

UNIVERZITET U BEOGRADU  
FAKULTET ORGANIZACIONIH NAUKA

Lena S. Đorđević

**DETEKCIJA I ANALIZA GREŠAKA  
U IMPLEMENTACIJI DINAMIČKIH  
DISKRETNIH MODELA UPRAVLJANJA  
ZALIHAMA**

doktorska disertacija

Beograd, 2016

UNIVERSITY OF BELGRADE  
FACULTY OF ORGANIZATIONAL SCIENCES

Lena S. Đorđević

**ERROR DETECTION AND ANALYSIS  
IN IMPLEMENTATION OF DYNAMIC  
DISCRETE INVENTORY CONTROL  
MODELS**

Doctoral Dissertation

Belgrade, 2016

**MENTOR:**

Dr Danica Lečić-Cveković, vanredni profesor  
Fakultet organizacionih nauka, Univerzitet u Beogradu

**ČLANOVI KOMISIJE:**

Dr Jasmina Omerbegović-Bijelović, redovni profesor  
Fakultet organizacionih nauka, Univerzitet u Beogradu

Dr Slađan Babarogić, docent  
Fakultet organizacionih nauka, Univerzitet u Beogradu

Dr Dragana Makajić-Nikolić, docent  
Fakultet organizacionih nauka, Univerzitet u Beogradu

Dr Nebojša Bojović, redovni profesor  
Saobraćajni fakultet, Univerzitet u Beogradu

Datum odbrane: \_\_\_\_\_2016. godine.

# DETEKCIJA I ANALIZA GREŠAKA U IMPLEMENTACIJI DINAMIČKIH DISKRETNIH MODELA UPRAVLJANJA ZALIHAMA

## **Rezime:**

Predmet istraživanja ove doktorske disertacije je pristup za detekciju i analizu grešaka u dinamičkim diskretnim spredšit modelima upravljanja zalihama, zasnovan na karakteristikama problema i načinu modeliranja. Postojeći pristupi za obezbeđenje kvaliteta spredšit modela pokazali su se kao perspektivni, ali su nedovoljno ispitani i prilagođeni konkretnim problemima. Spredšit modeli korišćeni za evaluaciju ovih pristupa su najčešće namenski kreirani, nisu realni primeri, relativno su mali i nisu prilagođeni kompleksnim problemima sa velikim brojem zavisnosti, tako da njihova primena u realnim okolnostima, kao i skalabilnost nisu potvrđeni. Ovi pristupi, u najvećem broju slučajeva, zasnovani su na osobinama spredšit aplikacija, ali i idejama iz različitih oblasti kao što su: softversko inženjerstvo, operaciona istraživanja i druge. Postojeći pristupi ne uzimaju u obzir karakteristike problema i način modeliranja, koji značajno utiču na nastanak, ali i mogućnost otkrivanja grešaka. Oni podrazumevaju da su izlazne vrednosti modela unapred poznate ili da korisnik može da obezbedi sve informacije o strukturi i ograničenima modela, što veoma često nije moguće. U skladu sa navedenim, opravdan je i neophodan razvoj novog pristupa, koji bi omogućio unapređenje kvaliteta spredšit modela za upravljanja zalihama, odnosno detekciju i analizu grešaka u njima.

Osnovni cilj ove disertacije je kreiranje novog pristupa za obezbeđenje višeg kvaliteta dinamičkih diskretnih modela upravljačkih problema operacionog menadžmenta, konkretno upravljanja zalihama, razvijenih u spredšitovima, razvojem algoritma za detekciju i analizu grešaka u navedenim modelima i utvrđivanje njihovih uzroka na brz i efikasan način.

Novorazvijeni pristup za detekciju i analizu grešaka u dinamičkim diskretnim spredšit modelima upravljanja zalihama, predstavljen u ovoj disertaciji, pogodan je za upotrebu

od strane krajnjih korisnika i omogućava podizanje kvaliteta spređit modela za upravljanje zalihama. Pristup se zasniva na karakteristikama problema i načinu modeliranja, kao i metodama operacionih istraživanja, tehnikama razvijenim u oblasti informacionih sistema i spređit inženjerstva. Razvoj i evaluacija pristupa je realizovana nad spređit modelom ekonomične količine naručivanja zaliha za više proizvoda sa ograničenim prostorom skladištenja.

U cilju analize efikasnosti i skalabilnosti, razvijeni pristup je testiran na određenom broju eksperimenata. Eksperimenti su sprovedeni nad modelima koji razmatraju 3, 20 i 100 proizvoda, za različiti broj vremenskih perioda posmatranja. U skladu sa rezultatima sprovedenih eksperimenata, pristup za detekciju i analizu spređit grešaka predložen u ovoj disertaciji, zasnovan na karakteristikama problema i načinu modeliranja, pokazao se kao veoma efikasan i skalabilan, dok je vreme realizacije pristupa prihvatljivo.

**Ključne reči:** detekcija grešaka, analiza grešaka, diskretni sistemi, dinamički modeli, upravljanje zalihama, spređitovi.

**Naučna oblast:** Operacioni menadžment

**Uža naučna oblast:** Upravljanje proizvodnjom i uslugama

**UDK broj:** 004.4 : 519.8

# **ERROR DETECTION AND ANALYSIS IN IMPLEMENTATION OF DYNAMIC DISCRETE INVENTORY CONTROL MODELS**

## **Abstract:**

The subject of this doctoral thesis refers to error detection and debugging approach for dynamic discrete spreadsheet models of inventory control problems, based on problem characteristics and modelling method. Existing quality assurance approaches are very perspective, but insufficiently tested and adapted to actual problems. Spreadsheet models used for evaluation of those approaches are usually created just in that purpose and aren't real practical examples. Experimental spreadsheet models are generally small and aren't adjusted to complex problems with many dependencies. Applicability and scalability of existing quality assurance approaches has not been proven in wider context. In most cases, approaches are based on spreadsheet application characteristics and ideas from different scientific areas, such as: software engineering, operations reaserch and others. Existing approaches do not consider problem characteristics and modelling method, which significantly influence error occurrence and error detection. Those approaches consider that model output values are known in advance or that all informations about model structure and constraints are provided by user. Very often, both presumptions are unacceptable for models used in practice. In accordance with aforementioned, development of new improved quality assurance approach for inventory control spreadsheet models is justified and necessary.

Main goal of this dissertation is to create new spreadsheet quality assurance approach for dynamic discrete models of operations management problems, specifically inventory control problems, developed in spreadsheets, by developing algorithm for error detection and quick and efficient debugging for mentioned models.

Newly developed error detection and debugging approach for dynamic discrete spreadsheet models of inventory control problems, presented in this dissertation, is adapted to users and allows quality improvement of inventory control spreadsheet

models. The approach is based on problem characteristics and modelling method and includes operation research methods and techniques developed in software and spreadsheet engineering fields. Development and evaluation of approach is realized on multiproduct economic order quantity based inventory control model with storage space constraint, developed in spreadsheet.

In order to analyze efficiency and scalability of developed approach, multiple experiments are conducted. Experiments were conducted on models with 3, 20 and 100 products for different number of time periods. In accordance with results of experiments, presented in this doctoral dissertation, spreadsheet error detection and debugging approach, based on problem characteristics and modelling approach, proved to be very efficient and scalable. CPU time of approach is acceptable.

**Keywords:** error detection, error analysis, discrete systems, dynamic models, inventory control, spreadsheets.

**Academic Expertise:** Operations management

**Major in:** Production and service management

**UDC:** 004.4 : 519.8

# Sadržaj

<b>1. Uvod.....</b>	<b>1</b>
1.1. Problem, predmet, cilj i hipoteze doktorske disertacije .....	4
1.2. Struktura doktorske disertacije .....	9
<b>2. Upravljački problemi operacionog menadžmenta.....</b>	<b>12</b>
2.1. Problemi upravljanja zalihama .....	18
2.2. Osnovni troškovi u procesu upravljanja zalihama .....	24
2.3. Tipovi upravljačkih modela za upravljanje zalihama.....	29
<b>3. Dinamički diskretni upravljački modeli .....</b>	<b>37</b>
3.1. Pojam modela .....	37
3.1.1 Tipovi modela .....	38
3.1.2 Simulacioni model i simulacija .....	40
3.2. Upravljanje diskretnim sistemima.....	44
3.2.1 Pojam objekta diskretnog upravljanja .....	44
3.2.2 Optimalno upravljanje diskretnim objektom.....	46
3.2.3 Osnovni elementi modela objekta diskretnog upravljanja .....	48
<b>4. Modelovanje upravljačkih procesa u spredšitovima .....</b>	<b>52</b>
4.1. Istorijat razvoja spredšitova.....	52
4.2. Analitički modeli upravljačkih problema u spredšitovima .....	57
4.3. Simulacija i simulacioni model procesa upravljanja u spredšitovima .....	64
<b>5. Model upravljanja zalihama kao dinamički diskretni proces     razvijen u spredšitu.....</b>	<b>69</b>
5.1. Klasični deterministički EOQ model .....	69
5.2. EOQ model kao dinamički diskretni upravljački model u spredšitu.....	73
5.3. EOQ model za više proizvoda sa ograničenim skladišnim prostorom.....	81
5.3.1 Dinamički diskretni EOQ model za više proizvoda sa ograničenim skladišnim prostorom .....	81
5.3.2 Dinamički diskretni EOQ model za više proizvoda sa ograničenim skladišnim prostorom razvijen u spredšitu.....	85



<b>6. Pristupi za obezbeđenje kvaliteta modela u spredšitovima .....</b>	<b>88</b>
6.1. Greške u spredšit modelima .....	88
6.2. Taksonomija spredšit grešaka .....	91
6.3. Pristupi za detekciju i analizu grešaka u spredšitovima .....	96
6.3.1 Vizualizacija spredšitova.....	101
6.3.2 Statičke analize i izveštaji .....	105
6.3.3 Testiranje spredšitova.....	107
6.3.4 Automatska detekcija i ispravka grešaka u spredšitovima.....	110
6.3.5 Pristupi razvoja spredšitova vođeni modelom .....	113
6.3.6 Projektovanje i održavanje spredšitova.....	116
6.4. Uporedni prikaz pristupa za detekciju i analizu grešaka u spredšitovima .....	120
<b>7. Pristup za detekciju i analizu grešaka u dinamičkim diskretnim modelima upravljanja zalihama razvijenim u spredšitovima .....</b>	<b>124</b>
7.1. Osnovni koncept pristupa za detekciju i analizu grešaka u DDMUZRS .....	124
7.2. Algoritmi pristupa za detekciju i analizu grešaka u DDMUZRS.....	126
7.3. Generisanje test slučajeva .....	137
7.4. Implementacija pristupa za detekciju i analizu grešaka u DDMUZRS i numerički rezultati izvedenih eksperimenata .....	140
7.4.1 Detekcija i analiza grešaka u DDMUZRS za tri proizvoda .....	142
7.4.2 Detekcija i analiza grešaka u DDMUZRS za 20 i 100 proizvoda.....	146
7.5. Analiza rezultata.....	150
7.6. Uporedna analiza postojećih i novog pristupa za detekciju i analizu grešaka u spredšitovima.....	156
<b>8. Zaključak .....</b>	<b>159</b>
8.1. Pregled istraživanja, naučni i stručni doprinosi, hipoteze .....	159
8.2. Pravci budućih istraživanja .....	173
<b>Literatura.....</b>	<b>174</b>
<b>Biografija.....</b>	<b>192</b>

## Spisak tabela

Tabela 2-1. Ukupan broj tema u oblasti OM-a u istraživanjima.....	14
Tabela 4-1. Karakteristike kvaliteta elemenata ASM.....	61
Tabela 4-2. Karakteristike kvaliteta ASM na opštem nivou.....	63
Tabela 5-1. Karakteristike klasičnog EOQ modela.....	70
Tabela 5-2. Formule za predstavljanje dinamičkog EOQ modela u spredšitu.....	77
Tabela 5-3. Spredšit formule dinamičkog diskretnog EOQ modela za više proizvoda sa ograničenim skladišnim prostorom.....	86
Tabela 6-1. Osnovne kategorije pristupa automatizovanog obezbeđenja kvaliteta spredšitova.....	98
Tabela 6-2. Pristupi automatizovanog obezbeđenja kvaliteta spredšitova.....	100
Tabela 7-1. Opseg vrednosti ulaznih parametara.....	141
Tabela 7-2. Rezultati evaluacije pristupa nad DDMUZRS za tri proizvoda, T=7 dana, sa jednom ugrađenom greškom.....	143
Tabela 7-3. Rezultati evaluacije pristupa nad DDMUZRS za tri proizvoda, T=7 dana, sa dve ugrađene greške.....	144
Tabela 7-4. Uporedni prikaz rezultata evaluacije pristupa nad DDMUZRS za tri proizvoda.....	145
Tabela 7-5. Uporedni prikaz rezultata evaluacije pristupa nad DDMUZRS za 20 proizvoda.....	147
Tabela 7-6. Uporedni prikaz rezultata evaluacije pristupa nad DDMUZRS za 100 proizvoda.....	149
Tabela 7-7. Uporedni prikaz rezultata evaluacije pristupa nad DDMUZRS.....	151
Tabela 7-8. Uporedni prikaz rezultata evaluacije pristupa nad DDMUZRS za T=7.....	152
Tabela 7-9. Uporedni prikaz rezultata evaluacije pristupa nad DDMUZRS za T=28.....	153
Tabela 7-10. Uporedni prikaz rezultata evaluacije pristupa nad DDMUZRS za T=365.....	154
Tabela 7-11. Uporedni prikaz rezultata evaluacije pristupa nad DDMUZRS sa jednom ugrađenom greškom.....	154
Tabela 7-12. Uporedni prikaz rezultata evaluacije pristupa nad DDMUZRS sa dve ugrađene greške.....	155
Tabela 7-13. Uporedni prikaz rezultata evaluacije pristupa nad DDMUZRS sa više od pet ugrađenih greška.....	156

