izgradnja osnovnog izgleda poslovnog sustava koji se pokušava simulirati. Kako bi se mogli simulirati svi ti različiti elementi, u 3D model se dodaju objekti. Objekti su osnovni građevni blokovi 3D simulacijskog modela. Različite vrste objekata imaju različite svrhe i funkcije unutar simulacijskog modela.

Četiri glavna dijela svakog FlexSim modela su Source, Queue, Processor te Sink. Source (izvor) stvara entitete. Zatim queue (red čekanja) služi kako bi se spremile, odnosno pohranili entiteti dok nisu spremne za slanje drugom objektu. Potom slijedi Processor koji služi za obrađivanje entiteta, a to se najčešće prikazuje kao vremensko kašnjenje. Na kraju se nalazi Sink koji služi za uklanjanje entiteta iz simulacijskog modela. Sink-ovi se koriste kako bi se uklonili entitete koji napuštaju model.

#### 4.1.4. Primjer 3D modela u FlexSim-u

U ovom primjeru modelira se niz zadataka za utovar i istovar entiteta koje će koristiti dva različita operatera u 3D modelu. Prvi korak je otvoriti novu stranicu FlexSim-a, te na nju postaviti 1 Source, 2 Processora, 1 Sink, 2 Operatera te 1 Shape. Sve navedene objekte treba preimenovati, a to se radi klikom na svaki objekt, zatim se s desne strane u gornjem desnom kutu pojavi njegovo ime te se ono izmjeni u navedeno ili potrebno ime. Raspored objekata zajedno s imenima prikazan je na slici 15.

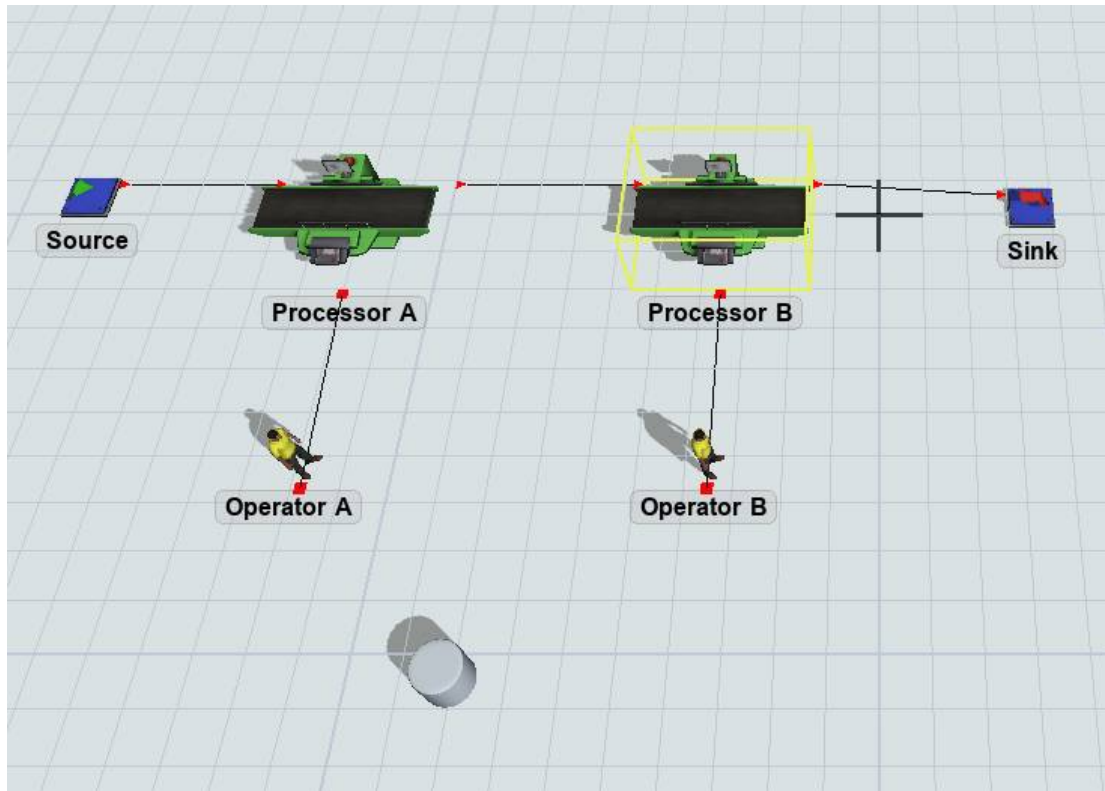


**Slika 15 – Raspored objekata modela**

Izvor: FlexSim aplikacija

Zatim slijedi povezivanje objekata. U gornjem lijevoj kutu u programu FlexSim-a nalazi se ikona slova A, klikom na strelicu uz nju otvara se izbornik i potrebno je kliknuti na Connect Objects. Strelica na ploči se mijenja u znak lanca te klikom na jedan od objekata i povlačenjem prema narednom objektu stvori se poveznica. U ovom slučaju potrebno je povezati Source sa Processorom A, Processor a sa Processorom B, Processor B sa Sink-om.

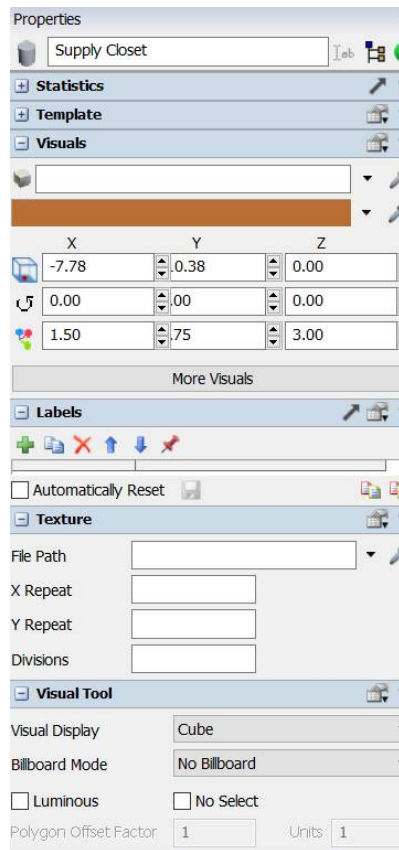
Ponovnim klikom na ikonu slova A otvara se izbornik, ali sada se bira Connect Central Ports, te s time se povezuje Processor A i Operater A, te Processor B i Operater B. Slika 16 prikazuje kako bi objekti trebali na kraju biti povezani.



**Slika 16 – Povezivanje objekata**

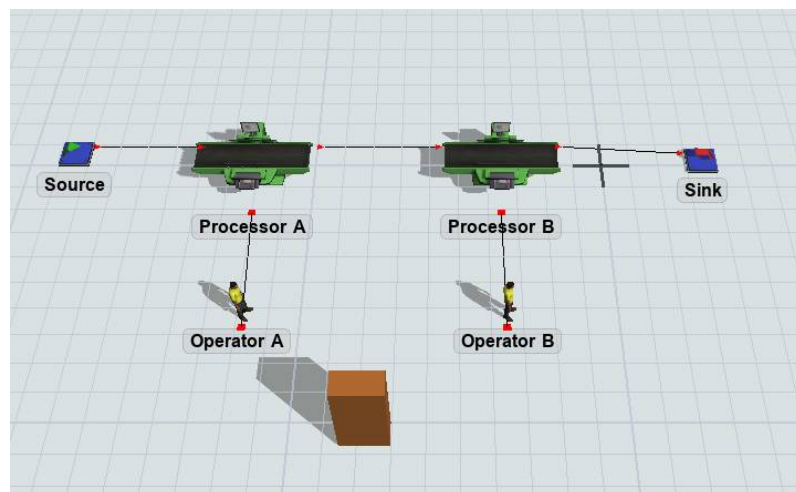
Izvor: FlexSim aplikacija

Nakon povezivanja potrebno je kliknuti na Shape Closet, a s desne strane će se pokazati njegova svojstva. Pod Visual Toll, nalazi se Visual Display menu, i od ponuđenih treba izabrati Cube. Zatim u sekciji Visual, treba kliknuti na strelicu pored Color, te izabrati tamnosmeđu. Također u sekciji Visual, pod x, y z što predstavlja poziciju, rotaciju i veličinu, potrebno je upisati: pod slovom X 1,50, pod slovom Y 0,75, a pod slovom Z 3.00. Svojstva te veličina objekta trebaju izgledati kao na slikama 17 i 18.



