



УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА У
НОВОМ САДУ



Darko Avramović

**RAZVOJ MODELA INFORMACIONOG SISTEMA ZA
PODRŠKU UPRAVLJANJU GRAFIČKIM
PROCESIMA**

DOKTORSKA DISERTACIJA

Novi Sad, 2014.



УНИВЕРЗИТЕТ У НОВОМ САДУ • ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА
21000 НОВИ САД, Трг Доситеја Обрадовића 6

КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

Редни број, РБР :	
Идентификациони број, ИБР :	
Тип документације, ТД :	Монографска публикација
Тип записа, ТЗ :	Текстуални штампани материјал
Врста рада, ВР :	Докторска дисертација
Аутор, АУ :	Дарко Аврамовић
Ментор, МН :	Проф. Др Бранко Милосављевић
Наслов рада, НР :	Развој модела информационог система за подршку управљању графичким процесима
Језик публикације, ЈП :	Српски
Језик извода, ЈИ :	Српски
Земља публиковања, ЗП :	Република Србија
Уже географско подручје, УГП :	АП Војводина
Година, ГО :	2014.
Издавач, ИЗ :	Ауторски репринт
Место и адреса, МА :	Нови Сад, Трг Доситеја Обрадовића 6
Физички опис рада, ФО : (поглавља/страна/ цитата/табела/слика/графика/прилога)	10/125/133/6/71/4/1
Научна област, НО :	Графичко инжењерство и дизајн
Научна дисциплина, НД :	Графичко инжењерство и дизајн
Предметна одредница/Кључне речи, ПО :	Графички процеси, информациони системи, модел информационог система
УДК	
Чува се, ЧУ :	Нови Сад, Трг Доситеја Обрадовића 6
Важна напомена, ВН :	
Извод, ИЗ :	У дисертацији је приказан поступак израде модела информационог система за подршку управљању графичким процесима. Овај модел за резултат има успешно праћење параметара графичке производње. У раду су дате основне теоријске поставке графичке производње, графичких процеса, информационих система и њиховог моделовања. Направљен је апстрактни (high-level) и реални (low-level) модел информационог система. Као валидација модела извршена је симулација рада система у лабораторијским условима.
Датум прихватања теме, ДП :	
Датум одбране, ДО :	
Чланови комисије, КО :	Председник: Др Драгољуб Новаковић
	Члан: Др Немања Кашиковић
	Члан: Др Сандра Дедијер
	Члан: Др Александар Милосављевић
	Члан: Др Милан Видаковић
	Члан, ментор: Др Бранко Милосављевић
	Потпис ментора



KEY WORDS DOCUMENTATION

Accession number, ANO :	
Identification number, INO :	
Document type, DT :	Monograph
Type of record, TR :	Printed text document
Contents code, CC :	Ph. D. thesis
Author, AU :	MSc Darko Avramović
Mentor, MN :	Branko Milosavljević, Ph. D.
Title, TI :	Model of a Management Information System For Graphic Processes
Language of text, LT :	Serbian
Language of abstract, LA :	Serbian
Country of publication, CP :	Republic of Serbia
Locality of publication, LP :	AP Vojvodina
Publication year, PY :	2014.
Publisher, PB :	Author's reprint
Publication place, PP :	21000 Novi Sad, Trg Dositeja Obradovića 6
Physical description, PD : <small>(chapters/pages/ref./tables/pictures/graphs/appendixes)</small>	10/125/133/6/71/4/1
Scientific field, SF :	Graphic engineering and design
Scientific discipline, SD :	Graphic engineering and design
Subject/Key words, S/KW :	Graphic production processes, Information Systems, MIS, Information System model
UC	
Holding data, HD :	21000 Novi Sad, Trg Dositeja Obradovića 6
Note, N :	
Abstract, AB :	The dissertation presents a model of Management Information System for Graphic processes. This model results in a successful monitoring of graphic production parameters. Basic theoretical assumptions involving graphic production, printing processes, information systems, and their modeling were presented. An abstract (high-level) and real (low-level) model of the information system were also created. As a validation of the model, a simulations was carried out within the laboratory.
Accepted by the Scientific Board on, ASB :	
Defended on, DE :	
Defended Board, DB :	President: Dragoljub Novaković, Ph. D.
	Member: Nemanja Kašiković, Ph. D.
	Member: Sandra Dedijer, Ph. D.
	Member: Aleksandar Milosavljević, Ph. D.
	Member: Milan Vidaković, Ph. D.
	Member, Mentor: Branko Milosavljević, Ph. D.
	Mentor's sign

SADRŽAJ

1	UVOD	7
1.1	Cilj istraživanja.....	9
1.2	Plan istraživanja.....	11
1.3	Hipoteza istraživanja.....	11
2	UPRAVLJANJE GRAFIČKIM PROCESIMA	13
3	INFORMACIONI SISTEMI	16
3.1	Model informacionog sistema i njegove komponente.....	18
3.2	Resursi informacionih sistema.....	19
3.3	Principi i modeli razvoja informacionih sistema.....	20
3.4	Metodi modelovanja.....	27
3.5	Sistemi za upravljanje radnim tokovima.....	29
3.6	Jezici za modeliranje radnih tokova.....	33
3.6.1	Petri mreže.....	33
3.6.1.1	Metode analize.....	38
3.6.2	YAWL.....	40
4	POVEZANA ISTRAŽIVANJA	43
5	MODEL INFORMACIONOG SISTEMA	53
5.1	Definicija.....	53
5.2	Infrastruktura i arhitektura.....	54
5.3	Procesi.....	58
5.3.1	Serverski modul.....	58
5.3.2	Klijentski modul.....	72
5.4	Struktura podataka.....	77
5.5	Detalji implementacije.....	78
6	STUDIJA SLUČAJA	81
6.1	Rad sa sistemom.....	82
6.1.1	Serverski modul.....	82
6.1.2	Klijentski modul.....	95
6.2	Verifikacija modela i simulacija rada sistema.....	99
7	ZAKLJUČAK	107
8	PRILOZI	109
8.1	Prilog 1 – Listing tabela baza podataka.....	109
9	LITERATURA	114
10	BIOGRAFIJA	125

INDEKS SLIKA

Slika 2.1: KBA Complete Workflow Solution koncept	15
Slika 3.1: Model informacionog sistema	18
Slika 3.2: Šema modela vodopada	21
Slika 3.3: Inkrementalni model.....	24
Slika 3.4: Model prototipskog razvoja.....	25
Slika 3.5: Spiralni model.....	26
Slika 3.6: Abstraktno (a) naspram low-level modelovanja (b) i vizuelno (a) naspram tekstualnog modelovanja (b)	28
Slika 3.7: Referentni model SURT	31
Slika 3.8: Primer rada softvera Staffware, modul Graphical Workflow Definer (Eindhoven University of Technology (TU/e), 2013).....	32
Slika 3.9: FLOWer.....	32
Slika 3.10: YAWL primer.....	33
Slika 3.11: Primer proste Petri mreže	34
Slika 3.12: Stanja Petri mreže	36
Slika 3.13: Primer YAWL toka	40
Slika 3.14: YAWL simboli.....	42
Slika 4.1: Koeficijent ε za predložena rešenja.....	49
Slika 4.2: Broj istih rešenja.....	49
Slika 4.3: KBA Complete Workflow Solutions koncept.....	50
Slika 4.4: GWFM koncept.....	51
Slika 5.1: Infrastruktura GWFM sistema.....	55
Slika 5.2: Jedna od uloga Message komponente u komunikaciji komponenata .	56
Slika 5.3: Komponente klijentskog modula.....	56
Slika 5.4: Komponente serverskog modula	57
Slika 5.5: Upotreba serverskog modula	58
Slika 5.6: Dijagram aktivnosti prijave u serverski modul	62
Slika 5.7: Dijagram aktivnosti generisanja liste sistema	62
Slika 5.8: Dijagram aktivnosti unosa tehnoloških sistema	64
Slika 5.9: Dijagram aktivnosti izmene sistema.....	65
Slika 5.10: Dijagram aktivnosti pozicioniranja sistema - inicijalizacija	66
Slika 5.11: Dijagram aktivnosti pozicioniranja sistema - zapis podataka	66
Slika 5.12: Dijagram aktivnosti unosa definicije posla.....	68
Slika 5.13: Dijagram aktivnosti generisanja liste poslova.....	69
Slika 5.14: Dijagram aktivnosti kreiranja tokova segmenata	70

Slika 5.15: Upotreba klijentskog modula	72
Slika 5.16: Dijagram aktivnosti podešavanja klijentskog modula - inicijalizacija	74
Slika 5.17: Dijagram aktivnosti podešavanja klijentskog modula - čuvanje podataka.....	74
Slika 5.18: Dijagram aktivnosti procesa prijave korisnika u klijentski modul	75
Slika 5.19: Dijagram aktivnosti kreiranja liste zaduženja	76
Slika 5.20: Dijagram aktivnosti kreiranja liste zaduženja - klik događaj.....	76
Slika 5.21: Dijagram baze podataka	78
Slika 5.22: Klasni dijagram klijentskog modula.....	79
Slika 6.1: Infrastruktura test okruženja	82
Slika 6.2: O GWFM sistemu	82
Slika 6.3: Prijava korisnika u serverski deo sistema.....	83
Slika 6.4: Komandna tabla.....	83
Slika 6.5: Pregled grafičkih sistema	84
Slika 6.6: Izmena jednog od sistema	85
Slika 6.7: Definisane prostorije	85
Slika 6.8: Određivanje pozicije sistema	86
Slika 6.9: Lista registrovanih korisnika	87
Slika 6.10: Registracija novog korisnika sistema.....	87
Slika 6.11: Unos nove definicije posla	89
Slika 6.12: Definisane grupe poslova.....	90
Slika 6.13: Pregled poslova	91
Slika 6.14: Detaljan prikaz posla	91
Slika 6.15: Praćenje statusa poslova po segmentima i fazama procesa	92
Slika 6.16: Primer statusa (u toku)	93
Slika 6.17: Grananje celog proizvoda na segmente koji se posebno obrađuju	93
Slika 6.18: Pregled klijenata	94
Slika 6.19: Interfejs za unos klijenata	94
Slika 6.20: Info o GWFM klijentu	95
Slika 6.21: Prijava u sistem.....	95
Slika 6.22: Podešavanje klijenta.....	96
Slika 6.23: Prikaz 'Moji podaci'	96
Slika 6.24: Lista zaduženja	97
Slika 6.25: Pregled zaduženja	98
Slika 6.26: Praćenje tokova segmenata	98

Slika 6.27: Aktivno zaduženje	99
Slika 6.28: Radni tokovi prema definicijama posla	101
Slika 6.29: Markeri stanja mreža sa slike 6.28.....	102
Slika 6.30: Prikaz podataka o proizvodnim procesima	103
Slika 6.31: Statusi sistema u toku rada sistema.....	104
Slika 6.32: Odnos predviđenih i realno ostvarenih vremena za pojedinačne proizvode	104
Slika 6.33: Odstupanja vremena realizacije pojedinačnih poslova	105
Slika 6.34: Prikaz odstupanja pri opterećenju sistema.....	105
Slika 6.35: Ostvarena vremena po grupama proizvoda	106

INDEKS TABELA

Tabela 4.1: Poređenje atributa sličnih rešenja (1. deo)	46
Tabela 4.2: Poređenje atributa sličnih rešenja (2. deo)	47
Tabela 4.3: Poređenje atributa sličnih rešenja (3. deo)	48
Tabela 5.1: Funkcije GWFM sistema dostupne korisnicima	54
Tabela 6.1: Podaci iz grupe 'Osnovni podaci vezani za posao'	88
Tabela 6.2: Parametri mreža sa slike 6.28.....	100

1 UVOD

Radni tok (*engl. workflow*) generalizovano je zasnovan na idejama, metodama, tehnikama i softveru koji se koriste kao podrška strukturiranim proizvodnim i poslovnim objektima. Zadatak upravljanja radnim tokovima je postizanje usmerenog i lakog za održavanje procesa proizvodnje.

Sistemi za upravljanje radnim tokovima (*engl. Workflow Management Systems*) su sistemi koji su razvijeni da podrže poslovne i proizvodne procese. (WfMC, 2012) Proizvodni proces se sastoji iz niza koraka (aktivnosti) koje se mogu izvršiti automatski, manuelno ili korišćenjem kombinacije prethodna dva slučaja. Sistem za upravljanje radnim tokovima ne samo da obezbeđuje izvršavanje aktivnosti nego vodi računa i o distribuciji zadataka i informacija zaposlenima. Pored toga omogućava praćenje događaja unutar procesa proizvodnje i aktivnosti zaposlenih u toku procesa, kao i kreiranje statističkih podataka koji mogu dovesti do određenih, za proces proizvodnje, nekada bitnih zaključaka. Izvršavanje zadataka kojima koordinira sistem za upravljanje radnim tokovima bazirano je na modelu procesa odnosno definiciji procesa.

Proces radnog toka je specifičan tip proizvodnog procesa sa sledećim osobinama:

- svaki korak procesa radnog toka može biti dodeljen tačno jednom slučaju.
- specifična situacija pokreće inicijalizaciju procesa.

Ove ključne akcije radnog toka podrazumevaju obradu velike količine informacija i shodno tome mogu biti okarakterisane kao zahtevne operacije u smislu količine informacija.

Za razvoj jednog radnog toka potrebno je kreirati tokove procesa. Ti tokovi procesa predstavljaju definiciju datog radnog toka. Jedna takva definicija može biti podeljena u dva dela: procesni model i klasifikacija resursa. Procesni model sadrži razne korake i stanja procesa dok klasifikacija resursa indicira koje vrste radnih operacija određeni zaposleni učesnik proizvodnje može da izvrši.

Jedan od načina odnosno tehnika modelovanja proizvodnih procesa su mreže tokova procesa (*engl. workflow nets*). Pomenuti oblik mreže predstavlja poseban oblik Petri mreže (van der Aalst i van Hee, 2000; van der Aalst, 2000; van der Aalst, 1997).

WFMC Reference model poznat i kao The Workflow Management Coalition model (van der Aalst i van Hee, 2000; van der Aalst, 2000; Lawrence, 1997; WFMC, 1996) razvio je referentni model upravljanja radnim tokovima da bi kreirao generalnu okosnicu razvoja sistema za upravljanje radnim tokovima. Referentni model ilustruje glavne komponente i interfejse jednog sistema za upravljanje radnim tokovima. Delovi referentnog modela pomenutog sistema su: Workflow engine,

Workflow klijent aplikacije, Administracioni alati i alati za održavanje sistema, definicije procesa i ostale aplikacije. Sve komponente su povezane interfejsima (van der Aalst i van Hee, 2000).

Veliki broj sistema za upravljanje radnim tokovima je dostupan na tržištu danas pored postojećih open source proizvoda i akademskih prototipa. Neki od tih sistema su: Staffware, FLOWer, akademski sistem YAWL i na kraju ADEPT (van der Aalst i van Hee, 2000, Huang i ostali, 2006, Alonso i ostali, 1996, Reichert i Dadam, 1998). Staffware je jedan od najupotrebljenijih sistema u svetu. Za primer, pretpostavlja se da je Staffware 1998. godine držao 25% svetskog tržišta (van der Aalst i van Hee, 2000). Staffware se sastoji iz nekoliko komponentata koje se generalno koriste za definisanje modela procesa, korisnika i njihovih uloga u sistemu, kao i izvršavanje instanci procesnih modela.

Sistemi za upravljanje radnim tokovima predstavljaju jedan oblik informacionih sistema. Svrha informacionih sistema je dostava pravih informacija pravim osobama u pravo vreme u pravim količinama i formatu – ukratko informacioni sistemi su sistemi namenjeni da obezbeđuju korisne informacije. Arhitektura na kojoj je baziran jedan informacioni sistem naziva se arhitektura informacione tehnologije ili IT arhitektura (*engl. information technology (IT) architecture*) i predstavlja plan ili mapu visokog nivoa informacionih segmenata u jednom organizovanom obliku. IT arhitektura objedinjuje kompletne potrebe poslovne organizacije za informacijama, IT infrastrukturu i sve aplikacije i može se reći da je analogna arhitekturi jedne kuće. IT arhitektura prikazuje kako svi aspekti jedne informacione tehnologije unutar organizacije funkcionišu zajedno. Infrastruktura informacionih tehnologija ili IT infrastruktura (*engl. information technology (IT) infrastructure*) se sastoji od fizičkih komponentata, IT komponentata, IT servisa i IT osoblja koji održavaju kompletan sistem organizacije. IT komponente predstavljaju kompjuterski hardver, softver i opremu potrebnu za ostvarivanje komunikacija odn. komunikacione tehnologije koje predstavljaju temelj svakog informacionog sistema. IT osoblje koristi IT komponente za proizvodnju IT servisa što uključuje upravljanje informacijama i podacima, razvoj sistema i brigu oko sigurnosti sistema (Rainer i Cegielski, 2011).

Uloga jednog informacionog sistema je da obezbedi informacije od značaja upravljačkom delu koje će im omogućiti da donose prave odluke koje će održati kompletnu organizaciju pod kontrolom.

Svi procesi proizvodnje zahtevaju određeno vreme za izvršenje. Problem koji se postavlja pred upravljački deo proizvodnog sistema je kako iskoristiti vreme i ispuniti ga tako da bude maksimalno iskorišćeno i optimalno raspoređeno. U grafičkoj proizvodnji postoje tokovi i procesi koje je potrebno pratiti i koordinirati. Da bi jedan poslovni sistem ostvario svoj cilj mora biti organizovan u organizacione celine. Te celine su povezane funkcionalnim vezama. Celokupno preduzeće sa svojim organizacionim celinama i vezama između njih predstavlja jedan poslovni sistem. Pored upravljačkog dela organizacije poslovni sistem poseduje i deo odgovoran za samu proizvodnju koji se naziva proizvodni sistem.

U okviru svakog proizvodnog sistema postoji redosled ili plan po kom se određeni procesi izvršavaju. Svaki proces zahteva određeno vreme za izvršavanje. Maksimalno vreme izvršenja svake procesne operacije se može izračunati upotrebom poznatih metoda uz pomoć kreiranih normativa. Na osnovu vremena izvršenja određenih procesa proizvodnje mogu se izračunati i troškovi datog procesa proizvodnje na osnovu normativa troškova rada mašina i troškova materijala. U okviru grafičke proizvodnje postoje kalkulatívni procesi koji se odvijaju pre početka samog proizvodnog procesa određenog proizvoda. Na osnovu pomenutih kalkulatívni procesa se formiraju okvirni ili maksimalni troškovi proizvodnje određenog proizvoda. Određivanje troškova koristi se da se odredi troškovni uticaj određenog procesa proizvodnje na ukupno poslovanje proizvodnog sistema. Proračun troškova se obavlja prema zahtevima naručioca ili klijenta (kupca) sa posebnim osvrtom na metodologiju izrade samog proizvoda u okviru odgovarajućeg proizvodnog procesa.

Komunikacija unutar proizvodnog sistema, pored konvencionalnih audio/vizuelnih tehnika, obavlja se korišćenjem za proizvodni sistem standardizovanih dokumenata - radnih naloga. Faznim radnim nalogom pojedine procesne celine se upoznaju sa proizvodom koji je trenutno u fazi izrade. Centralnim radnim nalogom kompletan proizvodni sistem se upoznaje sa opštim informacijama o proizvodu koji je trenutno u proizvodnom procesu. Problem ovakvog vida komunikacije i razmene informacija mogle bi biti netačne i nedovoljno precizne ili dvosmislene informacije koje mogu dovesti do zastoja proizvodnje, gubitka kapitala, gubitka vremena i neminovnog odlaganja drugih proizvodnih procesa, gubljenja klijenata ili kupaca proizvoda.

Unapređenje tokova informacija, bolja organizacija raspodele poslova osoblja proizvodnog sistema kao i bolja komunikacija i informisanost svih delova proizvodnog sistema značajno bi unapredili kompletan proizvodni sistem u celini. Nove tehnologije omogućavaju lakšu razmenu informacija i čine da te informacije budu preciznije, detaljnije i bolje predstavljene. Postoje problemi u klasičnim oblicima komunikacije unutar proizvodnog sistema koji bi se mogli prevazići novim modelima komunikacije i razmene informacija. Razvijeni model informacionog sistema unapredio bi parametre grafičkog proizvodnog procesa i izmenio koncept komunikacije i razmene informacije unutar jednog grafičkog proizvodnog sistema. Takođe kontrola procesa, osoblja i stanja sistema bila bi podignuta na viši nivo. Princip dodele poslova i planiranje zauzetosti postrojenja mogao bi biti pojednostavljen aktivnim uključivanjem pojedinaca proizvodnog procesa u proces odlučivanja i planiranja raspodele opterećenja proizvodnih postrojenja.

1.1 Cilj istraživanja

Cilj istraživanja je razvoj modela informacionog sistema za podršku upravljanju grafičkim procesima kojim će se unaprediti tokovi informacija između procesa

proizvodnje unutar jednog grafičkog proizvodnog sistema kroz razvoj konkretnog rešenja informacionog sistema korišćenjem najsavremenijih tehnologija bežičnih i mobilnih komunikacija.

Model informacionog sistema koji se predlaže objedinjuje nove tehnologije u jednu celinu koja ima za cilj da unapredi komunikaciju i raspodelu informacija između delova jednog grafičkog proizvodnog sistema. Za razliku od postojećih modela koji funkcionišu na konvencionalnim sistemima komunikacije, model informacionog sistema koji se predlaže ovim istraživanjem objedinjuje savremene tehnologije komunikacija sa ciljem olakšavanja komunikacije i boljeg informisanja pojedinaca učesnika procesa grafičke proizvodnje.

Pored unapređenja sistema komunikacije na jedan inovativan način, predloženi model informacionog sistema ima za cilj i smanjenje vremena pojedinih pripremnih operacija skladištenjem postojećih informacija prikupljenih iz proizvodnog procesa i koristeći ih za bolju organizaciju i smanjenje vremena proračuna trajanja operacija. Pojedinci koji učestvuju u proizvodnom procesu su u konstantnom kontaktu sa centralnim sistemom koji prati i beleži sve aktivnosti svakog pojedinca. Podaci zabeleženi i sačuvani u bazi podataka kasnije se mogu iskoristiti za ocenu uspešnosti svakog pojedinca učesnika proizvodnog procesa. Pored toga zabeleženi podaci koji se tiču vremena izvršenja pojedinih operacija proizvodnog procesa mogu biti iskorišćeni za predviđanje trajanja proizvodnih procesa iste kategorije i time se smanjuje vreme kalkulativnih procesa proizvodnje.

Model informacionog sistema biće potpuno prilagodljiv većini proizvodnih sistema grafičke proizvodnje. Proizvodni sistem prilagodiće informativni sistem, baziran na predloženom modelu, svojim potrebama i uslovima koji vladaju unutar proizvodnog sistema. Poslovni sistem može kreirati grupe proizvoda prema sopstvenim potrebama. Svakoj grupi proizvoda pripadaće određeni proizvodi proizvodnog procesa. Pripadnost proizvoda nekoj od grupa zavisice od kategorije proizvoda, dimenzije proizvoda, vrste obrade proizvoda i slično. Osobine koje su navedene određivaće pojedinci upravljačkog dela sistema prema potrebama organizacije. Ne postoje unapred određena pravila određivanja datih osobina. Za svaki grafički proizvod koji spada u neku od kreiranih grupa proizvoda beleži se vreme trajanja izrade po operacijama. Kasnije se takvi podaci mogu koristiti za predikciju vremena završetka sličnih po grupi ili istih proizvoda i samim tim lakšu organizaciju proizvodnog procesa.

Model informacionog sistema će omogućiti i zamenu tradicionalnih vidova komunikacije unutar proizvodnog procesa. Tradicionalno, informacije su se prenosile standardizovanim dokumentima (radnim nalozima) koji su sadržali ograničenu količinu informacija. Te informacije nekada su mogle da budu nepotpune, nedovoljno razumljive ili netačne što može da izazove niz grešaka unutar proizvodnog procesa. Model informacionog sistema koji se predlaže je novi model komunikacije i prenosa informacija korišćenjem savremenih tehnologija. Takav vid komunikacije i raspodele informacija obezbeđuje trenutno obaveštavanje svih

učesnika grafičkog proizvodnog procesa o statusu proizvodnje, eliminiše moguće greške koje nose klasični vidovi raspodele informacija. U slučaju grešaka, iste se otklanjaju centralno i o tome su istog momenta obavješteni svi učesnici proizvodnog grafičkog procesa koji sada poseduju ispravne informacije o tekućem delu procesa. Još jedno poboljšanje koje predviđa predloženi model informacionog sistema je raspoloživost svih informacija o svim započetim ili procesima proizvodnje koji su u pripremi. Svaki učesnik proizvodnog procesa će moći da se upozna sa parametrima proizvodnog procesa pre početka istog i pripremi se na odgovarajući način za realizaciju dodeljenog zadatka.

Svaki pojedinac koji učestvuje u proizvodnom procesu može imati dodeljen zadatak od strane pretpostavljenih i taj ili ti zadaci se nalaze u spisku njegovih obaveza. Ukoliko operater/zaposleni ne poseduje niti jedan dodeljen zadatak i neraspoređen je u nekom od procesa proizvodnje, zadatak mu može biti dodeljen od strane pretpostavljenog. Sistem bi u potpunosti funkcionisao na bazi aktuelnih bežičnih i mobilnih tehnologija zastupljenih danas. Model informacionog sistema predložen ovim istraživanjem uključuje upotrebu najsavremenijih tehnologija uređaja i bežičnih komunikacija. Takav oblik razmene informacija i podataka daleko je precizniji i savremeniji od konvencionalnih metoda.

Svaki zadatak (posao) ima nekoliko svojstava:

- *aktivnost* – određuje da li je posao aktivan u smislu da li je dostupan za početak proizvodnog procesa
- *da li je dodeljen?* – da li je posao dodeljen nekom od učesnika proizvodnog procesa ili tek treba da bude dodeljen.

Model nudi visok stepen prilagođavanja svojih osobina potrebama proizvodnog grafičkog sistema na koji se primenjuje.

1.2 Plan istraživanja

Planom rada obuhvaćeno je definisanje svrhe i zadatka istraživanja, pregled literature ranijih istraživanja na temu, definisanje modela informacionog sistema, simuliranje tokova informacija unutar modela informacionog sistema, određivanje potrebnih tehnologija i tehnika za realizaciju rešenja modela informacionog sistema, definisanja mesta izvođenja testova i eksperimenata, definisanje metoda korišćenih u izradi i prezentovanje rezultata rada.

1.3 Hipoteza istraživanja

Moguće je unaprediti proces grafičke proizvodnje razvojem i primenom modela informacionog sistema za podršku upravljanju grafičkim procesima uz primenu savremenih programskih i računarskih sistema.

Za realizaciju postavljene hipoteze moguće je razviti model za laboratorijske uslove koji imaju i proizvodnu grafičku opremu. Takav razvijeni model je moguće aplicirati u konkretnoj grafičkoj proizvodnji.

2 UPRAVLJANJE GRAFIČKIM PROCESIMA

Proizvodnja predstavlja osnovno područje ljudske delatnosti. Neophodna je za zadovoljenje potreba učesnika procesa rada, radnih sistema i stabilnog razvoja društva u skladu sa utvrđenim ciljevima. Predstavlja usmerenu aktivnost koja ima za cilj dobijanje proizvoda korisnih za društvo.

Proizvodni sistemi predstavljaju sisteme za ostvarenje proizvodnje i dobijanje proizvoda neophodnih za zadovoljavanje potreba društva (Zelenović, 2003).

Grafički procesi predstavljaju skup procesa koji se odvijaju unutar jednog grafičkog proizvodnog sistema. U grafičkim procesima učestvuju grafički sistemi koji se mogu definisati kao osnovni produkcionni element grafičkog procesa. Grafički sistemi predstavljaju integrisanu celinu ljudskog rada i rada tehničko tehnološkog sistema čiji je rezultat realizacija grafičkog proizvoda.

Pojam grafički (*grč. graphikos*) pripada veštini pisanja, crtanja, slikanja. Pojam proces (*lat. processus*) predstavlja put kojim se odvija nastanak nečega. Prema tome grafički procesi predstavljaju tok ili put kojim se dobijaju ili nastaju grafički proizvodi.

Razlikuju se dve vrste vođenja procesa:

- upravljanje
- regulacija

Upravljanje predstavlja način vođenja procesa. Može biti ručno, automatizovano ili automatsko. Regulacija predstavlja način vođenja procesa pri kome se odabrana procesna veličina održava u određenoj vrednosti ili u određenom rasponu (Novaković, 2002).

Pojam upravljanja vezan je za još mnogo definicija. Još jedna od definicija upravljanja mogla bi biti da je to skup akcija kojim se vrši delovanje na sistem u toku vremena čime se želi postići ciljevi pomenutog sistema. Upravljanje sistemom se odvija promenom vrednosti parametara posmatranog sistema. Takođe može se reći da je upravljanje prevođenje sistema iz jednog stanja u drugo (Grladinović, 1999).

Za ostvarenje funkcije upravljanja neophodno je prisustvo nekog oblika informacionog sistema. Danas je više nego očigledno izuzetno intenzivno prisustvo računarskih tehnologija u procesima proizvodnje. Situacija nije drugačija ni kada su u pitanju procesi grafičke proizvodnje. Za upravljanje grafičkim procesima koriste se složeni oblici informacionih sistema. Takvi oblici sistema su prisutni na svim nivoima poslovnog i proizvodnog procesa. Kroz faze grafičke proizvodnje evidentna je prisutnost manjih ili većih oblika informacionih sistema u svakom koraku. Počevši od administracije, preko pripreme za štampu i štampe do završne grafičke obrade upotrebljavaju se različiti oblici informacionih sistema različitih stepena složenosti i namena.

U pripremi grafičke proizvodnje koriste se office informacioni sistemi koji su neretko opremljeni bazama podataka. Često koriste mrežne resurse za potrebe komunikacije između sebe ili sa drugim delovima poslovnog sistema. Takođe, postoje oblici informacionih sistema koji se bave praćenjem i upravljanjem grafičkom proizvodnjom. Takvi sistemi su u današnje vreme izuzetno kompleksni i predstavljaju sveobuhvatna rešenja koja se bave upravljanjem informacijama, materijalima i sirovinama, ljudima i drugim delovima sistema. Arhitekture ovih sistema se pak razlikuju od slučaja do slučaja. Neki su web bazirani dok su neki vezani za desktop operativne sisteme. Neki su otvorenog pristupa (u smislu da trećim licima dopuštaju pristup svojim resursima) dok su neki zatvorenog tipa (upotrebljavaju se samo unutar poslovnog ili proizvodnog sistema). Neki se bave direktnim upravljanjem sistema proizvodnog procesa dok neki služe za upravljanje informacijama. Pravilna i pravovremena distribucija elektronskih dokumenata je od ključnog značaja za funkcionisanje kompletnog proizvodnog sistema. Postoje i sistemi koji su zaduženi isključivo za tokove informacija i materijala. Namena informacionog sistema određuje način njegove izrade i upotrebe. U zavisnosti od potreba za informacionim sistemom kreira se arhitektura prilagođena uslovima funkcionisanja informacionog sistema.

U grafičkim procesima poseban izazov predstavlja upravljanje tokovima materijala. Za ovu namenu postoje posebni oblici informacionih sistema koji služe za evidenciju skladištenja i upravljanje tokovima materijala koji se koriste u procesu proizvodnje. Ovi informacioni sistemi upravljaju i transportom materijala kroz proizvodni sistem potpuno automatski.

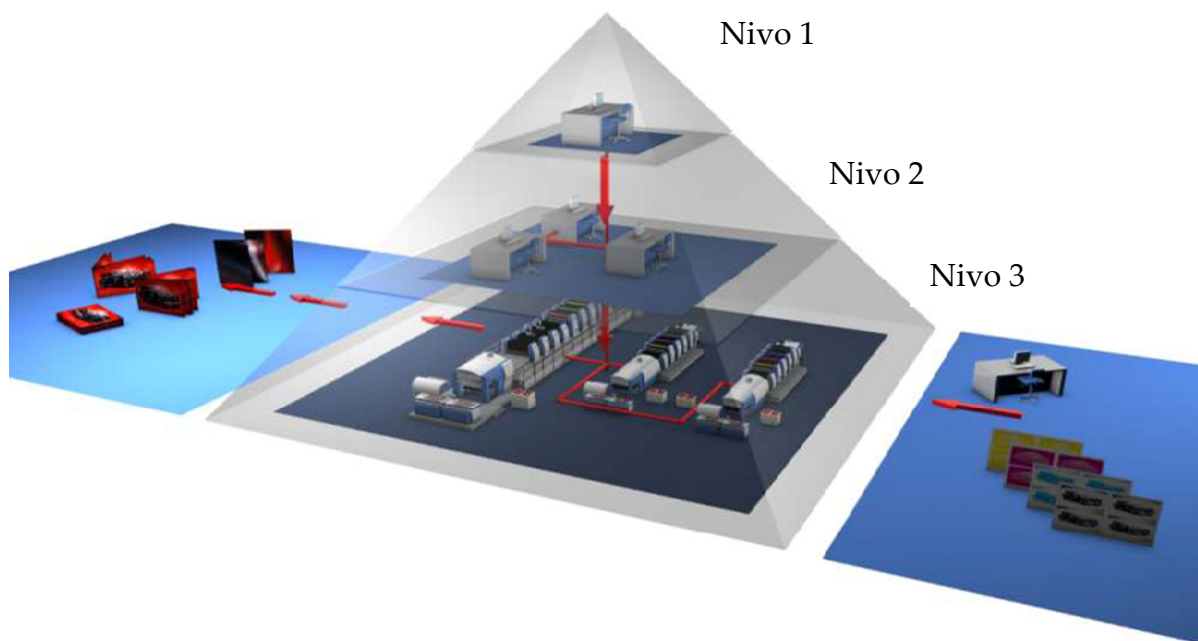
Veliki deo grafičkih sistema danas u sebi sadrži određeni oblik softvera koji upravlja pomenutim sistemom. Takvi oblici softvera mogu sadržati mogućnost integracije u druge sveobuhvatnije sisteme ali to nije obavezno. Najčešće sa ostatkom proizvodnog sistema komuniciraju upotrebom nekog od standardnih protokola mrežne komunikacije. Ovakvi oblici softvera su posebno dizajnirani za sistem na kome se primenjuju i kojim upravljaju. U većini slučajeva se ne mogu primeniti na drugi tip grafičkog sistema.

Ozbiljan pristup upravljanju takođe je potreban i u finalnoj fazi - isporuci proizvoda. Kompletne proizvode je potrebno evidentirati i uskladištiti prema rasporedu isporuke i različitim drugim faktorima.

U grafičkim proizvodnim sistemima upravljanje je prisutno i u najsitnijim segmentima proizvodnje i pravilno funkcionisanje sistema koji vrše upravljanje je od esencijalnog značaja za pravilno funkcionisanje proizvodnog sistema. Upravljanje se vrši od nivoa sastavnih delova grafičkih sistema do nivoa kompletnog proizvodnog sistema.

U ovoj oblasti dat je uski prikaz osnovnih pojmova proizvodnje, grafičkih sistema i grafičkih procesa i upravljanja istim. Dati su neki primeri upravljanja i oblici informacionih sistema koji se koriste. Pored nabrojanih oblika postoji još oblika informacionih sistema koji se koriste u grafičkim proizvodnim sistemima. Jedan od

primera upravljanja grafičkom proizvodnjom, koje vredi pomenuti, jeste sistem KBA Complete Workflow Solution (KBA, 2013). Ovaj sistem namenjen je upravljanju radnim tokovima u sklopu procesa štampe. Komunicira sa tehnološkim sistemima putem JDF i JMF standardizovanih formata dokumenata. Koncept rada pomenutog sistema prikazan je na slici 2.1.



Slika 2.1: KBA Complete Workflow Solution koncept

Nivo 1 predstavlja MIS (Management Information System). Na nivou 2 nalazi se *gateway* solucija (serverski oblik). Nivo 3 predstavlja proizvodni sistem. Više o ovom sistemu biće rečeno u 4. poglavlju.

3 INFORMACIONI SISTEMI

Organizacija posla kako unutar tako i između poslovnih i proizvodnih sistema postaje sve zahtevnija i komplikovanija. Koordinacija podacima i komunikacija postaje sve teža zbog izražene kompleksnosti procesa proizvodnje. Funkcionisanje organizacije i upravljanje organizacijom i njenim procesima je postalo nezamislivo bez odgovarajućih podataka, informacija i znanja. Tako oni postaju jednako značajan resurs organizacije kakvi su sirovine, energija, radna snaga, finansije i dr. Podaci se registruju, pripremaju, unose, organizuju, čuvaju u bazama podataka informacionog sistema. Informacioni sistem organizacije iste obrađuje i obezbeđuje informacije i znanje za realizaciju poslovnih operacija i rešavanje poslovnih problema organizacije. Zbog pomenutih razloga razvijeni su kompjuterski sistemi zaduženi za upravljanje informacijama i procesima i njihovo koordiniranje. Postoji veliki broj kategorija informacionih sistema. Neki od njih poseduju veoma malo informacija o kompletnom procesu proizvodnje i namena im je samo da pomognu ljudima učesnicima proizvodnje. Sa druge strane, postoje sistemi koji poseduju kompletne informacije o procesu proizvodnje i sposobni su da potpuno zamene čoveka u procesu proizvodnje (van der Aalst i van Hee, 2000; Alter, 2008; Kirikova i ostali, 2009; Balaban i ostali, 2010; Guetat i Dakhli, 2012; Pearlson i Saunders, 2013). Informacioni sistemi se mogu definisati i kao disciplina koja se bavi dizajnom, konstrukcijom i upotrebom informacionih tehnologija (March i Smith, 1995; Dahlbom, 1996; Orlikowski i Iacono, 2001; Benbasat i Zmud, 2003; Hevner et al., 2004, Hart i Gregor, 2005; Hardcastle, 2008). Može se definisati i kao sistem međusobno povezanih komponenata koje prikupljaju, obrađuju i čuvaju podatke na osnovu kojih proizvode povratne informacije (Ramiller i Pentland, 2009; Stair i Reynolds, 2011). Informacioni sistemi mogu biti kompjuterizovani ili manuelni (Lederman i Johnston, 2007) za koje postoje brojni primeri (Benson i Standing, 2002; Boddy et al., 2005; Stair i Reynolds, 2003).

Prema Brooks-u (1987 i 1995), informacioni sistemi imaju 4 osnovne osobine:

- kompleksnost
- usaglašenost
- promenljivost
- nevidljivost

Primarni cilj jednog informativnog sistema jeste da na jednostavan i pre svega ekonomičan način transformiše podatke u iskoristive informacije ili znanje. U nastavku će biti definisani ovi osnovni pojmovi: Podaci elementarno opisuju stvari, događaje, aktivnosti i transakcije koji su zabeleženi, klasifikovani i uskladišteni, ali nisu organizovani da prenesu neko konkretno značenje. Oni mogu biti različitih oblika: numerički, alfanumerički, zvuci, slike i dr. i skladište se u bazu podataka, organizuju na način da se lako pronalaze. Informacije su organizovani podaci na

način da imaju značenje i vrednost za primaoca. Najčešće se podaci obrađuju aplikativnim programima, kako bi se proizvela veća njihova korisnost od one koja se postiže u slučaju direktnog i jednostavnog pozivanja iz baze podataka. Ovakvi procesi se nazivaju i *informacioni procesi* (Georgakopoulos i ostali, 1995). Znanje čine podaci ili informacije koji se organizuju i obrađuju da prenesu razumevanje, iskustvo, akumulirano učenje i stručnost u primeni na određeni aktuelni problem ili aktivnost. Podaci se posebno obrađuju na način da reflektuju iskustvo i ekspertizu, pruže primaocu organizaciono znanje visoke potencijalne vrednosti. Tri osnovna razloga za primenu informacionih tehnologija u poslovanju, koji su u neposrednoj vezi sa tri vitalne uloge informacionog sistema, koje on može imati za poslovnu organizaciju su: podrška poslovnih procesa i aktivnosti organizacije, podrška donošenju odluka od strane zaposlenih i menadžera (rukovodilaca) i podrška strategiji u realizaciji konkurentskih prednosti. (Balaban i ostali, 2010; Hardcastle, 2008, Rainer i Cegielski, 2011; Polišćuk, 2007).

U nastavku će biti obrađene samo pojedine vrste informacionih sistema potrebne za upravljanje radnim tokovima.

- *Kancelarijski informacioni sistemi* su sistemi namenjeni za pomoć kadrovima koji se bave upravljačkim operacijama sa osnovnim nivoom obrade podataka kao što su pisanje, crtanje, kalkulacije i slično. Takvi informacioni sistemi podrazumevaju u svoj sklopu procesore ili formatere teksta, softver za upravljanje tabelama, proste sisteme za upravljanje bazama podataka i elektronsku poštu. Ovakvi sistemi ne poseduju nikakve informacije o procesu proizvodnje i namenjeni su za pomoć ljudima učesnicima procesa proizvodnje (van der Aalst i van Hee, 2000; Wamelink i ostali, 2002; Polišćuk, 2007).
- *Sistemi za procesiranje transakcija* (*engl. transaction-processing systems*) su sistemi još nazvani i registracioni sistemi (*engl. registrational systems*). Namena im je praćenje transakcija i promena unutar procesa. Ovakvi tipovi sistema koji su specijalizovani za među-organizacijsku komunikaciju nazivaju se i među-organizacioni sistemi (*engl. interorganizational information systems*). Takvi sistemi obično komuniciraju upotrebom standardizovanih načina elektronske komunikacije i razmene podataka kao što je za primer XML. Srce ovih sistema je obično sistem za upravljanje bazama podataka (van der Aalst i van Hee, 2000, van Belle i ostali, 2001; Polišćuk, 2007; Rainer i Cegielski, 2011).
- *Sistemi za upravljanje znanjem* (*engl. knowledge-management systems*) su sistemi čiji je zadatak pribavljanje i distribucija znanja na mesta koja to zahtevaju. Znanje koje oni prenose je eksplicitno znanje koje se može iskazati u elektronskoj/digitalnoj formi. Najjednostavniji primer jednog ovakvog sistema je *endžin* za pretragu u kombinaciji sa sistemom za upravljanje podataka. Na taj način pojedinac može pronaći podatak koji je lično zabeležio ili je zabeležen od strane drugog pojedinca. Pretraga je

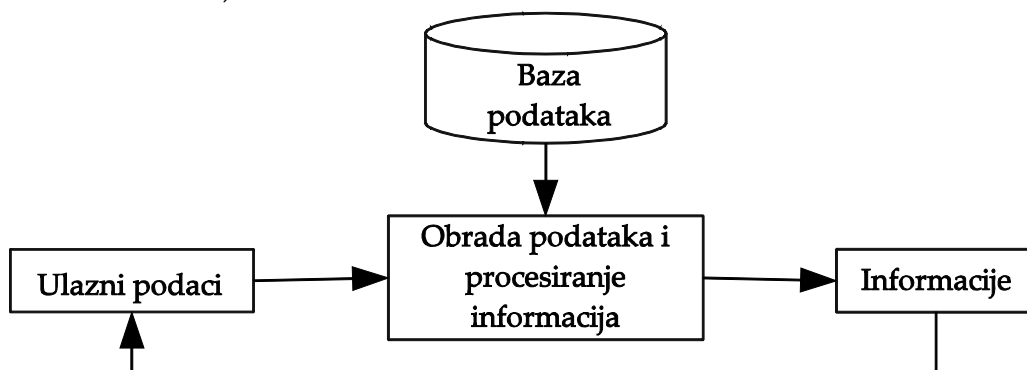
bazirana na prostom tekstu. Napredniji primer ovakvog sistema baziran je na slučajevima. Baza se pretražuje za najboljim slučajevima prema zahtevu. Rešenja pronađena na sličnim slučajevima nekada je , u celini ili delimično, moguće primeniti na trenutni slučaj (van der Aalst i van Hee, 2000; Hart i Gregor, 2005; Rainer i Cegielski, 2011; Hafeez-Baig i Gururajan, 2012; van Belle i ostali, 2001).

- *Sistemi za podršku u odlučivanje (engl. decision-support systems)* su informacioni sistemi koji donose odluke kroz interakciju sa ljudima. Postoje dva oblika ovih sistema: prvi su zasnovani na matematičkim modelima dok su drugi bazirani na logičkim sistemima rasuđivanja (primer su ekspertske sistemi) (van der Aalst i van Hee, 2000; Chaffey i Wood, 2005; Arnott i Pervan, 2007; Poliščuk, 2007; Rainer i Cegielski, 2011).
- *Kontrolni sistemi (engl. control systems)* su sistemi koji odluke donose potpuno automatski na bazi trenutnog stanja procesa.

Informacioni sistemi obično predstavljaju kombinaciju prethodno opisanih tipova sistema. Sa stanovišta efikasnosti, kontrolni sistemi predstavljaju idealno rešenje zato što ne zahtevaju ljudsku intervenciju. U praksi se pokazalo da ovakvi sistemi zapravo imaju izrazito ograničenu upotrebu u posebnim slučajevima. Dobro se pokazuju kod nekih problema operacionog menadžmenta. Sa druge strane, sistemi za podršku u odlučivanju koji rešavaju probleme kroz interakciju sa ljudima nude najveći potencijal zato što kombinuju ljudski faktor sa ogromnim mogućnostima računara (van der Aalst i van Hee, 2000).

3.1 Model informacionog sistema i njegove komponente

Informativni sistem predstavlja celinu koja se sastoji od bitnih komponenata koje izvršavaju aktivnosti sakupljanja i skladištenja podataka, njihove obrade i procesiranja. Takva obrada za rezultat ima nove informacije od kojih se mogu dobiti novi podaci. Slika 3.1 daje opšti prikaz modela jednog informacionog sistema (Balaban i ostali, 2010).



Slika 3.1: Model informacionog sistema

Ovaj model opisuje relevantne odnose i veze između ključnih delova jednog informacionog sistema. Ulaz predstavlja aktivnost koja je zadužena za pripremu i unos podataka u sistem odnosno bazu podataka. Obrada podataka pretvara ulazne podatke u informacije na osnovu odgovarajućih transformacija. To obuhvata izračunavanja, upoređivanja, sortiranja, sumiranja, klasifikacije i različite druge operacije. Podaci u elektronskom obliku se čitaju i pretvaraju u informacije. Izlaz ima ulogu dostavljača informacija koje mogu biti u obliku poruka, izveštaja, dokumenata sa ili bez grafičke vizuelizacije i slično (Balaban i ostali, 2010).

3.2 Resursi informacionih sistema

Informacioni sistem koristi određene resurse u svrhu svog funkcionisanja. Ti resursi mogu biti (Balaban i ostali, 2010; Hardcastle, 2008):

- ljudski resursi,
- hardverski resursi,
- softverski resursi,
- resursi podataka i
- resursi računarskih mreža.

Ljudske resurse jednog informacionog sistema čine IT specijalisti i krajnji korisnici informacionog sistema. Ljudski resursi su neophodni za funkcionisanje informacionog sistema. IT specijalisti su ljudi koji razvijaju, implementiraju i održavaju informacioni sistem. Ostali deo ljudskih resursa jednog informacionog sistema čine projektanti informacionog sistema, programeri, administratori baza, sistem analitičari, softver inženjeri, specijalisti za hardver i mreže, menadžeri, analitičari, istraživači, komercijaliste, tehničko osoblje i drugi (Balaban i ostali, 2010; Hardcastle, 2008).

Hardverske resurse predstavljaju kompletni računarski resursi koji podržavaju funkcionisanje informacionog sistema (Balaban i ostali, 2010; Hardcastle, 2008).

Softverski resursi uključuju sve vrste softvera upotrebljenog za funkcionisanje informativnog sistema. To mogu biti operativni sistemi, prevodioci, sistemi za upravljanje bazama podataka, aplikativni softver i slično (Balaban i ostali, 2010; Hardcastle, 2008).