## Spisak slika

Slika 2-1. Zastupljenost pojedinačnih tema u radovima iz oblasti OM-a.....	15
Slika 2-2. Zastupljenost kombinacija tema u radovima u oblasti OM-a.....	16
Slika 2-3. Metodologije za izradu naučnih radova u oblasti OM-a.....	17
Slika 2-4. Grafički prikaz relevantnih TR zaliha u zavisnosti od veličine narudžbine.....	29
Slika 3-1. Prikaz stanja sistema tačkom u prostoru stanja.....	37
Slika 3-2. Dopustivi prostori stanja, okolnosti i upravljanja.....	46
Slika 3-3. Prevođenje diskretnog objekta upravljanja iz početnog u krajnje stanje.	47
Slika 4-1. Plimpton 322 predstavlja Pitagorine trojke.....	52
Slika 4-2. Prikaz ekrana <i>VisiCalc</i> .....	53
Slika 4-3. Izgled prozora u <i>Excel 1.0</i> .....	55
Slika 4-4. Upotreba alata krajnjih korisnika.....	56
Slika 4-5. Domen analitičkih spredšit modela.....	58
Slika 5-1. Kriva ukupnih troškova u EOQ modelu.....	73
Slika 5-2. EOQ model u kompaniji BEDROCK.....	77
Slika 5-3. Diskretni EOQ model u spredšitu.....	78
Slika 5-4. Dinamika akumulacije za izračunavanje promenljivih veličina modela.	79
Slika 5-5. Dinamički diskretni višestapni EOQ spredšit model za više proizvoda sa ograničenim skladišnim prostorom.....	87
Slika 6-1. Metrike za merenje spredšit grešaka.....	92
Slika 6-2. Revidirana taksonomija spredšit grešaka.....	93
Slika 6-3. Taksonomija spredšit grešaka.....	94
Slika 6-4. Žbunasta taksonomija spredšit grešaka.....	95
Slika 6-5. Binarna taksonomija spredšit grešaka.....	96
Slika 7-1. Pristup za detekciju i analizu grešaka u DDMUZRS.....	126
Slika 7-2. DDMUZRS za tri proizvoda i horizont posmatranja od sedam dana.....	142
Slika 7-3. Prosečno vreme potrebno za detekciju grešaka u DDMUZRS za tri proizvoda.....	146
Slika 7-4. Prosečno vreme potrebno za detekciju grešaka u DDMUZRS za 20 proizvoda.....	148
Slika 7-5. Prosečno vreme potrebno za detekciju grešaka u DDMUZRS za 100 proizvoda.....	150

## 1. Uvod

U današnjem poslovnom okruženju, koje karakteriše neizvesnost, nesigurnost i izražena konkurentnost tržišta, preduzeća se nalaze u nezavidnoj situaciji, koja iziskuje stalno „gledanje“ unapred, predviđanje i prilagođavanje. Operacioni menadžeri moraju stalno da razmatraju okolnosti pod kojima će se odvijati procesi poslovanja. Oni nemaju značajan uticaj na dinamiku i intenzitet tih okolnosti, ali imaju mogućnost da svojim upravljačkim akcijama iskoriste povoljna delovanja okolnosti, odnosno preduprede njihova nepovoljna delovanja. Kompanije postavljaju svoje strateške i operativne ciljeve, a zatim utvrđuju rezultate kojima će ispuniti te ciljeve, odnosno utvrđuju meru ostvarenosti ciljeva, ili kriterijum upravljanja. Nakon ovog koraka, potrebno je odrediti akcije koje u najvećoj meri omogućavaju postizanje rezultata za ostvarenje postavljenih ciljeva (Kostić, 2008b).

Događaji u budućnosti su neposredno ili posredno povezani, tako da se uticaj na jedan od događaja pre ili kasnije prenosi do svih događaja u budućnosti. Veze među događajima u budućnosti su veoma složene, pa se zato budućnost i smatra neizvesnom, u onoj meri u kojoj nije moguće da se sagledaju sve te veze. Zato je, pored intuitivnih sposobnosti i iskustva, nemoguće doneti prave poslovne odluke bez kvantitativnih analiza. Neophodno je primeniti određenu racionalnu tehniku sagledavanja budućih događaja, njihove međupovezanosti, kao i načina i brzine prenošenja uticaja kroz njihove međuveze.

Poslovanje i procesi u preduzeću mogu se posmatrati kao sistem matematičkih relacija. Takav sistem se zove matematički model, (Kostić, 2008b). U tom sistemu, jedan broj obuhvaćenih veličina se uzima kao nezavisne (unapred poznate), a drugi broj veličina se smatra zavisnim (nepoznatim) veličinama. Što je problem kompleksniji, kako sa aspekta broja veličina koje ulaze u proračun tako i sa aspekta vremenskog horizonta u kome će se odvijati poslovni poduhvat, to je i proračun zavisnih promenljivih kompleksniji i vremenski zahtevniji.

Jedan od veoma pogodnih načina za modelovanje i rešavanje realnih problema obezbeđen je konceptom diskretnog objekta upravljanja. Ovaj koncept omogućava razdvajanje diskretnog objekta (zakona ponašanja i oblasti upravljanja), ciljnog funkcionala i metode za nalaženje što boljeg rešenja. Struktura spredšita izuzetno je pogodna za predstavljanje različitih vrednosti promenljivih u vremenskim periodima, kao što je prikazano u Kostić (2008, 2009, 2012). Implementacija koncepta diskretnog objekta upravljanja čini spredšitove još moćnijim alatom za simulaciju, pogodnim za razumevanje i modelovanje poslovne dinamike. Takođe, pristup je u skladu sa preporukama za dizajn i projektovanje spredšit aplikacija i modela. Ipak, spredšit modeli kreirani na ovaj način su izraženo podložni nastanku grešaka, usled procesa razvoja, koji uključuje prevođenje verbalnih i matematičkih izraza u spredšit formule i funkcije.

Zahvaljujući karakteristikama današnjih računara i mogućnostima rada sa različitim softverskim aplikacijama, rešavanje takvih problema je značajno olakšano. Ipak, treba imati u vidu da stručnjaci iz različitih poslovnih oblasti nisu profesionalni programeri, rad sa složenim softverskim paketima im nije blizak, a često im metodologije i postupci kreiranja matematičkih modela i primene matematičkih metoda nisu poznati. Kao posledica toga, neophodno je maksimalno pojednostaviti postupak kreiranja modela problema koji se javljaju u poslovnim procesima.

U cilju što boljeg upravljanja poslovanjem, od poslovnog čoveka se očekuje da razume strukturu poslovnog poduhvata, okolnosti, nameravane akcije, da opiše moguće scenarije odvijanja poslovnih situacija i da na modelu proveri svoje pretpostavke sa aspekta planiranih akcija i očekivanih efekata. Drugim rečima, od operacionog menadžera se očekuje da bude sposoban da izvede analizu „šta će biti ako...?“ i tako dođe do nekog dopustivog odnosno zadovoljavajućeg rešenja.

Ukoliko je operacionom menadžeru omogućena primena tehnike modeliranja poslovnih situacija, komunikacija među članovima radnog tima je značajno olakšana. Članovi tima, u tom slučaju, razgovaraju o dinamičkom aspektu poduhvata i u svakom trenutku mogu simulacijom da provere različite scenarije budućnosti. Takođe, pregovori sa

poslovnim partnerima su mnogo uspješniji kada se obavljaju uz korišćenje dinamičkih simulacionih modela koji su svima razumljivi i dostupni. Pošto je modelar u stanju da matematički korektno definiše sistem i zadatak optimalnog upravljanja, omogućena je uspešna saradnja poslovnog tima sa matematičarima i operacionim istraživačima, koji pokušavaju da odrede rešenja kompleksnih problema.

U današnjem poslovnom okruženju koje je, kao što je već spomenuto, veoma konkurentno i ubrzano, spredšitovi imaju važnu ulogu u obavljanju svakodnevnih aktivnosti i postaju skoro neophodni za funkcionisanje mnogih kompanija. Spredšitovi se upotrebljavaju za obavljanje kancelarijskih zadataka, modelovanje, analize, predviđanje, donošenje odluka itd. Ove aplikacije našle su svoju primenu u širokom opsegu organizacionih funkcija, u različitim oblastima poslovanja, radi ostvarenja ciljeva preduzeća. Spredšitovi su najčešće korišćen i najpopularniji alat kod većine krajnjih korisnika. Mnogi profesionalci iz različitih oblasti donose značajne odluke na osnovu spredšit analiza, a organizacije se oslanjaju na njih pri vođenju evidencija, predviđanju, analizama itd. Usled lakoće učenja rada u spredšitovima, sa jedne strane, i mogućnosti izvođenja kompleksnih analiza sa druge, oni su prihvaćeni od strane velikog broja korisnika od početnika do eksperata, (Lawson et al., 2009). Fleksibilnost spredšitova omogućava njihovo korišćenje, bez primene striktnih pravila. Upravo ta karakteristika čini spredšitove sklonim razvoju grešaka i lošem dizajnu modela i aplikacija. Greške se, usled nedovoljne obučenosti korisnika, lako prave, ali teško uočavaju. Eksperimenti prikazani u (Powell, Baker, & Lawson, 2008) pokazali su da su modeli i aplikacije razvijeni u spredšitovima podložniji greškama nego ostali softveri. U okviru EuSpRiG<sup>1</sup> konferencija, u poslednjih petnaest godina, prikupljeni su i predloženi značajni dokazi o slučajevima grešaka u spredšitovima, koje su prouzrokovale velike finansijske gubitke u kompanijama. Rizik pojave grešaka u spredšitovima izaziva potrebu za unapređenjem metoda i alata za detekciju istih. Svest o riziku ovog tipa u mnogome se razvila u poslednje dve decenije. U skladu sa tim, naučnici i istraživači su predlagali više tehnika i automatizovanih alata za otkrivanje grešaka, koje bi krajnji korisnici primenjivali. Pregled najnovije literature na ovu temu i klasifikacija pristupa za obezbeđenje kvaliteta spredšitova od vizualizacije spredšitova, preko statičkih

---

<sup>1</sup> <http://www.eusprig.org/>

analiza i izveštaja, testiranja do podrške zasnovane na modelima razvoja prikazana je u (Jannach et al., 2014).

U skladu sa svim navedenim činjenicama, osnovni predmet ovog rada predstavlja razvoj i evaluacija novog pristupa za detekciju i analizu grešaka u diskretnim dinamičkim modelima realnih problema operacionog menadžmenta, razvijenim u spredšitovima. Pristup je baziran na karakteristikama problema i načinu modeliranja. Operacioni menadžment (OM), čak i u užem smislu, obuhvata širok skup problema i podproblema koji mogu biti modelirani na različite načine. Kao primer za evaluaciju pristupa korišćen je model upravljanja zalihama, konkretno model ekonomske količine naručivanja za više proizvoda sa ograničenim prostorom skladištenja. Modelovanje je realizovano u skladu sa konceptom diskretnog objekta upravljanja, a model razvijen u spredšitu.

### **1.1. Problem, predmet, cilj i hipoteze doktorske disertacije**

Problemi upravljanja zalihama predstavljaju jednu od najznačajnijih tema u oblasti operacionog menadžmenta, a najčešće korišćeni pristupi za rešavanje ovih problema matematičko modelovanje i simulacija. Koncept diskretnog objekta upravljanja je veoma pogodan za modelovanje i rešavanje realnih problema operacionog menadžmenta, između ostalog problema upravljanja zalihama. On omogućava jasno definisanje zakona ponašanja, oblasti upravljanja, ciljnog funkcionala i metode rešavanja modeliranog problema upravljanja, u cilju dobijanja optimalnog rešenja. Kod diskretnih dinamičkih modela problema operacionog menadžmenta stanje sistema se menja kontinualno, ali se promene stanja registruju na krajevima definisanih vremenskih perioda (*engl. Discrete Time Processes*). Dinamičko okruženje podrazumeva da se razmatraju posledice odluka iz jednog vremenskog perioda na buduće vremenske periode. Opisivanje dinamike ovih sistema se vrši upotrebom diskretnih jednačina i nejednačina. Struktura sistema je poznata, odnosno ona je determinističkog karaktera, dok promenljive u sistemu mogu biti determinističkog i/ili stohastičkog karaktera. Matematički aparat koji se koristi za rešavanje problema upravljanja je optimalno upravljanje diskretnim sistemima.

Spredštovi su veoma pogodni za modelovanje, predstavljanje i simulaciju diskretnog objekta upravljanja. Struktura spredšita je odgovarajuća za predstavljanje različitih tipova i vrednosti promenljivih u vremenskim periodima. Sistem povezanih formula, kreiran u skladu sa elementima diskretnog objekta upravljanja, predstavlja simulacioni model. Simulacija se izvodi izborom vrednosti upravljačkih promenljivih iz jednog u naredni vremenski period, pri čemu su vrednosti svih elemenata modela u svakom periodu povezane. Naredni period se ispituje kada su vrednosti oblasti upravljanja nenegativne za tekući period. Na taj način, sve vrednosti oblasti upravljanja iz prethodnog perioda sigurno obezbeđuju nenegativnost. Određivanje zadovoljavajućih vrednosti upravljačkih promenljivih umnogome zavisi od definicije oblasti upravljanja, kao i drugih relevantnih formula u modelu. Implementacija koncepta diskretnog objekta upravljanja čini spredšitove još prikladnijim alatom za simulaciju, pogodnim za razumevanje i modelovanje poslovne dinamike. Takođe, pristup je u skladu sa preporukama za dizajn i projektovanje spredšit modela i aplikacija. Diskretnim dinamičkim simulacionim spredšit modelom se, na verodostojan način, mogu predstaviti i rešavati problemi operacionog menadžmenta sa kompleksnim matematičkim aparatima, i oni se, na taj način, mogu relativno jednostavno primeniti u realnim sistemima. Spredšit model diskretnog objekta upravljanja implementira konceptualni matematički model, koji odražava znanje i iskustvo domenskog eksperta. Ovakav model omogućava analizu jednostavnom promenom ulaznih i sagledavanjem vrednosti izlaznih promenljivih. Analize su prilagodljive potrebama korisnika u širokom poslovnom opsegu, a izlazi se mogu projektovati u skladu sa njihovim potrebama. Krajnji korisnici spredšit modela i aplikacija su obično domenski eksperti kojima je potreban efikasan alat za rešavanje problema. Oni najčešće nisu zainteresovani za učenje o kompleksnim načinima modeliranja, kao ni metodologija za razvoj softvera. Istovremena primena koncepta diskretnog objekta upravljanja, spredšit modeliranja i simulacije rezultuje alatom za krajnje korisnike, koji je jednostavan za kreiranje, razumevanje i primenu.

Lakoća rada u spredšitovima omogućava krajnjim korisnicima brzu i jednostavnu izradu modela i aplikacija bez striktnih pravila. S obzirom da spredšitovi predstavljaju alat veoma jednostavan i prigodan za upotrebu od strane krajnjih korisnika, oni se mogu



posmatrati i kao izuzetna veza između složenih metoda operacionih istraživanja i tehnika razvijenih u oblasti informacionih sistema, koje bi se mogle primeniti za rešavanje realnih problema u praksi. Međutim, diskretni dinamički spredšit modeli upravljačkih problema operacionog menadžmenta su dodatno podložni nastanku grešaka, usled procesa razvoja koji uključuje prevođenje verbalnih i matematičkih izraza u spredšit formule i funkcije. Krajnji korisnici obično nisu profesionalni programeri i nisu upoznati sa pravilima, metodologijama i standardima razvoja softvera, što rezultuje nizom grešaka i lošim dizajnom modela i aplikacija. Greške se, usled nedovoljne obučenosti korisnika, lako prave, ali vrlo teško uočavaju.