**Slika 17 – Svojstva i veličina objekta modela**

Izvor: FlexSim aplikacija



**Slika 18 – Prikaz modela**

Izvor: FlexSim aplikacija

## 5. FLEXSIM PROCESS FLOW

### 5.1. Process Flow grafičko sučelje

Process Flow View je glavni radni prostor gdje se gradi Process Flow.<sup>17</sup> Novi Process Flow se može stvoriti klikom na gumb Process Flow na glavnoj alatnoj traci ili korištenjem Toolbox-a. To je mjesto koje sadrži svaki dio, odnosno svaki alat FlexSim-a i u njemu se mogu vidjeti sve komponente koje su se koristile za model. Toolbox se može koristiti i za upravljanje komponentama alata.

Kada se prvotno stvori novi Process Flow, Process Flow pane će se stvoriti pored Process Flow modela. Svaki put kada se napravi novi tok procesa, on će se prikazati kao vlastiti okvir s karticama.

Najosnovnija komponenta Process Flow-a je token. Tokeni su objekti koji prolaze razne aktivnosti u toku procesa tijekom izvođenja simulacije. Funkcioniraju vrlo slično entitetima u standardnom 3D modelu. Poput entiteta, tokeni se kreću od jedne aktivnosti do sljedeće, slično kao što se entitet kreće od izvora do reda čekanja do procesora i tako dalje. Ali za razliku od entiteta, tokeni ne moraju nužno predstavljati fizički objekt koji se kreće kroz poslovni sustav. Tokeni mogu biti apstraktniji, što znači da mogu predstavljati sve što osoba poželi da predstavljaju. Na primjer, mogu predstavljati narudžbu koja se predaje, poziv pozivnom centru, logično grupiranje paleta koje se na kraju utovaraju na kamion za isporuku i tome slično. Tokeni će često biti logički povezani s fizičkim objektima u standardnom sustavu, ali ne moraju biti. Na osobi koja je izradila model je da definira prirodu veza između tokena i fizičkih elemenata u simulacijskom modelu.

Postoje tri vrste tokena. Prvi je ID kada se token stvara uz pomoć jedne od aktivnosti stvaranja tokena, automatski mu se dodjeljuje jedinstveni ID broj koji se može upotrebljavati kao referentna točka. ID broj će biti jedinstven među svim objektima toka procesa i instancama unutar FlexSim modela. Sljedeći je Name odnosno imenovanje tokena od velike je pomoći pri tome da se bolje identificira njihova svrha, funkcija ili mjesto u tijeku procesa. Tokeni ne moraju imati ime, ali im ga se može dati kada ih kreira jedna od aktivnosti izvora toka procesa. Također se može promijeniti naziv u kasnijoj fazi tijeka procesa korištenjem aktivnosti prilagođenog koda s opcijom odabira Setname ili naredbom setname. I posljednji je

---

<sup>17</sup> FlexSim priručnik – Process Flow, <https://docs.flexsim.com/en/22.2/ModelLogic/ProcessFlowBasics/OverviewProcessFlowInterface/OverviewProcessFlowInterface.html> (20.08.2022.)

Labels, koje su ključne za izgradnju složenog i dinamičnog tijeka procesa. Oznake pohranjuju prilagođene informacije o tokenu koje se mogu koristiti za utjecaj na ono što se događa s tim tokenom dok napreduje kroz tijek procesa. Razne aktivnosti mogu dodijeliti podatke oznakama tokena dok se kreće kroz aktivnosti u dijagramu toka.

Kada je Process Flow aktivan, knjižnica odnosno Library će se promijeniti kako bi prikazala sve aktivnosti i dijeljena sredstva koja se mogu dodati tijeku procesa. Aktivnosti se mogu direktno povući u Process Flow kako bi se dodale.

Aktivnost odnosno Activity je logična operacija ili korak u tijeku procesa. Kao takve, aktivnosti su osnovni građevni blokovi svakog tijeka procesa. One se povlače iz knjižnice u tijek procesa i međusobno povezuju konektorima. Kako token ulazi u aktivnost, on izvodi logiku povezanu s tom vrstom aktivnosti. To može uključivati dodjeljivanje oznaka, premještanje entiteta, odgodu tokena itd. U nekim aktivnostima, kao što je aktivnost dodjele oznaka, token će započeti i završiti aktivnost bez protoka vremena u simulacijskom satu. Tijekom pokretanja modela token neće biti vidljiv u toj aktivnosti. Međutim, korištenjem gumba Korak na kontrolnoj traci simulacije, može se vidjeti kako se token kreće kroz svaku aktivnost korak po korak. U drugim aktivnostima kao što je nabava resursa, token može biti odgođen i čekati u toj aktivnosti dok se ne ispune određeni kriteriji.

Zajednička svojstva odnosno Shared Asset, slika 19, ograničen je resurs koji tokeni mogu zatražiti ili osloboditi u određenim točkama tijeka procesa. Iako su vrlo slične aktivnostima, postoji nekoliko razlika. Prema zadanim postavkama imaju drugačiju boju od ostalih aktivnosti, kao što je prikazano na sljedećoj slici:



**Slika 19 – Shared Asset**

Izvor:

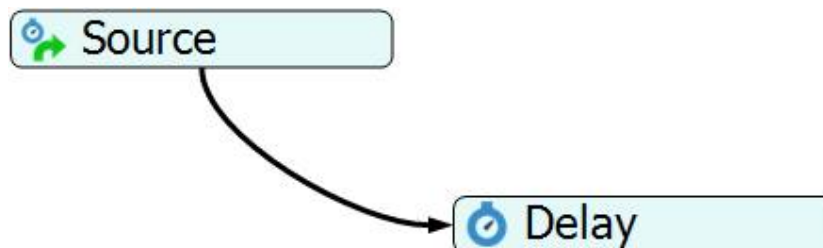
<https://docs.flexsim.com/en/22.2/ModelLogic/ProcessFlowBasics/OverviewProcessFlowInterface/OverviewProcessFlowInterface.html> (20.08.2022.)

Shared Assets može nametnuti ograničenja na tokene tako što će token čekati ako tražena svojstvo nije dostupno. Primjer ovoga iz stvarnog života bio bi određeni alat koji dijele tri različite radne stanice. Ako jedna stanica treba alat dok ga je već preuzela druga stanica, ta



stanica mora pričekati dok alat ne postane dostupan. Na isti način, ako je tokenu potrebno sredstvo za prelazak na sljedeću aktivnost, token će čekati na svojoj trenutnoj aktivnosti dok sredstvo ne bude dostupno.

Connector je spojnica između dva objekta odnosno aktivnosti, kao što je prikazano na slici 20.



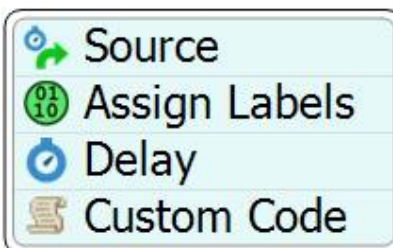
**Slika 20 – Connector**

Izvor:

<https://docs.flexsim.com/en/21.2/ModelLogic/ProcessFlowBasics/OverviewProcessFlowInterface/OverviewProcessFlowInterface.html> (20.08.2022.)

Dok simulacija traje, tokeni će koristiti konektore, odnosno Connectors kako bi mogli ići od jedne aktivnosti prema drugoj, u smjeru prema dolje. Neke aktivnosti dozvoljavaju više od jednog odlaznog konektora, ali velika većina aktivnosti to ne dopušta.

Naslagani blok ili blok je riječ koja se koristi za opisivanje skupa aktivnosti koje su spojene zajedno u jedan pomični blok, kao što je prikazano na slici 21:



**Slika 21 – Pomični blok**

Izvor:

<https://docs.flexsim.com/en/21.2/ModelLogic/ProcessFlowBasics/OverviewProcessFlowInterface/OverviewProcessFlowInterface.html> (20.08.2022.)

