PRIMJENA SIMULACIJSKOG MODELIRANJA U PLANIRANJU PROIZVODNJE

Mijolović, Toni

Master's thesis / Diplomski rad

2023

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

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:190:734703

Rights / Prava: Attribution 4.0 International/Imenovanje 4.0 međunarodna

Download date / Datum preuzimanja: **2024-11-30**

Repository / Repozitorij:

Repository of the University of Rijeka, Faculty of Engineering

SVEUČILIŠTE U RIJECI

TEHNIČKI FAKULTET

Diplomski sveučilišni studij strojarstva

Diplomski rad

PRIMJENA SIMULACIJSKOG MODELIRANJA U PLANIRANJU PROIZVODNJE

Rijeka, srpanj 2023.

Toni Mijolović

0069077742

SVEUČILIŠTE U RIJECI

TEHNIČKI FAKULTET

Diplomski sveučilišni studij strojarstva

Diplomski rad

PRIMJENA SIMULACIJSKOG MODELIRANJA U PLANIRANJU PROIZVODNJE

Mentor: prof. dr. sc. Mladen Perinić

Komentor: doc. dr. sc. David Ištoković

Rijeka, srpanj 2023.

Toni Mijolović

0069077742

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
POVJERENSTVO ZA DIPLOMSKE ISPITE

Rijeka, 15. ožujka 2022.

Zavod: **Zavod za industrijsko inženjerstvo i menadžment**
Predmet: **Računalna simulacija proizvodnih procesa**
Grana: **2.11.03 proizvodno strojarstvo**

ZADATAK ZA DIPLOMSKI RAD

Pristupnik: **Toni Mijolović (0069077742)**
Studij: Diplomski sveučilišni studij strojarstva
Modul: Tehnološko informatičko inženjerstvo

Zadatak: **Primjena simulacijskog modeliranja u planiranju proizvodnje / Application of simulation modeling in manufacturing planning**

Opis zadatka:

Prikazati prednosti i nedostatke metoda simulacije. Klasificirati metode simulacije. Opisati osnovne značajke simulacije diskretnih događaja i kontinuirane simulacije. Primijeniti simulacijske metode na primjeru proizvodnje dijelova. Provjeriti ispravnost rada izrađenog simulacijskog modela. Na kraju je potrebno obrazložiti dobivene rezultate na osnovi provedenih simulacijskih eksperimenata.

Rad mora biti napisan prema Uputama za pisanje diplomskih / završnih radova koje su objavljene na mrežnim stranicama studija.

Zadatak uručen pristupniku: 21. ožujka 2022.

Mentor:

Prof. dr. sc. Mladen Perinić

Doc. dr. sc. David Ištoković (komentor)

Predsjednik povjerenstva za diplomski ispit:

Prof. dr. sc. Kristian Lenić

IZJAVA

U skladu s *Pravilnikom o diplomskom radu, diplomskom ispitu i završetku diplomskih sveučilišnih studija* Tehničkog fakulteta Sveučilišta u Rijeci izjavljujem da je diplomski rad izrađen samostalno pod vodstvom mentora prof. dr. sc. Mladena Perinića i komentora doc. dr. sc. Davida Ištokovića.

Rijeka, srpanj 2023.

Toni Mijolović
0069077742

ZAHVALA

Veliko hvala mentoru prof. dr. sc. Mladenu Periniću i komentoru doc. dr. sc. Davidu Ištokoviću na stručnoj pomoći, podršci i savjetovanju tijekom pisanja ovog diplomskog rada. Hvala vam na prenesenom znanju.

Zahvalan sam svim mojim kolegama i prijateljima koji su olakšali i uljepšali moje studentske dane.

Neizmjerno hvala mojoj obitelji na podršci u svakom smislu te riječi.

SADRŽAJ

1. UVOD 1

2. RAČUNALNA SIMULACIJA 2

2.1. OSNOVE ZA IZRADU SIMULACIJSKOG MODELA 2

2.1.1. KORACI POTREBNI ZA IZRADU SIMULACIJSKOG MODELA 3

2.1.2. PRAVILA IZRADE SIMULACIJSKOG MODELA 4

2.2. PREDNOSTI I NEDOSTATCI RAČUNALNIH SIMULACIJA 5

2.3. PODJELA SIMULACIJSKIH MODELA 5

2.3.1. SIMULACIJA DISKRETNIH DOGAĐAJA 6

2.3.2. KONTINUIRANA SIMULACIJA 7

3. PRIMJENA SIMULACIJSKIH MODELA U PROIZVODNJI DIJELOVA 8

3.1. IDEJNI PROIZVODNI PROCES 9

4. IZRADA SIMULACIJSKOG MODELA 12

4.1. DETALJNI POSTUPAK IZRADE 12

4.1.1. DEFINIRANJE VRSTA PROIZVODA 12

4.1.2. KREIRANJE TABLICA ZA UNOS ULAZNIH PODATAKA 14

4.1.3. KODIRANJE LOGIČKIH METODA - RAZVOJ 17

4.1.4. KODIRANJE LOGIČKIH METODA - UNIVERZALNA VERZIJA 21

4.1.5. SLAGANJE PROSTORNIH ELEMENATA 25

4.1.6. POSTAVKE PROSTORNIH ELEMENATA 26

4.2. VERIFIKACIJA I VALIDACIJA SIMULACIJSKOG MODELA 29

4.2.1. POSTUPAK VERIFICIRANJA SIMULACIJSKOG MODELA 29

4.2.2. POSTUPAK VALIDIRANJA SIMULACIJSKOG MODELA 35

4.3. UNOS IDEJNOG PROBLEMA U RAČUNALNI SIMULACIJSKI MODEL 40

4.3.1. UNOS I ODREĐIVANJE TOKA MATERIJALA 41

4.3.2. INICIJALNI TOK MATERIJALA 42

5. INTERPRETACIJA I ANALIZA REZULTATA 43

5.1. OPĆA ANALIZA REZULTATA 43

5.2. RJEŠENJE IDEJNOG PROBLEMA 46

5.2.1. PRVO POBOLJŠANJE TOKA MATERIJALA 47

5.2.2. DRUGO POBOLJŠANJE TOKA MATERIJALA 48

5.2.3. KONAČNI TOK MATERIJALA 49

5.2.4. RJEŠENJE I ODGOVOR NA UPIT 50

6. ALTERNATIVNO RJEŠENJE 52
6.1. KANBAN 52
6.2. IZRADA SIMULACIJSKOG MODELA 53
7. ZAKLJUČAK 60
LITERATURA 63
POPIS SLIKA 64
POPIS TABLICA 66
POPIS OZNAKA I KRATICA 67
SAŽETAK 69
SUMMARY 70

1. UVOD

U današnje vrijeme računalne simulacije predstavljaju moćne alate za brojna složena ispitivanja. Pomoću njih može se predvidjeti ponašanje nekog sustava na temelju obrazaca ponašanja koji su prethodno zabilježeni. Mogu se primjenjivati u gotovo svim granama strojarstva, a svojim mogućnostima mogu se izvrsno uklopiti u planiranje proizvodnje.

Primjer iz prakse upućuje na to da se računalna simulacija slabo primjenjuje u planiranju proizvodnje. Polazeći od stava da se pomoću računalnih simulacija mogu utvrditi i ispitati mnogi problemi koji isprva nisu vidljivi prilikom planiranja proizvodnje te da se pomoću simulacije značajno može unaprijediti proces planiranja, u radu se nastoji ideju o računalnoj simulaciji u planiranju proizvodnje približiti krajnjem korisniku.

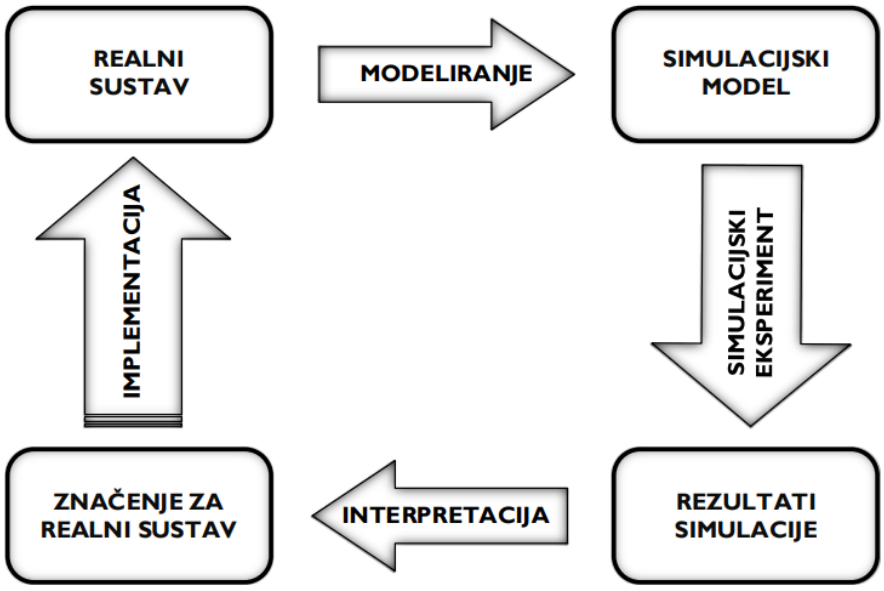
Ovaj rad ima za cilj postići jednostavan pristup krajnjeg korisnika korištenju računalne simulacije u proizvodnji. U radu će se objasniti osnovna funkcija simulacija te njihova podjela, između ostalog na kontinuirane simulacije i simulacije diskretnih događaja na koje će se staviti naglasak. Dakle, govorit će se o općim pravilima koje treba primjenjivati prilikom izrade simulacijskih modela i o pravilima kako primijeniti simulacijske modele i implementirati njihove rezultate te o važnosti simulacija, kako u proizvodnji, tako i šire.

Računalne simulacije karakteriziraju brojne prednosti i nedostatci. Najznačajnija je prednost omogućavanje ispitivanja rada nekog realnog sustava bez ometanja ili zaustavljanja njegovog rada u realnom okruženju. Naravno, postoje i određeni nedostatci, ponajprije kompleksnost izrade simulacijskog modela. Taj glavni nedostatak pokušat će se pojednostaviti i umanjiti kroz ovaj rad te korisniku omogućiti samostalnu izradu simulacijskog modela i njegovo jednostavno korištenje. Osim objašnjenja same izrade simulacijskog modela, provest će se postupci verifikacije i validacije simulacijskog modela te prikazati rješavanje ovog problema na konkretnom primjeru. Izrada simulacijskog modela te provođenje simulacijskih eksperimenata izvodit će se pomoću softvera *Tecnomatix Plant Simulation 15* koji je razvila tvrtka *Siemens*.

2. RAČUNALNA SIMULACIJA

Računalna simulacija vrlo je moćan alat u planiranju proizvodnje. Njom se oponaša rad realnog procesa u određenom vremenu. Računalnom simulacijom moguće je proučavati složene sustave, analizirati ih te upravljati njima.[1]

Koncept simulacije može se opisati kao kružni proces (slika 1). Na osnovi realnog sustava modelira se simulacijski model te se iz njega pomoću simulacijskog eksperimenta dolazi do rezultata simulacije. Interpretacijom rezultata dolazimo do podataka koji su vezani za realni sustav te oni, ako imaju pozitivno značenje za sustav, mogu biti implementirani. [1]



Slika 1 Koncept simulacije[1]

2.1. OSNOVE ZA IZRADU SIMULACIJSKOG MODELA

Kao i svaki postupak, izrada simulacijskih modela podliježe određenim pravilima. U svrhu ispravnosti rada simulacijskih modela te točnosti rezultata dobivenih simulacijskim eksperimentima, potrebno je pratiti određene korake i pravila prilikom izrade simulacijskog modela i provođenja simulacijskih eksperimenata.

2.1.1. KORACI POTREBNI ZA IZRADU SIMULACIJSKOG MODELA

Za početak potrebno je odrediti cilj prema kojem se želi krenuti jer se simulacijski model izrađuje prema postavljenim zahtjevima. Zatim je potrebno identificirati sustav u realnosti koji želimo ispitati te utvrditi je li simulacijsko modeliranje adekavatan izbor za rješavanje promatranog problema. Prije same izrade simulacijskog modela potrebno je prikupiti i analizirati podatke realnog sustava te identificirati one podatke koji će biti potrebni za izradu simulacijskog modela i pronalazak rješenja. S definiranim ciljem, odabranim sustavom i potrebnim podatcima pristupa se izradi simulacijskog modela. Simulacijsko modeliranje također treba provesti prema određenim pravilima (više u poglavlju 2.1.2.). Kako bi se potvrdila ispravnost rada simulacijskog modela i točnost dobivenih rezultata, potrebno je provesti verifikaciju i validaciju simulacijskog modela. Verifikacija se provodi na način da se utvrdi ispravno ponašanje rada simulacijskog modela te odvija li se sve kako je predviđeno. U najvećoj se mjeri to radi vizualno, ali postoje i drugi načini (brojčanici, hodogrami, monitori…). Validacija simulacijskog modela vezana je za rezultate koji se dobiju provođenjem simulacijskog eksperimenta. Potrebno je simulirati realan slučaj te usporediti stvarno dobivene rezultate s onima dobivenim simulacijskim eksperimentom. Da bi simulacijski model bio validiran, izrazito je važno da se dobiveni podatci podudaraju s onima dobivenim u realnom okruženju. Nakon što je simulacijski model verificiran i validiran, mogu se provoditi neovisni simulacijski eksperimenti te se njihovi rezultati mogu analizirati. Iz njih se može zaključivati i prenositi preporuke na realni sustav.

Gore opisano može se sažeti u devet koraka koje treba slijediti redom[1]:

1) Definicija cilja
2) Identifikacija sustava
3) Prikupljanje i analiza podataka o sustavu
4) Izrada simulacijskog modela
5) Verifikacija modela
6) Validacija modela
7) Planiranje i izvođenje simulacijskih eksperimenata
8) Analiza rezultata
9) Zaključci i preporuke

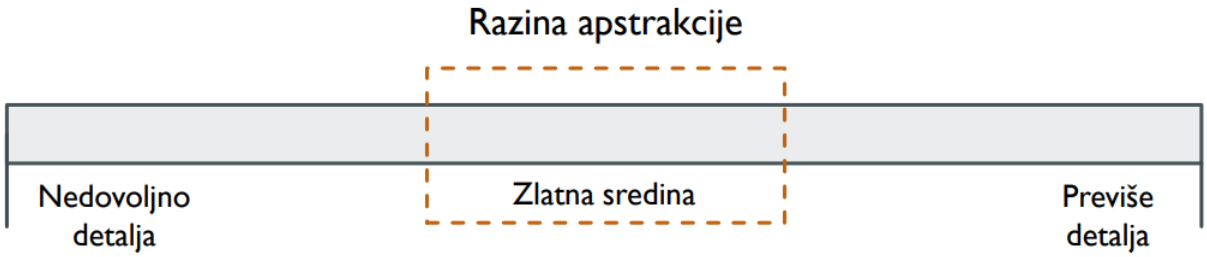
2.1.2. PRAVILA IZRADE SIMULACIJSKOG MODELA

Iskustveno je kroz razvoj simulacijskog modela znanstvena zajednica postavila neka opća pravila kojih se treba pridržavati prilikom modeliranja.[2] Sažeto se to može zapisati u obliku deset načela:

1) Simulacijski model treba biti najjednostavniji mogući da zadovolji postavljene zahtjeve
2) Problem se ne smije prilagođavati tehnici rješavanja
3) Zaključci se moraju donositi strogo
4) Prije primjene simulacijskog modela u praksi, potrebno ga je testirati
5) Kritički analizirati rezultate
6) Simulacijski model ne može dati rješenje problema za koji nije projektiran
7) Simulacijski model koristiti u granicama za koje je predviđen
8) Prilikom izrade simulacijskog modela može se doći do određenih rezultata i spoznaja
9) Rezultati dobiveni simulacijskim modelom ne mogu biti bolji od realnih rezultata
10) Simulacijski model nije zamjena za donošenje odluka o realnom sustavu

Iz gore navedenih pravila može se zaključiti kako simulacijski modeli nisu u potpunosti točni, odnosno u sebi sadrže razinu apstrakcije (slika 2).

„*Svi modeli su netočni, ali su neki korisni*.“ – *George Box*



Slika 2 Razina apstrakcije [1]

2.2. PREDNOSTI I NEDOSTATCI RAČUNALNIH SIMULACIJA

Kao glavne prednosti korištenja računalnih simulacija može se navesti relativno jednostavna primjena i oponašanje realnog sustava bez njegovog ometanja i prekidanja. Također, računalni model može služiti za više različitih simulacijskih eksperimenata te ga se lako može modificirati i manipulirati njime. Vrlo je koristan u vizualnoj interpretaciji sustava.

Osim prednosti, računalne simulacije imaju i određene nedostatke. Treba naglasiti da je proces modeliranja relativno dug te je potrebno poznavanje odgovarajućih metoda i alata za izradu računalnih simulacijskih modela. Također, ne postoji garancija za pronalazak optimalnog rješenja, a uz to cijena izrade samog simulacijskog modela može biti vrlo velika.

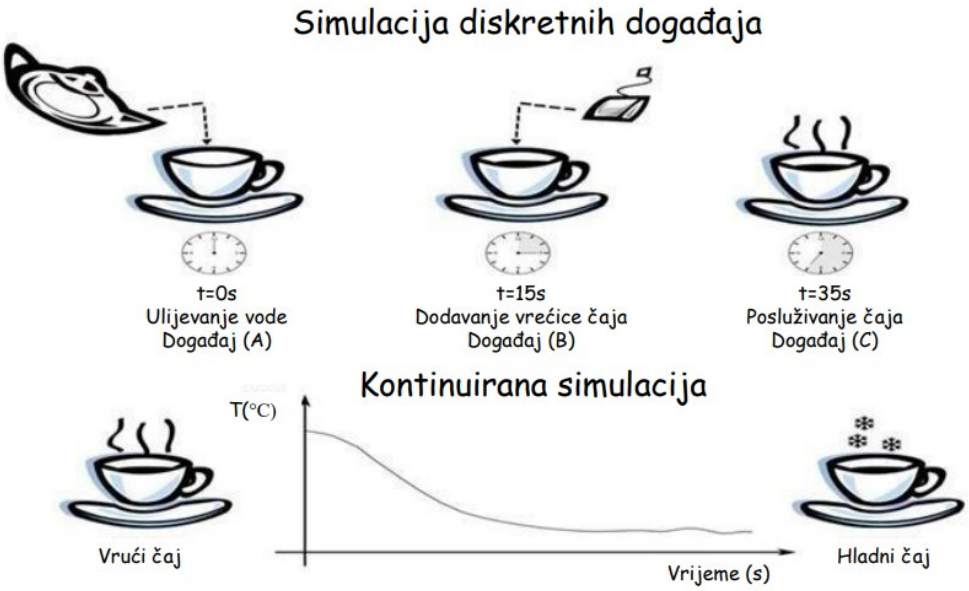
2.3. PODJELA SIMULACIJSKIH MODELA

Kada se općenito govori o modelima, oni se mogu podijeliti u nekoliko skupina, a to ovisi o metodi kojom se rješavaju sami modeli. Mogu se podijeliti:

- prema strukturi na fizičke i apstraktne modele,
- prema ponašanju na statičke i dinamičke modele,
- prema načinu rješavanja na konceptualne, matematičke i simulacijske modele.

Dakle, simulacijske metode zahtijevaju simulacijske modele koji se općenito dijele na dvije grane. Prema vrsti varijabli u modelu mogu se podijeliti na determinističke i stohastičke modele, a prema načinu promjene stanja modela u vremenu na kontinuirane (sistemska dinamika) i diskretne (simulacija diskretnih događaja).[1]

Na slici 3 prikazana je razlika između diskretnih i kontinuiranih događaja. Vidljivo je da kod diskretnih događaja postoji točan broj događaja i promjena stanja te je za njih poznato točno vrijeme. Kontinuirani događaji pak mijenjaju svoje stanje s infinitezimalnom promjenom vremena.



Slika 3 Razlika diskretnih događaja i kontinuiranih [4]

2.3.1. SIMULACIJA DISKRETNIH DOGAĐAJA

Ako znamo točne događaje i njihovo vrijeme pojavljivanja, tada govorimo o simulaciji diskretnih događaja. Za tu se vrstu događaja može reći da vrijeme „skače“ s događaja na događaj, a određeni se događaj izvrši u jednom trenutku.

Simulacija diskretnih događaja odnosi se na modeliranje dinamičkog sustava u kojem se varijable stanja sustava mijenjaju na diskretan način (mogu se opisati diskretnim matematičkim funkcijama) te se i vrijeme mijenja na isti način. Za prikaz vremena u diskretnoj simulaciji koristi se koncept niza događaja. Kada se dogodi događaj u sustavu, izvode se određene aktivnosti vezane uz taj događaj. Zatim se određuju nove vrijednosti varijabli stanja sustava i bilježi statistika sustava te se trenutno vrijeme pomiče do sljedećeg događaja. Diskretna simulacija temelji se na objektno orijentiranom modeliranju i modeliranju uz korištenje mreža čekanja i evaluacijskih mreža.[5]

Simulacijsko modeliranje sustava s diskretnim događajima, u kojima se varijabla stanja mijenja samo u diskretnom skupu točaka, u vremenskim simulacijskim modelima analizira se numeričkim, a ne analitičkim metodama. Analitičari koriste deduktivno matematičko zaključivanje kako bi „riješili“ model. Na primjer, diferencijalni izračun može se koristiti za izračunavanje politike

minimalnih troškova za neke modele zaliha. Brojčane metode koriste računalne postupke za „rješavanje“ matematičkih modela. U slučaju modela simulacije, koji koriste numeričke metode, modeli se „pokreću“, umjesto da se rješavaju, odnosno umjetna povijest sustava generira se iz pretpostavki modela, a opažanja se prikupljaju da bi se analizirala i procijenila stvarna izvedba sustava. Primjena simulacije u stvarnom svijetu prilično je velika, a količina je podataka koji se pohranjuju i kojima se manipulira ogromna, pa se takva testiranja obično provode pomoću računala.[6]

2.3.2. KONTINUIRANA SIMULACIJA

Ako se ipak radi o nekom procesu koji svoje stanje mijenja konstantno u ovisnosti o vremenu, tada je riječ o kontinuiranoj simulaciji. Jedan je od najboljih primjera promjena temperature, za taj proces ne možemo reći da se dogodio u jednom trenutku, već je trajao određeno vrijeme i u svakom trenutku poprimao novu vrijednost. Dakle, takav je proces kontinuiran.

Kontinuirani procesi, za razliku od diskretnih procesa, odvijaju se pod fiksnim uvjetima u kojima se promjene odvijaju istovremeno na neprekinut način. Takav proces može se podijeliti u faze, od kojih svaka ispunjava navedene uvjete. Svaka od faza provodi se u uvjetima koji se ne mijenjaju tijekom vremena, a promjene se kontinuirano usmjeravaju u sljedeće faze.