Brojni radovi objavljeni u poslednjih desetak godina: (Hofer et al., 2013), (Panko, 2008), (Powell, Baker, & Lawson, 2009), (Panko, 2005), (Panko & Aurigemma, 2010) i drugi, pokazali su da su spredšitovi podložniji greškama nego ostali softveri za modelovanje poslovnih procesa, usled, kao što je već rečeno, razvoja od strane krajnjih korisnika, bez primene standardnih procedura za obezbeđenje kvaliteta softvera. Za razliku od drugih softvera, koji su validirani i dobro kontrolisani, za spredšitove ne postoje odgovarajuće procedure ni standardi za otkrivanje i analizu grešaka, radi njihovog otklanjanja. Iako je, u poslednjih nekoliko godina, predloženo više metodologija, metoda, tehnika i automatizovanih alata za otkrivanje grešaka, kako se navodi u (Abreu, Ribeiro, & Wotawa, 2012), (Aurigemma & Panko, 2010), (Cunha et al., 2011), (Hermans, 2013), (Hofer & Wotawa, 2014), (Jannach & Schmitz, 2014), (Jannach et al., 2014), (Ruthruff et al., 2005) i drugim radovima, od strane različitih autora, koje bi krajnji korisnici mogli da primenjuju za pomenutu svrhu, većina njih je usmerena na smanjenje grešaka pri razvoju modela u spredšitovima, ali se značajno manji broj radova bavi testiranjem već gotovih modela. Iz tih razloga, ističe se potreba za unapređenjem metoda i alata za detekciju grešaka u već kreiranim spredšit modelima.

Dosadašnji pristupi za obezbeđenje kvaliteta spredšit modela pokazali su se perspektivnim, ali nedovoljno ispitanim. Spredšitovi korišćeni za evaluaciju ovih pristupa obično su namenski napravljeni, ne odnose se na realne primere i relativno su mali, tako da njihova primenljivost u realnim okolnostima, kao i skalabilnost nisu potvrđeni. Rezultati eksperimenata postojećih metoda, dizajniranih za testiranje i

analizu grešaka u spredšitovima (Jannach et al. 2014; Jannach and Schmitz 2014), ostavljaju značajan prostor za unapređenje. Ove metode, u najvećem broju slučajeva, zasnovane su na osobinama spredšit aplikacija ali i idejama iz različitih oblasti kao što su softversko inženjerstvo, operaciona istraživanja i druge. Skoro nijedan od postojećih pristupa ne uzima u obzir karakteristike problema i način modeliranja, koji značajno utiču na nastanak ali i mogućnost otkrivanja grešaka.

U skladu sa prethodno navedenim, opravdan je i neophodan razvoj algoritama koji bi omogućio detekciju i analizu grešaka u modelima razvijenim u spredšitovima, čime bi se postiglo unapređenje kvaliteta dinamičkih diskretnih modela upravljačkih problema operacionog menadžmenta, konkretno upravljanja zalihama.

Predmet istraživanja ove doktorske disertacije je pristup za detekciju i analizu grešaka u dinamičkim diskretnim spredšit modelima upravljanja zalihama, zasnovan na karakteristikama problema i načinu modeliranja, sa akcentom na model ekonomske količine naručivanja zaliha za više proizvoda sa ograničenim prostorom skladištenja.

Spredšitovi i testiranje spredšit aplikacija predstavljaju subjekat istraživanja različitih disciplina iz oblasti informacionih sistema i računarskih nauka. Problem koji se u ovom radu razmatra u kontekstu detekcije i analize grešaka u spredšit modelu, tj. problem upravljanja zalihama modeliran kao dinamički diskretni proces upravljanja u spredšitu, objedinjava probleme iz domena operacionog menadžmenta, operacionih istraživanja, upravljanja materijalnim tokovima i spredšit inženjerstva. Na osnovu navedenog može se reći da analizirani problem i njegovo rešavanje ima multidisciplinarni karakter.

Naučni cilj istraživanja je kreiranje novog pristupa za obezbeđenje višeg kvaliteta dinamičkih diskretnih modela upravljačkih problema operacionog menadžmenta, konkretno upravljanja zalihama, razvijenih u spredšitovima, razvojem algoritma za detekciju i analizu grešaka u navedenim modelima.

Opšti cilj istraživanja je da se primenom ovog pristupa omogući podizanje kvaliteta spredšit modela za upravljanje zalihama, kao i da se kreira alat pogodan za upotrebu od

stane krajnjih korisnika, koji uključuje metode operacionih istraživanja, tehnike razvijene u oblasti softverskog inženjerstva i spredšit inženjerstva.

Na osnovu analize dostupne literature i postavljenog predmeta i cilja istraživanja, postavljena je opšta hipoteza, kao i posebne hipoteze doktorske disertacije, koje su u radu testirane.

Opšta hipoteza istraživanja glasi:

H0: Pristup detekcije i analize grešaka u dinamičkim diskretnim spredšit modelima upravljanja zalihama, zasnovan na karakteristikama problema i načinu modeliranja, kao i na primeni metoda operacionih istraživanja, tehnika razvijenih u oblasti informacionih sistema i spredšit inženjerstva, može da unapredi kvalitet ovih modela, odnosno da omogući detekciju grešaka i utvrđivanje njihovih uzroka na brz i efikasan način.

Posebne hipoteze istraživanja su sledeće:

H1: Problemi upravljanja zalihama modelovani kao dinamički diskretni procesi upravljanja u spredšitu podložni su nastanku grešaka, zbog kompleksnog procesa razvoja strukture i međuzavisnosti elemenata modela.

H2: Postoji širok skup potencijalnih uzroka grešaka u problemima upravljanja zalihama, modelovanim kao dinamički diskretni procesi upravljanja u spredšitu, iz koga je potrebno izdvojiti prave uzroke.

H3: Primenom postojećih, opštih pristupa za detekciju i analizu grešaka u spredšitovima ne može se pouzdano i računarski efikasno identifikovati podskup pravih uzroka grešaka u problemima modelovanim kao dinamički diskretni procesi upravljanja.

H4: Pristup zasnovan na karakteristikama problema i načinu modeliranja može da unapredi detekciju i analizu grešaka spredšit modela.

- H5: Moguće je odrediti intervale za vrednosti elemenata problema upravljanja zalihama modelovanih kao dinamički diskretni procesi upravljanja u spredšitu, koji omogućavaju detekciju grešaka.
- H6: Uvođenjem dodatnih pravila o osobinama problema upravljanja zalihama širok skup potencijalnih uzroka grešaka u spredšit modelu može se suziti na redukovani skup.
- H7: Dinamički diskretni spredšit modeli upravljanja zalihama sadrže pravila kojima se dodatno može redukovati početni širok skup potencijalnih uzroka grešaka modela.

## **1.2. Struktura doktorske disertacije**

U prvom poglavlju doktorske disertacije definisani su predmet i problem istraživanja, kao i način na koji će se analizirati postavljeni problem istraživanja. Postavljen je cilj i hipoteze koje će biti testirane eksperimentalnom proverom, primenom naučnog metoda opisanog u nastavku disertacije. Uvodni deo rada u kratkim crtama prikazuje i opisuje strukturu rada, sažetim pregledom svakog poglavlja.

Drugo poglavlje disertacije opisuje koncepte operacionog menadžmenta, kao i značajnosti i aktuelnosti tema i problema koje obuhvata. Na osnovu analizirane literature, utvrđene su karakteristike problema upravljanja zalihama, kao jednog od najznačajnijih predstavnika problema operacionog menadžmenta, koji u užem smislu obuhvata veoma širok dijapazon poslovnih i naučnih disciplina. U okviru ovog poglavlja klasifikovani su tipovi modela za upravljanje zalihama.

Treće poglavlje odnosi se na dinamičke diskretne upravljačke modele. U ovom delu disertacije razmatraju se pojam i tipovi modela, posebno simulacija i simulacioni model, koji će biti korišćeni za predstavljanje upravljačkih problema, kao dinamičkih diskretnih modela, razvijenih u spredšitu. Dodatno, definiše se pojam diskretnog objekta upravljanja, optimalno upravljanje diskretnim objektom, kao i osnovni elementi modela objekta diskretnog upravljanja.

U četvrtom poglavlju prikazan je istorijat razvoja spredšitova, značaj spredšitova kao analitičkog alata i alata za modelovanje opšteprihvaćenog u današnjem poslovnom okruženju, koncept programiranja od strane krajnjih korisnika, kao i karakteristike kvaliteta analitičkih spredšit modela upravljačkih problema. Objasnjene su koncept spredšit simulacije, uz navođenje mogućnosti i prednosti koje on pruža korisnicima.

Peto poglavlje predstavlja sublimaciju prethodnih poglavlja, opisivanjem problema upravljanja zalihama modelovanih kao dinamički diskretni procesi upravljanja u spredšitu. U ovom poglavlju prikazan je dinamički diskretni model ekonomske količine naručivanja zaliha (*Economic Order Quantity* - EOQ), uz konkretizaciju kroz model ekonomske količine naručivanja za više proizvoda, sa ograničenim prostorom skladištenja. Navedeni modeli su predstavljeni kao objekti diskretnog upravljanja u spredšitu, definisanjem odgovarajućih elemenata. Opisane su prednosti modelovanja u spredšitu u skladu sa konceptom diskretnog objekta upravljanja. Modeli čiji je razvoj prikazan u ovom poglavlju, u narednim delovima disertacije korišćeni su kao primeri za evaluaciju pristupa za detekciju i analizu grešaka u implementaciji dinamičkih diskretnih modela upravljanja zalihama.

U šestom poglavlju rada predstavljeni su postojeći pristupi za obezbeđenje kvaliteta modela u spredšitovima, njihova analiza i komparacija. Prikazane su kategorije pristupa koje obuhvataju: vizualizaciju spredšitova, statičke analize i izveštaje, testiranje, automatsku detekciju i ispravku grešaka, razvoj spredšitova vođen modelom, projektovanje i održavanje, kao i podkategorije svake navedene grupe. U ovom delu disertacije prikazane su i definicije grešaka u spredšit modelima, kao i taksonomije ovih grešaka, definisane od strane različitih autora.

Sedmo poglavlje rada sadrži prikaz dizajna, implementacije i testiranja novog pristupa za detekciju i analizu grešaka u dinamičkim diskretnim modelima upravljanja zalihama razvijenim u spredšitu (DDMUZRS). U ovom delu rada prikazani su osnovni koncepti i koraci pristupa, zasnovanog na karakteristikama problema i načinu modeliranja. Kompletan pristup za detekciju i analizu grešaka u DDMUZRS predstavljen UML 2.0 dijagramom aktivnosti, dok su pojedinačne faze pristupa prikazane i objašnjene

izdvojenim algoritmima. Kao dodatak opisu koncepta, predstavljeni su i različiti načini za generisanje test slučajeva, koji su neophodni za evaluaciju pristupa. Nakon prikaza novog pristupa za detekciju i analizu grešaka u DDMUZRS, predstavljena je implementacije istog i eksperimentalni rezultati testiranja. Eksperimenti su sprovedeni nad modelima prikazanim u petom poglavlju, koji razmatraju 3, 20 i 100 proizvoda, za različiti broj vremenskih perioda posmatranja. Analizom rezultata potvrđena je efikasnost i skalabilnost razvijenog pristupa, dok se vreme potrebno za realizaciju pokazalo prihvatljivim u odnosu na postojeće pristupe. U poslednjem potpoglavlju sedmog poglavlja doktorske disertacije prikazana je uporedna analiza postojećih i novog pristupa za detekciju i analizu grešaka u spredšitovima.

Osmo poglavlje sadrži zaključke istraživanja, potvrde ostvarenja postavljenih ciljeva i rezultate testiranja hipoteza istraživanja. Takođe, sadrži i pregled ostvarenih naučnih i stručnih doprinosa, kao i pravce budućih istraživanja.

Poslednje poglavlje doktorske disertacije sadrži spisak literature koja je korišćena prilikom izrade ove doktorske disertacije. Literatura sadrži skup relevantnih i korišćenih referenci iz oblasti na kojima se zasniva doktorska disertacija, ukupno 199 referenci, koje su korišćene tokom istraživanja. Doktorska disertacija je napisana na 191 strani, ilustrovana sa 27 slika i 21 tabelom.

## 2. Upravljački problemi operacionog menadžmenta

Operacioni menadžment predstavlja amalgam više naučnih disciplina i polje na kome se praktično primenjuju dostignuća tih disciplina. Stoga se Operacioni menadžment može posmatrati kao: a) Nauka o upravljanju procesima i resursima organizacionih sistema (iz privrede i vanprivrede, tj. kod proizvodnje ili/i pružanja usluga), b) Praksa upravljanja operacionom funkcijom (funkcijom proizvodnje ili/i pružanja usluga), c) Grupa ljudi koji se bave upravljanjem operacionom funkcijom u organizacionim sistemima, (Omerbegović-Bijelović i drugi, 2010).

Operacioni menadžment obuhvata, između ostalog, upravljanje procesima i resursima koje preduzeća koriste, da bi isporučila vrednost koja će zadovoljiti očekivanja kupaca (potrošača i/ili korisnika njihovih proizvoda i/ili usluga), sa naglaskom na efikasnost i efektivnost tih procesa. Vrednost se ovde definiše kao odnos benefita i troškova. Osnovni elementi vrednosti su kvalitet, troškovi i vremenski rokovi, (Vasiljević & Jovanović, 2008).

Terminom operacije se označavaju sve neposredno povezane aktivnosti sa proizvodnjom proizvoda i/ili usluga, dok se terminom operacioni menadžment označava funkcija preduzeća koja doprinosi da se vrednost isporuči kupcima, (Shtub, 1999). Operacioni menadžment predstavlja funkciju koja omogućava organizacijama da ispune svoje ciljeve, kroz efikasnu nabavku i korišćenje resursa, (Chase, Jacobs, & Aquilano, 2006). Prema (Vasiljević, 2001), operacioni menadžment se prepoznaje kao poslovna funkcija kojom se uspostavlja sistem dodavanja vrednosti u proizvodnim i uslužnim preduzećima, dok se operacijama dodaje vrednost, koju preduzeće isporučuje kupcu.

Savremeni koncept operacionog menadžmenta omogućava upravljanje procesom, kojim se resursi na ulazu konvertuju u proizvode na izlazu, (Antić, 2012). Ulazi su sirovine, ljudski resursi, postrojenja, mašine i oprema, energija i informacije. Izlazi su u obliku proizvoda, usluga, a najčešće paketa proizvoda-usluga. Kupci imaju ključnu ulogu pri specificiranju izlaza. Najčešće kupci očekuju mnogo više od samog proizvoda ili

usluge, te se od jednog prodavca očekuje integrisana vrednost u obliku paketa proizvod-usluga.

Operacioni menadžment, u širem smislu, obuhvata donošenje odluka i rešavanje problema koji uključuju primenu kvantitativnih metoda operacionih istraživanja i nauke o menadžmentu za podršku efikasne i efektivne alokacije deficitarnih resursa, u vezi sa operacijama organizacije, prema (Mentzer, Stank, & Esper, 2008). Operacioni menadžment, u užem smislu, obuhvata: agregatno planiranje, planiranje kapaciteta, projektovanje objekata i lokacija, predviđanje, održavanje, raspoređivanje radnika, projektovanje procesa, upravljanje projektima, kvalitet radnog života, raspoređivanje poslova i merenje rada, prema (Mentzer, Stank, & Esper, 2008).