Složeni blokovi, odnosno Stacked Block, omogućavaju jednostavno povezivanje skupa aktivnosti u jedan niz koraka, samim tim se olakšava njihovo premještanje i uređivanje kao da su jedan blok. Kada su aktivnosti povezane zajedno u naslagani blok, one su automatski

povezane kao da je svaka aktivnost povezana konektorima. Isto tako i kada token uđe u naslagani blok, uvijek će započeti s najvišom aktivnošću, a zatim se spuštati do svake naredne aktivnosti. Sve veze koje se nalaze u takozvanom naslaganom bloku bit će povezane s najvišom aktivnošću, bez obzira na to kamo veza pokazuje u naslaganom bloku. Sve odlazne veze iz naslaganog bloka povezat će se iz donje aktivnosti.

## **5.2. Process Flow komponente**

### **5.2.1. Process Flow aktivnosti**

Process Flow se sastoji od različitih vrsta aktivnosti koje su grupirane po svojim namjenama. Prva grupa se naziva Token Creation i sastoji se od Inter - Arrival source, Schedule Source, Date Time Source, Event – Triggered Source. Njihova svrha je stvaranje tokena koji će teći kroz tijek procesa. Većina tokova procesa započinje nekom vrstom aktivnosti stvaranja tokena. Zatim slijede Basic aktivnosti, a u njima se nalazi: Assign Labels, Delay, Custom Code, Decide, Batch, Wait for Event, Create Tokens, Sink. Ove osnovne aktivnosti imaju razne vrste općih namjena u povećem broju tokova procesa. Potom se niz nastavlja na Sub Flow aktivnosti a to su: Run Sub Flow, Start, Finish. One su potrebne kako bi se stvorili i pokrenuli Sub Flow-si. Zatim postoje Visuals, a u tu grupu pripadaju: Change visuals i Run Animation. Služe kako bi se promijenili i izmijenili vizualni prikazi objekata ili pokretanja animacije u 3D modelu. Za njima slijede Objects, a u to ulaze Create Object, Move Object i Destroy Object. Ove aktivnosti mogu stvoriti, premjestiti ili uništiti objekte kao što su entiteti, fiksni resursi, izvršitelji zadataka itd. u 3D simulacijskom modelu. Potom postoje Task Sequences, a u njih se ubrajaju: Travel, Load, Unload, Delay, Travel To Lock, Custom Task, Create Task Sequence, Dispatch Task Sequence. Koriste se za izradu nizova zadataka koji se mogu dodijeliti izvršiteljima zadataka u simulacijskom modelu. Zatim slijede Shared Assets, a dijele se na List, Push To List, Pull From List, Resource, Acquire Resource, Release Resource, Variable, Get Variable, Set Variable, Zone, Enter Zone, Exit Zone. Služe za upravljanje i stvaranje popisa, resursa te zona u simulacijskom modelu. Niz nastavlja Coordination, a u to ulaze Split, Join, Synchronize. Koriste se za koordinaciju višebrojnih tokena u Process Flow-u. Potom slijedi Preemption, a u njega se ubrajaju Save token Context, Release Token, Restore Token Context. Ove se aktivnosti mogu koristiti za upravljanje sustavima preempcije u tijeku procesa. I za kraj grupacija slijedi Display. U njemu se nalaze Text, Image, Arrow. Oni su generalno bez funkcije, ali služe za vizualni dio Process Flow-a.

Aktivnosti u Process Flow-u se dodaju na razne načine, a neki od njih su sljedeći:

1. Jedan od načina je dodavanje aktivnosti iz biblioteke tijeka procesa. Provjeriti je li lijevi okvir otvoren za karticu Library, zatim pritisnuti negdje unutar tijeka procesa da bi učinili prikaz tijeka procesa aktivnim. Knjižnica će se promijeniti kako bi prikazala aktivnosti tijeka procesa. Povuci aktivnost iz knjižnice u tijek procesa.
2. Drugi način je dodavanje aktivnosti iz Quick Library. Potrebno je dvaput kliknuti u toku procesa da bi se otvorila Quick Library i zatim pritisnuti ikonu za aktivnost koju se želi dodati. Ako se želi dodati aktivnost na kraj naslaganog bloka pomoću Quick Library, potrebno je usmjeriti miš na donji rub aktivnosti ili naslaganog bloka. Ikona miša promijenit će se u kariku lanca, zatim Dvaput kliknuti na rub da se ponovno otvori Quick Library, te pritisnuti ikonu za aktivnost koju se želi dodati i aktivnost će biti dodana na kraj naslaganog bloka.
3. Treći način je dodavanje aktivnosti iz Library unutar naslaganog bloka: potrebno je povuci aktivnost iz biblioteke Process Flow u bilo koje područje tijeka procesa. Te dok se pritišće tipka miša, povuci aktivnost u naslagani blok na bilo koje mjesto u naslaganom bloku i zatim otpustiti tipku miša kada je u željenom položaju.
4. Četvrti način je dodavanje aktivnosti iz Quick Library. Usmjeriti miš na liniju između dvije aktivnosti dok linija ne postane žuta. Dvaput kliknuti redak da se ponovno otvorit Quick Library, zatim pritisnuti ikonu za aktivnost koju želite dodati. Aktivnost će biti umetnuta u sredinu naslaganog bloka. Kako bi se neka aktivnost izbrisala ili premjestila iz naslaganog bloka potrebno je pritisnuti te držati tipku Ctrl. Zatim pritisnuti aktivnost koju se želi premjestiti i povuci je na drugu poziciju u naslaganom bloku.

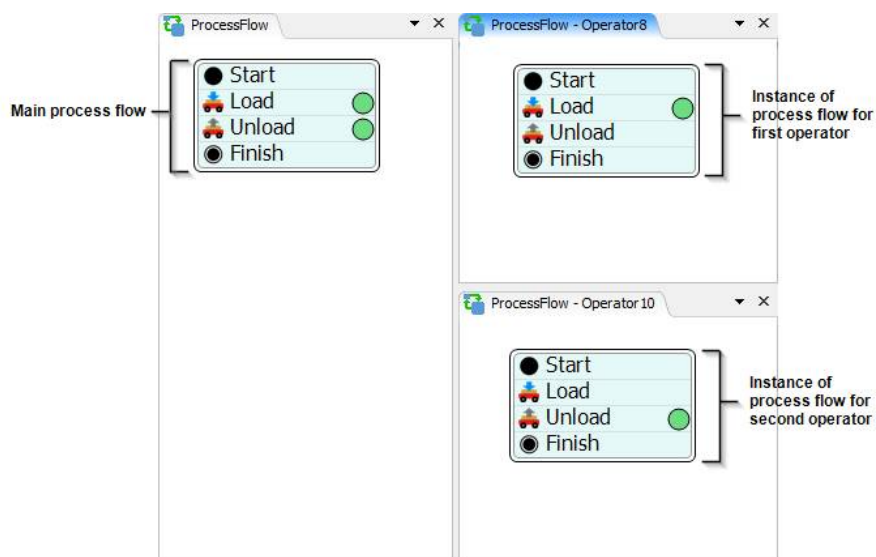
Svojstva aktivnosti mogu se uređivati i preuređivati na dva načina. Prvi je direktno u aktivnostima u Process Flow-u, a drugi je u Quick properties Pane-u.