U simulaciji kontinuiranih procesa, u kojoj model dinamičkog sustava ima oblik diferencijalne jednadžbe ili sustava takvih jednadžbi, on se naziva sustavom jednadžbi stanja. Model sustava jednadžbi stanja kreiran je na temelju odabranih značajki sustava. Ovim značajkama dodijeljene su vrijednosti koje definiraju trenutno stanje sustava.[6]

3. PRIMJENA SIMULACIJSKIH MODELA U PROIZVODNJI DIJELOVA

Ovaj rad bavi se problemom kako predstaviti rješenje krajnjem korisniku pomoću simulacijskog modeliranja te mu omogućiti što lakšu upotrebu istoga. U daljnjem radu pokušat će se prikazati nekoliko rješenja. Ideja polazi od zamisli da korisnik unese nekoliko osnovnih parametara te da se simulacijski model prilagodi tome. Konkretno, pokušaj će ići u smjeru da korisnik unosi nekakav željeni cilj (primarno broj željenih proizvoda koje treba proizvesti) te će se korisniku povratno dati rješenje pomoću provedenih simulacijskih eksperimenata. Simulacijski modeli i eksperimenti bit će modelirani i provedeni unutar računalnog programa za simulaciju proizvodnje *Siemens Tecnomatix Plant Simulation 15.*

Jedan je od načina upotreba kanban metode. S obzirom na to da je kanban zasnovan na karticama pomoću kojih se može naručiti repromaterijal s prethodih radnih stanica kada zaliha na određenoj radnoj stanici postigne minimalnu vrijednost. Drugo moguće rješenje upotreba je logičkih metoda pomoću kojih će se regulirati protok materijala, redoslijed operacija, njihova vremena i radne stanice na kojima se obavljaju operacije. Upotreba logičkih metoda upotrijebit će se kao baza za univerzalni simulacijski model jer je znatno fleksibilnija u usporedbi s kanban modelom.

Izradi univerzalnog simulacijskog modela pristupit će se s idejom da korisnik unosi neke od ulaznih varijabli na osnovi kojih će se cijeli simulacijski eksperiment provoditi samostalno bez dodatnih modifikacija na samom simulacijskom modelu. Dakle, korisnik bi naručivao, odnosno unosio ulazne varijable kao što su količina određenih proizvoda, vrijeme trajanja pojedinih operacija i redoslijed radnih mjesta pojedine operacije te pomoću toga generirao ulazni materijal i tok materijala. Također, pokušaj će ići u smjeru da radne stanice same prepoznaju koji je proizvod i koja operacija u pitanju i prema tome određuju vrijeme operacije i sljedeći korak toka materijala.

Izrada samog simulacijskog modela bit će opisana detaljno, korak po korak. Počet će se od definiranja vrste proizvoda i njihovog kreiranja u računalnom programu. Zatim će biti kreirane tablice koje će služiti kao određena vrsta sučelja za unos ulaznih informacija u simulacijski model te će one biti jedino čemu bi krajnji korisnik trebao pristupati prilikom postavljanja uvjeta za

simulacijski eksperiment. Srž će simulacijskog modela biti kodiranje logičkih metoda prema čijoj će se definiciji sve odvijati. Ideja je zapisati dvije vrste metoda, jednu ulaznu i jednu izlaznu. Ulazna metoda izvršavala bi se prilikom ulaska određenog proizvoda na određenu stanicu te bi pomoću identificirane vrste proizvoda pokretala potragu odgovarajućih varijabli iz tablica ulaznih varijabli. Iščitane varijable, informacije, usvajale bi se kao određeni režimi radne stanice na kojoj se aktivira metoda. Na isti način trebala bi se ponašati izlazna metoda koja će se aktivirati prilikom izlaska određenog proizvoda s određene radne stanice. Ručno će se definirati varijabla svakom pojedinom proizvodu te će ona pomoću jedne od metoda biti manipulirana te će se na taj način brojati redni broj operacije na nekom od proizvoda. Prostorni raspored strojeva bit će zanemaren, postavit će se nekolicina strojeva koja će zadovoljiti svrhu. Također, sva vremena potrebna za postavljanje i namještanje stroja bit će zanemarena, kao i mogući nastanak greške bilo koje prirode. Ostale karakteristike i postavke strojeva bit će jasno pojašnjene i u koracima će se opisati kako ih definirati.

3.1.IDEJNI PROIZVODNI PROCES

Za početak će se osmisliti narudžba potencijalnog naručitelja te opisati idejni proizvodni proces za koji će se ponuditi rješenje. Narudžba potencijalnog naručitelja izrada je nekolicine proizvoda. Konkretno, radi se o proizvodnji tri vrste proizvoda (proizvod AB, proizvod C i proizvod D) u definiranim količinama kao u tablici 1 *Proizvodni program*.

	Proizvod		
	AB	C	D
Količina	30	50	70

Tablica 1 Proizvodni program

Potrebna količina proizvoda za proizvesti jest 30 jedinica proizvoda AB, 50 jedinica proizvoda C i 70 jedinica proizvoda D. Iz tehnoloških podataka poznato je da proizvod A u svojoj proizvodnji mora proći kroz proces tokarenja i dva procesa glodanja, proizvod B mora proći četiri procesa glodanja te se na kraju moraju spojiti u omjeru 1:1 na montažnoj stanici gdje postaju proizvod AB koji zatim odlazi na proces glodanja. Proizvod C mora proći proces tokarenja i dva procesa glodanja, a proizvod D tri procesa glodanja. Dalje su u radu navedeni proizvodi iz narudžbe

proizvoljno imenovani kao A, B, C, D i AB. Navedeni proizvodi proizvest će se na postojećih pet proizvodnih kapaciteta. Kupac zahtijeva informaciju o mogućem roku isporuke te je potrebno odrediti hodograme kojima se postiže što kraće vrijeme proizvodnje, a time i krajnji rok isporuke. Raspoloživi proizvodni kapaciteti popisani su u tablici 2 *Popis proizvodnih kapaciteta*. Dakle, za ovaj proizvodni proces koristit će se jedna tokarilica, tri glodalice i jedna montažna stanica.

Popis proizvodnih kapaciteta	**Oznaka**	**Količina**
Tokarilica	S1	1
Glodalica 1	S2	1
Glodalica 2	S3	1
Glodalica 3	S4	1
Montažna stanica	AS	1

Tablica 2 Popis proizvodnih kapaciteta

Ako bi se proizvodnja odvijala jednosmjerno, što podrazumijeva da nema povratnih hodova, problematika bi ovog proizvodnog procesa bila minimalna, odnosno svela bi se na usklađivanje vremena operacija. Međutim, u ovome slučaju zahtijeva se proizvodnja triju različitih proizvoda na ograničenom broju i vrsti proizvodnih kapaciteta. Kada bi se ovakva narudžba sastavljena od različitih proizvoda proizvodila jednosmjerno (linijski), sam proizvodni proces zahtijevao bi puno veći broj proizvodnih kapaciteta, a uz to bi bio vremenski i financijski vrlo neekonomičan. Dakle, sada će od presudne važnosti biti raspodjela toka proizvoda kako bi sam proizvodni proces bio vremenski učinkovitiji i samim time financijski isplativiji.

Tehnološki podatci o vremenima pojedinih operacija određeni su od strane tehnologa te su zadani u tablici 3 *Vremena operacija*. Potrebno je dodati kako se bilo koja operacija tokarenja odvija na tokarilici (S1), a bilo koja operacija glodanja može se odvijati na bilo kojoj od raspoloživih glodalica (S2, S3 ili S4), također operacija montaže odvija se na montažnoj stanici (AS).

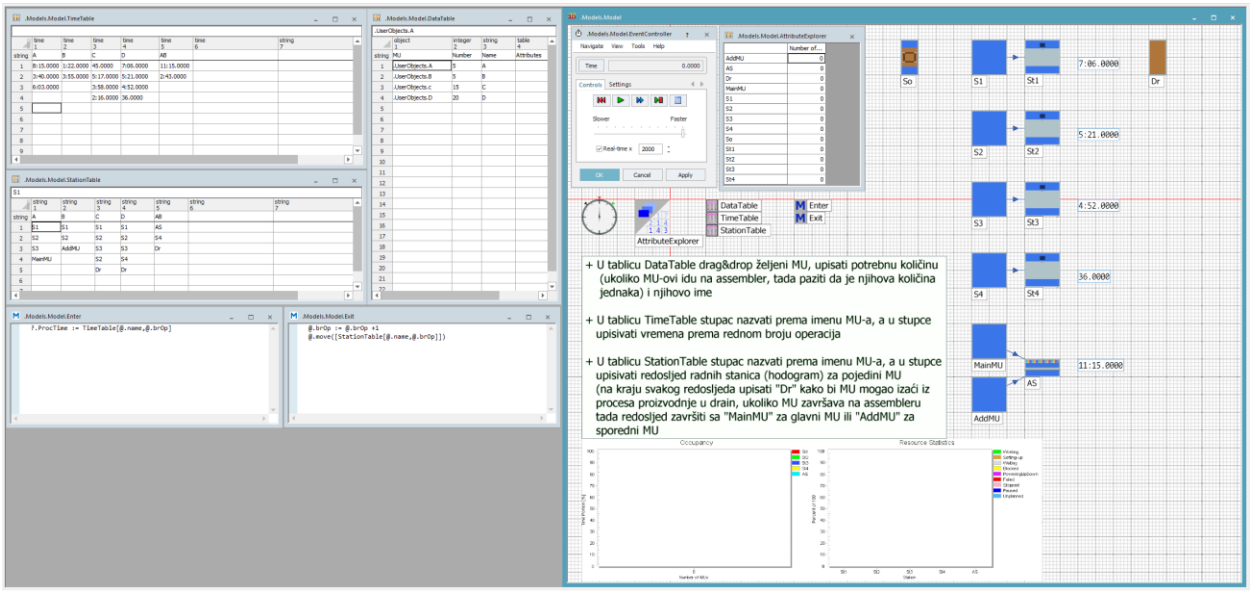
Vrijeme operacija [min]						
Rbr.	**Vrsta operacije**	**A**	**B**	**AB**	**C**	**D**
1	Tokarenje	5:20	-	-	6:10	-
2	Glodanje	3:45	1:11	-	5:35	7:18
3	Glodanje	8:12	3:17	-	6:30	3:57
4	Glodanje	-	2:45	-	-	7:46
5	Glodanje	-	7:15	-	-	-
6	Montaža	-	-	5:50	-	-
7	Glodanje	-	-	10:20	-	-

Tablica 3 Vremena operacija

Dakle, narudžba zahtijeva proizvodnju triju različitih proizvoda u različitom broju jedinica u što kraćem vremenskom roku. Stoga je, s obzirom na raspoloživost proizvodnih kapaciteta, od ključne važnosti uspostaviti ispravan tok materijala. Problem koji tek iziskuje rješenje jest kako rasporediti određene operacije prema određenim radnim mjestima, odnosno proizvodnim kapacitetima, kako bi se postigao traženi učinak. Rješenje će se pokušati dati osmišljavanjem simulacijskog modela koji će udovoljiti promjenama ulaznih parametara, odnosno koji će biti univerzalan. Zamisao je da takav simulacijski model krajnjem korisniku (onome koji planira proizvodnju) omogući da unosom tri vrste podataka (količine proizvoda, vremena pojedinih operacija i radna oprema za izvršavanje pojedine operacije) brzo i jednostavno dođe do određene analize zadanog proizvodnog procesa. Također, simulacijski će model u zahtjevu imati brzu i jednostavnu izmjenu ulaznih podataka u svrhu što lakšeg prilagođavanja univerzalnog modela zadanom proizvodnom procesu.

4. IZRADA SIMULACIJSKOG MODELA

Simulacijski model (slika 4) bazirat će se na korištenju naredbe *Method* unutar samog programa. Pomoću metode koja se zapisuje u obliku računalnog koda bit će kontroliran ulaz i izlaz pojedinog entiteta unutar simulacijskog modela. Od korisnika će se tražiti da ulazne podatke uvrsti u tri tablice. U prvu će se unositi željeni proizvodi i njihova količina, u drugu tablicu vremena operacija prema njihovom redoslijedu obavljanja, a u treću redoslijed strojeva (radnih stanica) na kojima se obavlja pojedina operacija (hodogram).



Slika 4 Simulacijski model izrađen unutar programa Tecnomatix Plant Simulation

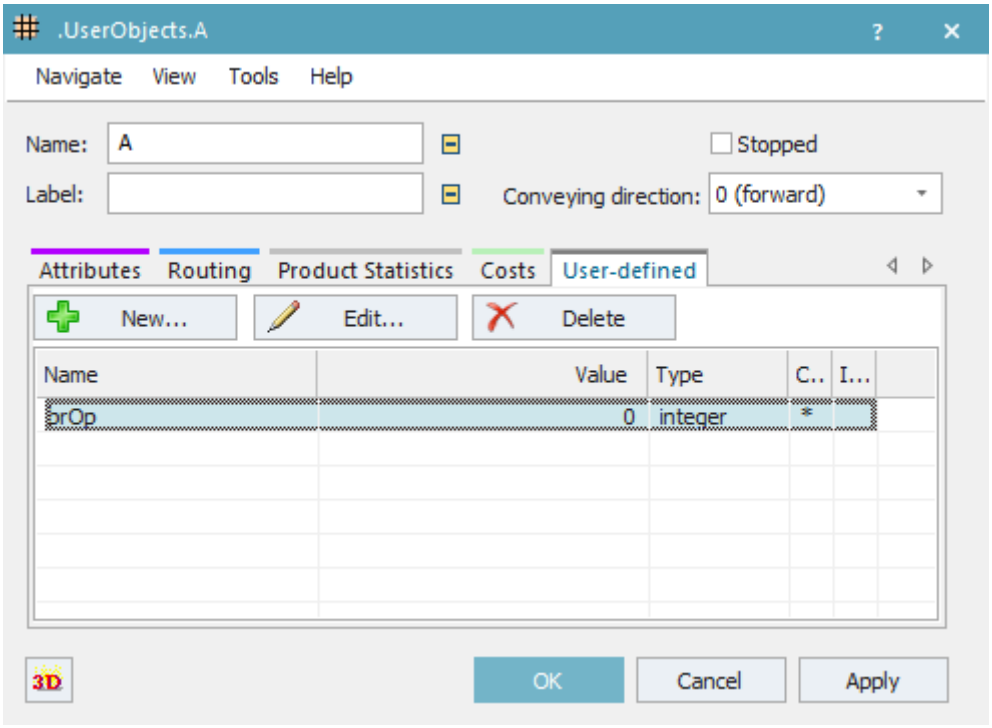
4.1. DETALJNI POSTUPAK IZRADE

U narednim potpoglavljima opisat će se detaljni postupak izrade univerzalnog simulacijskog modela s naglaskom na sve elemente potrebne za njegovu izradu te sve potrebne postavke.

4.1.1. DEFINIRANJE VRSTA PROIZVODA

Za početak, kreirat će se nekoliko vrsta proizvoda (*Mobile Units – MU*). To se radi na način da se u *Class Libraryju* u mapi *MUs* pritiskom na desni klik na *Part* odabere opcija *Duplicate*. Nakon

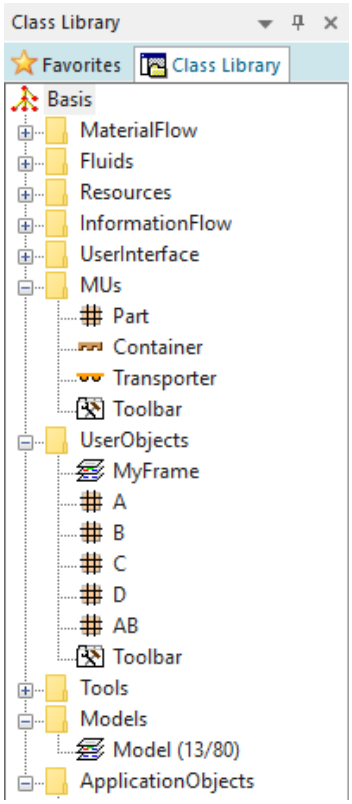
toga, kreirat će se duplicirani proizvod – *Part* unutar mape *UserObjects*. Dvostrukim klikom na njega ulazi se u postavke gdje se mijenja njegov naziv (u ovom slučaju nazvat će ga se „A“). Također, unutar postavki pod karticom *User-defined* dodat će se novi atribut „brOp“ koji će biti tipa integer te će nositi vrijednost 0 (slika 5). To se može napraviti klikom na *New…*. Razlog stvaranja novog atributa „brOp“ jest taj što će se isti koristiti kasnije prilikom kodiranja metoda za brojanje rednog broja operacije na način da će se na svakom izlazu s pojedinog stroja, odnosno radne stanice, dodavati vrijednost +1 (pogledati poglavlje 3.1.3.).



Slika 5 Postavke za Part A

Na isti način dodat će se proizvodi „B“, „C“, „D“ i „AB“, ali će se za razliku od ostalih proizvoda kod proizvoda „AB“ za početnu vrijednost varijable „brOp“ staviti vrijednost 1 jer se proizvod „AB“ kreira tek kasnije u procesu direktno na stanici (*AssemblyStation – AS*).

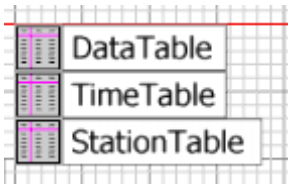
Dakle, za sve proizvode koji nastaju na ulazu, te tako ulaze u proces, atribut „brOp“ treba biti vrijednosti 0, a za sve one koji nastaju na određenoj stanici unutar procesa atribut „brOp“ treba biti vrijednosti 1. Svi definirani proizvodi pojavit će se unutar trake *ClassLibrary* kako je prikazano na slici 6.



Slika 6 Popis proizvoda u Class Library

4.1.2. KREIRANJE TABLICA ZA UNOS ULAZNIH PODATAKA

Nakon kreiranja proizvoda koje treba proizvesti, potrebno je napraviti tri tablice za unos podataka (slika 7). To se radi na način da se iz trake *Toolbox* pod karticom *Information Flow* na radni prostor povuče objekt pod nazivom *DataTable*. Potrebno je preimenovati tablice u skladu sa slikom 7, a to se radi na način da se pritisne desni klik na pojedinu tablicu i odabere opcija *Rename*. Tablice će se preimenovati u *DataTable*, *TimeTable* i *StationTable*.



Slika 7 Naslovi tablica postavljenih u radni prostor

U prvu tablicu pod nazivom *DataTable* unose se podatci o tome koji će se proizvodi pojavljivati na izvoru te u kolikom broju – proizvodni program. Tablicu je potrebno organizirati na način da se uključi *Column Index* te se prvi stupac nazove „MU“ tipa *object*, drugi stupac „Number“ tipa *integer* i treći stupac „Name“ tipa *string*. U prvi stupac pomoću *drag&drop* metode iz *Class Libraryja* potrebno je dovući ranije definirane proizvode. Pri tome valja napomenuti da je potrebno povlačiti jedan po jedan proizvod, ali njihov redoslijed nije bitan. Proizvode koji se nalaze u tablici (kao što je prikazano na slici 8) bit će moguće generirati na ulazu u onolikom broju koji se unese u drugi stupac pod nazivom *Number*. Također se u treći stupac treba upisati ime proizvoda. Treba pripaziti da u slučaju ako dva proizvoda završavaju na stanici za montažu njihova zahtijevana količina bude jednaka (ili u omjeru koji zahtjeva stanica za montažu), kako ne bi ostajali proizvodi unutar međuskladišta.

.Models.Model.DataTable

	object 1	integer 2	string 3	table 4
string	MU	Number	Name	Attributes
1	.UserObjects.A	5	A	
2	.UserObjects.B	5	B	
3	.UserObjects.c	15	C	
4	.UserObjects.D	20	D	
5				

Slika 8 DataTable

Unutar tablice pod nazivom *TimeTable* (slika 9) potrebno je također uključiti *Column Index* te stupce nazvati prema nazivu proizvoda koji se proizvode. Također, tip svih stupaca mora biti *time*. Ovaj su put uključeni svi proizvodi, oni koji se generiraju na izvoru i oni koji se kasnije generiraju na nekoj od stanica. U stupac koji je u skladu s imenom pojedinog proizvoda potrebno je upisati vremena operacija prema redoslijedu obavljanja.

.Models.Model.TimeTable

	time 1	time 2	time 3	time 4	time 5
string	A	B	C	D	AB
1	8:15.0000	1:22.0000	45.0000	7:06.0000	11:15.0000
2	3:40.0000	3:55.0000	5:17.0000	5:21.0000	2:43.0000
3	6:03.0000		3:58.0000	4:52.0000	
4			2:16.0000	36.0000	
5					

Slika 9 TimeTable

Tablica pod nazivom *StationTable* služi kao svojevrsni hodogram. U nju se upisuju nazivi radnih stanica – strojeva prema redu obavljanja operacija (slika 10). Potrebno je uključiti *Column Index* te stupce nazvati prema proizvodu za koji će se unositi redoslijed radnih stanica u taj stupac. Treba obratiti pozornost na to da će se kasnije u simulacijskom modelu radne stanice morati preimenovati u točne nazive kako su zadane u ovoj tablici (ili, ako je već napravljen simulacijski model, u tablicu unijeti točne nazive radnih stanica). Ako je definiran konačni hod proizvoda, na kraj stupca potrebno je dodati naziv izlaza materijala iz sustava, u ovom slučaju „Dr“. Međutim, ako je proizvod potrebno poslati na stanicu za montažu (*AssemblyStation – AS*), na kraj stupca potrebno je upisati „MainMU“ (naziv *Buffera* koji isporučuje glavni proizvod u stanicu za montažu) ili „AddMU“ (naziv *Buffera* koji isporučuje sporedni proizvod u stanicu za montažu), ovisno radi li se o glavnom ili sporednom proizvodu.