Operacioni menadžment uključuje sledeće oblasti i discipline, prema (Pawar, 2011):

- Projektovanje i analiza proizvoda i usluga;
- Projektovanje i analiza proizvodnih/uslužnih procesa;
- Upravljanje kvalitetom i tehnike kvaliteta;
- Predviđanje, planiranje, raspoređivanje i kontrola;
- *Just-in-Time* proizvodnja, štedljiva proizvodnja i teorija ograničenja;
- Upravljanje lancima snabdevanja;
- Projektovanje lokacija i objekata;
- Projektovanje radnog mesta i merenje radnih učinaka;
- Upravljanje ljudskim resursima u operacijama, učenje, kontinualno unapređenje i integracija;
- Upravljanje projektima;
- Kvantitativne metode i modeli, optimizacija i simulacija;
- Operaciona strategija preduzeća.

Prema (Schroeder, 2007), operacioni menadžmenta se može definisati na sledeći način:

- Operacije su odgovorne za snabdevanje organizacije proizvodima i uslugama;
- Operacioni menadžeri donose odluke shodno operacijama i njihovoj vezi sa drugim funkcijama. Operacioni menadžer planira i kontroliše proizvodni ili uslužni proces i njegov uticaj na organizaciju i spoljno okruženje;



- Operacioni menadžment je oblast koja se bavi proizvodnjom dobara i usluga, i fokusirana je na unapređenje procesa.

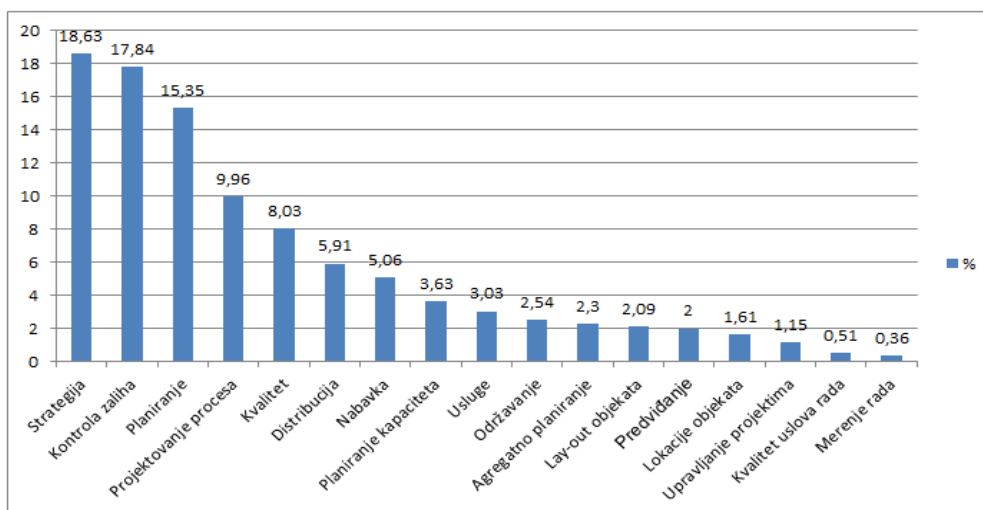
**Tabela 2-1.** Ukupan broj tema u oblasti OM-a u istraživanjima izvedenim 1989, 1999 i 2014, (Tomašević, Stojanović, & Simeunović, 2014)

Teme	[Amo89]		[Pen99]		[Tom14]	
	Rang	%	Rang	%	Rang	%
Kontrola zaliha	1	28,15	2	16,19	2	17,84
Planiranje	2	16,89	1	25,54	3	15,35
Projektovanje procesa	3	15,23	4	11,29	4	9,96
Agregatno planiranje	4	11,26	15	0,66	11	2,30
Usluge	5	6,29	8	2,71	9	3,03
Kvalitet	6	3,97	3	11,34	5	8,03
Strategija	7	3,64	5	10,88	1	18,63
Upravljanje projektima	8	3,31	12	1,74	15	1,15
Planiranje kapaciteta	9	2,98	9	2,09	8	3,63
Održavanje	10	2,65	10	2,04	10	2,54
Nabavka	11	1,66	11	1,99	7	5,06
Lay-out objekata	12	1,66	6	7,61	12	2,09
Predviđanje	13	1,32	14	1,02	13	2,00
Kvalitet uslova rada	14	0,33	17	0,20	16	0,51
Lokacije objekata	15	0,33	13	1,07	14	1,61
Distribucija	16	0,33	7	3,12	6	5,91
Merenje rada	17	0,00	16	0,51	17	0,36
Ukupno		100,00		100,00		100,00

Aktuelnost tema i problema koji pripadaju oblasti operacionog menadžmenta može se analizirati na osnovu njihove zastupljenosti u međunarodnim časopisima. U radu

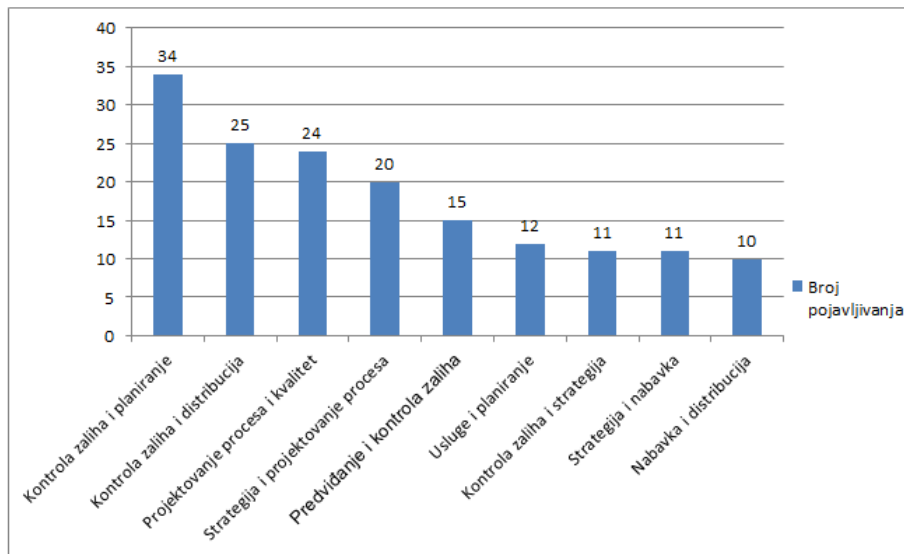
(Tomašević, Stojanović, & Simeunović, 2014) prikazano je istraživanje o zastupljenosti tema iz oblasti operacionog menadžmenta u međunarodnim časopisima, objavljenih u periodu od 2009. do 2012. godine.

Osnovni cilj rada je uočavanje trendova u oblasti OM-a i poređenje sa prethodnim istraživanjima sprovedenim 1980. i 1990. godine, koja su predstavljena u radovima (Amoako-Gyampah & Meredith, 1989) i (Pennirselvam, Ferguson, & Ash, 1999), a da bi se utvrdili trendovi u oblasti OM-a u 21. veku. Istraživanja prikazana u (Amoako-Gyampah & Meredith, 1989) i (Pennirselvam, Ferguson, & Ash, 1999), odnose se na međunarodne časopise iz perioda 1982-1987. i pokazuju da su najzastupljenije bile teme iz domena upravljanja zalihama i kontrole zaliha, a korišćene metode istraživanja su modelovanje i simulacija. Uporedni prikaz rezultata istraživanja dva navedena rada i rada (Tomašević, Stojanović, & Simeunović, 2014) prikazan je Tabelom 2-1. U cilju istraživanja analizirano je 2802 radova. Svi radovi su klasifikovani prema 17 ključnih kriterijuma, odnosno tema koje se pojavljuju u časopisima iz oblasti OM-a. U istraživanju su korišćeni časopisi: *International Journal of Operations and Production Management (IJOPM)*, *International Journal of Production Research (IJOPR)*, *Journal of Operations Management (JOM)*, *Production and Operations Management (POM)*, *Decision Sciences (DS)*, *Management Science (MS)*, *IIE Transactions (IET)* and *International Journal of Production Economics (IJPE)*.



**Slika 2-1.** Zastupljenost pojedinačnih tema u radovima iz oblasti OM-a, (Tomašević, Stojanović, & Simeunović, 2014)

Na osnovu istraživanja može se zaključiti da tri najznačajnije teme u oblasti OM-a čine preko 50% posmatranog uzorka, odnosno da su najznačajnije teme: strategija u OM-u, upravljanje i kontrola zaliha i planiranje, kao što je prikazano na Slici 2.1. Kada se govori o kombinacijama tema iz domena OM-a u analiziranim radovima, može se uočiti da dve najznačajnije kombinacije tema (od prikazanih devet) čine skoro 60% posmatranog uzorka, a to su: upravljanje zalihama/planiranje i upravljanje zalihama/distribucija, što se može videti na Slici 2.2.



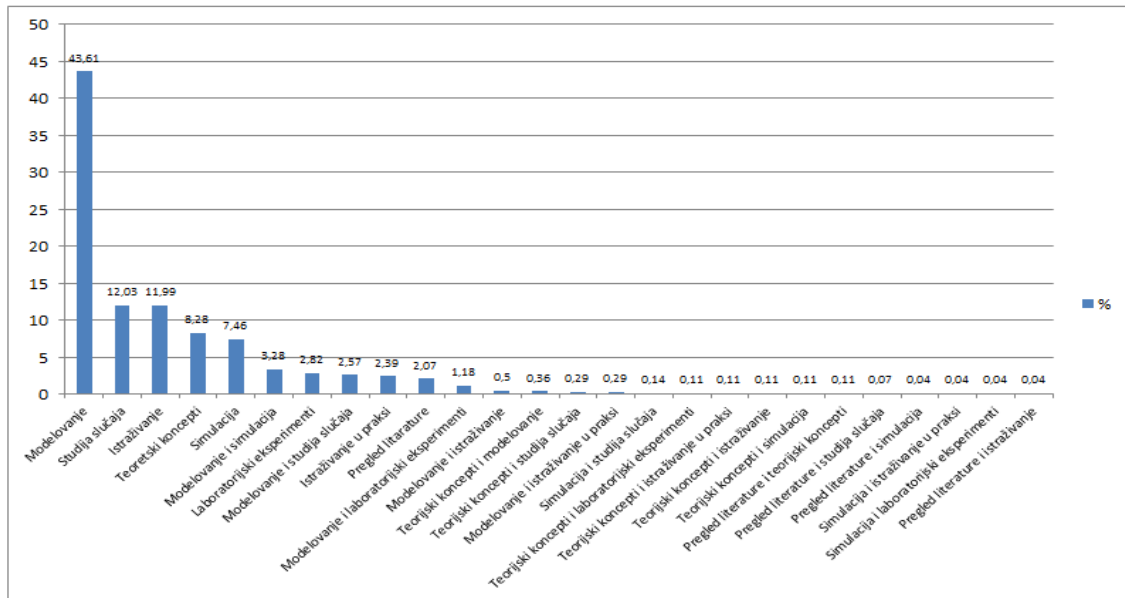
**Slika 2-2.** Zastupljenost kombinacija tema u radovima u oblasti OM-a, (Tomašević, Stojanović, & Simeunović, 2014)

Kada se govori o metodologiji korišćenoj za izradu posmatranih naučnih radova, može se zaključiti da prvih pet različitih metodoloških pristupa (od ukupno 26) čini preko 83% posmatranog uzorka.

Najčešće korišćeni metodološki pristupi su: matematičko modelovanje, studija slučaja, istraživanje, teorijski koncepti i simulacija, što se može videti na Slici 2-3.

Na osnovu predstavljenog istraživanja (Tomašević, Stojanović, & Simeunović, 2014), može se zaključiti da je u poslednje četiri godine najzastupljenija tema akademskih radova iz oblasti OM-a, upravljanje i kontrola zaliha, kao upravljački problemi operacionog menadžmenta. Takođe, najzastupljenija metodologija za izradu radova je matematičko modelovanje i simulacija.

U skladu sa ovim zapažanjem, tema ovog rada je izuzetno aktuelna i u naučnim istraživanjima veoma zastupljena. Kako operacioni menadžment, u užem smislu, obuhvata veoma širok dijapazon poslovnih i naučnih disciplina i oblasti, ovaj rad je usmeren samo na probleme upravljanja zalihama u operacionom menadžmentu.



**Slika 2-3.** Metodologije za izradu naučnih radova u oblasti OM-a, (Tomašević, Stojanović, & Simeunović, 2014)

Jedna od ključnih funkcija u upravljanju preduzećem je funkcija upravljanja zalihama. Značaj funkcije upravljanja zalihama proizilazi iz veoma visoke novčane vrednosti zaliha, kojima preduzeće raspolaže. Osnovni problem koji se rešava upravljanjem zalihama je zadovoljavanje potreba krajnjih kupaca i rešavanje vremenskog nesklada između tražnje i snabdevanja. Zbog toga se zalihama poklanja ogromna pažnja i problemi u upravljanju zalihama predstavljaju jedan od najčešće rešavanih problema u praksi. Značaj upravljanja zalihama se može videti na mnogim realnim primerima u praksi, a u većini kompanija postoje i posebna odeljenja koje se bave upravljanjem zalihama. Upravljanje zalihama u realnosti, ne podrazumeva više samo: praćenje nivoa zaliha u skladištu (u količinskim jedinicama i u broju dana vezivanja zaliha), jednostavno poručivanje ne vodeći računa o predviđanju, kao i određivanje količine za naručivanje samo na osnovu prethodne prodaje, već se sve više traži upotreba savremenih metoda operacionih istraživanja u cilju rešavanja ovog problema. Može se

tvrditi da se u kompanijama danas intenzivno traži znanje matematičkog modeliranja, kao i znanje heurističkih metoda, u cilju rešavanja problema u upravljanju zalihama, u što kraćem vremenskom periodu.

Metodološki pristupi koji će se primeniti su veoma aktuelni, a kao najznačajniji se mogu izdvojiti: istraživanje literature, matematičko modelovanje, simulacija, metode optimizacije, metoda eksperimenta (studije slučaja) u spremit okruženju.

## **2.1. Problemi upravljanja zalihama**

Jedan od ključnih procesa operacionog menadžmenta je proces upravljanja zalihama (*engl. Inventory Management/Inventory Control*), (Antić, 2012). Značaj funkcije upravljanja zalihama proizilazi iz vrednosti zaliha kojima preduzeće raspolaže. U industrijski razvijenim zemljama, upravljanju zalihama se poklanja značajno veća pažnja nego u industrijski manje razvijenim zemljama. Osnovni problem koji se rešava upravljanjem zalihama jeste vremenski nesklad između tražnje i snabdevanja, odnosno nesklad između potreba i raspoloživosti materijala koji se koristi u proizvodnji. Zahtev koji treba da ispuni upravljanje zalihama jeste obezbeđenje potrebnih materijala, tako da se proces proizvodnje može odvijati ravnomerno i neprekidno. Postavljeni zahtev treba ispuniti uz što je moguće niže ukupne troškove, koji nastaju u upravljanju zalihama, (Krčevinac i drugi, 2004).

U poslovanju sa kupcima može se samo predviđati šta kupci žele da kupe, odnosno koje proizvode žele da kupe i kada će određeni proizvod kupiti, pod uslovom da se proizvod već nalazi na tržištu. U cilju prevazilaženja neizvesnosti, držanje zaliha proizvoda omogućava zadovoljenje tražnje kupaca, a zalihe predstavljaju rezervu koja treba da nadomesti neizvesnost između intenziteta snabdevanja i intenziteta tražnje, (Antić, 2012). Kao najprostiji primer, prema (Barlow, 2003), mogu se posmatrati banke, koje moraju imati određenu zalihu novca da bi zadovoljile tražnju kupaca za novcem.