### **5.2.2. Process Flow instance**

U alatu Process Flow, instanca je jedno pojavljivanje podtoka, fiksnog resursa procesa podtoka ili podtoka izvršitelja zadatka. Kada objekt, kao što je fiksni resurs ili izvršitelj zadatka koristi tijek procesa tijekom simulacije, postat će zasebna instanca tog toka procesa. Baš iz tog razloga, postoji mogućnost za više instanci istog toka procesa koji se izvršavaju u isto vrijeme tijekom simulacije. Koncept instanci je pomalo apstraktan. Baš zbog toga moglo

bi pomoći da ga se uspoređi s nečim konkretnim, kao što je kuhanje po receptu. Kada se izgradi bilo koja vrsta tijeka procesa osim općeg tijeka procesa, to je kao da se stvara osnovni recept za određeno jelo. Pojedinačni kuhari tada mogu sami napraviti taj recept. Svaki put kad netko napravi recept, napravi primjerak tog recepta. Nekoliko osoba može kuhati obrok koristeći recept u isto vrijeme. Ali svi koriste isti osnovni recept kao polazište. Također, ako se promijeni recept, promijenit će se sve instance koje koriste taj recept od tog trenutka nadalje. Na taj isti način, kada se stvara tijek procesa, prije svega se gradi "recept" za osnovnu logiku koja će kontrolirati objekte u osobnom simulacijskom modelu. Tijekom simulacije, razni različiti objekti mogli bi koristiti istu logiku toka procesa, ali svaki bi bio zasebna instanca. Ako se promijeni tok glavnog procesa, to će promijeniti i sve instance tog toka procesa. Vodič za niz zadataka, odnosno The Task Sequence Tutorial pruža koristan primjer instanci toka procesa. U ovom vodiču stvara se podtok koji se sastoji od jednostavnog niza zadataka za učitavanje i uklanjanje entiteta iz procesora. Model ima dva procesora, svaki sa svojim operaterom. Oba operatera koriste niz zadataka u glavnom tijeku procesa za učitavanje i istovar stavki tijeka.

Slika 22 prikazuje kako bi to izgledalo da se gleda glavni tok procesa i svaka pojedinačna instanca u isto vrijeme:



**Slika 22 – Glavni tok procesa i instance**

Izvor:

<https://docs.flexsim.com/en/21.2/ModelLogic/ProcessFlowBasics/OverviewProcessFlowInterface/OverviewProcessFlowInterface.html> (20.08.2022.)

Za prikaz instance Process Flow-a, potrebno je kliknuti na prazno mjesto u tijeku procesa kako bi se osiguralo da ništa nije odabrano. U Quick Properties pronaći grupu Attached Objects (instances). Ovo navodi objekte koji su pripojeni tijeku procesa. Zatim pritisnuti jedan od objekata na popisu da bi ga se moglo odabrati i zatim kliknuti gumb Open Process Flow View da bi se otvorio prikaz tijeka procesa za tu određenu instancu. Ovaj prikaz će prikazati samo tokene povezane s tom instancom.

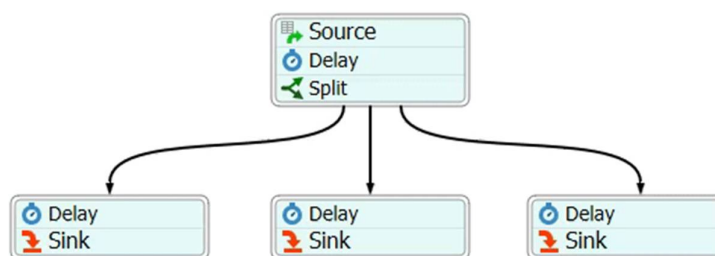
### **5.2.3. Process Flow varijable**

U računalnom programiranju varijabla je vrijednost koja se može izmjenjivati ovisno o različitim uvjetima. Varijable toka procesa rade na isti princip. U tijeku procesa može se stvoriti varijabla kako bi se imala bolja kontrola nad aspektima vlastitog simulacijskog modela koji se mogu redovito mijenjati pod različitim uvjetima. Konkretno u ovom slučaju, Process Flow varijable se najčešće primjenjuju onda kada se procesira više od jedne instance Process Flow-a. Također se mogu primjenjivati i koristiti kao centralizirano mjesto za elemente u vlastitom simulacijskom modelu koje se često mijenja. Slično kao i kod nadzornih ploča, varijable toka procesa mogu ponuditi globalno spremište u kojem se mogu brzo ažurirati entiteti procesa koje bi se možda trebale češće mijenjati od jedne simulacije do druge. Mnogi predlošci toka procesa koriste varijable toka procesa u tu svrhu. Isto tako, varijable se koriste kako bi se model dobro skalirao kada mu se pridruži nekoliko objekata Process Flow-a. Ako se priloži nekolicina različitih objekata tijeku procesa onda kada se izvodi nekoliko različitih instanci tijeka procesa, mogu se koristiti varijable tijeka procesa za brzo postavljanje varijabli koje se mogu promijeniti za svaki objekt u tijeku procesa.

### **5.2.4. Process Flow aktivnosti koordinacije**

Aktivnosti koordinacije omogućavaju jednostavan način putem kojeg se dijeli token na više tokena te koordinira vrijeme između više tokena. Oni također poboljšavaju vidljivost vlastite logike korištenjem konektora za vizualni raspored kada se tokeni stvaraju, kada čekaju jedni druge i kada se vraćaju zajedno. Aktivnosti koordinacije mogu se podijeliti u dvije grupe, a to su koordinirani nizovi zadataka te istodobni procesi. Kod koordiniranih nizova zadataka, za zajedničko izvršavanje zadataka, vrlo vjerojatno će biti potrebno više operatera. Mogu se koristiti aktivnosti koordinacije kako bi se za svakog operatera ponaosob napravio svoj token,

zatim sinkronizacija kada se završe određeni zadaci i za kraj mogu se uništiti dodatni tokeni kada je zadatak završen. Istodobni procesi mogu poslužiti kada se treba pokrenuti više procesa odjednom, gdje se trebaju odvijati istovremeno i zatim pričekati da se završe. Jedan od primjera je kada liječnik mora naručiti niz testova, čiji se uzorci šalju u odvojene laboratorije, gdje se paralelno obrađuju. Tada liječnika treba obavijestiti kada su rezultati obrađeni i spremni za pregled. Aktivnosti koordinacije mogu se koristiti za izradu tokena koji predstavlja svaki test koji bi paralelno provodio logiku testa, a zatim bi se svi tokeni ponovno spojili prije nego što liječnik pogleda rezultate. U Process Flow-u, koriste se tri koordinirane aktivnosti. Prva je Split, odnosno odvajanje. Kada token uđe u aktivnost Split, on podijeli token na više aktivnosti te ih oslobađa kroz svoje odlazne konektore. Koliko ima odlaznih konektora, toliko tokena oslobađa. Slika 23 objašnjava navedeno:



**Slika 23 – Process Flow aktivnosti koordinacije**

Izvor: <https://docs.flexsim.com/en/21.2/ModelLogic/AdditionalConcepts/Coordination/Coordination.html>  
(25.08.2022.)

Kada se izabere Split, vrlo je važan broj konektora te oznake. Broj konektora je važan zbog dolaznih i odlaznih konektora kojih bude nekolicina, dok su oznake važne zbog toga što aktivnost Split dodaje oznaku svakom novom tokenu koji sadrži referencu na već izvorni token. Sljedeća koordinirana aktivnost je Join, odnosno pridruživanje. Ona će pričekati dokle god se ne dobije token svakog od svojih dolaznih konektora prije nego se pusti jedan token. Kod odabira Join bitno je uzeti u obzir broj konektora, pregrade te Max idle i Max Wait Timers. Broj konektora se odnosi na dolazne i odlazne konektore. Aktivnost pridruživanja može imati onoliko dolaznih konektora koliko je potrebno, ali može imati samo jedan odlazni konektor. Aktivnost Join u kombinaciji s aktivnošću Split se koristi ako je potrebno imati različiti broj dolaznih u odnosu na odlazne konektore. Partitions služe kako bi se, ako je potrebno, u Join aktivnosti postavilo da se spajaju samo oni tokeni koji imaju isti ID partition. Max Idle i Max Wait Timers služi ako se želi od aktivnosti da oslobodi tokene nakon isteka maksimalnog vremena, čak i ako još nema token od svakog dolaznog konektora. I zadnja