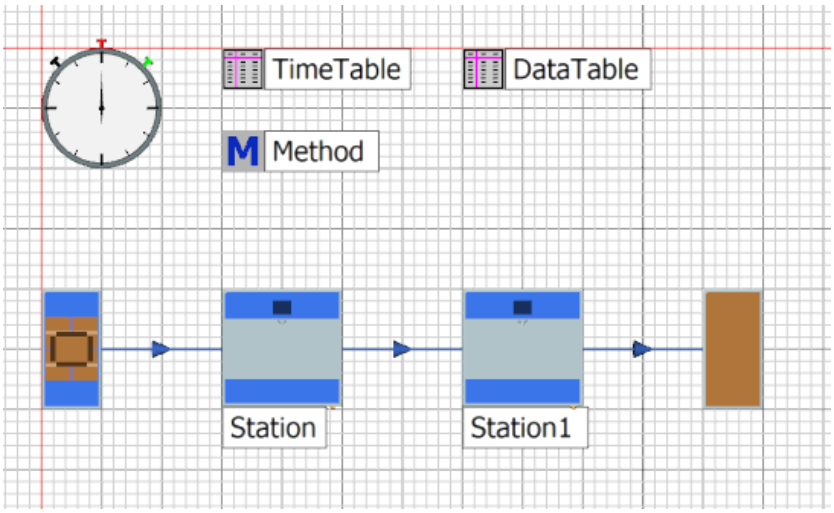
.Models.Model.StationTable

	string 1	string 2	string 3	string 4	string 5	st 6
string	A	B	C	D	AB	
1	S1	S1	S1	S1	AS	
2	S2	S2	S2	S2	S4	
3	S3	AddMU	S3	S3	Dr	
4	MainMU		S2	S4		
5			Dr	Dr		
6						

Slika 10 StationTable

4.1.3. KODIRANJE LOGIČKIH METODA - RAZVOJ

Ideja je izrada jednostavnog simulacijskog modela baziranog na ručno uređenim tablicama i zapisivanju logičkih metoda pomoću kojih će se iščitavati podatci iz tablica. Razvoj je počeo jednostavnim simulacijskim modelom (slika 11) s pokušajem promjene vremena operacije u ovisnosti o vrsti proizvoda koji ulazi u radnu stanicu. Ideja je bila da se točno na određeno mjesto u tablici upiše vrijeme koje je sortirano prema stroju i proizvodu.



Slika 11 Testni simulacijski model za razvoj i pisanje metoda

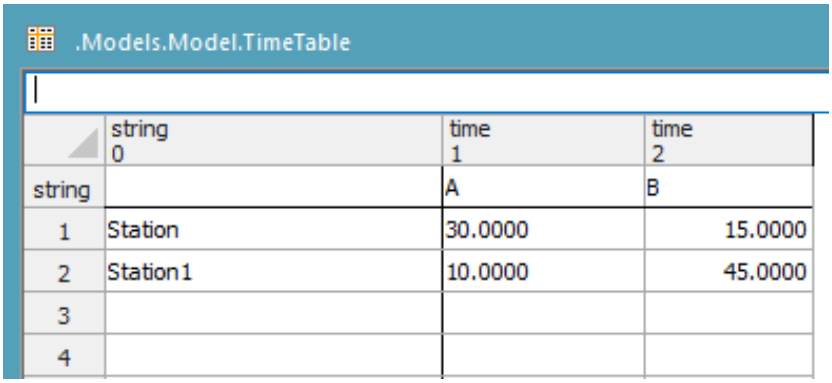
Proizvodi koji će se proizvoditi jesu „A“ i „B“ te će se zbog jednostavnosti proizvoditi po jedan komad od svakog proizvoda. To će biti zapisano u tablici *DataTable*, kao na slici 12, koja će se postaviti na izvor koji će prema tome kreirati proizvode (*MU selection: Sequence*).

.Models.Model.DataTable

	object 1	integer 2	string 3	table 4
string	MU	Number	Name	Attributes
1	.UserObjects.A	1	A	
2	.UserObjects.B	1	B	
3				

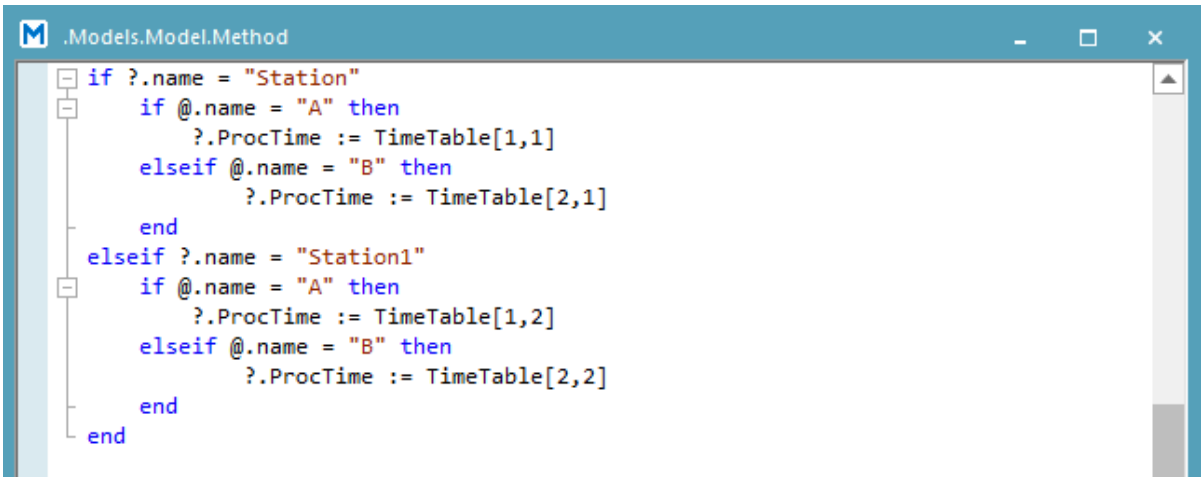
Slika 12 DataTable - osnovni simulacijski model

Kreirana je osnovna tablica vremena *TimeTable* koja će služiti kao izvor za iščitavanje pomoću logičkih metoda koje je potrebno kodirati (slika 13). Prva metoda zapisana je na vrlo bazičan način. Funkcija je fiksna i ne može se prilagođavati određenim promjenama lokacija (koordinata; stupac, redak) unutar tablice *TimeTable.* Riječ je o višestrukoj *if* petlji koja u slučaju ispravnosti čita točan položaj unutar tablice (slika 14).



	string 0	time 1	time 2
string		A	B
1	Station	30.0000	15.0000
2	Station1	10.0000	45.0000
3			
4			

Slika 13 TimeTable - osnovni simulacijski model



Slika 14 Prvi zapis metode za čitanje tablice TimeTable

Zapisani kod metode glasi:

```
if ?.name = "Station"
    if @.name = "A" then
        ?.ProcTime := TimeTable[1,1]
    elseif @.name = "B" then
            ?.ProcTime := TimeTable[2,1]
```

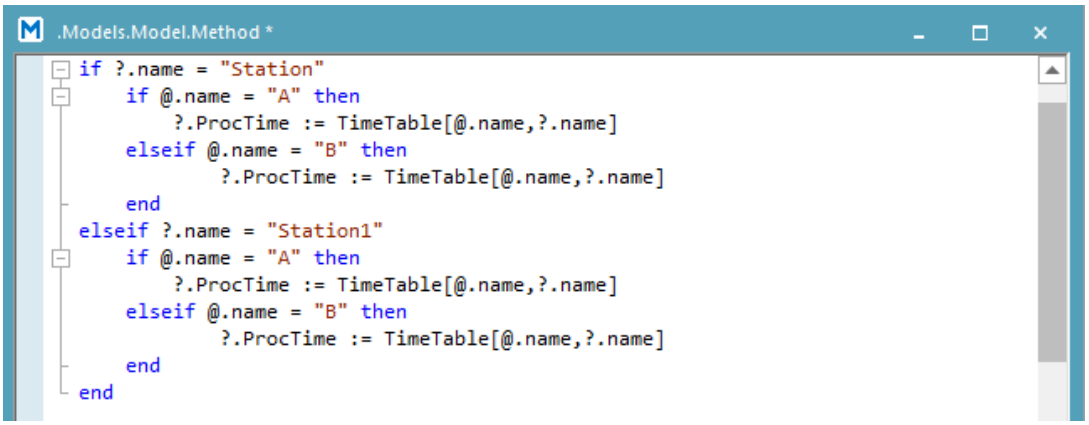
```
    end
elseif ?.name = "Station1"
    if @.name = "A" then
        ?.ProcTime := TimeTable[1,2]
    elseif @.name = "B" then
            ?.ProcTime := TimeTable[2,2]
    end
end
```

te se čita:

„Ako se radi o objektu pod imenom „Station“ tada ako je ulazni proizvod pod imenom „A“ usvoji vrijeme operacije iz tablice *TimeTable* pod koordinatama [1,1], a ako je ulazni proizvod pod imenom „B“ usvoji vrijeme operacije iz tablice *TimeTable* pod koordinatama [2,1].

Ako se radi o objektu pod imenom „Station1“ tada ako je ulazni proizvod pod imenom „A“ usvoji vrijeme operacije iz tablice *TimeTable* pod koordinatama [1,2], a ako je ulazni proizvod pod imenom „B“ usvoji vrijeme operacije iz tablice *TimeTable* pod koordinatama [2,2].“

U ovom slučaju vidljivo je da se problem može pojaviti ako se u tablicu *TimeTable* unesu dodatna vremena. Također, može se opaziti da se koordinate mijenjaju nekakvim pravilom, odnosno mijenjaju se u ovisnosti o tome koji je objekt u pitanju i koji se proizvod pojavljuje. Taj zakon promjene nalaže kodiranje poboljšane metode koja će se mijenjati prema njemu. Poboljšana metoda (slika 15) obuhvaća čitanje podataka iz tablice *TimeTable* u ovisnosti o promjeni imena radne stanice i imena proizvoda.



Slika 15 Poboljšana verzija metode za čitanje tablice TimeTable

Zapis koda glasi:

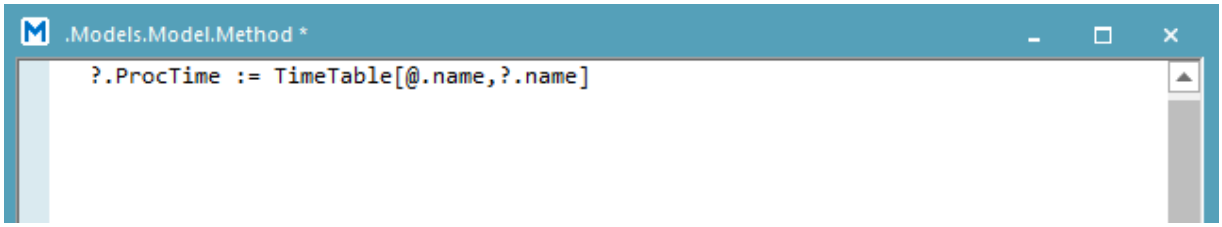
```
if ?.name = "Station"
    if @.name = "A" then
        ?.ProcTime := TimeTable[@.name,?.name]
    elseif @.name = "B" then
            ?.ProcTime := TimeTable[@.name,?.name]
    end
elseif ?.name = "Station1"
    if @.name = "A" then
        ?.ProcTime := TimeTable[@.name,?.name]
    elseif @.name = "B" then
            ?.ProcTime := TimeTable[@.name,?.name]
    end
end
```

te se kod može pročitati na sljedeći način:

„Ako se radi o objektu pod imenom „Station“ tada ako je ulazni proizvod pod imenom „A“ usvoji vrijeme operacije iz tablice *TimeTable* pod koordinatama [ime proizvoda, ime objekta], a ako je ulazni proizvod pod imenom „B“ usvoji vrijeme operacije iz tablice *TimeTable* pod koordinatama [ime proizvoda, ime objekta].

Ako se radi o objektu pod imenom „Station1“ tada ako je ulazni proizvod pod imenom „A“ usvoji vrijeme operacije iz tablice *TimeTable* pod koordinatama [ime proizvoda, ime objekta], a ako je ulazni proizvod pod imenom „B“ usvoji vrijeme operacije iz tablice *TimeTable* pod koordinatama [ime proizvoda, ime objekta].“

Ako se metoda aktivira zasebno na svakoj radnoj stanici, cijeli kod može se zapisati puno kraće (slika 16).



Slika 16 Konačni oblik metode za čitanje tablice TimeTable

Konačni je kod za čitanje *TimeTable* tablice sljedeći:

```
?.ProcTime := TimeTable[@.name,?.name]
```

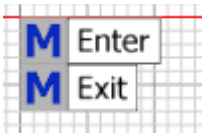
Čita se na sljedeći način te vrijedi za svako „mjesto“ na kojem se aktivira:

„Učitaj vrijeme operacije iz tablice *TimeTable* u ovisnosti o imenu ulaznog proizvoda i objekta na kojemu se aktivira.“

Ovo je konačni kod prema kojemu su na isti način kasnije iščitavani podatci iz tablica *TimeTable* i *StationTable* u konačnom univerzalnom modelu.

4.1.4. KODIRANJE LOGIČKIH METODA - UNIVERZALNA VERZIJA

Sada je potrebno postaviti dva objekta *Method* u univerzalni simulacijski model u koje će se unositi logički kod za ulaznu i izlaznu kontrolu na radnim stanicama. *Drag&drop* metodom ubacuje se *Method* iz *Information Flow* trake te je potrebno preimenovati objekte u „Enter“ i „Exit“, kako je prikazano na slici 17.



Slika 17 Naslovi metoda postavljenih u radni prostor

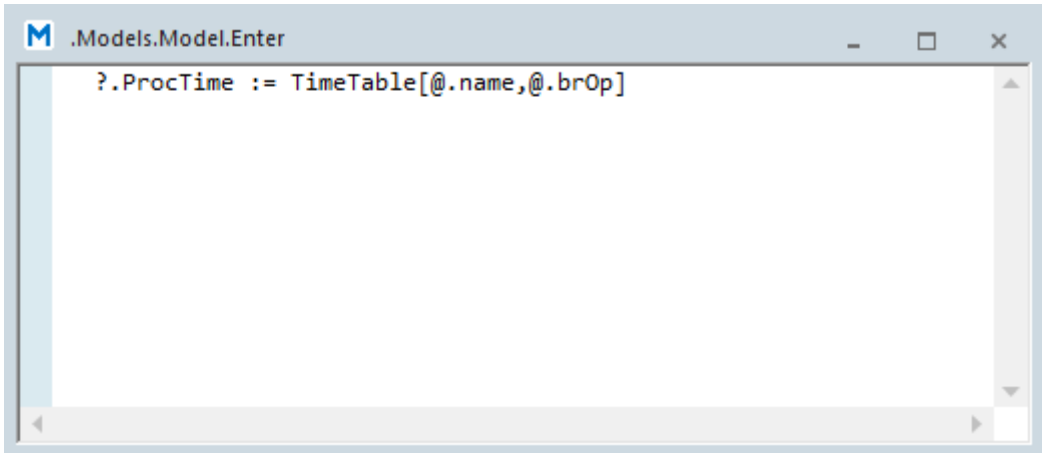
Prilikom pisanja univerzalnih metoda, potrebno je navesti nekoliko važnih informacija koje uvelike olakšavaju razumijevanje i pisanje metoda. Sve potrebne i raspoložive atribute te njihov tip za svaki od objekata moguće je pronaći na popisu pritiskom desnog klika na objekt i odabiranjem *Show Attributes and Methods.* Na tom popisu nalazit će se i atributi koji su ručno definirani (u promatranom slučaju atribut „brOp“).

Još jedna važna stvar za razlikovati način je pozivanja varijabli i/ili atributa. Ako se poziva atribut vezan za radnu stanicu (objekt na kojem se izvršava zadano), tada se koristi sintaksa „?.nazivAtributa“. U slučaju kada se poziva atribut vezan za proizvod (mobilnu jedinicu koja ulazi i izlazi s radne stanice), tada se koristi sintaksa „@.nazivAtributa“.

Ako se pridružuju određene vrijednosti, potrebno je koristiti sintaksu „ := “, a ako se radi o provjeri jednakosti, tada je dovoljno staviti samo znak jednakosti „ = “ unutar sintakse.

Način iščitavanja podataka iz tablice izvršava se prema logici „nazivTablice[stupac, redak]“. Treba dodati da se za stupac i/ili redak može koristiti numerički zadana koordinata iz tablice ili naziv stupaca i/ili redaka.

Nakon razumijevanja gore navedenog, mogu se zapisati kodovi za ulaznu i izlaznu metodu (*Enter* i *Exit*). Prvo se dvostrukim klikom otvori metoda pod nazivom *Enter* te se zapiše sintaksa prema prikazanoj slici 18:



Slika 18 Kod ulazne metode Enter

Zapisani je kod za *Enter* metodu sljedeći:

```
?.ProcTime := TimeTable[@.name,@.brOp]
```

te se može pročitati kao:

„Usvojiti vrijeme operacije (ili određene aktivnosti) na pridruženoj stanici iz tablice *TimeTable* iz stupca koji odgovara nazivu ulaznog proizvoda i reda koji odgovara atributu *brOp* koja je pridružena ulaznom proizvodu.“

Dakle, ovom će se metodom kontrolirati ulaz proizvoda na pojedinu stanicu te će se u ovisnosti o tome koji je proizvod i koja je njegova operacija u pitanju mijenjati vrijeme operacije na toj radnoj stanici.

Primjer:

Na neku radnu stanicu, koja za ulaznu metodu ima postavljenu metodu *Enter*, ulazi proizvod pod nazivom „C“. Na toj će se stanici obavljati treća operacija na proizvodu „C“ (vrijednost „brOp“ varijable je 3). Dakle, metoda će iščitati podatak iz tablice *TimeTable* na sljedeći način, prikazan na slici 19, te ga usvojiti kao vrijeme operacije za navedeni slučaj:

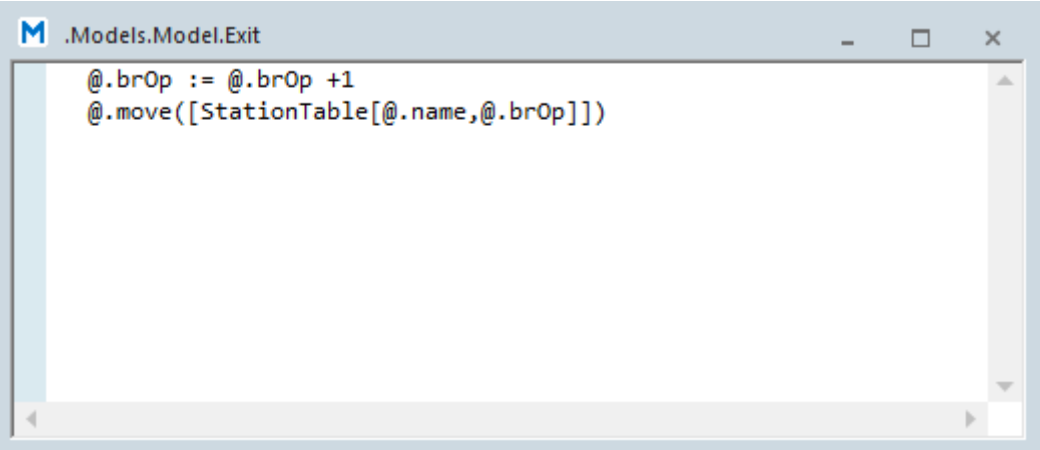
.Models.Model.TimeTable

3:58.0000

	time 1	time 2	time 3	time 4	time 5	time 6
string	A	B	C	D	AB	
1	8:15.0000	1:22.0000	45.0000	7:06.0000	11:15.0000	
2	3:40.0000	3:55.0000	5:17.0000	5:21.0000	2:43.0000	
3	6:03.0000		3:58.0000	4:52.0000		
4			2:16.0000	36.0000		
5						
6						
7						
8						
9						

Slika 19 Primjer čitanja tablice TimeTable (Proizvod „C“ operacija 3)

Istim postupkom potrebno je stvoriti metodu *Exit* koja će izgledati kao na slici 20:



Slika 20 Kod izlazne metode Exit

Zapisani je kod za *Exit* metodu sljedeći:

```
@.brOp := @.brOp +1
@.move([StationTable[@.name,@.brOp]])
```

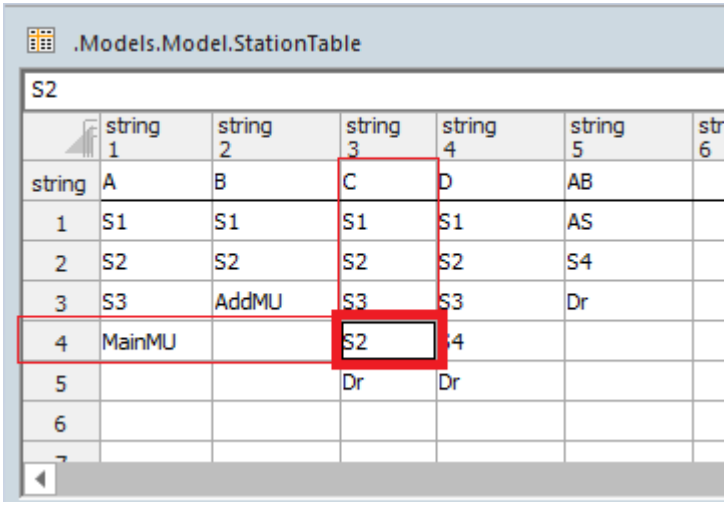
te se čita kao:

„Proizvodu pridružen atribut *brOp* povećati za jedan (+1) i pomaknuti proizvod na objekt zapisan u tablici *StationTable* u stupcu koji se zove kao izlazni proizvod i redu koji je jednak atributu *brOp* koji je pridružen izlaznom proizvodu.“

Ova će se metoda aktivirati na izlazu proizvoda s određene radne stanice. Njom će se povećati vrijednost atributa „brOp“ za jedan (+1) te će se odrediti na koju se sljedeću radnu stanicu pomiče proizvod.

Primjer:

S neke stanice, koja za izlaznu metodu ima metodu *Exit*, izlazi proizvod pod nazivom „C“. Na tom je proizvodu upravo izvršena operacija broj 3 (vrijednost „brOp“ varijable je 3). Metoda će povećati vrijednost s 3 na 4 te iščitati podatak iz tablice *StationTable.* Prema tome, proizvod „C“ bit će poslan na radnu stanicu „S2“ (slika 21).