Kao osnovne funkcije zaliha, prema (Antić, 2014), mogu se navesti sledeće:

- Zadovoljavanje neizvesnosti i promena u tražnji za proizvodima;

- Obezbeđivanje neprekidnog toka proizvodnje, odnosno zadovoljenje fleksibilnih proizvodnih planova;
- Zaštita sistema od proboja (nedostatka) zaliha (*engl. Stock-out*);
- Obezbeđivanje stabilnosti u lancu snabdevanja, odnosno obezbeđivanje stabilnog ciklusa nabavljanja i proizvodnje;
- Zaštita sistema od posledica povećanja cena zaliha na tržištu, kupovinom većih količina zaliha po nižim cenama;
- Omogućavanje povećanja prodaje, kroz davanje određenih popusta na količinu;
- Stvaranja boljih uslova prilikom nabavljanja, ostvarujući popuste na kupljenu količinu;
- Smanjenje organizacionih napora u planiranju u lancu snabdevanja;
- Omogućavanje apsorpcije kašnjenja učesnika u operacijama lanca snabdevanja.

Upravljanje zalihama je jedan od najvažnijih zadataka operacionog menadžmenta, zato što zalihe predstavljaju veliki deo uloženog kapitala i direktno utiču na isporuke robe kupcima. Upravljanje zalihama ima veliki uticaj na ostale poslovne funkcije u kompaniji, posebno na logistiku, marketing, finansije i itd. U (Schroeder, 2007) zalihe se definišu kao količina materijala koja se koristi za održavanje procesa proizvodnje ili za zadovoljenje tražnje kupaca. Ako se posmatraju zalihe u jednom poslovnom sistemu, može se reći da one uključuju zalihe sirovina, zalihe nedovršene proizvodnje ili poluproizvoda i zalihe gotovih proizvoda. U slučaju sagledavanja operacija jednog proizvodnog transformacionog procesa, može se reći da zalihe prelaze iz jednog procesa u drugi proces, menjajući svoje transformacione karakteristike, tako sirovine ulaze ili čekaju na proces proizvodnje, u procesu proizvodnje sirovine postaju zalihe u proizvodnji, a na kraju transformacije predstavljaju gotove proizvode, (Schroeder, 2007). Prema (Ballou, 2004), zalihe se definišu kao skupovi sirovina, komponenata, nedovršene proizvodnje i gotovih proizvoda, koje se pojavljuju na više mesta (akumulacija) u okviru proizvodnje i logistike. U kompanijama zalihe se nalaze na različitim mestima, kao što su: zatvorena skladišta, otvorena skladišta, prodajna mesta, u transportu, na prodajnim policama maloprodajnih objekata. Procenjuje se da držanje zaliha može da košta godišnje između 20% i 40% njihove vrednosti. Stoga, pažljivo upravljanje nivoom zaliha omogućava smanjenje troškova angažovanog kapitala. Neki

autori definišu zalihe kao potencijalnu ekonomsku dodatnu vrednost, ali većina autora i kompanija, u praksi, smatra da kapital u zalihama, predstavlja pogrešno vezan ili zarobljen kapital. Ovo se posebno dešava u slučajevima kada je trošak držanja zaliha veći od profita ostvarenog prodajom istih. U takvim situacijama, kompanije usvajaju strategiju nedostatka (proboja) zaliha, koje omogućavaju da potrebe za proizvodima od strane kupca budu zadovoljene tek posle određenog vremenskog perioda, nakon što je porudžbina za proizvodima poslata od strane kupca. Ovakva strategija upravljanja zalihama omogućava manje vezivanje kapitala kompanije u zalihama, ali i pretpostavlja činjenicu da je kupac voljan da čeka određeni vremenski period isporuku proizvoda.

Kao osnovne vrste zaliha, prema (Barlow, 2003), mogu se navesti: zalihe sirovina, zalihe nedovršenih proizvoda (proizvodnje), zalihe gotovih proizvoda i zalihe rezervnih delova - potrošnih materijala i zalihe u tranzitu (u transportu od skladišta do kupaca). Prema (Hopp & Spearman, 1996), u zavisnosti od prirode tražnje, definišu se tri osnovna tipa zaliha:

- Zalihe sa nezavisnom tražnjom - predstavljaju zalihe proizvoda koje se prodaju krajnjim kupcima;
- Zalihe sa zavisnom tražnjom - predstavljaju zalihe čija je tražnja uslovljena tražnjom za drugim zalihama. Kao primer može se navesti proizvodnja automobila, čija tražnja direktno utiče na tražnju i zalihe automobilskih guma. Proizvodi sa zavisnom tražnjom se koriste u proizvodnji proizvoda sa nezavisnom tražnjom;
- Zalihe ostalog materijala - predstavljaju zalihe proizvoda kao što su kancelarijski papir, sredstva za čišćenje itd., koji ne koriste direktno u proizvodnji proizvoda sa nezavisnom tražnjom.

U zavisnosti od specifične svrhe ili lokacije u proizvodno-distribucionom procesu, zalihe mogu biti klasifikovane na različite načine, (Salvendy, 2007):

- Prema tipovima subjekata materijalnih tokova;
- Prema nameni (prim. autora disertacije), na proizvodne i distributivne zalihe;
- Prema svrsi (prim. autora disertacije), na tranzitne i organizacione zalihe.

Prema istom izvoru, zalihe se prema tipu subjekata materijalnih tokova mogu podeliti na:

- Zalihe sirovina – odnose se na zalihe koje se nabavljaju u cilju proizvodnje pojedinačnih delova, komponenti i podkomponenti proizvoda, koje se dalje koriste za finalnu montažu gotovih proizvoda;
- Zalihe gotovih sastavnih delova – odnose se na sastavne delove koji nezavisno ulaze u sastav finalnog proizvoda, zajedno sa sirovinama za proizvodnju;
- Zalihe komponenti – ova klasa zaliha podrazumeva zalihe koje su nabavljene kao gotove komponente, koje nije potrebno dodatno obrađivati i koje ulaze u sastav proizvoda tokom montaže u proizvodnji;
- Zalihe podsklopova i sklopova – odnose se na zalihe poluproizvoda, koji su nastali spajanjem sastavnih delova i komponentata u proizvodnji. Firme proizvode podsklopove i sklopove, skladište iste na posebna mesta u proizvodnji, da bi u slučaju potrebe, proizvodni proces mogao biti fleksibilniji i u što kraćem roku odgovoriti na zahteve tražnje;
- Zalihe nedovršene proizvodnje – odnose se na zalihe koje nastaju na transportnim putevima između mašina u proizvodnom procesu, prilikom čekanja komponente ili podsklopa na obradu na određenoj mašini;
- Zalihe gotovih proizvoda – odnose se na zalihe finalnih proizvoda, koje su spremne za isporuku kupcima.

Prema drugoj klasifikaciji (Salvendy, 2007), zalihe se mogu podeliti na proizvodne i distributivne zalihe. Distributivne zalihe se odnose na zalihe gotovih proizvoda, koje su spremne za isporuku kupcima. Ukoliko kompanija ima geografski veoma raširen sistem distribucije, ove zalihe gotovih proizvoda se nalaze po skladištima distributivnih centara ili u maloprodajnim objektima kompanije. Fabrika, odnosno proizvođač, nema nikakvog uticaja na politiku upravljanja ovim zalihama, pa se iz tog razloga ove zalihe zovu distributivne zalihe. Za razliku od njih, postoje proizvodne zalihe, koje uključuju sve od sirovih materijala do zaliha podsklopova i sklopova u proizvodnji. Osnovi razlog zbog koga se pravi ova podela zaliha je organizaciono-upravljačke prirode, jer su često kompanije organizovane tako da proizvodnja predstavlja jednu organizacionu celinu sa posebnom upravljačkom strukturom, dok distribucija predstavlja drugu organizacionu celinu sa svojom nezavisnom upravljačkom strukturom.



Prema trećoj klasifikaciji (Salvendy, 2007) Zalihe se mogu podeliti na tranzitne i organizacione zalihe. Tranzitne zalihe se mogu dalje klasifikovati kao zalihe u „kretanju“ ili zalihe u „cevovodima“. Prva klasa zaliha predstavlja zalihe sirovina ili gotovih proizvoda, koje se transportuju unutar ili izvan kompanije. Glavna karakteristika ovih zaliha je da se njihov transport odvija u etapama. Kompanije na različite načine pokušavaju da smanje troškove transporta zaliha i u tom cilju teže da transport zaliha bude kontinualan. Nekada vrsta i fizičko stanje zaliha to omogućavaju (tečnosti, granulat i itd.), pa kompanije ulažu ogromna sredstva u izradu cevovoda, u kojima će se ostvariti kontinualan tok zaliha, a time i niži troškovi transporta po jedinici proizvoda. Zalihe u „cevovodima“ su zalihe koje omogućavaju kontinualno snabdevanje krajnjih korisnika proizvodima.

Organizacione zalihe su definisane prema specifičnoj svrsi za koju su namenjene. Zalihe predstavljaju važnog posrednika između procesa nabavke i tražnje (Antić, 2014). Zalihe gotovih proizvoda smanjuju zavisnost proizvodnje od promene u tražnji kupaca. One smanjuju organizacione napore potrebne za balansiranje između nabavke i tražnje. Sa druge strane, smanjenje zaliha često zahteva efikasnije planiranje i kontrolu, da bi se operacije u lancu snabdevanja realizovale sa istom lakoćom, kao kada su te zalihe bile na višem nivou. Organizacione zalihe se mogu podeliti u tri kategorije, (Antić, 2014): ciklične zalihe, sigurnosne zalihe i anticipirane zalihe. Ciklične zalihe se obnavljaju u konstantnim intervalima (ciklusima) vremena ili se proizvede u serijama koje su veće od postojeće potrebe. Često se nazivaju i serijskim zalihama (*engl. Lot Size Inventory*). Sigurnosne ili rezervne zalihe, imaju za cilj zadovoljenje tražnje prilikom iznenadnih događaja, koji nisu predviđeni u prethodnim periodima (kvar mašine u proizvodnoj liniji, nerealizovane isporuke i itd.). Anticipirane zalihe se često nazivaju i sezonskim zalihama, koje nastaju predviđanjem potreba u budućim periodima poslovanja. Ukoliko kompanija prodaje sezonske proizvode, u periodima kada nije sezona kompanija mora da izvrši predviđanje i nabavku zaliha koje će se prodavati u sezonskim periodima. Anticipirane zalihe se, takođe, formiraju zbog ograničene dostupnosti sirovinama u određenim periodima (npr. period žetve, kada se skupljaju usevi radi trošenja u periodima kada nema žetve).

Osnovna svrha upravljanja zalihama i kontrole zaliha (Barlow, 2003) je da se osigura da tačna količina tačno određenog proizvoda bude poručena u pravo vreme. Da bi ova pretpostavka bila ispunjena neophodno je prvo odgovoriti na sledeća pitanja:

1. Koje proizvode (sirovine) treba naručiti;
2. Kada porudžbinu treba napraviti;
3. Koliko proizvoda ili sirovina treba poručiti.

Prvo pitanje ukazuje na važnost izbora proizvoda (sirovina) koje treba poručiti. Osnovni razlog držanja određenih proizvoda (sirovina) na zalihama je (Barlow, 2003):

- Omogućavanje neometanog i efikasnog procesa proizvodnje;
- Dobijanje popusta (rabata) na kupovinu veće količine zaliha proizvoda (sirovina);
- Zaštita od mogućnosti nastanka proboja zaliha usled većih fluktuacija na strani tražnje;
- Pravljenje rezervi zaliha usled sezonskih oscilacija u nabavci proizvoda (sirovina);
- Zaštita od inflacije i promena cena na tržištu.

Kada se radi o vremenu za naručivanje zaliha, kao odgovor na drugo pitanje, mogu se definisati dva osnovna pristupa (Barlow, 2003):

- Prema prvom pristupu, se u svakom postupku nabavke poručuje ista količina zaliha. Bilo kakva varijacija u tražnji za proizvodima ili sirovinama se prevazilazi promenom vremena između porudžbina, jer se uvek naručuju iste količine;
- Prema drugom pristupu, porudžbine se lansiraju u fiksnim vremenskim intervalima u zavisnosti od količine zaliha proizvoda ili sirovina na stanju. U slučaju bilo kakve varijacije u tražnji za proizvodima ili sirovinama menja se veličina porudžbine. Bilo kakva varijacija u tražnji za proizvodima ili sirovinama se prevazilazi menjanjem naručene količine, odnosno veličine porudžbine, jer su periodi između dva naručivanja uvek isti.

Odgovor na treće pitanje je značajno teže odrediti, jer je potrebno imati model na osnovu kojeg će se izračunati optimalna količina zaliha za poručivanje. Postoje različiti

matematički modeli za određivanje optimalne veličine narudžbine sa najnižim troškovima. Najpoznatiji matematički model za određivanje optimalne veličine porudžbine po najnižim troškovima je EOQ model ili model ekonomske količine naručivanja zaliha (*engl. Economic Order Quantity*).

## **2.2. Osnovni troškovi u procesu upravljanja zalihama**

Kako troškovi zaliha predstavljaju bitan deo ukupnih troškova proizvodnje, efikasnost upravljanja zalihama značajno se ogleda u efikasnom merenju i kontroli troškova držanja zaliha (Krčevinac i drugi, 2004). Kada su u pitanju problemi upravljanja zalihama, moguće je definisati različite vrste troškova. Prema istom izvoru, u analizi troškova zaliha razmatraju se četiri grupe troškova:

- Troškovi nabavke zaliha – obuhvataju sve troškove koji nastaju od trenutka utvrđivanja potrebe za robom, do trenutka dobijanja robe na skladištu firme. U širem smislu, ovi troškovi obuhvataju i nabavnu cenu same robe. Većina modela u okviru ovog troška ne uračunava cenu robe, nego se samo razmatraju troškovi pripremanja narudžbine, njene realizacije i transporta robe do skladišta firme. U modelima koji se odnose na sopstvenu proizvodnju u samoj organizaciji, umesto troškova nabavke, računaju se i troškovi pripreme proizvodnje. Kao najznačajniji troškovi u okviru ovog troška mogu se navesti: plate zaposlenih u nabavnoj službi, ptt usluge, kancelarijski materijal i transport. Obračun troškova nabavke se vrši tako, što se svi sastavni troškovi sabere za jedan obračunski period, tako da se dobiju ukupni troškovi naručivanja u obračunskom periodu. Kada se ovi troškovi podele sa brojem narudžbina u datom periodu, dobijaju se prosečni jedinični troškovi po narudžbini;
- Troškovi držanja zaliha – nastaju iz činjenice da preduzeće poseduje zalihe. U najznačajnije stavke ovog troška spadaju: troškovi kapitala, troškovi prostora i troškovi zaposlenih koji rade u skladištu. Troškovi držanja zaliha se obično izražavaju procentom od prosečne vrednosti zaliha. Da bi se dobio ovaj podatak potrebno je znati vrednost prosečnih zaliha u toku obračunskog perioda i sve troškove koji nastaju usled držanja zaliha. Takođe, jedinični troškovi držanja zaliha se izražavaju u vrednosti po jedinici zalihe u jedinici vremena;

- Troškovi nedostatka zaliha (troškovi penala) – obično se klasifikuju u dve grupe: troškovi usled prekida rada i zastoja u proizvodnji i troškovi usled vanrednih postupaka zbog nedostatka zaliha. Zastoj u proizvodnji usled nedostatka zaliha ima kao posledicu propuštanje šanse da se zaradi, odnosno dolazi do gubitka oportuniteta. Nedostatak zaliha izaziva slabljenje poverenja i dobre volje kod klijenata. Vanredni postupci i aktivnosti mogu stvoriti dodatne troškove u vezi sa tehnološkim procesima, odnosno troškove hitne nabavke, vanrednog naručivanja, hitne izrade i itd. Troškovi nedostatka zaliha se obično izražavaju u vrednosti po jedinici robe i jedinici vremena. Ovu vrstu troškova je izuzetno teško meriti, pa se oni po pravilu procenjuju. Narušavanje ugleda firme, usled nedostatka zaliha je izuzetno teško meriti i izraziti kvantitativno putem iznosa troška. Stoga se u matematičkim modelima umesto troškova nedostatka zaliha koriste drugi pokazatelji u vezi sa snabdevenošću skladišta potrebnom robom. To su koeficijent snabdevenosti, kao verovatnoća da će na skladištu biti raspoloživa roba u trenutku kada se traži i očekivani broj nezadovoljenih porudžbina u posmatranom vremenskom periodu;
- Ostali troškovi – su dodatni troškovi, koji se teško razvrstavaju u jednu od gore navedenih kategorija. Tipičan primer ovih troškova su troškovi rashodovanja zastarelih zaliha, troškovi fleksibilnosti proizvodnje usled neočekivane promene tražnje i itd.