Podaci su resurs koji ima veliko značenje kako za informacioni sistem tako i za organizaciju koja ga koristi. Čuva se u različitim oblicima baza podataka ili formatima dokumenata (Balaban i ostali, 2010; Hardcastle, 2008).

Resursi mreža su jedan od najznačajnijih delova informacionog sistema zbog toga što omogućavaju komunikaciju između delova informacionog sistema. Pravilno funkcionisanje ove komponente obezbeđuje nesmetan rad informacionog sistema. U navedenu grupu spadaju (Balaban i ostali, 2010; Hardcastle, 2008):

- komunikacioni mediji, kao što su koaksijalni kablovi, fiber-optički kablovi, satelitski komunikacioni sistemi i drugi,

- mrežna oprema: ruteri, svičevi, modemi, habovi, razne vrste priključaka i druga mrežna oprema, i
- komunikacioni kontrolni softver.

3.3 Principi i modeli razvoja informacionih sistema

Opšte poznati principi razvoja (aktivnosti u razvoju) informacionih sistema su (Balaban i ostali, 2010):

- Modelovanje - koje podrazumeva izbor modela, rukovanje različitim nivoima detaljnosti modela.
- Apstrakcija - kao rezultat modelovanjem stečenih rezultata, daje specifikaciju modela koja je nedvosmislena, razumljiva, jedinstvena, nezavisna od sistema i lako izmenljiva.
- Iteracija - ponovljiv ciklus aktivnosti koji je uzrokovan različitim razlozima.
- Arhitektura - dekomponovanje složene strukture proizvoda na njegove segmente odnosno arhitekturu bilo u analizi i dizajnu, bilo u održavanju. Razlozi za strukturiranje su: lakša razumljivost složenih sistema, jednostavnije organizovanje razvojnog procesa, lakša ugradnja elemenata u celinu sistema, upravljivost kompleksnih i složenih sistema.
- Dokumentovanje - podrazumeva neophodnost istovremene izrade dokumentacije u toku realizacije pojedinih aktivnosti razvoja. Na taj način dokumentacija obezbeđuje njihovo nesmetano odvijanje. Dokumentacija doprinosi kvalitetnijoj komunikaciji i razmeni informacija između svih učesnika razvoja, kao i njihovoj međusobnoj usaglašenosti. Takođe, ona pruža osnovu za kontrolu procesa razvoja. Dokumentacija je jednako potrebna ekspertima u procesu razvoja informacionog sistema, kao i krajnjim korisnicima budućeg sistema. Upravo zbog toga se proces izrade dokumentacije, sadržaj dokumentacije, način izrade, upravljanja i čuvanja dokumentacije uređuju međunarodnim standardima.

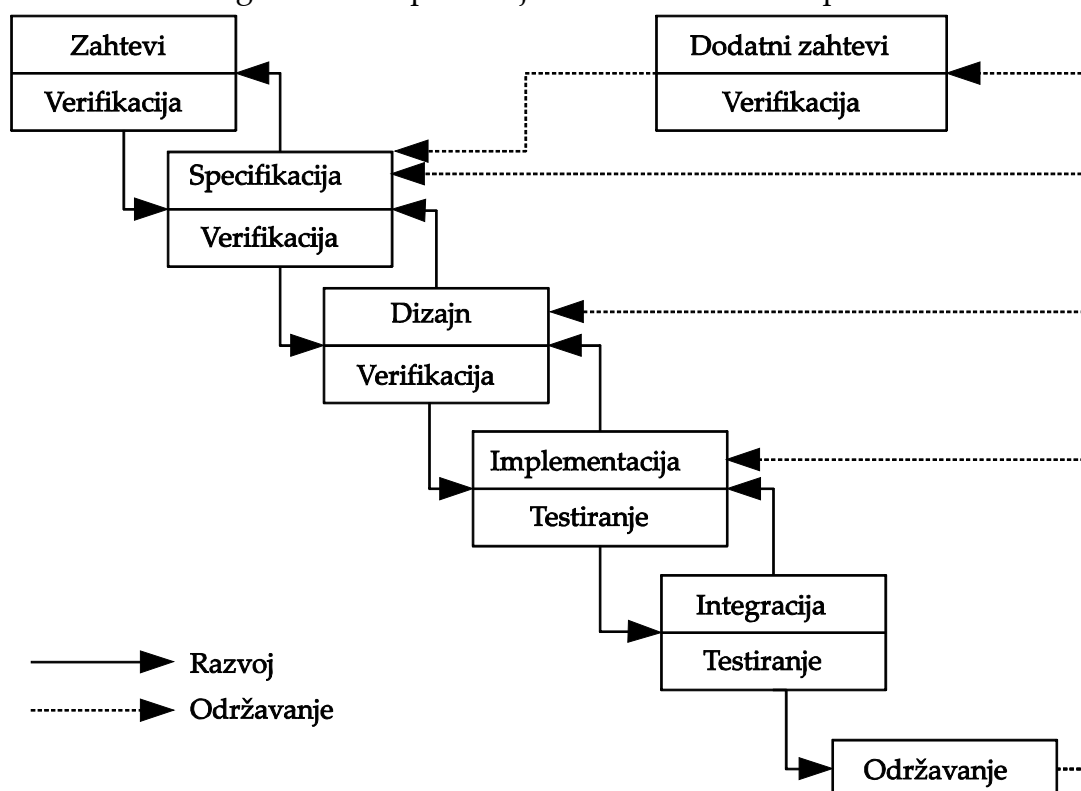
Model razvoja je apstraktna (teorijska) predstava procesa razvoja. Svaki model predstavlja proces na poseban i jedinstven način, te tako obezbeđuje samo delimične informacije o njemu. Model razvoja se bira u zavisnosti od prirode zadatka odnosno projekta, tehničke orijentacije osoba koje učestvuju u razvoju, metoda i alata koji se upotrebljavaju pri razvoju, načina kontrole i prirode samog proizvoda koji se zahteva. Najčešće, modeli se strukturiraju na sledeći način od kojih će biti opisani samo pojedini (Poliščuk, 2007; Balaban i ostali, 2010; Hardcastle, 2008):

- Model vodopada,
- Modifikovani model vodopada,
- Inkrementalni model,
- RAD model.

- Model prototipskog razvoja,
- Spiralni model,
- Model zasnovan na komponentama,
- Model unificiranog procesa razvoja (Unified Process)
- Modeli agilnog razvoja (Extreme Programming (XP), Adaptive Software Development (ASD), Dynamic Systems Development Method (DSDM), Scrum, Feature Driven Development (FDD), Agile Modeling (AM),
- Kombinovani modeli.

U nastavku će biti prikazane osnovne karakteristike samo nekih od navedenih modela (Poliščuk, 2007; Balaban i ostali, 2010; Hardcastle, 2008; Milićev, 2009).

Model vodopada (engl. Waterfall model) je model koji je uveden 1970. godine. Uveo ga je W. Royce. Svoj naziv ovaj model duguje toku izvođenja operacija i načinu prenošenja informacija iz jedne faze u drugu – nešto slično vodopadu. Kod ovog modela je prisutan sekvencijalni redosled obavljanja operacija. Informacije se prenose od jedne faze do druge. Slika 3.2 prikazuje šemu modela vodopada.



Slika 3.2: Šema modela vodopada

Razvoj se odvija kroz sledeće faze (Balaban i ostali, 2010; Hardcastle, 2008, Milićev, 2009; Poliščuk, 2007):

- Analiza i specifikacija zahteva – U ovoj fazi identifikuju se problem i ciljevi koji se žele postići razvojem proizvoda, zahtevane funkcije, performanse i njihove međusobne veze. Takođe, identifikuju se potencijalna ograničenja u razvoju. Zahtevi se analiziraju i pregledaju sa

korisnicima i na kraju dokumentuju. Specifikacija zahteva jednoznačno i jasno definiše funkcije budućeg proizvoda.

- Projektovanje - Projektovanje ili dizajn proizvoda je faza razvoja, kojom se definiše celokupna arhitektura sistema. U projektovanje spadaju aktivnosti razvoja korisničkog interfejsa, ulazne ekranske forme, izlaza, baze podataka, procedura obrade i systemske kontrole.
- Implementacija - Ovom fazom se izvršava zadatak prevođenja rezultata projektovanja u mašinski prepoznatljivu formu. U fazu implementacije spadaju aktivnosti programiranja, testiranja, konverzije i obuke korisnika.
- Integracija – Unutar ove faze svi generisani programi se integrišu da bi se utvrdilo da li funkcionišu kao celina i da li takav sistem zadovoljava specifikaciju zahteva.
- Funkcionisanje i održavanje – U toku ove faze vrši se testiranje i neprekidno inoviranje proizvoda. Najčešći razlozi izmena i dopuna softvera su proširenja mogućnosti, ispravljanje eventualnih grešaka i poboljšanje sveukupnih performansi.

Prema modelu svaka od faza ima ulaz, proces obrade i izlaz. Razvoj softvera prolazi kroz nekoliko sukcesivnih faza. Svaka faza daje određenu vrstu proizvoda ili dokumentaciju. U realnosti, razvoj nužno ne prati sve ove faze. Iz tog razloga nakon svake faze postoji revizija faze pre prihvatanja rezultata (Balaban i ostali, 2010).

Model vodopada je najstariji i najrasprostranjeniji model koji se danas koristi. Može se kombinovati sa ostalim modelima razvoja.

Prednosti ovog modela su:

- strogo definisani i kontrolisani proces, kojeg karakterišu standardizovane i detaljno opisane aktivnosti u svim fazama razvoja,
- uključeno testiranje odnosno verifikacija izvršenih aktivnosti i dobijenih rezultata na kraju svake faze razvoja,
- detaljna i kvalitetna dokumentacija, koja se generiše u svim fazama razvoja, istovremeno kada se izvršavaju pojedine aktivnosti
- relativno laka zamena pojedinih učesnika u procesu razvoja.

Nedostaci modela vodopada su:

- nefleksibilna podela aktivnosti razvoja u posebne faze i nedostatak povratne sprege između faza,
- greške koje se ne otklone u pojedinim fazama razvoja, kada se vrši testiranje ili verifikacija proizvoda, mogu imati stravično distorziono dejstvo na razvoj u celini,
- nemogućnost obavljanja iteracija tokom realizacije razvoja jer iste izazivaju ozbiljne probleme i konfuziju u primeni modela,
- teška prilagodljivost neizvesnosti koja uglavnom egzistira na startu projekta, kada je korisniku veoma teško da eksplicitno navede sve svoje zahteve prema softveru,

- dugotrajan proces razvoja te korisnik mora biti veoma strpljiv i istrajan jer su mu radne verzije softvera dostupne tek na kraju aktivnosti razvoja, a do tada postoji samo pisana specifikacija funkcionalnosti budućeg softvera,
- samo potpuno gotov proizvod je upotrebljiv od strane korisnika,
- visoki razvojni troškovi.

Postoji i modifikovani model vodopada koji je razvijen da bi se otklonila dva najveća nedostatka klasičnog linearnog modela vodopada. To je mogućnost preklapanja aktivnosti razvoja i realizacija povratne sprege između faza razvoja (Balaban i ostali, 2010).

Inkrementalni model (slika 3.3) je nastao od modela vodopada. Predstavlja kombinaciju klasičnog modela vodopada i interaktivnih elemenata. Kod ovog modela prvobitno se razvija inicijalni podskup funkcija proizvoda, a zatim se u narednim koracima razvijaju uvek novije i komplikovanije verzije proizvoda. Svakim inkrementom se razvijaju nove funkcionalnosti koje se dodaju već razvijenom proizvodu, pri čemu se postojeće funkcionalnosti zadržavaju. Proizvod je razvijen kada zadovolji sve identifikovane korisničke zahteve odnosno sve funkcionalnosti. Međutim, on je upotrebljiv i nakon razvijenog prvog inkrementa i bez razvijenih svih ostalih funkcionalnosti. Razvoj se može prekinuti pri razvoju bilo kojeg inkrementa bez rizika za njegovu trenutnu upotrebljivost. Inkrementalni model razvoja je posebno popularan i koristi se u softverskim kućama (Balaban i ostali, 2010; Poliščuk, 2007).

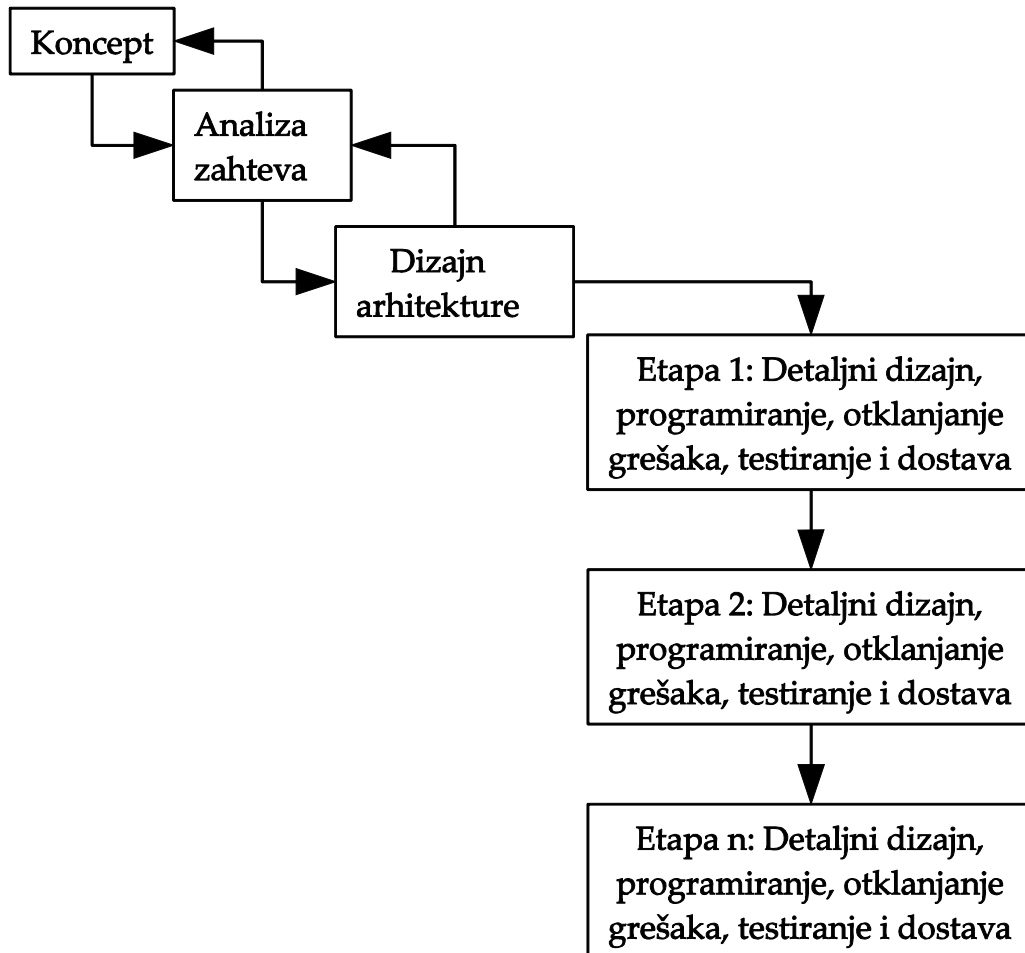
Prednosti inkrementalnog modela su:

- obezbeđuje transparentan razvoj proizvoda, sa stalno vidljivim rezultatima,
- uvek raspoloživ funkcionalno upotrebljiv proizvod, koji zadovoljava određeni podskup korisničkih zahteva,
- lako razumevanje i testiranje novorazvijenih inkremenata proizvoda jer oni samo dodaju nove funkcionalnosti postojećem upotrebljavanom softveru i na taj način obezbeđuju premošćavanje traumatskih efekata uvođenja kompletno novog proizvoda odjednom, postojanje povratne sprege i permanentne mogućnosti ugradnje bogatog korisničkog iskustva u redefinisani proizvod na manje skup način, putem novih inkremenata odnosno novih funkcionalnosti proizvoda,
- umanjeni rizik od neuspeha razvoja celine, jer se problemi uglavnom uočavaju u pojedinim inkrementima,
- skromniji obim kapitalnih ulaganja u razvoj proizvoda i brži povrat investicija, manji broj angažovanih osoba u procesu razvoja.

Nedostaci inkrementalnog modela su:

- dekompozicija proizvoda na inkremente, da bi se oni mogli integrisati, nije trivijalan zadatak, kao ni sam proces integracije, a da se pri tome ne ugrozi kvalitet već postojećeg proizvoda,

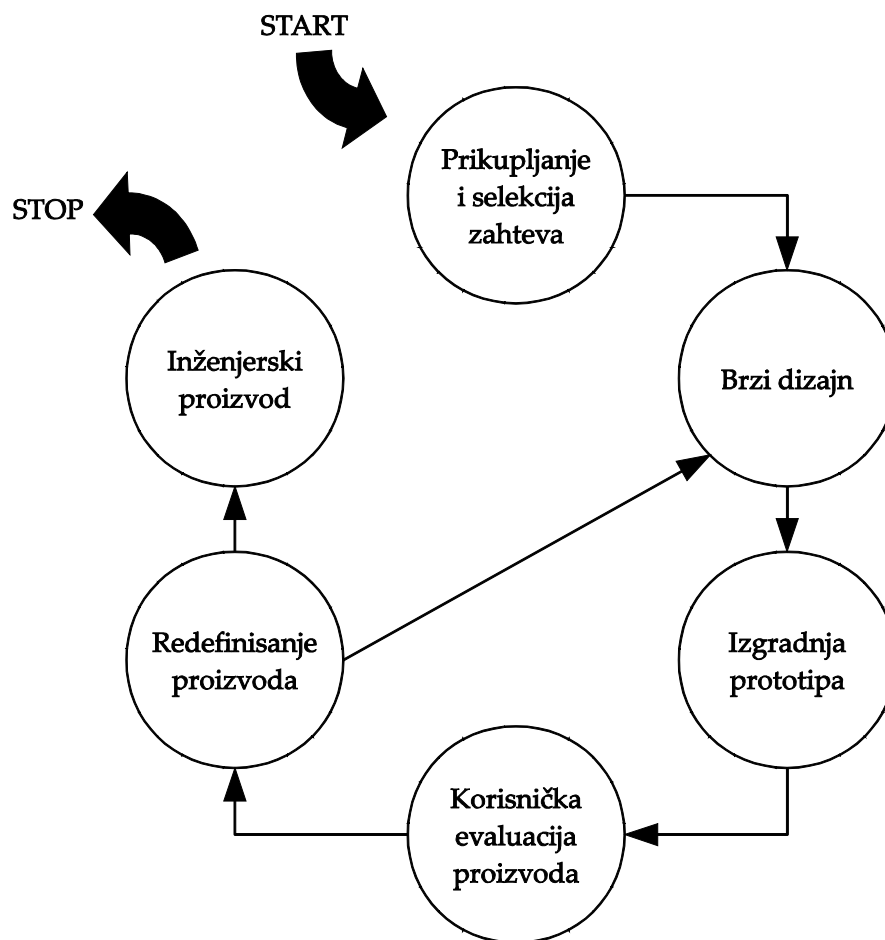
- specifikacija detaljnih korisničkih zahteva se kod svakog inkrementa izrađuje neposredno pre nego što se on razvija,
- integracija može uvek doneti neočekivane probleme i potrebe za reorganizacijom, koja može imati posledice po efikasnost i održavanje,
- korisnici imaju stalnu želju da menjaju svoje zahteve.



Slika 3.3: Inkrementalni model

Jedna modifikovana verzija ovog modela je RAD model koji omogućava izradu proizvoda za manje od 90 dana.

Model prototipskog razvoja predstavlja iterativni model. U okviru ovog modela pored projekatanta razvojnom timu se priključuju i korisnici. Na osnovu korisničkih zahteva kreira se inicijalni proizvod (najčešće samo interfejs) koji simulira osobine stvarnog proizvoda. Na osnovu inicijalnog proizvoda korisnik odlučuje koji su njegovi zahtevi vezani za finalni proizvod. U okviru ovog modela se izrađuje prototip koji se po potrebi doraduje i na osnovu koga se na kraju izrađuje finalni proizvod.



Slika 3.4: Model prototipskog razvoja

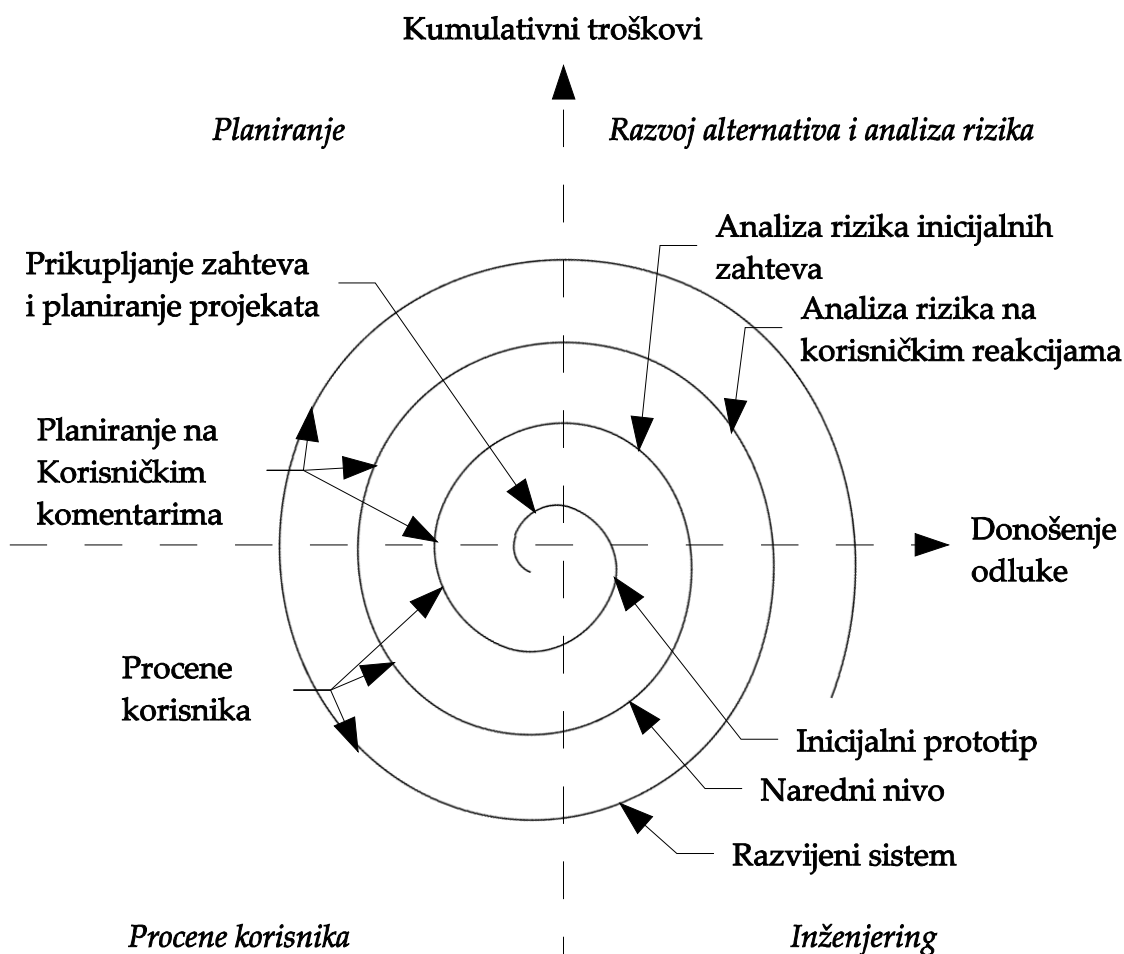
Proces razvoja proizvoda upotrebom ovog modela započinje prikupljanjem i selekcijom zahteva (slika 3.4). Uz konsultacije sa korisnicima definišu se zahtevi koje proizvod treba da ispuni. Nakon ove faze sledi brzi dizajn proizvoda. U ovom delu fokus je na onim delovima proizvoda koji su vidljivi korisniku i sa kojim korisnik ima neposrednu interakciju. Na osnovu brzog dizajna proizvoda izrađuje se prototip proizvoda. On obično služi za potvrdu svih specifikacija i zahteva i njihovu eventualnu ispravku i dopunu. U ovom trenutku korisnik zaista može da isproba proizvod koji očekuje. Zahtevi se dopunjavaju i ispravljaju sve dok prototip ne zadovolji sve potrebe korisnika. Pored modifikacije postojećih zahteva moguće je i pojavljivanje novih zahteva što zahteva dodatne radnje. Kada je evaluacija proizvoda od strane korisnika završena, vrši se eventualno redefinisavanje proizvoda. Krajnja faza izrade proizvoda po ovom modelu je izrada inženjerskog odnosno finalnog proizvoda.

Dobre strane ovog modela su poboljšana kreativnost u toku razvoja, više radnih verzija proizvoda koje prikazuju njegovu funkcionalnost i prisustvo korisnika što u mnogome olakšava proces izrade. Loše strane ovog modela su nemogućnost procene realnih troškova proizvodnje, nepoznavanje principa rada proizvoda od strane korisnika, razne vrste kompromisa koje projektanti moraju da čine da bi prototip što pre počeo da funkcioniše, velika verovatnoća da inženjerski proizvod baziran na

prototipu ne funkcionise kako treba i nepotpuna dokumentacija koja često ne bude ni napravljena (Balaban i ostali, 2010; Polišćuk, 2007).

Različita istraživanja su sprovedena kako bi se utvrdio uticaj participacije korisnika na uspeh informacionog sistema (Cavaye, 1995; Hwang i Thorn, 1999). Rezultati su izrazito konfliktni. Neka istraživanja su pokazala da participacija korisnika doprinosi uspehu informacionih sistema, druga su pak dala obrnutu tvrdnju (Mattia i Weistroffer, 2009).

Spiralni model je napravljen kao model koji će objediniti najbolje osobine modela vodopada i modela prototipskog razvoja.



Slika 3.5: *Spiralni model*

Ovaj model razvoja obuhvata četiri faze razvoja (prikazane na slici 3.5) (Balaban i ostali, 2010; Hardcastle, 2008; Polišćuk, 2007).

- planiranje – faza u kojoj se uspostavljaju kontakti između projektanata i korisnika budućeg sistema. Određuju se ciljevi i ograničenja
- razvoj alternativa i analiza rizika – faza u kojoj se analiziraju alternativna rešenja i rizici
- inženjering – faza namenjena za razvoj i testiranje novih nivoa proizvoda uz podršku korisnika

- procene korisnika – procena realizovanih rešenja od strane korisnika sistema koji se projektuje

Svakom iteracijom kreira se složeniji i kompletniji proizvod. U prvom ciklusu se prikupljaju zahtevi potrebni za planiranje projekta. Po donošenju odluke o daljem razvoju obavlja se inženjering u svakom ciklusu. Po završetku svakog ciklusa razvoja korisnik ocenjuje proizvod i daje sugestije za poboljšanja i prepravke. Na osnovu datih informacija inicijalizuju se nove faze planiranja novog ciklusa razvoja. Svaki od ciklusa razvoja zahteva analizu rizika i donošenje odluke da li nastaviti razvoj ili ne. Iteracije kroz cikluse se ponavljaju analizirajući rezultate prethodnih faza.

Prednosti ovog modela su kratko vreme realizacije funkcionalnog proizvoda, fleksibilnost u upravljanju fazom inženjeringa i mogućnost kombinovanja različitih pristupa razvoju, mogućnost izvođenja i procene rizika u svakom trenutku i td. Neki od nedostataka su odsustvo veze prema postojećim standardima, to što model zahteva više uniformnosti i konzistenciju u razvoju, kao model je relativno skup za primenu na malim projektima, problemi koji se pojavljuju neotkrivanjem ili zakasnelim otkrivanjem rizika, relativno kratko vreme primene (Balaban i ostali, 2010; Hardcastle, 2008, Milićev, 2009; Polišćuk, 2007).

3.4 Metodi modelovanja

Modeli, generalno posmatramo, mogu biti apstraktnog oblika kao za primer matematički model, šeme ili nacrti. Sa druge strane, modeli mogu biti fizički modeli. Obično su to makete napravljene od plastike, drveta ili sličnih materijala. Modeli omogućavaju testiranje karakteristika sistema pre nego što se sam sistem izradi. Softversko inženjerstvo takođe koristi modele za istu namenu. U ovom slučaju modeli nisu pravljeni od drveta ili metala nego predstavljaju formalnu i nedvosmislenu specifikaciju koja može biti direktno interpretirana od strane nekog softverskog sistema ili može biti transformisana u oblik izvršiv od strane hardvera ili nekog drugog softverskog sistema. U ovom slučaju model je izvršiv i predstavlja praktično model u razvoju (Milićev, 2009).

Glavna uloga modela jeste obezbeđivanje boljeg razumevanja osobina i načina funkcionisanja sistema koji se razvija. Postoje četiri cilja koji se postižu modelima (Booch, 1999; Milićev, 2009):

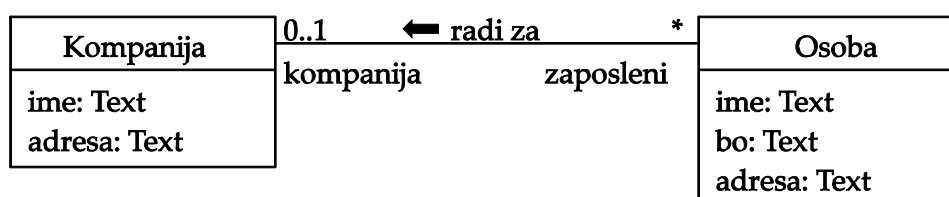
- vizuelizacija – modeli doprinose vizuelizaciji sistema,
- specifikacija – dozvoljava dizajneru određivanje strukture i osobina sistema,
- konstrukcija – daje šablone koji se koriste pri konstrukciji sistema,
- dokumentacija – dokumentovanje odluka i akcija preduzetih tokom razvoja.

Programi kreirani tradicionalnim programskim jezicima mogu se tretirati kao modeli iz razloga što specificiraju sistem na formalni i izvršni način. Sa druge strane,

programi kreirani tradicionalnim programskim jezicima se ne smatraju modelima zato što (Milićev, 2009):

- modeli treba da poseduju apstraktnu notaciju,
- modeli mogu da budu nekompletni ili neprecizni, prvenstveno u ranim fazama
- modeli su obično specificirani vizuelnom notacijom (dijagramima) u kombinaciji sa tekstualnim elementima, dok kod programskih jezika dominiraju tekstualne forme.

Slika 3.6 daje prikaz poređenja vizuelnog i tekstualnog modelovanja. Opisuje situaciju u kojoj jedna osoba može da radi samo za jednu kompaniju, dok za kompaniju može da radi više osoba. Osobu odlikuje ime, broj osiguranja, adresa i kompanije u kojoj je zaposlena. Kompaniju odlikuje samo ime i adresa.



a)

```

class Kompanija {
    public:
    String ime;
    String adresa;
    Collection<Osoba*> zaposleni;
};

class Osoba {
    public:
    String ime;
    String bo;
    String adresa;
    Kompanija* kompanija;
};
    
```

b)

Slika 3.6: Abstraktno (a) naspram low-level modelovanja (b) i vizuelno (a) naspram tekstualnog modelovanja (b)

Alati koji se koriste za modelovanje nazivaju se prosto model editori. Pored model editora postoje i alati koji su zaduženi za prevođenje modela u low-level formu i nazivaju se transformeri ili translator (u specifičnim situacijama generatori ili kompajleri). Kod generatori (nekada nazvani model kompajleri) su alati koji prevode modele visokog nivoa u kod (npr. C++, Java ili C#). Na kraju liste alata nalaze se kompajleri koji kod programskog jezika prevode u binarnu ili interpretabilnu formu (Milićev, 2009).

Prema Ahlemann-u, modeli igraju važnu ulogu u razvoju i implementaciji informacionih sistema. U zavisnosti od faze razvoja informacionog sistema, postoje tri tipa modela se mogu izdvojiti (Ahlemann, 2009):

- Konceptualni modeli – koji pomažu dokumentovanju, analiziranju i razumevanju zahteva koje informacioni sistem treba da ispuni,

- Dizajn modeli – daju specifikaciju arhitekture informacionog sistema opisujući veće delove sistema (komponente). Opisi ovakvog tipa nisu detaljni,
- Implementacioni modeli – su usko vezani za programiranje softvera i zavise od tehnologije.

Kada su u pitanju jezici za modeliranje, postoji nekoliko jezika koji se često koriste pri modelovanju informacionih sistema. Jedan od najrasprostranjenijih jezika jeste Unified Modeling Language (UML). Za modeliranje podataka mogu se iskoristiti klasni dijagrami dok se za modelovanje procesa koriste dijagrami aktivnosti. I jedni i drugi su deo UML jezika.

3.5 Sistemi za upravljanje radnim tokovima

Prema van der Aalst-u i Van Hee-u radni tok (*eng. workflow*) predstavlja ideje, metode, tehnike i softver (aplikacije) koje se koriste kao podrška struktuiranim poslovnim procesima. Zadatak upravljanja radnim tokovima jeste ostvarivanje lako održivih i lako upravljivih radnih procesa (van der Aalst i Van Hee, 2000; Georgakopoulos, 1995). Još jedna definicija radnog toka koju su dali isti autori glasi: radni tok je definisan kao mreža zadataka ili zaduženja koja potpuno ili delimično određuje redosled kojim će pomenuti zadaci ili zaduženja biti izvršeni (van der Aalst i Van Hee, 2000; Shi i ostali, 2008; Bider, 2005; Joeris, 1999, Adams i ostali, 2007, Reichert i Bauer, 2007; Agarwal, 2000).

Takođe, može se reći da je radni tok „kompjuterizovano olakšavanje“ ili automatizacija celih poslovnih procesa ili nekih njihovih delova (Zhuge i ostali, 2001; WfMC, 1999, Cicirelli i ostali, 2010; Georgakopoulos i ostali, 1995; Smith, 1993). Takva automatizacija je uspešno ostvarena korišćenjem i poštovanjem pravila koja zadaje radni tok (Perez i Rojas, 2000).

Chundi (2011) je rekao da svaki radni tok predstavlja logičku celinu zaduženja ili zadataka koji se izvršavaju po određenom redosledu i tačno definisanim međuzavisnostima. Izvršenje radnog toka mora da očuva date zavisnosti i da se eventualno privede i kraju (Chundi i ostali, 2011).

Radni tok (*eng. Workflow*) često se poistovećuje i navodi kao sinonim za poslovni proces (*eng. business process*) (van der Aalst i Van Hee, 2000; WfMC, 1996; Jablonski i Bussler, 1996). Predstavlja izraz dat procedurama koje uključuju usmeravanje zadataka od osobe do osobe u sekvenci odnosno nizu, dozvoljavajući pri tom svakom pojedincu da doprinese procesu pre nastavka na sledeći korak procesa proizvodnje (Fakas i Karakostas, 1999). Koncept radnog toka razvijen iz ideje automatizacije fabričke proizvodnje, a prvi prototip je razvijen u toku 1970-ih godina (Koulopoulos, 1995; Salimifard i Wright, 2001).

Sistemi za upravljanje radnim tokovima (u daljem tekstu SURT) (*eng. Workflow Management Systems*) svojim mogućnostima privlače veliku pažnju kao sistemi

dizajnirani da podrže i pomognu poslovne procese. Poslovni proces se sastoji od niza koraka (aktivnosti) koje se mogu izvršiti automatski, ručno ili kombinacijom dva navedena načina. SURT se ne bavi samo podržavanjem izvršenja aktivnosti (zadataka) (Salimifard i Wright, 2001, van der Aalst, 1998) nego vodi računa i o raspodeli i dodeli zaduženja zaposlenima i omogućava praćenje izvršenja pomenutih definisanih zaduženja. Pored praćenja, sistem za upravljanje radnim tokovima ima mogućnost prikupljanja i statističke obrade podataka o procesu proizvodnje ili zaposlenima (Chundi i ostali, 2011; WfMC, 1999; Hales i Lavery, 1991; Ellis i Nutt, 1996; Hollingsworth, 1995; Leymann i Roller, 1997). Do skoro je smatrano da predstavljaju jedan od glavnih primera napretka u oblasti informacionih sistema (Chiu i ostali 2001).

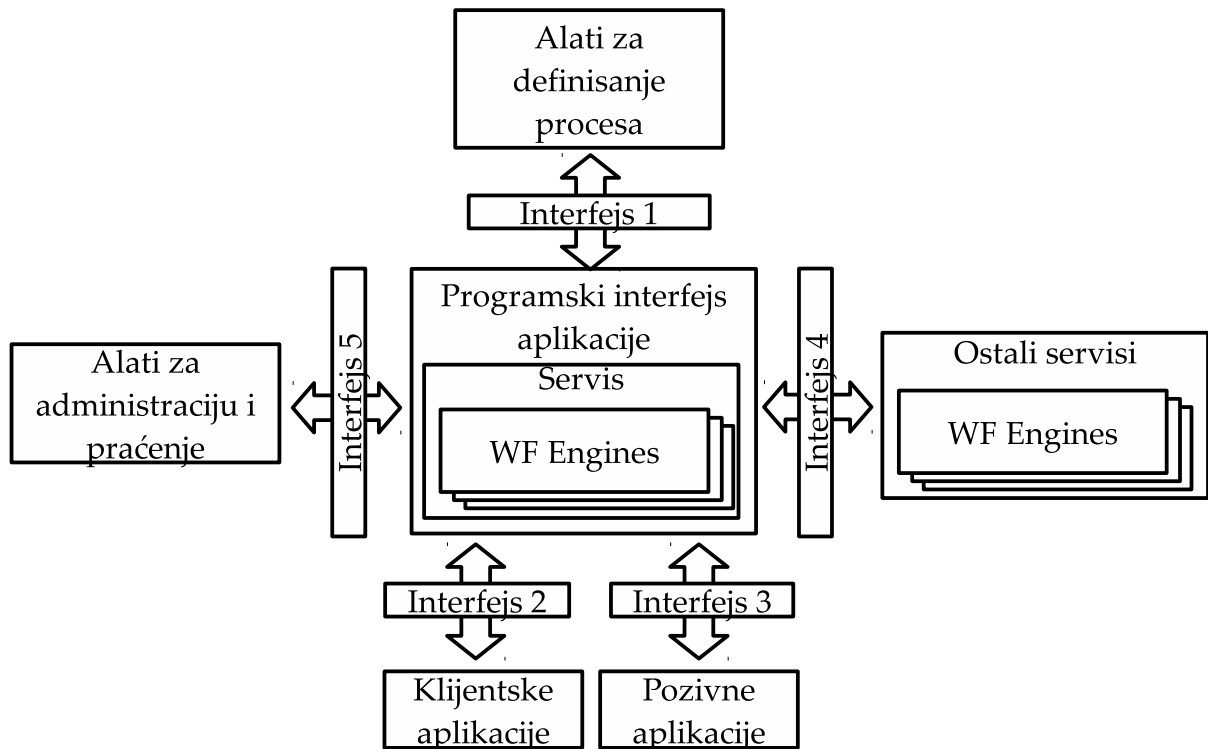
SURT se može definisati i kao sistem koji kompletno definiše, izvršava i upravlja radnim tokom kroz izvršavanje softvera čiji tok izvršavanja je vođen računarskom reprezentacijom logike radnog toka (Zhuge i ostali, 2001, WfMC, 1999, Salimifard i Wright, 2001) koji mora da poseduje zadovoljavajući nivo fleksibilnosti kako bi uspeo da podrži širok spektar procesa (Reichert i Dadam, 1998; Weske, 1998; Reichert i Bauer, 2007; Lenz i Reichert, 2007) što nekada podrazumeva i ad-hok promene toka procesa u slučaju izuzetnih situacija (Weber i ostali, 2007; Lenz i Reichert, 2007). Ovakve mogućnosti poseduje ADEPT (Rinderle i ostali, 2004). Generalno predstavljaju softver za podršku kontroli i upravljanju procesima kroz automatizaciju zadataka (Bae i ostali, 1999). SURT koordiniraju izvršenje međusobno logički povezanih zadataka (Chundi i ostali, 2011, Choi i ostali, 2008; WfMC, 1999).

Obično se sastoji iz tri dela (Zhuge i ostali, 2001; WfMC, 1999):

- funkcija koje su odgovorne za definisanje i modelovanje radnog toka
- funkcija koje su odgovorne za izvršavanje, praćenje i upravljanje radnim tokom
- funkcija koje su odgovorne za omogućavanje interakcije sistema sa ljudima

Tehnologija radnih tokova je izuzetno napredovala i postala izvodljiva zahvaljujući novim tehnologijama komunikacije među ljudima. Poslovni sistemi koriste SURT za praćenje proizvodnog procesa, za iniciranje određenih segmenata proizvodnje, praćenje rokova, statistiku, praćenje i utvrđivanje efikasnosti zaposlenih i celog proizvodnog procesa (Sinderen i ostali, 1999).

Referentni model jednog SURT prikazan je na slici 3.7 (Hollingsworth, 1995; WfMC, 1999, Cicirelli i ostali, 2010). Razvijen je od strane Workflow Management Coalition (WfMC) koja je osnovana 1993. kao internacionalna organizacija čija je misija promocija radnih tokova i uspostavljanje standarda vezanih za SURT (van der Aalst, 1996, Carlsen, 1997).



Slika 3.7: Referentni model SURT

Referentni model razvijen je da bi se dobile uniformne smernice za razvoj SURT. Model ilustruje osnovne komponente i interfejsse SURT.

Servis predstavlja srce SURT. WfMC je definisala servis kao softverski servis koji se može sastojati od jednog ili više workflow enginea uz pomoć kojih se kreiraju, upravljaju i izvršavaju procesi radnog toka. (WfMC, 1999, Ellis i Kedara, 1993). Ovaj deo je odgovoran za kreiranje novih slučajeva, generisanje radnih stavki po ugledu na opis procesa, odabir resursa, podršku aktivnostima kao i zapis performansi radnog toka. Ukratko, ovaj servis omogućava da se pravi zadaci obave u pravo vreme na pravi način od strane pravih pojedinaca.

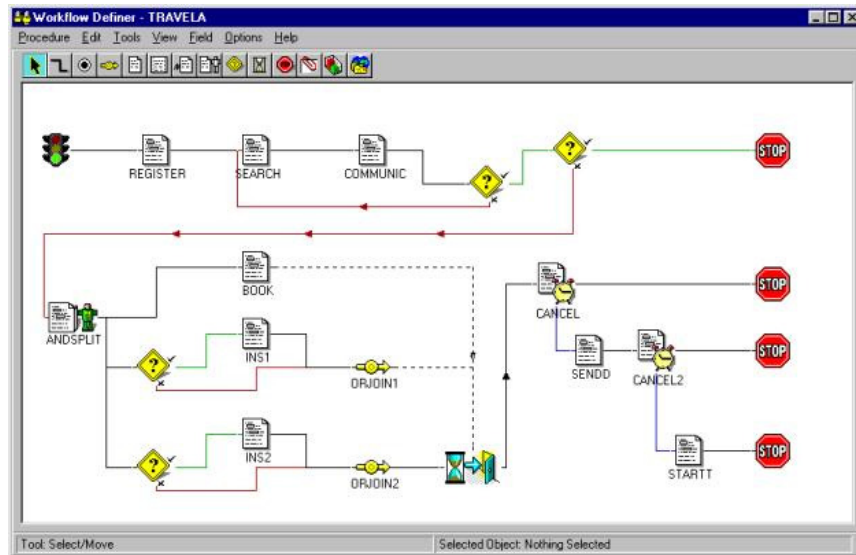
Interfejsi su zaduženi za komunikaciju servisa sa ostalim komponentama sistema (van der Aalst, 2000; WfMC, 1999):

- Alati za definisanje procesa služi za definisanje i konstrukciju definicije radnog toka.
- Klijentske aplikacije služe isključivo u svrhu izvršenja radnog procesa. Koriste ih pojedinci učesnici proizvodnog procesa. Klijentske aplikacije sadrže liste aktivnosti (zaduženja) koje treba izvršiti.
- Dodatne aplikacije ne predstavljaju deo SURT. To su spoljne aplikacije koje se pozivaju po potrebi, ukoliko je to zahtevano od strane SURT. One pripadaju globalnom radnom sistemu. Mogu biti potpuno automatizovane ili interaktivne aplikacije.
- Ostali servisi – jedan SURT može imati više workflow engine-a. Ovi servisi se integrišu u SURT na isti način kao i glavni servis.

- Alati za administraciju i praćenje služe za praćenje i administraciju sistema.

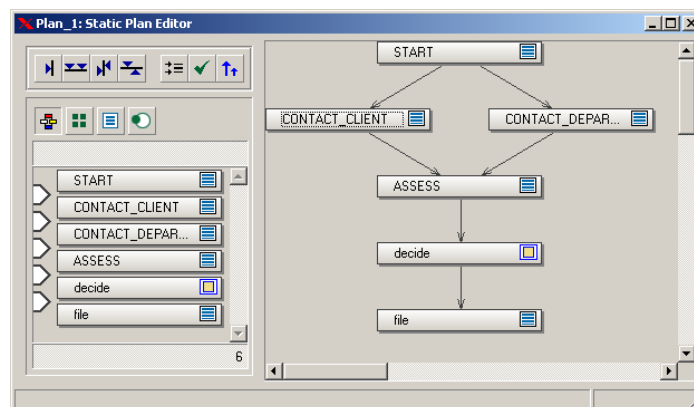
Danas postoji veliki broj SURT prisutnih na tržištu na koje se mogu dodati i modeli otvorenog koda i akademski modeli (van der Aalst, 2000). U nastavku će biti dat prikaz nekih popularnih SURT.

Staffware se sastoji od nekoliko komponenata koje se koriste za definisanja modela procesa, korisnika i njihovih uloga i izvršavanje procesnih modela. Slika 3.8 prikazuje jedan primer rada Staffware-a (<http://www.is.win.tue.nl>, 2013).



Slika 3.8: Primer rada softvera Staffware, modul Graphical Workflow Definer (Eindhoven University of Technology (TU/e), 2013)

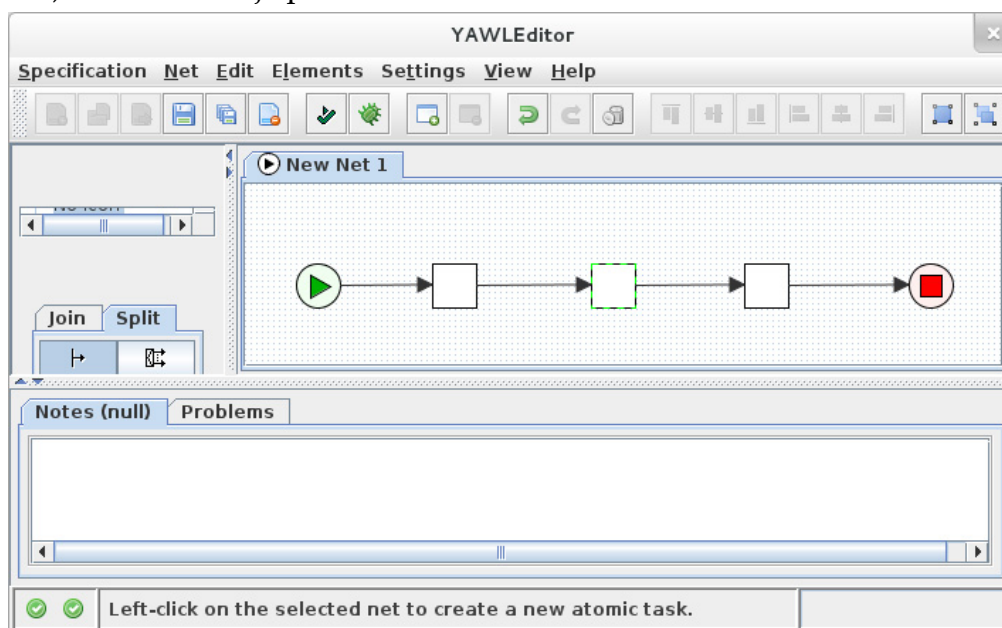
FLOWer je oblik SURT koji se bavi slučajevima (van der Aalst, 2001; van der Aalst, 2005; Reijers i ostali, 2003; van der Aalst i ostali, 2003). Slika 3.9 daje prikaz rada pomenutog sistema.