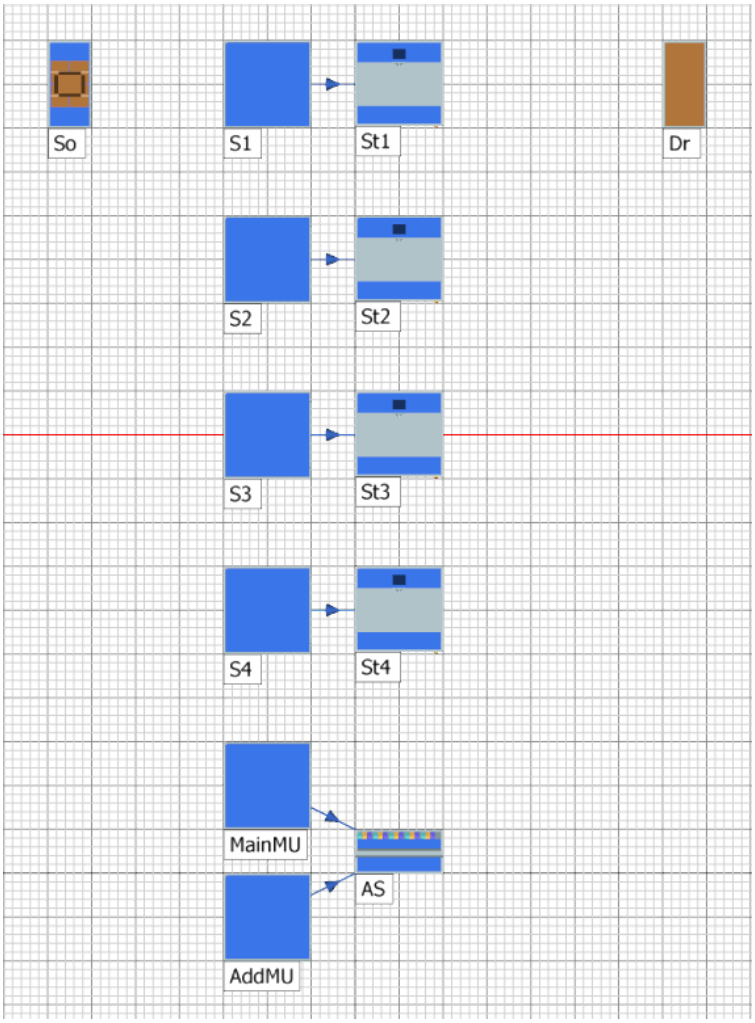


Slika 21 Primjer čitanja tablice StationTable (Proizvod „C“ operacija 4)

4.1.5. SLAGANJE PROSTORNIH ELEMENATA

Kod slaganja radnih stanica pozornost treba obratiti na povratni hod proizvoda koji može dovesti do poremećaja toka materijala, odnosno do zastoja, jer se može dogoditi da dva proizvoda u isto vrijeme trebaju ući na istu radnu stanicu. To je izraženo kod uravnoteženijih vremena trajanja operacija. Kako bi se izbjeglo nastajanje ovakve situacije, najjednostavnije je postaviti mala međuskladišta ispred svake radne stanice te ih ostaviti na standardnim postavkama. Tada se skladište i radna stanica mogu gledati kao jedan element te se *Connector* može postaviti između njih. Kada je u pitanju montažna stanica, kod nje je u potpunosti nemoguće postaviti ulaz materijala direktno pomoću metode. Kod nje se moraju postaviti dva međuskladišta prije te ih povezati direktno pomoću *Connectora*. U njih treba razvrstati glavni proizvod i sporedni proizvod, kako je i zadano u tablici *StationTable* (pogledati sliku 10).

Za promatrani simulacijski model postavit će se stanice prema gore navedenim pravilima, a u skladu s prikazanom slikom 22. Ulaz (*Source*) će se nazvati „So“, izlaz (*Drain*) „Dr“, radne stanice „St1“, St2“, „St3“, „St4“ i „AS“, a pripadajuća međuskladišta, kojima će biti dodijeljeni nazivi koji se zapravo nalaze u tablici *StationTable* i označavaju cjelinu međuskladište-radna stanica, „S1“, „S2“, „S3“, „S4“, „MainMU“ i „AddMU“.

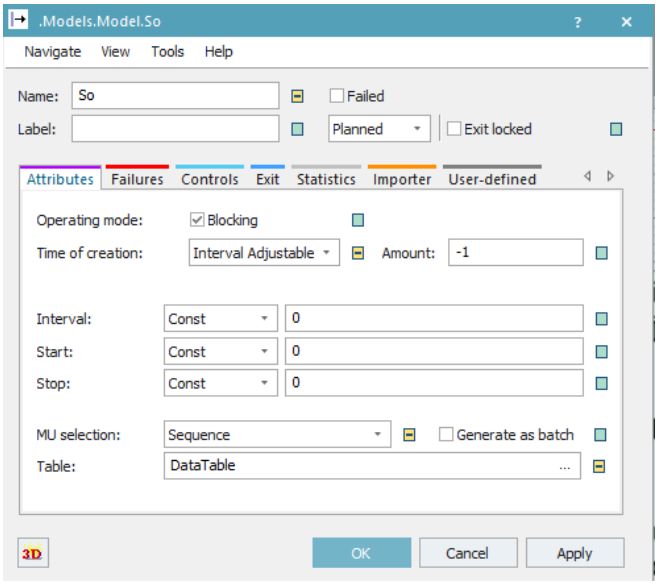


Slika 22 Raspored prostornih elemenata

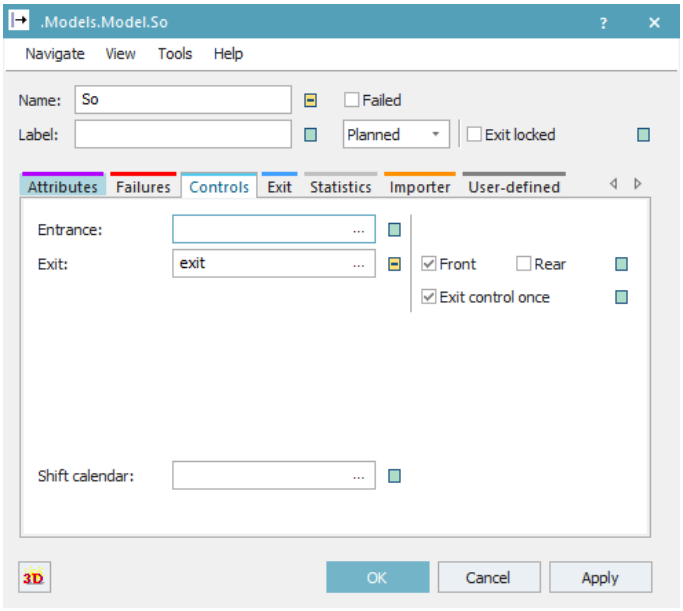
4.1.6. POSTAVKE PROSTORNIH ELEMENATA

Još je potrebno postaviti postavke pojedinih prostornih elemenata, uvrstiti ulaznu i izlaznu metodu na odgovarajuće mjesto te odrediti konačni tok materijala. Kada se definiraju postavke za ulaz proizvoda „So“, važno je pod karticom *Attributes* za *MU selection* postaviti *Sequence* te ispod u ćeliju *Table drag&drop* metodom postaviti *DataTable* (slika 23). Na taj će način izvor generirati točno one proizvode iz *DataTable* tablice prema zadanom broju za količinu.

Također, potrebno je pod karticom *Controls* za izlaznu strategiju (*Exit)* postaviti ranije napisanu metodu pod nazivom *Exit* (kako prikazuje slika 24), tako će proizvodi generirani na ulazu „So“ povećati svoju varijablu „brOp“ s 0 na 1 te će biti spremni za prvu operaciju, odnosno bit će pomaknuti na radnu stanicu za operaciju broj 1 iz tablice *StationTable*.

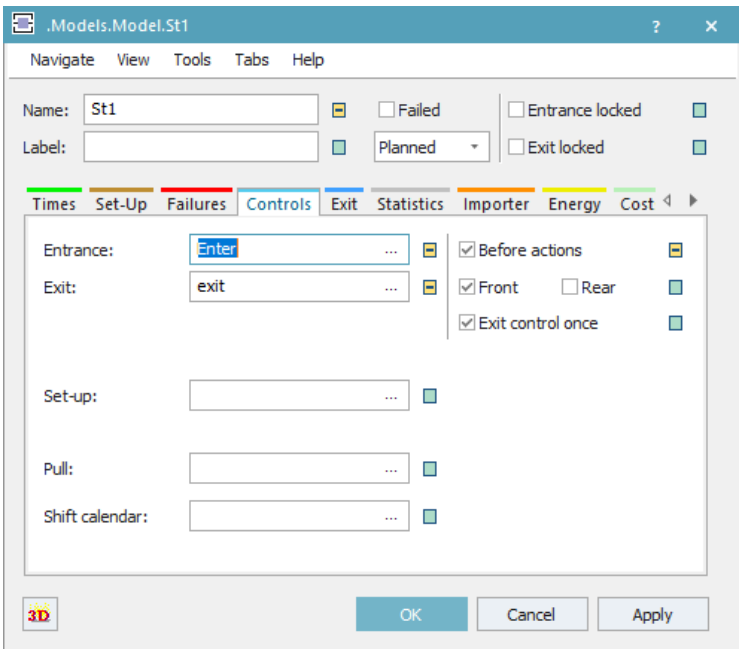


Slika 23 Postavke izvora „So“ - Attributes



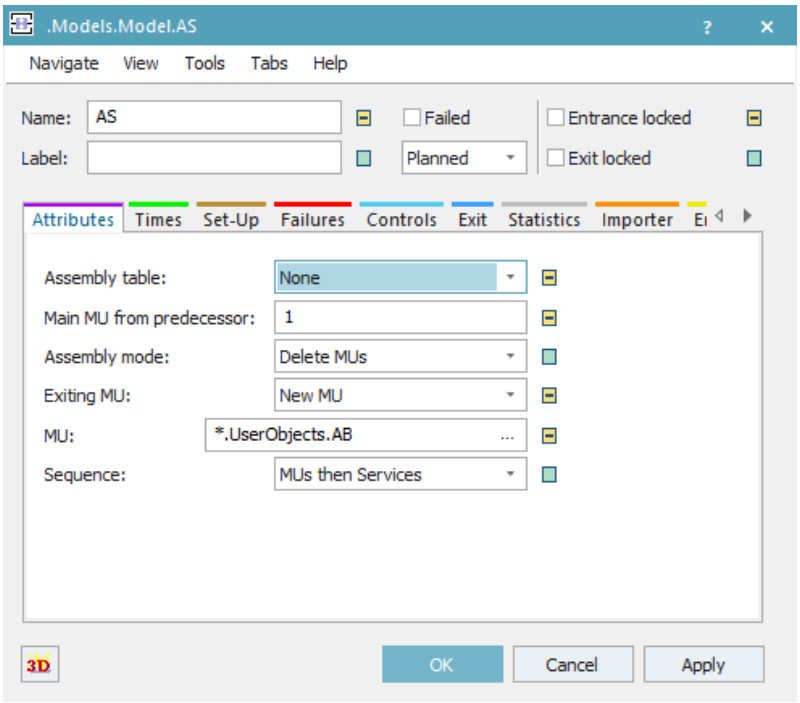
Slika 24 Postavke ulaza „So“ -Controls

Radne stanice (*Station*) postavljaju se na sljedeći način (slika 25). Unutar kartice *Controls* za ulaznu strategiju (*Entrance)* postavlja se metoda *Enter*, a za izlaznu strategiju (*Exit)* metoda *Exit*. To vrijedi za sve radne stanice unutar simulacijskog modela (*St1, St2, St3 i St4*). Na taj će način, pomoću ulazne metode, radna stanica poprimiti vrijednost zadanog vremena operacije ovisno o proizvodu koji je ušao u nju te će taj podatak o vremenu povlačiti iz tablice *TimeTable*. Pomoću izlazne metode na izlazu iz radne stanice proizvodu se povećava broj operacije („brOp“) te se šalje na sljedeću radnu stanicu prema tablici *StationTable*.



Slika 25 Postavke stanice „St1“ -Controls

Za montažnu stanicu „AS“ pod karticom *Controls* potrebno je postaviti identične postavke kao i za prethodno gore navedene radne stanice te se time ostvaruju isti uvjeti kao i kod „običnih“ radnih stanica. Dodatno, kod montažne stanice potrebno je postaviti karticu *Attributes* prema dolje prikazanoj slici 26.



Slika 26 Postavke montažne stanice „AS“ - Attributes

Ovim postavkama definirano je s kojeg prethodnika ulazi glavni proizvod. To se odnosi na *Connector* kojim je „AS“ povezan na međuskladišta, odnosno konkretno na međuskladište „MainMU“. Budući da se ovdje sklapa potpuno novi proizvod pod nazivom „AB“, za *Assembly mode* postavlja se *Delete Mus* čime se svi ulazni proizvodi brišu. Za *Exiting MU* odabire se *New MU* te se u ćeliju *MU* postavi part „AB“ iz *Class Libraryja*.

Svi ostali objekti unutar simulacijskog modela ostaju na početnim zadanim postavkama koje za promatrani problem nisu važne. Ovime su svi objekti i sva kretanja unutar simulacijskog modela definirani te se može pokrenuti simulacijski eksperiment nakon što se ispune ulazne tablice: *DataTable*, *TimeTable* i *StationTable*.

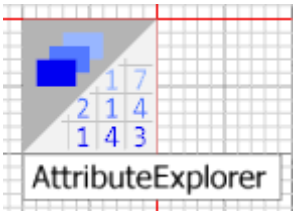
4.2. VERIFIKACIJA I VALIDACIJA SIMULACIJSKOG MODELA

Nakon izrade simulacijskog modela i postavljanja svih traženih postavki provest će se verifikacija i validacija, kako je već navedeno u poglavlju 2.1.1, s ciljem potvrde ispravnosti rada i točnosti dobivenih rezultata izrađenog simulacijskog modela. Prema definiranim pravilima za izradu simulacijskih modela, svakom simulacijskom modelu treba na određeni način provjeriti ispravnost njegovog rada i točnost dobivenih rezultata. Verifikacijom će se potvrditi ispravnost rada simulacijskog modela, odnosno potvrditi da simulacijski model ispravno funkcionira i da se sve radnje odvijaju kako je predviđeno. Validacijom će se utvrditi točnost rezultata na način da se rezultati izračunati za promatrani idejni proces podudaraju s rezultatima unutar računalne simulacije.

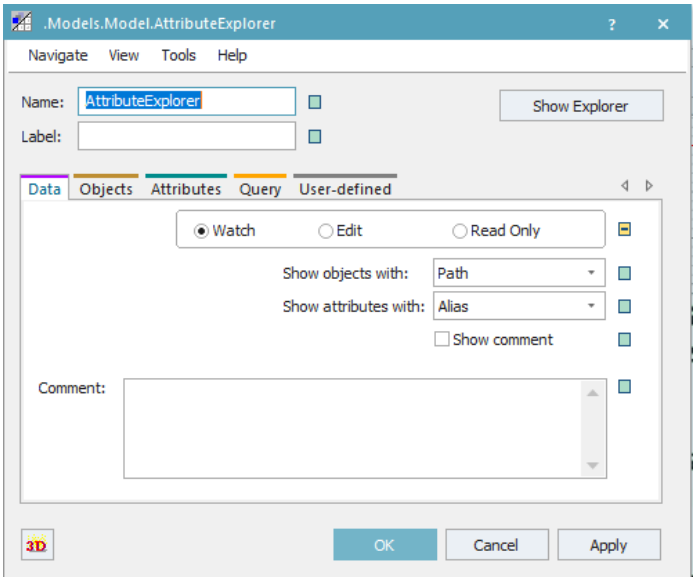
4.2.1. POSTUPAK VERIFICIRANJA SIMULACIJSKOG MODELA

Postupkom verifikacije potvrđuje se ispravnost rada simulacijskog modela, a to će se provesti na način da će se proizvesti po jedan komad od svake vrste proizvoda te će se pratiti količina proizvoda koji su izašli iz cjelokupnog procesa. Također, prikazat će se hodogrami za svaki od proizvoda te pomoću njih točno vidjeti odgovara li tok proizvoda onom predviđenom. Verifikacija će se dodatno kontrolirati direktno pomoću vizualnog sučelja.

Za početak postavit će se *Attribute Explorer* iz trake *Information Flow* te u postavkama namjestiti da broji proizvode na svakom objektu unutar simulacijskog modela. Pritiskom na desnu tipku miša, prelazi se preko svih objekata unutar simulacijskog modela dok ne pozelene, a tada se, i dalje držeći pritisnutom desnu tipku miša, povlače zeleno označeni objekti na ikonu *AttributeExplorer* (slika 27). Tim su postupkom odabrani svi objekti koji će se prikazivati unutar *Attribute Explorera* te će se na njima izvršavati brojanje proizvoda. Sada je potrebno dvostrukim klikom na ikonu otvoriti postavke. U postavkama pod karticom *Data* potrebno je označiti kućicu pored *Watch*, kao što je to prikazano na slici 28.



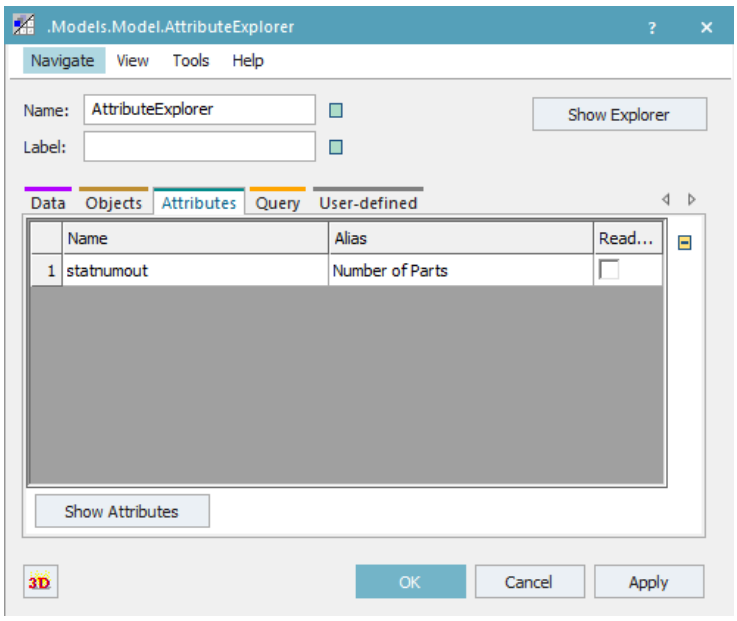
Slika 27 Attribute Explorer



Slika 28 Postavke AttributeExplorer -Data

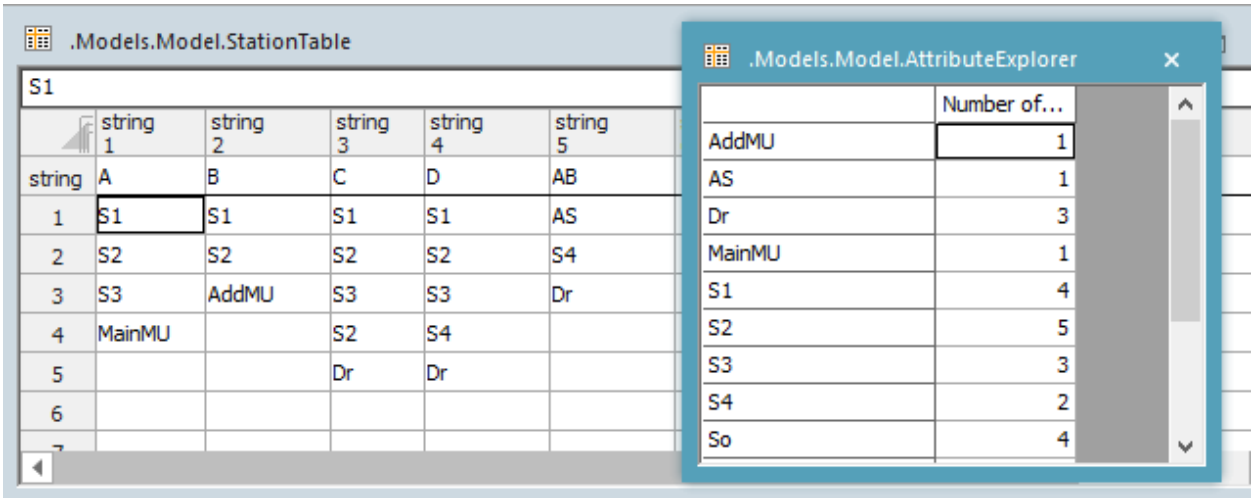
Pod karticom *Objects* mogu se vidjeti svi objekti koji su ubačeni u *Attribute Explorer.* Unutar kartice *Attributes* potrebno je omogućiti uređivanje klikom na kvadratić desno od same tablice (slika 29). Sada je moguće ručno upisati atribute koji se žele promatrati – mjeriti. U stupac *Name* upisuje se „statnumout“, naziv atributa koji broji količinu proizvoda koji su izašli s objekta, a u

stupac *Alias* postavlja se naslov koji će se prikazivati uz rezultate (u ovom slučaju to je „Number of Parts“). Potvrđuju se postavke klikom na *Apply* te se otvara prozor za prikaz klikom na *Show Explorer*.



Slika 29 Postavke AttributeExplorer -Attributes

Sada je potrebno pokrenuti simulacijski eksperiment na način da u tablicu *DataTable* za svaki od proizvoda unesemo količinu 1. Nakon izvršenog simulacijskog eksperimenta, pratit će se podudaranje podataka prikazanih u *AttributeExploreru* i onih unesenih u tablicu *StationTable.*



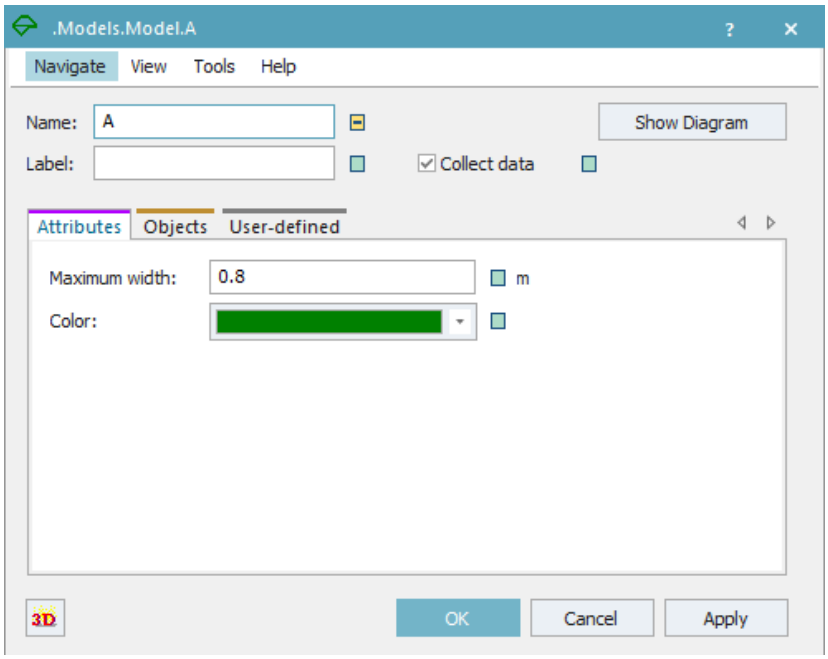
Slika 30 Usporedba StationTable i AttributeExplorer

Jedan od pokazatelja ispravnosti rada simulacijskog modela jest da se količina zapisanih istih objekata u tablici *StationTable* podudara s brojem koji za taj objekt prikazuje *AttributeExplorer* (slika 30).