Prema (Barlow, 2003), celokupni troškovi zaliha se mogu podeliti na pet različitih vrsta troškova:

- Jedinični troškovi proizvoda (*engl. Unit Cost*) – predstavljaju cenu koštanja jednog proizvoda, koju obračunava dobavljač;
- Troškovi naručivanja zaliha (*engl. Ordering/Setup Cost*) – uključuju troškove koji nastaju u pripremi i slanju porudžbine;
- Troškovi držanja zaliha (*engl. Holding/Carrying Costs*) – predstavljaju trošak držanja pojedinačnog artikla na zalihama u jednom periodu vremena. Može biti iskazan kao procenat od jediničnog troška proizvoda. Ovaj trošak treba da pokrije kamatu na kapital koji je vezan u zalihama, trošak osiguranja, troškove

skladištenja (zakupa prostora, osvetljenja, grejanja, rashlađivanja i itd.), troškove gubitka kvaliteta robe na zalihama;

- Troškovi proboja zaliha ili nedostatka zaliha (*engl. Shortage/Stock-out Cost*) – su troškovi koji nastaju ukoliko sistem ostane bez zaliha proizvoda za prodaju ili proizvodnju. U slučaju da proizvodnja ostane bez jedne komponente proizvoda, koja se ne nalazi na zalihama, proces proizvodnje može biti ugrožen, čak i zaustavljen. Takođe, ista situacija je i u prodaji, kada kompanija može izgubiti kupce, ukoliko nema na zalihama artikla u količinama koje kupci traže, jer kupci često nisu spremni da čekaju nabavku ili proizvodnju nedostajućih artikala;
- Troškovi popusta (*engl. Discount Cost*) – nastaju kada preduzeće kupuje veće količine zaliha da bi ostvarilo niže cene kupovine. Međutim, često uštede koje se naprave u kupovini ne pokrivaju troškove držanja zaliha, koji značajno rastu zbog nabavke veće količine zaliha.

Prema (Lawrence & Pasternack, 2002) troškovi zaliha mogu se kategorizovati na sledeći način:

- Troškovi držanja zaliha (*engl. Holding/Carrying Costs*) – u preduzeću iznose od 10% do 40% prosečne vrednosti zaliha. Ovi troškovi obuhvataju: vrednost kamate na uloženi novac, troškove energije, zakupa i radne snage, troškove čuvanja zaliha i osiguranja, troškove različitih vrsta taksi i troškove opadanja kvaliteta robe na zalihama;
- Troškovi naručivanja zaliha (*engl. Ordering/Setup Cost*) – nastaju kad kompanija kupuje robu od dobavljača. U ove troškove spadaju: troškovi poštanskih i telefonskih usluga, troškovi kancelarijskog materijala, troškovi radne snage i troškovi obuke zaposlenih;
- Troškovi zadovoljstva kupca (*engl. Customer Satisfaction Cost/Stock-out Cost*) – predstavlja jednu vrstu troška nedostajućih zaliha. Osnovna razlika od osnovnog troška proboja zaliha je što se u ovaj trošak, pored troška propuštene prodaje, ponderiše se i uračunava kao trošak i stepen zadovoljstva, odnosno nezadovoljstvo kupca. Na ovaj način se pokušava kvalitativno izraziti buduća volja kupca za kupovinom artikala posmatrane kompanije;

- Troškovi nabavke i proizvodnje (*engl. Procurement/Manufacturing Cost*) – predstavljaju troškove koji pored troškova poručivanja obuhvataju i troškove koje nastaju ukoliko se nabavljaju veće količine sa popustima, koje značajno utiču na ostale troškove držanja zaliha. Ukoliko je artikal za isporuku napravljen kod drugog proizvođača, koji proizvodi za preduzeće, onda je neophodno uračunati u trošak nabavke i troškove koje ima dobavljač u pripremi proizvodnje i proizvoda za isporuku za matičnu kompaniju.

Prema (Thonemann, 2006), troškovi upravljanja zalihama obuhvataju:

- Varijabilne troškove naručivanja zaliha (*engl. Variable Order Cost*) – predstavljaju troškove nabavke zaliha, čija visina zavisi od naručene količine zaliha. Ovi troškovi uključuju jediničnu cenu koštanja proizvoda, transportne troškove, itd.;
- Fiksne troškovi naručivanja zaliha (*engl. Fixed Order Cost*) – troškovi lansiranja svake pojedinačne porudžbine. Ne zavise od količine zaliha koje se naručuju i uključuju administrativne troškove postavljanja porudžbine, materijalne troškove svake postavljanja svake pojedinačne porudžbine. U ove troškove ne spadaju troškovi razvoja informacionih sistema za prenos porudžbina;
- Troškove držanja zaliha (*engl. Inventory Holding Cost*) – koji zavise od prosečnog nivoa zaliha koje sistem poseduje. Uključuju oportunitetne troškove kapitala (zarada na uloženi kapital u zalihe van kompanije), skladišne troškove i itd.;
- Troškove nedostatka ili proboja zaliha (*engl. Shortage Cost*) – koji direktno zavise od nivoa nezadovoljene tražnje, odnosno izgubljene tražnje. Uključuju troškove dodatnih isporuka, troškove gubitka imidža kompanije (*Goodwill*) i itd.;

Prema (Ballou, 2004), postoje tri glavne klase troškova, koje su od izuzetne važnosti za definisanje strategije upravljanja zalihama:

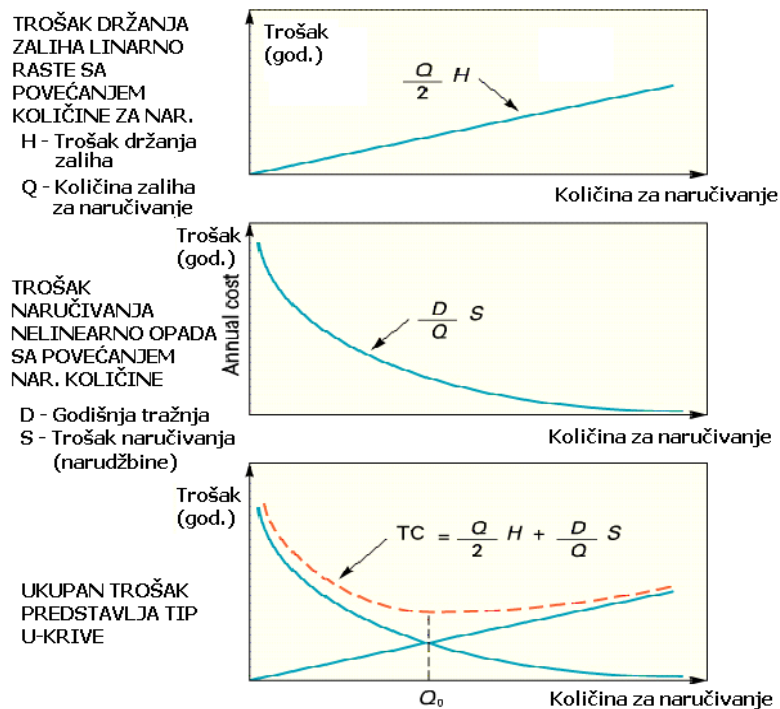
- Troškovi nabavke zaliha (*engl. Procurement Cost*) – nastaju prilikom pripreme i pravljenja porudžbine dobavljaču. Ovi troškovi mogu da uključuju jediničnu

cenu proizvoda ili jediničnu cenu proizvodnje proizvoda, trošak pripreme proizvodnog procesa, trošak obrade narudžbine u službi računovodstva ili nabavke, troškovi slanja porudžbine, troškovi transporta, materijalni troškovi obrade ili slanja porudžbine do tačke prijema porudžbine i itd. Neki od ovih troškova su fiksni, dok drugi zavise od veličine narudžbine;

- Troškovi držanja zaliha (*engl. Carrying Cost*) – su rezultat skladištenja ili držanja zaliha. Mogu se podeliti u četiri klase: troškovi prostora (zakup prostora), troškovi kapitala (troškovi kapitala vezanog u zalihama, mogu da iznose i do 80% vrednosti ukupnih troškova zaliha), troškovi ostalih usluga za držanje zaliha (troškovi osiguranja i različitih taksi), troškovi rizika u držanju zaliha (troškovi rastura i loma zaliha, troškovi gubitka kvaliteta zaliha, troškovi oštećenja robe i povraćaja robe dobavljačima);
- Troškovi nedostatka zaliha ili proboja zaliha (*engl. Stockout Cost*) - nastaju kada se postojeće porudžbine od strane kupaca ne mogu zadovoljiti iz postojećeg nivoa zaliha. Postoje dve vrste ovih troškova: troškovi izgubljene prodaje i troškovi dodatnih narudžbina od strane kupaca, kada prodaja nije izgubljena nego odložena. Ove troškove je izuzetno teško meriti.

Pored navedenih, postoje i mnoge druge podele troškova upravljanja zalihama koje se mogu naći u relevantnoj literaturi. Da bi se što bolje razumeli ukupni troškovi zaliha, neophodno je sagledati ponašanje ovih troškova u odnosu na količinu robe koja se naručuje za zalihe, odnosno u odnosu na veličinu narudžbine, kao što je prikazano na Slici 2-4.

Iz svega navedenog u ovom poglavlju, može se zaključiti da većina navedenih izvora u literaturi definiše i podrazumeva, pored ostalih troškova, postojanje dve osnovne vrste troškova: troškove naručivanja zaliha i troškove držanja zaliha. Osnovni cilj upravljanja zalihama jeste uravnoteženje raspoloživosti proizvoda ili tražnje kupaca sa troškovima, koji istovremeno ovaj proces omogućavaju. Zadovoljavanje potreba kupaca, može se ostvariti na veliki broj načina, ali treba izabrati onaj koji minimizira troškove držanja zaliha, a istovremeno maksimalno zadovoljava tražnju kupaca (Antić, 2014).



**Slika 2-4.** Grafički prikaz relevantnih troškova zaliha u zavisnosti od veličine narudžbine, (Ballou, 2004)

### 2.3. Tipovi upravljačkih modela za upravljanje zalihama

Savremeni pristupi upravljanju zalihama značajno se oslanjaju na intenzivno korišćenje informacionih sistema i pogodnih matematičkih modela. Prvi matematički model, za računanje optimalne količine naručivanja, nastao je početkom dvadesetog veka. Do značajnog razvoja ove teme došlo je šezdesetih godina dvadesetog veka u oblasti operacionih istraživanja. Interesovanje za ovu oblast ne prestaje, a uzrok tome je značaj zaliha u proizvodnji i raznovrsnim sistemima u kojima se koriste. Danas postoje nacionalna i međunarodna udruženja posvećena istraživanju zaliha, kako se navodi u (Krčevinac i drugi, 2004). Prema istom izvoru, modelovanje zaliha može biti usmereno ka razvoju računarskih metoda i algoritama koje se mogu koristiti u konkretnim sistemima zaliha, za rešavanje pojedinačnih praktičnih problema, a takođe može biti usmereno ka izučavanju teorije zaliha, u cilju izučavanja odnosa i jednačina koje opisuju sistem, da bi se dokazali rezultati u obliku optimalnih politika. Postoji veliki



broj tipova modela zaliha i različiti kriterijumi njihove klasifikacije. Faktori koji su ključni prilikom određivanja tipa i strukture modela su (Krčevinac i drugi, 2004):

1. Nadgledanje (kontrola) zaliha – neprekidno ili periodično;
2. Karakter tražnje – deterministička, slučajna ili nepoznata tražnja;
3. Stacionarnost – da li se parametri raspodele verovatnoća, koje opisuju slučajne veličine u modelu menjaju u vremenu;
4. Period planiranja – samo jedan period, konačan vremenski horizont podeljen na više perioda, beskonačan vremenski horizont;
5. Nezadovoljena tražnja – da li se nezadovoljena tražnja gubi ili nagomilava;
6. Troškovi – prosečni ili diskontovani;
7. Isporuka – trenutna ili postoji vreme isporuke, koje može biti konstantno ili slučajno;
8. Trajnost zaliha – neograničena ili ograničena, posebno ako su zalihe kvarljive;
9. Broj artikala – modeli za jedan artikal ili više artikala.