Slika 3.9: FLOWer

YAWL je baziran na poznatim šablonima radnih tokova. (Leonardi i ostali, 2007; Holman i Lorig, 2004). Predstavljen je kao sistem koji nudi visok nivo fleksibilnosti (van der Aalst i Jablonski 2000; Agostini i De Michelis, 2000; Casati i

ostali, 1996; Ellis i Keddara, 2000; Herrmann i ostali, 2000; Klein i ostali, 1998; Klein i ostali, 2000). Slika 3.10 daje primer YAWL editora:



Slika 3.10: YAWL primer

3.6 Jezici za modeliranje radnih tokova

U SURT, poslovni procesi igraju dominantnu ulogu. Jezici za modeliranje radnih tokova su neophodni za modeliranje i analizu procesa. Popularni jezici za modeliranje radnih tokova su: Petri Nets (Petri mreže), Web Service Business Process Execution Language (WS-BPEL) i Yet Another Workflow Language (YAWL). Uz pomoć tih jezika moguće je modelirati procese proizvodnje koristeći uniformnu sintaksu i strukturu. Neki od tih jezika su izvršivi, što znači da proces proizvodnje koji je njima definisan može biti direktno izvršen od strane sistema čiji softver podržava i razume oblik zapisa datih modela.

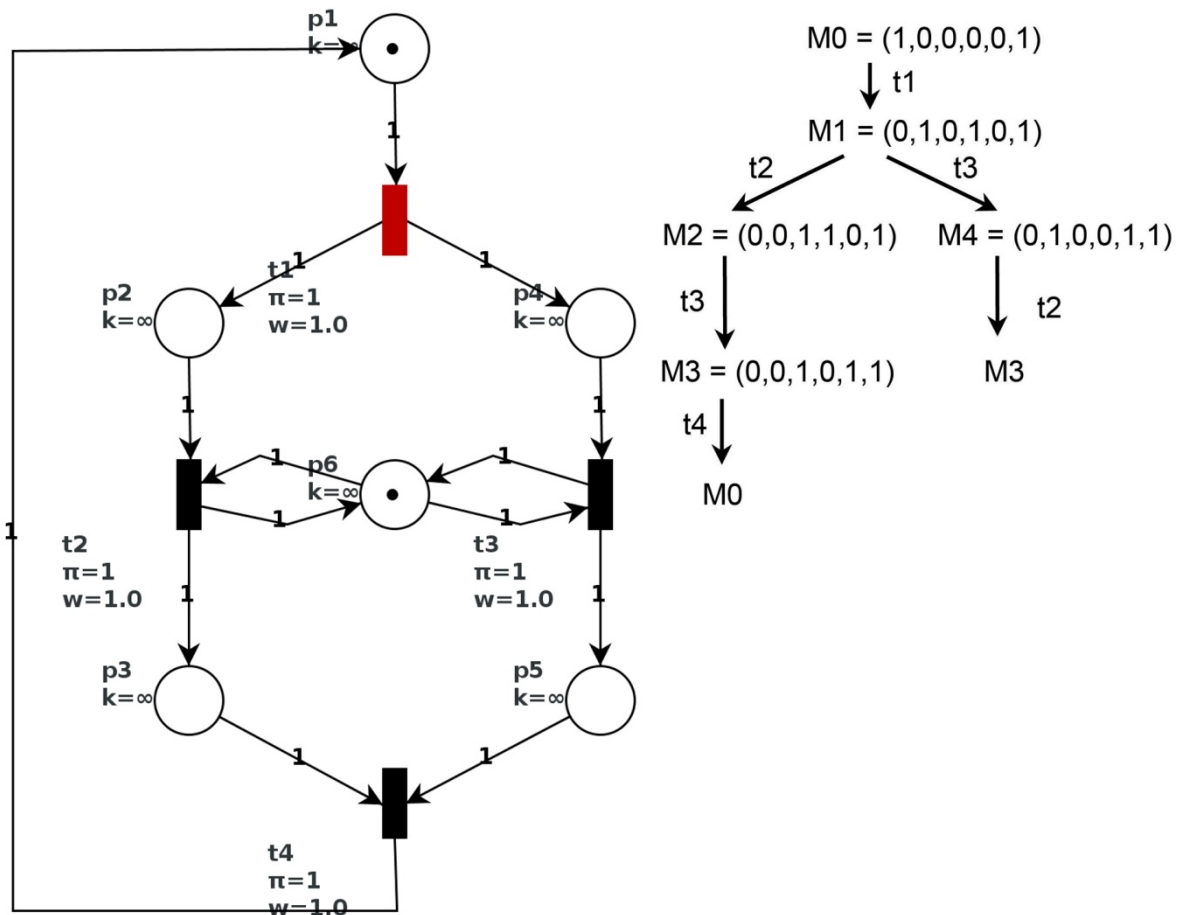
3.6.1 Petri mreže

Glavna ideja transformacija mreža je njihova modifikacija bazirana na slučajevima gde svaka upotreba zadatog pravila znači jedan korak transformacije mreže. Poznati sistem tokena u slučaju Petri mreža ne menja strukturu mreže, ovaj sistem važi za sistem dinamičke promene Petri mreža koji se zasniva na pravilima. Petri mreže predstavljaju bipartitni graf i sva pravila transformacije grafika se mogu primeniti za definisanje promena ovih mreža (Ehrig i ostali, 2006; Ehrig i ostali, 1999; Ehrig i Padberg, 2004; Ehrig i ostali, 1999*; Rozenberg, 1997).

Petri mreže predstavljaju jedan od načina modelovanja različitih vrsta sistema (Murata, 1989; Huang i ostali, 2006) i tokova čiji početak razvoja se vezuje za 1960-e godine, tačnije 1962. godinu kada su razvijene od strane Carl Adam Petri-a kao alat

za model i analizu procesa. Značajan napredak u istraživanjima vezanim za ove mreže postignut je tokom 1980-ih godina, a i dalje se aktivno izučavaju (Murata, 1989; Salimifard i Wright, 2001). Predstavljaju idealno rešenje za definisanje i analizu kompleksnih procesa. Značajna prednost ovog sistema leži u činjenici da isti omogućava grafičku reprezentaciju različitih procesa. Pored toga što Petri mreže predstavljaju oblik grafa, poseduju jaku matematičku osnovu (Salimifard i Wright, 2001). Zahvaljujući formalizovanoj osnovi moguće je izvoditi jasne zaključke o svojstvima modelovanih procesa (van der Aalst, 2000). Mogu biti u potpunosti primenjene na različite dinamičke sisteme bazirane na događajima koje karakterišu paralelizam i sinhronizacija (Dicesare i ostali, 1993; van Hee, 1994; Chen, 2010).

Petri mreže se sastoje od mesta ili pozicija (engl. place) i tranzicija (engl. transition). Početno stanje Petri mreže naziva se inicijalno stanje i obeležava se sa M_0 . Pozicije se označavaju simbolom kruga, dok se tranzicije označavaju simbolom pravougaonika. Slika 3.11 daje prikaz jednog primera Petri mreže (Murata, 1989; van der Aalst, 2000, van der Aalst, 1998; Bause i Kritzinger, 2002; Piedrafita i ostali, 2010).



Slika 3.11: Primer proste Petri mreže

Za Petri mrežu sa slike 3.11 važi (Murata, 1989; Cicirelli i ostali, 2010; Mireles i ostali, 2006; Bause i Kritzinger, 2002; Sun i Jiang, 2009):

$$P = \{1, 2, 3, 4, 5, 6\}$$

$$\begin{aligned}
T &= \{t1, t2, t3, t4\} \\
F &\subseteq (P \times T) \cup (T \times P) \\
M_0: P &\rightarrow \{0, 1, 2, 3, \dots\} \\
P \cap T &= \emptyset \text{ i } P \cup T = \emptyset
\end{aligned}$$

- P predstavlja skup mesta p (engl. Place). Pozicije p poseduju parametar k koji označava kapacitet mesta.
- T predstavlja skup tranzicija t (engl. Transition). Tranzicije poseduju parametre π i w koji predstavljaju prioritet i težinu tranzicije, respektivno.
- M_q predstavljaju markere stanja pozicija kroz tranzicije.
- Token se obeležava crnim krugom unutar mesta/pozicije p.

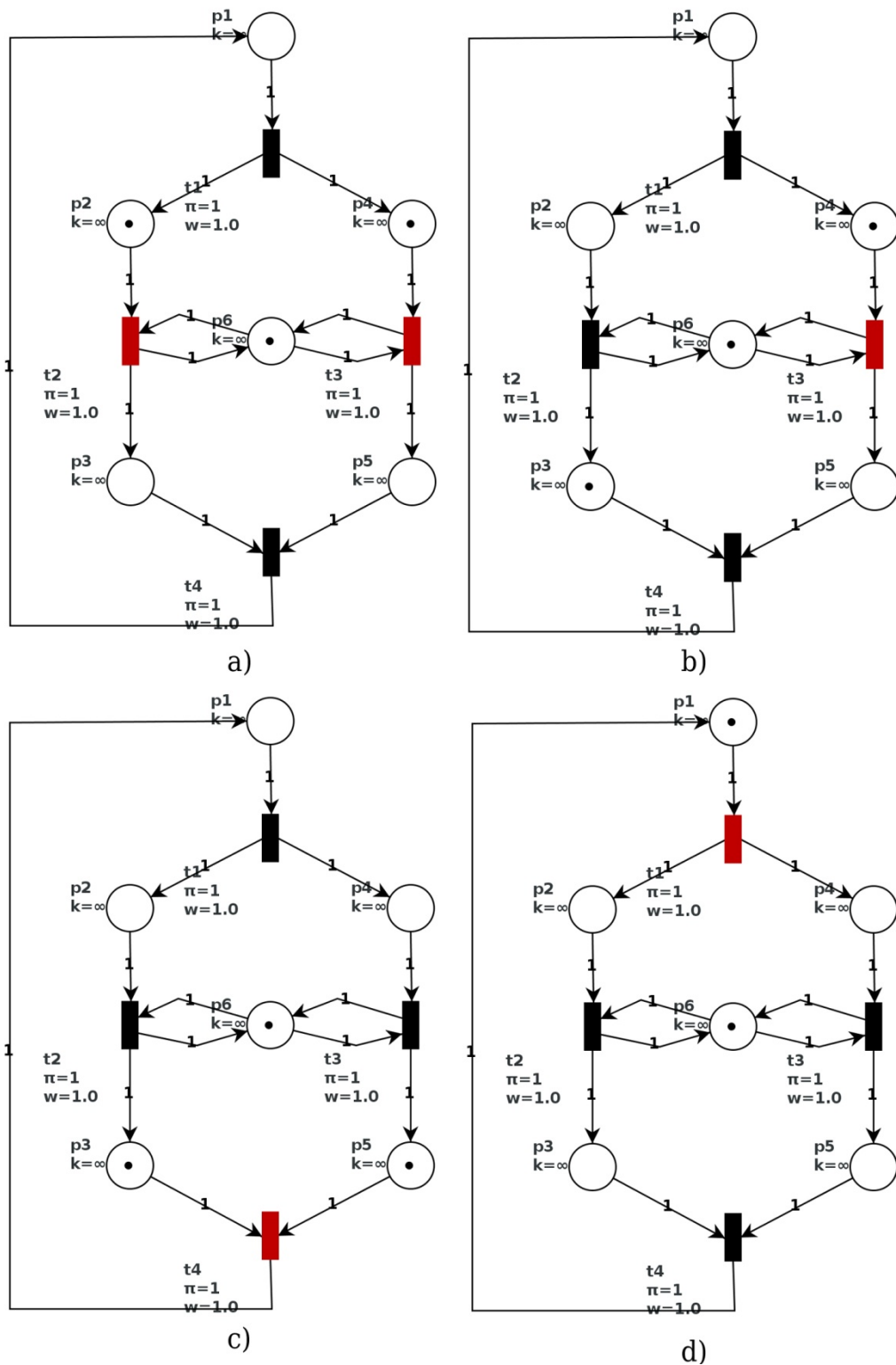
Mesta i tranzicije su povezani direktnim lukom koji ima svoju težinu w. Na slici 3.11 se vidi veza između mesta i tranzicija u obliku strelica. Postoji dve vrste veza između mesta i tranzicija:

- veze koje idu od mesta do tranzicije,
- veze koje idu od tranzicije do mesta.

Veze od mesta do mesta, odn. tranzicije do tranzicije nisu moguće. Svako mesto može se povezati samo sa tranzicijom i obrnuto. Na osnovu smera veze (strelice/luka) može se zaključiti koja pozicija je *ulazna* (engl. *input place*) za odgovarajuću tranziciju a koja *izlazna* (engl. *output place*). Tranzicija bez ulaznih pozicija naziva se izvorna tranzicija, a tranzicija bez izlaznih pozicija završna ili odvodna tranzicija. Izvorna tranzicija je uvek aktivna, dok izvršavanje završne tranzicije troši tokene ali ih ne proizvodi (Murata, 1989; Mireles i ostali, 2006). Ulazne (U) i izlazne (I) pozicije Petri mreže sa slike 3.11 su:

$$I = \begin{bmatrix} 0 & t1 & t2 & t3 & t4 \\ p1 & 0 & 0 & 0 & 1 \\ p2 & 1 & 0 & 0 & 0 \\ p3 & 0 & 1 & 0 & 0 \\ p4 & 1 & 0 & 0 & 0 \\ p5 & 0 & 0 & 1 & 0 \\ p6 & 0 & 1 & 1 & 0 \end{bmatrix} \quad U = \begin{bmatrix} 0 & t1 & t2 & t3 & t4 \\ p1 & 1 & 0 & 0 & 0 \\ p2 & 0 & 1 & 0 & 0 \\ p3 & 0 & 0 & 0 & 1 \\ p4 & 0 & 0 & 1 & 0 \\ p5 & 0 & 0 & 0 & 1 \\ p6 & 0 & 1 & 1 & 0 \end{bmatrix}$$

Stanje Petri mreže određuje distribucija tokena po pozicijama. Početno stanje Petri mreže sa slike 3.11 može se opisati vektorom (1,0,0,0,0,1) što znači da se po jedan token nalazi u pozicijama 1 i 6 (ilustrovano). Prva tranzicija koja se aktivira je t1. Tranzicija postaje dostupna (omogućena za aktivaciju) kada svaka od njenih ulaznih pozicija ima bar po jedan token u sebi. Za tranziciju t1 taj uslov je ispunjen na početku. Nakon izvršenja tranzicije t1 postoje dva puta: preko pozicije p2 i pozicije p4, a stanje petri mreže se može opisati vektorom (0,1,0,1,0,1) što znači da se nakon izvršenja tranzicije t1 tokeni nalaze u pozicijama p2, p4 i p6 (slika 3.12a). Sada su za izvršavanje omogućene tranzicije t2 i t3. U slučaju pozicije p2, naredna tranzicija koja će se izvršiti je tranzicija t2.



Slika 3.12: Stanja Petri mreže

Ulazne pozicije tranzicije t2 su pozicije p2 i p6. Nakon izvršenja tranzicije t2 stanje Petri mreže se može opisati vektorom $(0,0,1,1,0,1)$ kao što je prikazano na slici 3.12b.

Jedan token je prebačen u poziciju p3, jedan u p6 iz koje je u isto vreme i izbačen jedan token koji nije prenet dalje zbog kapaciteta veze t2-p3. Token unutar

pozicije p3 ne može nastaviti put dok se ne omogući izvršenje tranzicije t4. Izvršenje tranzicije t4 će biti omogućeno izvršenjem tranzicije t3 koja se u ovom trenutku jedino može izvršiti (slika 3.12b). Nakon izvršenja tranzicije t3 stanje mreže se može opisati vektorom (0,0,1,0,1,1) i jedina preostala tranzicija koja može da se aktivira je tranzicija t4 (slika 3.12c). Ulazne pozicije tranzicije t4 su pozicije p3 i p5 koje sada poseduju tokene što je i uslov za aktivaciju tranzicije t4. Nakon izvršenja tranzicije t4 mreža se vraća u inicijalno stanje (M0 sa slike 3.11) opisano vektorom (1,0,0,0,0,1).

Za Petri mrežu se kaže da je čista (eng. pure) ako nema povratnih petlji. Povratna petlja predstavlja par pozicije p i tranzicije t pri čemu je pozicija p u isto vreme i ulazna i izlazna pozicija za tranziciju t. Petri mreža je obična (engl. ordinary) ako je težina w svih njenih lukova jednaka 1. Svaka pozicija Petri mreže može da sadrži ograničen ili neograničen broj tokena. Petri mreže sa pozicijama koje mogu da sadrže neograničen broj tokena nazivaju se mreže beskonačnog kapaciteta (engl. infinite capacity net), dok se mreže čije pozicije imaju ograničen kapacitet k nazivaju mreže konačnog (ograničenog) kapaciteta (engl. finite capacity net) (Murata, 1989).

Tranzicije se izvršavaju po posebno definisanim pravilima (Murata, 1998; Mireles i ostali, 2006):

- za tranziciju t se kaže da je aktivna ako svaka ulazna pozicija p od t poseduje barem $w(p,t)$ tokena, gde je $w(p,t)$ predstavlja težinu luka od p do t.
- omogućena ili dostupna tranzicija može se ili ne mora izvršiti (u zavisnosti da li se događaj izvršava ili ne)
- izvršenje (engl. firing) aktivne tranzicije t uklanja $w(p,t)$ tokena iz svake ulazne pozicije p od t i dodaje $w(t,p)$ tokena u svaku izlaznu poziciju p od t gde je $w(t,p)$ predstavlja težinu luka od t do p.

Nakon izrade modela sistema upotrebom Petri mreža postavlja se pitanje šta raditi sa tako izrađenim modelom. Odgovor na to pitanje leži u najvećoj snazi Petri mreža a to je mogućnost analize mnogih svojstava i problema vezanih za sistem. Postoje dve vrste svojstava koja se mogu analizirati: svojstva zavisna od inicijalnog markera (stanja) i svojstva koja to nisu.

Svojstva Petri mreža

- Dostupnost (engl. Reachability) – predstavlja fundamentalnu osnovu za izučavanje dinamičkih svojstava sistema. Izvršavanjem bilo koje od aktivnih tranzicija promeniće raspored tokena (markiranje) mreže prema pravilima tranzicije. Za marker M_n kaže se da je dostupan sa markera M_0 ako postoji sekvenca izvršavanja tranzicija koji transformiše M_0 u M_n . Takva sekvenca transformacija označava se sa $\sigma = t_1 t_2 t_3 \dots t_n$. U ovom slučaju M_n je dostupno od M_0 po σ i to se može zapisati u obliku $M_0[\sigma > M_n$. Skup dostupnih markiranja od M_0 označava se sa $R(N, M_0)$ ili prostije $R(M_0)$. Skup svih sekvenci transformacija od M_0 u mreži (N, M_0) označava se sa $L(N, M_0)$ ili prostije $L(M_0)$.

- Ograničenost (engl. Boundedness) – Petri mreža je k-ograničena ili jednostavno ograničena ako broj tokena u svakoj poziciji ne prelazi krajnji broj k za bilo koji marker dostupan sa M_0 .
- Živost mreže (engl. Liveness) je usko povezana sa odsutnošću zastoja u operativnim sistemima. Mreža (N, M_0) smatra se živom ako uvek postoji barem jedna tranzicija koja se može izvršiti. To znači da živa mreža obezbeđuje izvršenje operacija bez zastoja.
- Reverzibilnost (engl. Reversibility) je osobina Petri mreže kod koje je M_0 dostupno sa M . Iz svakog stanja moguće se vratiti na M_0 .
- Istrajnost (engl. Persistence) je osobina mreže kod koje kada postoje dve aktivne tranzicije, izvršenjem jedne od njih druga neće biti deaktivirana. Tranzicija koja je aktivna ostaje aktivna sve dok se ne izvrši.
- Sinhrona distanca označava nivo usklađenosti dve tranzicije t_i i t_j i odgovara razlici brojeva izvođenja tranzicija (Murata, 1989, Cicirelli i ostali, 2010, Bause i Kritzinger, 2002; Sun i Jiang, 2009).

3.6.1.1 Metode analize

U nastavku će biti prikazane neke metode analize Petri mreža. Prvi metod uključuje pregled svih dostupnih markera. Sa druge strane, tehnike matrica i redukcije su moćne ali primenjive samo na određene slučajeve Petri mreža.

Drvo pokrivenosti (dostupnosti) (engl. the Coverability tree) – za zadata mrežu (N, M_0) , iz markera M_0 može se dobiti onoliko novih markera koliko ima dostupnih tranzicija. Iz svakog narednog markera može se dobiti još novih markera pod uslovom da postoje aktivne tranzicije. Ovaj proces rezultuje reprezentacijom markera u obliku drveta (ranije prikazano na slici 3.11). Stavke drveta predstavljaju markere stanja počevši od M_0 ka njegovim naslednicima, dok svaki luk (grana) predstavlja izvršenje jedne tranzicije kojom se jedan marker transformiše u drugi (Murata, 1989; Bause i Kritzinger, 2002).

Matrica incidencije (engl. Incidence Matrix) – ponašanje mnogih sistema može se opisati različitim oblicima jednačina. Bilo bi izuzetno korisno kada bi se osobine svake Petri mreže mogle opisati jednačinom ali to često nije moguće. Rešavanje jednačina dobijenih analizom Petri mreža nekada je ograničeno zbog nedeterminističke prirode Petri mreža i zbog ograničenja da rezultati moraju biti pozitivni brojevi.

Matrica incidencije za Petri mrežu N sa n tranzicija i m pozicija bila bi matrica $A=[a_{ij}]$ dimenzija $n \times m$. Stavka matrice bi se mogla definisati kao:

$$a_{ij} = a_{ij}^+ - a_{ij}^-$$

gde $a_{ij}^+ = w(i, j)$ predstavlja težinu luka od tranzicije i do njene izlazne pozicije j , dok $a_{ij}^- = w(j, i)$ predstavlja težinu luka ka tranziciji i od njene ulazne pozicije j (Murata, 1989; Cicirelli i ostali, 2010; Mireles i ostali, 2006; Bause i Kritzinger, 2002).

Prema tome, matrica incidencije Petri mreže sa slika 3.11 i 3.12 je:

$$A = \begin{bmatrix} -1 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Takođe, mreža je ograničena, sigurna i nema zastoja. Jednačina stanja mreže je oblika (Murata, 1989):

$$M_k = M_{k-1} + A^T u_k \text{ za } k=1, 2, \dots$$

gde M_k predstavlja trenutni marker stanja mreže. M_{k-1} predstavlja stanje mreže pre izvršavanja tranzicije k . A^T predstavlja transponovanu matricu incidencije A . Vektor u_k služi za označavanje tranzicije. Dakle, marker M_1 mreže sa slika 3.11 i 3.12 se računa iz inicijalnog markera na sledeći način:

$$M_1 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} -1 & 1 & 0 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 1 & 0 & -1 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Na isti način računaju se i ostali markeri stanja:

$$M_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} -1 & 1 & 0 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 1 & 0 & -1 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} -1 & 1 & 0 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 1 & 0 & -1 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$M_4 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} -1 & 1 & 0 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 1 & 0 & -1 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

Mreža se vraća u inicijalno stanje izvršavanjem tranzicije t_4 :

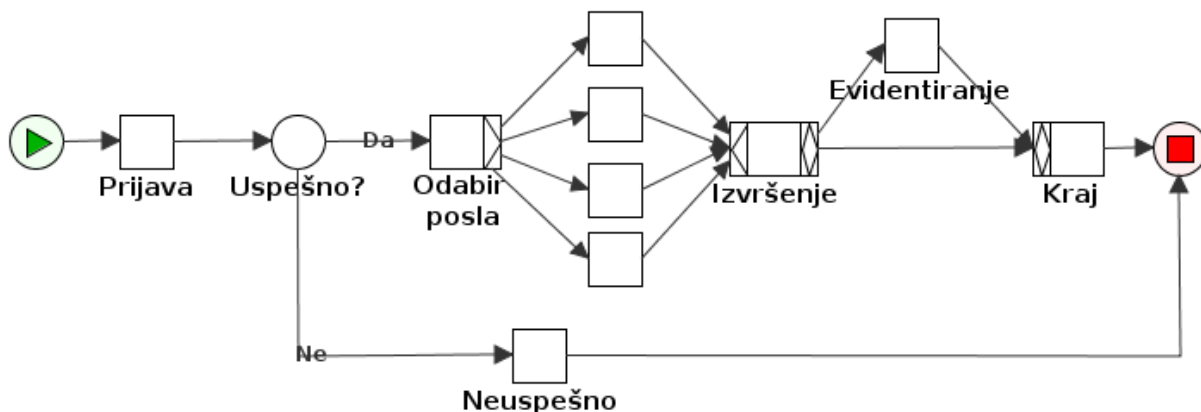
$$M_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} -1 & 1 & 0 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 1 & 0 & -1 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

3.6.2 YAWL

Reč YAWL predstavlja akronim od Yet Another Workflow Language i predstavlja jezik za modelovanje radnih tokova baziran na takozvanim šablonima radnih tokova. Jezik je razvijen u Holandiji na Univerzitetu za Tehnologiju, Ajndhoven (*Eindhoven*) i Univerzitetu za Tehnologiju, Kvinslend (*Quinsland*) (van der Aalst, 2005).

Glavni ciljevi ovog jezika su bili da podrži što više šablona radnih tokova (van der Aalst i ostali, 2003) i formalna semantika. Zasniva se na Petri mrežama visokog nivoa (Workflow Patterns, 2013), mrežama koje su proširene podrškom za boje, vreme i hijerarhiju, ali predstavlja potpuno nov jezik za nezavisnom semantikom (van der Aalst i ostali, 2004; van der Aalst i ostali, 2005).

Kao i Petri mreže, YAWL poseduje formalnu osnovu i grafičku reprezentaciju. Pored toga, YAWL specifikacija radnog toka može biti predstavljena i u XML formatu zbog toga što YAWL poseduje XML sintaksu i specificiran je po ugledu na XML šemu. Primer je dat na slici 3.13.



Slika 3.13: Primer YAWL toka

U pozadini toka sa slike 3.13 krije se XML kao što je prethodno naglašeno (nekompletno):

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <specificationSet
  xmlns="http://www.yawlfoundation.org/yawlschema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  version="2.2"
  xsi:schemaLocation="http://www.yawlfoundation.org/yawlschem
  a
  http://www.yawlfoundation.org/yawlschema/YAWL_Schema2.2.xsd
  ">
3   <specification uri="slika00011">
4     <metaData>
5       <creator>darko</creator>
6       <description>No description has been
  given.</description>
7     <version>0.2</version>
```

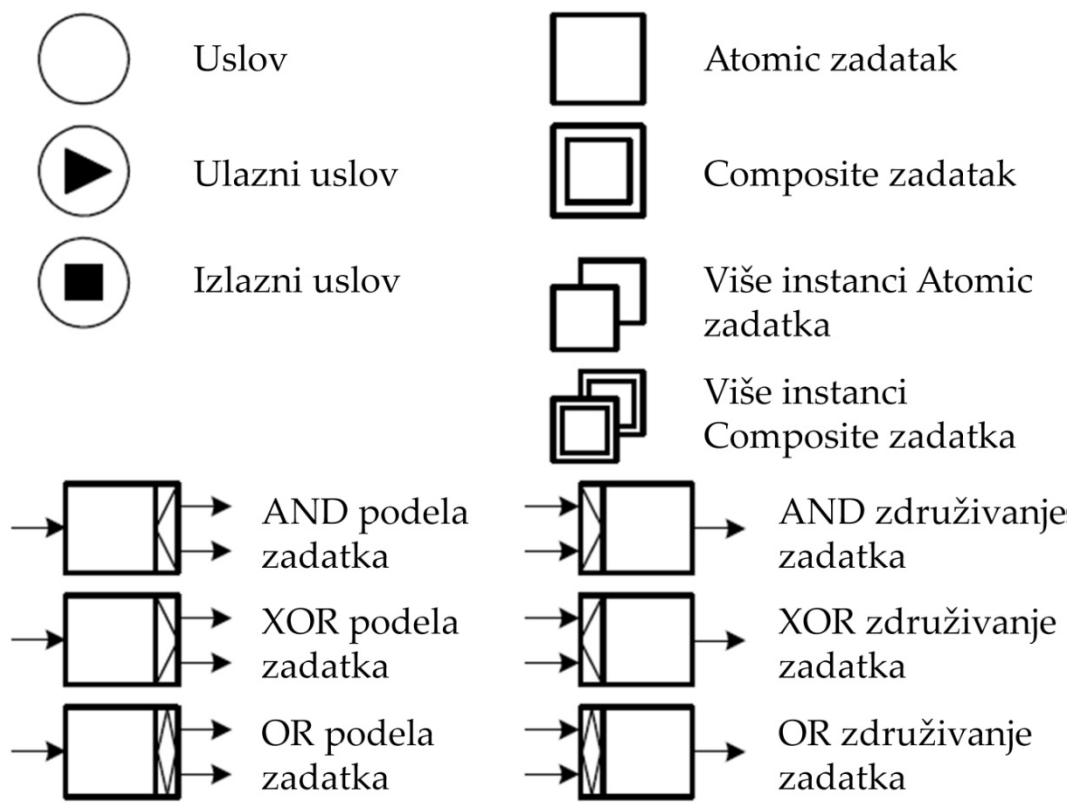
```