Primjer:

U tablici *StationTable* promatra se koliko se puta pojavljuje objekt „S2“. Dakle, za proizvod „A“ objekt „S2“ pojavljuje se jednom, za proizvod „B“ jednom, za „C“ dva puta, za „D“ također jednom i za proizvod „AB“ ni jednom (nula puta). Sve ukupno, objekt „S2“ pojavljuje se pet puta, što se podudara s rezultatom u *AttributeExploreru.*

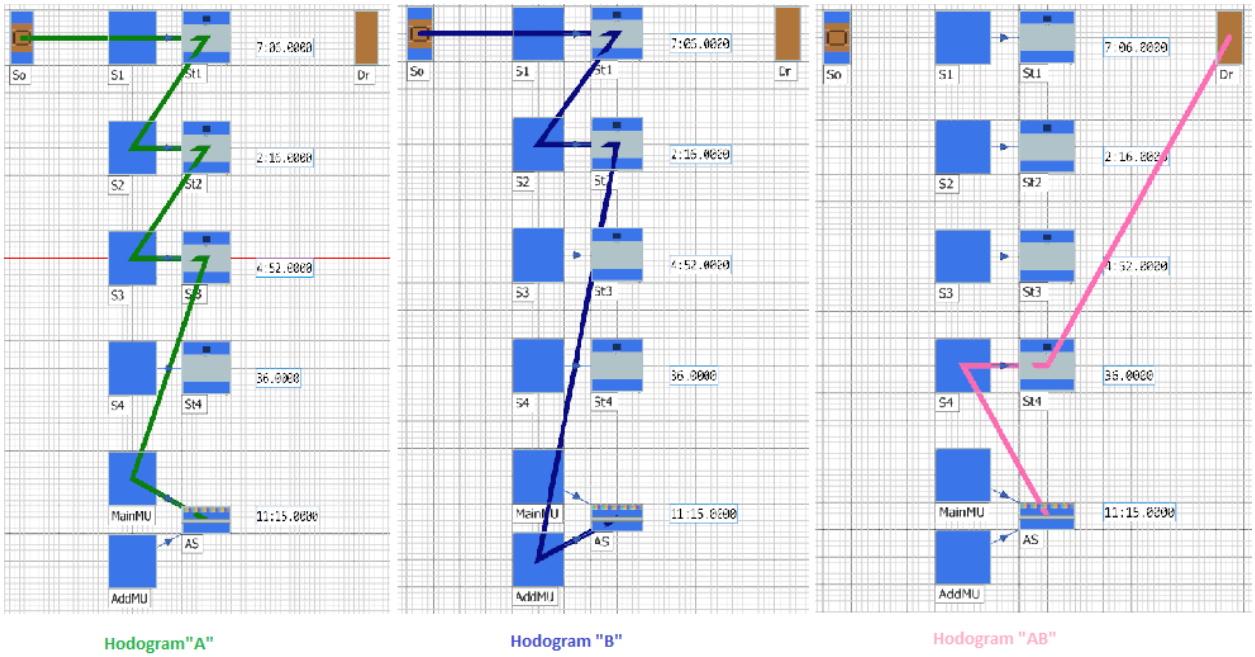
Kao dodatni način provjere ispravnosti, uključit će se grafički prikaz toka proizvoda, odnosno *SankeyDiagram*, koji se nalazi u traci *User Interface*. Za svaki od proizvoda postavit će se po jedan dijagram (u ovom slučaju 5 dijagrama). Dvostrukim klikom na lijevu tipku miša odredit će se postavke za svaki od dijagrama. Primjer je dan na slici 31 za proizvod A..



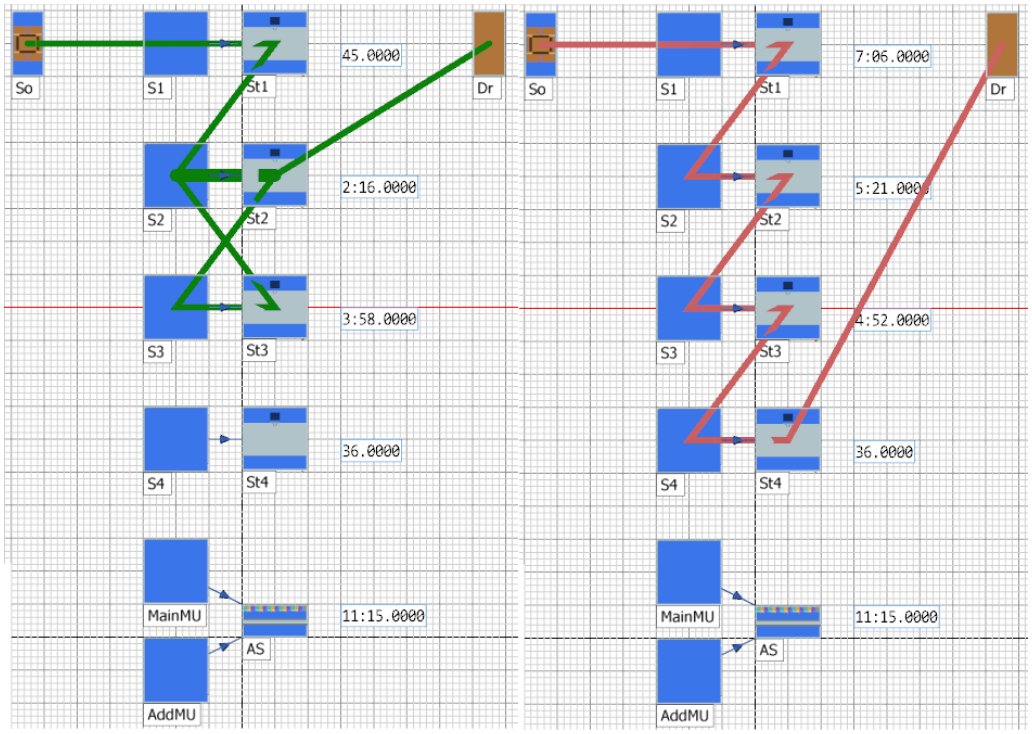
Slika 31 SankeyDiagram - postavke

Dijagrame će se nazavati prema imenu proizvoda koji će im se pridružiti te će se svakom postaviti drugačija boja kako bi se lakše razlikovali. Proizvode je najlakše pridružiti na način da se iz

ClassLibraryja povuče određeni proizvod na ikonu dijagrama. Klikom na *Show Diagram*, nakon pokretanja simulacijskog eksperimenta, uključuje se prikaz dijagrama (slike 32 i 33) iz kojih se jasno može vidjeti kretanje pojedinih proizvoda te ih se može usporediti s tablicom *StationTable*.



Slika 32 Hodogram "A", "B" i "AB"



Slika 33 Hodogram "C" i "D"

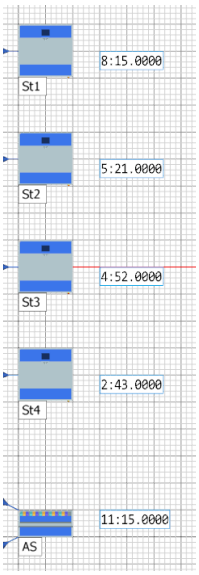
Primjer: (Slika 32)

Proizvod „A“ prema prikazanom dijagramu ima sljedeći hod prema stanicama:

1) „So“ - ulaz
2) „S1“ - radna stanica
3) „S2“ - radna stanica
4) „S3“ - radna stanica
5) „MainMU“ - međuskladište montažne stanice

Taj redoslijed podudara se s tablicom *StationTable*, što je još jedan od pokazatelja ispravnosti rada izrađenog simulacijskog modela.

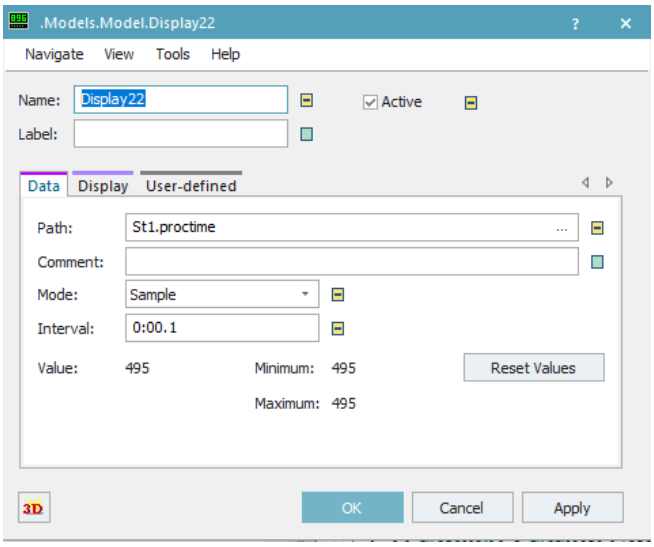
Još jedan način praćenja ispravnosti bit će pomoću objekta *Display* (prikazivač) koji se nalazi pod trakom *User Interface*. Postavit će se po jedan *Display* za svaku od radnih stanica na sljedeći način (slika 34).



Slika 34 Display

Cilj je svakog od ovih prikazivača da u svakom trenutku provedenog simulacijskog eksperimenta prikazuje vrijeme operacije na pridruženoj radnoj stanici. Kako bi se to ostvarilo, potrebno je u postavkama zadati put do određenog atributa koji će se prikazivati. Na slici 35 prikazane su

postavke jednog od prikazivača. Za put (*Path*) potrebno je upisati atribut koji označava vrijeme operacije, formata „stanica.proctime“. Konkretno za ovaj slučaj, treba upisati „St1.proctime“. Unutar ćelije *Mode* postavi se *Sample*, jer u protivnom neće biti moguće mjerenje vremena (ako se postavi *Watch*). Najvažnija je stvar kod postavljanja ovih prikazivača *Interval* postavka.



Slika 35 Display postavke

Standardni interval za mjerenje vremena jedna je sekunda te će se, ako se ostavi na toj postavci, svi rezultati mjerenja vremena produžiti upravo za tu jednu sekundu. To se događa zbog toga što interval obnavljanja, koji je na *EventControlleru,* iznosi 0,2 sekunde. Odnosno, *EventController* zaustavit će simulaciju 0,2 sekunde nakon što se obavi zadnja akcija, tj. nakon što 0,2 sekunde ne bude ikakve promjene unutar simulacijskog modela. S obzirom na to da je započeta promjena na *Displayu* i traje jednu sekundu, što je duže od 0,2 sekunde, *EventController* neće se zaustaviti još dodatnu sekundu. Za rješenje ovog problema potrebno je pod *Interval* unutar postavki za *Display* upisati vrijednost manju od 0,2 sekunde. To će povećati broj intervala, ali će zadnji interval upasti u interval obnavljanja *EventControllera*. Konkretno, upisat ćemo 0,1 sekundu te na taj način biti sigurni da se vrijeme neće mijenjati.

4.2.2. POSTUPAK VALIDIRANJA SIMULACIJSKOG MODELA

Validacija simulacijskog modela označava potvrdu točnosti rezultata simulacijskog eksperimenta dobivenih na određenom simulacijskom modelu. Konkretno, za ovaj slučaj validacija će se

provesti na način da će se ručno izračunati vrijeme proizvodnje svakog pojedinog proizvoda. U račun će se uzeti isključivo vrijeme operacije jer unutar simulacijskog modela nismo uključili transportna sredstva, samim time ni transportna vremena. Također, zbog jednostavnosti i lakše usporedbe računat će se proizvod po proizvod, a ne svi skupa. Izračunate rezultate potrebno je usporediti s onima koje zabilježi *EventController*. Postupak računanja provest će se na sljedeći način. Iz tablice *TimeTable* (slika 36) zbrojit će se sva vremena u pojedinim stupcima, odnosno za pojedini proizvod.

.Models.Model.TimeTable

	time 1	time 2	time 3	time 4	time 5
string	A	B	C	D	AB
1	8:15.0000	1:22.0000	45.0000	7:06.0000	11:15.0000
2	3:40.0000	3:55.0000	5:17.0000	5:21.0000	2:43.0000
3	6:03.0000		3:58.0000	4:52.0000	
4			2:16.0000	36.0000	
5					
6					
7					
8					
9					

Slika 36 TimeTable

Ukupno strojno vrijeme t_{uk}:

$$t_{uk} = t_1 + t_2 + t_3 + t_i + \cdots + t_n$$

Gdje je:

t_{uk} ukupno vrijeme operacija *i*-tog proizvoda

t_i vrijeme *i*-te operacije

Proizvod A:

$$t_{ukA} = t_1 + t_2 + t_3 = 8{:}15 + 3{:}40 + 6{:}03$$

$$t_{ukA} = 17{:}58 \text{ min}$$

Proizvod B:

$$t_{ukB} = t_1 + t_2 = 1{:}22 + 3{:}55$$

$$t_{ukB} = 5{:}17 \text{ min}$$

Proizvod C:

$$t_{ukC} = t_1 + t_2 + t_3 + t_4 = 0{:}45 + 5{:}17 + 3{:}58 + 2{:}16$$

$$t_{ukC} = 12{:}16 \text{ min}$$

Proizvod D:

$$t_{ukD} = t_1 + t_2 + t_3 + t_4 = 7{:}06 + 5{:}21 + 4{:}52 + 0{:}36$$

$$t_{ukD} = 17{:}55 \text{ min}$$

Proizvod AB:

$$t_{ukAB} = t_1 + t_2 = 11{:}15 + 2{:}43$$

$$t_{ukAB} = 13{:}58 \text{ min}$$

Budući da su sada poznata ukupna vremena operacija pojedinih proizvoda, u tablicu *DataTable*, kao na slici 37, upisuje se 1 za željenu količinu proizvoda A. Količina je ostalih proizvoda 0.

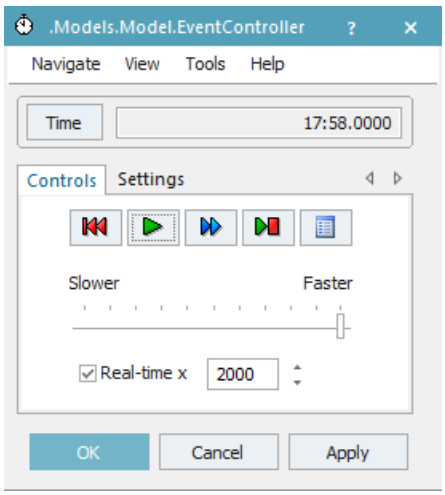
.Models.Model.DataTable

	object 1	integer 2	string 3	table 4
string	MU	Number	Name	Attributes
1	.UserObjects.A	1	A	
2	.UserObjects.B	0	B	
3	.UserObjects.c	0	C	
4	.UserObjects.D	0	D	
5				

Slika 37 DataTable za validaciju (proizvod A)

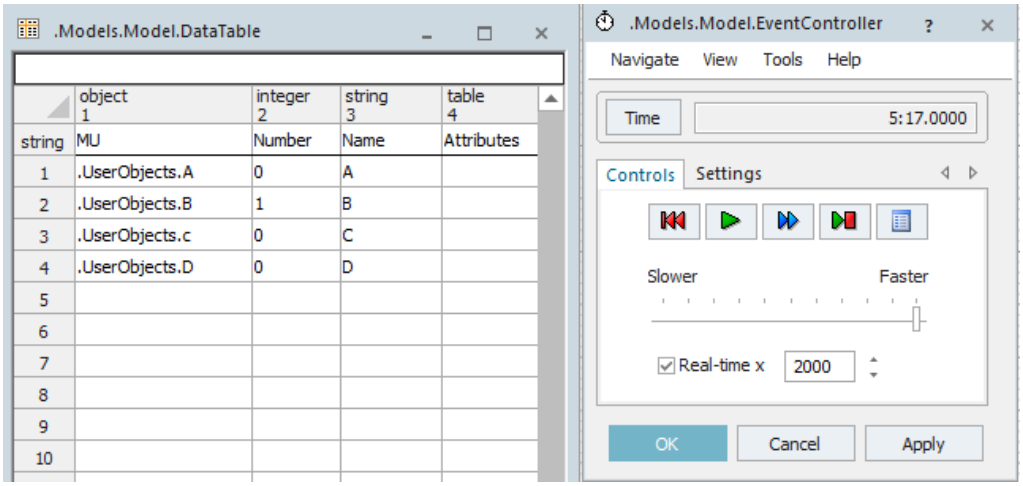
Tada se pokreće simulacijski eksperiment te se očitava vrijeme ispisano u *EventControlleru.* Vrijeme potrebno da se odradi simulacija jest 17:58. Format vremena koji se koristi unutar

korištenog softvera jest dd:hh:mm:ss, odnosno dani:sati:minute:sekunde. Prema tome, prikazan rezultat (slika 38) predstavlja vrijeme od 17 minuta i 58 sekundi, a to se vrijeme poklapa s onim ručno izračunatim vremenom.

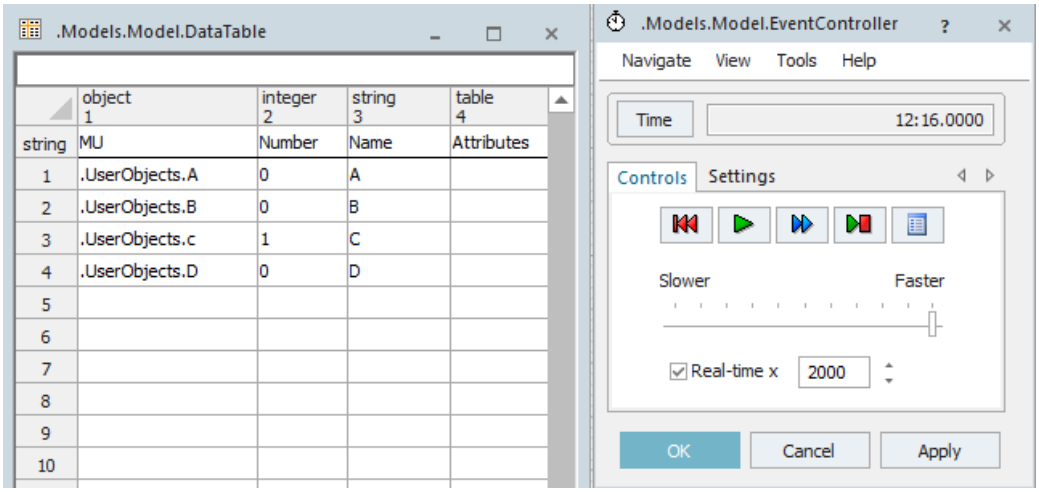


Slika 38 EventController - vrijeme izrade za proizvod A

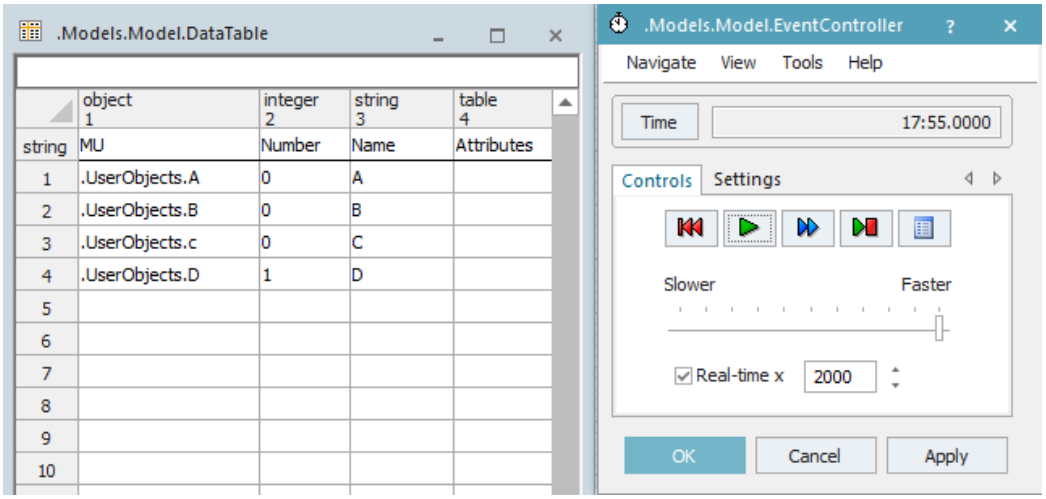
Kontrolirat će se i vremena za ostale proizvode. Potrebno je postaviti da se proizvodi samo jedan proizvod B te izmjeriti vrijeme na objektu *EventController.* Iz usporedbe rezultata s objekta *EventController* i onih ručno izračunatih, vidljivo je da se vremena podudaraju i za proizvod B, kao i za proizvode C i D, što je prikazano na slikama 39, 40 i 41.



Slika 39 Vrijeme simulacije za B



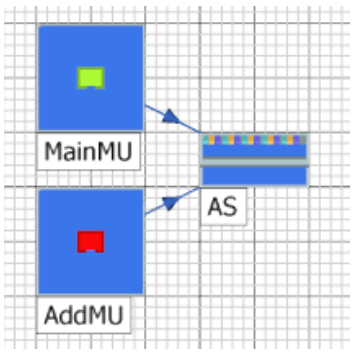
Slika 40 Vrijeme simulacije za C



Slika 41 Vrijeme simulacije za D

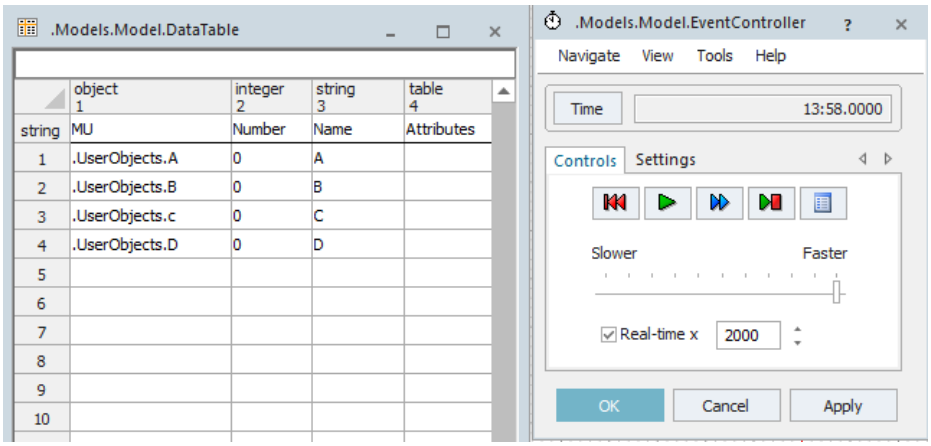
Jedini dio simulacijskog modela koji nije validiran zapravo je montažna stanica. Za njezinu validaciju koristit će se drugačiji postupak ubacivanja proizvoda u proces. Budući da se na montažnoj stanici *„AS“* spajaju proizvodi A i B u proizvod AB, pomoću tog će se objekta i vršiti validacija.

Iz *ClassLibraryja* povući će se jedan proizvod *A* te će se postaviti na međuskladište *„MainMU“* (*drag&drop*). Zatim će se istim postupkom povući proizvod *B* na međuskladište *„AddMU“*. Zbog lakšeg raspoznavanja (slika 42), proizvodu A dodijeljena je zelena boja, a proizvodu B crvena boja.



Slika 42 Proizvodi A i B postavljeni u međuskladišta

Za ovaj način ubacivanja proizvoda u proces stvaranje entiteta na izvoru mora biti isključeno kako se ne bi stvarali dodatni procesi i računala njihova vremena. To se postiže na način da se u tablicu *DataTable* za sve proizvode unese količina nula kao na slici 43. Sada se pokrene simulacijski eksperiment i očita vrijeme. Vidljivo je da se i vrijeme za proizvod AB podudara s ranije ručno izračunatim vremenom. Stoga se nakon ovog koraka može s velikom sigurnošću potvrditi da simulacijski model daje dobra i istinita rješenja, odnosno može se reći da je simulacijski model validan.