Modeli upravljanja zalihama koji se često pojavljuju u realnim situacijama razmatrani su u radu (Winston, Albright, & Broadie, 2001). Prema navedenom izvoru, svi modeli za upravljanje zalihama mogu biti podeljeni na:

- Determinističke i probabilističke modele zaliha (*engl. Deterministic and Probabilistic Models*) – Kada se govori o determinističkim modelima zaliha, pretpostavka je da su svi ulazi u model, posebno tražnja kupaca, poznati u trenutku donošenja odluka. U realnim situacijama, kompanija mora predviđati buduće potrebe kupaca, određenim modelom predviđanja tražnje. Izlazni rezultat iz ovih modela predviđanja je prosečna tražnja i standardna devijacija tražnje. Međutim, u determinističkim modelima se koriste samo prosečne vrednosti i isključuju se bilo kakve informacije o neizvesnosti, kao što su u ovom slučaju standardna devijacija. Ovakvo posmatranje determinističkih modela ima za rezultat stvaranje jednostavnijih modela, ali obično manje realnih. Probabilistički modeli za razliku od determinističkih modela koriste informacije o neizvesnosti. Ovi modeli su tipično znatno teži za analizu, ali za rezultat imaju bolje odluke, posebno kada je nivo neizvesnosti u procesu odlučivanja visok;

- Modeli sa nezavisnom (eksternom) i zavisnom (internom) tražnjom (*engl. External and Internal Demand*) – Nezavisna tražnja se javlja kada kompanija prodaje proizvode krajnjem kupcu i kada ne može direktno kontrolisati veličinu i vreme nastajanja tražnje kupaca. Tipičan primer je maloprodavac koji naručuje proizvode od dobavljača i onda čeka da vidi koliko će kupaca tražiti te proizvode u eksternoj tražnji. U ovom slučaju odluka o nabavci ne utiče na nivo tražnje kupaca. Suprotno od nezavisne tražnje javlja se zavisna tražnja ili interna tražnja pretežno u montažnim i proizvodnim procesima. Kao primer, u cilju lakšeg razumevanja, navodi se kompanije koja proizvodi frižidere. Za frižiderima, kao gotovim proizvodima postoji nezavisna (eksterna) tražnja kupaca, ali za sastavnim delovima koji ulaze u gotov proizvod – frižider postoji zavisna tražnja. Kada kompanija izvrši predviđanje tražnje za budući period (mesec dana) i definiše planove proizvodnje (za naredni mesec) u broju frižidera, koje je neophodno napraviti, pristupa se nabavci tačno određenog broja zaliha komponenti i sklopova, u tačno potrebnim količinama, za proizvodnju planiranog broja frižidera. Osnovni problem modela za zavisnom tražnjom je osigurati da sve komponente budu na pravom mestu u pravo vreme;
- Modeli za naručivanje i za proizvodnju (*engl. Ordering and Production Models*) – Modeli za upravljanje zalihama se razlikuju i prema tome da li kompanija zalihe naručuje od dobavljača ili iste proizvodi interno u sopstvenoj proizvodnji. Ako se zalihe proizvoda naručuju eksterno postoji tzv. vreme kašnjenja u naručivanju zaliha (*engl. Lead Time*), koje predstavlja vreme koje protekne od trenutka kada je porudžbina lansirana ka dobavljaču do trenutka prijema zaliha na skladište. U modelima naručivanja zaliha postoji fiksni trošak naručivanja zaliha (*engl. Setup or Ordering Cost*), koji nastaje svaki put kada je narudžbina napravljena. Ovaj trošak je nezavisan od poručene količine proizvoda i predstavlja trošak pripreme svake narudžbine. Suprotno, ako proizvode proizvodimo interno, postoji vreme potrebno za proizvodnju jedne jedinice proizvoda ili jedne serije proizvoda. Ovo vreme predstavlja stopu produktivnosti proizvodnje (*engl. Production Rate*), koja može da iznosi 10 komada na sat. Takođe, ovo vreme može da predstavlja vreme pripreme proizvodnje (*engl. Setup Time*), odnosno vreme za podešavanje opreme (mašina) da bi se proizveo

specifičan proizvod. Kao u modelima naručivanja, mogu takođe postojati troškovi pripreme proizvodnje (*engl. Setup Cost*) svaki put kada je potrebno proizvesti određenu seriju proizvoda. Ovaj trošak je nezavisan od veličine serije koja se proizvodi;

- Modeli kontinualne (neprekidne) kontrole (nadgledanja) zaliha i periodične kontrole (nadgledanja) zaliha (*engl. Continuous and Periodic Review Models*) – U modelima kontinualne kontrole zaliha, zalihe se kontrolišu (nadgledaju) neprekidno i porudžbinu je moguće napraviti u bilo kom trenutku vremena. Postoji tačka (trenutak) ponovnog naručivanja zaliha (*engl. Reorder Point*), odnosno specifičan nivo zaliha, koji zalihe vremenom dostižu i pri kome se vrši ponovno poručivanje zaliha. Aktivnost naručivanja zaliha se može desiti u bilo kom danu ili delu određenog dana tokom meseca. U suprotnom, periodična kontrola (nadgledanje) zaliha podrazumeva standardna vremena za naručivanje zaliha, kao što su jedanput mesečno ili početkom svake sedmice. U praksi, modeli neprekidne kontrole zaliha se zasigurno mogu primeniti, ukoliko kompanija poseduje kompjuterizovan pristup nadgledanju zaliha u realnom vremenu. Ovi modeli, sigurno, mogu da doprinesu snižavanju nivoa zaliha u kompaniji, jer su znatno fleksibilniji na promene u okruženju kompanije. Međutim, kada kompanija skladišti hiljade artikala na zalihama, ponekad je prigodnije koristiti periodične modele kontrole zaliha i zalihe poručivati na primer jednom nedeljno (npr. ponedeljkom);
- Modeli za jedan artikal i za više artikala (*engl. Single-Product and Multiple-Product Models*) – jedan od faktora prema kome se mogu podeliti zalihe su broj proizvoda koje model tretira u poručivanju. Modeli koji su pravljani za poručivanje ili upravljanje zalihama jednog proizvoda su konceptualno i matematički jednostavniji. U realnosti, većina kompanija istovremeno upravlja zalihama sa više različitih proizvoda, odnosno od jednog ili više dobavljača poručuje više proizvoda odjednom. Ovakav način poručivanja više artikala odjednom ili sinhronizacija više narudžbina, dovodi do smanjenja troškova poručivanja proizvoda.

Modeli za upravljanje zalihama imaju dve glavne karakteristike, koje su, prema (Barlow, 2003), usko povezane fenomenom tražnje za proizvodima, i mogu se podeliti na:

- Modeli sa nezavisnom tražnjom (deterministički i probabilistički modeli) - U determinističke modele se ubrajaju: model ekonomične količine naručivanja (*EOQ*), model sa popustom u zavisnosti od nabavljene količine zaliha (*engl. Quantity Discount Model*), model sa planiranim probojem (nedostatkom) zaliha (*engl. Planned Shortages Model*). U ovim modelima tražnja za jednom komponentom je konstantna i poznata, ali i nezavisna od tražnje za drugim komponentama. U probabilističke modele se spadaju sledeći modeli: problem prodavca novina (*engl. The Newsboy Problem*), modeli nivoa usluge (*engl. Service Level Models*) i modeli periodične kontrole (nadgledanja) zaliha (*engl. Periodic Review Systems*);
- Modeli sa zavisnom tražnjom (*engl. Material Requirements Planning - MRP*, *engl. Just-In-Time - JIT*) - Kod ovih modela tražnja za određenom komponentom je u zavisnosti od postojećih planova proizvodnje.

Modeli za upravljanje zalihama, klasifikovani prema broju vremenskih perioda, kako se navodi u (Hesse, 2005), mogu se podeliti na:

- Modeli upravljanja zalihama u jednom vremenskom periodu (*engl. Single Period Inventory System*). Jedan od najpoznatijih modela ovog tipa je model prodavca dnevnih novina. Prodavac ima zadatak da donese odluku koliko novina treba da nabavi kako bi zadovoljio tražnju. Sve novine koje ne uspe da proda istog dana, sledećeg dana neće biti aktuelne i biće ih nemoguće prodati, tako da predstavljaju trošak. Ukoliko nabavi manju količinu od realne tražnje, izgubiće mogući profit;
- Modeli upravljanja zalihama u većem broju vremenskih perioda (*engl. Multiperiod Inventory System*). Postoje dve opšte grupe u okviru sistema upravljanja zalihama u više vremenskih perioda:
  1. Modeli fiksne količine naručivanja (*engl. Fixed Order Quantity Models*) - poznatiji su kao model ekonomične količine naručivanja (*EOQ*) odnosno *Q* modeli;

2. Modeli fiksnog vremenskog perioda (*Fixed Time Period Models*) - u ovu grupu spadaju različiti sistemi i modeli, kao što su: periodični sistemi zaliha i (*engl. Periodic System*) i periodični sistemi nadgledanja zaliha (*engl. Periodic Review System*).

Prema istom izvoru, modeli za upravljanje zalihama se, prema vremenu za naručivanje zaliha, mogu podeliti na:

- Sistem fiksne količine naručivanja zaliha (*engl. Fixed Order Quantity System*) – u kome se isti iznos zaliha i u vremenu konstantan iznos zaliha, poručuje svaki put u postupku poručivanja. Bilo kakva varijacija u tražnji za zalihama se prevazilazi menjanjem vremena između porudžbina;
- Sistem fiksnog perioda ili periodične količine naručivanja zaliha (*engl. Fixed Time Quantity System*) – u ovom sistemu porudžbine se lansiraju u fiksnim intervalima vremena u zavisnosti od veličine zaliha proizvoda (sirovina) na stanju. U slučaju bilo kakve varijacije u tražnji za proizvodima (sirovinama), ista se kompenzuje menjanjem količine porudžbine. Sistem fiksnog perioda se naziva i sistem periodične količine naručivanja zaliha (*engl. Periodic Order Quantity System - POQ*).

Matematički modeli za upravljanju zalihama se prema (Anderson, Sweeney, & Williams, 2003) mogu podeliti na:

- Modeli fiksne količine poručivanja (*engl. Fixed Order Size*) – su modeli u kojima postoji promenljiv interval (period) naručivanja, dok je količina koja se poručuje fiksna. U ove modele se ubrajaju:
  1. Model ekonomske količine poručivanja
  2. Model ekonomske količine proizvodnje (*engl. Economic Production Quantity Model – EPQ model*);
  3. Model sa planiranim probojem (nedostatkom) zaliha (*engl. Inventory Model with Planned Shortages*);
  4. Model ekonomske količine poručivanja sa popustima na količinu (*engl. EOQ with Quantity Discounts*). Razlikuju se dva tipa ovih modela sa: konstantnim troškovima držanja zaliha (*engl. Constant Holding Cost*) i

promenljivim proporcionalnim troškovima držanja zaliha (*Proportional Holding Cost*);

5. Modeli sa nivoom obnavljanja zaliha (*engl. Reorder Point – ROP*) – u ove modele se ubrajaju modeli sa kašnjenjem u obezbeđenju nivoa usluge (*engl. Lead Time Service Level*) i modeli sa konstantnom ili promenjivom stopom popunjavanja zaliha (*engl. Fill Rate*);
  - Modeli sa fiksnim intervalom poručivanja (*engl. Fixed Order Interval*) - su modeli u kojima postoji promenjiva količina naručivanja, dok je vremenski interval u kome se vrši poručivanje fiksni;
  - Model jedne porudžbine (*engl. Single Order Model*) – u ovu grupu modela spada model prodavca novina (*Newsboy Model*).

Prema istom izvoru, matematički modeli za upravljanje zalihama se mogu podeliti i na determinističke i probablističke. U determinističke modele za upravljanje zalihama spadaju:

- Model ekonomske količine poručivanja (*engl. Economic Order Quantity Model – EOQ*);
- Model ekonomske količine proizvodnje (*engl. Economic Production Lot Size Model- EPQ*);
- Model sa planiranim probojem (nedostatkom) zaliha (*engl. Inventory Model with Planned Shortages*);
- Model ekonomske količine poručivanja sa popustima na količinu (*engl. Quantity Discounts for the EOQ*).

U probablističke modele za upravljanje zalihama mogu se ubrojati:

- Model upravljanja zalihama u jednom vremenskom periodu pri probablističkom tipu tražnje (*engl. Single Period Inventory Model with Probabilistic Demand*);
- Model poručivanja zaliha sa nivoom obnavljanja zaliha pri probablističkom tipu tražnje (*engl. Order Quantity – Reorder Point Model with Probabilistic Demand*);
- Model periodične kontrole (nadgledanja) zaliha pri probablističkom tipu tražnje (*engl. Periodic Review Model with Probabilistic Demand*).

Optimalno upravljanje zalihama može se vršiti i primenom koncepta upravljanja diskretnim sistemima ili diskretnim objektom, kako se navodi u (Kostić, 2009). Diskretni objekat upravljanja karakteriše zakon ponašanja, ograničenja i kriterijum performansi (ciljna funkcija). Simulacionim modelima diskretnog objekta upravljanja može se opisati dinamika zaliha. Dinamika zaliha u akumulaciji (stanju), kojim se upravlja, može se opisati pomoću određenog tipa priliva i odliva u sistemu. Ako se posmatraju materijalni tokovi zaliha u jednoj kompaniji, od trenutka priliva zaliha do trenutka isporuke proizvoda kupcu, mogu se izdvojiti četiri osnovna slučaja uticaja na dinamiku akumulacije, koji takođe određuju i tip modela za upravljanje zalihama. Prema (Kostić, 2009), mogu se navesti četiri osnovne kategorije, u koje se mogu svrstati modeli za upravljanje zalihama:

- Modeli zaliha sa diskretnim (etapnim ili periodičnim) prilivom i kontinualnim (neprekidnim) odlivom;
- Modeli zaliha sa diskretnim (etapnim ili periodičnim) prilivom i diskretnim (etapnim ili periodičnim) odlivom;
- Modeli zaliha sa kontinualnim (neprekidnim) prilivom i kontinualnim (neprekidnim) odlivom;
- Modeli zaliha sa kontinualnim (neprekidnim) prilivom i diskretnim (etapnim ili periodičnim) odlivom.

Podela modela zaliha (*engl. Lot-size Problems*), odnosno podela determinističkih modela sa dinamičkom tražnjom, predstavljena je u (Robinson, Narayanan, & Sahin, 2009):

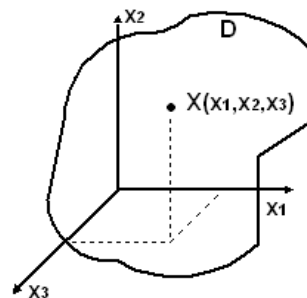
- Deterministički modeli zaliha sa statičkom tražnjom (*engl. Deterministic Static Demand Models*);
- Deterministički modeli zaliha sa dinamičkom tražnjom (*engl. Deterministic Dynamic Demand Models*).

### 3. Dinamički diskretni upravljački modeli

Definicije pojma modela se razlikuju u zavisnosti od autora koji su ih postavili. Ipak, u odnosu na većinu izvora može se reći da je model uprošćena predstava ili slika realnosti, odnosno realnog sistema. U (Kostić, 2008a) polazi od definicije sistema i elemenata sistema, da bi definisao pojam modela. Autor navodi da sistem predstavlja skup elemenata među kojima postoje odnosi i međusobna delovanja koji ih povezuju u celinu, u kome celina ima svoju svrhu ili cilj.

#### 3.1. Pojam modela

Prema autoru Kostiću, (Kostić, 2008a), stanje sistema se može definisati na sledeći način: skup vrednosti veličina, kojima se određuje ponašanje sistema, naziva se stanje sistema. Stanje sistema se može prikazati u vidu tabele, tačkom na brojnoj osi ili na neki drugi način. Prikazivanje stanja sistema tačkom na brojnoj osi odgovara prikazivanju stanja sistema koje je određeno jednom veličinom. Stanje sistema, koje se određuje većim brojem veličina može se prikazati tačkom u višedimenzionom prostoru stanja, a taj prostor se onda naziva prostor stanja sistema, kako je prikazano Slikom 3-1. Broj dimenzija prostora stanja odgovara broju nezavisnih veličina kojima se određuje stanje sistema. Stanje sistema može da uzme vrednosti iz oblasti dopuštenih stanja. Pri tome se može razlikovati diskretni i kontinualni prostor stanja sistema.