8     <persistent>>false</persistent>
9     <identifier>UID_3e5d12f1-0fa6-4dc7-8e53-
adabda4a4029</identifier>
10    </metaData>
11    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
/>
12    <decomposition id="New_Net_1" isRootNet="true"
xsi:type="NetFactsType">
13        <processControlElements>
14            <inputCondition id="InputCondition">
15                <flowsInto>
16                    <nextElementRef id="Prijava" />
17                </flowsInto>
18                .....
19                <flow source="Neuspešno" target="OutputCondition">
20                    <ports in="13" out="11" />
21                    <attributes>
22                        <lineStyle>11</lineStyle>
23                    </attributes>
24                </flow>
25            </net>
26        </specification>
27    </layout>
28 </specificationSet>

```

Da bi prevazišao nedostatke Petri mreža, YAWL je nadograđen osobinama koje olakšavaju implementaciju više instanci šablona, naprednom sinhronizacijom šablona i šablonima otkazivanja.

YAWL definicija procesa se sastoji od zadataka i uslova koje se mogu tumačiti kao tranzicije i pozicije u Petri mrežama respektivno. Zadaci mogu biti atomic ili composite tipa. Composite tip se obično koristi u nižim nivoima definicije procesa dok se atomic tip koristi u perifernim delovima. Svaka YAWL definicija procesa sadrži jedinstveni ulazni i jedan jedinstveni izlazni uslov kao početnu i krajnju tačku respektivno. Zadaci i uslovi su povezani direktnim lukom. Za razliku od Petri mreža, moguće je povezati dva zadatka direktno bez upotrebe uslova. Kod Petri mreža pozicija se može povezati sa drugom samo posredno preko tranzicije. Ovo na neki način pojednostavljuje definiciju izbegavanjem nepotrebnih objekata. Simboli korišćeni u YAWL definicijama procesa prikazani su na slici 3.14.



Slika 3.14: YAWL simboli

4 POVEZANA ISTRAŽIVANJA

Sørensen i ostali (2010) su kreirali konceptualni model informacionog sistema namenjen upravljanju poslovima farme i farmom generalno. Za potrebe kreacije pomenutog modela koristili su pored proizvoljnih šema koristili su i UML dijagrame za preciznu specifikaciju sistema koji je trebao da se razvije na osnovi modela. Zhu i Li (2013) su razvili web-bazirani informacioni sistem namenjen za podršku projektima izgradnje tunela. Takođe su koristili sve mogućnosti UML jezika. Informacioni sistem je podeljen na serverski i klijentski deo za šta su definisali i arhitekturu. Jedan ovakav sistem funkcioniše veoma slično kao i sistem koji bi bio razvijen na osnovu modela koji predlaže ova disertacija. Kwapisz (2009) je dao prikaz mogućnosti UML-a pri modelovanju složenih poslovnih procesa finansijskih sistema. Zaključio je da je UML moguće iskoristiti u ogromnom broju slučajeva iz oblasti softverskog inženjerstva. Posebno je akcentirao mogućnosti definisanja funkcionalnosti sistema u razvoju koji poseduju kompleksnu arhitekturu. Takođe je konstatovao visok nivo primenljivosti UML-a u projektovanju baza podataka. Nasuprot do sada pomenutim pristupima modelovanja uz pomoć UML-a, Pawlak je 1981. godine model informacionog sistema predstavio isključivo matematičkom notacijom što predstavlja još jedan od prihvaćenih načina modeliranja. Informacioni sistem je predstavljao u obliku niza elemenata. Elementi bi predstavljali delove samog informacionog sistema. Informacioni sistem je definisao kao:

$$S = \langle X, A, V, \rho \rangle$$

gde X predstavlja konačan niz objekata (ljudi, mašine i slično). A predstavlja konačan niz atributa dok je $V = \cup_{a \in A} V_a$, gde V_a predstavlja konačan niz vrednosti atributa a , pri čemu je $\#V > 1$. Za predstavljanje određenih osobina elementa koristi se funkcija $\rho: X \times A \rightarrow V$ takva da

$$\forall x \in X \wedge \forall a \in A: \rho(x, a) \in A$$

Primer jednog informacionog sistema mogao bi da bude:

$$\begin{aligned} X &= \{x_1, x_2, x_3, x_4, x_5\} \\ A &= \{pol, plata, godište\} \\ V &= \{V_{pol}, V_{plata}, V_{godište}\} \end{aligned}$$

gde je $V_{pol} = \{muško, žensko\}$, $V_{godište} = \{mlad, srednje, star\}$, $V_{plata} = \{niska, srednja, visoka\}$. Funkcija ρ bi prema ovom bila definisana sledećom tabelom

X	POL	PLATA	GODIŠTE
x_1	muški	visoka	srednje
x_2	ženski	visoka	srednje
x_3	muški	srednja	mlad
x_4	muški	niska	mlad
x_5	ženski	srednja	mlad

Za svako $x \in X$ možemo definisati funkciju $\rho: A \rightarrow V$ takvu da $\rho_x(a) = \rho(x, a)$ i nazvaćemo je informacija o x u S , tako da je, za primer,

$$\rho_{x_2} = \begin{matrix} POL & PLATA & GODIŠTE \\ \hline \text{ženski} & \text{visoka} & \text{srednje} \end{matrix}$$

Parovi vrednosti a i V_a mogu se opisati deskriptorom v , pri čemu $v \in V_a$. Pomenuti deskriptor bi imao vrednost (POL, ženski), (PLATA, visoka) ili (GODIŠTE, srednje). Deskriptori takođe mogu da se zapišu i u obliku (POL=ženski). Prema ovome, informacije o objektu x_2 se mogu zapisati kao $\{(POL=\text{ženski}), (PLATA=\text{visoka}) \text{ ili } (GODIŠTE=\text{srednje})\}$.

Prema Pawlak-u, svojstva informacionih sistema se mogu opisati funkcijom $\varphi: A \rightarrow V$ za svako a . Funkcija $\varphi(a) \in V_a$ biće nazvana informacija unutar sistema S . Postoji ukupno $\prod_{a \in A} \text{card}(V_a)$ različitih informacija u sistemu S . Shodno tome, prema prethodnom primeru, ukupan broj informacija u sistemu S (još označen i kao $\text{Inf}(S)$) bio bi

$$\text{Inf}(S) = \text{card}(V_{pol}) \cdot \text{card}(V_{plata}) \cdot \text{card}(V_{godiste}) = 2 \cdot 3 \cdot 3 = 18$$

Ovo je samo deo modela koji je Pawlak predstavio 1981. Ostatak modela dotiče zavisnost argumenata i podsisteme i neće biti prikazan ovde.

Pored pomenutih načina modelovanja informacionih sistema ili softvera uopšte postoji (između ostalih) još jedan koji je u širokoj upotrebi – Petri mreže. Yang i ostali su 2012. iskoristili Petri mreže (stohastički oblik) za pravljenje modela sigurnosti informacionih sistema. Aoumeur i Saake (2002) su iskoristili Petri mreže za modelovanje i validaciju kooperativnih informacionih sistema. Chema i ostali (2012) su ih takodje iskoristili za modeliranje web servisa. Pored navedenih primera upotrebe postoji još mnogo načina i mogućnosti upotrebe Petri mreža u softverskom inženjerstvu (Mireles i ostali, 2006; Chen, 2010; Salimifard i Wright, 2001; Yu i ostali, 2010; Rodriguez i ostali, 2010; Piedrafita i ostali, 2010; Flores i López, 2010; Miao i He, 2010; van der Aalst, 1996).

Pored prikaza modela različitih informacionih sistema moguće je i napraviti poređenje funkcija postojećih rešenja i rešenja razvijenog za potrebe ovog istraživanja. U grafičkim procesima postoji veliki broj informacionih sistema koji se bave različitim aspektima proizvodnje. Neki upravljaju samo jednim segmentom proizvodnje dok se neki bave sveobuhvatnim upravljanjem kompletnom proizvodnjom.

Kao što je već naglašeno, postoje apstraktni modeli i realni modeli. Pored prikazanih apstraktnih modela i načina modelovanja, u nastavku će biti prikazana poređenja prototipskog rešenja razvijenog prema modelu koji se u ovom radu predlaže. Pomenuto prototipsko rešenje nosiće naziv GRID Workflow Manager (skraćeno GWFM) i biće upoređeno sa prisutnim aktuelnim tržišnim rešenjima softvera sličnih ili istih karakteristika.

Za potrebe ovog istraživanja osmišljeno je 24 atributa koje jedno softversko rešenje navedenog tipa može da poseduje. Ti atributi su:

1. Upravljanje poslovima
2. Upravljanje dokumentima
3. JDF
4. Upravljanje sistemima
5. Upravljanje zaposlenima
6. Upravljanje klijentima
7. Upravljanje porudžbinama
8. Upravljanje plaćanjem
9. Upravljanje dostavom
10. Upravljanje skladištem
11. Upravljanje prodajom
12. Upotreba mrežnih resursa
13. Upravljanje tokovima
14. Baza podataka
15. Kalkulacije
16. Predviđanje troškova na bazi parametara
17. Predviđanje troškova na bazi akumuliranih podataka
18. Izveštaji
19. Desktop platforma
20. Prisustvo Mobilnih platformi
21. Prisustvo Mobilnih platformi u proizvodnji
22. Prisustvo Web platforme
23. Obaveštenja klijentima
24. Praćenje sistema u realnom vremenu

Rešenja koja su iskorišćena za upoređivanje su:

- A. RPM Elite (Brooks, 2013)
- B. Print Manager Plus (Print Manager Plus, 2013)
- C. EPRO (Print MIS, 2013)
- D. Digital Factory (CADLink, 2013)
- E. EFI Digital StoreFront (EFI, 2013)
- F. EFI Monarch (EFI, 2013)
- G. ESKO Suite 12 (ESKO, 2013)
- H. Fiery JobFlow (Fiery, 2013)
- I. FreeFlow Output Manager (Xerox FreeFlow, 2013)
- J. Heidelberg Business Manager (Heidelberg BM, 2013)
- K. PressWise Print MIS (Presswise, 2013)
- L. Konica Minolta Printgroove (Konica Minolta Printgroove, 2013)
- M. Canon PRISMA-production (Océ, 2013)
- N. KBA Complete Workflow Solution (KBA, 2013)

Tabela 4.1: Poređenje atributa sličnih rešenja (1. deo)

	RPM Elite	Print Manager Plus	EPRO	Digital Factory	EFI Digital StoreFront
Upravljanje poslovima	x		x	x	x
Upravljanje dokumentima			x	x	x
JDF					x
Upravljanje sistemima		x	x	x	
Upravljanje zaposlenima		x			x
Upravljanje klijentima			x		
Upravljanje porudžbinama					
Upravljanje plaćanjem					x
Upravljanje dostavom			x		
Upravljanje skladištem			x		
Upravljanje prodajom					
Upotreba mrežnih resursa				x	x
Upravljanje tokovima		x			x
Baza podataka	x		x	x	x
Kalkulacije			x		
Predviđanje troškova na bazi parametara			x		
Predviđanje troškova na bazi akumuliranih podataka					
Izveštaji			x		x
Desktop platforma	x	x	x	x	x
Prisustvo Mobilnih platformi					
Prisustvo Mobilnih platformi u proizvodnji					
Prisustvo Web platforme					
Obaveštenja klijentima					
Praćenje sistema u realnom vremenu					

Poređenje atributa pomenutih rešenja prikazano je u tabelama 4.1, 4.2 i 4.3. Veliki broj informacija nije mogao biti prikazan u okviru jedne tabele te je podeljen u tri. Sve tri tabele zajedno čine jednu veliku tabelu. Skupovi atributa softverskih rešenja su označeni slovima A do N (čime su označena i sama rešenja) dok su sami atributi pojedinačno označeni brojevima od 1 do 24. Skup atributa GWFm će biti označen slovom O .

Ako je skup svih softverskih rešenja označen sa Δ , prema tabelama 4.1 do 4.3 može se reći da postoje atributi softvera za koje je

$$\forall x \in \{A, B, \dots, N\} \wedge \forall y \in \{A, B, \dots, N\}: \Delta_x \cap \Delta_y = \emptyset \text{ i } \Delta_x \cap \Delta_y \neq \emptyset$$

pri čemu važi da je $x \neq y$. Postoje softverska rešenja kod kojih je $card(\Delta_x \cap \Delta_y)$ veći nego kod drugih softverskih rešenja. Na osnovu toga može se uvesti koeficijent sličnosti rešenja ε_{xy} po definisanim atributima:

$$\varepsilon_{xy} = card(\Delta_x \cap \Delta_y)$$

za svako $x \in \{A..N\} \wedge y \in \{A..N\} \wedge x \neq y$ i definisan i ograničen niz atributa. Prema ovome ako je $\varepsilon_{xy} = 0$, softversko rešenje Δ_x ne poseduje sličnosti sa rešenjem Δ_y .

Tabela 4.2: Poređenje atributa sličnih rešenja (2. deo)

	EFI Monarch	ESKO Suite 12	Fiery JobFlow	FreeFlow Output Manager	Heidelberg Business Manager
Upravljanje poslovima	x	x	x	x	x
Upravljanje dokumentima	x	x		x	x
JDF	x			x	x
Upravljanje sistemima		x		x	x
Upravljanje zaposlenima	x				
Upravljanje klijentima	x				x
Upravljanje porudžbinama					x
Upravljanje plaćanjem	x				x
Upravljanje dostavom	x				x
Upravljanje skladištem	x				x
Upravljanje prodajom	x				x
Upotreba mrežnih resursa	x	x	x	x	x
Upravljanje tokovima	x	x			
Baza podataka	x	x	x	x	x
Kalkulacije					x
Predviđanje troškova na bazi parametara	x				x
Predviđanje troškova na bazi akumuliranih podataka					
Izveštaji	x				x
Desktop platforma	x	x	x	x	x
Prisustvo Mobilnih platformi	x				
Prisustvo Mobilnih platformi u proizvodnji					
Prisustvo Web platforme	x	x			
Obaveštenja klijentima			x		
Praćenje sistema u realnom vremenu	x	x	x		x

Ako je $\varepsilon_{xy} > 0$, tada softverska rešenja Δ_x i Δ_y poseduju određeni nivo sličnosti. Što je koeficijent ε_{xy} veći to je i broj zajedničkih atributa veći.

Tabela 4.3: Poređenje atributa sličnih rešenja (3. deo)

	PressWise Print MIS	Konica Minolta Printgroove	Canon PRISMA- production	KBA Complete Workflow Solution	GWFM
Upravljanje poslovima	x	x	x	x	x
Upravljanje dokumentima	x	x	x	x	x
JDF	x	x		x	x
Upravljanje sistemima	x	x	x	x	
Upravljanje zaposlenima				x	x
Upravljanje klijentima	x		x	x	x
Upravljanje porudžbinama	x	x	x	x	
Upravljanje plaćanjem		x	x		
Upravljanje dostavom	x	x		x	
Upravljanje skladištem	x			x	
Upravljanje prodajom		x		x	
Upotreba mrežnih resursa			x	x	x
Upravljanje tokovima			x	x	x
Baza podataka	x	x	x	x	x
Kalkulacije				x	
Predviđanje troškova na bazi parametara	x	x		x	
Predviđanje troškova na bazi akumuliranih podataka				x	x
Izveštaji				x	x
Desktop platforma	x	x	x	x	x
Prisustvo Mobilnih platformi				x	x
Prisustvo Mobilnih platformi u proizvodnji					x
Prisustvo Web platforme		x	x	x	x
Obaveštenja klijentima			x		x
Praćenje sistema u realnom vremenu		x		x	x

Za softverska rešenja za koja važi

$$card(\Delta_x) = card(\Delta_y) \wedge \varepsilon_{xy} = card(\Delta_x)$$

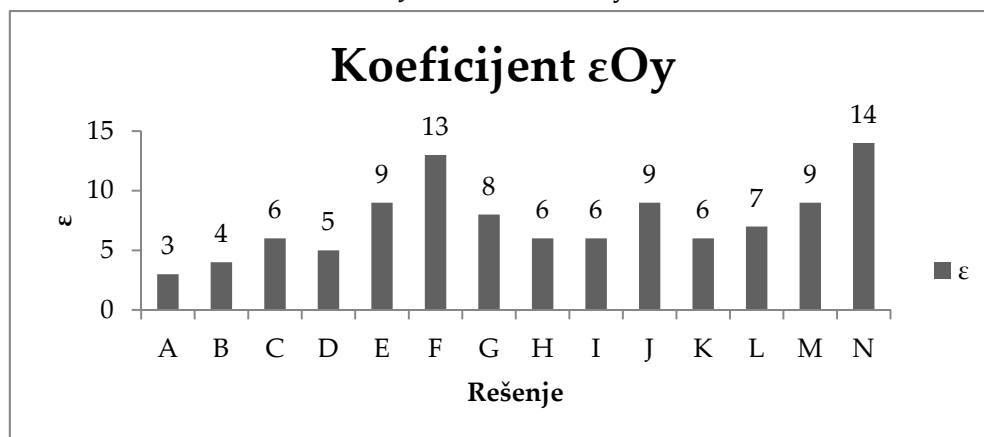
kažemo da su ista po svojim atributima.

Ukoliko je, za primer, $card(\Delta_x) > card(\Delta_y) \wedge \varepsilon_{xy} = card(\Delta_y)$ možemo reći da sve attribute rešenja Y poseduje rešenje X, ali dva rešenja nisu identična po specificiranim atributima. Maksimalna moguća vrednost ε_{xy} bi bila:

$$\varepsilon_{xy(maks)} = \min\{card(\Delta_x), card(\Delta_y)\}$$

Koeficijent sličnosti softverskih rešenja i rešenja O iz tabela 4.1 do 4.3 prikazan je na slici 4.1. Za svako $y \in \{A, B, \dots, N\}$ koeficijent je računat kao

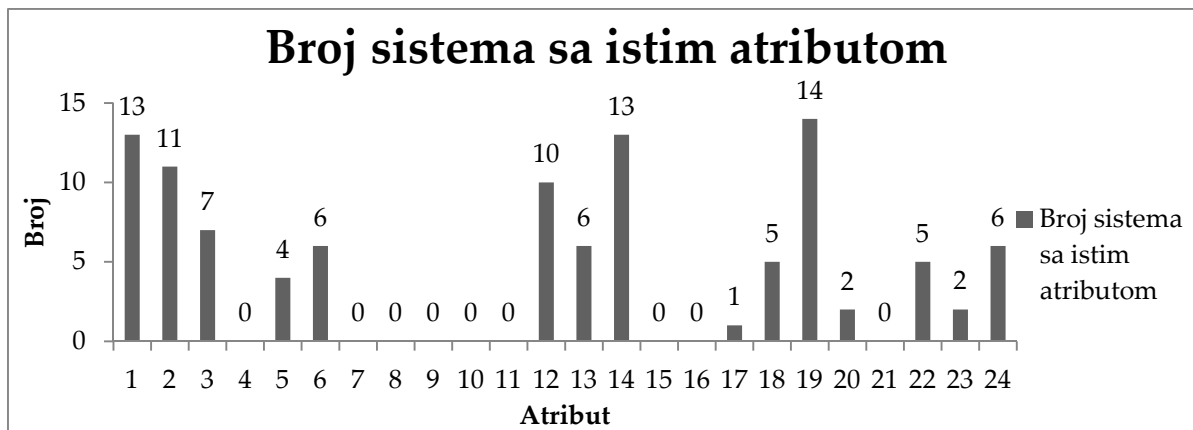
$$\varepsilon_{Ny} = \text{card}(O \cap \Delta_y)$$



Slika 4.1: Koeficijent ε za predložena rešenja

Sa slike 4.1 se može zaključiti da je najslićnije (prema zadatim atributima) softversko rešenje rešenju koje se daje ovim radom rešenje N. Najmanje slićnih elemenata prisutno je kod rešenja A i B. Uzevši u obzir vrednosti $\text{card}(O)$ i $\overline{\varepsilon_{Oy}}$ može se izvesti zaključak da većina rešenja skupa Δ nema velikih slićnosti prema zadatim atributima sa rešenjem O. Naravno, svi kriterijumi nisu podjednako bitni i ne nose u sebi podjednak nivo kompleksnosti tako da ovo poređenje ne dokazuje da je jedno rešenje po nečemu bolje od drugog.

Na slici 4.2 prikazan je broj istih atributa rešenja O i ostalih rešenja. Primetno je da rešenje O sa većinom rešenja deli attribute 1, 2, 14 i 19. Atributi 1 i 2 odnose se na upravljanje tokovima i dokumentima što je primarni zadatak rešenja O. Atributi 14 i 19 označavaju upotrebu baza podataka i desktop platformu. Pomenuta četiri atributa su danas široko prisutna kod softverskih rešenja ovakvog tipa. Takođe, rešenje O poseduje mogućnost upotrebe mrežnih resursa i tu osobinu deli sa još 10 rešenja. Funkciju upravljanja tokovima deli sa 6 rešenja. Sa dovoljnim brojem drugih rešenja deli i attribute koji se tiču upravljanja klijentima, generisanje izveštaja, prisutnost web platforme i praćenje sistema. Vrednosti 0 koje se vide na slici 4.2 označavaju situaciju kada rešenje O ne poseduje odgovarajući atribut ili isti ne deli ni sa jednim rešenjem. Upravo u toj grupi se kriju atributi po kojima se rešenje O izdvaja od drugih.

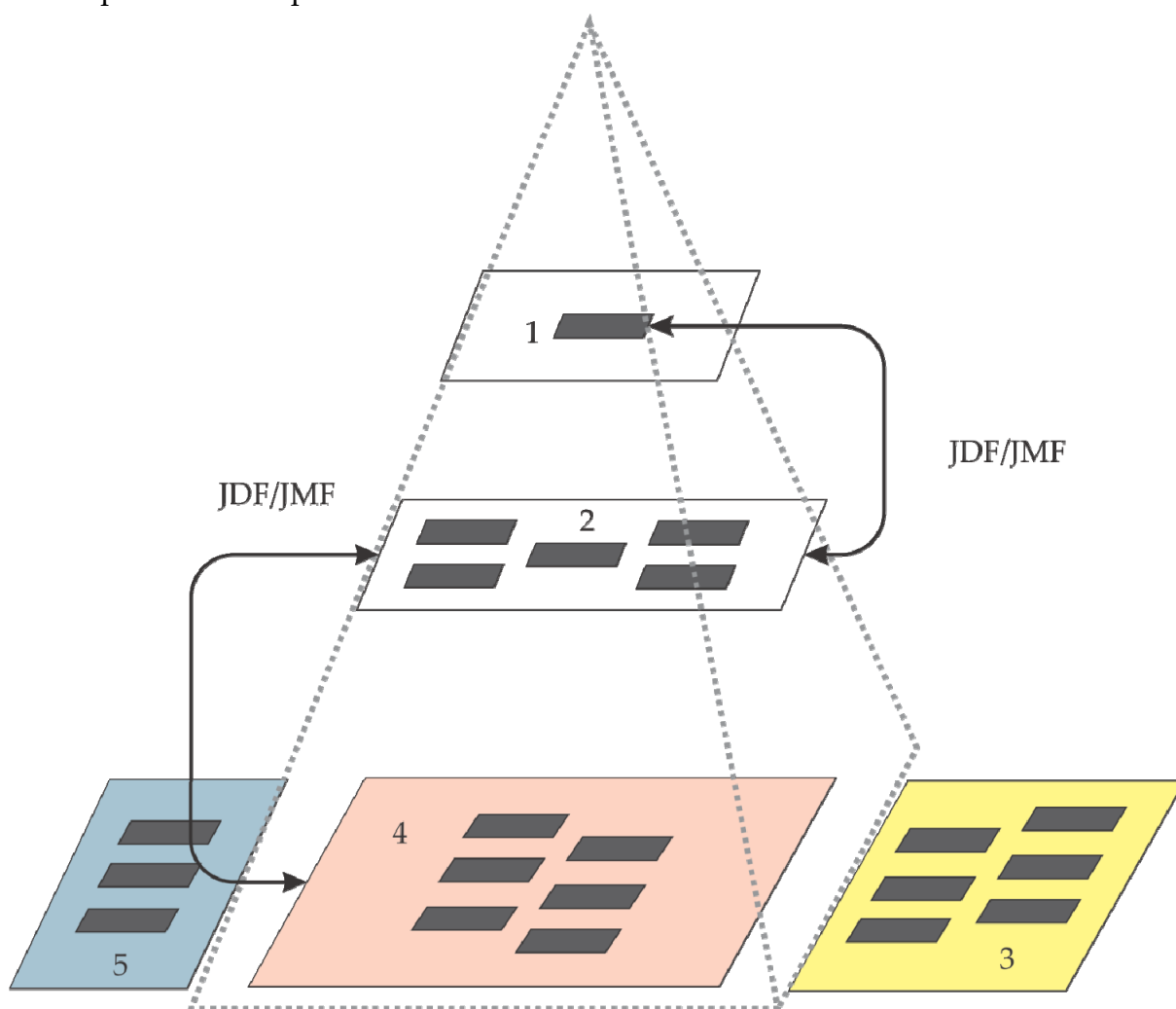


Slika 4.2: Broj istih rešenja

Oznakom O' biće obeležen skup atributa koje rešenje O ne deli ni sa jednim rešenjem.

$$O' = \{21\}$$

Skupu O' bi se mogao pridružiti i atributi 20 (iz razloga što je prisutan kod svega još dva rešenja) i 17 (prisutan je samo još kod jednog rešenja). Ono po čemu se rešenje O izdvaja od ostalih jeste upotreba mobilnih tehnologija i bežičnih komunikacija u proizvodnji. Takav način funkcionisanja do sada nije primenjen. Ogleda se u upotrebi mobilnih uređaja direktno od strane operatera. Mobilni uređaji su iskorišćeni u ovoj situaciji iz razloga što za funkcionisanje sistema O nije potrebna desktop hardverska podrška.

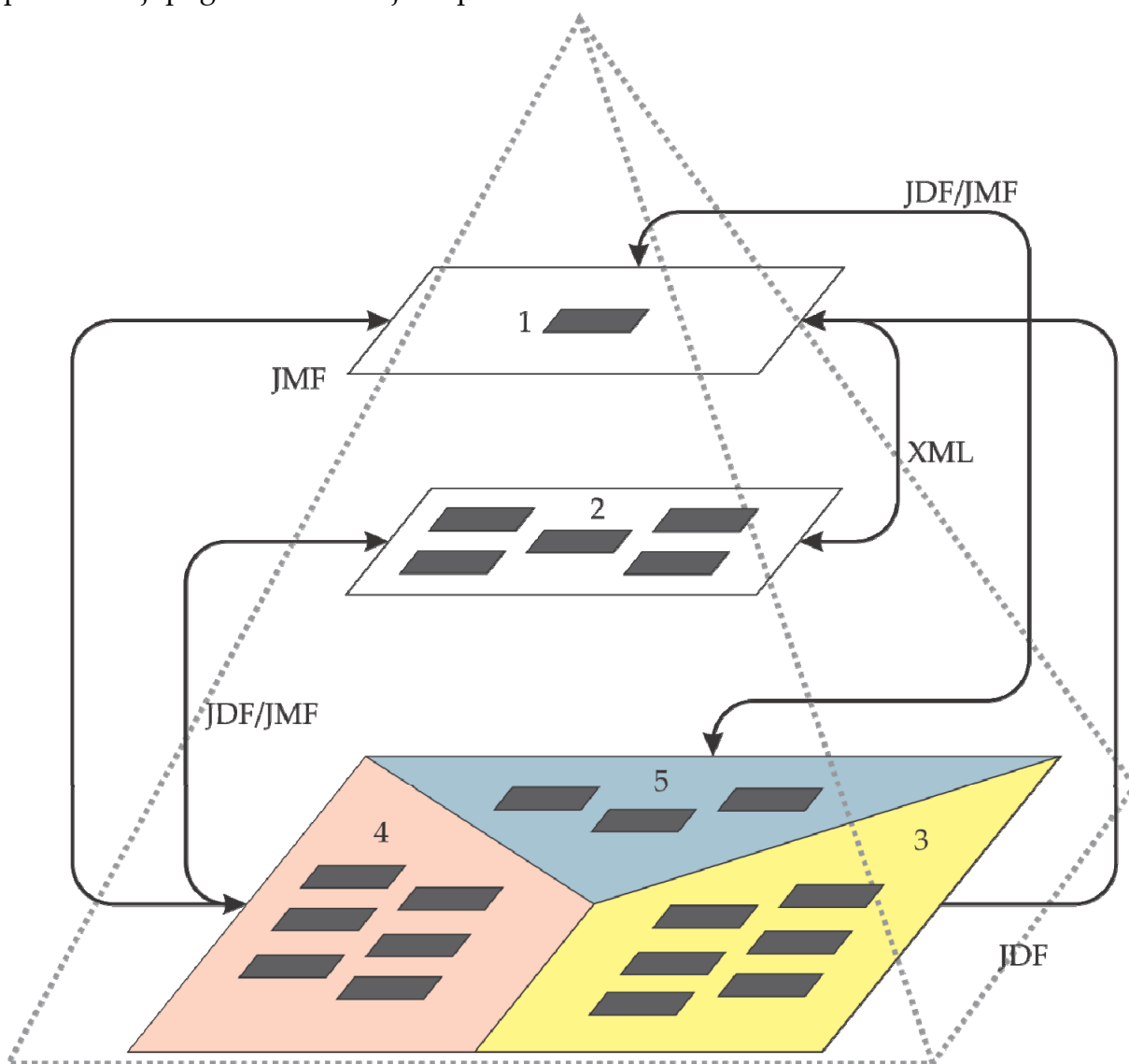


Slika 4.3: KBA Complete Workflow Solutions koncept

Drugi atribut po kome se rešenje O izdvaja od ostalih je predviđanje troškova i smanjenje vremena kalkulacija troškova proizvoda na osnovu akumuliranih podataka. Predviđanje se vrši grupisanjem proizvoda u smislene grupe za koje se posle akumulacije podataka o vremenu izrade proizvoda mogu izvući zaključci o potrebnom vremenu izrade. Na osnovu pomenutih akumuliranih podataka mogu se izvući i zaključci o efikasnosti sistema i pojedinaca učesnika procesa proizvodnje. Ovaj sistem pogodan je i primenljiv za polu automatizovane ili neautomatizovane

sisteme. Visoko automatizovani sistemi ove operacije vrše samostalno sa ograničenjima proizvođača. Softver koji prati mašine često nema pristup podacima mašina drugih proizvođača.

Pošto je rešenje N prema navedenim atributima najbližije rešenju O, biće izvršeno posebno poređenje pomenutih rešenja. Slika 4.3 daje prikaz koncepta funkcije rešenja N. Stavka 1 predstavlja MIS, stavka 2 *gateway* servise dok stavke 3, 4 i 5 predstavljaju faze proizvodnog procesa. Na pomenutoj slici se vidi visok nivo automatizacije, standardizovan način komunikacije ali kontrola samo jedne od faza grafičkog proizvodnog procesa. Cilj je potpuna kontrola procesa proizvodnje. Kontrolni sistemi u ovom slučaju nemaju potpunu sliku trenutnih dešavanja u proizvodnji pogotovo u slučaju nepodržanih tehnoloških sistema.



Slika 4.4: GWFM koncept

Slika 4.4 daje koncept funkcionisanja GWFM rešenja. Na slici se vidi da se uz pomoć softverskog rešenja napravljenog prema modelu koji se ovde predlaže potencijalno može kontrolisati kompletan sistem proizvodnje. Stavka 1 predstavlja

GWFM dok su u stavku 2 uključeni ostali softverski sistemi. GWFM sa nivoa 1 može da komunicira i sa njima a i da prihvata informacije direktno od tehnoloških sistema sa nivoa 3. Time se stiče uvid u trenutno stanje kompletne proizvodnje.

5 MODEL INFORMACIONOG SISTEMA

Model sistema koji predlaže ovo istraživanje biće predstavljen korišćenjem aktuelnih jezika i tehnika modeliranja. Informacioni sistem razvijen za potrebe ovog istraživanja posedovaće sledeće attribute (definisane u poglavlju 4):

- Upravljanje poslovima
- Upravljanje dokumentima
- Upravljanje zaposlenima
- Upravljanje klijentima
- Upotreba mrežnih resursa
- Upravljanje tokovima
- Baza podataka
- Predviđanje troškova na bazi akumuliranih podataka
- Izveštaji
- Desktop platforma
- Prisustvo Mobilnih platformi
- Prisustvo Mobilnih platformi u proizvodnji
- Prisustvo Web platforme
- Obaveštenja klijentima
- Praćenje sistema u realnom vremenu

5.1 Definicija

GWFM informacioni sistem uopšteno možemo definisati kao

$$S = \langle K, E, A, V, F \rangle$$

gde K predstavlja skup komponenata sistema, dok E predstavlja skup elemenata komponente. Skup A predstavlja skup atributa elementa komponente čije vrednosti se sadrže u skupu $V = \bigcup_{a \in A} V_a$. Skup F predstavlja skup funkcija sistema S . Funkcija koja bi bila zadužena za prikupljanju informacija o objektu skupa E mogla bi biti definisana kao $\rho: E \times A \rightarrow V$ takva da $\forall e \in E \wedge \forall a \in A: \rho_e(a) \in V$.

Skup komponenata sistema možemo definisati kao

$$K = \left\{ \begin{array}{l} \text{Poslovi, Sistemi, Korisnici, Kontrolna tabla,} \\ \text{Message komponenta, Image manipulation,} \\ \text{File management, Generator grafova} \end{array} \right\}$$

gde je $\#K = \text{const}$ i $k \in K$. Svako k predstavlja jednu komponentu sistema.

Elementi skupa E predstavljaju objekte ili elemente svake od komponenata sistema. Skup E_k predstavlja skup elemenata komponente k . Za primer, E_{Poslovi} predstavlja skup elemenata komponente *Poslovi*. Atributi i vrednosti atributa biće navedeni i objašnjeni u narednim delovima ovog poglavlja.

Tabela 5.1: Funkcije GWFM sistema dostupne korisnicima

Br.	Elementi komponente	Delovi
1	$E_{Poslovi}$	Pregled Arhiva Unos Izmena Grupe poslova Praćenje poslova Klijenti
2	$E_{GrafickiSistemi}$	Pregled Unos Izmena Pozicije Prostorije
3	$E_{Korisnici}$	Lista Unos Izmena
4	$E_{KomandnaTabla}$	Stanje Podaci Log Pretraga

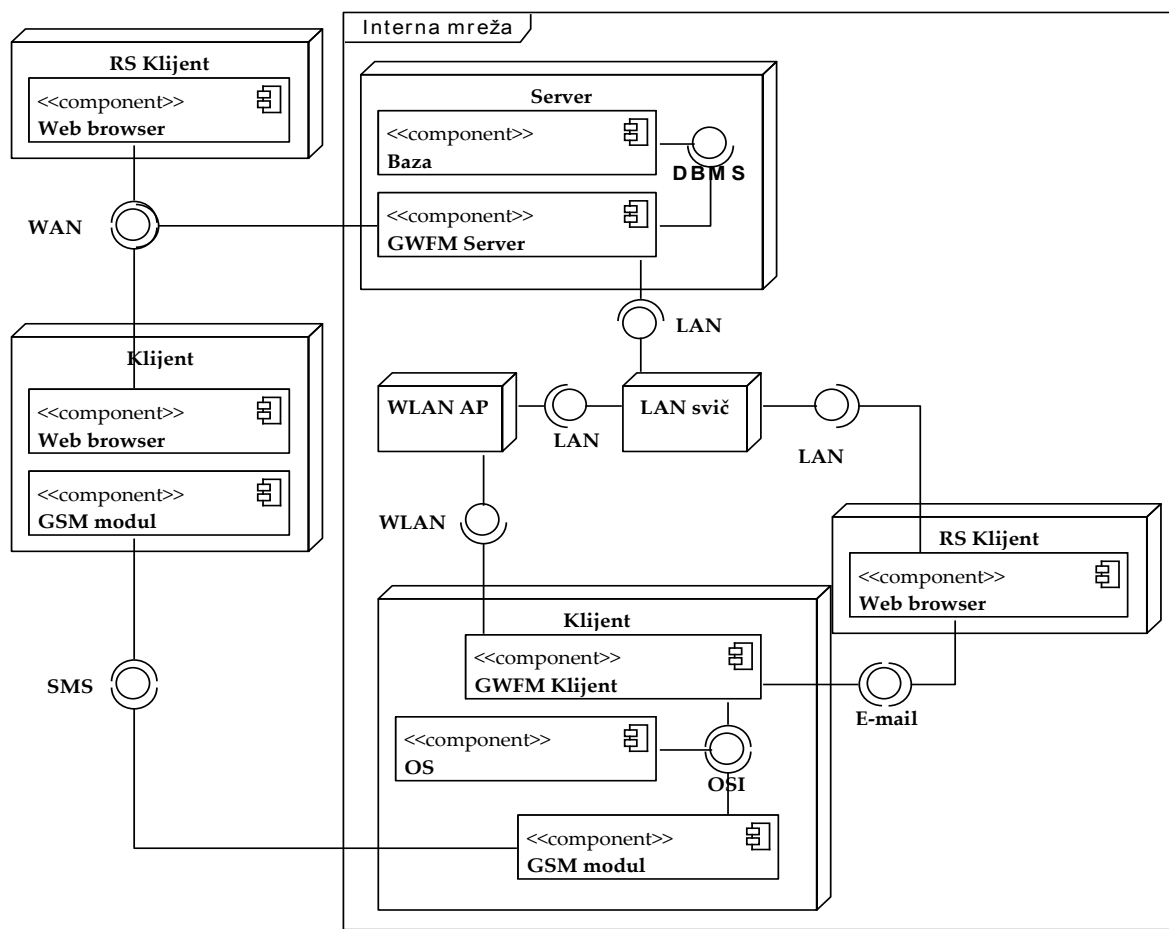
Funkcije komponentata, kao što je već navedeno, opisane su skupom F . F_k predstavlja skup funkcija pojedine komponente gde k predstavlja oznaku komponente i $k \in K$. Funkcije pojedinih komponentata GWFM sistema prikazane su tabelom 5.1.

Komponente *Message komponenta*, *Image manipulation*, *File management* i *Generator grafova* su interne komponente sistema i korišćene su isključivo od strane drugih komponentata. Korisnici sistema nemaju direktan pristup ovim komponentama. Pomenute komponente će biti naknadno objašnjene.

5.2 Infrastruktura i arhitektura

Infrastruktura GWFM sistema predstavljena je na slici 5.1. GWFM sistem se sastoji iz 2 dela:

- GWFM servera i
- GWFM klijenta



Slika 5.1: Infrastruktura GWFM sistema

Infrastruktura sistema se sastoji jednog servera, jednog ili više klijenata, uređaja za komunikaciju i uređaja okruženja. Potrebne komponente servera su GWFM server komponenta i komponenta baze podataka. Dve komponente komuniciraju putem interfejsa za upravljanje bazom podataka.

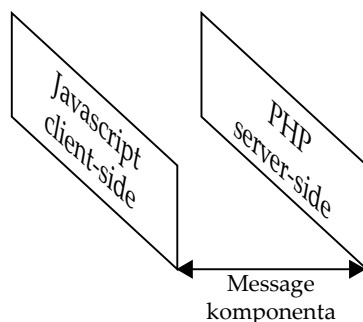