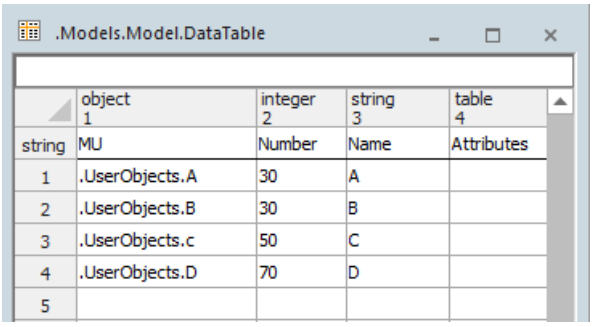


Slika 43 Vrijeme simulacije za AB

4.3. UNOS IDEJNOG PROBLEMA U RAČUNALNI SIMULACIJSKI MODEL

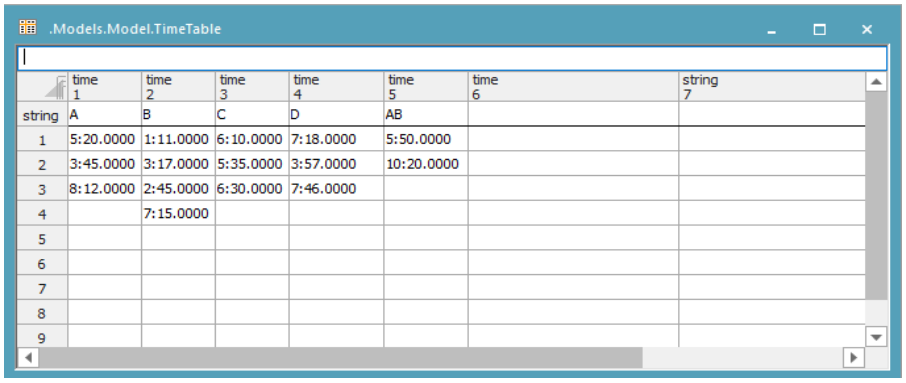
Kako je objašnjeno u poglavlju 3.1.2., svi podatci potrebni za izvršavanje računalne simulacije unose se u tri tablice (DataTable, TimeTable i StationTable). Način unošenja podataka također je ranije objašnjen. U *DataTable* unese se tražena količina proizvoda (slika 44). Budući da postoji

zahtjev za 30 proizvoda „AB“, u tablicu *DataTable* unosi se zahtjev za izradu 30 jedinica proizvoda „A“ i 30 jedinica proizvoda „B“. Proizvodi „C“ i „D“ ne idu na montažu te se njihov zahtjev zapisuje kako je i zadano u *tablici 3*, a zahtijeva se 50 jedinica proizvoda „C“ i 70 jedinica proizvoda „D“. Zapisivanje vremena operacija u tablicu *TimeTable* radi se na način da ih se zapisuje redom kako slijede operacije za pojedini proizvod, pri tome ne ostavljajući prazna mjesta, kako je vidljivo na slici 45.



	object 1	integer 2	string 3	table 4
string	MU	Number	Name	Attributes
1	.UserObjects.A	30	A	
2	.UserObjects.B	30	B	
3	.UserObjects.c	50	C	
4	.UserObjects.D	70	D	
5				

Slika 44 DataTable za traženu količinu



	time 1	time 2	time 3	time 4	time 5	time 6	string 7
string	A	B	C	D	AB		
1	5:20.0000	1:11.0000	6:10.0000	7:18.0000	5:50.0000		
2	3:45.0000	3:17.0000	5:35.0000	3:57.0000	10:20.0000		
3	8:12.0000	2:45.0000	6:30.0000	7:46.0000			
4		7:15.0000					
5							
6							
7							
8							
9							

Slika 45 TimeTable za traženu narudžbu

Količina proizvoda i vremena operacija u ovom se slučaju ne mijenjaju. Dakle, tablice *DataTable* i *TimeTable* ostaju nepromijenjene i pomoću njih se ne smije manipulirati ukupnim trajanjem procesa. Nakon unošenja podataka o količini proizvoda i vremenu operacija, potrebno je odrediti tok materijala u tablici *StationTable* unošenjem naziva pojedinih strojeva redom.

4.3.1. UNOS I ODREĐIVANJE TOKA MATERIJALA

Ključnu ulogu u rješavanju promatranog problema ima tok materijala, odnosno hodogram proizvodnje svakog pojedinog proizvoda. Za početak, pretpostavit ćemo neki početni tok materijala te ispitati njegove nedostatke i moguća poboljšanja. Na osnovi dobivenih rezultata pokušat će se pronaći bolja rješenja te ih potvrditi računalnom simulacijom.

4.3.2. INICIJALNI TOK MATERIJALA

Prateći Tablicu 2 ispunjava se *StationTable*. Pri tome je važno paziti na vrstu operacije, odnosno radi li se o tokarenju, glodanju ili montaži, jer o tome ovisi raspoloživost tog radnog mjesta. Prvi hodogram bit će usklađen prema rednom broju stanice, odnosno pokušat će se pratiti redoslijed S1, S2, S3 pa S4, ako to dozvoljava vrsta operacije (slika 46).

.Models.Model.StationTable

	string 1	string 2	string 3	string 4	string 5	string 6
string	A	B	C	D	AB	
1	S1	S2	S1	S2	AS	
2	S2	S3	S2	S3	S2	
3	S3	S4	S3	S4	Dr	
4	MainMU	S2	Dr	Dr		
5		AddMU				
6						

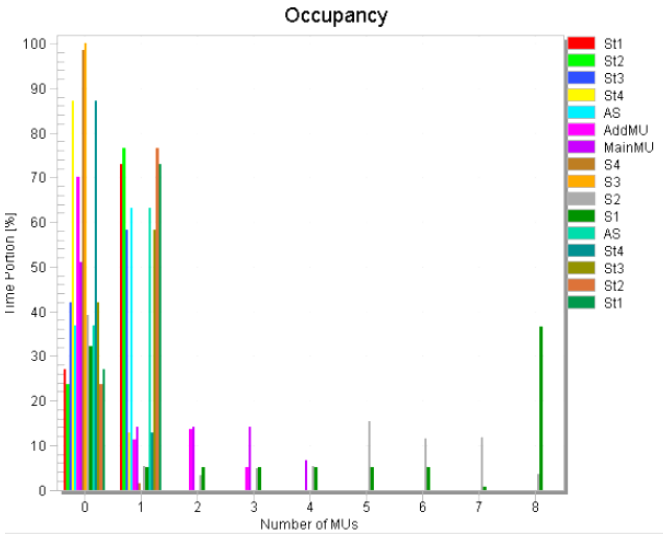
Slika 46 StationTable za početni tok materijala

5. INTERPRETACIJA I ANALIZA REZULTATA

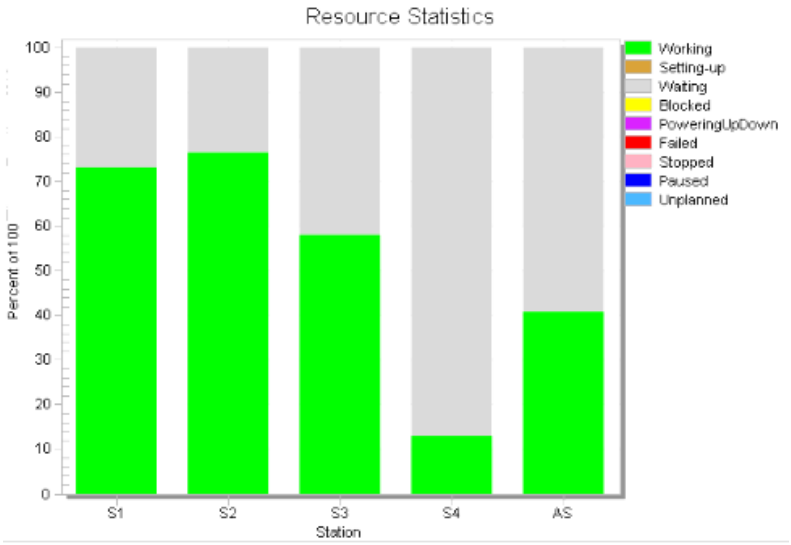
Rezultati računalnih simulacija ovog tipa mogu se interpretirati na više načina, ovisno o tome što se želi promatrati. Dalje će biti prikazano nekoliko načina s kojima se može pratiti određene parametre, odnosno neki su već prikazani u prijašnjim poglavljima (*AttributeExplorer* i *Sankey diagram)*.

5.1. OPĆA ANALIZA REZULTATA

Jedan od najefikasnijih načina interpretacije jesu dijagrami. Nijh je moguće kreirati na način da se iz kartice *User Interface* postavi objekt pod nazivom *Chart* unutar simulacijskog modela. Zatim se odaberu svi objekti koji se želi promatrati te ih se metodom *Drag&Drop* povuče na *Chart.* Ulaskom u postavke kreiranog objekta te klikom na *Show Chart*, aktivira se grafički prikaz tog objekta. Na slikama 47 i 48 prikazani su grafovi okupiranosti pojedinih objekata u ovisnosti o broju proizvoda te raspoloživosti pojedinih objekata unutar promatranog procesa/sustava. Iz takvog se načina interpretacije rezultata analizom može doći do zaključka o tome koji su objekti najviše opterećeni, odnosno na kojima se stvaraju zastoji i/ili uska grla. Pomoću toga može se pokušati preusmjeriti tok materijala na druge objekte (ako to dozvoljava tehnološki proces) koji su manje okupirani te tako rasteretiti kompletno proizvodno postrojenje.

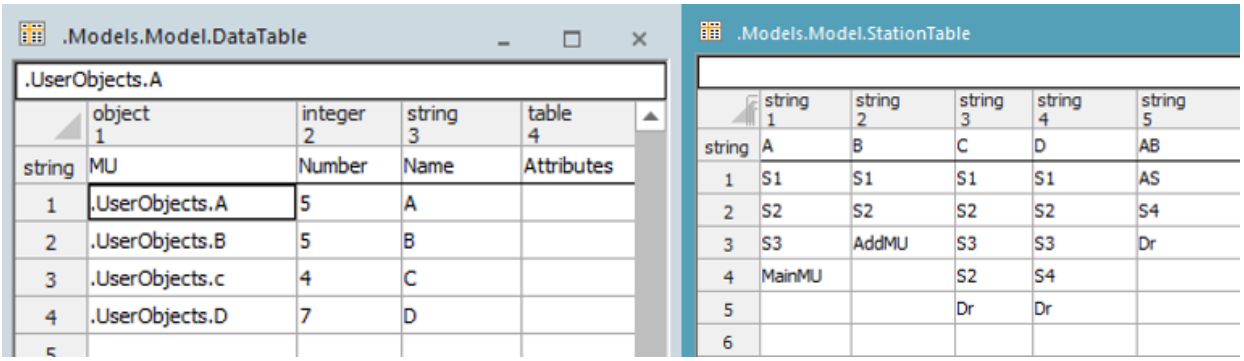


Slika 47 Graf okupiranosti proizvodnih kapaciteta



Slika 48 Graf iskorištenja proizvodnih kapaciteta

Gore prikazani grafovi rezultati su zahtjeva količine proizvoda i raspodjele radnih mjesta prema sljedećim tablicama.



.Models.Model.DataTable

.UserObjects.A

	object 1	integer 2	string 3	table 4
string	MU	Number	Name	Attributes
1	.UserObjects.A	5	A	
2	.UserObjects.B	5	B	
3	.UserObjects.c	4	C	
4	.UserObjects.D	7	D	
5				

.Models.Model.StationTable

	string 1	string 2	string 3	string 4	string 5
string	A	B	C	D	AB
1	S1	S1	S1	S1	AS
2	S2	S2	S2	S2	S4
3	S3	AddMU	S3	S3	Dr
4	MainMU		S2	S4	
5			Dr	Dr	
6					

Slika 49 Zahtjevi za proizvodnju 1

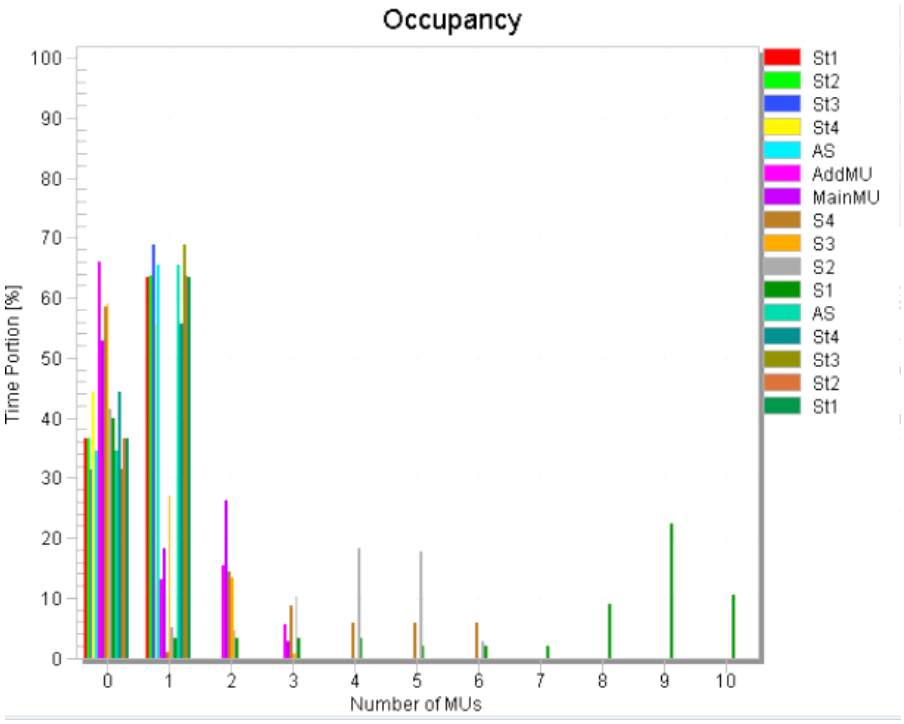
Konkretno, u ovom je slučaju vidljivo da je najviše zauzeta radna stanica „S2“ te nešto manje radna stanica „S1“. Također, na tim objektima nema zastoja, što je dobra vijest jer pruža visoku okupiranost i kratak prazan hod. Dakle, iz toga se vidi da bi na neki način trebalo dodatno rasteretiti te dvije radne stanice. Pokušaj rasterećenja ide s pretpostavkom da su sve radne stanice univerzalne te da mogu obavljati sve vrste operacija. U cilju rasterećenja promjenit će se tablica koja prikazuje redoslijed odvijanja pojedinih aktivnosti. Novi tok materijala bit će kao što je prikazan na slici 50.

.Models.Model.StationTable

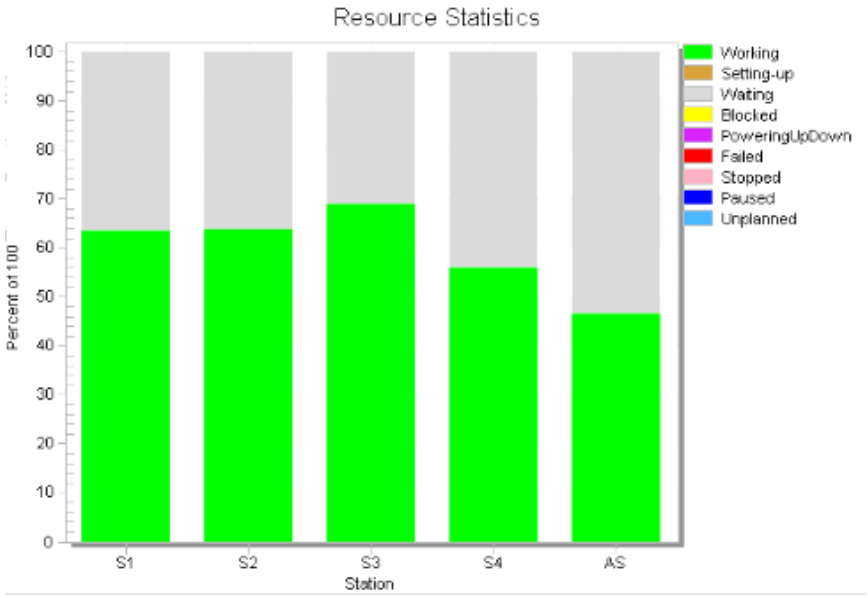
	string 1	string 2	string 3	string 4	string 5
string	A	B	C	D	AB
1	S1	S1	S3	S4	AS
2	S2	S1	S2	S2	S4
3	S3	AddMU	S3	S3	Dr
4	MainMU		S1	S4	
5			Dr	Dr	
6					

Slika 50 Preraspoređeni tok materijala

Gore prikazani raspored radnih mjesta puno je pogodniji te rasterećuje radna mjesta i uklanja zastoje. Jedini je problem što se za takav raspored treba povećati kapacitet međuskladišta s 8 na 10 jedinica proizvoda, kako ne bi dolazilo do blokade zbog povratnog hoda. Iz grafova na slici 51 i 52 vidljivo je kako su potencijalne blokade i nastanci uskih grla eliminirani. Također je usklađena okupiranost svih pojedinih strojeva. Isto tako, može se vidjeti da se broj proizvoda na određenim međuskladištima povećao. Ako se može ispuniti zahtjev za povećanjem međuskladišta, na taj se način može poboljšati realni sustav.



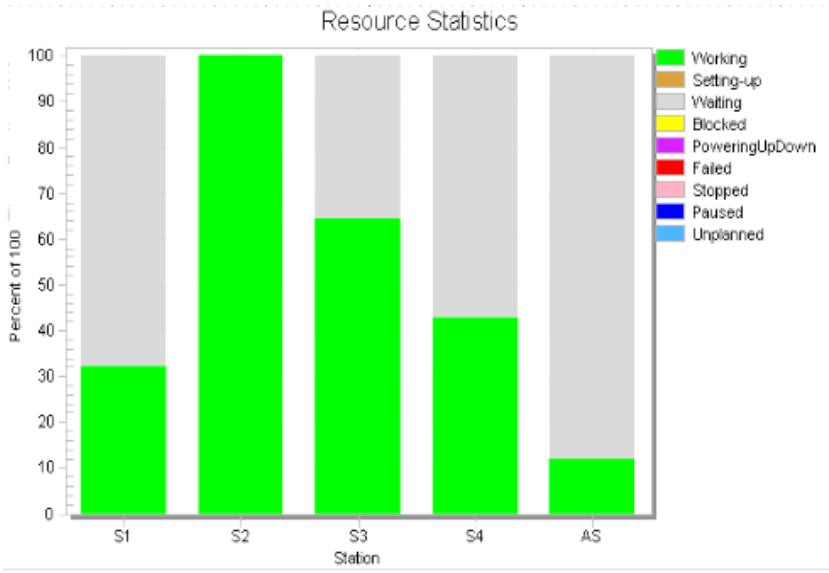
Slika 51 Grafovi okupiranosti proizvodnih kapaciteta - preraspoređeni tok



Slika 52 Graf iskorištenja proizvodnih kapaciteta - preraspoređeni tok

5.2. RJEŠENJE IDEJNOG PROBLEMA

Kao rezultat hodograma inicijalnog toka (poglavlje 4.3.2.), dobije se ukupno vrijeme trajanja kompletnog procesa izrade 1 dan i 26 minuta. To je vrijeme koje treba pokušati smanjiti u svrhu što brže isporuke naručene količine proizvoda. Za pronalazak kritičnog dijela, odnosno radne stanice koja je najviše opterećena, koristi se rezultati grafa udjela poslova na određenoj radnoj stanici (slika 53).



Slika 53 Udio poslova po radnim stanicama

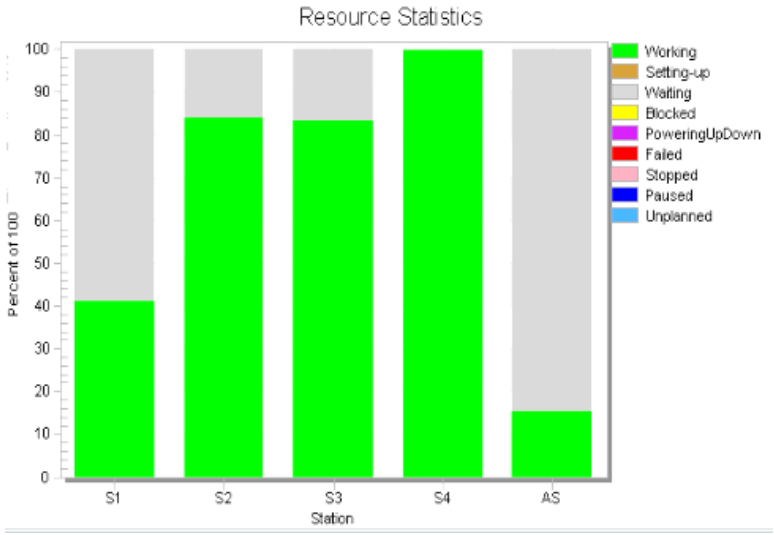
5.2.1. PRVO POBOLJŠANJE TOKA MATERIJALA

Iz grafa na slici 53 vidljivo je kako je stanica „S2“ cijelo vrijeme zauzeta. Stoga će se pokušati prebaciti dio operacija sa stanice „S2“ (glodalica) na neku manje zauzetu radnu stanicu – glodalicu, točnije na „S4“ (najmanje zauzeta glodalica). Prema slici 54, u tablici *StationTable* mijenja se podatak o radnoj stanici koja obavlja prvu operaciju za proizvod „D“ jer proizvod „D“ ima najviše jedinica proizvoda te je vrijeme za obavljanje operacije najveće. Tako izmijenjeni tok materijala dovodi do znatnog poboljšanja. Ukupno vrijeme proizvodnje iznosi 18 sati i 58 minuta, što je za 5 sati i 28 minuta manje od prethodnog rasporeda. Ako se pogleda udio zauzetosti stanica, vidljivo je da se razlika na glodalicama (S1, S2 i S3) ujednačila (slika 55).