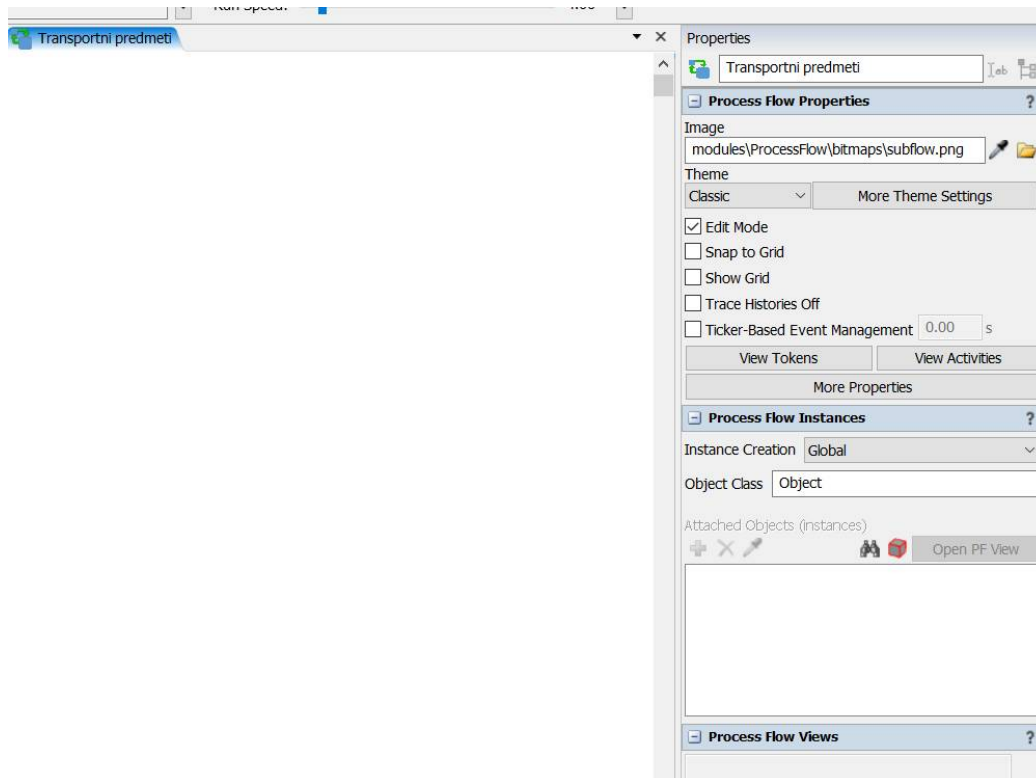
koordinirana aktivnost je Synchronize. Aktivnost sinkronizacije čeka dok ne dobije token od svakog od svojih dolaznih konektora, a zatim otpušta sve tokene u isto vrijeme. Aktivnost sinkronizacije slična je aktivnosti pridruživanja, ali umjesto da otpušta samo jedan token, ona otpušta sve tokene istovremeno.

### **5.3. Primjer Process Flow modela**

U ovom modelu Process Flow-a izradit će se primjer sa jednostavnim nizom zadataka koji će oba operatera koristiti kada god trebaju učitati ili isprazniti stavku toka. Nadalje će se napraviti sub flow jer putem njega Process Flow može najjednostavnije koristiti sub flow za dodjeljivanje sekvenci zadataka onima koji izvršavaju zadatke.

Za početak prvo će se postaviti tok procesa da pokrene zasebnu instancu svaki put kada operater pokrene podtok. To znači da će svaki operater koristiti isti podtok kako bi im pružio osnovni predložak za niz zadataka. Ali svaki će operater pokrenuti vlastitu instancu toka procesa, što je kao pokretanje vlastite kopije toka procesa. Jedna od prednosti ove metode je da će sva zajednička sredstva koja koriste operateri biti lokalna za dva operatera.

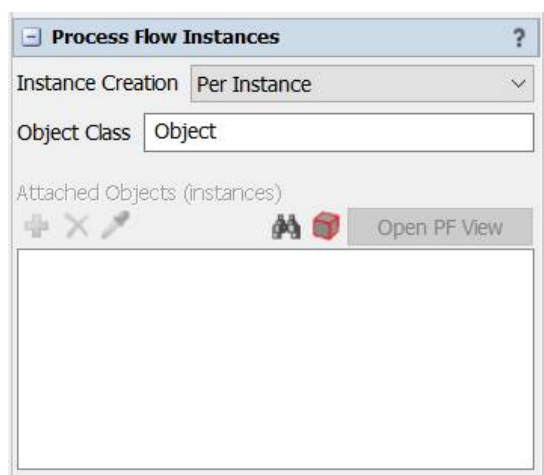
Na glavnoj alatnoj traci u FlexSim-u, na sredini na vrhu stranice potrebno je kliknuti na tipku Process Flow, zatim na add a Sub Flow te ga na taj način otvoriti. Kada se Process Flow otvori, na desnoj strani se otvore njegova svojstva. Prvo je potrebno promijeniti njegovo ime, iz Sub Flow u npr. Transporti predmeti.



**Slika 24 – Process Flow svojstva**

Izvor: FlexSim aplikacija

Zatim pod Process Flow Instance, potrebno je otvoriti Instance Creation meni te izabrati Per Instance, slika 25.

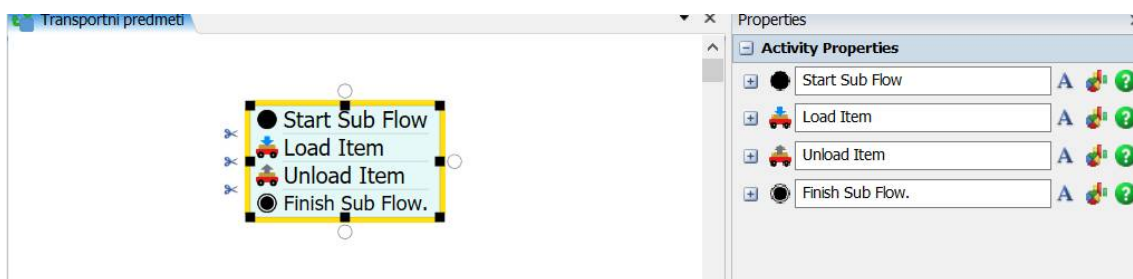


**Slika 25 – Process Flow instance**

Izvor: FlexSim aplikacija



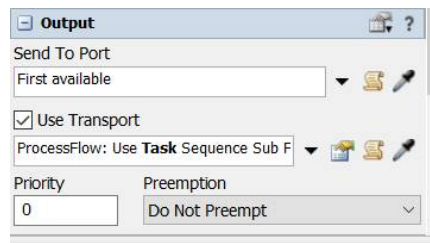
Iz Library, na lijevoj strani potrebno je dodati objekte u Process Flow. Prvo se dodaje start, koji se naziva u Sub Flow grupi. Zatim se dodaje Load, koji se nalazi u Task Sequence grupi. Slijedi Unload, koji se nalazi također u Task Sequence grupi. I za kraj se dodaje Finish, koji se nalazi pod Sub Flow grupom objekata. Svaki objekt od navedenih se postavlja jedan ispod drugog. Kako bi bilo jednostavnije i lakše svakom objektu je potrebno promijeniti ime. Klikom na navedene objekte s desne strane se pojave njihova imena. Start promijeniti u Start Sub Flow, Load u Load Item, Unload u Unload Item, a Finish u Finish Sub Flow, slika 26.



**Slika 26 – Objekti u Process Flow-u**

Izvor: FlexSim aplikacija

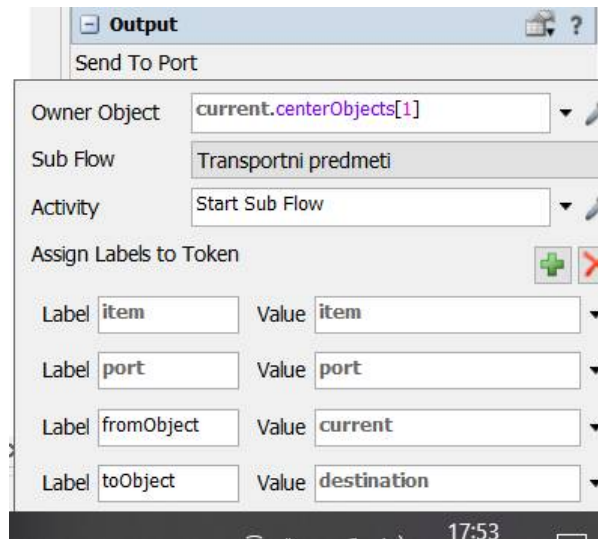
Zatim je potrebno povezati Sub Flow sa objektima simulacije modela. Na FlexSim modelu, kliknuti na Processor A kako bi se s desne strane pokazala njegova svojstva. Pod sekcijom Output nudi se Use transport, potrebno je kliknuti na njega te ga označiti kvačicom. Uz Use transport nalazi se strelica, a klikom na nju se otvara meni. Potrebno je izabrati Customized Logic, a zatim Process Flow: Use Task Sequence Sub Flow kako bi se otvorile daljnje opcije slika 27.