Pored serverske komponente prisutna je i klijentska komponenta. Pomenuta komponenta u sebi sadrži GWFM klijent komponentu koja putem interfejsa, podržanih od strane operativnog sistema uređaja u sklopu kojeg radi, komunicira sa serverskom komponentom i ostalim delovima infrastrukture sistema. Putem interne mreže klijentski modul komunicira sa serverom prvenstveno. U slučaju da hardver klijent uređaja podržava 3G komunikacije, klijentski modul može uspostaviti i komunikaciju sa drugim uređajima putem GSM protokola.

Osnovu mrežne komunikacije čini jedan ili više LAN switch uređaja koji su povezani sa jednim ili više WLAN AP uređaja. Svi WLAN uređaji moraju funkcionisati u AP modu. ROUTER mode nije izvodljiv i ne zadovoljava potrebe komunikacija sistema. Brzine neophodne za nesmetanu komunikaciju delova sistema su 100Mbit/s i 54Mbit/s za LAN i WLAN uređaje respektivno. WLAN AP i LAN uređaji su povezani LAN interfejsom. Hardver klijenta komunikaciju vrši isključivo putem WLAN interfejsa, dok, za primer, server komunicira isključivo putem LAN

interfejsa. Radne stanice, koje spadaju ostale delove infrastrukture, mogu koristiti oba načina komunikacije. Na slici 5.1 dat je primer upotrebe LAN interfejsa.

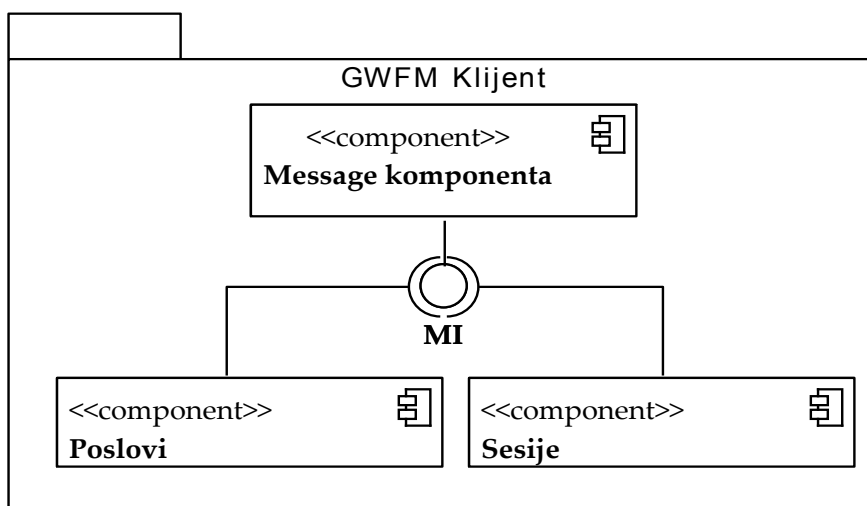
U ostale delove infrastrukture sistema mogu se svrstati i klijenti koji pristupaju sistemu putem WAN interfejsa. U takve uređaje mogu da spadaju radne stanice ili mobilni uređaji.

Arhitekturu GWFM sistema čine slike 5.3. i 5.4 zajedno. Na slici 5.3 prikazan je dijagram komponenata klijentskog modula dok je na slici 5.4 prikazan dijagram serverskog modula.



Slika 5.2: Jedna od uloga Message komponente u komunikaciji komponenata

Komponente klijentskog modula *Poslovi* i *Sesije* komuniciraju sa okruženjem uz pomoć *Message* komponente. Pomenuta komponenta odgovorna je za slanje podataka serveru i prihvatanje paketa podataka od strane servera. Ova komponenta služi kao veza između PHP i Javascript apstraktnih slojeva (slika 5.2). Ovo je vitalna komponenta bez koje komunikacija unutar sistema ne bi bila moguća. Prenos informacija između komponenata i unutar samih komponenata ne bi funkcionisao. Na dijagramima komponenata vidi se zavisnost velikog broja komponenata od *Message* komponente.



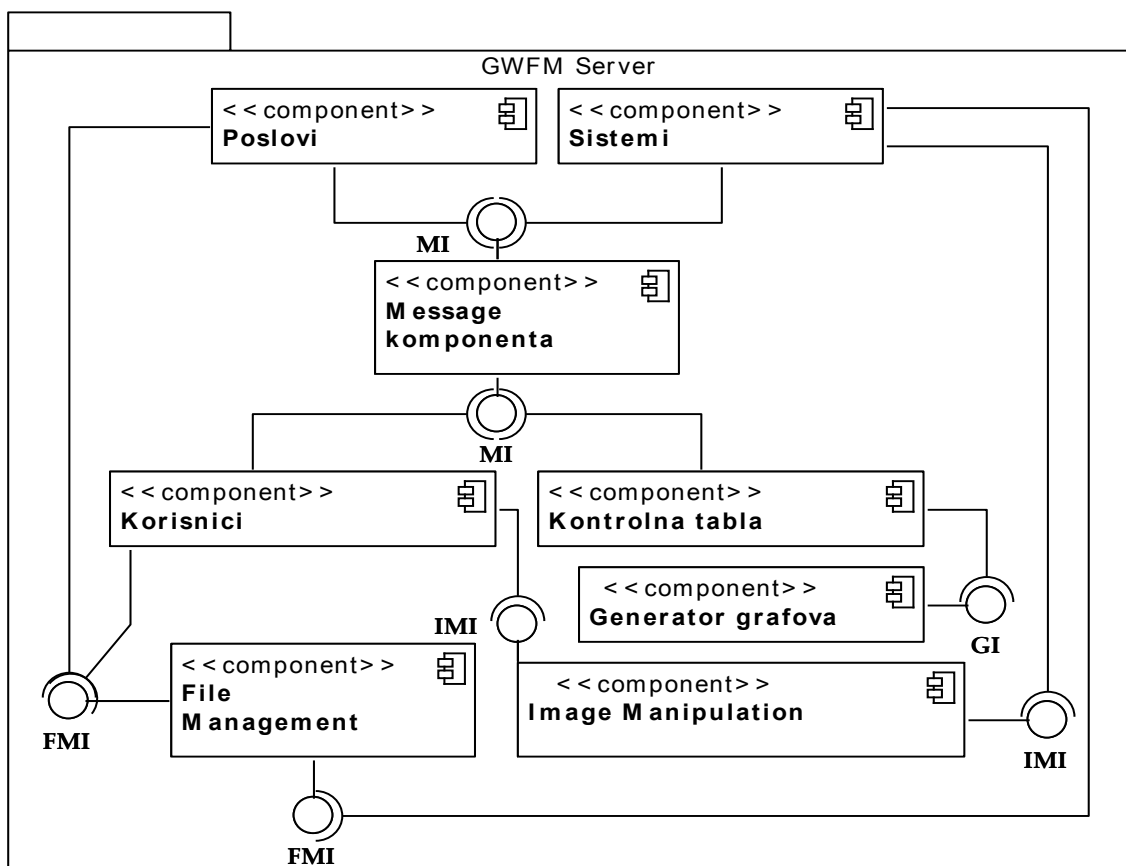
Slika 5.3: Komponente klijentskog modula

Komponente serverskog modula rade na vrlo sličan način kao i komponente klijentskog modula. Za prenos informacija unutar komponente (slika 5.2) kao i za

međusobnu komunikaciju koristi se *Message* komponenta. Sve četiri osnovne komponente serverskog modula sistema se oslanjaju na ovu komponentu i zavise od nje. Pomenute četiri komponente (posredno navedene i u tabeli 5.1) su:

- Poslovi
- Sistemi
- Korisnici
- Kontrolna tabla

Navedene komponente su osnovne komponente i služe za upravljanje sistemom. Ovim komponentama se upravlja uz pomoć GUI (grafičkog korisničkog interfejsa). Ostale komponente su internog tipa i koriste ih isključivo osnovne komponente. *Image manipulation* komponenta je zadužena za operacije obrade slika potrebne za funkcionisanje komponenata *Sistemi* i *Korisnici*.



Slika 5.4: Komponente serverskog modula

Komponentu *File management* koriste komponente *Poslovi*, *Korisnici* i *Sistemi* za svoje potrebe upravljanja dokumentima različitih formata. *Generator grafova* koristi isključivo komponenta *Komandna tabla* za potrebe generisanja grafova (za primer) stanja i opterećenja sistema.

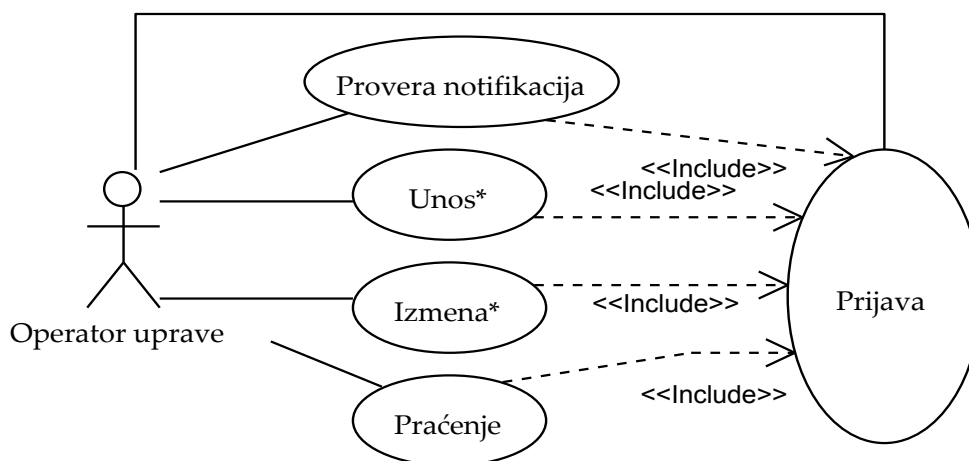
5.3 Procesi

GWFM sistem je sastavljen iz dva dela:

- GWFM Servera i
- GWFM Klijenta

Oba dela GWFM obavljaju različite uloge u celokupnom sistemu. Serverski deo je prvenstveno zadužen za upravljanje podacima sistema, upravljanje poslovima, sistemima i korisnicima. Putem serverskog interfejsa definišu se svi podaci koji se kasnije koriste i kojima se pristupa iz proizvodnje. Klijentski deo je zadužen za praćenje i upravljanje pojedinačnim zaduženjima.

5.3.1 Serverski modul



Slika 5.5: Upotreba serverskog modula

Slika 5.5 prikazuje dijagram upotrebe serverskog modula. Dijagram je opšteg karaktera, zvezdicom su označene stavke koje se odnose na više vrsta operacija. Za primer, *Unos* se odnosi na unos definicija tehnoloških sistema, na unos definicija poslova i na unos definicija korisnika. Pored ovih osnovnih grupa postoje i pod grupe koje neće biti posebno spomenute a biće predstavljene dijagramima aktivnosti u nastavku poglavlja. Ista izuzetna situacija je i sa stavkom *Izmena* koja se takođe odnosi na više slučajeva izmene različitih podataka u različitim modulima.

Detaljan opis slučajeva za serverski modul dat je u tabelama u nastavku teksta.

Naziv	Unos
Kratak opis	Operater unosi podatke u bazu
Učesnici	Operater uprave
Pretpostavke	Da je uspešno prijavljen u sistem
Opis	Nakon prijave u sistem korisnik može da izabere unos različitih vrsta podataka.
Izuzeci	-
Uslovi zadovoljeni posle izvršavanja	Unet podatak

Naziv	Izmena
Kratak opis	Operater menja podatke u bazi
Učesnici	Operater uprave
Pretpostavke	Da je uspešno prijavljen u sistem i da postoje uneti podaci
Opis	Nakon unosa podataka u sistem korisnik može da izabere izmenu različitih vrsta podataka.
Izuzeci	-
Uslovi zadovoljeni posle izvršavanja	Izmenjen podatak

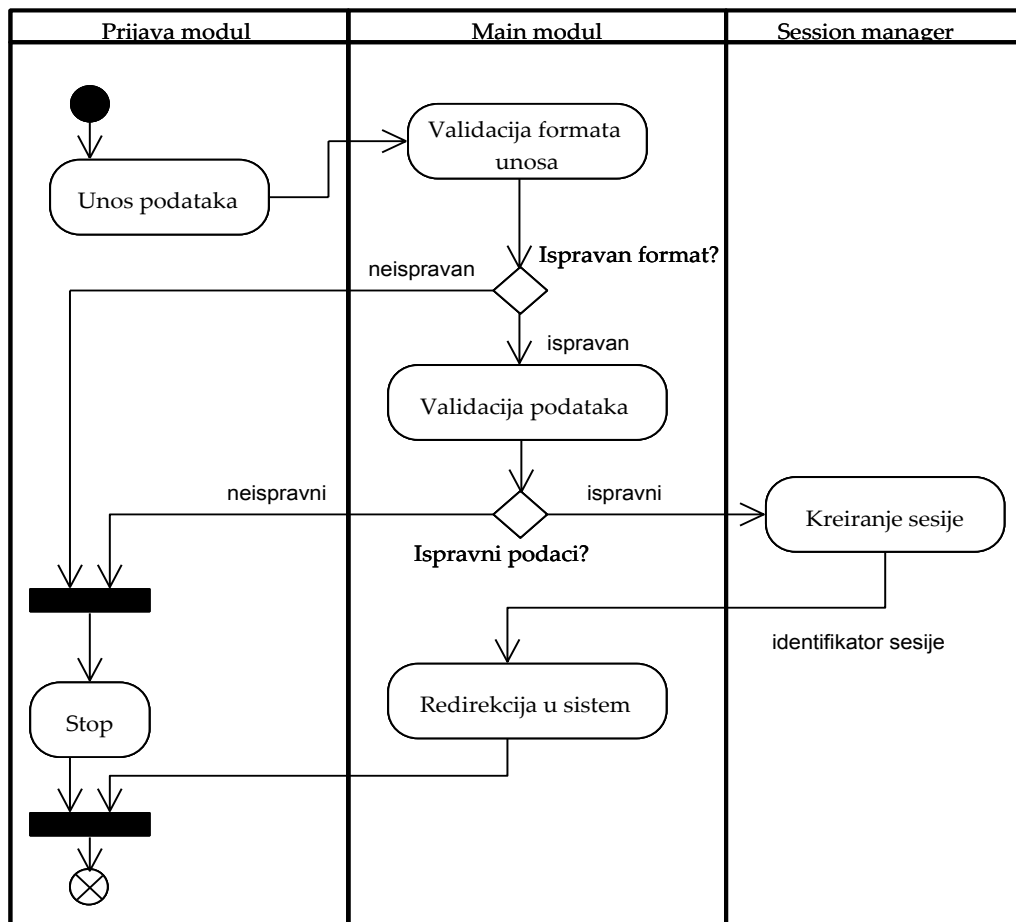
Naziv	Praćenje
Kratak opis	Operater prati stanje sistema i proizvodnje u realnom vremenu
Učesnici	Operater uprave
Pretpostavke	Da je uspešno prijavljen u sistem i da postoje unete odgovarajuće definicije poslova
Opis	Nakon prijave u sistem korisnik može da izabere praćenja različitih vrsta podataka koji se tiču proizvodnje.
Izuzeci	-
Uslovi zadovoljeni posle izvršavanja	Uvid u stanje sistema

Naziv	Provera notifikacija
Kratak opis	Operater proverava notifikacije
Učesnici	Operater uprave
Pretpostavke	Da je uspešno prijavljen u sistem i da postoje unete odgovarajuće definicije poslova
Opis	Operater proverava notifikacije u koje spadaju i zahtevi za dodelu poslova
Izuzeci	-
Uslovi zadovoljeni posle izvršavanja	Uvid u status notifikacija

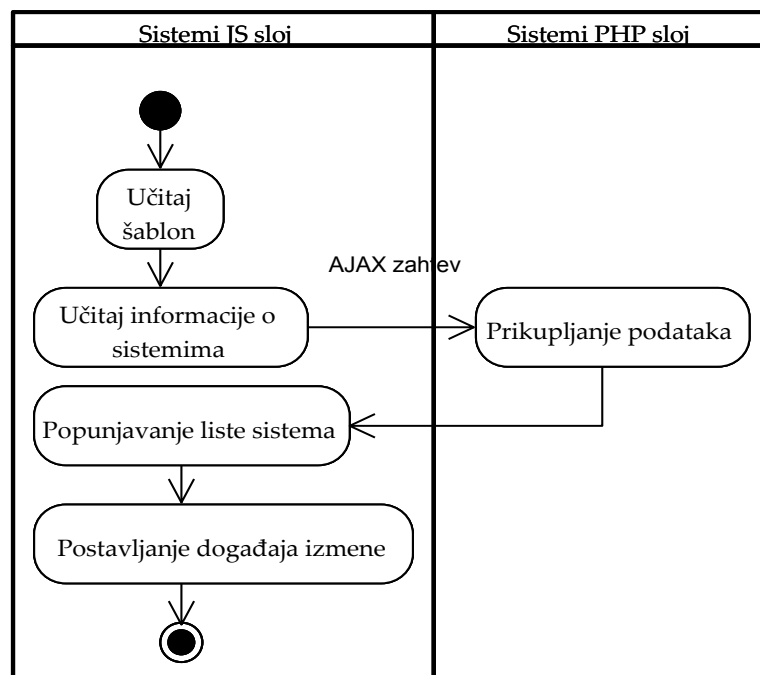
Naziv	Prijava
Kratak opis	Operater se prijavljuje u sistem
Učesnici	Operater uprave
Pretpostavke	-
Opis	Operater se prijavljuje u sistem
Izuzeci	-
Uslovi zadovoljeni posle izvršavanja	Operator je prijavljen

U nastavku teksta biće prikazane aktivnosti koje se izvršavaju u pojedinim delovima sistema. Jednostavniji slučajevi će biti izostavljeni.

Slika 5.6 prikazuje dijagram aktivnosti procedure prijave u sistem. Inicijalni korak jeste unos korisničkih podataka od strane Operatera uprave. Ti korisnici su korisničko ime i lozinka za pristup sistemu. Nakon unosa vrši se validacija unetih podataka. Validacija se vrši metodom upotrebe regularnih izraza. Ovaj šablon zahteva određeni stepen kompleksnosti i dužine korisničkih podataka i sprečava različite načine zloupotrebe istih. Ukoliko je format unetih podataka korektan sistem nastavlja dalje sa validacijom podataka. U suprotnom kompletan proces prijave staje. Validacija podataka se vrši poređenjem unetih vrednosti sa vrednostima u bazi. Ukoliko su podaci koje je korisnik uneo neispravni, proces prijave takođe staje. Ukoliko su ispravni, sistem kreira sesiju za korisnika i vrši preusmeravanje korisnika u sistem. Kreiranje sesije se vrši kreiranjem ključa sesije. Ključ sesije je nasumični broj kreiran posebnim metodima. Nakon kreiranja ključa proverava se postojanje sesija korisnika koji momentalno pristupa. Ukoliko postoje, sesije se uklanjaju. Informacije o sesiji se smeštaju u bazu podataka i u memoriju web pretraživača koji korisnik koristi.



Slika 5.6: Dijagram aktivnosti prijave u serverski modul

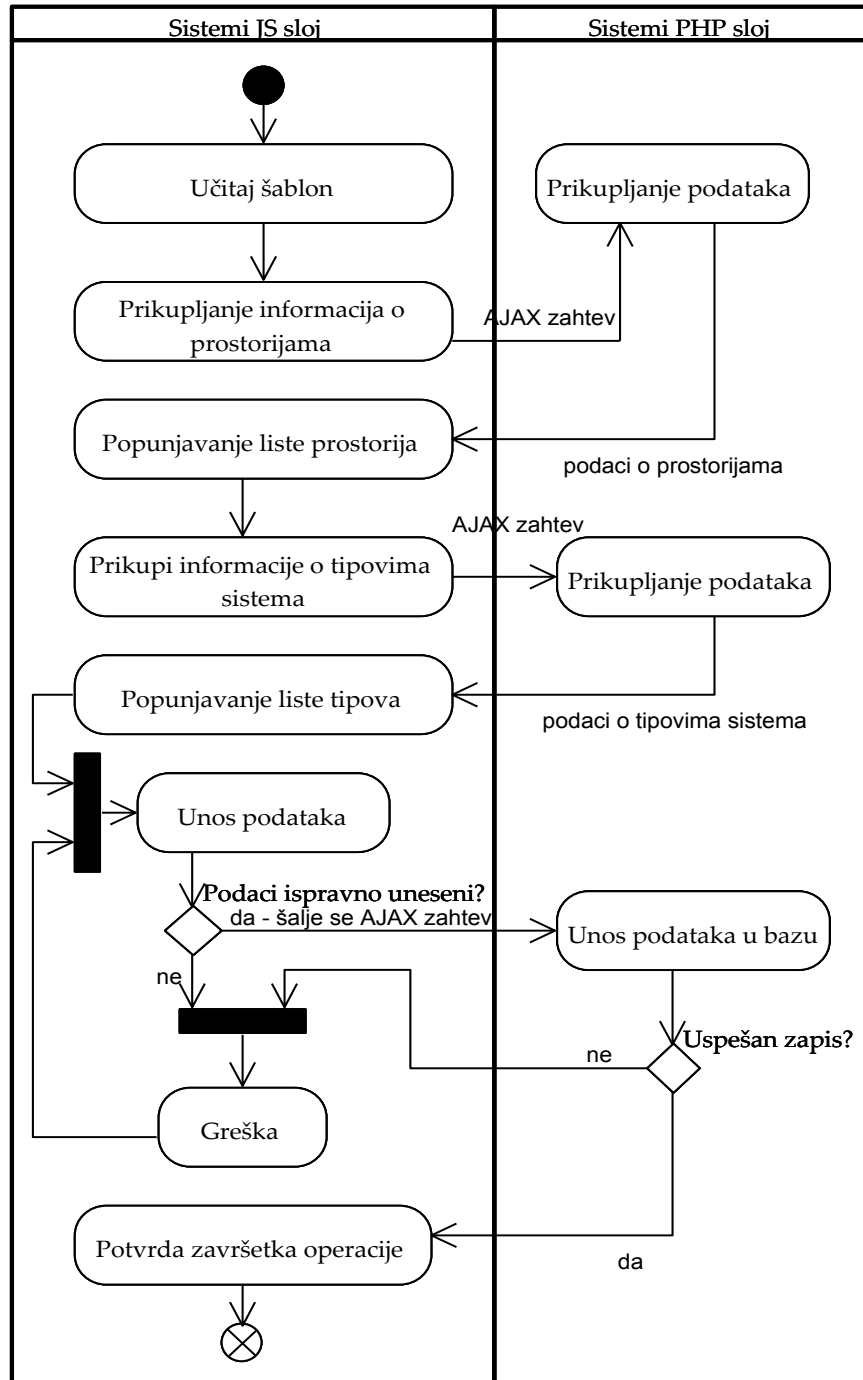


Slika 5.7: Dijagram aktivnosti generisanja liste sistema

Dijagram sa slike 5.7 daje prikaz aktivnosti koje se odvijaju u procesu generisanja liste tehnoloških sistema. Prvi korak predstavlja učitavanje šablona. Šabloni bivaju popunjeni podacima i predstavljaju kompleksne celine elemenata koje nije lako generisati jedan po jedan ili u manjim grupama. Elementi se "ubrizgavaju" u šablon i time se dobijaju realni podaci. Informacije se prikupljaju iz baze podataka. Proces funkcionise slanjem AJAX zahteva PHP servisnim skriptama koje imaju ulogu dostavljanja podataka Javascript (u daljem tekstu JS) sloju. Na osnovu parametara koji su poslani iz JS sloja skripta iz PHP sloja prikuplja i dostavlja tražene informacije. U ovom slučaju to su svi podaci vezani za definisane tehnološke sisteme. U procesu popunjavanja šablona pored tekstualnih elemenata, koji nose informacije o sistemu, generišu se i elementi koji obezbeđuju interakciju korisnika sa tabelom sistema. U te elemente spadaju tasteri koji omogućuju izmenu i brisanje pojedinačnih definicija sistema. Nakon generisanja pomenutih elemenata potrebno im je dinamički dodeliti događaje. Metodi događaja izmene kao glavni parametar poseduju identifikacioni broj definicije tehnološkog sistema. Moguće je izmeniti sve podatke sistema. Pored događaja izmene dinamički se kreiraju i događaji za uklanjanje sistema. Proces uklanjanja je moguć isključivo ukoliko sistem nije uključen niti u jednu aktivnost.

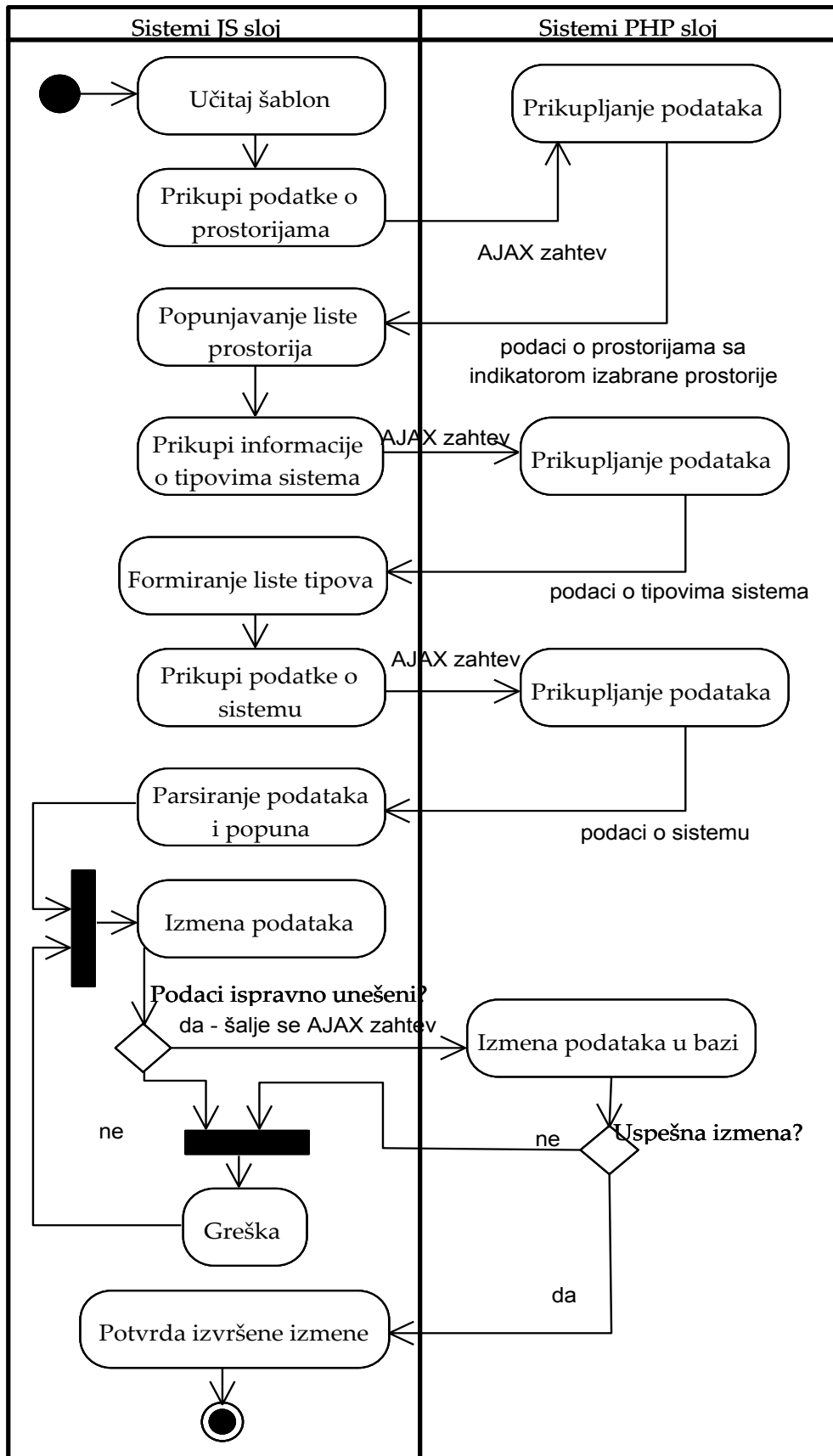
Slika 5.8 daje ilustraciju aktivnosti koje se odvijaju u procesu kreiranja definicije jednog tehnološkog sistema. Kao prvi korak izvršava se učitavanje šablona. Šablon u sebi sadrži polja za unos informacija o sistemu. Neka od polja inicijalno nisu prazna i potrebno ih je popuniti. Ta polja su polje koje sadrži spisak definicija prostorija i polje koje sadrži spisak tipova sistema. Iz JS sloja potrebno je poslati dva odvojena AJAX zahteva. Prvi bi trebao da pribavi informacija o trenutno definisanim prostorijama. Potrebno je samo ime i identifikacioni broj prostorije. Definicija tehnološkog sistema će se vezati upravo za identifikacioni broj prostorije. Ime se koristi za prikaz korisniku. Željene prostorije moraju biti definisane pre unosa definicije tehnološkog sistema. Drugi zahtev pribavlja informacije o tipovima sistema. Ovi tipovi zavise od polja primene ovog modela informacionog sistema. Moguće ih je prilagoditi potrebama proizvodnog i poslovnog sistema. Kada su pomenute dve operacije završene korisniku se ostavlja mogućnost popunjavanja ostalih polja i odabira po jedne vrednosti od dva pomenuta dinamički popunjena polja. Po unosu svih potrebnih podataka vrši se proces provere da li su svi podaci uneti u korektnom formatu *preg_match* metodom korišćenjem odgovarajućih regularnih izraza. U slučaju nekorektno unetih podataka sistem korisnika obavestava o grešci i zahteva ispravku unesenih podataka. U suprotnom, sistem podatke u posebnom obliku iz JS sloja šalje u PHP sloj koji vrši upis podataka u bazu. PHP sloj vraća potvrdu uspešnosti zapisa podataka u JS sloj i ukoliko je odgovor pozitivan, sistem preusmerava korisnika na spisak definisanih sistema.

Aktivnosti izmene podataka jednog tehnološkog sistema slične su upravo nabrojanim koje se tiču unosa. Prikazane su na slici 5.9.



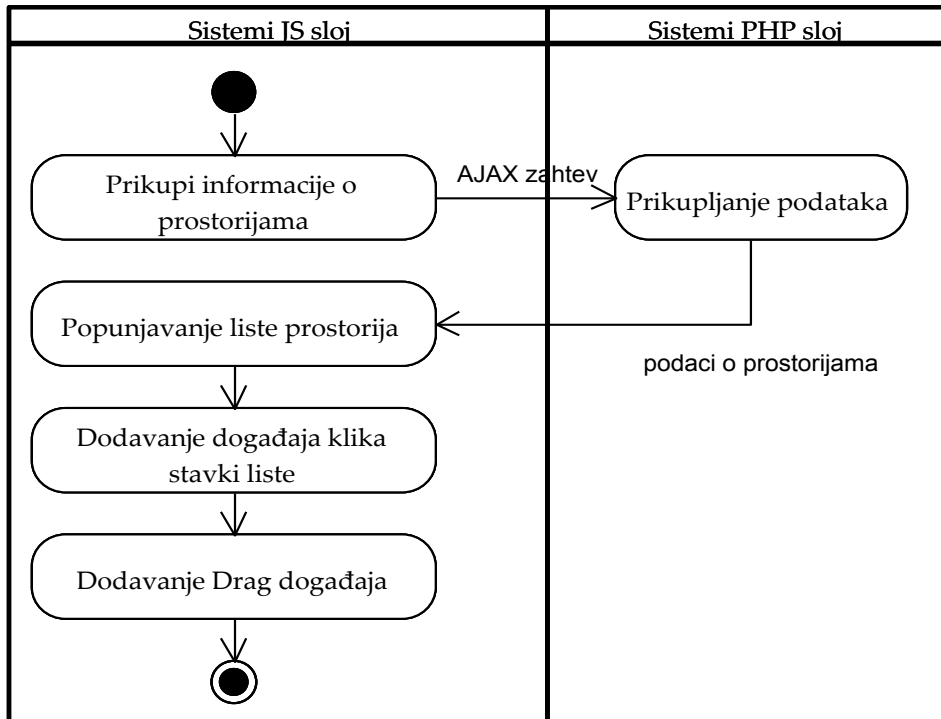
Slika 5.8: Dijagram aktivnosti unosa tehnoloških sistema

Aktivnosti procesa izmene tehnološkog sistema (slika 5.9) uključuju prve dve akcije iz procesa unosa sistema - formiranje lista tipova sistema i prostorija. Pored ovoga postoji i akcija pribavljanja informacija o sistemu koji se menja. PHP sloju se putem AJAX zahteva prosleđuje identifikacioni broj definicije tehnološkog sistema na osnovu koga skripte PHP sloja učitavaju informacije iz baze podataka. Te informacije se u posebnom obliku prosleđuju nazad u JS sloj u kome se raščlanjuju. Tako raščlanjene i pripremljene informacije se raspoređuju po poljima interfejsa koje je prethodno formiran iz šablona u prvoj akciji.

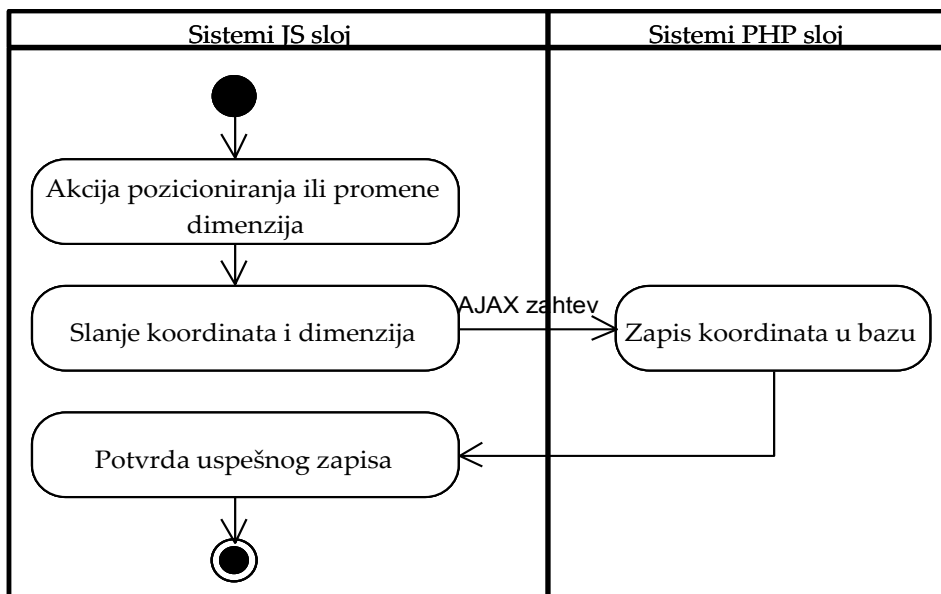


Slika 5.9: Dijagram aktivnosti izmene sistema

Nakon raspoređivanja podataka na odgovarajuća mesta korisnik ima uvid u trenutne podatke koji se nalaze u bazi podataka, može da ih menja i sačuva ponovo. Podaci moraju takođe proći kontrolu unosa, tj. moraju biti uneti u ispravnom formatu. Metod kontrole je isti kao i kod aktivnosti unosa definicije sistema.



Slika 5.10: Dijagram aktivnosti pozicioniranja sistema - inicijalizacija



Slika 5.11: Dijagram aktivnosti pozicioniranja sistema - zapis podataka

Aktivnosti pozicioniranja sistema prikazane su na slici 5.10 i 5.11. Podeljene su u dva dela: inicijalizacija i pozicioniranje. U sklopu inicijalizacije prikupljaju se podaci

vezani za prostorije. U sklopu prostorija formiraju se objekti koji čine tehnološke sisteme. Pomenutim objektima se dodeljuju akcije prevlačenja elemenata kursom miša i to predstavlja finalnu akciju prvog dela. Drugi deo aktivnosti vezanih za pozicioniranje sistema jeste akcija koju izvodi korisnik. Svakom promenom pozicije ili dimenzija elementa koji označava sistem koordinate i dimenzije se u posebnom obliku šalju iz JS sloja u PHP sloj koji u realnom vremenu registruje podatke u bazi podataka. Za ove akcije nije potrebna nikakva potvrda korisnika, zahtevi se izvršavaju u pozadini. Potvrda o uspešnosti zapisa namenjena je za interne potrebe, korisnik ne dobija nikakvu informaciju.

Kreiranje definicije posla je nešto komplikovanije u odnosu na proces unosa sistema. Za kreiranje definicije posla potrebno je odrediti nekoliko celina podataka: osnovne podatke, dokumente vezane za definiciju posla, segmente proizvoda i faze kroz koje prolazi proizvod u toku izrade. Za unos osnovnih podataka potrebno je uneti osnovne podatke o definiciji posla i odrediti dva podataka koji moraju biti ranije određeni. Ti podaci su: grupa posla i klijent. Pribavljanju tih podataka su posvećene i prve akcije dijagrama sa slike 5.12.

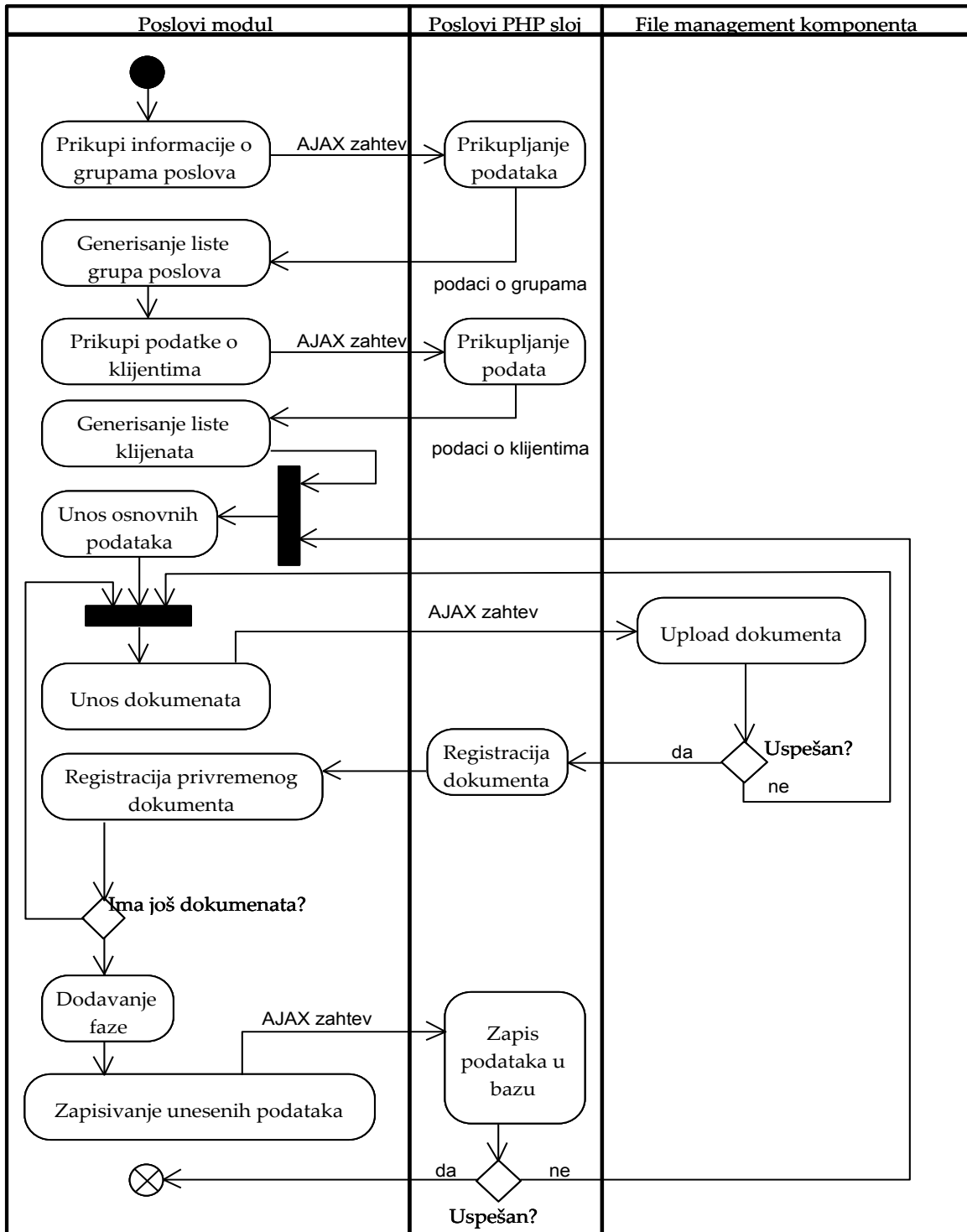
Prvi koraci aktivnosti predstavljaju akcije prikupljanja podataka o grupama poslova. PHP sloju se šalje AJAX zahtev sa potrebnim parametrima. Nakon obrade zahteva podaci se prikupljaju i šalju nazad u JS sloj gde se isti parsiraju i raspoređuju na potrebna mesta. Istom procedurom se dobijaju i informacije o trenutno definisanim klijentima. Ukoliko pribavljanje ovih podataka ne uspe, neće biti moguć unos definicije posla (zbog dimenzija dijagrama, to nije prikazano na dijagramu aktivnosti sa slike 5.12, smatra se da zahtev uvek uspe). Nakon izvršavanja prvih nekoliko akcija dolazi se do akcije u kojoj korisnik unosi osnovne podatke.

Akcije koje se tiču određivanja dokumenata koji prate definiciju posla mogu se izvršiti najmanje jednom a najviše neograničen broj puta (uslovno rečeno). Za svaku definiciju posla mora biti definisan barem jedan prateći dokument. Definiciju posla nije moguće drugačije kreirati. Nakon definicije dokumenta dokumenti se privremeno registruju u memoriji web čitača. Nakon preuzimanja dokumenta (upload) vrši se slanje zahteva PHP sloju koji vrši trajnu registraciju dokumenta. Pri preuzimanju dokumenata kontroliše se naziv dokumenta. Ukoliko naziv dokumenta ne zadovoljava zahteve vrši se prilagođavanje i zaštita od pojave duplikata dokumenata. Po preuzimanju dokumenta vrši se formiranje niza elemenata koji korisniku daju uvid u izvršenu operaciju. Elementi obuhvataju odgovarajuću ikonu (koja odgovara tipu dokumenta), naziv dokumenta i opciju za brisanje dokumenta. U slučaju odabira opcije brisanja dokumenta, isti biva obrisan sa privremene lokacije i iz privremenog registra.

Određivanje faza obrade segmenata proizvoda predstavlja najkomplikovaniji deo kreiranja definicije posla. Podaci koje treba odrediti mogu varirati u zavisnosti od potreba proizvodnog sistema. Primer podataka koje je potrebno odrediti može biti:

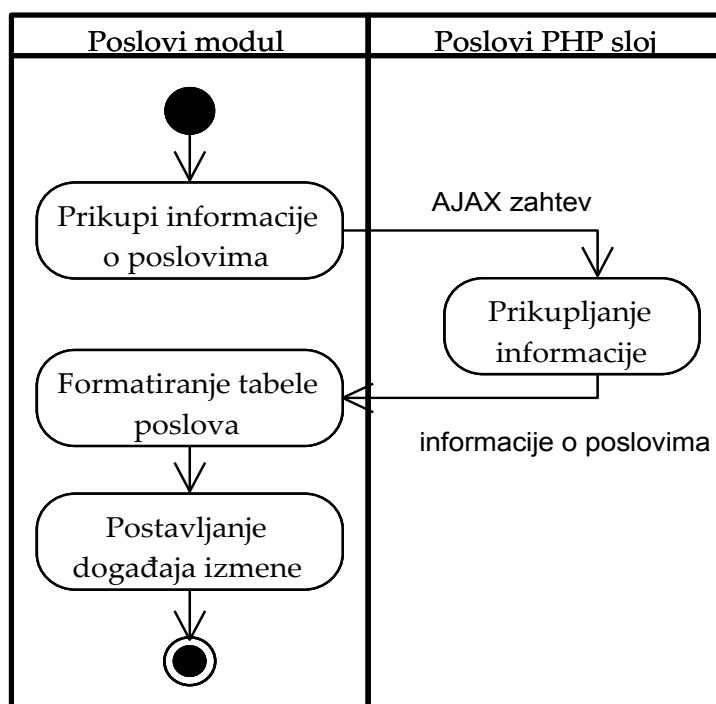
- naziv segmenta koji se obrađuje,

- tehnološki sistem na kom se vrši obrada - ovim se obrada segmenta posredno vezuje i za učesnika proizvodnje (operatera),
- predviđeno vreme završetka operacije izrade ili obrade segmenta proizvoda.



Slika 5.12: Dijagram aktivnosti unosa definicije posla

Uneti podaci se smeštaju u privremene registre. Formiraju se složeni objekti tačno određene strukture u koje se smeštaju podaci. Za svaki podatak kreira se poseban token kao indentifikator podatka. Na osnovu pomenutih tokena sistem prepoznaje podatke i zna šta koji podatak nosi i gde koji podatak treba iskoristiti. Svi podaci prolaze odgovarajuću proveru uz pomoć regularnih izraza kako bi format informacija u bazi bio odgovarajući. Nakon kontrole podaci se konvertuju u poseban oblik i šalju is JS sloja u PHP sloj na dodatnu proveru i zapis u bazu podataka. PHP skripte vrše dodatnu proveru dolaznog zahteva i ukoliko provera prođe bez grešaka pristupa se upisu podataka u bazu. Pored upisa tekstualnih tipova podataka vrši se obrada i transfer dokumenata na finalne lokacije na kojima će biti korišćeni u funkcionisanju sistema. Nakon obrade dokumenata vrši se generisanje faza obrade proizvoda i segmenata.

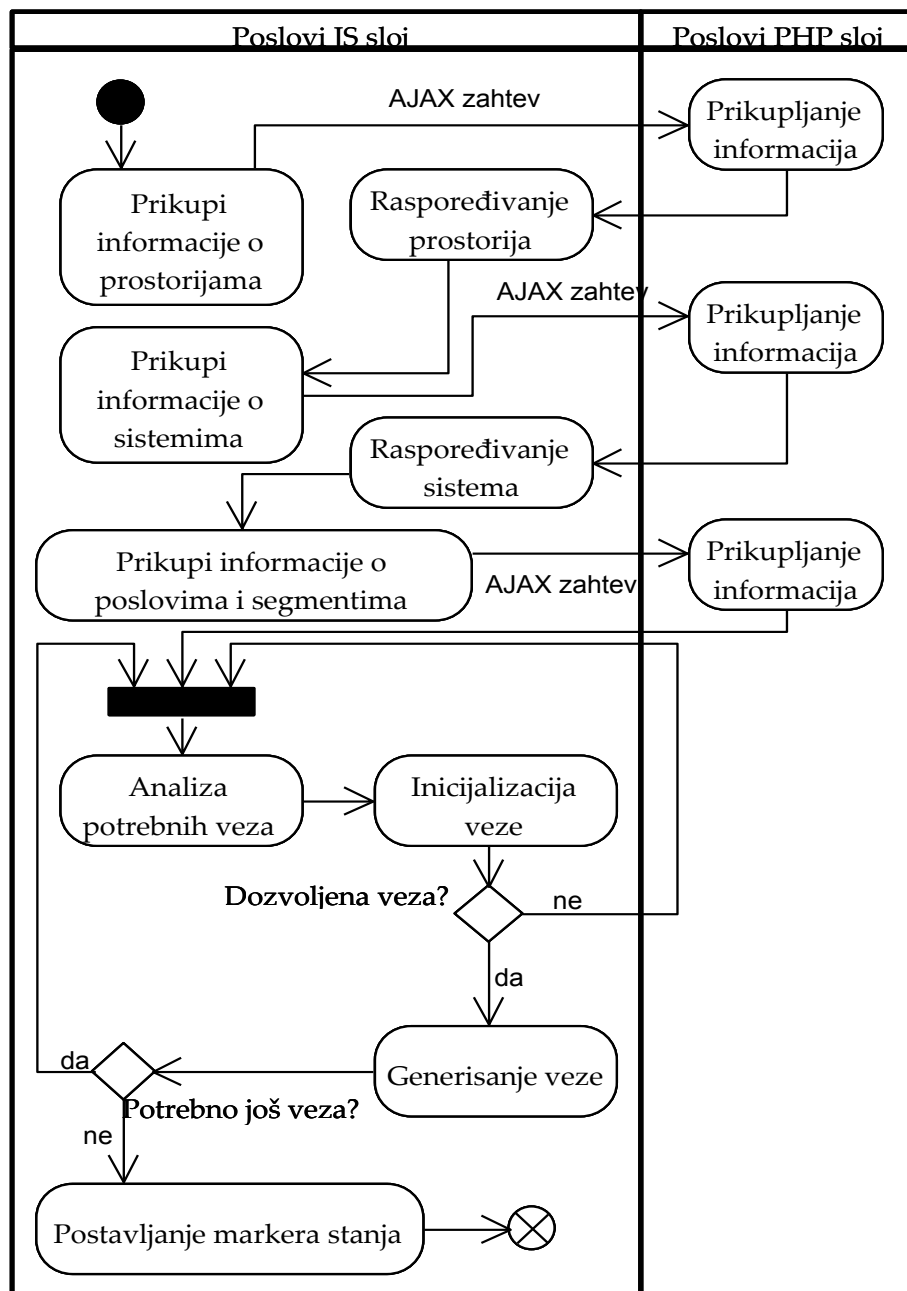


Slika 5.13: Dijagram aktivnosti generisanja liste poslova

Operacije generisanja liste poslova opisane su na slici 5.13. Inicijalno se vrši prikupljanje informacija o definicijama poslova koje su do tog momenta kreirane. Nakon prikupljanja informacija formira se tabela poslova sa potrebnim informacijama koje korisniku nedvosmisleno daju na uvid potpunu sliku o zadatku. Tabela se formira učitavanjem šablona koji se naknadno popunjavaju odgovarajućim informacijama. Te informacije predstavljaju tekstualne elemente kao i elemente koji obezbeđuju interakciju sa korisnikom. Finalna faza predstavlja dodeljivanje događaja elementima zaduženim za interakciju sa korisnikom. Ti elementi su zaduženi da korisniku obezbede izmenu definicije posla u nekoliko celina (dijagram aktivnosti za izmenu definicije nije kreiran zbog velike sličnosti sa dijagramom aktivnosti unosa sa slike 5.12, sličnosti će biti napomenute). Jedan element obezbeđuje izmenu osnovnih

informacija, drugi izmenu dokumenata i treći izmenu faza. Za svaku celinu izmene definicije posla prikuplja se određeni skup podataka. Na dijagramu aktivnosti sa slike 5.12 akcija bi bila prikazana neposredno posle akcije 'Generisanje liste klijenata'.

Jedna od mogućnosti sistema koje direktno zavise od definicije posla jeste i mogućnost kreiranja tokova segmenata proizvoda. Treća faza kreiranja definicije posla obuhvata definisanje faza kroz koje prolaze segmenti proizvoda u roku proizvodnje. U zavisnosti od informacija određenih od strane korisnika i uopšte potrebe izrade proizvoda sistem uz pomoć posebnog algoritma generiše putanje kretanja segmenata proizvoda. Akcije kreiranja tokova segmenata prikazane su na slici 5.14.



Slika 5.14: Dijagram aktivnosti kreiranja tokova segmenata

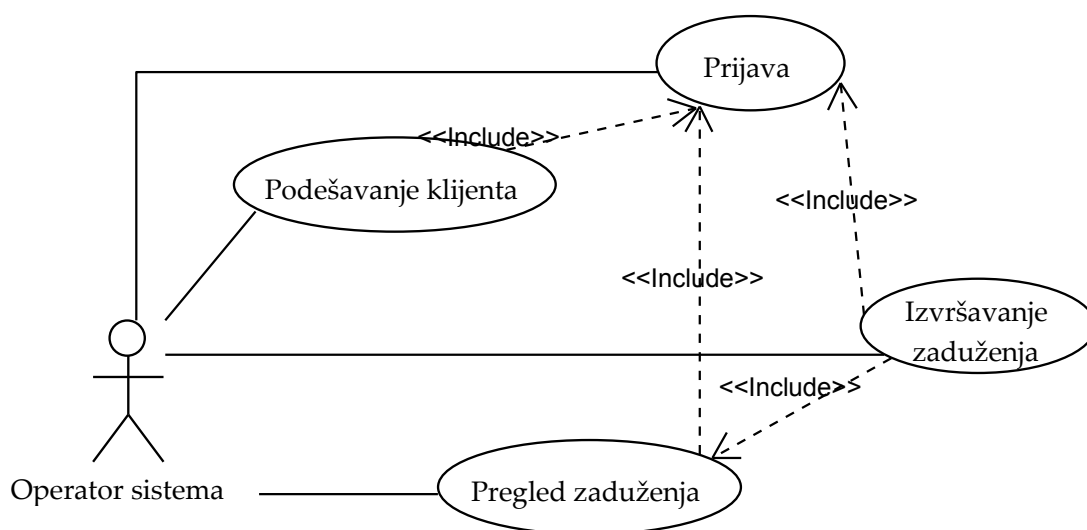
U prvom momentu prikupljaju se informacije o prostorijama koje postoje u listi definicija prostorija (moraju biti ranije definisane). Zatim se aktiviraju procedure koje raspoređuju prostorije unutar radne površine dostupne korisniku. Kada su prostorije raspoređene, sistem učitava informacije o tehnološkim sistemima i kreira objekte unutar svake od raspoređenih prostorija. Svaki objekat predstavlja jedan tehnološki sistem. Objektima sistema dodeljuje se naziv i simbol statusa. Nakon raspoređivanja sistema, sistem učitava informacije o definicijama poslova i segmenata, vrši analize i povezuje odgovarajuće objekte. Povezivanje objekata se vrši po određenim pravilima definisanim unutar logike povezivanja. Nakon generisanja veza raspoređuju se markeri stanja (prikazani u poglavlju 6) čime je i završen proces generisanja kompletnog interfejsa.

Za akcije generisanja grafova nema posebne potrebe kreirati dijagrame aktivnosti. U sklopu low-level modela napravljena su četiri primera prikaza različitih informacija koje prikazuju stanje sistema i slične podatke. Jedan primer grafa jeste prikaz u obliku tipa Pie sa tri kategorije: nezapočetim, započetim i završenim poslovima. Za kreiranje ovog grafika nije potreban poseban algoritam, u pitanju je prebrajanje ćelija tabele. Drugi primer može prikazivati grafik tipa Bar sa podacima o predviđenom i realnom trajanju poslova. Predviđeno vreme se računa prostim sabiranjem predviđenih vrednosti obrade svakog segmenta dok se realno vreme t_r računa prema formuli u nastavku:

$$t_r = \sum_{m=1}^n tsdif(t_{pm}, t_{zm})$$

gde n predstavlja broj segmenata posla dok je $tsdif$ funkcija koja računa razliku tajmstempova početnog vremena t_{pm} i završnog vremena t_{zm} svake sesije obrade segmenta. Završno vreme t_{zm} je jednako trenutnom vremenu ukoliko sesija još nije završena. Sumom svih vremena dobija se ukupno realno vreme trajanja posla. Istim metodom može se izračunati i vremensko zauzeće tehnoloških sistema koje je prikazano trećim grafikom. Pored ovih prikaza moguće je kreirati još varijacija i kombinacija podataka, sve u zavisnosti od potreba proizvodnog ili poslovnog sistema.

5.3.2 Klijentski modul



Slika 5.15: Upotreba klijentskog modula

Detaljan opis slučajeva za klijentski modul dat je u tabelama u nastavku teksta.

Naziv	Podešavanje klijenta
Kratak opis	Operater izvodi operacije podešavanja klijenta.
Učesnici	Operater sistema
Pretpostavke	Da je uspešno prijavljen u sistem putem klijenta
Opis	Operater izvodi operacije podešavanja klijenta. Operacije podrazumevaju podešavanje adrese sajta i dužine dijagonale klijentskog uređaja