.Models.Model.StationTable

	string 1	string 2	string 3	string 4	string 5	string 6
string	A	B	C	D	AB	
1	S1	S2	S1	S4	AS	
2	S2	S3	S2	S3	S2	
3	S3	S4	S3	S4	Dr	
4	MainMU	S2	Dr	Dr		
5		AddMU				
6						

Slika 54 Prva izmjena toka materijala



Slika 55 Udio poslova po radnim stanicama nakon prve izmjene toka materijala

5.2.2. DRUGO POBOLJŠANJE TOKA MATERIJALA

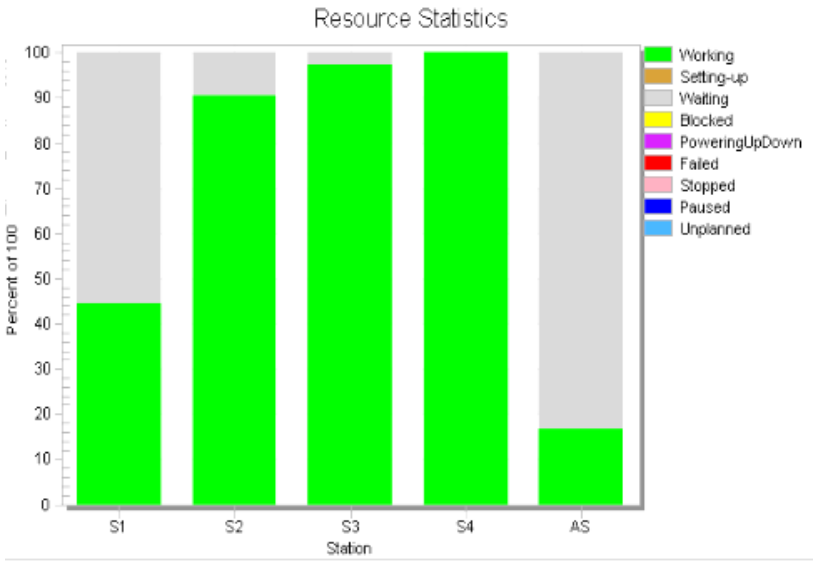
Sljedeći pokušaj poboljšanja bit će usmjeren prema rasterećenju „S4“. Treću operaciju proizvoda „B“ preselit će se s radne stanice „S4“ na radnu stanicu „S3“, a to je prikazano slikom 56.

.Models.Model.StationTable

	string 1	string 2	string 3	string 4	string 5
string	A	B	C	D	AB
1	S1	S2	S1	S4	AS
2	S2	S3	S2	S3	S2
3	S3	S3	S3	S4	Dr
4	MainMU	S2	Dr	Dr	
5		AddMU			
6					

Slika 56 Druga izmjena toka materijala

Takav tok materijala omogućuje još jedno poboljšanje u vidu konačnog vremena trajanja proizvodnje. Sada je vrijeme 1 sat i 23 minute manje od prijašnjeg, odnosno iznosi 17 sati i 35 minuta.



Slika 57 Udio poslova po radnim stanicama nakon druge izmjene toka materijala

5.2.3. KONAČNI TOK MATERIJALA

Budući da su svi udjeli zauzetosti na radnim stanicama – glodalicama (S2, S3 i S4) iznad 90%, rezultat se može proglasiti zadovoljavajućim i zaključiti konačni hodogram materijala za sve pojedine proizvode. Stanice „S1“ i „AS“ ne mogu postići veći udio zauzetosti jer se na njima odvijaju točno određene operacije. Ako želimo ispitati postojanost još kraćih vremena, potrebno je primijeniti neku od tehnika optimiziranja kojom se to može postići ili rješenje pronaći u samom tehnološkom procesu izrade dijelova, odnosno optimizaciji njihovih vremena operacija. Na slici 58 prikazan je konačni redoslijed operacija svakog pojedinog proizvoda. Također se može iščitati vrijeme svake pojedine operacije. Slijedi detaljno iščitavanje redoslijeda aktivnosti za svaki pojedini tip proizvoda u formatu: broj operacije (OPxx), vrsta operacije, vrijeme operacije, radna stanica.

Br.	Vrsta operacije	Oznaka opreme	Proizvod					Vrijeme (min)				
			A	B	AB	C	D	t_A	t_B	t_{AB}	t_C	t_D
0.	Skladištenje	So						--	--	--	--	--
1.	Tokarenje	S1	10			10		5:20	--	--	6:10	--
2.	Glodanje	S2	20	10 40	20	20		3:45	1:11 7:15	10:20	5:35	--
3.	Glodanje	S3	30	20 30		30	20	8:12	3:17 2:45	--	6:30	3:57
4.	Glodanje	S4					10 30	--	--	--	--	7:18 7:46
5.	Montaža	AS			10			--	--	5:50	--	--
6.	Skladištenje	Dr						--	--	--	--	--

Slika 58 Redoslijed aktivnosti proizvodnog procesa

Proizvod A:

- Skladište So
- OP10 tokarenje 5:20 stroj S1
- OP20 glodanje 3:45 stroj S2
- OP30 glodanje 8:12 stroj S3
- Međuskladište montažna stanica AS

Proizvod B:

- Skladište So
- OP10 glodanje 1:11 stroj S2
- OP20 glodanje 3:17 stroj S3
- OP30 glodanje 2:45 stroj S3
- OP40 glodanje 7:15 stroj S2
- Međuskladište montažna stanica AS

Proizvod AB:

- Međuskladište montažna stanica AS
- OP10 montaža 5:50 montažna stanica AS
- OP20 glodanje 10:20 stroj S2
- Skladište Dr

Proizvod C:

- Skladište So
- OP10 tokarenje 6:10 stroj S1
- OP20 glodanje 5:35 stroj S2
- OP30 glodanje 6:30 stroj S3
- Skladište Dr

Proizvod D:

- Skladište So
- OP10 glodanje 7:18 stroj S4
- OP20 glodanje 3:57 stroj S3
- OP30 glodanje 7:46 stroj S4
- Skladište Dr

5.2.4. RJEŠENJE I ODGOVOR NA UPIT

Budući da je postignuto dovoljno dobro rješenje, može se usvojiti potrebno vrijeme izrade traženih proizvoda (ukupno strojno vrijeme). Na to vrijeme potrebno je dodati još vremena za dobavu repromaterijala i potrebnih alata te pripremno-raspremna vremena. Također, treba uračunati

vrijeme potrebno za otklon potencijalnog problema/kvara. Ukupna suma tih vremena treba biti odgovor na upit o vremenu isporuke tražene narudžbe.

Nakon što klijent prihvati ponudu, redoslijed aktivnosti (slika 58) može se poslati u proizvodnju kao konačni tok materijala prema kojem se organizira proizvodni proces.

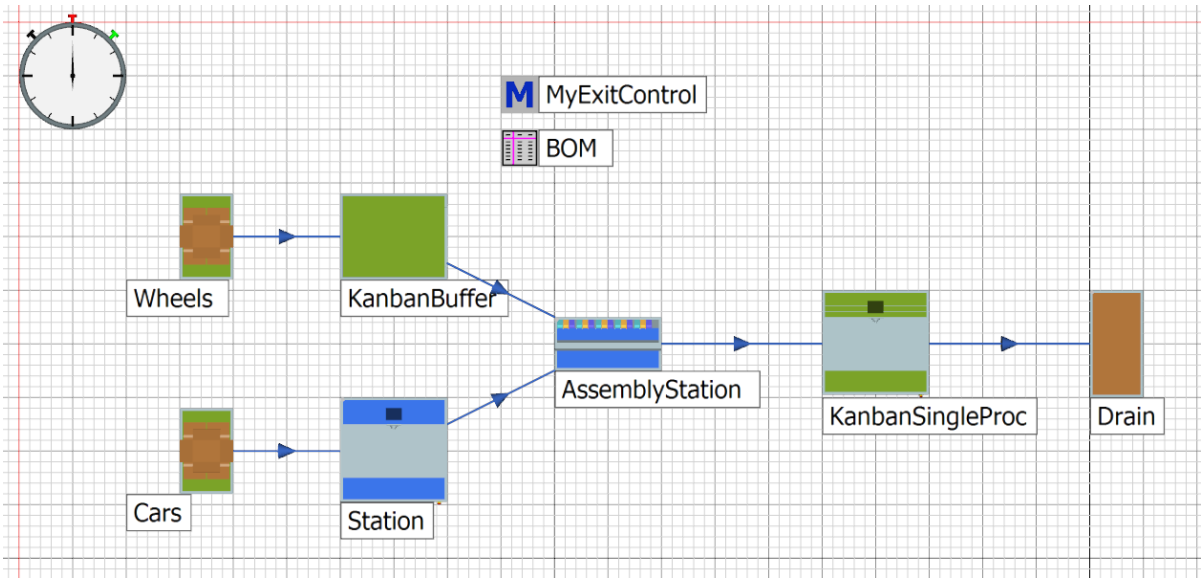
6. ALTERNATIVNO RJEŠENJE

Prilikom osmišljanja rješenja za prethodno navedeni problem, koji je riješen pomoću simulacijskog modela, pojavile su se neke od dodatnih ideja. Alternativna ideja bila je implementacija kanban metode pomoću koje bi se naručivali proizvodi.

6.1. KANBAN

Ideja kanban metode leži u karticama pomoću kojih se može naručivati potreban repromaterijal. Proizvodnja kreće tek nakon što se utvrdi konačna narudžba. Isto tako, koristeći *Just-in-time* (JIT) metodu u kombinaciji s *Kanban* metodom postiže se zahtjev da se proizvede točno potreban broj proizvoda.

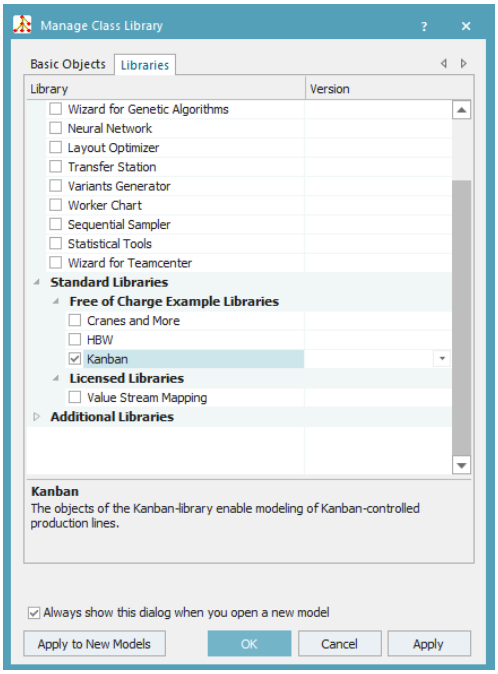
U daljnjem primjeru pokazat će se način izrade i funkcije simulacijskog kanban modela (slika 59) u kojemu će se proizvoditi dvije vrste proizvoda (u ovom slučaju automobila) te će se na njih, ovisno o vrsti, montirati potrebna vrsta kotača.



Slika 59 Simulacijski model - Kanban metoda

6.2. IZRADA SIMULACIJSKOG MODELA

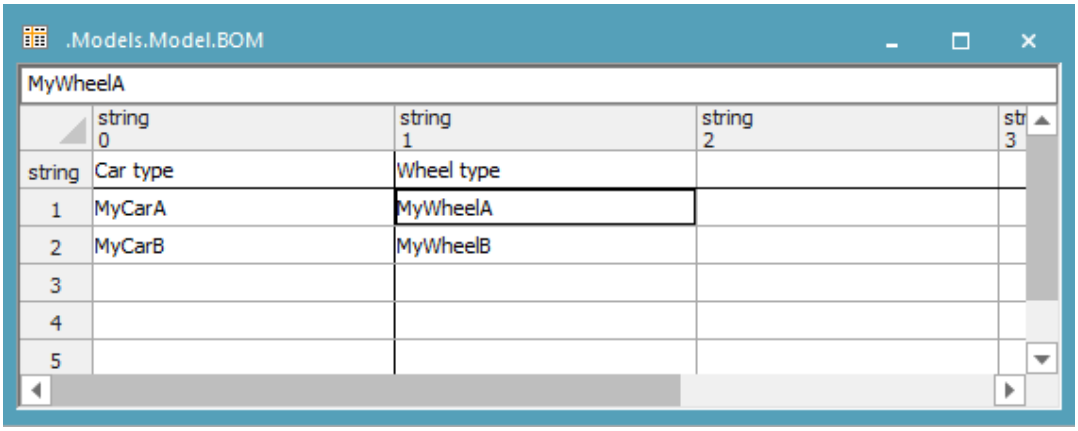
Kako bi se uopće moglo pristupiti kanban elementima, potrebno je prilikom kreiranja novog simulacijskog modela u prozoru *Manage Class Library* pod karticom *Libraries* označiti kućicu pokraj *Kanbana* (na slici 60).



Slika 60 Manage Class Library - Kanban

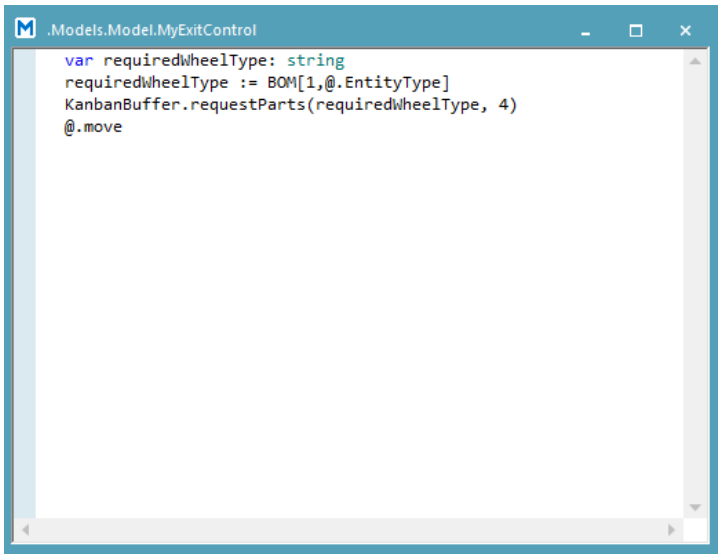
Sada je unutar *Toolbox* odjeljka vidljiva kartica *Tools: Kanban* u kojoj se nalaze elementi koji će se koristiti za izradu takvog simulacijskog modela.

Za početak, potrebno je ubaciti elemente unutar simulacijskog modela prema rasporedu danom na slici 59. Postavit će se dva elementa *KanbanSource* i preimenovati u „Wheels“ i „Cars“. Zatim će se postaviti *KanbanBuffer, Station, AssemblyStation, KanbanSingleProc* i *Drain* te ih povezati s naredbom *Connector* prema rasporedu. Uz ove elemente, potrebno je još postaviti jedan *Method* te ga nazvati „MyExitControl“ i jedan *DataTable* pod nazivom „BOM“. U tablici „BOM“ potrebno je uključiti *ColumnIndex i RowIndex.* Zatim, kao što je prikazano slikom 61, stupce nazvati „Car type“ i „Wheel type“ te ispuniti tablicu na način da se u redove za određeni tip automobila unese potreban tip kotača.



Slika 61 Tablica "BOM"

Također je potrebno stvoriti logičku metodu „MyExitControl“ kao što je prikazano na slici 62. Metoda bi trebala funkcionirati pomoću atributa *.requestParts* koji se nalazi kao sastavni atribut *KanbanBuffera*. Taj atribut odnosi se na zahtjev vrste proizvoda koji trebaju ući na kanban međuskladište te na njihovu količinu. Metoda se aktivira na izlazu iz radne stanice „Station“ gdje je moguće reći koja vrsta automobila ulazi u stanicu za montažu. Dakle, navedena metoda postavit će se kao izlazna metoda na radnoj stanici „Station“,na način da se u postavkama radne stanice pod karticom *Controls* za *Exit* povuče pomoću *drag&drop* metode.



Slika 62 Metoda MyExitControl

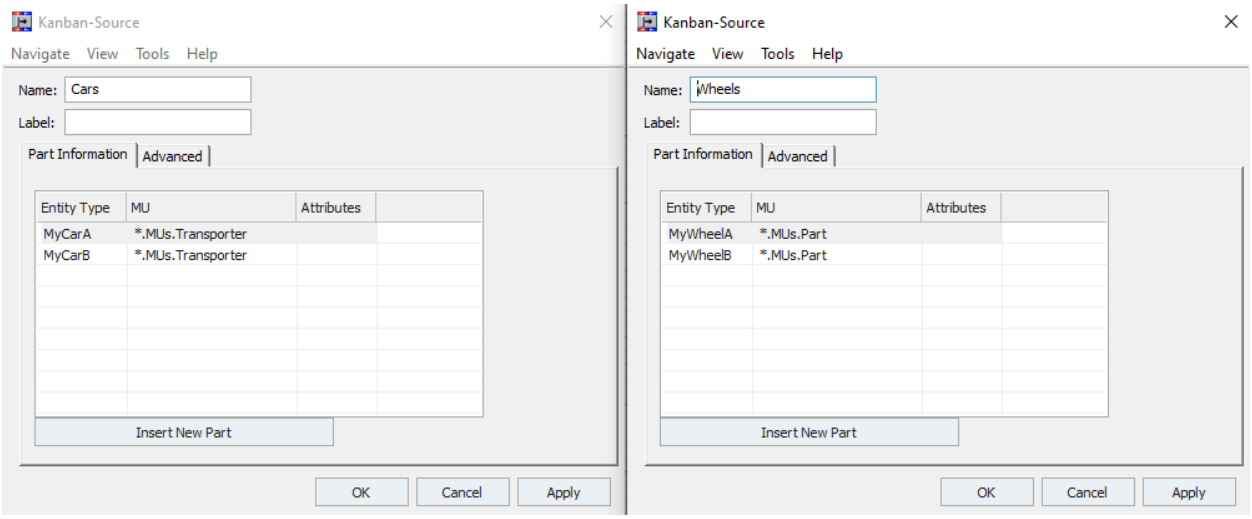
Konkretno, kreirat će se dodatan atribut *requiredWheelType* koji će biti tipa *string*. Taj atribut poprimat će vrijednost iz tablice „BOM“ u ovisnosti o *@.EntityType*, odnosno o tipu proizvoda

koji ulazi u radnu stanicu „Station“. Na osnovi te vrijednosti i vrijednosti količine (4 kotača), pridružit će se vrijednost varijabli *KanbanBuffer.requestParts* (slika 62).

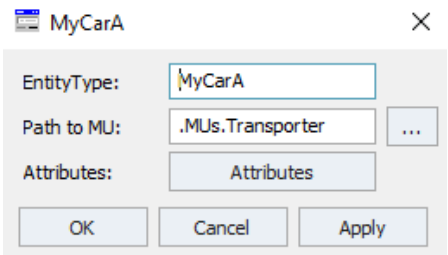
Metoda „MyExitControl“ može se pročitati na sljedeći način:

„Naruči potreban tip kotača u ovisnosti o tipu automobila u količini od 4 komada.“

Postavke za kanban izvore „Cars“ i „Wheels“ dane su slikom 63. Za izvor *„Cars“* odabire se opcija *Insert New Part* (slika 64) te se u ćeliju *EntityType* upisuje ime proizvoda. U ovom slučaju, to su automobili te će se na isti način kreirati dva proizvoda pod nazivom *„MyCarA“* i *„MyCarB“*. U ćeliju *Path to MU* pomoću *drag&drop* metode iz *ClassLibraryja* povući će se objekt *Transporter* jer se radi o automobilu. Treba napomenuti, ako se odabere *Part* umjesto *Transporter,* da se neće moći pridružiti 4 proizvoda (4 kotača) na montažnoj stanici *AssemblyStation.*



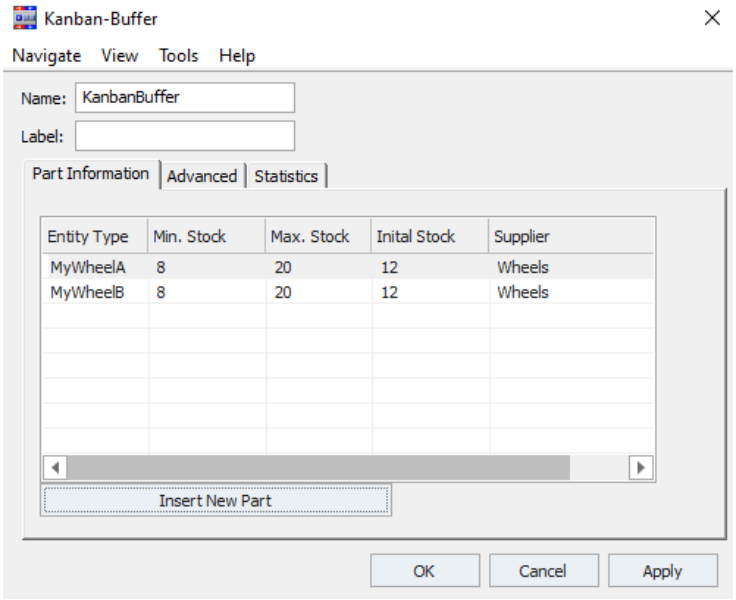
Slika 63 Postavke za kanban izvore - Cars (lijevo), Wheels (desno)



Slika 64 Insert New Part

Na isti način postavit će se i izvor „Wheels“. Jedina razlika bit će u ćeliji *Path to MU*, gdje je potrebno iz *ClassLibraryja* povući objekt *Part* i ta dva proizvoda nazvati *„MyWheelA“* i *„MyWheelB“*.

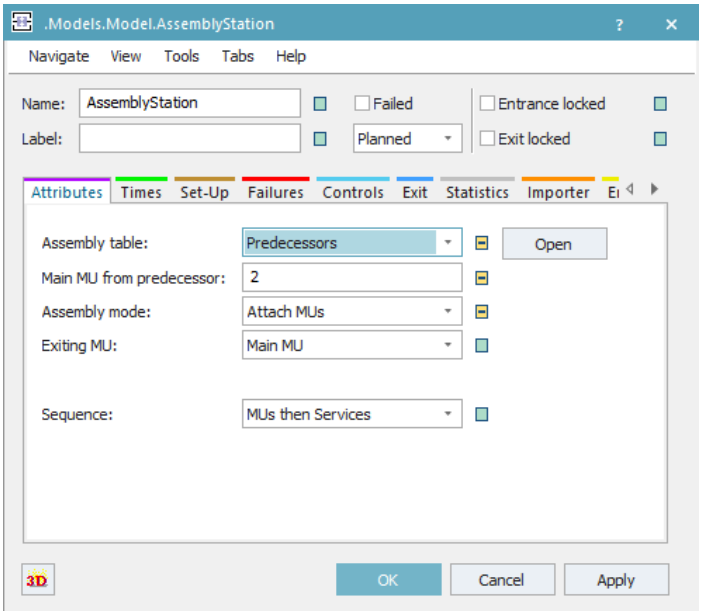
KanbanBuffer zapravo je međuskladište koje ima opciju potraživanja proizvoda, kada proizvod na stanju padne ispod minimalnog broja, i zaustavljanja potražnje, kada broj dosegne maksimalnu vrijednost (slika 65). Stoga je potrebno postaviti listu proizvoda i njihove brojčane granice. Klikom na *Insert New Part* otvara se tablica u koju se unose podatci. U stupac *Entity Type* upisuje se ime proizvoda za koji se postavljaju granice. *Min. Stock* minimalna je vrijednost proizvoda koja aktivira naručivanje novih proizvoda, a *Max. Stock* suprotna granica koja zaustavlja naručivanje. *Inital Stock* označava početnu vrijednost međuskladišta, a u stupac *Supplier* potrebno je upisati s kojeg se objekta dobavljaju naručeni proizvodi, u ovom slučaju to je izvor „Wheels“.