**Slika 27 – Task Sequence Sub Flow**

Izvor: FlexSim aplikacija

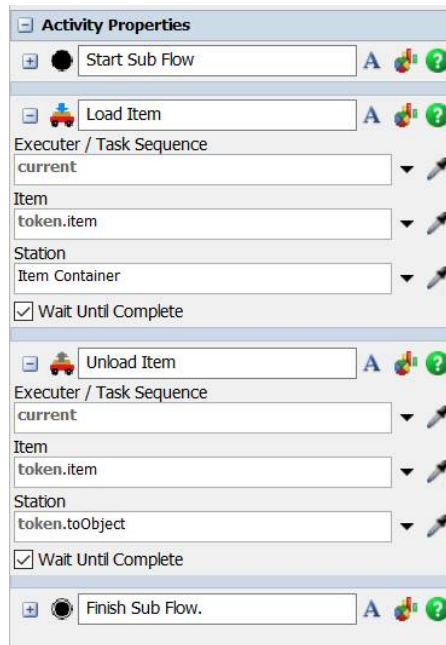
Kada se otvore daljnje opcije treba kliknuti na strelicu pored Owner Object kako bi se ponovno otvorio meni. Zatim treba provjerite je li odabrani Sub Flow u okviru Sub Flow TransportItems, što je podtok stvoren u prethodnom koraku. I za kraj, na otvorenom meniju pod Activity treba izabrati Start Sub Flow. Meni treba izgledati kao na slici 28.



**Slika 28 – Activity meni**

Izvor: FlexSim aplikacija

Isto se treba napraviti i za Processor B. Klikom na naslagani blok u Process Flow-u, na desno se otvore njihova svojstva. Klikom na plus pored Load Item, otvore se njegova svojstva. Pod executer/Task Sequence treba kliknuti na strelicu i zatim odabrati current(Intance Object). Taj isti postupak ponoviti i za Unload Item i kod Unload Item i kod Load Item u kućici Item, treba pisati token, item. Kod Unload Item, u kućici Station, treba izbrisati tekst te napisati token.toObject. Njihova svojstva na kraju trebaju izgledati kao na slici 29.



**Slika 29 – Activity svojstva**

Izvor: FlexSim aplikacija

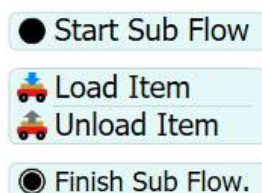
Kako bi se model pokrenuo, potrebno je kliknuti na reset, a zatim run, u gornjem desnom kutu FlexSim-a. Model treba raditi dokle god oba operatera ne krenu raditi sa sekvencama, odnosno sa loading task sequences. Kliknuti na prazno polje Process Flow-a kako bi se uvjerali da ništa nije trenutno odabrano. Klikom na prazno polje s desne strane se otvore svojstva Process Flow-a. Pod Process Flow Instances, trebaju se prikazivati dva operatera pod Attached Objects. Treba kliknuti na operator 1 kako bi ga se izabralo. Zatim kliknuti na Open PF View te će se time otvoriti instance Process Flow-a za operatora 1. Isti taj postupak treba napraviti i za processor 2, te nakon toga ponovno resetirati model a zatim ga ponovno pokrenuti.

Dok simulacija radi, uočljivo je kako operateri učitavaju i istovaruju entitete u 3D modelu. U tijeku procesa TransportItems, vidi se kako se jedan ili dva tokena kreću kroz tok procesa svaki put kada operater učitava i istovaruje entitet. Također je uočljivo i da je svaka od dvije instance toka procesa operatora u korelaciji s operatorima u 3D modelu i s tokom procesa TransportItems.

S obzirom da je do sada izrađeni Process Flow jako jednostavan, u narednim zadacima dodati će se dodatni prilagođeni zadaci koji znatno olakšavaju dodjeljivanje složenih nizova zadataka mobilnim objektima. Kako bi dodali prvu narednu aktivnost potrebno je stisnuti reset u

gornjem lijevom kutu FlexSim-a. Kako nebi došlo do zabune, potrebno je zatvoriti i dvije instance Process Flow-a te samo pustiti prvu, TransporterItems.

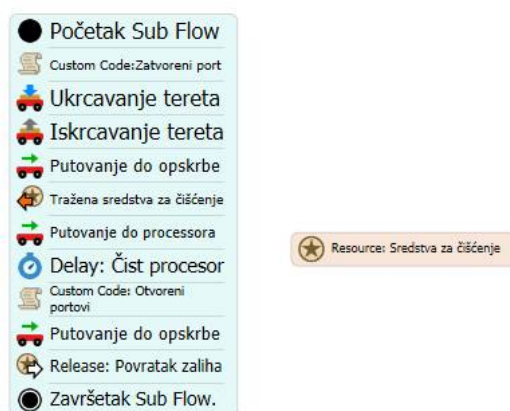
U Process Flow-u kliknuti na blok aktivnosti. Sa lijeve strane bloka se prikazala ikona škara. Tim škarama potrebno je odvojiti Start Sub Flow, te Finish Sub Flow. Blok na kraju treba izgledati kao na slici 30.



**Slika 30 – Izgled bloka**

Izvor: FlexSim aplikacija

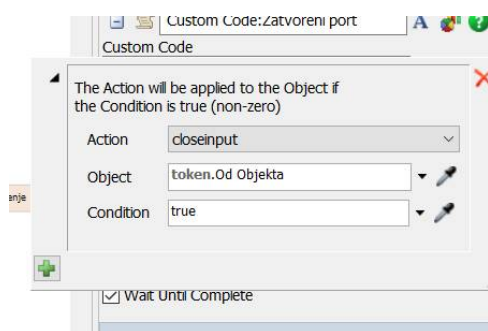
Pod Library, zatim Basic, treba povući Custom Code aktivnost ispod Start Sub Flow, a iznad Load Item kako bi se blok ponovo spojio. Ispod Unload Item, treba ovim redoslijedom dodati naredne aktivnosti: Travel, nalazi se pod Task Sequence. Zatim Acquire Resource, koji se nalazi pod Shared Assets. Zatim ponovno Travel. Nakon toga Delay, koji se nalazi pod Basic. Zatim Custom Code, koji se nalazi također pod Basic. Za njim slijedi Travel, koji se nalazi pod Task Sequence. I za kraj Release Resource, koji se nalazi pod Shared Assets. I ponovno spojiti Finish Sub Flow sa ostatkom bloka, slika 31.



**Slika 31 – Prilagodeni blok**

Izvor: FlexSim aplikacija

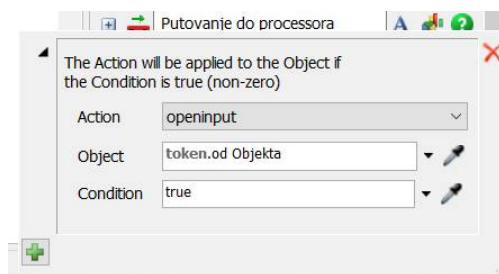
Klikom na blok aktivnosti za početak će se urediti dva Custom code-a tako da zatvaraju i otvaraju portove procesora u pravoj točki u nizu zadataka. Treba kliknuti na plus uz Custom code kako bi se otvorila njegova svojstva, zatim kliknuti na zeleni plus kako bi se otvorio meni. Izabrati control, zatim close and open ports kako bi se otvorio još jedan meni. Pod Action pustiti closeinput tako da se zatvori dolazni port trenutnog procesora. Zatim pored Object kliknuti na strelicu kako bi se otvorio meni. Izabrati token.labelName promijeniti label name u ime from object. Ta postavka će napraviti dinamičku referencu na objekt naveden u oznaci fromObject tokena, slika 32.