Izuzeci	-
Uslovi zadovoljeni posle izvršavanja	Podešen klijentski modul

Naziv	Prijava
Kratak opis	Operater se prijavljuje u sistem
Učesnici	Operater sistema
Pretpostavke	-
Opis	Operater se prijavljuje u sistem
Izuzeci	-
Uslovi zadovoljeni posle izvršavanja	Operater je prijavljen

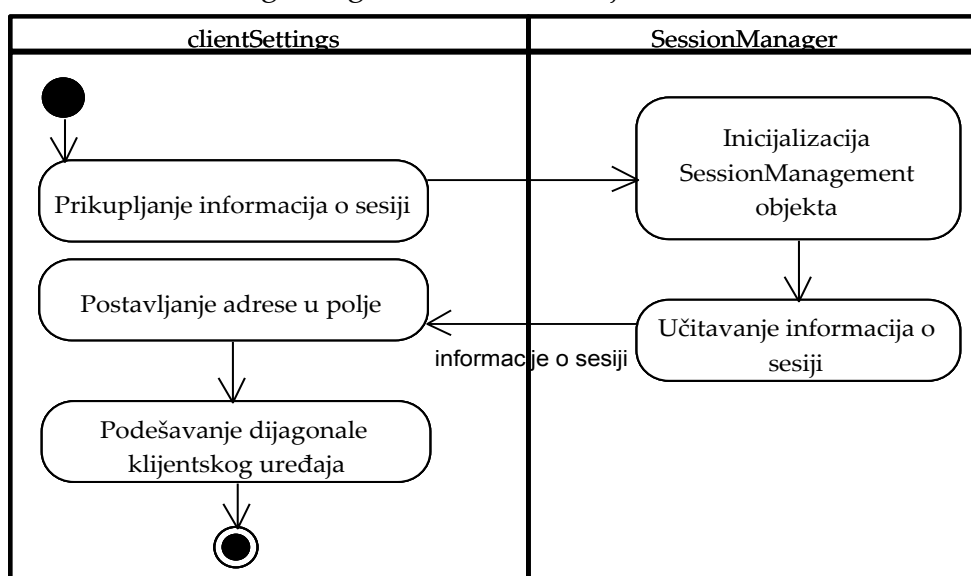
Naziv	Pregled zaduženja
Kratak opis	Operater pregleda dodeljena zaduženja
Učesnici	Operater sistema
Pretpostavke	Da je uspešno prijavljen u sistem i da postoji uneta definicija posla koji se posmatra
Opis	Operater pregleda dodeljena zaduženja od strane operatora uprave.
Izuzeci	-
Uslovi zadovoljeni posle izvršavanja	Stečen uvid u zaduženja

Naziv	Izvršavanje zaduženja
Kratak opis	Operater izvršava zaduženje
Učesnici	Operater sistema
Pretpostavke	Da je uspešno prijavljen u sistem, da postoji uneta definicija posla koji se posmatra i da je zaduženje odabrano
Opis	Operater izvršava dodeljena zaduženja od strane operatora uprave.
Izuzeci	-
Uslovi zadovoljeni posle izvršavanja	Izvršeno zaduženje

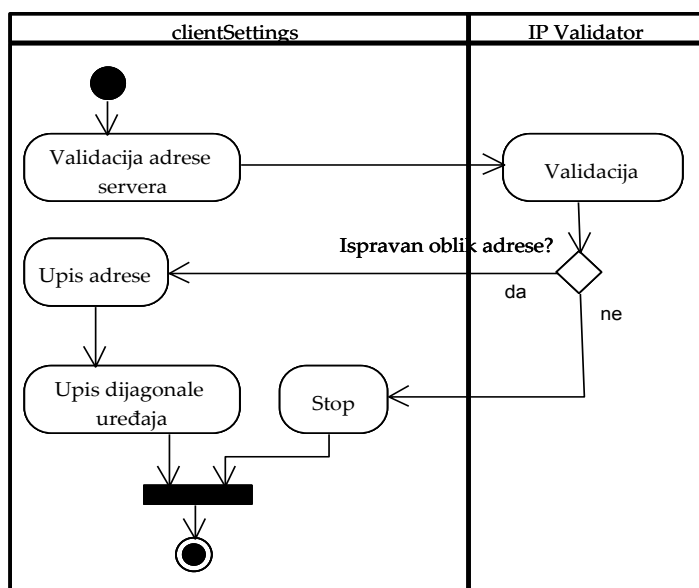
Početak upotrebe klijentskog modula nije moguć bez prijave. Aktivnosti prijave za klijentski modul su slične aktivnostima prijave za serverski modul i neće biti posebno objašnjene.

Nakon prijave korisnik ima obavezu da podesi klijent (slika 5.16). Podešavanje klijenta je obavezno samo jednom po instalaciji softvera. Nakon inicijalnog podešavanja nije potrebno više podešavati parametre klijenta ukoliko se parametri servera ne menjaju. Parametri koje treba podesiti su:

- adresa servera - predstavlja IP adresu servera koji sadrži bazu podataka
- dimenzije klijentskog uređaja - određuje se dužina dijagonale klijentskog uređaja. Dostupne vrednosti su 7 ili 10 inča. Odabirom prve vrednosti mogu se koristiti i klijentski uređaji dijagonale 6 ili 8 inča dok se odabirom druge mogu koristiti i uređaji 9 ili 11 inča.



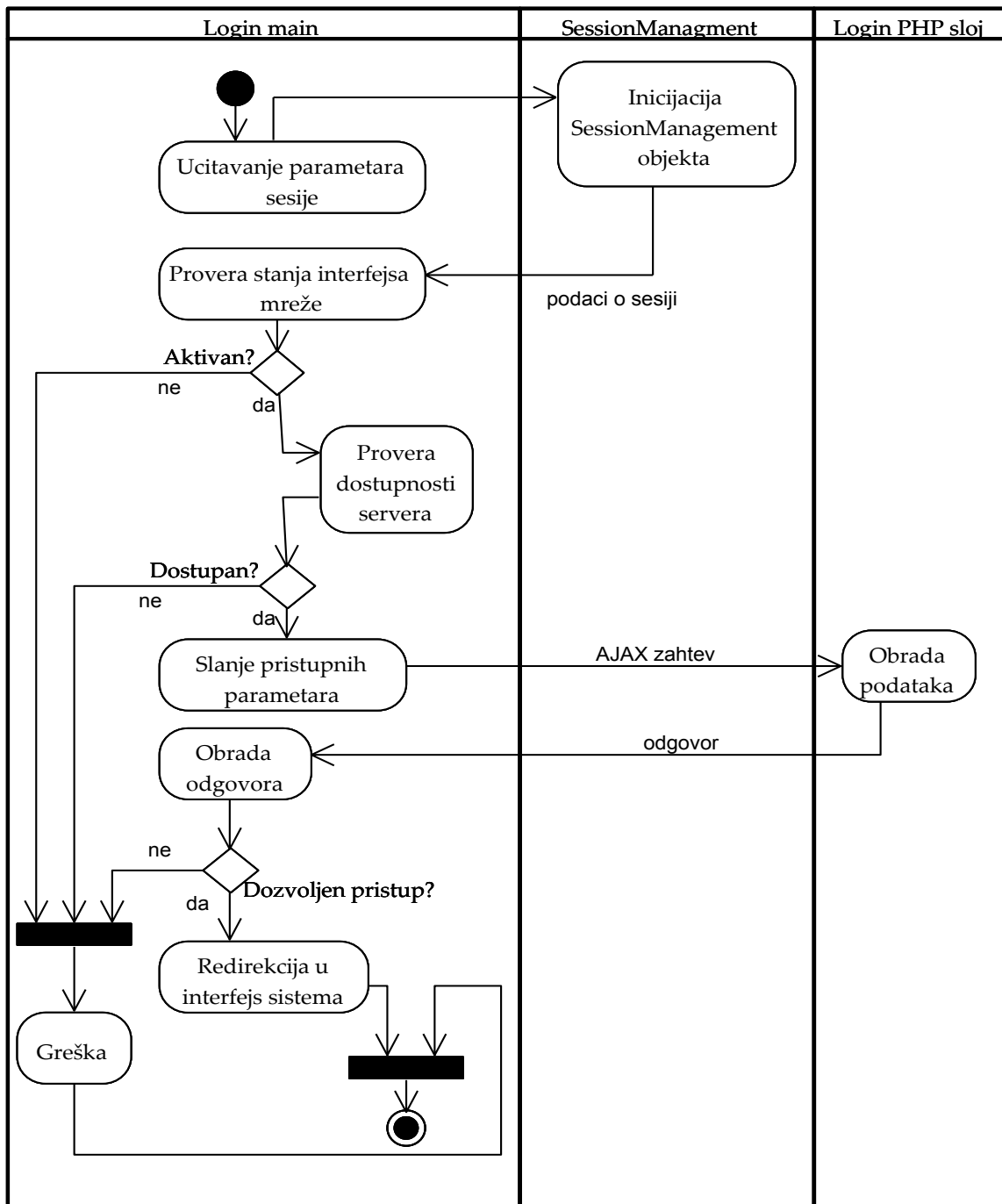
Slika 5.16: Dijagram aktivnosti podešavanja klijentskog modula - inicijalizacija



Slika 5.17: Dijagram aktivnosti podešavanja klijentskog modula - čuvanje podataka

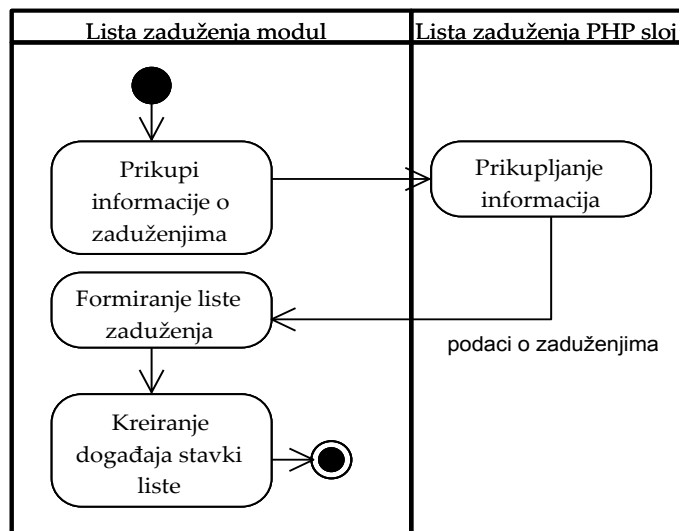
Prvi korak aktivnosti predstavlja inicijalizacija SessionManager-a i učitavanje podataka o sesiji. SessionManager sadrži i podatke u podešavanjima klijenta. Ukoliko su podešavanja prisutna biće učitana i iskorišćena, ukoliko nisu, sistem će iskoristiti osnovna podešavanja.

Drugi deo aktivnosti jeste samo čuvanje podešavanja. Nakon unetih podataka isti se čuvaju u registrima operativnog sistema klijentskog uređaja. IP adresa servera se proverava regularnim izrazom kako bi bila pravilno uneta.

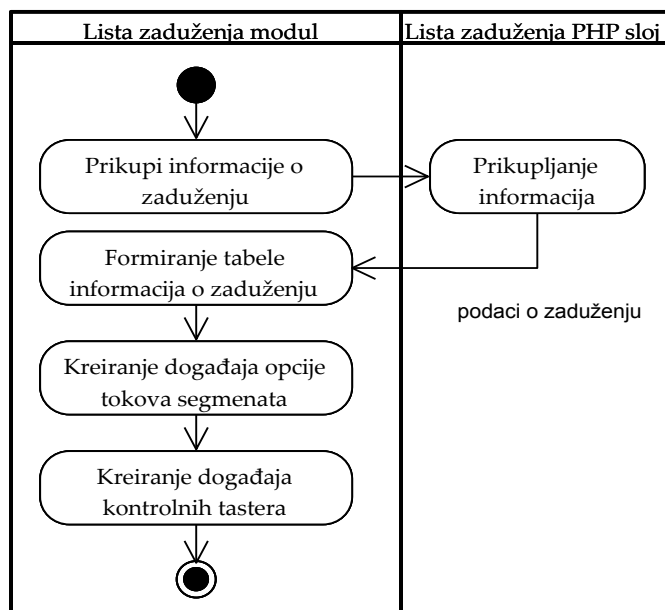


Slika 5.18: Dijagram aktivnosti procesa prijave korisnika u klijentski modul

Po podešavanju klijenta počinje procedura pristupa sistemu, tj. prijava (slika 5.18). Učitavaju se parametri sesije, adresa servera i ostala podešavanja klijenta. Pre nego što se pošalje zahtev za prijavu proverava se stanje mrežnog interfejsa klijentskog uređaja. Ukoliko je mrežni uređaj klijenta aktivan pokušava se sa povezivanjem sa serverom. Vreme čekanja je 3 sekunde i ukoliko se veza ne uspostavi u tom vremenskom intervalu, klijent odustaje od povezivanja i daje korisniku do znanja da povezivanje nije uspelo ili server nije dostupan. U tom slučaju korisnik treba da proveri podešavanja klijenta, parametre i dostupnost servera. Ukoliko je uspostavljanje veze uspešno šalju se pristupni parametri. Ukoliko server odobri pristup, korisnik biva preusmeren u interfejs sistema. U protivnom, korisnik dobija obaveštenje o nedozvoljenom pristupu.



Slika 5.19: Dijagram aktivnosti kreiranja liste zaduženja



Slika 5.20: Dijagram aktivnosti kreiranja liste zaduženja - klik događaj

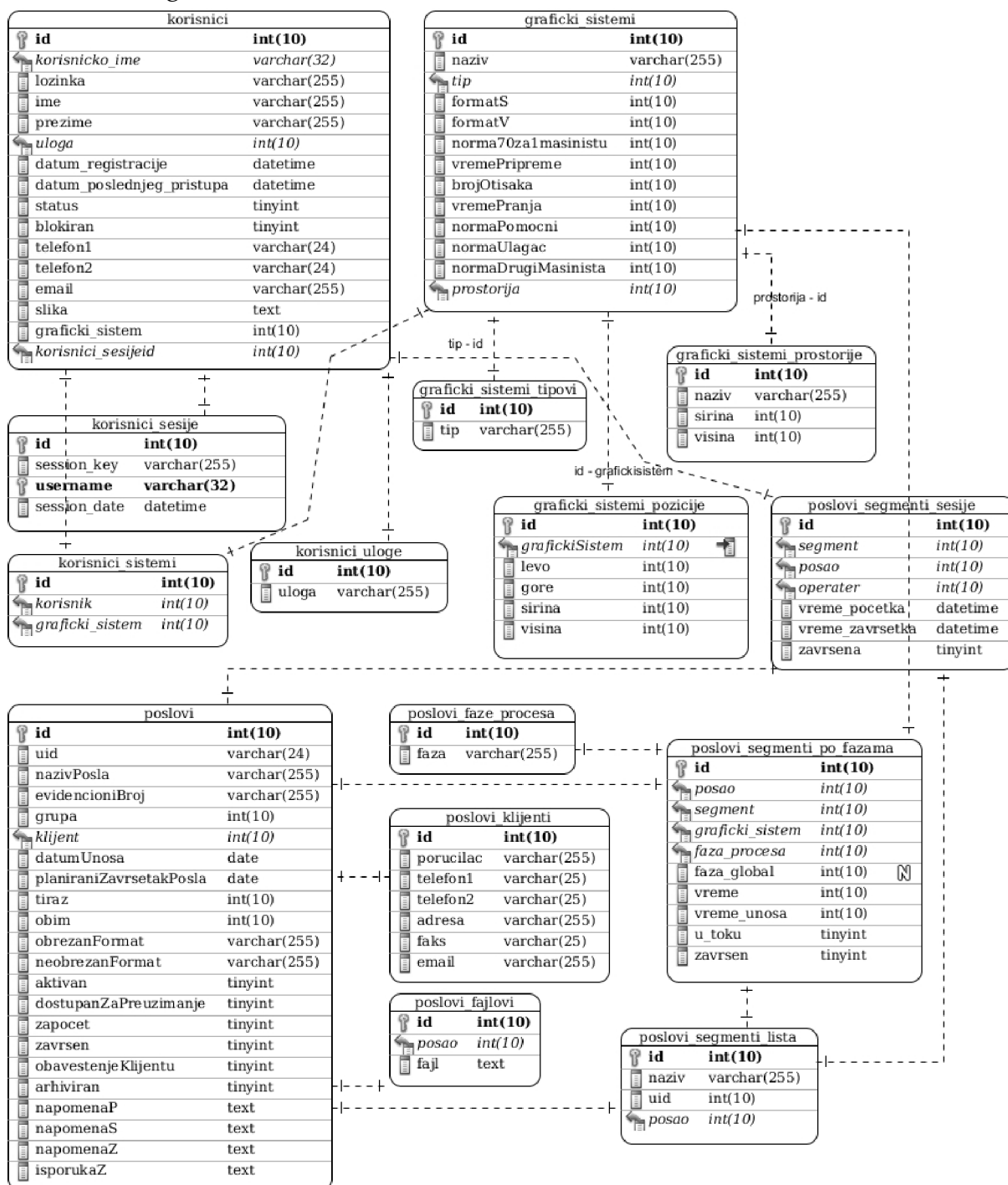
Slike 5.19 i 5.20 daju prikaz aktivnosti kreiranja liste zaduženja u sklopu interfejsa klijentskog modula. Prva opisuje kreiranje liste dok druga opisuje akcije koje se dešavaju kada korisnik izabere neku od stavki liste tj. neki od poslova iz liste.

5.4 Struktura podataka

Listing tabela prikazan je u sklopu priloga 1 (poglavlje 9). Dijagram baze podataka prikazan je na slici 5.21. Tabele potrebne za funkcionisanje sistema su:

1. graficki_sistemi – zadužena za skladištenje osnovnih informacija o tehnološkim sistemima koji učestvuju u simulaciji.
2. graficki_sistemi_pozicije – sadrži informacije o dimenzijama i koordinatama sistema koje je korisnik definisao i rasporedio u sklopu virtuelnih prostorija. Sve vrednosti su izražene u pikselima. Sa tabelom pod 1 je povezana putem id-a.
3. graficki_sistemi_prostorije – u ovoj tabeli su sadržane informacije o virtuelnim prostorijama koje je korisnik prethodno definisao i u koje se smeštaju tehnološki sistemi. Tabela pod 1 se vezuje za ovu tabelu putem primarnog ključa.
4. graficki_sistemi_tipovi – sadrži informacije o dostupnim tipovima tehnoloških sistema. Ova lista je izmenjiva i moguće je uneti bilo koji tip po potrebi ili želji. Tabela pod 1 se vezuje za ovu tabelu preko primarnog ključa.
5. korisnici – sadrži kompletne informacije o korisničkim naložima. Sve tabele koje koriste ovu tabelu povezuju se putem primarnog ključa.
6. korisnici_sesije – sadrži podatke o korisničkim sesijama. Svaka sesija ima svoj jedinstveni ključ koji je vezan za korisničko ime i vreme.
7. korisnici_uloze – u ovoj tabeli nalaze se definicije uloga korisnika u sistemu. Ove uloge se mogu menjati u zavisnosti od potreba.
8. poslovi – sadrži osnovne informacije o definiciji posla.
9. poslovi_fajlovi – sadrži podatke o dokumentima koji se vezuju za definiciju posla. Za tabelu pod 8 vezuje se preko primarnog ključa.
10. poslovi_faze_procesa – sadrži podatke o mogućim operacijama u toku procesa proizvodnje. Ova tabela se može prilagoditi potrebama proizvodnog sistema.
11. poslovi_klijenti – sadrži informacije o podacima klijenata. Klijenti se direktno povezuju sa definicijom posla. Mogu biti iskorišćeni za više definicija poslova.
12. poslovi_segmenti_lista – sadrži definicije i nazive svih segmenata svih poslova. Svaki segment sadrži jedinstveni identifikator.
13. poslovi_segmenti_po_fazama – sadrži podatke o segmentima poslova kao i identifikatore statusa segmenata.

14. poslovi_segменти_sesije – sadrži informacije o sesijama obrade pojedinih segmenata.



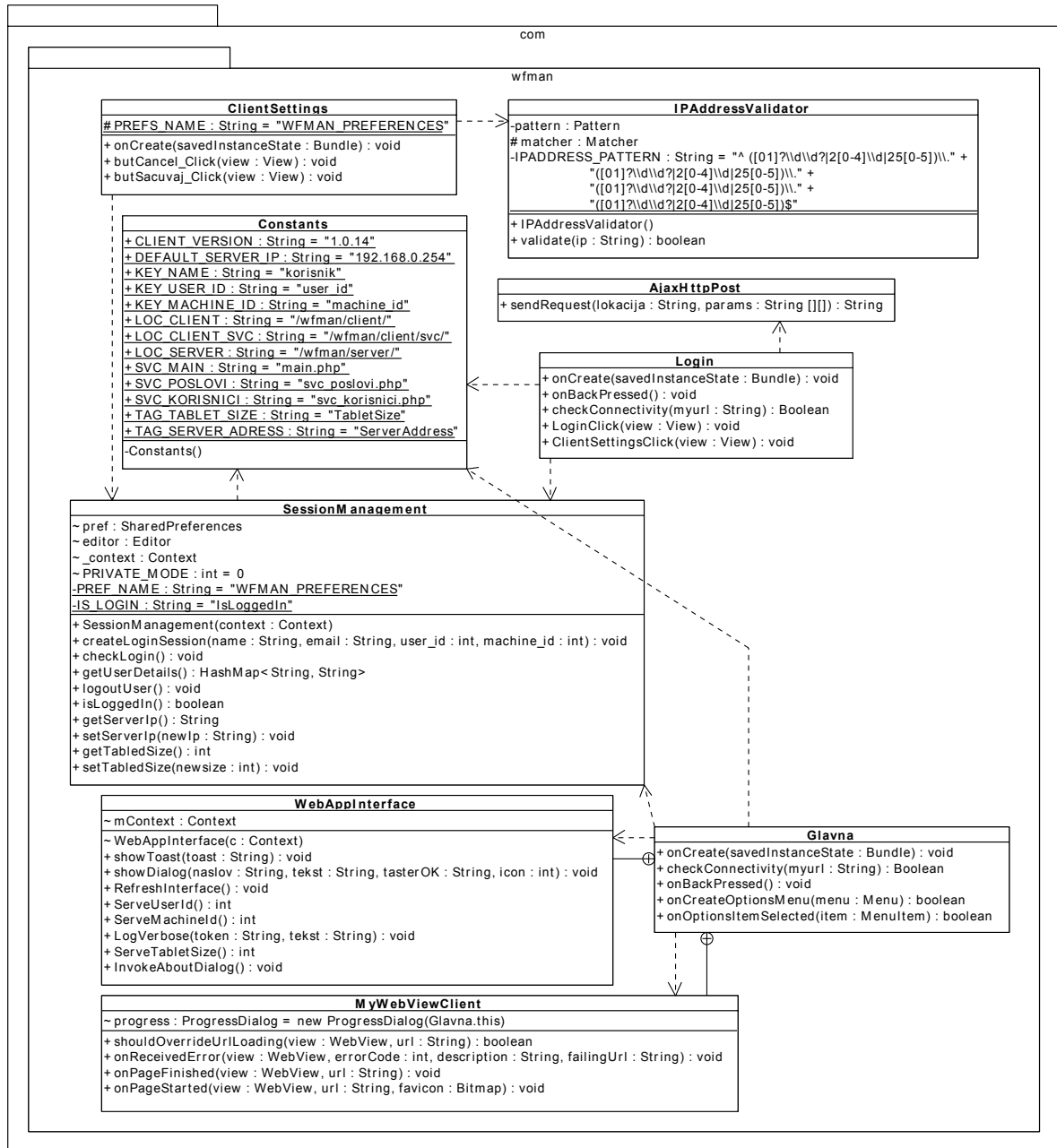
Slika 5.21: Dijagram baze podataka

5.5 Detalji implementacije

Klijentski modul je sastavljen iz nekoliko delova (klasa) (slika 5.22):

- clientSettings klasa je zadužena za operacije podešavanja uređaja klijenta. To podešavanje se ogleda u podešavanjima parametara veze sa serverom

i vrednosti dužine dijagonale uređaja na kome radi. Mobilni uređaji koji su podržani ovim podešavanjima mogu posedovati dužinu dijagonale od 6" do 11".



Slika 5.22: Klasni dijagram klijentskog modula

- *IPAddressValidator* klasa je zadužena za validaciju adrese servera. Validacija se vrši na bazi predefinisanih regularnih izraza. Ovu klasu koristi isključivo *clientSettings* klasa.
- *Login* klasa je zadužena za operacije autentifikacije korisnika sistema, tačnije klijenta. U sklopu ove klase nalaze se i instrukcije za proveru veze. Takođe, ova klasa obezbeđuje prilaz *clientSettings* interfejsu.

- *AjaxHttpPost* klasa je zadužena za realizaciju AJAX zahteva. Posедуje instrukciju za slanje zahteva. Koristi se isključivo od strane *Login* klase.
- Klasa *Constants* obezbeđuje pristup konstantama.
- *SessionManagement* klasa ima zaduženje upravljanja korisničkim sesijama. Takođe obavlja i ulogu učitavanja i čuvanja parametara klijenta.
- *Glavna* klasa je nosilac funkcionisanja klijenta. U sebi sadrži dve pod klase:
 - *WebAppInterface* – zaduženu za komunikaciju lokalnih i serverskih skripti.
 - *MyWebViewClient* – zaduženu za prikaz rada i interakciju sa serverskim skriptama.

Serverski modul je pisan kombinacijom Javascript i PHP jezika i za njega nije moguće napraviti klasni dijagram. Struktura podataka za serverski modul dostupna je u dijagramu baze podataka.

6 STUDIJA SLUČAJA

Za potrebe ovog istraživanja kreirano je prototipsko rešenje informacionog sistema namenjeno za testiranje modela u laboratorijskim uslovima. Rešenje je predviđeno za testiranje u okviru jednog grafičkog proizvodnog sistema.

Primer infrastrukture test okruženja i infrastrukture sistema kreiranog po modelu prikazan je na slici 6.1. Sistem se sastoji iz više elemenata:

- dve ili više radnih stanica
- dva ili više mobilnih uređaja
- mrežne opreme
 - LAN rutera
 - WLAN AP
- servera koji poseduje bazu i DBMS

Specifikacije mobilnog uređaja su:

- poseduje instaliran klijentski modul.
- Android platforma
- dijagonala ekrana: minimalno 6 inča, maksimalno 11 inča
- WiFi modul
- GSM modul (opciono)

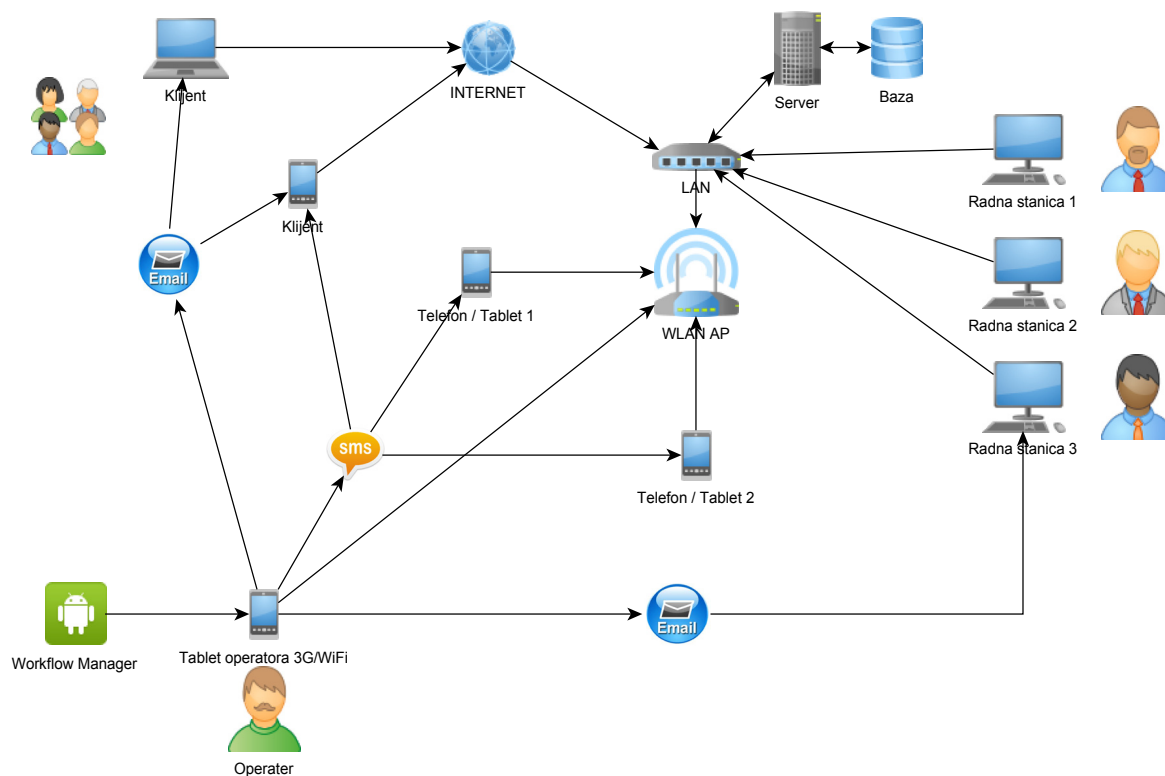
Specifikacije servera su:

- Debian baziran GNU, kernel 2.6 ili noviji
- Apache 2.2 ili noviji
- PHP 5.3 ili noviji
- MySQL server 5.0 ili noviji
- Hardverska konfiguracija:
 - Procesor: Intel Core i3
 - RAM: 4GB
 - Realtek Lan 100Mbit/s
 - Hard disk kapaciteta 320GB

Specifikacije mrežne opreme:

- WLAN ruter, tip nebitan, najmanje 54Mbit/s, AP mod
- LAN svič 100Mbit/s ili 1Gbit/s

Slika 6.1 opisuje i veze i interakciju elemenata infrastrukture. Kompletna komunikacija obavlja se putem LAN interfejsa. Radne stanice sa serverom komuniciraju LAN interfejsom. Uređaji koji ne poseduju mogućnost komunikacije LAN interfejsom umreženi su putem WLAN interfejsa. WLAN AP je direktno povezan sa LAN uređajem. Na slici je naznačena pozicija GWFM klijenta dok se GWFM server nalazi na samom Serveru.



Slika 6.1: Infrastruktura test okruženja

6.1 Rad sa sistemom

6.1.1 Serverski modul

Serverski modul (slika 6.2) zadužen je za upravljanje sistemom i njegovim praćenjem. Posедуje celine za praćenje sistema, definiciju grafičkih sistema, definiciju i upravljanje poslovima proizvodnje i definiciju i upravljanje korisnicima. Po pokretanju serverske komponente potrebna je prijava u sistem (slika 6.3). Prijavu u serverski modul mogu izvršiti samo operatori uprave.



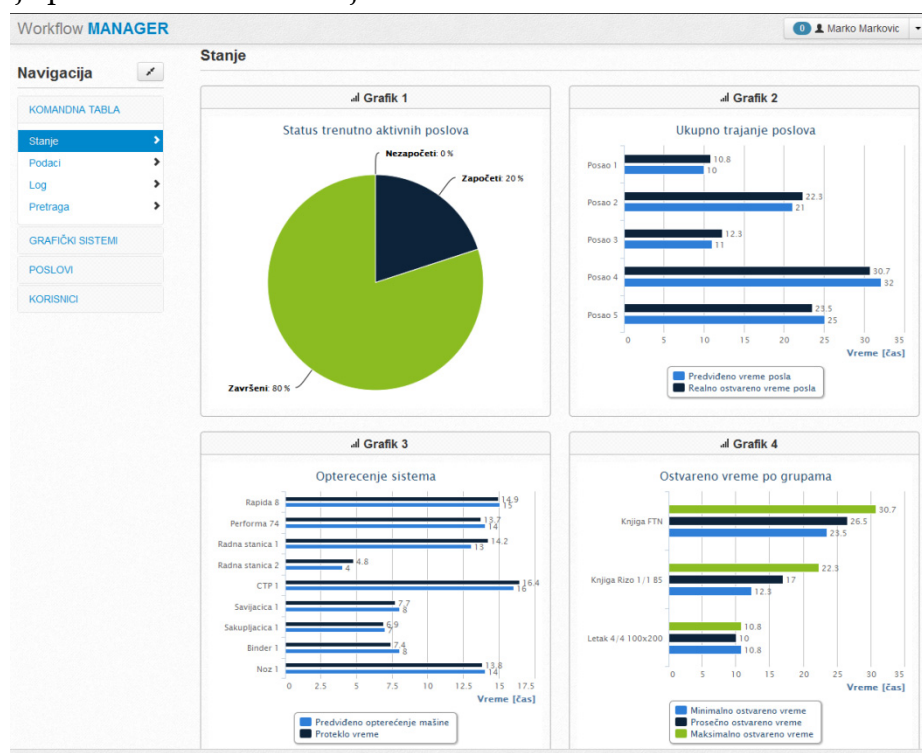
Slika 6.2: O GWFM sistemu

Slika 6.3: Prijava korisnika u serverski deo sistema

Nakon prijave u sistem dobija se prikaz sa slike 6.4. U pitanju je komandna tabla uz čiju pomoć je moguće pratiti kompletno stanje sistema i proizvodnje. U low-level modelu koji je razvijen za potrebe ovog istraživanja implementirana su četiri primera praćenja stanja koja su i prikazana na slici 6.4. Jedan primer ilustruje trenutno stanje nezapočetih, započetih i završenih poslova proizvodnje. Drugi primer daje ilustraciju ukupnog trajanja poslova. U prikazu postoje dve skale:

- jedna prikazuje predviđeno trajanje poslova koji su u toku dok
- druga prikazuje realno ostvareno trajanje poslova.

Na osnovu ovih pokazatelja direktno je dostupna informacija o premašenju vremenskih granica predviđenih za odgovarajuću akciju i mogu se preduzeti akcije za rešavanje problematične situacije.



Slika 6.4: Komandna tabla

Treći primer prikazuje opterećenje tehničko tehnoloških sistema. Obrazac je sličan prethodnom slučaju, poredi se predviđeno i realno opterećenje sistema. Na osnovu ovih parametara može se doći do brzih zaključaka o dostupnosti pojedinih

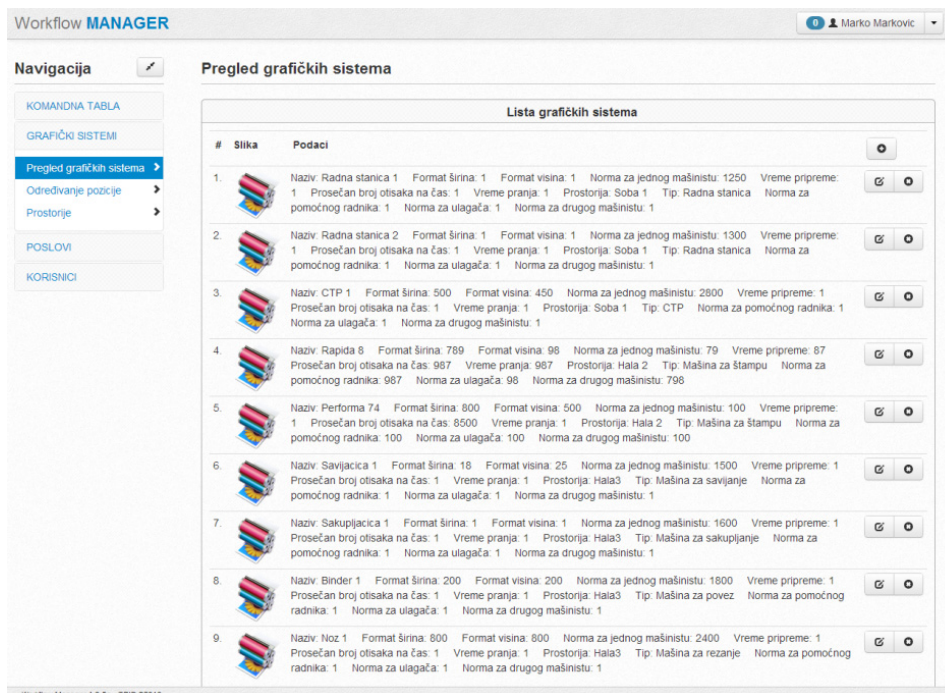
sistema, problemima, zastojsima i slično. Četvrti primer daje uvid u ostvarena vremena po grupama proizvoda.

Kao što je napomenuto, ovo su samo četiri primera. Moguće je razviti više primera praćenja na osnovu potreba poslovnog ili proizvodnog sistema.

Drugi modul serverske komponente je deo koji se bavi definicijom i upravljanjem definisanim grafičkim sistemima (slika 6.5). Ovaj modul se sastoji iz tri dela:

- Pregled grafičkih sistema
- Određivanje pozicije (sistema)
- Prostorije

U delu 'Pregled grafičkih sistema' vrši se pregled i definisanje grafičkih sistema unošenjem potrebnih podataka koji detaljno opisuju jedan grafički sistem. Pomenuti podaci kasnije služe za kreiranje različitih vrsta informacija koje koriste kako korisnici sistema tako i ostali moduli. Pregled se sastoji iz liste definisanih sistema sa potrebnim detaljima.



Slika 6.5: Pregled grafičkih sistema

Lista daje mogućnost dodavanja novih, izmene ili brisanja postojećih sistema. Na slici 6.6 prikazan je interfejs forme izmene jednog sistema. Ova forma je identična formi definicije novog sistema. Pored osnovnih podataka o sistemu određuje se i prostorija u kojoj se sistem nalazi (kasnije će biti reči o definisanju prostorija) kao i tip sistema. Lista tipova sistema može se prilagoditi odgovarajućem proizvodnom sistemu. U primeru sa slike dati su samo neki tipovi.

Grafički sistemi - izmena

Osnovni podaci vezani za sistem

Naziv grafičkog sistema:

Format širina:

Format visina:

Norma za 70% produktivnosti za 1 mašinistu (din/č):

Vreme pripreme mašine za rad (č):

Prosečan broj otisaka u radnom času (ot/č):

Vreme pranja (č):

Prostorija:

Tip grafičkog sistema

- Radna stanica
- CTP
- Mašina za štampu
- Mašina za savijanje
- Mašina za sakupljanje
- Mašina za povez
- Mašina za rezanje
- Mašina za plastifikaciju

Norma za

Pomoćnog radnika:

Ulagača:

Drugog mašinistu:

Slika 6.6: Izmjena jednog od sistema

Pre nego što se obrazloži metod pozicioniranja sistema potrebno je objasniti ulogu prostorija u definiciji sistema. Prostorije predstavljaju virtuelno okruženje u kome se nalaze proizvodni sistemi. Odlikuju ih naziv i dimenzije (širina i visina). Slika 6.7 daje prikaz primera unosa jedne prostorije. Dimenzije nisu realne, izražene su u pikselima i namenjene isključivo za ekranski prikaz.

Workflow **MANAGER** 0 Marko Markovic

Navigacija

- KOMANDNA TABLA
- GRAFIČKI SISTEMI
- Pregled grafičkih sistema >
- Određivanje pozicije >
- Prostorije >**
- POSLOVI
- KORISNICI

Određivanje prostorija

Definisane prostorije			
#	Prostorija	Širina	Visina
1.	Soba 1	500	500
2.	Hala 2	700	500
3.	<input type="text" value="Hala3"/>	<input type="text" value="500"/>	<input type="text" value="500"/>

Pregled

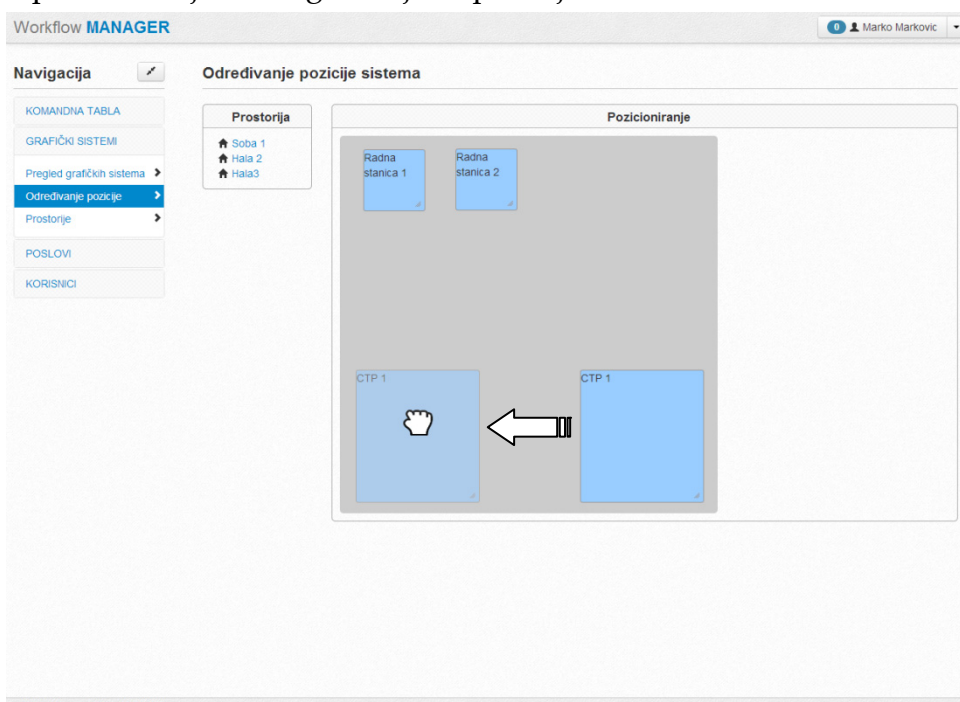
Hala3

Slika 6.7: Definisane prostorije

Pri definiciji grafičkog sistema upravo je prostorija jedna od stavki koja se treba izabrati. Iz ovog razloga prostorije je potrebno definisati pre definisanja grafičkog sistema. Naziv prostorije može biti proizvoljan ili se povezuje sa realnim okruženjem

dok se dimenzije prilagođavaju rezoluciji ekrana i ne povezuju se sa realnim dimenzijama prostorija.

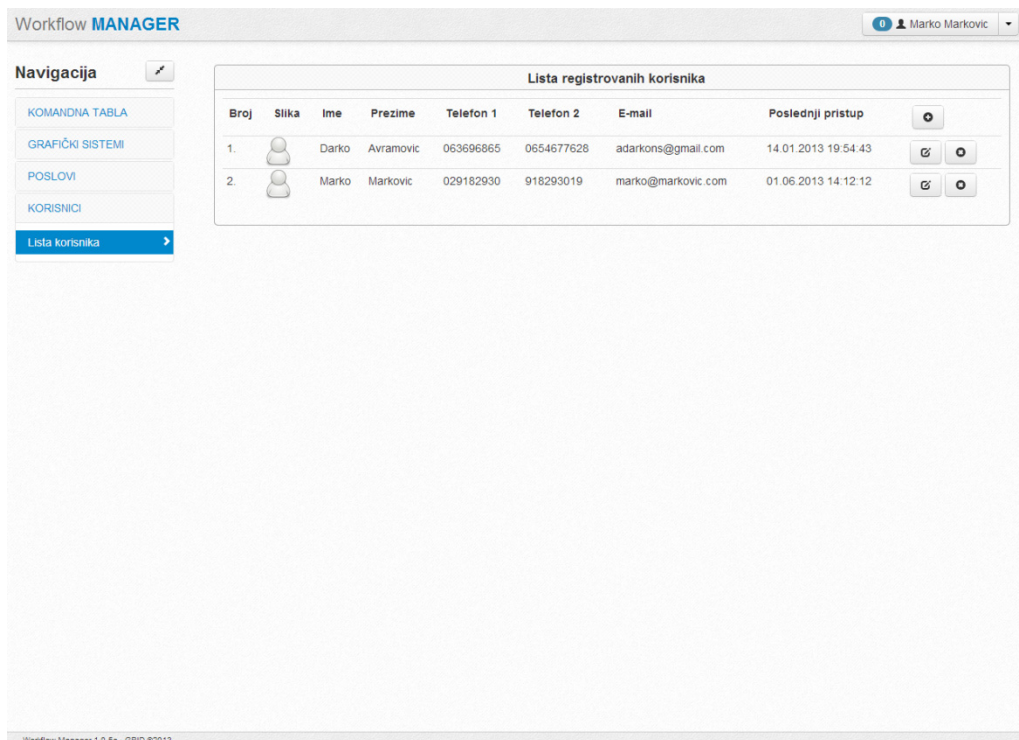
Nakon definicije prostorija i grafičkih sistema potrebno je odrediti poziciju grafičkih sistema u virtuelnim prostorijama. Na osnovu ovih pozicija GWFM sistem će generisati putanje tokova informacija i segmenata proizvoda. Pozicioniranje grafičkih sistema je proizvoljno, može se vezati za realni raspored proizvodnih sistema ali i ne mora (zavisi od potreba korisnika GWFM sistema). Primer interfejsa za pozicioniranje sistema prikazan je na slici 6.8. Odabirom prostorija iz grupe 'Prostorija' dobija se prikaz iste sa vezanim proizvodnim sistemima. Nakon pozicioniranja sistema prelazi se na drugu (ostale) prostorije dok se svi proizvodni sistemi ne pozicioniraju na odgovarajuću poziciju.



Slika 6.8: Određivanje pozicije sistema

Modul 'Korisnici' ima zaduženja definisanja i upravljanja korisničkim nalogima (slika 6.9).

Lista korisnika sadrži lične i korisničke podatke o korisniku (slika 6.10). U lične podatke spadaju lične informacije, slika i kontakt informacije. Korisničke podatke čine podaci za pristup sistemu, uloga korisnika, indikatori aktivnosti i blokade i veza sa proizvodnim sistemom. Korisnici mogu biti operatori uprave (Administratori) ili operatori proizvodnih sistema (Operater).



Slika 6.9: Lista registrovanih korisnika

The form is titled 'Registracija novog korisnika'. It is divided into two main sections:

- Korisnički podaci:**
 - Korisničko ime:
 - Lozinka:
 - Uloga:
 - Aktivan:
 - Blokiran:
 - Zadužen za:
- Lični podaci:**
 - Ime:
 - Prezime:
 - Telefon 1:
 - Telefon 2:
 - E-mail:
 - Slika: No file chosen

At the bottom, there are two buttons: 'Sačuvaj' and 'Odustani'.

Slika 6.10: Registracija novog korisnika sistema

Operatori uprave mogu koristiti isključivo serverski modul dok operatori grafičkih sistema mogu koristiti isključivo klijentski modul.

Poslednji i najkomplikovaniji modul jeste modul 'Poslovi'. Modul poseduje nekoliko celina:

- Pregled poslova
- Arhiva
- Unos
- Grupe poslova
- Zahtevi
- Praćenje poslova
- Klijenti

Prvi od nabrojanih delova biće objašnjen deo 'Unos' kako bi se stekao uvid u način kreiranja definicije posla (slika 6.11). Definicija posla se sastoji iz 6 celina. Svaka celina predstavlja skup na poseban način srodnih informacija. Te celine su:

- Osnovni podaci vezani za posao
- Dokumenti vezani za posao
- Segmenti proizvoda
- Priprema
- Štampa
- Završna grafička obrada

U delu 'Osnovi podaci vezani za posao' unose se osnovne informacije koje određuju posao (Tabela 6.1).

Tabela 6.1: Podaci iz grupe 'Osnovni podaci vezani za posao'

	Podatak	Opis
1.	Naziv	Određuje naziv posla. Predstavlja tekstualnu vrednost
2.	Evidencioni broj	Određuje evidencioni broj posla. Predstavlja kombinaciju slova i brojeva (u zavisnosti od potreba)
3.	Grupa	Određuje grupu kojoj posao pripada. Grupa mora biti ranije uneta a detaljnije objašnjenje o grupama proizvoda dato je u nastavku poglavlja
4.	Datum i planirani završetak	Datum početka odnosno rok završetka posla
5.	Tiraž	Određuje ukupan tiraž
6.	Obim	Određuje obim proizvoda
7.	Klijent	Vezuje posao za klijenta
8.	Obrezan format	Određuje obrezan format proizvoda
9.	Neobrezan format	Određuje neobrezan format proizvoda
10.	Aktivan	Određuje da li je posao započet, odnosno da li je dostupan za započinjanje
11.	Dostupan	Određuje da li je posao dostupan za preuzimanje od strane operatera proizvodnje
12.	Arhiviran	Određuje da li je posao arhiviran nakon završetka
13.	Obaveštenje klijentu	Određuje da li će klijent biti obavešten o završetku posla. Obaveštenje može biti u obliku e-mail ili sms poruke

Podaci iz tabele 6.1 predstavljaju esencijalne informacije o poslu. Ove informacije koriste se u više delova GWFM sistema.

Workflow **MANAGER** Marko Markovic

Navigacija

- KOMANDNA TABLA
- GRAFIČKI SISTEMI
- POSLOVI
 - Pregled poslova
 - Arhiva
 - Unos**
 - Grupe poslova
 - Praćenje poslova
 - Klijenti
- KORISNICI

Unos novog posla

Osnovni podaci za posao

Naziv posla:	<input type="text" value="Knjiga 1"/>	Obrezan format:	<input type="text"/>
Evidencioni broj:	<input type="text"/>	Neobrezan format:	<input type="text"/>
Grupa:	<input type="text" value="Knjiga FTN"/>	Aktivan:	<input type="checkbox"/>
Datum:	<input type="text"/>	Dostupan za preuzimanje:	<input type="checkbox"/>
Planiran završetak posla:	<input type="text"/>	Arhiviran:	<input type="checkbox"/>
Tiraž:	<input type="text"/>	Obaveštenje klijentu:	<input type="checkbox"/>
Obim:	<input type="text"/>		
Klijent:	<input type="text" value="GOMEX trejd"/>		

Dokumenti vezani za posao

Upload: Document con...orkflow.pdf

Document-configuration-control-processes-captured-in-a-workflow.pdf

Segmenti proizvoda

Dodaj novi segment:

Priprema

Dodaj novu operaciju:

Faza: Priprema	<input type="text" value="Korice"/>	na:	<input type="text" value="Radna stanica 1"/>	za (vreme):	<input type="text" value="4"/>
Faza: Priprema	<input type="text" value="Blok"/>	na:	<input type="text" value="Radna stanica 2"/>	za (vreme):	<input type="text" value="4"/>
Faza: CTP	<input type="text" value="Ceo proizvod"/>	na:	<input type="text" value="CTP 1"/>	za (vreme):	<input type="text" value="3"/>

Napomena:

Štampa

Dodaj novu operaciju:

Faza: Štampa	<input type="text" value="Korice"/>	na:	<input type="text" value="Rapida 8"/>	za (vreme):	<input type="text" value="3"/>
Faza: Štampa	<input type="text" value="Blok"/>	na:	<input type="text" value="Performa 74"/>	za (vreme):	<input type="text" value="7"/>

Napomena:

Završna grafička obrada

Dodaj novu operaciju:

Faza: Savijanje	<input type="text" value="Blok"/>	na:	<input type="text" value="Savijacica 1"/>	za (vreme):	<input type="text" value="4"/>
Faza: Sakupljanje	<input type="text" value="Blok"/>	na:	<input type="text" value="Sakupljacica 1"/>	za (vreme):	<input type="text" value="4"/>
Faza: Povez	<input type="text" value="Ceo proizvod"/>	na:	<input type="text" value="Binder 1"/>	za (vreme):	<input type="text" value="4"/>
Faza: Rezanje	<input type="text" value="Ceo proizvod"/>	na:	<input type="text" value="Noz 1"/>	za (vreme):	<input type="text" value="3"/>

Isporuka:

Napomena:

Workflow Manager 1.0.5a - GRID ©2013

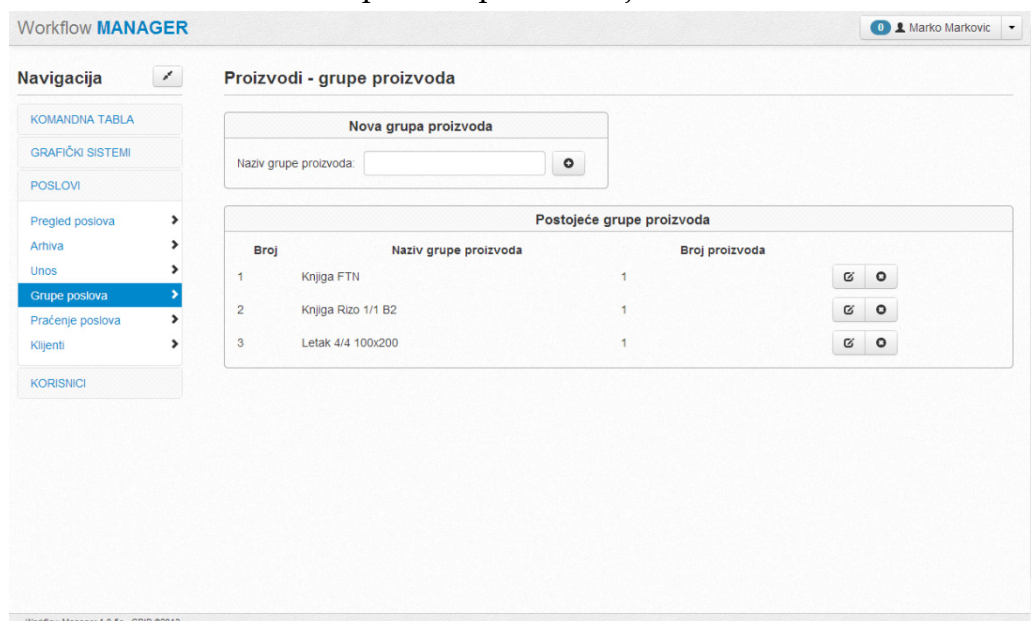
Slika 6.11: Unos nove definicije posla

Drugu celinu definicije posla čine 'Dokumenti vezani za posao'. Ovaj deo definicije zadužen je za upravljanje dokumentima koji se pridružuju poslu i koji nose detaljnije informacije o strukturi i osobinama proizvoda. Kao obavezan dokument u ovoj kategoriji se očekuje jedan oblik radnog naloga. Ostali dokumenti mogu biti dodati prema potrebi ili posebnim zahtevima. Ograničenja formata dokumenata ne postoje prema modelu.