Slika 65 KanbanBuffer - postavke

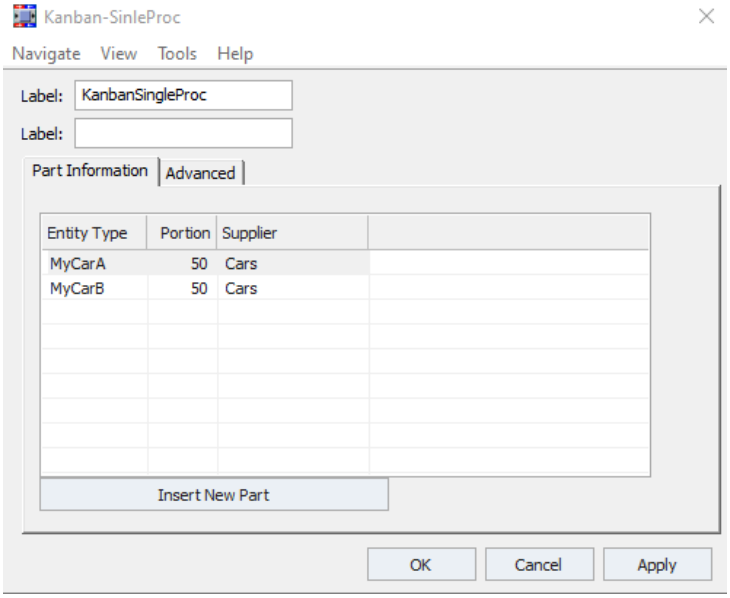
Postavke za montažnu stanicu postavljaju se na isti način kao i u glavnom primjeru u poglavlju 3.1.5. Konkretno, za ovaj slučaj treba postaviti da se montaža odvija prema glavnom proizvodu koji ovisi o tome na kojem se ulazu pojavi. Dakle, za *Assembly table* treba postaviti *Predecessors* i za *Main MU from predecessor* broj 2 jer u ovom slučaju glavni proizvod ulazi pomoću konektora 2, tj. prati se gdje je ulaz proizvoda koji označava automobil. Kotači se dodaju na automobil te se

zbog toga za *Assembly mode* koristi *Attach MU*. Isto tako, proizvod koji izlazi s montaže također je automobil.Stoga se za *Exiting MU* ostavlja *Main MU* (slika 66).



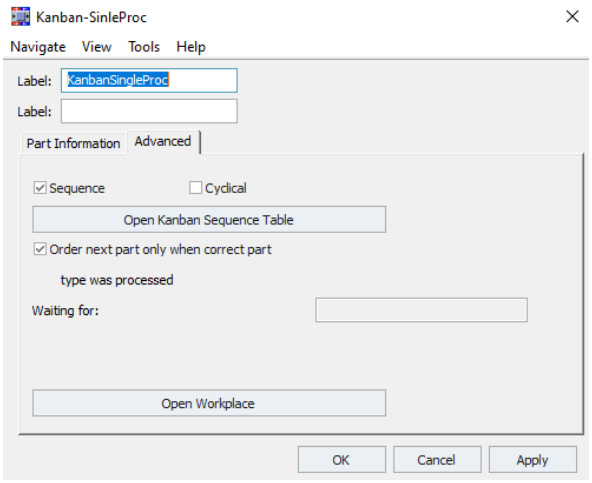
Slika 66 Postavke AssemblyStation

Zadnji element koji se treba postaviti jest *KanbanSingleProc*. To je stanica koja ima mogućnost zahtijevanja određenog broja proizvoda s određenog izvora, odnosno ovisi što je zadano kao *Supplier*. Unutar postavki (slika 67) odabire se opcija *Insert New Part* te se u nastalu tablicu unosi željene proizvode („MyCarA“ i „MyCarB“), njihov omjer u postotcima unosi se u stupac *Portion*, a u stupac *Supplier* unosi se *„Cars“* (ime izvora koji predstavlja nabavu, odnosno na kojemu se kreiraju traženi proizvodi).

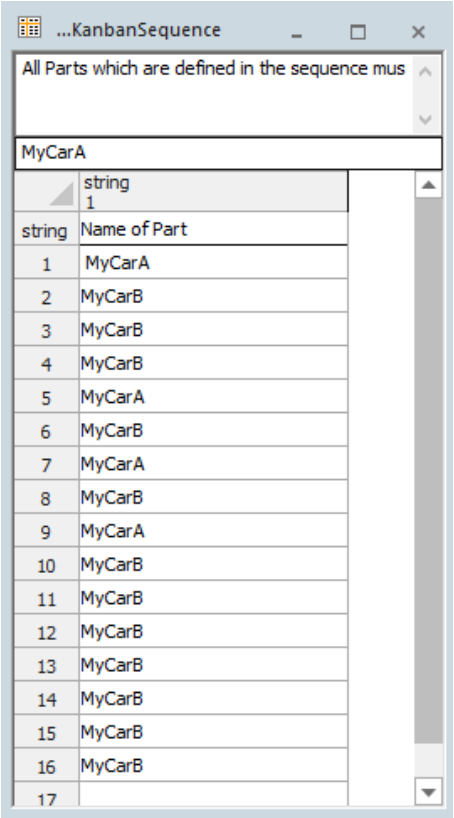


Slika 67 KanbanSngleProc

Ako se pod karticom *Advanced* označi opcija *Sequence*, pojavit će se mogućnost otvaranja *Kanban Sequence Table*, kao što je prikazano slikom 68. U tablicu se redom upisuju željeni proizvodi kako će se proizvoditi, od prvog do zadnjeg. Označavanjem opcije *Cyclical* tablica se ispunjava ciklički od prvog do zadnjeg, ali beskonačan broj puta. Ideja je da korisnik pristupa ovoj tablici (slika 69) i unosi koje proizvode želi i kojim redoslijedom.



Slika 68 KanbanSingleProc - -Advanced



Slika 69 Kanban Sequence Table

Ovaj pristup ispunjava osnovnu zamisao prema krajnjem korisniku da sam unosi željene proizvode, ali sadrži nekoliko problema, kao što su komplikacije ukoliko se ubaci povratni hod materijala, problem različitih vremena obrade pojedinih proizvoda i dr. Zbog tih nedostataka, kao bolje rješenje pokazala se kontrola procesa pomoću metoda.

7. ZAKLJUČAK

Računalna se simulacija pokazala kao primjeren alat za planiranje proizvodnje. Prednosti upotrebe ovakvog načina planiranja upravo su u tome što se u kratkom vremenu bez ometanja realnog sustava može predvidjeti određeno ponašanje istoga. S namjerom da se ovaj alat približi krajnjem korisniku i olakša njegova upotreba, u radu je kreiran univerzalni simulacijski model koji je lako primjenjiv na sve vrste i oblike proizvodnih sustava. Također, olakšan je unos ulaznih parametara u simulacijski model pomoću tri tablice u koje se unosi vrijeme operacije, količina proizvoda i radne stanice za obavljanje operacija (strojevi). Kao potvrda poboljšanja i pojednostavljenja, dobiveni su rezultati konačnog eksperimenta na realnom slučaju gdje je prikazano kako se na jednostavan način, bez izmjene postavki i promjene modela, može postići drugačiji raspored obavljanja aktivnosti koje obuhvaća proizvodni proces.

Izradi univerzalnog simulacijskog modela pristupilo se s idejom da korisnik unosi neke od ulaznih podataka te da se pomoću toga cijeli simulacijski eksperiment provodi samostalno bez izmjena samog simulacijskog modela. Ideja je proizašla iz zamisli o naručivanju proizvoda. Dakle, korisnik ima mogućnost naručiti, odnosno unositi ulazne podatke, kao što su količina određenih proizvoda, vrijeme trajanja pojedinih operacija i redoslijed radnih mjesta pojedine operacije, te se pomoću toga generira ulazni materijal i tok materijala. Također, zahtijevalo se da radne stanice same prepoznaju koji je proizvod i koja operacija u pitanju i prema tome automatski određuju vrijeme operacije i sljedeći korak toka materijala. Izrada simulacijskog modela započela je definiranjem vrste proizvoda i njihovim kreiranjem u programu, zatim su kreirane tablice koje služe kao određena vrsta sučelja za unos ulaznih informacija u simulacijski model i koje su jedino čemu krajnji korisnik treba pristupati prilikom postavljanja uvjeta za simulacijski eksperiment. Srž simulacijskog modela kodiranje je logičke metode prema čijoj se definiciji sve odvija. Ideja je bila zapisati dvije vrste metoda, jednu ulaznu i jednu izlaznu. Ulazna metoda izvršava se prilikom ulaska određenog proizvoda na određenu radnu stanicu, nakon čega pomoću identificirane vrste proizvoda pokreće potragu odgovarajućih podataka iz tablica ulaznih atributa. Iščitani atributi, informacije, usvajaju se kao određeni režimi radne stanice na kojoj se aktivira metoda. Na isti se način ponaša izlazna metoda koja se aktivira prilikom izlaska određenog proizvoda s određene radne stanice. Ručno se definira atribut svakom pojedinom proizvodu te se s njim pomoću jedne od metoda manipulira i na taj se način broji redni broj operacije na nekom od proizvoda. Prostorni je raspored strojeva zanemaren, a postavlja se nekolicina proizvodnih kapaciteta – strojeva koja će

zadovoljiti svrhu. Također, sva su vremena potrebna za postavljanje i namještanje strojeva zanemarena, kao i mogući nastanak greške bilo koje prirode. Ostale karakteristike i postavke strojeva jasno su definirane.

Nakon izrade simulacijskog modela i postavljanja svih traženih postavki, provedena je verifikacija i validacija simulacijskog modela. Kako je definirano u pravilima za izradu simulacijskih modela, svakom simulacijskom modelu treba na određeni način provjeriti ispravnost njegova rada i točnost dobivenih podataka. Verifikacijom je potvrđena ispravnost rada simulacijskog modela, odnosno da model ispravno funkcionira, dok je validacijom potvrđena točnost rezultata na način da se ručno izračunati rezultati podudaraju s rezultatima računalne simulacije.

Realni problem postavljen u radu riješen je pomoću izrađenog računalnog simulacijskog modela. Plan je bio postići određeni cilj u pogledu što kraćeg vremena proizvodnje za neku određenu narudžbu. Za početak, bilo je ponuđeno inicijalno rješenje koje je naknadno poboljšano do određene razine u nekoliko stupnjeva poboljšanja. Cilj je bio postići zadovoljavajuće rezultate te na osnovi toga izraditi svojevrsni plan i raspored aktivnosti za određeni proizvodni sustav.

U simulacijski su model pomoću tablica jednostavno uneseni svi zahtjevi te je u samo par minuta bio spreman za pokretanje prvog simulacijskog eksperimenta. Smjer u kojem su išla poboljšanja bio je prikazan vizualno pomoću grafa te je bio vrlo jasan. Bilo je potrebno rasteretiti najopterećeniju radnu stanicu i opterećenje preusmjeriti na neku od manje opterećenih radnih stanica. U iznimno kratkom vremenu od par minuta učinjena su dva koraka unapređenja toka materijala i postignut je zadovoljavajući hodogram. Istina je da se radi o metodi pokušaja i pogrešaka, ali je jasno vidljiv smjer pokušaja te se promjene izvršavaju brzo, tj. promjenom imena radne stanice na određenom mjestu u tablici. Konkretni dobiveni rezultati dokazuju da se izrađeni računalni simulacijski model može uspješno primijeniti na realni proizvodni proces. Vrijeme trajanja proizvodnog procesa pri inicijalnom (pretpostavljenom) toku bilo je 1 dan i 26 minuta. Uz unesene minimalne i očigledne izmjene, vrijeme trajanja proizvodnog procesa skratilo se za 5 sati i 28 minuta te je iznosilo 18 sati i 58 minuta. Još jednom sličnom izmjenom vrijeme proizvodnog procesa dodatno se skratilo za 1 sat i 23 minute te iznosilo 17 sati i 35 minuta. Dakle, ukupno trajanje proizvodnog procesa skratilo se za 6 sati i 51 minutu u samo dvije iteracije. Ako se želi ispitati postojanost još kraćih vremena, potrebno je primijeniti neku od tehnika optimiziranja

kojom se to može postići ili rješenje pronaći u samom tehnološkom procesu izrade dijelova, odnosno optimizaciji njihovih vremena operacija.

Treba naglasiti kako je u ovom radu ponuđeno i rješenje u vidu izrađenog kanban simulacijskog modela. S obzirom na to da je kanban zasnovan na karticama, pomoću njih može se naručiti repromaterijal s prethodih radnih stanica kada zaliha na određenoj radnoj stanici postigne minimalnu vrijednost. Ta vrsta rješenja nije se pokazala dovoljno zadovoljavajućom te je zbog toga prvobitna varijanta, tzv. univerzalni simulacijski model, usvojena kao bolje rješenje. Nedostatak kanban modela bio je u tome što se tok materijala nije mogao mijenjati, odnosno bio je fiksan i uspostavljen točnim vezama. Također, u kanban simulacijskom modelu nije se moglo jednostavno pristupiti izmjeni vremena u ovisnosti o proizvodu i broju operacije. Nakon izrade određene količine proizvoda, u međuskladištima bi ostajala određena količina repromaterijala. Na osnovi gore navedenih nedostataka, odustalo se od korištenja kanban metode u ovakvu svrhu.

Na osnovi primjene računalnih simulacija unutar programa *Tecnomatix Plant Simulation*, zaključuje se da ideja o univerzalnom simulacijskom modelu, poput ovog koji je dan u radu, uvelike olakšava planiranje proizvodnje. Pokazalo se da takvi univerzalni simulacijski modeli mogu uštedjeti vrijeme potrebno za izradu određenih količina različitog asortimana proizvoda i olakšati organizaciju proizvodnog procesa za krajnjeg korisnika. Simulacijsko modeliranje pokazalo se kao vrlo uspješno i korisno te se lako može primijeniti u planiranju proizvodnje, a uz primjenu univerzalnih modela, računalno simuliranje proizvodnje može postati još dostupnije i jednostavnije.

LITERATURA

[1] Ištoković, D. : „Računalna simulacija proizvodnih procesa - predavanje I“., s Interneta, https://moodle.srce.hr/2022-2023/pluginfile.php/7997143/mod_resource/content/1/RSPP%20-%20predavanje%201.pdf, 16. lipnja 2023.

[2] Stanković, R., Pašagić Škrinjar, J. : „Logistika i transportni modeli“, autorizirana predavanja, Fakutet prometnih znanosti, Zagreb, 2016., https://www.bib.irb.hr/932240

[3] Law, A. M., Kelton, W. D. : „Simulation modeling & analysis“, drugo izdanje, McGraw-Hill, SAD, 1991.

[4] Machado, T. S. : „Computer Simulation of a Photovoltaic System for Electricity Generation at Residencies and Condominiums“, s Interneta, https://www.researchgate.net/figure/Main-differences-between-continuous-and-discrete-event-simulation-Source-Adapted-from_fig31_322530271 , 5. lipnja 2023.

[5] Krenczyk, D., Pawlewski, P., Plinta, D. : „Symulacja procesów produkcyjnych“, Polskie Wydawnictwo Ekonomiczne, Warszava, Poljska, 2022.

[6] Banks, J., Carson, J. S., Nelson, B. L., Nicol, D. M. : „Discrete-Event System Simulation“, peto izdanje, Pearson, New Jersey, SAD, 2010.

POPIS SLIKA

Slika 1 Koncept simulacije[1] 2
Slika 2 Razina apstrakcije [1] 4
Slika 3 Razlika diskretnih događaja i kontinuiranih [4] 6
Slika 4 Simulacijski model izrađen unutar programa Tecnomatix Plant Simulation 12
Slika 5 Postavke za Part A 13
Slika 6 Popis proizvoda u Class Library 14
Slika 7 Naslovi tablica postavljenih u radni prostor 14
Slika 8 DataTable 15
Slika 9 TimeTable 16
Slika 10 StationTable 16
Slika 11 Testni simulacijski model za razvoj i pisanje metoda 17
Slika 12 DataTable - osnovni simulacijski model 17
Slika 13 TimeTable - osnovni simulacijski model 18
Slika 14 Prvi zapis metode za čitanje tablice TimeTable 18
Slika 15 Poboljšana verzija metode za čitanje tablice TimeTable 19
Slika 16 Konačni oblik metode za čitanje tablice TimeTable 20
Slika 17 Naslovi metoda postavljenih u radni prostor 21
Slika 18 Kod ulazne metode Enter 22
Slika 19 Primjer čitanja tablice TimeTable (Proizvod „C“ operacija 3) 23
Slika 20 Kod izlazne metode Exit 24
Slika 21 Primjer čitanja tablice StationTable (Proizvod „C“ operacija 4) 25
Slika 22 Raspored prostornih elemenata 26
Slika 23 Postavke izvora „So“ - Attributes 27
Slika 24 Postavke ulaza „So“ -Controls 27
Slika 25 Postavke stanice „St1“ -Controls 28
Slika 26 Postavke montažne stanice „AS“ - Attributes 28
Slika 27 Attribute Explorer 30
Slika 28 Postavke AttributeExplorer -Data 30
Slika 29 Postavke AttributeExplorer -Attributes 31
Slika 30 Usporedba StationTable i AttributeExplorer 31
Slika 31 SankeyDiagram - postavke 32
Slika 32 Hodogram "A", "B" i "AB" 33
Slika 33 Hodogram "C" i "D" 33
Slika 34 Display 34
Slika 35 Display postavke 35
Slika 36 TimeTable 36
Slika 37 DataTable za validaciju (proizvod A) 37
Slika 38 EventController - vrijeme izrade za proizvod A 38
Slika 39 Vrijeme simulacije za B 38
Slika 40 Vrijeme simulacije za C 39
Slika 41 Vrijeme simulacije za D 39
Slika 42 Proizvodi A i B postavljeni u međuskladišta 40
Slika 43 Vrijeme simulacije za AB 40
Slika 44 DataTable za traženu količinu 41

Slika 45 TimeTable za traženu narudžbu 41
Slika 46 StationTable za početni tok materijala 42
Slika 47 Graf okupiranosti proizvodnih kapaciteta 43
Slika 48 Graf iskorištenja proizvodnih kapaciteta 44
Slika 49 Zahtjevi za proizvodnju 1 44
Slika 50 Preraspoređeni tok materijala 45
Slika 51 Grafovi okupiranosti proizvodnih kapaciteta - preraspoređeni tok 45
Slika 52 Graf iskorištenja proizvodnih kapaciteta - preraspoređeni tok 46
Slika 53 Udio poslova po radnim stanicama 46
Slika 54 Prva izmjena toka materijala 47
Slika 55 Udio poslova po radnim stanicama nakon prve izmjene toka materijala 47
Slika 56 Druga izmjena toka materijala 48
Slika 57 Udio poslova po radnim stanicama nakon druge izmjene toka materijala 48
Slika 58 Redoslijed aktivnosti proizvodnog procesa 49
Slika 59 Simulacijski model - Kanban metoda 52
Slika 60 Manage Class Library - Kanban 53
Slika 61 Tablica "BOM" 54
Slika 62 Metoda MyExitControl 54
Slika 63 Postavke za kanban izvore - Cars (lijevo), Wheels (desno) 55
Slika 64 Insert New Part 55
Slika 65 KanbanBuffer - postavke 56
Slika 66 Postavke AssemblyStation 57
Slika 67 KanbanSngleProc 58
Slika 68 KanbanSingleProc - -Advanced 58
Slika 69 Kanban Sequence Table 59

POPIS TABLICA

Tablica 1 Proizvodni program 9
Tablica 2 Popis proizvodnih kapaciteta 10
Tablica 3 Vremena operacija 11

POPIS OZNAKA I KRATICA

Oznake:

t_{uk}	ukupno vrijeme operacije
t_i	vrijeme *i*-te operacije

Kratice:

A	proizvod A
B	proizvod B
C	proizvod C
D	proizvod D
AB	proizvod AB
So	ulaz materijala - izvor (engl. *Source*)
St1	Radna stanica 1 (tokarilica)
St2	Radna stanica 2 (glodalica)
St3	Radna stanica 3 (glodalica)
St4	Radna stanica 4 (glodalica)
MainMU	međuskladište glavnog proizvoda montažne stanice
AddMU	međuskladište sporednog proizvoda montažne stanice
S1	međuskladište (ulaz) radne stanice 1
S2	međuskladište (ulaz) radne stanice 2
S3	međuskladište (ulaz) radne stanice 3
S4	međuskladište (ulaz) radne stanice 4
AS	montažna stanica (engl. *Assembly Station*)
MU	mobilna jedinica (engl. *Mobile Unit*)

JIT	metoda upravo na vrijeme (engl. *Just-in-Time*)
BOM	tablica popisa materijala (engl. *Bill Of Materials*)

SAŽETAK

Računalna simulacija primjeren je alat za planiranje proizvodnog procesa jer omogućava predviđanje ponašanja realnog sustava bez ometanja istog. Kako bi upotreba računalnih simulacija u planiranju proizvodnog procesa bila još jednostavnija, u radu je kreiran univerzalni simulacijski model koji je lako primjenjiv na sve vrste i oblike proizvodnih sustava. Univerzalni simulacijski model baziran je na primanju tri vrste podataka (količina proizvoda, vremena operacija i tok proizvoda prema proizvodnim kapacitetima) koje krajnji korisnik unosi u zasebne tablice kako bi manipulirao računalnim simulacijskim modelom. Pokazalo se da univerzalni računalni simulacijski model štedi vrijeme i olakšava organizaciju proizvodnog procesa za krajnjeg korisnika. Nakon uvodnog dijela u kojemu se nudi teorijski pregled računalnih simulacija diskretnih događaja, slijedi objašnjenje idejnog proizvodnog procesa. Zatim slijedi detaljni postupak izrade univerzalnog simulacijskog modela, njegova validacija i verifikacija te unos konkretnih podataka iz idejnog procesa. Konačno, izvršena je analiza i interpretacija rezultata te je postavljeno konačno rješenje idejnog proizvodnog procesa. Na kraju je ponuđeno i alternativno rješenje promatranog problema.

Ključne riječi: planiranje proizvodnje, računalna simulacija, simulacija diskretnih događaja, simulacijski model, *Tecnomatix Plant Simulation*

SUMMARY

Computer simulation is a suitable tool for planning a production process since it allows to predict the behaviour of an actual system without obstructing the system itself. In the paper, in order to make the utilization of computer simulations even simpler when planning the production process, a universal simulation model was created that is easily applicable to systems of production of all types and forms. Universal simulation modeling is based on receiving three types of data (product quantity, duration of the process, and the movement of the product on the production line according to its production capacity) that the end user enters into separate tables in order to manipulate the simulation model. It has been shown that universal computer simulation modeling saves time and that

the organization of the production process is made easier for the end user. After the paper's introductory part in which a theoretical overview of discrete-event simulation is provided, an explanation of a conceptual production process is given. Afterwards, a detailed process of creating a universal simulation model follows together with its validation and verification, and the entry of specific data from the conceptual process. Finally, an analysis and interpretation of the results is conducted, and a final solution of the conceptual process is set. At the end of the paper, an alternative solution to an observed problem is offered.

Key words: production planning, computer simulation, discrete-event simulation, simulation model, Tecnomatix Plant Simulation