**Slika 32 – Custom code: Zatvoreni port**

Izvor: FlexSim aplikacija

Isti postupak potrebno je napraviti i za Custom code: Otvoreni portovi. Samo kod Action, umjesto closeinput staviti openinput, slika 33.

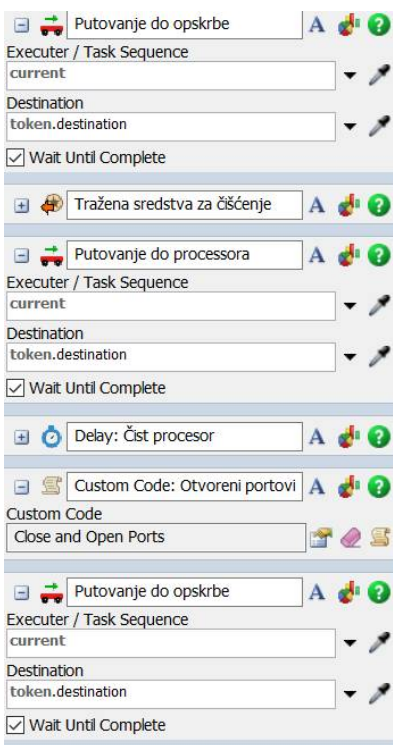


**Slika 33 – Custom code: Otvoreni portovi**

Izvor: FlexSim aplikacija


Sada treba urediti postavke za svaki Travel, u ovom slučaju ima ih čak tri. U svojstvima Putovanje do opskrbe treba kliknuti na strelicu uz Executer/Task Sequence kako bi se otvorio

meni, a u njemu izabrati current/Instance object. Isti postupak napraviti i za naredna dva Travel-a, slika 34.

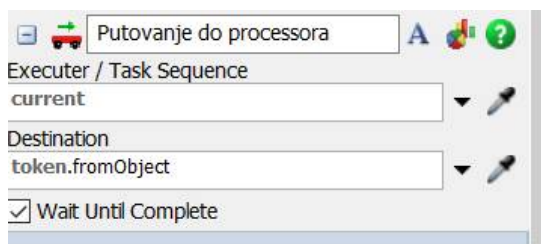


**Slika 34 – Travel instance**

Izvor: FlexSim aplikacija

Zatim treba još dodatno urediti svojstva Travel-a. U svojstvima kod prvog Travel, odnosno Putovanja do opskrbe, pored Destination nalazi se ikona  i nakon što se klikne na nju potrebno je kliknuti na Supply Closet u 3D modelu te izabrati Supply Closet. Isti postupak ponoviti i za drugi Travel, odnosno Putovanje do opskrbe.

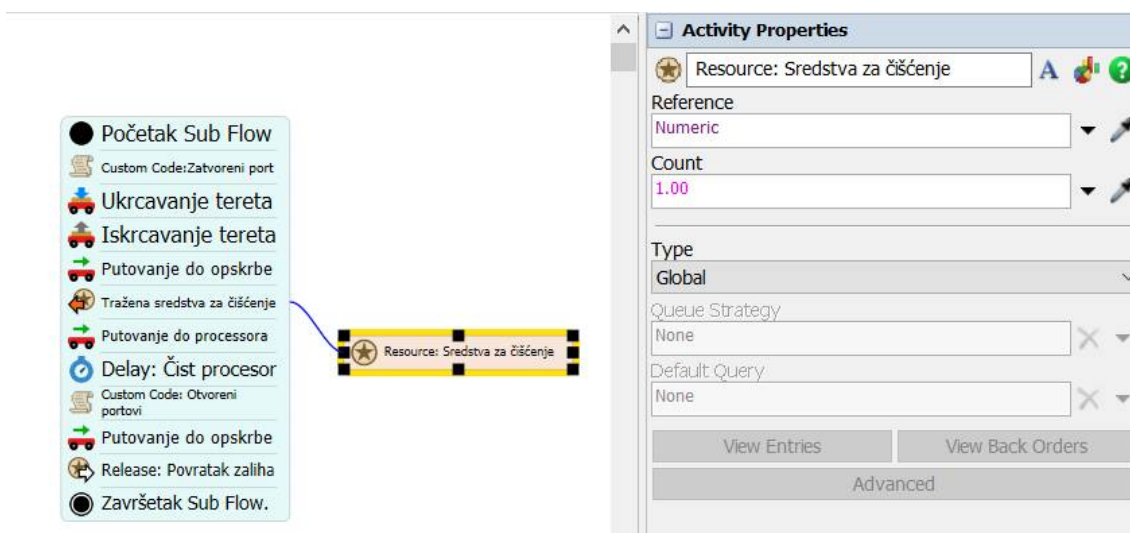
Pod svojstvima Travel to processor, odnosno Putovanje do Processora, treba kliknuti na Destination te upisati token.fromObject kako bi se napravila dinamička referenca na objekt naveden u oznaci tokena fromObject, slika 35.



**Slika 35 – Dinamička referenca**

Izvor: FlexSim aplikacija

Nadalje, treba spojiti Resource sa Process Flow modelom. Dok je blok odabran, treba kliknuti na uskličnik uz Acquire Cleaning Supplies kako bi se izabrao model te zatim kliknuti na Resource. Na taj način se stvori linija koja povezuje ta dva objekta. I zatim kliknuti na Resource, time će se otvoriti njegova svojstva i pod Type izabrati Global, slika 36.



**Slika 36 – Povezivanje objekata**

Izvor: FlexSim aplikacija

## 5.4. Prednosti i nedostaci FlexSim Process Flow-a

### Prednosti

Flexsim Process Flow nudi određene prednosti, a prva među njima je standardizacija vlastitih procesa. Ona pruža korisnicima uspostavljen i vremenski provjeren proces za korištenje i zbog toga se naglašava naspram drugih prednosti. Sljedeća prednost je povećanje resursa koja služi kako bi se nakon prepoznavanja komponenata procesa resursi mogli lakše distribuirati za maksimalnu izvedbu. Također jedna od prednosti je poboljšanje vlastitih procesa koji se javljaju kada sastavnice zajedno služe istom cilju, pri čemu je sljedeći korak njihovo unaprjeđenje. Još neke od prednosti su smanjivanje rizika, gdje određeni tokovi procesa mogu pomoći pri učinkovitijem otkrivanju i sprječavanju rizika te povećavanje transparentnosti u organizaciji. Pri tome svaki zaposlenik u organizaciji ima mogućnost uvida u jasnije upute o tome što se radi i kako se radi.

### Nedostaci

Pri korištenju FlexSim Process Flow-a postoje određeni nedostaci, a prvi među njima su izmjene. Njihova prva mana je što mogu postati otežavajuće pri korištenju dijagrama toka. Tome u prilog ide činjenica da pri izmjenama, odnosno kada je proces potrebno prilagoditi ili promijeniti iz određenih razloga, postoji vjerojatnost da će se morati ponovo nacrtati cijeli dijagram toka. Iz tog razloga to dovodi do vremenskog i financijskog gubitka te zbog toga se ovo smatra najvećim nedostatkom FlexSim Process Flow-a.

Zatim slijedi logička složenost koja se javlja kada je proces kompliciran i zahtjevan, što rezultira da dijagram toka procesa može biti neuredan i teško shvatljiv. Takav način rada otežava razumijevanje samog procesa te se to manifestira nedostatkom vremena u pokušaju određivanja točnog procesa ili područja. Zbog tog razloga je važno pregledati proces te uvidjeti postoje li dijelovi procesa koji se mogu ukomponirati zajedno ili postaviti na jednostavniji način.