Treća celina je zadužena sa segmentiranje proizvoda. U ovom delu proizvod je moguće razdvojiti na odgovarajuće delove (segmente) u zavisnosti od tehnološkog postupka izrade samog proizvoda. Iskustvo takođe predstavlja jedan od faktora koji odlučuju konačan izgled liste segmenata.

Ostale tri celine funkcionišu na identičan način. Svaka od celina predstavlja jednu od faza proizvodnje i u sebi sadrži definicije operacija kroz koje proizvod prolazi. Definicija operacije podrazumeva vrstu operacije koja se obavlja, na kom segmentu proizvoda se operacija odvija, na kom tehnološkom sistemu i za koje predviđeno vreme. Ovi podaci predstavljaju referentnu vrednost za GWFM.

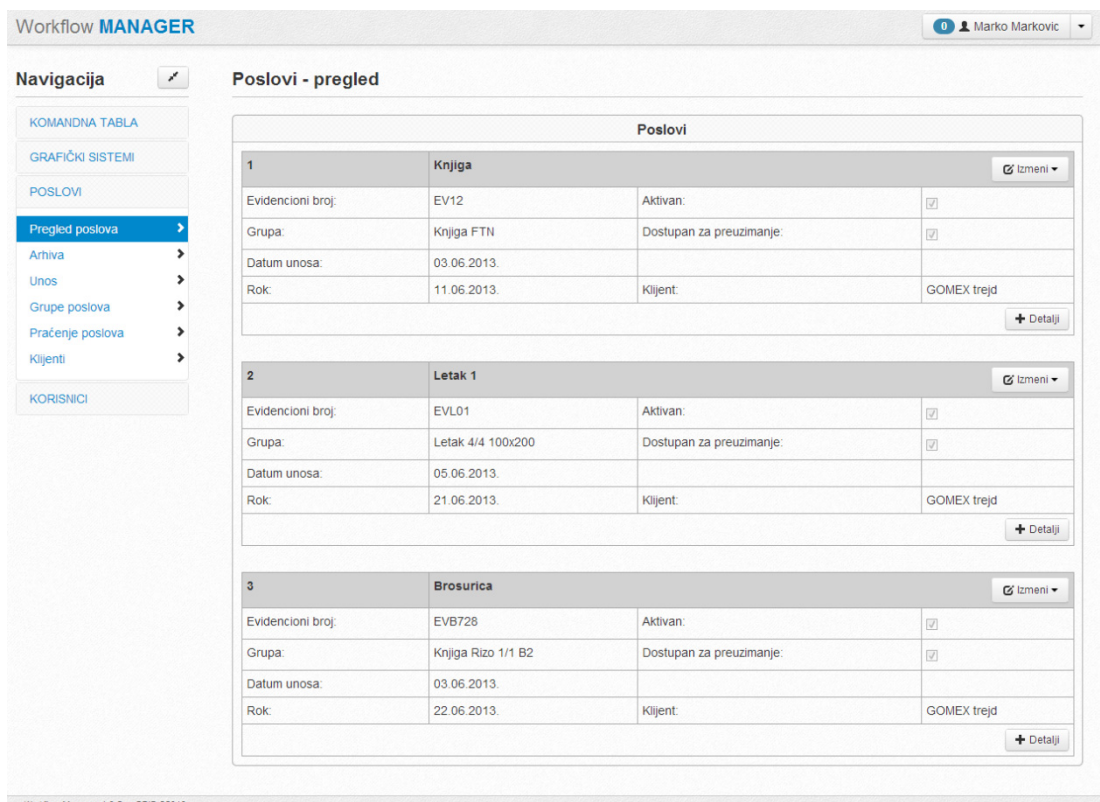
Deo 'Grupe poslova' (slika 6.12) predstavlja interfejs za definisanje grupa proizvoda. Naime, proizvodi se mogu posmatrati pojedinačno ili grupno. Na osnovu toga moguće je izvući različite zaključke vezane za, na primer, prosečno, maksimalno ili minimalno vreme završetka procesa proizvodnje i slično.



Slika 6.12: Definisane grupe poslova

Grupe se odlikuju samo imenom. Pored imena, u pregledu grupa nalazi se i broj proizvoda koji su svrstani u odgovarajuću grupu.

'Pregled poslova' (slika 6.13) daje uvid u trenutne aktivne poslove. Poslovi mogu biti započeti, nezapočeti ili završeni. Tabela prikaza posla daje osnovne informacije koje se mogu proširiti detaljima (slika 6.14).



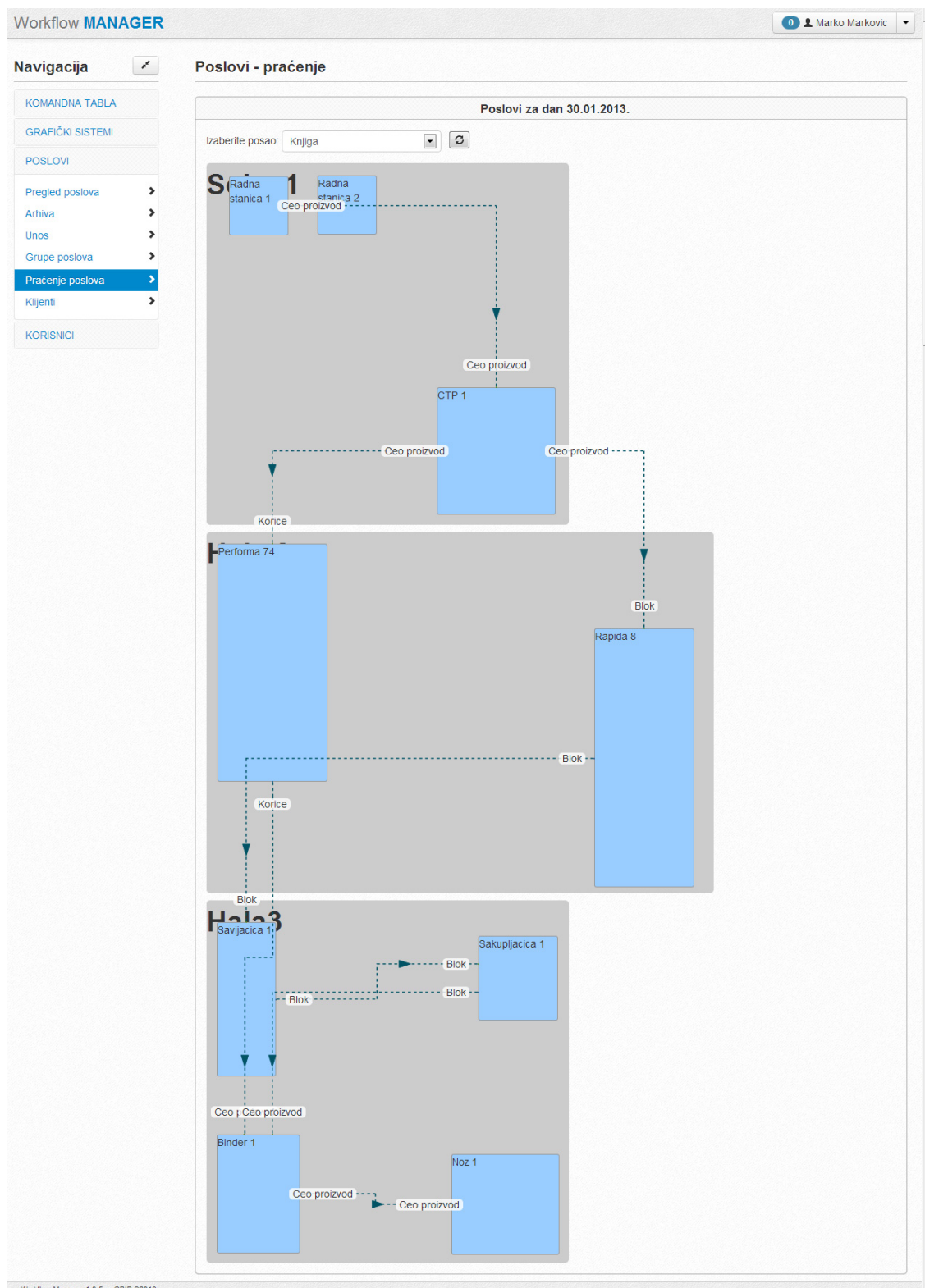
Slika 6.13: Pregled poslova

Detalji obuhvataju informacije o dokumentima vezanim za posao kao i trenutni status posla. Pod statusom se podrazumeva status segmenata proizvoda kroz faze procesa izrade.

2		Letak 1		<input type="checkbox"/> Izmeni	
Evidencioni broj:	EVL01	Aktivan:	<input checked="" type="checkbox"/>		
Grupa:	Letak 4/4 100x200	Dostupan za preuzimanje:	<input checked="" type="checkbox"/>		
Datum unosa:	05.06.2013.				
Rok:	21.06.2013.	Klijent:	GOMEX trejd		
<input type="button" value="+ Detalji"/>					
Tiraz:	333	Segmenti	Sistem	Status	Preostalo
Obim:	1	Ceo proizvod	Radna stanica 2	<input checked="" type="checkbox"/>	-
Dokumenti vezani za posao:	1. SGA-invitation.pdf	Ceo proizvod	CTP 1	<input checked="" type="checkbox"/>	-
		Ceo proizvod	Performa 74	<input checked="" type="checkbox"/>	-
		Ceo proizvod	Noz 1	<input checked="" type="checkbox"/>	-

Slika 6.14: Detaljan prikaz posla



Na slici 6.14 dat je primer proizvoda koji je završen ali nije arhiviran. Status segmenata sadrži notifikacije koje označavaju završen proces. Preostalo vreme takođe nije dostupno.

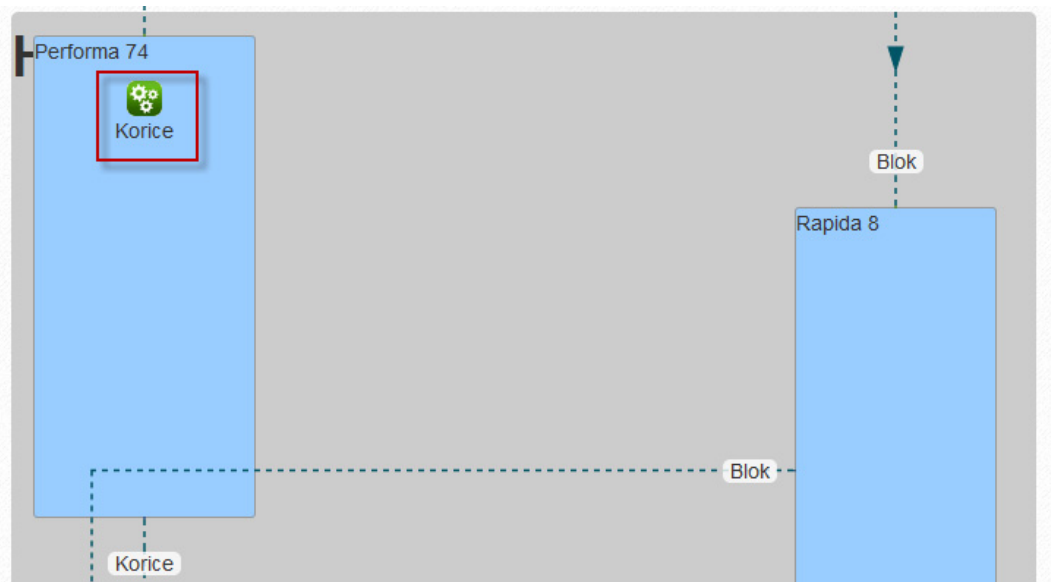


Slika 6.15: Praćenje statusa poslova po segmentima i fazama procesa

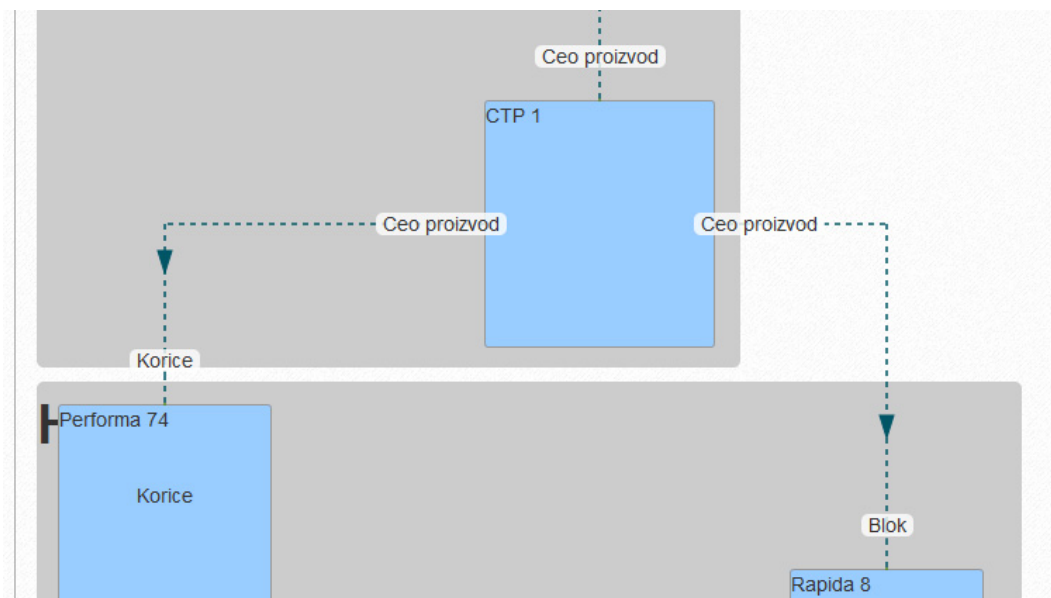
'Praćenje poslova' (slika 6.15) je celina koja se bavi pružanjem mogućnosti praćenja poslova, tačnije stanja ili faze procesa koja je u toku. Na slici 6.15 dat je primer praćenja statusa jedne knjige. Dat je prikaz svih ranije definisanih prostorija sa tehnološkim sistemima prethodno raspoređenim unutar odgovarajućih prostorija. Odabirom odgovarajućeg posla sistem generiše putanje kretanja segmenata

proizvoda kroz proizvodni sistem i daje uvid u stanje pojedinih faza procesa izrade. Na primeru sa slike 6.15, niti jedna faza procesa izrade nije u toku. Statusi procesa izrade segmenata proizvoda mogu biti sledeći (primer dat na slici 6.16):

- bez simbola - faza procesa nije započeta ili je privremeno zaustavljena.
- simbol  - označava da je faza procesa izrade segmenta u toku.
- simbol  - označava da je faza procesa izrade segmenta završena.



Slika 6.16: Primer statusa (u toku)

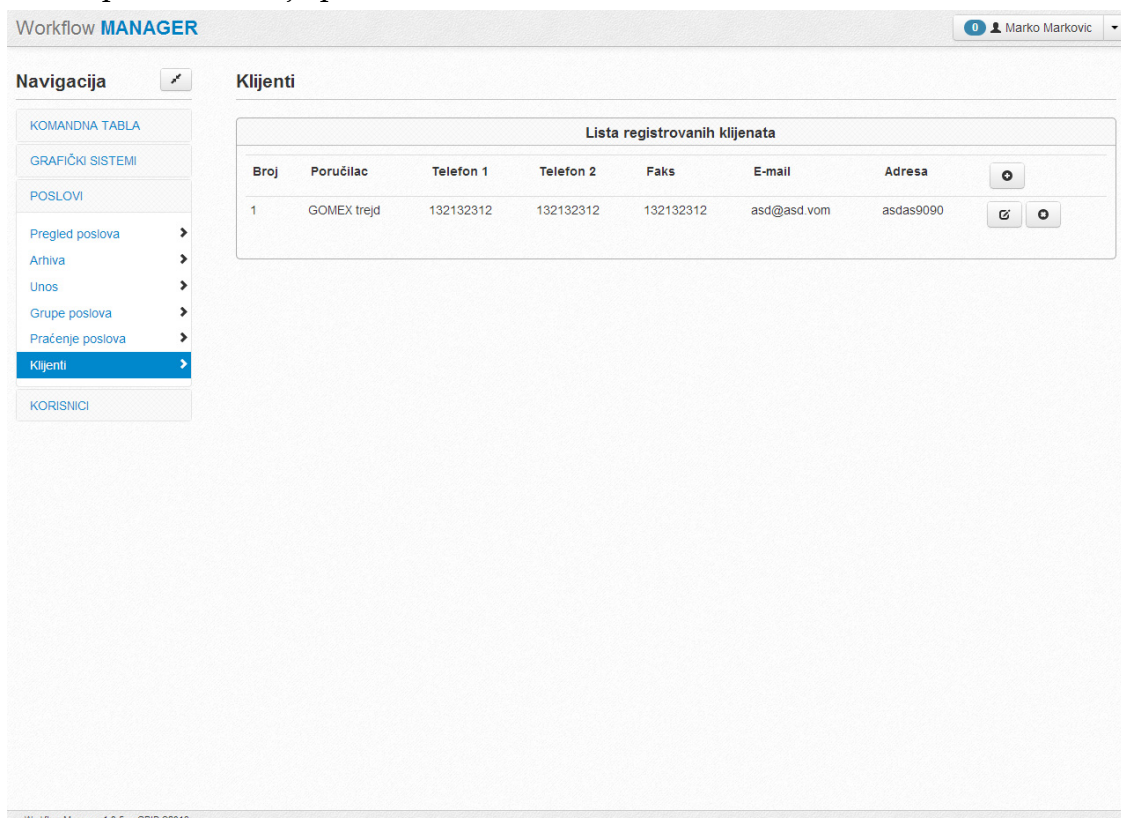


Slika 6.17: Grananje celog proizvoda na segmente koji se posebno obrađuju

Slika 6.17 daje prikaz deljenja celog proizvoda na segmente koji se u narednoj fazi procesa odvojeno obrađuju. To je jedna od glavnih uloga ove celine - lakše shvatanje putanje i praćenje proizvoda kroz proces proizvodnje.

Poslednja celina modula 'Poslovi' jeste celina 'Klijenti' (slika 6.18). Sadrži pregled i daje mogućnost upravljanja klijentima. Definicija klijenta sadrži

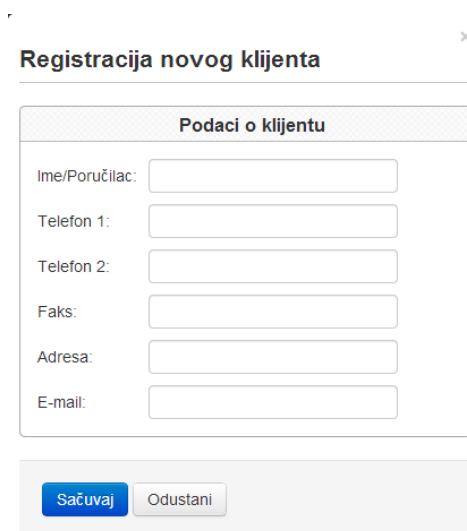
najosnovnije kontakt informacije o naručiocu posla (slika 6.19). Klijent mora biti definisan pre definisanja posla.



The screenshot shows the 'Workflow MANAGER' interface. On the left is a navigation menu with categories like 'KOMANDNA TABLA', 'GRAFIČKI SISTEMI', 'POSLOVI', and 'KORISNICI'. The 'POSLOVI' section is expanded to show 'Klijenti'. The main area displays a table titled 'Lista registrovanih klijenata' with the following data:

Broj	Poručilac	Telefon 1	Telefon 2	Faks	E-mail	Adresa	
1	GOMEX trejd	132132312	132132312	132132312	asd@asd.vom	asdas9090	[Share] [More]

Slika 6.18: Pregled klijenata



The screenshot shows a form titled 'Registracija novog klijenta'. It contains a section 'Podaci o klijentu' with the following fields:

- Ime/Poručilac:
- Telefon 1:
- Telefon 2:
- Faks:
- Adresa:
- E-mail:

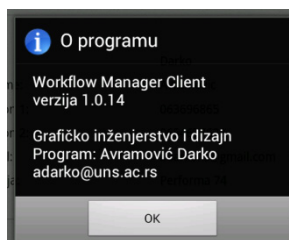
At the bottom of the form are two buttons: 'Sačuvaj' (Save) and 'Odustani' (Cancel).

Slika 6.19: Interfejs za unos klijenata

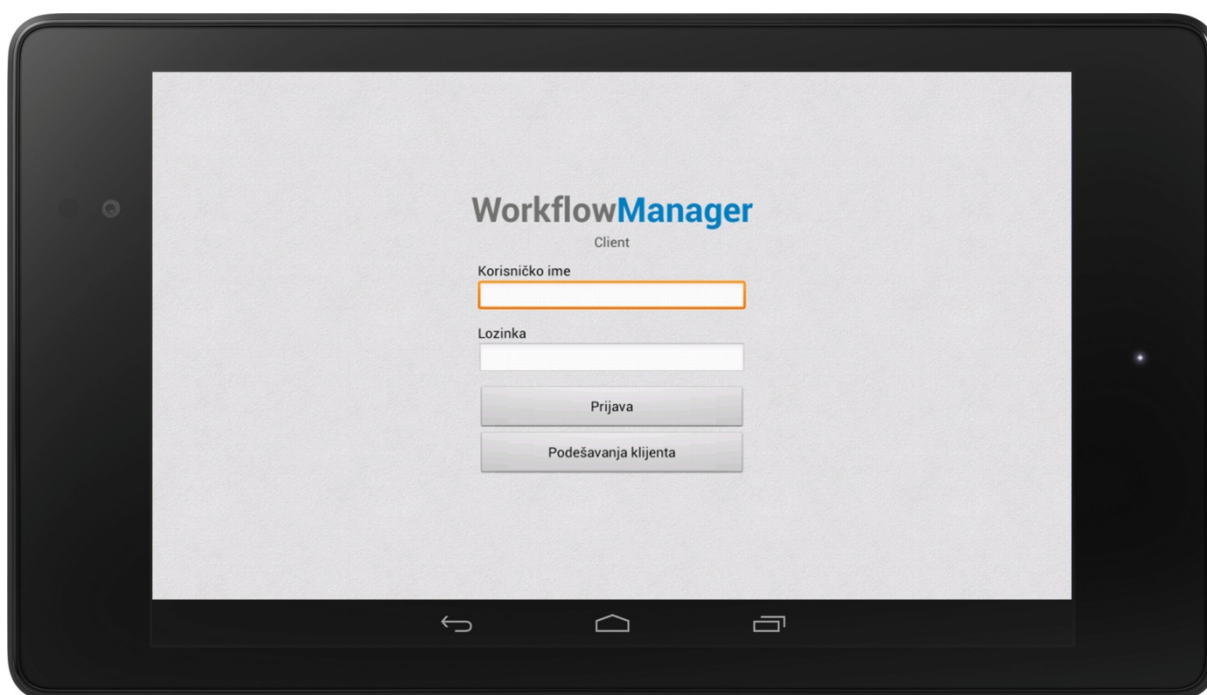
Ovim je priveden kraju opis funkcija i mogućnosti serverskog modula GWFM sistema. U nastavku teksta biće prikazane funkcije klijentskog modula.

6.1.2 Klijentski modul

Klijentski modul (slika 6.20 i dalje) poseduje manji opseg mogućnosti od serverskog modula. Uloga mu je isključivo upravljanje pojedinim informacijama posla koje ne uključuju menjanje definicije posla, dokumenata i sastava faza procesa izrade. Klijentski modul koriste isključivo Operatori (uloga ranije definisana). Klijentski modul funkcioniše u okviru mobilnih uređaja čija specifikacija je data na početku poglavlja.



Slika 6.20: Info o GWFM klijentu

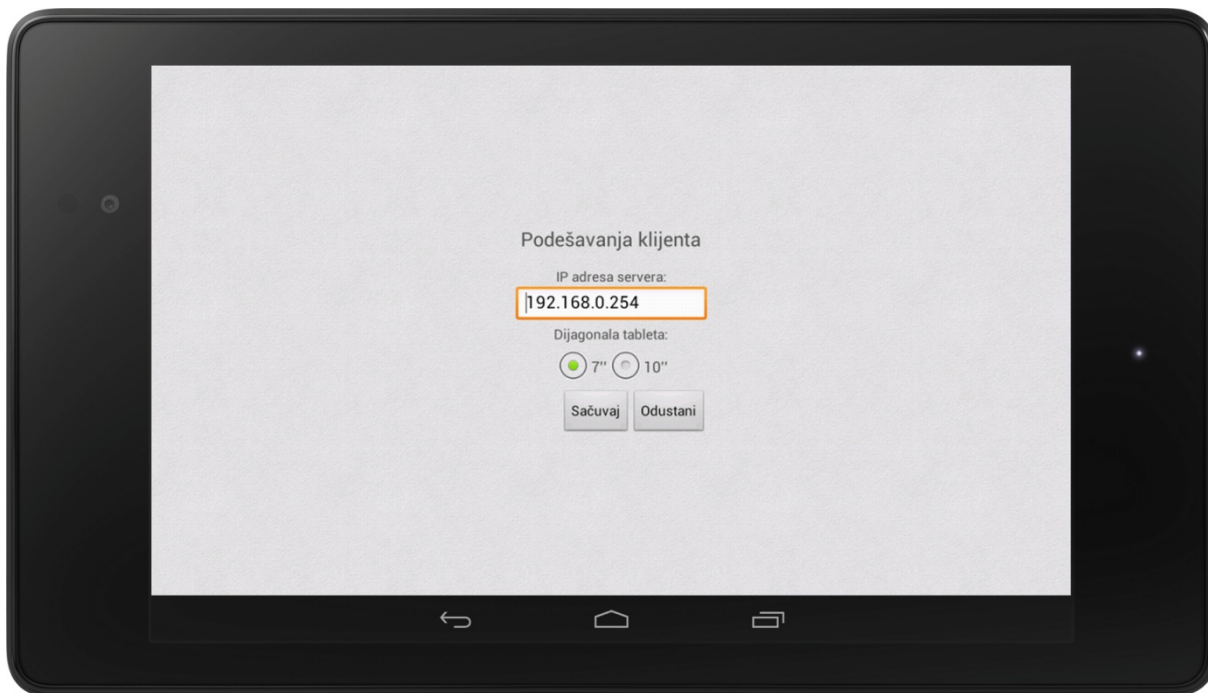


Slika 6.21: Prijava u sistem

Po pokretanju GWFM klijenta korisnik biva doveden na ekran za prijavu (slika 6.21). Putem ovog ekrana pristup imaju Operateri. Pored prijave Operateri mogu i da konfigurišu klijent aplikaciju za pristup serveru. Potrebno je uneti adresu servera i izabrati dužinu dijagonale ekrana klijentskog uređaja (slika 6.22). To predstavljaju inicijalna i obavezna podešavanja klijenta kako bi isti mogao da se koristi. Bez ovih podešavanja korišćenje klijenta nije moguće.

Po prijavi u sistem putem klijentske aplikacije, Operater dobija spisak ličnih informacija u obliku ekrana 'Moji podaci', što je prikazano na slici 6.23. U pomenute informacije spadaju lični i kontakt podaci kao i pozicija operatera (dodeljeni

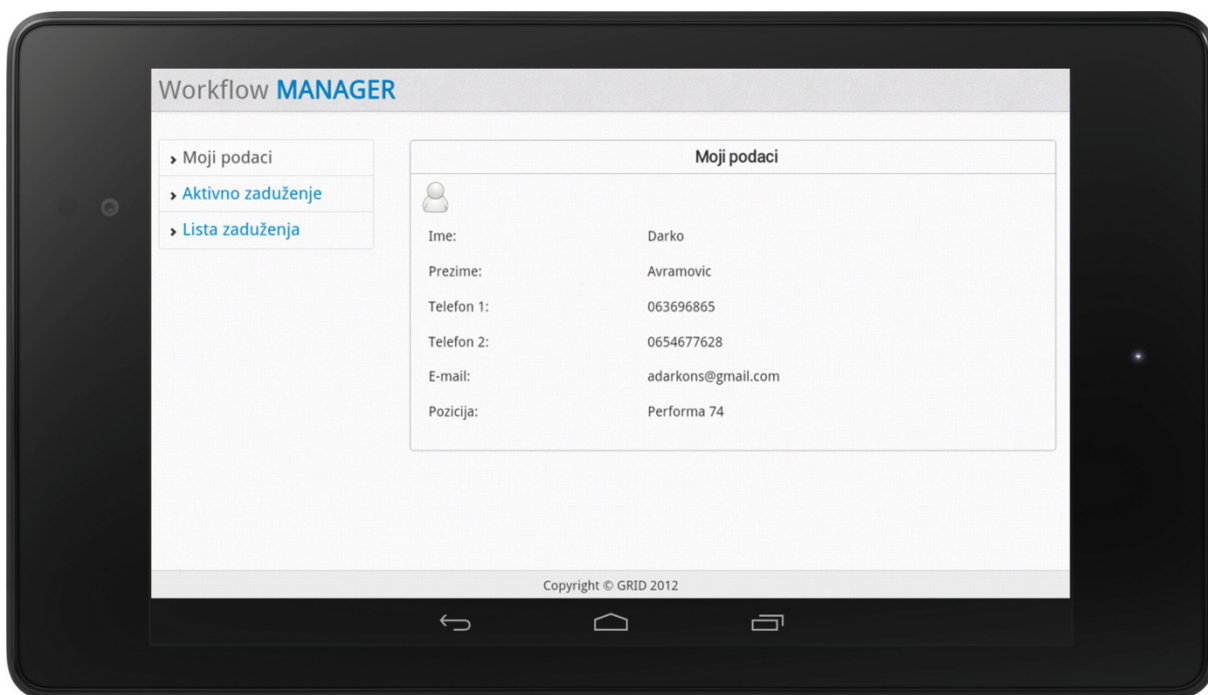
tehnološki sistem). Ovi podaci su informativnog karaktera i nije ih moguće menjati putem interfejsa.



Slika 6.22: Podešavanja klijenta

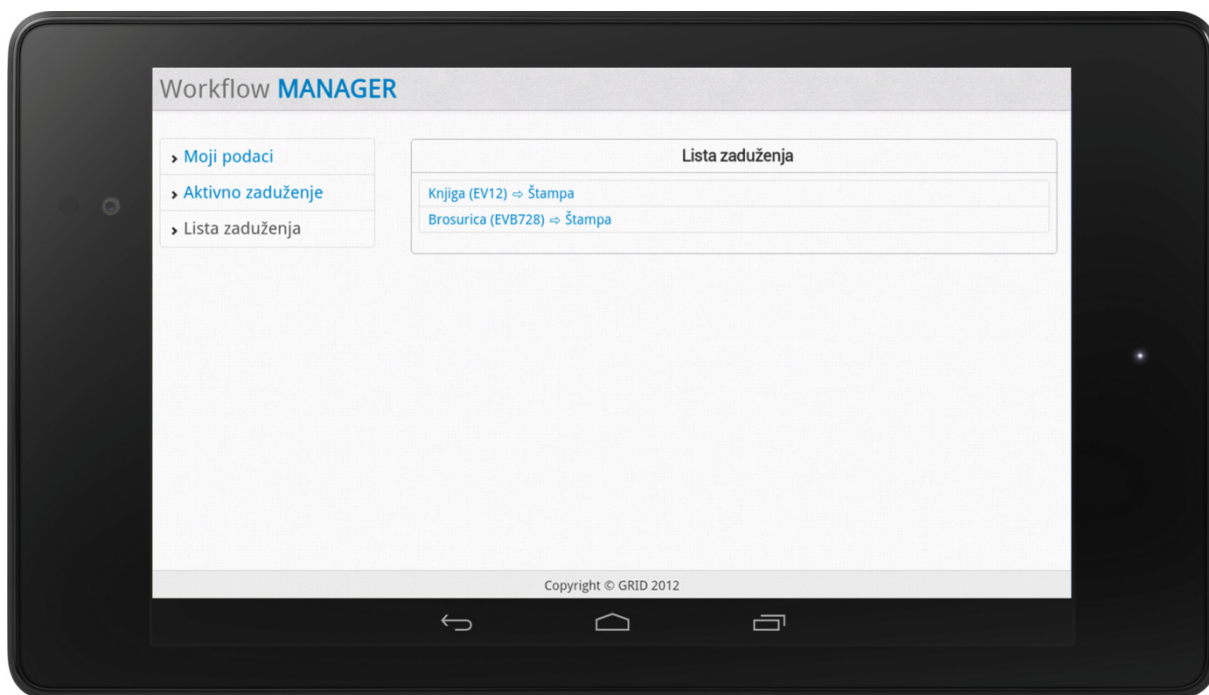
Nakon uvida u podatke korisnik sistema može da izabere jednu od tri preostale stavke iz navigacije koja je pozicionirana u levom delu ekrana klijentskog uređaja. Dostupne opcije su:

- Aktivno zaduženje
- Lista zaduženja



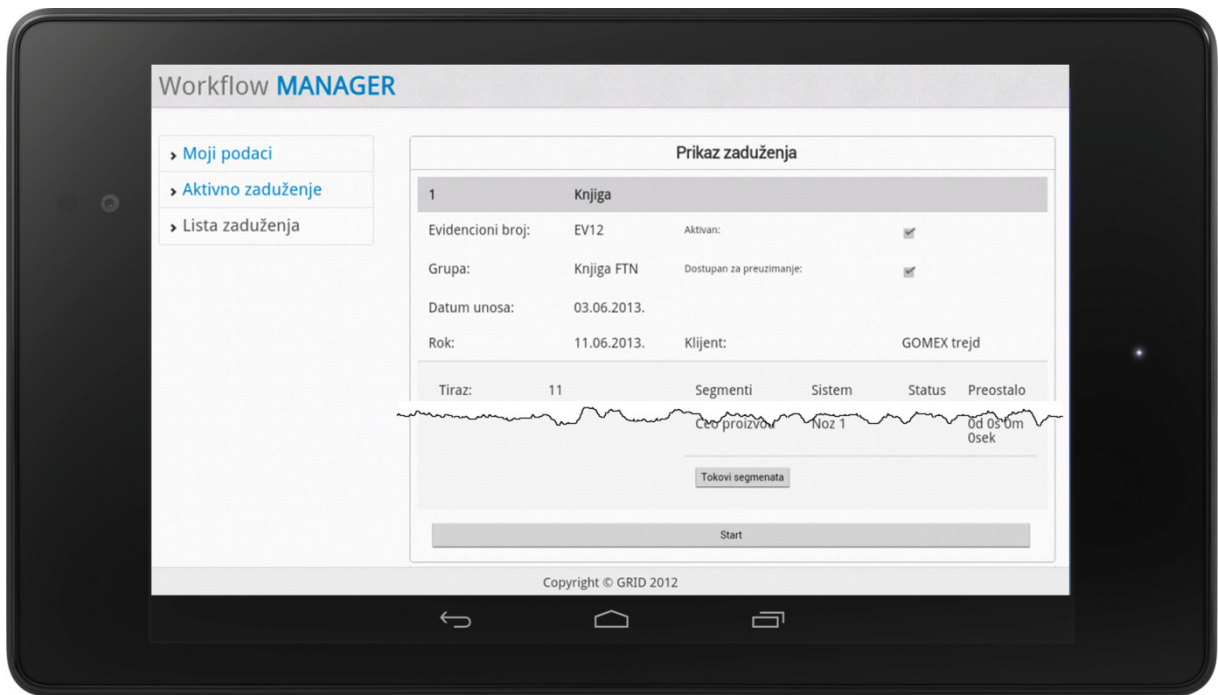
Slika 6.23: Prikaz 'Moji podaci'

Odabirom opcije *Lista zaduženja* (slika 6.24) korisnik dobija spisak svojih zaduženja. Zaduženje dodeljuje Administrator (operator uprave) pri pravljenju definicije posla. Praktično, zaduženje se ne vezuje za Operatora nego sa tehnološki sistem.

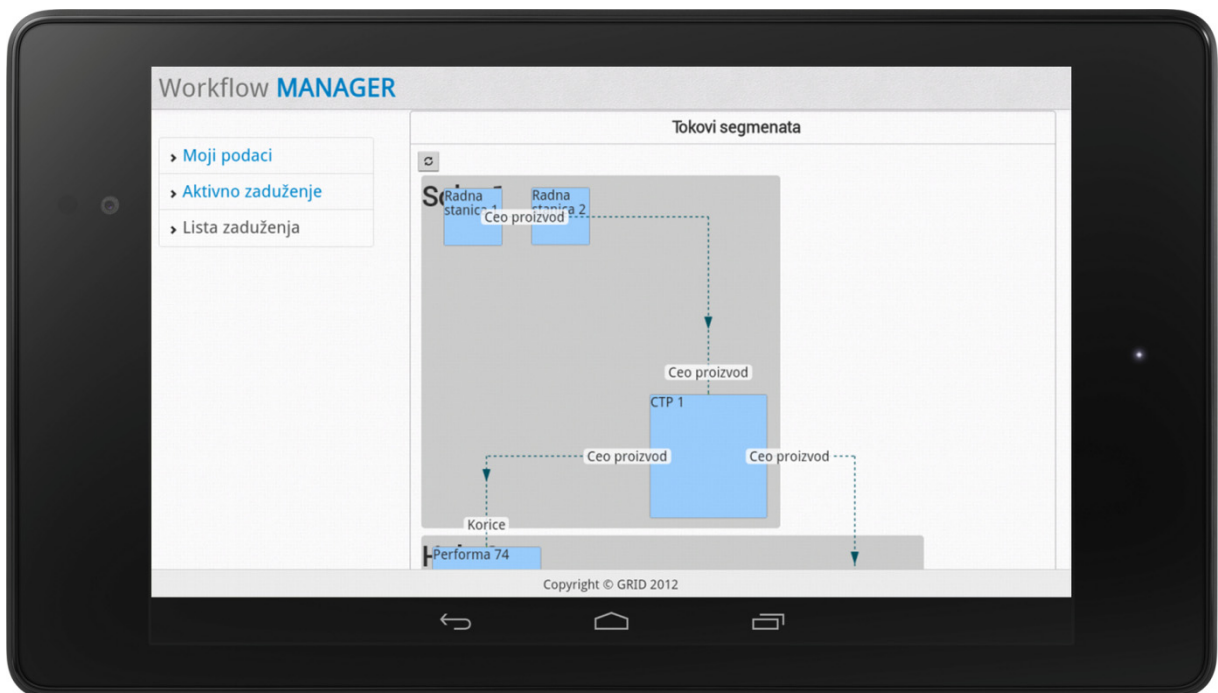


Slika 6.24: Lista zaduženja

Preko tehnološkog sistema posao se posredno vezuje za operatora. Nakon uvida u listu korisnik može da izabere jedno od zaduženja koje mu je dodeljeno. Odabirom zaduženja na ekranu se pojavljuju detaljne informacije o poslu koji treba sprovesti (slika 6.25). Pomenuti prikaz daje podatke iste kao na slici 6.13 i 6.14 što znači da korisnik dobija potpune informacije po zaduženju koje ga očekuje: kompletne osnovne informacije, pristup svim dokumentima vezanim za posao, uvid u protekle, započete ili nezapočete faze izrade, uvid u protekla i predviđena vremena za izradu segmenta proizvoda.

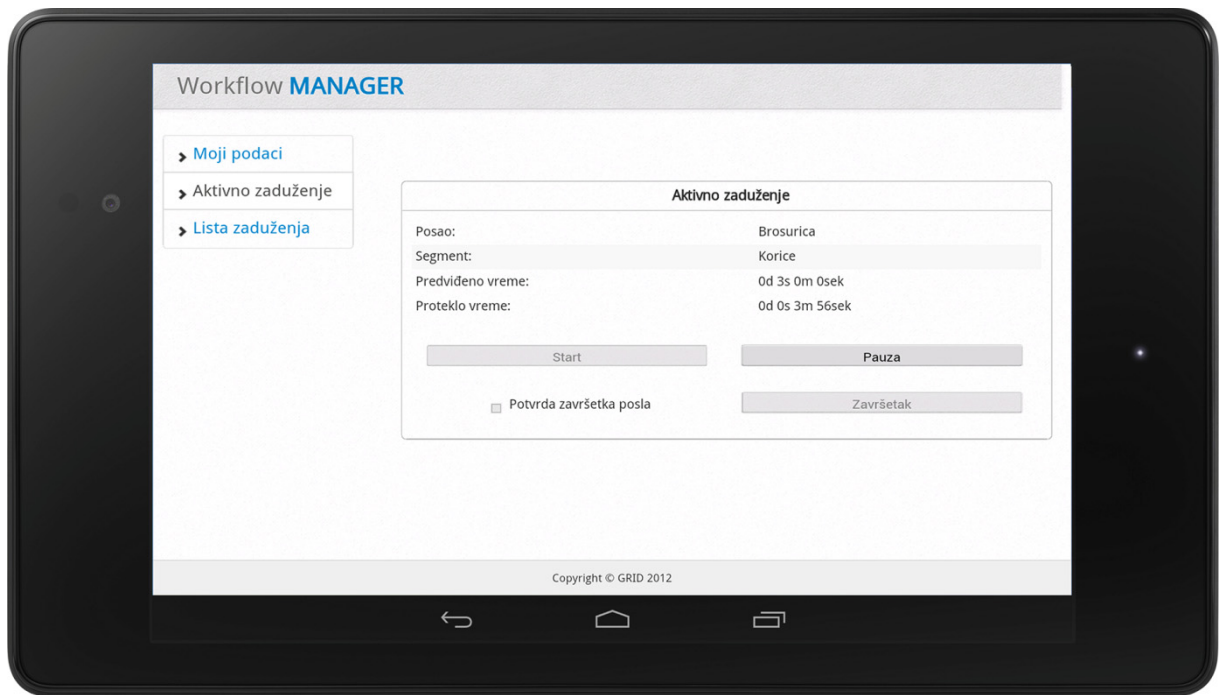


Slika 6.25: Pregled zaduženja



Slika 6.26: Praćenje tokova segmenata

Pored pomenutih stavki korisnik iz ovog prikaza može steći uvid i u tokove kretanja segmenata kroz proizvodni sistem (slika 6.26) kao i izvršiti započinjanje zaduženja pritiskom na taster *Start* (slika 6.25).



Slika 6.27: Aktivno zaduženje

Pritiskom na taster *Start* operater biva preusmeren na ekran *Aktivno zaduženje* (slika 6.27). Na pomenutom ekranu korisnik ima kontrolu svog zaduženja koje je upravo započeo. Dostupan je naziv posla, segment koji se obrađuje, predviđeno vreme trajanja i ukupno proteklo vreme. Aktivno zaduženje se u svakom momentu može pauzirati i ponovo nastaviti bilo kada. Ukoliko je proces izrade segmenta proizvoda završen, aktivno zaduženje se može i završiti. Za ovu potrebu postoji posebno dugme 'Završi' čiju aktivaciju je prethodno potrebno potvrditi check poljem 'Potvrda završenog posla'. Ovo je jedna mera sigurnosti kako ne bi došlo do slučajnog završetka posla pošto posao koji se završi više nije moguće ponovo započeti. Moguće je samo nastaviti pauziran posao. Ovo predstavlja i osnovnu razliku tastera 'Pauza' i 'Završi'.

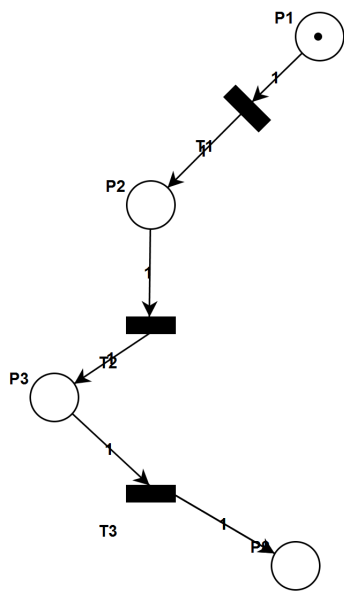
6.2 Verifikacija modela i simulacija rada sistema

U ovom poglavlju će biti prikazana verifikacija rada prototipskog rešenja GWFM sistema u laboratorijskim uslovima. Za potrebe simulacije kreirane su definicije laboratorijskih prostorija, tehnoloških sistema realnih laboratorijskih uslova i deset korisničkih naloga. Jedan korisnik predstavlja operatera uprave i upravlja serverskim modulom GWFM dok ostali nalozi imaju ulogu operatera tehnoloških sistema i upravljaju klijentskim modulima. Takođe, za potrebe simulacije rada prototipskog rešenja kreirane su odgovarajuće definicije poslova. Definicije poslova smeštene su u odgovarajuće grupe. Radni tokovi kreiranih definicija poslova u obliku tokova segmenata proizvoda prikazani su na slici 6.28. Tabela 6.2 daje detaljnije obrazloženje slike 6.28. Iz pomenute tabele se vidi da prvi proizvod (6.28a)

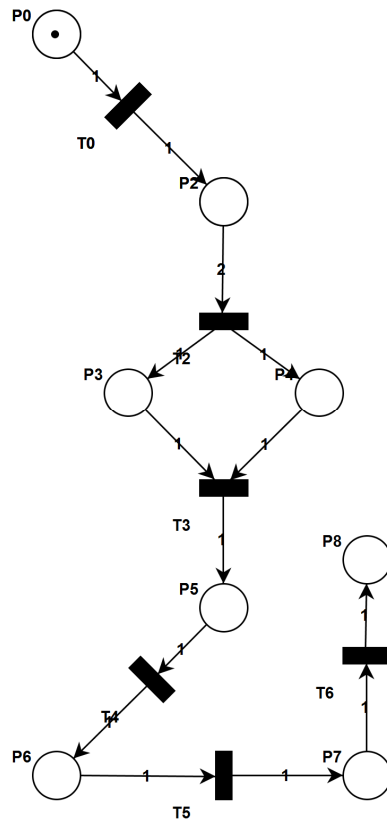
prolazi kroz 4 faze obrade tokom izrade, drugi proizvod (slika 6.28b) prolazi kroz 8 faza izrade, treći proizvod (slika 6.28c) prolazi kroz svega 3 faze, četvrti proizvod (slika 6.28d) kroz 8 i peti proizvod (slika 6.28e) kroz 8 faza. Pod 'fazom' se ne misli na faze izrade grafičkog proizvoda nego na pozicije u proizvodnom procesu. Segmentiranje proizvoda je prisutno kod proizvoda 2, 4 i 5.

Tabela 6.2: Parametri mreža sa slike 6.28

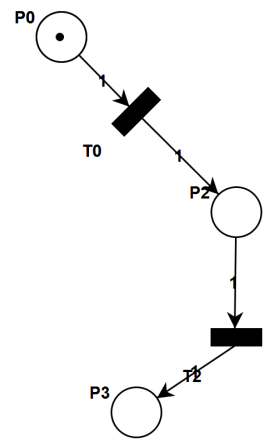
Br	Pozicija	Proizvod 1	Proizvod 2	Proizvod 3	Proizvod 4	Proizvod 5
1.	P0 - Radna stanica 1	-	x	x	x	x
2.	P1 - Radna stanica 2	x	-	-	-	x
3.	P2 - CTP	x	x	x	x	x
4.	P3 - Mašina za štampu 1	x	x	x	x	-
5.	P4 - Mašina za štampu 2	-	x	-	x	x
6.	P5 - Mašina za savijanje	-	x	-	x	x
7.	P6 - Mašina za sakupljanje	-	x	-	x	x
8.	P7 - Mašina za povez	-	x	-	x	x
9.	P8 - Mašina za rezanje	x	x	-	x	x



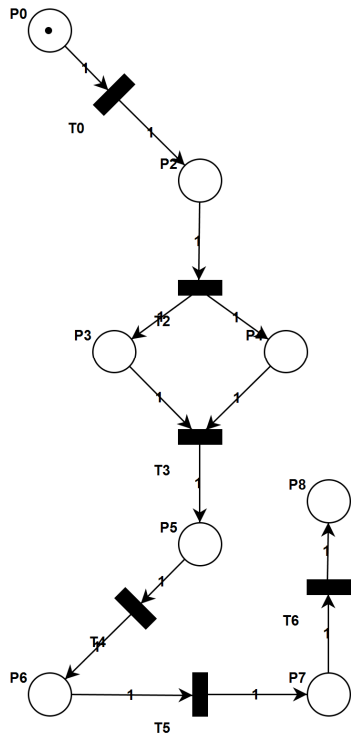
a) Proizvod 1



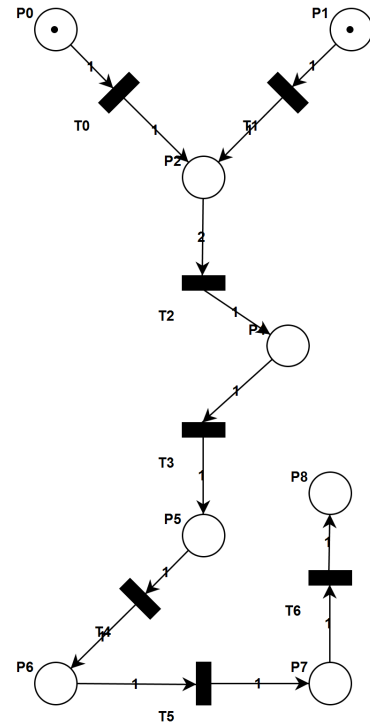
b) Proizvod 2



c) Proizvod 3



d) Proizvod 4



e) Proizvod 5

Slika 6.28: Radni tokovi prema definicijama posla

$M_0(1,0,0,0)$

↓ t_1

$M_1(0,1,0,0)$

↓ t_2

$M_2(0,0,1,0)$

↓ t_3

$M_3(0,0,0,1)$

a)

$M_0(1,0,0,0,0,0,0,0)$

↓ t_0

$M_1(0,1,0,0,0,0,0,0)$

↓ t_2

$M_2(0,0,1,1,0,0,0,0)$

↓ t_3

$M_3(0,0,0,0,1,0,0,0)$

↓ t_4

$M_4(0,0,0,0,0,1,0,0)$

↓ t_5

$M_5(0,0,0,0,0,0,1,0)$

↓ t_6

$M_6(0,0,0,0,0,0,0,1)$

b)

$M_0(1,0,0)$

↓ t_0

$M_1(0,1,0)$

↓ t_2

$M_2(0,0,1)$

c)

$M_0(1,0,0,0,0,0,0,0)$

↓ t_0

$M_1(0,1,0,0,0,0,0,0)$

↓ t_2

$M_2(0,0,1,1,0,0,0,0)$

↓ t_3

$M_3(0,0,0,0,1,0,0,0)$

↓ t_4

$M_4(0,0,0,0,0,1,0,0)$

↓ t_5

$M_5(0,0,0,0,0,0,1,0)$

↓ t_6

$M_6(0,0,0,0,0,0,0,1)$

d)

$M_0(1,1,0,0,0,0,0,0)$

↓ t_0

$M_1(0,1,1,0,0,0,0,0)$

↓ t_1

$M_2(0,0,2,0,0,0,0,0)$

↓ t_2

$M_3(0,0,0,1,0,0,0,0)$

↓ t_3

$M_4(0,0,0,0,1,0,0,0)$

↓ t_4

$M_5(0,0,0,0,0,1,0,0)$

↓ t_5

$M_6(0,0,0,0,0,0,1,0)$

↓ t_6

$M_6(0,0,0,0,0,0,1,0)$

e)

Slika 6.29: Markeri stanja mreža sa slike 6.28

Proizvodi sa čiji radni tokovi su prikazani na slici 6.28 su: pod a i c – prosti, pod b, d i e – složeni.

U nastavku su prikazani statusi sistema i mogućnosti praćenja efikasnosti i opterećenja sistema i učesnika proizvodnje u jednom momentu proizvodnog procesa. Preko prikazanih ekranskih interfejsa prati se broj sesija i vreme izrade proizvoda (slike 6.30 i 6.31).

Podaci u poslovima					
Broj	Naziv	Evid. broj	Predviđeno trajanje	Realno trajanje	Broj sesija
1	Posao 1	EVP1	00:10:00:00	00:10:47:33	7
2	Posao 2	EVP2	00:21:00:00	00:22:19:53	9
3	Posao 3	EVP3	00:11:00:00	00:12:19:56	4
4	Posao 4	EVP4	01:08:00:00	01:06:39:45	12
5	Posao 5	EVP5	01:01:00:00	00:23:32:31	11

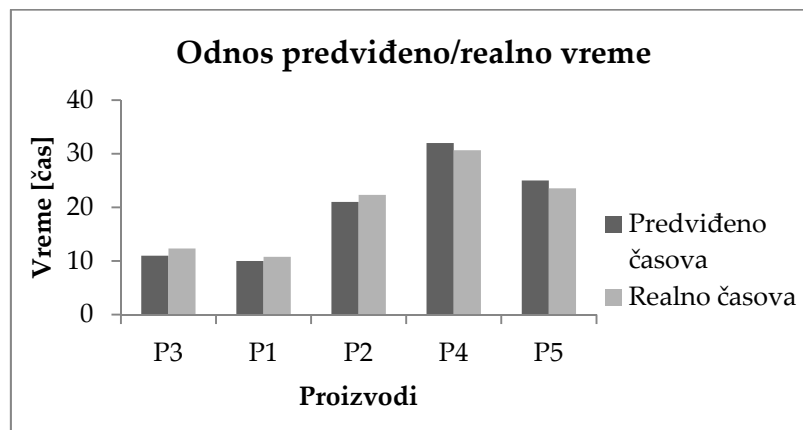
Slika 6.30: Prikaz podataka o proizvodnim procesima

Na slici 6.30 je prikaz predviđenih i ostvarenih vremena za svaki od poslova pojedinačno. Iz prikaza se može dobiti razlika predviđenih i realno ostvarenih vremena završetka poslova. Akumulacijom podataka tokom vremena dobili bi se znatno precizniji rezultati u slučaju proizvoda istih osobina. Sa druge strane moguće je pratiti i realna ostvarena vremena završetaka poslova kroz grupe koje su kreirane i prilagođene samo proizvodnom sistemu u kome se GWFm koristi (grafik 4 sa slike 6.31). Akumulacijom podataka vezanih za veći broj proizvoda u sklopu svake grupe dobijaju se sve preciznije informacije o grupi proizvoda.

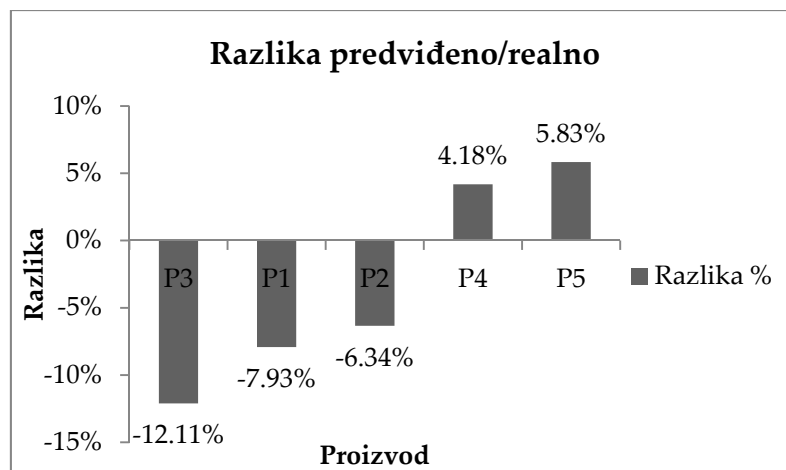
Na slici 6.32 dat je prikaz odnosa predviđenih i realnih ostvarenih vremena procesa izrade svakog od proizvoda. Može se zaključiti da odstupanja nisu velika. Odstupanja se kreću u opsegu od -12% do +~6%. Negativno odstupanje označava premašenje utvrđenog predviđenog vremena dok pozitivno odstupanje označava da je proizvod završen pre predviđenog vremenskog roka. Za primer, slika 6.33 daje prikaz svih odstupanja u procentualnom obliku. Sa ove slike možemo videti da je proizvod 5 završen za 5,83% manje vremena od predviđenog dok je proizvod 3 završen za 12,11% više vremena od predviđenog.



Slika 6.31: Statusi sistema u toku rada sistema

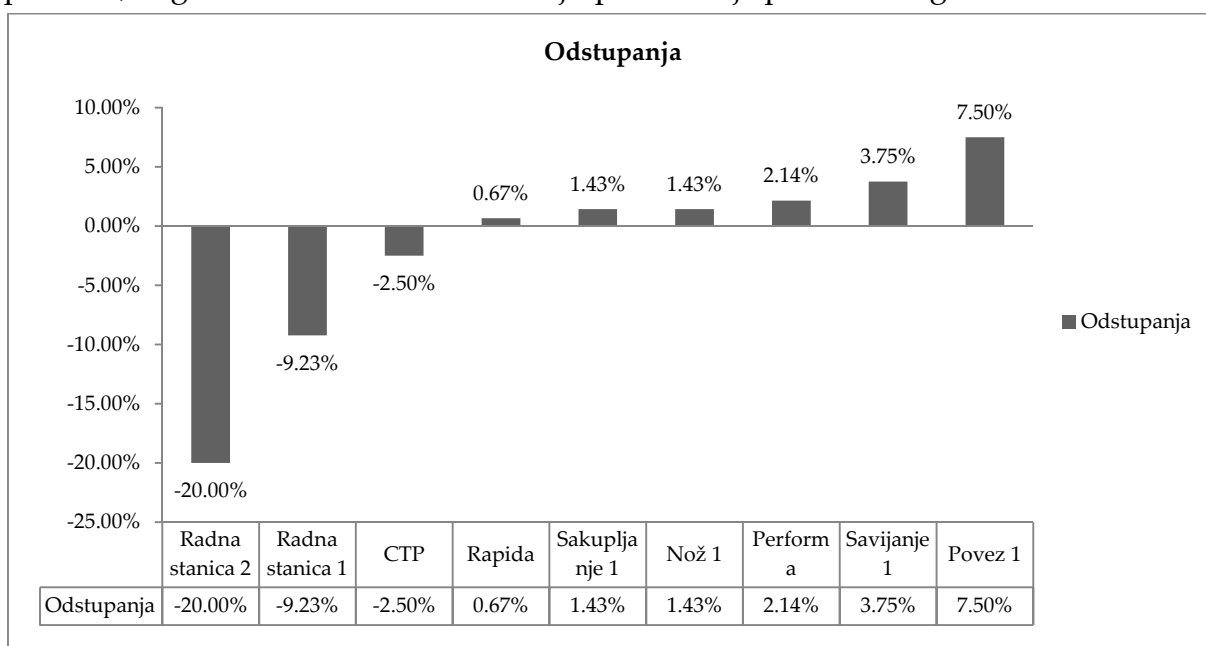


Slika 6.32: Odnos predviđenih i realno ostvarenih vremena za pojedinačne proizvode



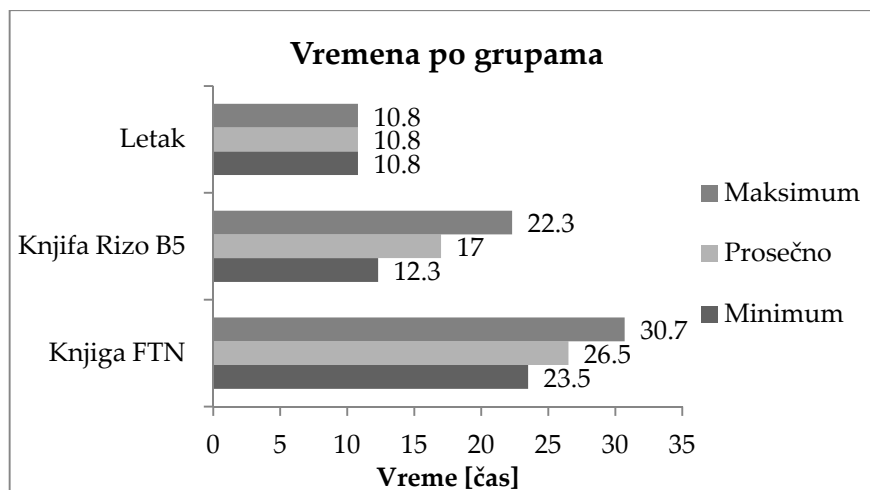
Slika 6.33: Odstupanja vremena realizacije pojedinačnih poslova

Na ekranskom prikazu sa slike 6.31 pored predviđenih i ostvarenih realnih vremena završetka poslova može se pratiti i opterećenje sistema. Slika 6.34 prikazuje predviđena i dosadašnja ili realna opterećenja svakog od sistema proizvodnog procesa. Vrednosti su prikazane u procentualnom obliku a, kao i na prethodnom primeru, negativne vrednosti označavaju premašenje predviđenog roka.



Slika 6.34: Prikaz odstupanja pri opterećenju sistema

Prema prikazu sa slike 6.34 može se utvrditi pojava premašenja rokova u delu pripreme za štampu što je indikator (za primer) ili problema sa tehnološkom opremom ili učesnicima procesa proizvodnje (pošto su operateri direktno povezani sa tehnološkim sistemom i mogu biti odgovorni za premašenje rokova), stoga treba posvetiti pažnju upravom tom segmentu, pronaći problem i otkloniti ga na prigodan način.



Slika 6.35: Ostvarena vremena po grupama proizvoda

Slika 6.35 daje prikaz ostvarenih vremena sada po grupama proizvoda. Grupe proizvoda mogu činiti proizvodi sličnih ili istih karakteristika. Popunjavanje grupa proizvodima vrši se na osnovu iskustva i potreba proizvodnog sistema. Akumuliranjem podataka dobijaju se tačnije informacije o proizvodnom procesu.

Pored primera praćenja stanja sistema koji su predstavljeni u ovom poglavlju moguće je razviti još sličnih primera koji bi kombinacijama različitih informacija davale potpunije i preciznije informacije o stanju sistema i proizvodnog procesa. Proširivanje mogućnosti sistema i njegova lakša integracija u proizvodno okruženje i druge sisteme predstavlja prioritet u daljim istraživanjima.

7 ZAKLJUČAK

Za prikaz dostignuća u razvoju modela informacionih sistema u prvom delu disertacije se daje pregled teorijskih osnova u ovoj oblasti. Opisane su osnove upravljanja proizvodnjom i grafičkim procesima. Date su osnovne definicije proizvodnje, proizvodnih sistema, grafičkih procesa, grafičkih sistema, upravljanja, regulacije, kao i osnovne smernice pri upravljanju.

Pored osnova upravljanja grafičkim procesima, u nastavku su date i osnovne informacije o informacionim sistemima, osnovne definicije i osobine informacionih sistema kao i definicije modela. Pored osnovnih informacija o informacionim sistemima prikazani su i principi i modeli razvoja informacionih sistema, objašnjeni su i neki jezici koji se koriste za modelovanje i metode analize.

U sklopu povezanih istraživanja predstavljena su određena softverska rešenja i izvršena je analiza poređenja tih rešenja sa modelom koji predlaže disertacija. Poređenje je izvršeno na osnovu posebno odabranih atributa.

Teorijski doprinos disertacije jeste apstraktni model informacionog sistema koji doprinosi boljoj komunikaciji i raspodeli informacija i zaduženja unutar jednog proizvodnog sistema. Za razliku od svih ostalih rešenja model koji predlaže disertacija daje novi pristup komunikaciji unutar proizvodnog sistema upotrebom novih tehnologija i modernih načina komunikacije. Prednosti ovakvog modela su:

- model daje mogućnost predviđanja troškova na osnovu akumuliranih podataka koji se prikupljaju u procesu proizvodnje i time daje realno stanje sistema za razliku od vrednosti dobijenih na osnovu normativa ili iskustveno. Na osnovu postojećih informacija moguće je kreiranje jednog oblika lokalnih normativa primenjivih samo na specifičan proizvodni sistem. Svi podaci se kreiraju na osnovu iskustvom i potrebama baziranim grupama proizvoda. Ova mogućnost je posebno primenljiva na polu automatizovane i neautomatizovane sisteme. Visoko automatizovani sistemi ovu operaciju izvode automatski ali uz niz ograničenja.
- model daje uvid u stanje i opterećenje tehnoloških sistema i učesnika proizvodnog procesa kroz razumljive grafičke prikaze.
- model daje mogućnost prevremenog upoznavanja učesnika proizvodnog procesa sa osobinama zaduženja.
- model daje mogućnost novog vida komunikacije unutar proizvodnog sistema i van njega upotrebom novih tehnologija. Upotreba mobilnih uređaja malih dimenzija u odnosu na dimenzije današnjih radnih stanica predstavlja još jednu od prednosti.
- model daje mogućnost lakog otklanjanja grešaka.

- status proizvodnog sistema može se pratiti kako uz pomoć radnih stanica tako i uz pomoć mobilnih uređaja sa bilo koje lokacije.

Praktični doprinos ove disertacije predstavlja low-level model sistema kreiran prema modelu koji se predlaže ovim istraživanjem. Nakon definicije modela izvršena je studija slučaja u kojoj je prikazano prototipsko rešenje informacionog sistema. Data je i infrastruktura test okruženja u sklopu koje je funkcionisao sistem tokom testiranja. Sistem je prikazan i opisan prema postojećim modulima. Nakon prikaza sistema dati su parametri upotrebe sistema u sklopu testiranja. Izvršena je simulacija i verifikacija rada sistema koja je demonstrirala rezultate rada sistema u laboratorijskim uslovima.

Sistem modifikuje postojeće modele komunikacije i praćenja i podiže ih na viši nivo. Moguće ga je integrisati u postojeće kanale komunikacija i koristiti ga paralelno sa istim.

Model sistema koncipiran je na način da omogućuje proširivanje i dublju integraciju u proizvodne sisteme. Time je stvorena osnova za rešavanje problema smanjenog opsega kontrole čitavog proizvodnog procesa. Ovim su stvoreni preduslovi većeg iskorišćenja novih komunikacionih tehnologija unutar proizvodnje.

Model je razvijen sa opštim konceptom koji omogućava približavanje drugim tipovima proizvodnih sistema. Model je testiran u sklopu laboratorijskih uslova grafičkog proizvodnog sistema ali nudi mogućnost, uz modifikacije, i integracije u druge tipove proizvodnih sistema. Ovim bi mogao biti iskorišćen za kreiranje softverskih rešenja sličnih namena u drugim granama industrije.

Razvijeni model daje mogućnosti integracije softverskih rešenja kreiranih po ovom modelu u sastav drugih postojećih rešenja. U ovom delu je potrebno rešiti problem platforme. Prikazano razvijeno rešenje je delom nezavisno od platforme na kojoj se izvršava a delom zavisno (klijentski modul je vezan za specifičnu mobilnu platformu). Razvijeni koncept ostavlja mogućnosti izbora platforme. Pri tome jedan od načina bi mogao da vodi ka Windows platformi iz razloga što je većina komercijalnih rešenja u čiji sastav bi se ovo rešenje moglo integrisati razvijena upravo u windows platformi. Drugi koncept koji omogućava razvijeni model bi mogao da vodi ka rešenju nezavisnom od platforme čiji kod se u potpunosti izvršava na korisničkom računaru (za razliku od web baziranog rešenja razvijenog po ovom modelu). U ovom slučaju model odlikuje lakoća integracije i povezivanja takvog rešenja u sklop postojećih razvijenih rešenja.

Model je koncipiran tako da se jednostavno mogu realizovati unapređenja kroz smanjenje uticaja ljudskog faktora na funkcionisanje sistema uz težnju ka automatizaciji.

Razvijeni model u osnovi daje mogućnost šireg razvoja uz integraciju sličnih modela.

8.1 Prilog 1 – Listing tabela baza podataka

- Tabela 'graficki_sistemi'

```

1 CREATE TABLE IF NOT EXISTS `graficki_sistemi` (
2   `id` int(11) NOT NULL AUTO_INCREMENT,
3   `naziv` varchar(255) COLLATE utf8_unicode_ci NOT NULL,
4   `tip` int(11) NOT NULL,
5   `formatS` int(11) NOT NULL,
6   `formatV` int(11) NOT NULL,
7   `norma70zalmasinistu` int(11) NOT NULL,
8   `vremePripreme` int(11) NOT NULL,
9   `brojOtisaka` int(11) NOT NULL,
10  `vremePranja` int(11) NOT NULL,
11  `normaPomocni` int(11) NOT NULL,
12  `normaUlagac` int(11) NOT NULL,
13  `normaDrugiMasinista` int(11) NOT NULL,
14  `prostorija` int(11) NOT NULL,
15  PRIMARY KEY (`id`)
16 ) ENGINE=InnoDB DEFAULT CHARSET=utf8
   COLLATE=utf8_unicode_ci AUTO_INCREMENT=13 ;

```

- Tabela 'graficki_sistemi_pozicije'

```

1 CREATE TABLE IF NOT EXISTS `graficki_sistemi_pozicije` (
2   `id` int(11) NOT NULL AUTO_INCREMENT,
3   `grafickiSistem` int(11) NOT NULL,
4   `levo` int(11) NOT NULL DEFAULT '5',
5   `gore` int(11) NOT NULL DEFAULT '5',
6   `sirina` int(11) NOT NULL DEFAULT '80',
7   `visina` int(11) NOT NULL DEFAULT '80',
8   PRIMARY KEY (`id`)
9 ) ENGINE=InnoDB DEFAULT CHARSET=utf8
   COLLATE=utf8_unicode_ci AUTO_INCREMENT=14 ;

```

- Tabela 'graficki_sistemi_prostorije'

```

1 CREATE TABLE IF NOT EXISTS `graficki_sistemi_prostorije` (
2   `id` int(11) NOT NULL AUTO_INCREMENT,
3   `naziv` varchar(255) COLLATE utf8_unicode_ci NOT NULL,
4   `sirina` int(11) NOT NULL DEFAULT '300',
5   `visina` int(11) NOT NULL DEFAULT '300',
6   PRIMARY KEY (`id`)
7 ) ENGINE=InnoDB DEFAULT CHARSET=utf8
   COLLATE=utf8_unicode_ci AUTO_INCREMENT=4 ;

```

- Tabela 'graficki_sistemi_tipovi'

```

1 CREATE TABLE IF NOT EXISTS `graficki_sistemi_tipovi` (
2   `id` int(11) NOT NULL,
3   `tip` varchar(255) COLLATE utf8_unicode_ci NOT NULL,
4   PRIMARY KEY (`id`)
5 ) ENGINE=InnoDB DEFAULT CHARSET=utf8
  COLLATE=utf8_unicode_ci;

```

- Tabela 'korisnici'

```

1 CREATE TABLE IF NOT EXISTS `korisnici` (
2   `id` int(11) NOT NULL AUTO_INCREMENT,
3   `korisnicko_ime` varchar(16) COLLATE utf8_unicode_ci NOT
  NULL DEFAULT 'gost',
4   `lozinka` varchar(255) COLLATE utf8_unicode_ci NOT NULL,
5   `ime` varchar(255) COLLATE utf8_unicode_ci NOT NULL,
6   `prezime` varchar(255) COLLATE utf8_unicode_ci NOT NULL,
7   `uloga` int(11) NOT NULL DEFAULT '4',
8   `datum_registracije` datetime NOT NULL,
9   `datum_poslednjeg_pristupa` datetime NOT NULL,
10  `status` tinyint(1) NOT NULL DEFAULT '1',
11  `blokiran` tinyint(1) NOT NULL DEFAULT '0',
12  `telefon1` varchar(24) COLLATE utf8_unicode_ci NOT NULL,
13  `telefon2` varchar(24) COLLATE utf8_unicode_ci NOT NULL,
14  `email` varchar(255) COLLATE utf8_unicode_ci NOT NULL,
15  `slika` text COLLATE utf8_unicode_ci NOT NULL,
16  `graficki_sistem` int(11) NOT NULL DEFAULT '0',
17  PRIMARY KEY (`id`)
18 ) ENGINE=InnoDB DEFAULT CHARSET=utf8
  COLLATE=utf8_unicode_ci AUTO_INCREMENT=7 ;

```

- Tabela 'korisnici_sesije'

```

1 CREATE TABLE IF NOT EXISTS `korisnici_sesije` (
2   `id` int(11) NOT NULL AUTO_INCREMENT,
3   `session_key` varchar(255) COLLATE utf8_unicode_ci NOT
  NULL,
4   `username` varchar(32) COLLATE utf8_unicode_ci NOT NULL,
5   `session_date` datetime NOT NULL,
6   PRIMARY KEY (`id`)
7 ) ENGINE=InnoDB DEFAULT CHARSET=utf8
  COLLATE=utf8_unicode_ci AUTO_INCREMENT=67 ;

```

- Tabela 'korisnici_sistemi'

```

1 CREATE TABLE IF NOT EXISTS `korisnici_sistemi` (
2   `id` int(11) NOT NULL AUTO_INCREMENT,
3   `korisnik` int(11) NOT NULL,
4   `graficki_sistem` int(11) NOT NULL,
5   PRIMARY KEY (`id`)
6 ) ENGINE=InnoDB DEFAULT CHARSET=utf8
  COLLATE=utf8_unicode_ci AUTO_INCREMENT=1 ;

```


- Tabela 'korisnici_uloge'

```

1 CREATE TABLE IF NOT EXISTS `korisnici_uloge` (
2   `id` int(11) NOT NULL AUTO_INCREMENT,
3   `uloga` varchar(255) COLLATE utf8_unicode_ci NOT NULL,
4   PRIMARY KEY (`id`)
5 ) ENGINE=InnoDB DEFAULT CHARSET=utf8
  COLLATE=utf8_unicode_ci AUTO_INCREMENT=3 ;

```

- Tabela 'poslovi'

```

1 CREATE TABLE IF NOT EXISTS `poslovi` (
2   `id` int(11) NOT NULL AUTO_INCREMENT,
3   `uid` varchar(24) COLLATE utf8_unicode_ci NOT NULL,
4   `nazivPosla` varchar(255) COLLATE utf8_unicode_ci NOT
  NULL,
5   `evidencioniBroj` varchar(255) COLLATE utf8_unicode_ci
  NOT NULL,
6   `grupa` int(11) NOT NULL,
7   `klijent` int(11) NOT NULL,
8   `datumUnosa` date NOT NULL,
9   `planiraniZavrsetakPosla` date NOT NULL,
10  `tiraz` int(11) NOT NULL,
11  `obim` int(11) NOT NULL,
12  `obrezanFormat` varchar(255) COLLATE utf8_unicode_ci NOT
  NULL DEFAULT '-',
13  `neobrezanFormat` varchar(255) COLLATE utf8_unicode_ci
  NOT NULL DEFAULT '-',
14  `aktivan` tinyint(1) NOT NULL DEFAULT '0',
15  `dostupanZaPreuzimanje` tinyint(1) NOT NULL DEFAULT '0',
16  `zapocet` tinyint(1) NOT NULL DEFAULT '0',
17  `završen` tinyint(1) NOT NULL DEFAULT '0',
18  `obavestjenjeKlijentu` tinyint(1) NOT NULL DEFAULT '0',
19  `arhiviran` tinyint(1) NOT NULL DEFAULT '0',
20  `napomenaP` text COLLATE utf8_unicode_ci NOT NULL,
21  `napomenaS` text COLLATE utf8_unicode_ci NOT NULL,
22  `napomenaZ` text COLLATE utf8_unicode_ci NOT NULL,
23  `isporukaZ` text COLLATE utf8_unicode_ci NOT NULL,
24  PRIMARY KEY (`id`)
25 ) ENGINE=InnoDB DEFAULT CHARSET=utf8
  COLLATE=utf8_unicode_ci AUTO_INCREMENT=15 ;

```

- Tabela 'poslovi_fajlovi'

```

1 CREATE TABLE IF NOT EXISTS `poslovi_fajlovi` (
2   `id` int(11) NOT NULL AUTO_INCREMENT,
3   `posao` int(11) NOT NULL,
4   `fajl` text COLLATE utf8_unicode_ci NOT NULL,
5   PRIMARY KEY (`id`)
6 ) ENGINE=InnoDB DEFAULT CHARSET=utf8
  COLLATE=utf8_unicode_ci AUTO_INCREMENT=18 ;

```

- Tabela 'poslovi_faze_procesa'

```

1 CREATE TABLE IF NOT EXISTS `poslovi_faze_procesa` (
2   `id` int(11) NOT NULL AUTO_INCREMENT,
3   `faza` varchar(255) COLLATE utf8_unicode_ci NOT NULL,
4   PRIMARY KEY (`id`)
5 ) ENGINE=InnoDB DEFAULT CHARSET=utf8
  COLLATE=utf8_unicode_ci AUTO_INCREMENT=9 ;

```

- Tabela 'poslovi_klijenti'

```

1 CREATE TABLE IF NOT EXISTS `poslovi_klijenti` (
2   `id` int(11) NOT NULL AUTO_INCREMENT,
3   `porucilac` varchar(255) COLLATE utf8_unicode_ci NOT
  NULL,
4   `telefon1` varchar(25) COLLATE utf8_unicode_ci NOT NULL,
5   `telefon2` varchar(25) COLLATE utf8_unicode_ci NOT NULL,
6   `adresa` varchar(255) COLLATE utf8_unicode_ci NOT NULL,
7   `faks` varchar(25) COLLATE utf8_unicode_ci NOT NULL,
8   `email` varchar(255) COLLATE utf8_unicode_ci NOT NULL,
9   PRIMARY KEY (`id`)
10 ) ENGINE=InnoDB DEFAULT CHARSET=utf8
  COLLATE=utf8_unicode_ci AUTO_INCREMENT=2 ;

```

- Tabela 'poslovi_segменти_lista'

```

1 CREATE TABLE IF NOT EXISTS `poslovi_segmenti_lista` (
2   `id` int(11) NOT NULL AUTO_INCREMENT,
3   `naziv` varchar(255) COLLATE utf8_unicode_ci NOT NULL,
4   `uid` int(11) NOT NULL,
5   `posao` int(11) NOT NULL,
6   PRIMARY KEY (`id`)
7 ) ENGINE=InnoDB DEFAULT CHARSET=utf8
  COLLATE=utf8_unicode_ci AUTO_INCREMENT=72 ;

```

- Tabela 'poslovi_segmenti_po_fazama'

```

1 CREATE TABLE IF NOT EXISTS `poslovi_segmenti_po_fazama` (
2   `id` int(11) NOT NULL AUTO_INCREMENT,
3   `posao` int(11) NOT NULL,
4   `segment` int(11) NOT NULL,
5   `graficki_sistem` int(11) NOT NULL,
6   `faza_procesa` int(11) NOT NULL,
7   `faza_global` int(11) DEFAULT NULL,
8   `vreme` int(11) NOT NULL,
9   `vreme_unosa` int(11) NOT NULL,
10  `u_toku` tinyint(1) NOT NULL DEFAULT '0',
11  `završen` tinyint(1) NOT NULL DEFAULT '0',
12  PRIMARY KEY (`id`)
13 ) ENGINE=InnoDB DEFAULT CHARSET=utf8
  COLLATE=utf8_unicode_ci AUTO_INCREMENT=60 ;

```

- Tabela 'poslovi_segmenti_sesije'

```

1 CREATE TABLE IF NOT EXISTS `poslovi_segmenti_sesije` (
2   `id` int(11) NOT NULL AUTO_INCREMENT,
3   `segment` int(11) NOT NULL,
4   `posao` int(11) NOT NULL,
5   `operater` int(11) NOT NULL,
6   `vreme_pocetka` datetime NOT NULL,
7   `vreme_zavrsetka` datetime NOT NULL,
8   `zavrshena` tinyint(1) NOT NULL DEFAULT '0',
9   PRIMARY KEY (`id`)
10 ) ENGINE=InnoDB DEFAULT CHARSET=utf8
    COLLATE=utf8_unicode_ci AUTO_INCREMENT=57 ;

```

- Tabela 'poslovi_zahtevi'

```

1 CREATE TABLE IF NOT EXISTS `poslovi_zahtevi` (
2   `id` int(11) NOT NULL AUTO_INCREMENT,
3   `poslao` int(11) NOT NULL,
4   `poslao_ime_prezime` varchar(255) COLLATE utf8_unicode_ci
    DEFAULT NULL,
5   `sistem` int(11) NOT NULL,
6   `sistem_naziv` text COLLATE utf8_unicode_ci NOT NULL,
7   `posao` int(11) NOT NULL,
8   `segment` int(11) NOT NULL,
9   `datum_slanja` datetime NOT NULL,
10  `odobreno` tinyint(1) NOT NULL DEFAULT '0',
11  `odbijeno` tinyint(1) NOT NULL DEFAULT '0',
12  `obradio` int(11) NOT NULL,
13  `obradio_ime_prezime` varchar(255) COLLATE
    utf8_unicode_ci NOT NULL,
14  `datum_obrađe` datetime NOT NULL,
15  PRIMARY KEY (`id`)
16 ) ENGINE=InnoDB DEFAULT CHARSET=utf8
    COLLATE=utf8_unicode_ci AUTO_INCREMENT=3 ;

```

- Tabela 'proizvodi_grupe'

```

1 CREATE TABLE IF NOT EXISTS `proizvodi_grupe` (
2   `id` int(11) NOT NULL AUTO_INCREMENT,
3   `grupa` varchar(255) COLLATE utf8_unicode_ci NOT NULL,
4   PRIMARY KEY (`id`)
5 ) ENGINE=InnoDB DEFAULT CHARSET=utf8
    COLLATE=utf8_unicode_ci AUTO_INCREMENT=27 ;

```

9 LITERATURA

1. Adams, M., Hofstede, A.H.M., van der Aalst, W.M.P., Edmond, D. (2007) Dynamic, Extensible and Context-Aware Exception Handling for Workflows, edicija Meersman, R. i Tari, Z. et al. (eds.): OTM 2007, Part I, LNCS 4803, pp 95–112.
2. Agarwal, R., Bruno, G., Torchiano, M. (2000) An operational approach to the design of workflow systems, *Information and Software Technology*, 42, 547–555.
3. Agostini, A. & De Michelis, G. (2000) Improving flexibility of workflow management systems. In: van der Aalst WMP, Desel J, Oberweis A, editors. *Business process management: models, techniques, and empirical studies LNCS*, Springer, 1806, 218–234.
4. Ahlemann, F. (2009) Towards a conceptual reference model for project management information systems, *International Journal of Project Management*, 27, 19–30.
5. Alonso, G., Agrawal, D., Abbadi, A.El., Kamath, M., Günthör, R., Mohan, C, (1996) *Advanced Transaction Models in Workflow Contexts*, Proceeding of 12th International Conference on Data Engineering, New Orleans, Louisiana: IEEE Computer Society Press, pp 574–581.
6. Alter, S. (2008) Defining information systems as work systems: implications for the IS field, *European Journal of Information Systems*, 17, 448–469
7. Aoumeur, N. & Saake, G. (2002) A component-based Petri net model for specifying and validating cooperative information systems, *Data & Knowledge Engineering*, 42, 143–187.
8. Arnott, D. & Pervan, G. (2007) *The Methodological and Theoretical Foundations of Decision Support Systems Research*, Information Systems Foundations, Theory, Representation and Reality, Published by ANU E Press, The Australian National University, Canberra ACT 0200, Australia, ISBN: 9781921313141.
9. Bae, J-S., Jeong, S-C., Seo, Y., Kim, Y., Kang, S-H. (1999) Integration of Workflow Management and Simulation, *Computers & Industrial Engineering*, 37, 203–206.
10. Balaban, N., Ristić, Ž., Đurković, J., Trninić, J., Tumbas, P. (2010) *Informacione tehnologije i informacioni sistemi*, Ekonomski fakultet Subotica, ISBN: 978-86-7233-244-5.

11. Bause, F. & Kritzinger, P. S. (2002) Stochastic Petri Nets An Introduction to the Theory.
12. Benbasat, I. & Zmud, R. W. (2003) The identity crisis within the discipline: Defining and communicating the discipline's core properties, *MIS Quarterly*, 27 (2), 183–194.
13. Benson, S. & Standing, C. (2002) *Information Systems: A Business Approach*, John Wiley and Sons.
14. Bider, I. (2005) Masking flexibility behind rigidity: Notes on how much flexibility people are willing to cope with. In: Castro, J., Teniente, E. (eds.) *CAiSE 2005 Workshops*, FEUP Edicoes, Porto, 1, pp 7–18.
15. Boddy, D., Boonnstra, A., Kennedy, G. (2005) *Managing Information Systems*, Pearson.
16. Booch, G., Rumbaugh, J., Jacobson, I. (1999) *The Unified Modeling Language User Guide*, Addison-Wesley.
17. Brooks (2013), RPM Remote Print Manager [Online], Dostupno sa <http://lpd.brooksnet.com/rpmelite-info.html> [Pristup 02.11.2013.].
18. Brooks, F.P. (1987) No Silver Bullet: Essence and Accidents of Software Engineering, *IEEE Computer*, 20 (4), 10–19.
19. Brooks, F.P. Jr. (1995) *The Mythical Man-Month*. Boston, MA: Addison Wesley Longman Inc.
20. CADLink (2013) [Online], Dostupno sa <http://www.cadlink.com/products/digitalFactory/> [Pristup 02.11.2013.].
21. Carlsen, S. (1997) *Conceptual Modeling and Composition of Flexible Workflow Models*, Informational Systems Group, Department of Computer and Information Science, Faculty of Physics, Informatics and Mathematics, Norwegian University of Science and Technology, Doctoral Thesis.
22. Casati, F., Ceri, S., Pernici, B., Pozzi, G (1996) Workflow evolution. In: *Proceedings of ER '96*, Cottubus, Germany, pp 438–455.
23. Cavaye, A.L.M. (1995) User participation in system development revisited, *Information and Management*, 28 (5), 311–323.
24. Chaffey, D. & Wood, S. (2005) *Business Information Management – Improving Performance using Information Systems*, Pearson Education Limited, ISBN: 978-0-273-68655-2.
25. Chema, S., Bachtarzi, F., Chaoui, A. (2012) A high-level Petri net based approach for modeling and composition of web services, *Procedia Computer Science*, 9, pp469–478.

26. Chen, Y. (2010) Production Process Object Modeling Based Petri Nets, *Advances in Petri Net Theory and Applications*, Aized, T. (ed.), ISBN: 978-953-307-108-4, InTech, [Online] Dostupno sa: <http://www.intechopen.com/books/advances-in-petri-net-theory-and-applications/production-process-objectmodeling-based-petri-nets> [Pristup 13.07.2013.].
27. Chiu, D.K.W., Li, Q., Karlapale, K. (2001) Web interface-driven cooperative exception handling Adome workflow management system, *Information Systems*, 26 (2), 93–120.
28. Choi, B. K., Lee, D., Kang, D. H. (2008) Devs modeling of run-time workflow simulation and its application, In *proceedings of the 22nd European Conference on Modeling and Simulation*.
29. Chundi, P., Xin, T., Ray, I. (2011) When does a workflow complete?, *Computer Standards & Interfaces*, 33, 292–300.
30. Cicirelli, F., Furfaro, A., Nigro, L. (2010) A service-based architecture for dynamically reconfigurable workflows, *The Journal of Systems and Software*, 83, 1148–1164
31. Dahlbom, B. (1996) The new informatics, *Scandinavian Journal of Information Systems*, 8 (2), 29–48.
32. Dicesare, F., Harhalakis, G., Proth, J.M., Silva, M., Vernadat, F.B. (1993) *Practice of Petri Nets in Manufacturing*, Chapman & Hall, London.
33. EFI (2013) [Online], Dostupno sa <http://w3.efi.com/en/WebToPrint/eCommerce-Web-to-Print-Software/Digital-Storefront/Resources/Materials-and-Demos/Brochures> [Pristup 02.11.2013.].
34. Ehrig, H. & Padberg, J. (2004) Graph Grammars and Petri Net Transformations. In *Lectures on Concurrency and Petri Nets*, Special Issue Advanced Course PNT, LNCS, Springer, 3098, pp 496–536.
35. Ehrig, H., Ehrig K., Prange, U., Taentzer, G. (2006) *Fundamentals of Algebraic Graph Transformation*. EATCS Monographs in Theoretical Computer Science. Springer, 2006.
36. Ehrig, H., Engels, G., Kreowski, H.J., Rozenberg, G. (eds) (1999) *Handbook of Graph Grammars and Computing by Graph Transformation, Volume 2: Applications, Languages and Tools*. World Scientific.
37. Ehrig, H., Kreowski, H.J., Montanari, U., Rozenberg, G. (eds) (1999) *Handbook of Graph Grammars and Computing by Graph Transformation. Volume 3: Concurrency, Parallelism and Distribution*. World Scientific.*

38. Eindhoven University of Technology (TU/e) [Online] Dostupno na <http://www.wis.win.tue.nl/~wvdaalst/workflowcourse/exercises/images/image084.jpg> [Pristup 06.07.2013.].
39. Ellis, C.A. & Keddara, K. (1993) Dynamic change within workflow systems, Technical Report, University of Colorado.
40. Ellis, C.A., Keddara, K. (2000) A workflow change is a workflow. In: van der Aalst WMP, Desel J, Oberweis A, editors. Business process management: models, techniques, and empirical studies LNCS, Springer, 1806, 201–217.
41. Ellis, C.A., Nutt, G.J. (1996) Workflow: the process spectrum, NSF Workshop on Workflow and Process Automation in Information Systems, Athens, Greece, pp. 140–145.
42. ESKO (2013) [Online], Dostupno sa <http://www.esko.com/en/Products/Overview/suite-12/overview> [Pristup 02.11.2013.].
43. Fakas, G. & Karakostas, B. (1999) A workflow management system based on intelligent collaborative objects, *Information and Software Technology*, 41, 907–915.
44. Fiery (2013) [Online], Dostupno sa <http://w3.efi.com/Fiery/Products/Workflow-Suite/Fiery-JobFlow> [Pristup 02.11.2013.].
45. Flores-Badillo, M. i López-Mellado, E. (2010) Multilevel Petri Nets for the Specification and Development of Workflow Automation Systems [Online], *Petri Nets Applications*, Pawel Pawlewski (ed.), ISBN: 978-953-307-047-6, InTech, Dostupno sa: <http://www.intechopen.com/books/petri-nets-applications/distributed-implementation-of-petrinets-in-control-applications> [Pristup 13.07.2013.].
46. Georgakopoulos, D., Hornick, M., Sheth, A. (1995) An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure, *Distributed and Parallel Databases*, 3, 119–153.
47. Grladinović, T. (1999) Upravljanje proizvodnim sustavima u preradi drva i proizvodnji nameštaja, Udžbenici sveučilišta u Zagrebu, Zagreb.
48. Guetat, S.B.A., Dakhli, S.B.D. (2012) Building Software Solutions in an Urbanized Information System: The 5+1 Software Architecture Model, *Procedia Technology*, 5, pp481–490.
49. Hafeez-Baig, A. & Gururajan, R. (2012) Critical Role of 'T-Shaped Skills & Incentive Rewards' as Determinants for Knowledge Management Enablers: A Case of Indian Study, *School of Information Systems*,

Faculty of Business and Law, University of Southern Queensland, Toowoomba, Queensland, Australia, Kalloniatis, C. (ed.) in MODERN INFORMATION SYSTEMS, Published by InTech, ISBN 978-953-51-0647-0, Croatia.

50. Hales, K. & Lavery, M. (1991) Workflow management software: the business opportunity, Ovum Ltd, London.
51. Hardcastle, E. (2008) Business Information Systems, Venus Publishing ApS, ISBN: 978-87-7681-463-2.
52. Hart, D.N. & Gregor, S.D. (2005) Information Systems Foundations: Constructing and Criticising, Published by ANU E Press, The Australian National University, Canberra ACT 0200, Australia, <http://epress.anu.edu.au>, ISBN 1 9209 4220 3.
53. Heidelberg BM (2013) [Online], Dostupno sa http://www.heidelberg.com/www/binaries/bin/files/dotcom/en/products/prinect/product_guide_business_manager_en.pdf [Pristup 02.11.2013.].
54. Herrmann, T., Hoffmann, M., Loser, K.U., Moysich, K. (2000) Semistructured models are surprisingly useful for user-centered design. In: De Michelis G, Giboin A, Karsenty L, Dieng R, editors. Designing Cooperative Systems Coop 2000. Amsterdam: IOS Press, pp 159–174.
55. Hevner, A., March, S., Park, J., Ram, S. (2004) Design science in information systems research, MIS Quarterly, 28 (1), 75–105.
56. Hollingsworth, D. (1995) The Workflow Management Coalition Specification – workflow Management Coalition The Workflow Reference Model. Doc No: TC00-1033., Issue: 1.1., Workflow Management Coalition, 2 Crown Walk, Winchester, Hampshire SO22 5XE, UK.
57. Holman, H. & Lorig, K. (2004) Patient self-management: a key to effectiveness and efficiency in care of chronic disease, Public Health Reports, May– June.
58. Huang, Y., Wang, H., Yu, P., Xia, Y. (2006) Property-Transition-Net-Based Workflow Process Modeling and Verification, Electronic Notes in Theoretical Computer Science, No159, 155–170.
59. Hwang, M. I. & Thorn, R. G. (1999) The effect of user engagement on system success: A meta-analytical integration of research findings. Information and Management, 35 (4), 229–236.
60. Jablonski, S. & Bussler, C. (1996) Workflow Management: Modelling, Concepts, Architecture and Implementation, International Thomson Computer Press.

61. Joeris, G. (1999) Defining flexible workflow execution behaviors. In: Dadam, P., Reichert, M. (eds.) *Enterprise-wide and Cross-enterprise Workflow Management: Concepts, Systems, Applications*. CEUR Workshop Proceedings, Paderborn, Germany, 24, pp 49–55.
62. KBA (2013) [Online], Dostupno sa <http://www.kba.com> [Pristup 28.11.2013.].
63. Kirikova, M., Finke, A., Grundspenkis, J. (2009) What Is CIM: An Information System Perspective, Riga Technical University, Latvia, Grundspenkis, J. i ostali (eds.) u *Advances in Databases and Information Systems, Associated Workshops and Doctoral Consortium of the 13th East European Conference, ADBIS 2009 Riga, Latvia*.
64. Klein, M., Dellarocas, C., Bernstein, A. (2000) Adaptive workflow systems. *J Comput Support Coop Work*, 9, 3–4. Special issue.
65. Klein, M., Dellarocas, C., Bernstein, A. (eds) (1998) In: *Proceedings of the CSCW-98 workshop towards adaptive workflow systems*, Seattle, WA.
66. Konica Minolta Printgroove (2013) [Online], Dostupno sa <http://www.biz.konicaminolta.com/production/printgroove/> [Pristup 02.11.2013.].
67. Koulopoulos, T.M. (1995) *The Workflow Imperatives: Building Real World Business Solutions*, Van Nostrand Reinhold, New York.
68. Kwapisz, S. (2009) UML 2.0 in the Modelling of the Complex Business Processes of Reporting and Control of Financial Information System, *Systems Analysis and Design for Advanced Modeling Methods: Best Practices*, Information science reference, ISBN 978-1-60566-345-6, pp115-133.
69. Lawrence, P. (ed) (1997) *Workflow Handbook, Workflow Management Coalition*, John Wiley and Sons, New York.
70. Lederman, R. & Johnston, R. (2007) *Are Routine Manual Systems Genuine Information Systems?*, Information Systems Foundations, Theory, Representation and Reality, Published by ANU E Press, The Australian National University, Canberra ACT 0200, Australia, ISBN: 9781921313141.
71. Lenz, R. & Reichert, M. (2007) IT Support for Healthcare Processes - Premises, Challenges, Perspectives. *DKE* 61, 82–111.
72. Leonardi, G., Panzarasa, S., Quaglini, S., Stefanelli, M., van der Aalst, W.M.P. (2007) Interacting agents through a web-based health serviceflow management system, *Journal of Biomedical Informatics*, 40, 486–499.
73. Leymann, F. & Roller, D. (1997) *Workflow-based applications*, IBM

- Systems Journal, 36 (1).
74. March, S.T., & Smith, G.F.: Design and natural science research on information technology, *Decision Support Systems*, 15, 251–66.
 75. Mattia, A. & Weistroffer, H.R. (2009) *Information Systems Development: Understanding User Participation as a Social Network, Systems Analysis and Design for Advanced Modeling Methods: Best Practices*, Information science reference, ISBN 978-1-60566-345-6, pp 58-67.
 76. Miao, H. & He, T. (2010) *Petri nets-based Models for Web Services Composition, Petri Nets Applications*, Pawel Pawlewski (Ed.), ISBN: 978-953-307-047-6, InTech, Dostupno sa: <http://www.intechopen.com/books/petri-nets-applications/distributed-implementation-of-petrinets-in-control-applications> [Pristup 13.07.2013.].
 77. Milićev, D. (2009) *Model-Driven Development with Executable UML*, Wiley Publishing, Inc., Published by Wiley Publishing, Inc., Indianapolis, Indiana, ISBN: 978-0-470-48163-9.
 78. Mireles, C., Noriega, A., Leyva, G. (2006) *Flexible Manufacturing System Simulation Using Petri Nets*, *Manufacturing the Future*, Kordic, V., Lazinica, A., Merdan, M. (eds.), ISBN: 386611-198-3, InTech [Online] Dostupno na: http://www.intechopen.com/books/manufacturing_the_future/flexible_manufacturing_system_simulation_using_petri_nets [Pristup 13.07.2013.].
 79. Murata, T. (1989) *Petri nets: Properties, analysis and applications*, in: *Proceedings of the IEEE*, 77, pp 541–580.
 80. Novaković, D. (2002) *Rukovanje materijalom u grafičkim procesima - monografija*, Edicija "Tehničke nauke - monografije", Fakultet tehničkih nauka, Novi Sad, Srbija.
 81. Océ (2013) [Online], Dostupno sa http://www7.oce.com/opp/PDF/PRISMAproduction_Server_brochure.pdf [Pristup 02.11.2013.].
 82. Orlikowski, W. J. & Iacono, C. S.: Research commentary: Desperately seeking the "IT" in IT research – A call to theorizing the IT artifact, *Information Systems Research*, 12 (2), 121–134.
 83. Pawlak, Z. (1981) *Information systems theoretical foundations*, *Information Systems*, 6 (3), 205–218.
 84. Pearlson, K.E. & Saunders, C.S. (2013) *Managing and Using Information Systems, A Strategic Approach*, 5th edition, Copyright 2013, 2010, 2006, 2004 John Wiley & Sons, ISBN 978-1-118-28173-4.
 85. Perez, M. & Rojas, T. (2000) *Evaluation of Workflow-type software*

- products: a case study, *Information and Software Technology*, 42, 489–503.
86. Piedrafita, R., Tardioli, D., Villarroel, J., L. (2010) Distributed Implementation of Petri nets in Control Applications, *Petri Nets Applications*, Pawlewski, P. (ed.), ISBN: 978-953-307-047-6, InTech, [Online] Dostupno sa: <http://www.intechopen.com/books/petri-nets-applications/distributed-implementation-of-petrinets-in-control-applications> [Pristup 13.07.2013.].
 87. Poliščuk, E.J. (2007) Projektovanje informacionih sistema, Elektrotehnički fakultet Podgorica.
 88. Presswise (2013) [Online], Dostupno sa <http://www.presswise.com/about/print-mis/> [Pristup 02.11.2013.].
 89. Print Manager Plus (2013) [Online], Dostupno sa http://www.printmanager.com/products/print_manager_plus_enterprise_edition.htm [Pristup 02.11.2013.].
 90. Print MIS (2013) [Online], Dostupno sa <http://www.printmis.com/> [Pristup 02.11.2013.].
 91. Rainer Jr, R. K. & Cegielski, C. G. (2011) *Introduction to Information Systems Third Edition*, Copyright © 2007, 2009, 2011 John Wiley & Sons, Inc.
 92. Ramiller, N.C. & Pentland, B.T. (2009) Management Implications in Information Systems Research, *Journal of the Association for Information Systems*, 10 (6), 474–494.
 93. Reichert, M. & Dadam, P. (1998) ADEPT flex Supporting Dynamic Changes of Workflows Without Losing Control, *Journal of Intelligent Information Systems - Special Issue on Workflow Management*, Kluwer Academic Publishers, 10, 93–129.
 94. Reichert, M., Bauer, T., van der Aalst, W.M.P., Edmond, D. (2007) Supporting Ad-Hoc Changes in Distributed Workflow Management Systems, edicija Meersman, R. i Tari, Z. et al. (eds.): *OTM 2007, Part I, LNCS 4803*, pp. 95–112.
 95. Reijers, H.A., Rigter, J., van der Aalst, W.M.P. (2003) The Case Handling Case. *International Journal of Cooperative Information Systems*, 12 (3), 365-391.
 96. Rinderle, S., Reichert, M., Dadam, P.: Flexible support of team processes by adaptive workflow systems. *Distributed and Parallel Databases* 16 (1), 91–116.
 97. Rodriguez, R., Rodriguez, O., Reyes, G., Cruz, V. (2010) Using Petri nets

for modeling and verification of Hybrid Systems [Online], Petri Nets Applications, Pawel Pawlewski (ed.), ISBN: 978-953-307-047-6, InTech, Dostupno sa: <http://www.intechopen.com/books/petri-nets-applications/distributed-implementation-of-petrinets-in-control-applications> [Pristup 13.07.2013.].

98. Rozenberg, G. (1997) Handbook of Graph Grammars and Computing by Graph Transformations, Volume 1: Foundations. World Scientific.
99. Salimifard, K., Wright, M. (2001) Petri net-based modelling of workflow systems: An overview, *European Journal of Operational Research*, 134, 664-676
100. Shi, J.J., Lee, D.E., Kuruku, E. (2008) Task-based modeling method for construction business process modeling and automation, *Automation in Construction*, 17, 633-640.
101. Smith, T. (1993) The Future of Work flow Software, INFORM.
102. Sørensen, C.G., Fountas, S., Nash, E., Pesonen, L., Bochtis, D., Pedersen, S.M., Basso, B., Blackmore, S.B. (2010) Conceptual model of a future farm management information system, *Computers and Electronics in Agriculture*, 72, 37-47.
103. Stair, R. & Reynolds, G. (2003) Principles of Information Systems, 6th ed., Boston, MA, Thomson Course Technology.
104. Stair, R.M. & Reynolds, G.W. (2011) Fundamentals of Information Systems, Sixth Edition, Copyright 2011 Cengage Learning. ISBN-13: 978-0-8400-6218-5.
105. Sun, P. & Jiang, C. (2009) Analysis of workflow dynamic changes based on Petri net, *Information and Software Technology*, 51, 284-292.
106. van Belle, J.P., Eccles, M., Nash, J. (2001) Discovering Information Systems, ISBN: 0-620-24194-2
107. van der Aalst, W.M.P. & Jablonski, S. (2000) Dealing with workflow change: Identification of issues and solutions, *Int J Comput Syst Sci Eng*, 15(5), 67-76.
108. van der Aalst, W. M. P. (1998) The Application of Petri Nets to Workflow Management, *Journal of Circuits, Systems, and Computers*, 8 (1), 21-66.
109. van der Aalst, W.M.P (1999) Process-oriented architectures for electronic commerce and interorganizational workflow, *Information Systems*, 24 (8), 639-671.
110. van der Aalst, W.M.P. & Berens, P.J.S. (2001) Beyond Workflow Management: Product- Driven Case Handling. In S. Ellis, T. Rodden, and I. Zigurs, editors, International ACM SIGGROUP Conference on

- Supporting Group Work (GROUP 2001), ACM Press, New York, pp 42–51.
111. van der Aalst, W.M.P. & ter Hofstede, A.H.M. (2005) YAWL: Yet Another Workflow Language. *Information Systems*, 30 (4), 245–275.
 112. van der Aalst, W.M.P. & van Hee, K. (2000) *Workflow Management: Models, Methods and Systems*, Eindhoven University of Technology
 113. van der Aalst, W.M.P. (1996) Three Good reasons for Using a Petri-net-based Workflow Management System. In S. Navathe and T. Wakayama, editors, *Proceedings of the International Working Conference on Information and Process Integration in Enterprises (IPIC'96)*, pp 179–201, Cambridge, Massachusetts.
 114. van der Aalst, W.M.P. (1997) Verification of workflow nets. In P. Azema and G. Balbo, editors, *Application and Theory of Petri Nets*, volume 1248 of *Lecture Notes in Computer Science*, pp. 407-426, Springer-Verlag, Berlin.
 115. van der Aalst, W.M.P., Aldred, L., Dumas, M., ter Hofstede, A.H.M. (2004) Design and implementation of the YAWL system. In: *Proc. of the 16th Int. Conf. on advanced information systems engineering (CaiSE 04)*, Riga, Latvia.
 116. van der Aalst, W.M.P., Stoffele, M., Wamelink, J.W.F. (2003) Case Handling in Construction. *Automation in Construction*, 12 (3), 303-320.
 117. van der Aalst, W.M.P., ter Hofstede, A.H.M., Kiepuszewski, B., Barros, A.P. (2003) Workflow Patterns, *Distributed and Parallel Databases*, 14 (1), 5–51.
 118. van der Aalst, W.M.P., Weske, M., Grunbauer, D. (2005) Case Handling: A New Paradigm for Business Process Support. *Data and Knowledge Engineering*, 53(2), 129–162.
 119. van Hee, K.M. (1994) *Information Systems Engineering*, Cambridge University Press, Cambridge.
 120. van Sinderen, M. J., Joosten, S.M.M., de Farias, C.R.G. (1999) Workflow automation based on OSI job transfer and manipulation, *Computer Standards & Interfaces*, 21, 403–415.
 121. Wamelink, J.W.F., Stoffele, M., van der Aalst, W.M.P. (2002) Workflow management in construction; opportunity for the future, *Conference Proceeding – distributed knowledge in building*, International Council for Research and Innovation in Building and Construction, 12 (14).
 122. Weber, B., Rinderle, S., Reichert, M. (2007) Change patterns and change support features in process-aware information systems. In: *CAiSE 2007*.

- Proc. 19th Int'l Conf. On Advanced Information Systems Engineering, pp 574–588.
123. Weske, M. (1998) Flexible Modeling and Execution of Workflow Activities. In: Proc. 31st Hawaii Int. Conf. on Sys Sciences, Hawaii, pp 713–722.
 124. WFMC (1996) Workflow Management Coalition Terminology and Glossary (WFMC-TC-1011). Technical report, Workflow Management Coalition, Brussels.
 125. WFMC (1999) Workflow Management Coalition. Workflow management coalition terminology & glossary.
 126. WFMC (2013) Workflow Standard – Interoperability Abstract Specification [Online] Dostupno na <http://www.wfmc.org>, [Pristup 13.07.2013.].
 127. Workflow Patterns (2013), [Online] Dostupno sa: <http://www.workflowpatterns.com/> [Pristup: 13.07.2013.].
 128. Xerox FreeFlow (2013) [Online], Dostupno sa <http://www.xerox.com/digital-printing/workflow/freeflow/core/enus.html> [Pristup 02.11.2013.].
 129. Yang, N., Yu, H., Qian, Z., Sun, H. (2012) Modeling and quantitatively predicting software security based on stochastic Petri nets, *Mathematical and Computer Modelling*, 55, 102–112.
 130. Yu, Z., Fu, X., Liu, Y., Wang, J., Cai, Y (2010) Modeling and Analyzing Software Architecture Using Object-Oriented Petri Nets and π -calculus [Online], *Petri Nets Applications*, Pawel Pawlewski (ed.), ISBN: 978-953-307-047-6, InTech, Dostupno sa: <http://www.intechopen.com/books/petri-nets-applications/distributed-implementation-of-petrinets-in-control-applications> [Pristup 13.07.2013.].
 131. Zelenović, D. M. (2003) Projektovanje proizvodnih sistema, tokovi materijala, Edicija "Tehničke nauke - udžbenici", Fakultet tehničkih nauka, Novi Sad, Srbija.
 132. Zhu, H. & Li, X. (2013) Development of a web-based information system for shield tunnel construction projects, *Tunnelling and Underground Space Technology*, 37, 146–156.
 133. Zhuge, H., Cheung, T., Pung, H. (2001) A timed workflow process model, *The Journal of Systems and Software*, 55, 231-243.

10 BIOGRAFIJA



Darko Avramović rođen je u Rumi 1983. godine. Nakon završene osnovne škole, 1998. godine upisao je srednju tehničku školu, grafički odsek. Nakon završene srednje škole, 2002. godine upisao je studije na Fakultetu tehničkih nauka, odsek Grafičko inženjerstvo i dizajn. U decembru 2007. godine odbranio je master rad i time stekao zvanje diplomirani inženjer grafičkog inženjerstva i dizajna - master. U januaru 2008. godine zaposlio se na Fakultetu tehničkih nauka kao saradnik u nastavi na Departmanu za Grafičko inženjerstvo i dizajn. U momentu pisanja disertacije, kandidat je zaposlen u zvanju asistenta. Od zaposlenja na Departmanu držao je nastavu na pet nastavnih predmeta. Doktorske studije upisao je u martu 2008 i do momenta pisanja disertacije položio je sve ispite propisane nastavnim planom i programom doktorskih studija na Departmanu za Grafičko inženjerstvo i dizajn. Nakon ispunjenja svih propisanih obaveza prijavio je temu doktorske disertacije pod nazivom "Razvoj modela informacionog sistema za podršku upravljanju grafičkim procesima". Objavio je više naučnih radova od kojih dva na SCI listi. Učesnik je na projektima ministarstva za nauku i tehnološki razvoj republike Srbije. Od stranih jezika vrlo dobro govori i piše engleski jezik.