I kao posljednji nedostatak koji se može odnositi na FlexSim Process Flow je reprodukcija dijagrama toka. Određeni simboli dijagrama se ne mogu upisati u proces i zbog toga je potrebno korištenje drugih programa kao što su Word, Excel ili tome slični programi koji omogućuje stvaranje novih simbola te umetanje riječi u njih. Upravo iz tog razloga takav oblik rada je otežavajući jer kako bi se mogao stvoriti određeni oblik potrebno je koristiti više vrsta programa što zahtijeva i više uloženog vremena.



## 6. ZAKLJUČAK

Simulacija je pokušaj kopiranja svojstava, izgleda i značajki realnoga sustava eksperimentiranjem, donošenjem zaključaka i provođenjem odluka temeljem rezultata simulacije. Razvitkom računalne znanosti, modeliranje i simulacije koriste se u sve većem broju projekata u raznim granama industrije i znanstvenog istraživanja. Brz razvoj, očitovao se i na napretku softverskih aplikacija za izradu simulacijskih modela što je omogućilo širokom krugu korisnika razvoj svojeg simulacijskog modela temeljenog na njihovim potrebama.

Cilj svih simulacijskih aplikacija je provesti eksperiment unutar same aplikacije, a ne u realnom vremenu i prostoru, kako bi se utvrdilo koliko je isplativije provesti planiranu promjenu kako bi se omogućila uštedu u vremenu i ulaganju. Postoje razni simulacijski alati koji se ističu u cijeni, kvaliteti, načinu i lakoći korištenja, dobroj podršci itd. FlexSim je analitički alat za prepoznavanje problema i procjenu alternativnih rješenja u dizajnu i radu stvarnog fizičkog sustava. U FlexSim-u se može izgraditi 3D model stvarnog fizičkog sustava, a zatim se sustav može proučavati u kraćem vremenskom periodu i uz djelić cijene stvarnog fizičkog sustava.

Korištenje dodatnih FlexSim alata, koji su integrirani unutar same aplikacije, kao što je Process Flow predstavlja brži i točniji proces konstrukcije modela te stoga omogućuje razvoj svih detalja, parametara i procjenu glavnih pokazatelja koji karakteriziraju stvarni fizički sustav. Korištenje dodatnog alata Process Flow omogućuje se razvoj preciznih i učinkovitih simulacijskih modela za različite namjene. Process Flow model omogućuje točan odabir najboljih parametara i procjenu glavnih pokazatelja koji karakteriziraju proizvodni proces. Rezultati simulacije mogu se koristiti u procesu planiranja i istovremeno mogu doprinijeti povećanju učinkovitosti proizvodnog procesa.

## LITERATURA

### KNJIGE

1. Banks, J., Carson, J.S.II, Nelson, B.L., Nicol, D.M.: *Discrete-Event System Simulation, 5th edition*, Pearson Education, Inc., Upper Saddle River, New Jersey, USA 2010.
2. Čerić, V.: *Simulacijsko modeliranje*, Školska knjiga, Zagreb, 1993.
3. Law, A.M., Kelton, W.D.: *Simulation Modelling and Analysis, 2nd edition*, McGraw-Hill, Inc., Singapore, 2000.
4. Robinson, S.: *Simulation: The Practice of Model Development and Use*, John Wiley & Sons, New York, 2004.
5. Schmidt, J.W., Taylor, R.E.: *Simulation and Analysis of Industrial Systems*, Richard D. Irwin, Inc., Homewood, USA, 1970.

### POGLAVLJE U KNJIZI

1. Božikov, J.: *Modeliranje i simulacija*, poglavlje u knjizi Medicinskoinformatičke metode, Medicinski fakultet Sveučilišta u Zagrebu i Medicinska Naklada, Zagreb, 2007., 85-108.

### ČLANCI U ČASOPISIMA

1. Banks, J.: *Introduction to simulation*. In Proceedings of the 31st conference on Winter simulation: Simulation - a bridge to the future - Volume 1 (WSC '99), Association for Computing Machinery, New York, NY, USA, 7–13. 1999.  
<https://doi.org/10.1145/324138.324142>
2. Mourtzis, D.: *Simulation in the design and operation of manufacturing systems: state of the art and new trends*, International Journal of Production Research, Vol. 58, No. 3, 1–23, 2019. doi:10.1080/00207543.2019.1636321
3. Nordgren, W. B.: *Flexsim simulation environment*. Proceedings of Winter Simulation Conference, Phoenix, Arizona, USA. IEEE, 197-200 Vol. 1. 2003.

### ELEKTRONIČKI IZVORI

1. AnyLogic, <https://www.anylogic.com/> (15.07.2022.)
2. ExtendSim, <https://extendsim.com/products/line/overview> (15.07.2022.)
3. FlexSim, <https://www.flexsim.com/flexsim/> (17.07.2022.)

4. FlexSim priručnik,  
<https://docs.flexsim.com/en/22.2/FlexSimUI/OverviewUserInterface/OverviewUserInterface.html> (25.07.2022.)
5. FlexSim priručnik – Process Flow,  
<https://docs.flexsim.com/en/22.2/ModelLogic/ProcessFlowBasics/OverviewProcessFlowInterface/OverviewProcessFlowInterface.html> (20.08.2022.)
6. Diskretne-simulacije, <http://www.efos.unios.hr/poslovne-simulacije/wp-content/uploads/sites/180/2013/04/Predavanje-4-Diskretne-simulacije.pdf> (11.07.2022.)
7. Informacijske mreže - Simulacija diskretnih događaja,  
[https://www.fer.unizg.hr/\\_download/repository/INFMRE-2019\\_05.pdf](https://www.fer.unizg.hr/_download/repository/INFMRE-2019_05.pdf) (10.07.2022.)
8. Rockwell Automation Arena, <https://www.rockwellautomation.com/en-us/products/software/arena-simulation.html> (15.07.2022.)

## POPIS SLIKA

<b>Slika 1 – Simulacijski proces</b> .....	13
<b>Slika 2 – ExtendSim Pro</b> .....	17
<b>Slika 3 - Rockwell Automation Arena</b> .....	17
<b>Slika 4 – AnyLogic</b> .....	18
<b>Slika 5 – FlexSim</b> .....	20
<b>Slika 6 – Izgled radnog okruženja simulacijskog softvera FlexSim</b> .....	21
<b>Slika 7 – Vremenska tablica</b> .....	23
<b>Slika 8 – MTBF/MTTR</b> .....	24
<b>Slika 9 – Dashboard</b> .....	25
<b>Slika 10 – Statistics Collector</b> .....	26
<b>Slika 11 – Flow Item Bin</b> .....	27
<b>Slika 12 – FlexSim properties</b> .....	28
<b>Slika 13 – FlexSim properties objekta</b> .....	29
<b>Slika 14 – Prikaz simulacijskog procesa</b> .....	29
<b>Slika 15 – Raspored objekata modela</b> .....	31
<b>Slika 16 – Povezivanje objekata</b> .....	32
<b>Slika 17 – Svojstva i veličina objekta modela</b> .....	33
<b>Slika 18 – Prikaz modela</b> .....	33
<b>Slika 19 – Shared Asset</b> .....	35
<b>Slika 20 – Connector</b> .....	36
<b>Slika 21 – Pomični blok</b> .....	36
<b>Slika 22 – Glavni tok procesa i instance</b> .....	39
<b>Slika 23 – Process Flow aktivnosti koordinacije</b> .....	41
<b>Slika 24 – Process Flow svojstva</b> .....	43
<b>Slika 25 – Process Flow instance</b> .....	43
<b>Slika 26 – Objekti u Process Flow-u</b> .....	44
<b>Slika 27 – Task Sequence Sub Flow</b> .....	44
<b>Slika 28 – Activity meni</b> .....	45
<b>Slika 29 – Activity svojstva</b> .....	46
<b>Slika 30 – Izgled bloka</b> .....	47
<b>Slika 31 – Prilagođeni blok</b> .....	47
<b>Slika 32 – Custom code: Zatvoreni port</b> .....	48
<b>Slika 33 – Custom code: Otvoreni portovi</b> .....	48

<b>Slika 34 – Travel instance</b> .....	49
<b>Slika 35 – Dinamička referenca</b> .....	50
<b>Slika 36 – Povezivanje objekata</b> .....	50