**Slika 3-1.** Prikaz stanja sistema tačkom u prostoru stanja, (Kostić, 2008a)

Do uprošćavanja polaznog sistema može se doći objedinjavanjem mnoštva njegovih stanja u jedno stanje. Taj uprošćeni sistem predstavlja uprošćeni model polaznog sistema. Uloga modela je da pomogne kako u procesu analize okruženja, u kome će se



realizovati odluka, tako i izboru najbolje alternative. Model ne zamenjuje donosioca odluke, on samo daje dodatne informacije koje pomažu pri donošenju odluke.

### 3.1.1 Tipovi modela

Klasifikacija različitih tipova modela zasniva se na načinu opisivanja uprošćenog modela, odnosno jezika kojim se model opisuje. Prema (Kostić, 2008a) mogu se definisati sledeći tipove modela u zavisnosti od načina na koji se opisuju realni sistemi:

- Verbalni model – definiše se verbalnim opisom modela. Od svih prikaza modela verbalni model je najčešće primenjivan, verovatno zato što je čoveku najbliži jezik kojim on govori. Izrada verbalnog modela se sastoji u tačnom opisu elemenata i definisanju pravila ponašanja elemenata modela rečima. Izrada verbalnog modela je prvi korak u izradi bilo kog drugog modela;
- Grafički model – grafičkim simbolima se daje grafički model. Sastoji se u slikovitom prikazu elemenata modela, njihovoj međupovezanosti u jednu celinu i načina njihovog delovanja radi ostvarenja određenog cilja. U tom smislu je i drugi korak kod izrade modela upravo izrada slikovitog prikaza modela na osnovu verbalnog modela. Grafički prikaz nekog procesa može da ima formu tabele. Za razliku od tabelarnog prikaza, modeli se mogu prikazivati i sistemskim dijagramima, blok dijagramima ili dijagramima tokova;
- Matematički model – nastaje opisujući model nekim formalnim jezikom. Struktura sistema se može lako definisati korišćenjem elemenata teorije grafova. Time se zalazi u sferu matematičkog modela, pošto se sistem opisuje formalnim jezikom matematike. Matematički model ne opisuje realne sisteme već njihove uprošćene modele. U matematičkom modelu svojstva sistema mogu biti zadata funkcionalnim zavisnostima ili graficima. Dinamičko ponašanje sistema može se prikazati nizom jednačina i nejednačina. Prelaz sistema iz jednog stanja u drugo može biti dat putem tablica, grafika i itd. Izrada matematičkog modela je treći korak u formiranju modela koji će kasnije koristiti unapred definisanim potrebama. Formiranje matematičkog modela je najteži čin stvaranja modela koji će se kasnije koristiti za unapred definisane svrhe. Matematički model spada

u klasu apstraktnih modela i može da bude komplikovan i za sagledavanje i za rešavanje;

Isti autor navodi postojanje takozvanih modela odlučivanja u okviru nauke o menadžmentu (*engl. Management Science*). U odnosu na okruženje odlučivanja, odnosno okruženje u kome se donosi odluka, modeli odlučivanja mogu biti (Kostić, 2008a):

- Deterministički i probabilistički modeli odlučivanja – Determinističko okruženje je ono u kome je neizvesnost u vezi sa događanjem događaja tako mala da se može ignorisati. Probabilističko okruženje je ono u kome je neizvesnost u vezi sa događanjem bar nekih događaja dovoljno velika da se ne može ignorisati;
- Statički i dinamički modeli odlučivanja - Ako vremenski horizont iznosi jednu godinu, on se može posmatrati kao jedan vremenski period od jedne godine, ili kao 12 vremenskih perioda od po mesec dana ili kao 365 vremenskih perioda od po jednog dana. Statičko okruženje je ono u kome postoji samo jedan vremenski period i taj vremenski period se razmatra nezavisno od budućih vremenskih perioda. Dinamičko okruženje je ono u kome postoji više od jednog vremenskog perioda i u kome se razmatraju posledice odluka iz jednog vremenskog perioda na buduće vremenske periode.

Takođe, prema (Kostić, 2008a), modeli za donošenje odluka se mogu klasifikovati kao:

- Optimizacioni modeli – bira se najbolja odluka među, onima koje se određuju modelom odlučivanja. Na primer, model koji minimizira ukupne troškove proizvodnje unutar ograničenja definisanih kapacitetom, potražnjom za proizvodom i rokovima proizvodnje može se smatrati optimizacionim modelom pošto se on koristi za izbor najboljeg plana proizvodnje;
- Simulacioni modeli – se ponekad zove "Šta - ako...?" model. On se projektuje u cilju davanja odgovora na pitanje "Šta će se desiti ako se realizuje određena odluka?". U model se unosi određena odluka i kao rezultat se dobija posledica te odluke. On ne nalazi najbolju alternativu. Međutim, analizom različitih alternativa dolazi se do korisnih informacija. Simulacioni modeli su korisni kada se ne raspolože metodama analize za kreiranje optimizacionih modela.

### 3.1.2 Simulacioni model i simulacija

Simulacija predstavlja izvođenje eksperimenata pomoću računara. Centralno mesto simulacije je model. Za predstavljanje modela koristi se računarski program koji obavlja potrebne proračune u modelu i daje rezultate u unapred definisanim formama izveštaja. Svi simulacioni modeli odgovaraju ovakvom opisu, kako se navodi u (Kostić, 2008b)

U cilju istraživanja ponašanja dinamičkih sistema koji su često veoma kompleksni i komplikovani, pristupa se izradi uprošćenih modela, a sa razvojem nauke i računara, izradi matematičkih modela. Međutim, pokazalo se da je teško analizirati kompleksno dinamičko ponašanje preko matematičkih relacija, kako se navodi u (Kostić, 2008b). Osim toga, i kada se izradi sistem relacija koje predstavljaju jedan model, mali deo matematičkih problema moguće je rešiti poznatim analitičkim metodama. Zato se pribegava numeričkim metodama, koje se razvijaju sa razvojem računara. Postupak rešavanja problema numeričkim metodama sastoji se u tome da se izabere neko početno stanje sistema koji se proučava, da se definišu zakoni - pravila promene vrednosti veličina, kojima se opisuje sistem, u jedinici vremena i da se korak po korak izračunava vrednost svake od veličina kroz vremenske periode. Na taj način se izbegavaju kompleksne matematičke analize i poteškoće koje se u vezi s njima javljaju. Postupak dobijanja vrednosti veličina, kojima se određuje ponašanje sistema, na način korak po korak, gde svaki korak predstavlja vremenski interval, naziva se simulacija. Prema (Kostić, 2008a), sistem matematičkih relacija, kao pravila izračunavanja ovih vrednosti, čini simulacioni model. Opravdanost izrade matematičkih modela i vođenje eksperimenata na njima, a ne na polaznom sistemu, nalazi se u troškovima i riziku uništenja polaznog sistema u toku eksperimenata. Polazeći od toga da je eksperiment osnovni oblik naučnog istraživanja i da je na velikim i skupim sistemima nemoguće i neopravdano izvoditi eksperimente, vođenje eksperimenata nad uprošćenim modelom preostaje kao jedina mogućnost. Vođenje eksperimenata na modelu naziva se simulacija. Putem simulacije može se proučavati svaki sistem koji se može opisati logičkim ili matematičkim izrazima. Model može da sadrži proizvoljan broj detalja. Simulacioni model može da obuhvati veći broj detalja od svih drugih logičkih ili matematičkih modela. Može se reći da je simulacija sredstvo za proučavanje dinamičkih

svojstava sistema, kako bi se dobio uvid u to kako oni rade. Prema (Kostić, 2008a), simulacija može poslužiti kao:

- pomoć u definisanju sistema;
- pomoć u utvrđivanju uzajamnih odnosa činilaca i njihovog uticaja na ponašanje;
- sredstvo da se dobije uvid u osetljivost sistema na promenu parametara;
- pomoć pri izboru konačnog rešenja između više alternativa;
- sredstvo za proveru svojstava koja se očekuju i itd.

Isti autor navodi da je za izvođenje simulacije potrebno obaviti više računskih operacija. Zato je simulacija postala praktično sredstvo za rešavanje problema tek nakon pojave računara. Prema ovom izvoru, simulacija omogućava da se:

- Na računaru u kratkom roku ispitaju sve varijante, pod uslovom da je simulacioni model postavljen, pri čemu najčešće treba promeniti samo vrednosti pojedinih parametara;
- U nekoliko minuta rada simulirati višečasovni ili čak višegodišnji rad sistema odnosno vrlo kratak period rada, koji nije moguće posmatrati pri stvarnom odvijanju procesa;
- U toku izvođenja eksperimenata zaustavi proces, kako bi se analizirali trenutni rezultati, a zatim nastavi proces, što je kod eksperimenata na realnim sistemima retko moguće;
- Ponoviti proces dobijanja rešenja bezbroj puta nad istim početnim uslovima, pri tome dodajući ili oduzimajući proizvoljan broj podataka.

Postupci izrade simulacionog modela i simulacije su prema (Kostić, 2008b) sledeći:

- Definirati i kvantifikovati značajna svojstva sistema koji treba da se simulira;
- Svojstva sistema predstaviti u obliku matematičkog modela, koji se programira na računaru;
- Testirati model, kako bi se proverila njegova tačnost rada. Tada se proverava da li su svi činioci uključeni u model i da li su zakoni ponašanja bili tačno predstavljeni;
- Simulirati efekte različitih scenarija na modelu.

U praksi je moguće razlikovati tri osnovna tipa simulacionih modela, kako se navodi u (Kostić, 2008a):

- Kontinualno promenljivi modeli - u ovom tipu modela ponašanje sistema je opisano sistemom diferencijalnih jednačina, koje omogućavaju da se promene vrednosti veličina, kojima je opisan sistem, prate kontinualno u vremenu. Zavisnost posledica od uzroka je determinističke prirode, bez stohastičkih efekata. Kontinualno promenljivi modeli su podesni za predstavljanje kretanja fizičkih sistema koji rade pod dejstvom Njutnovih ili klasičnih elektromehaničkih sila;
- Modeli fiksiranog vremena – u ovom tipu modela ponašanje sistema se može opisati sistemom diferencijalnih jednačina, ali se uzimaju vrednosti veličina, kojima je opisano ponašanje sistema, u diskretnim vremenskim tačkama koje izražavaju kraj nekog posmatranog vremenskog intervala (dan, nedelja, kvartal, polugodište, godina itd.). Promenljive su kontinualne veličine ali se njihove vrednosti očitavaju u diskretnim vremenskim tačkama. Kod takvih modela je mnogo pogodnije ponašanje sistema opisati sistemom diferencijalnih jednačina. U diferencijalne jednačine se često uključuju stohastički uslovi ili uslovi greške, kako bi se predstavila činjenica da su promenljive nešto slabije vezane u realnom sistemu. Model fiksiranog vremena najčešće se koristi u ekonometriji, gde su podaci dostupni samo na periodičnim osnovama;
- Model diskretnih događaja - u ovom tipu modela promenljive su diskretne veličine koje predstavljaju stanja elemenata sistema. Interakcija između elemenata dešava se samo u diskretnim tačkama vremena, odvojenim intervalima aktivnosti. Takve interakcije se nazivaju "događaji". Vreme se posmatra od događaja do sledećeg događaja. Stvarno određivanje vremena zavisi od stohastičkih činilaca. Model diskretnih događaja je primenljiv na široki opseg problema koji su u vezi sa sistemom pokretnih jedinica. Koristi se kod modela toka neutrona u nuklearnim reaktorima, transportnim sistemima i interakcijama borbenih jedinica na bojnopolju itd.

Ako je okruženje u kome se realizuje odluka probablističko, onda postoji više od jedne moguće posledice razmatrane odluke. U takvom slučaju simulacioni model svaki put

daje po jednu od mogućih posledica analizirane odluke, koja se zove scenario. Zato se simulacioni model primenjuje više puta nad istim polaznim podacima kako bi se istražili različiti scenariji i procenila njihova verovatnoća ostvarivanja.

### **3.1.2.1 Probabilistička i deterministička simulacija**

Simulacioni model prikazuje rezultat određenog skupa odluka. Kao ulaz se uzima skup odluka i simulira rezultat za svaku od njih. Dobijeni rezultati se analiziraju i odlučuje se da li da se isproba i drugi skup mogućih odluka. Rezultati, skupovi odluka, se proveravaju i upoređuju, dok se ne dođe do neke prihvatljive odluke (Kostić, 2008a). Prema istom izvoru, treba razlikovati determinističku simulaciju od probabilističke simulacije.

Deterministička simulacija predskazuje posledice određenog skupa odluka u determinističkom okruženju u kome će se realizovati odluka. To podrazumeva da je neizvesnost posledica dovoljno mala, pa se može ignorisati. Deterministički simulacioni model se koristi kod odlučivanja, tako što se različiti skupovi odluka vrednuju na osnovu posledica koje se predskazuju simulacijom. Bira se ona odluka koja daje najbolju vrednost odabranoj meri efikasnosti (na primer profitu).

Probabilistička simulacija se definiše kao generisanje vrednosti promenljivih modela iz neke raspodele verovatnoća, sakupljanje rezultata i njihovo korišćenje za ocenu verovatnoće i očekivane vrednosti u vezi sa određenom alternativom odluke. Kod probabilističke simulacije rezultat simulacije je neizvestan i eksplicitno se analizira uočena neizvesnost. Za svaku alternativu odluke procenjuje se očekivana vrednost odabrane mere efikasnosti. Bira se ona alternativa koja ima najveću očekivanu vrednost. Probabilistička simulacija se koristi za nalaženje najbolje odluke na isti način kao što se koristi deterministička simulacija. Alternativa odluke se simulira probabilističkom simulacijom i procenjuje se očekivana vrednost. Različite alternative odluke se upoređuju na osnovu dobijenih procena očekivane vrednosti odabrane mere efikasnosti. Izvođenje probabilističke simulacije zahteva generisanje slučajnih promenljivih iz specificirane raspodele verovatnoća. Raspodela verovatnoća se bira ili na osnovu teoretskih znanja ili na osnovu istorijskih podataka. Da bi se dobio željeni nivo

poverenja u rezultat simulacije za određenu alternativu odluke, simulaciju je potrebno ponoviti određeni broj puta.

## 3.2. Upravljanje diskretnim sistemima

### 3.2.1 Pojam objekta diskretnog upravljanja

Teoretske osnove za modelovanje optimalnog upravljanja diskretnim sistemima predstavljeno je u (Boltianski, 1978). Osnovna kategorija u ovom pristupu modeliranja je diskretno vreme  $t$ , koje može da uzima vrednosti samo u diskretnim vremenskim tačkama  $t=0,1,2,\dots,T$ , koje će predstavljati unapred definisan vremenski interval (čas, dan, kvartal, itd.). Vremenski period  $T$  je fiksiran prirodni broj. Prvu grupu izdvojenih veličina označava se kao promenljiva  $X_t^n$ , kojih će biti  $N$  ( $n=1,2,\dots,N$ ), gde je  $N$  fiksiran prirodni broj. To su veličine za koje je istraživanjem utvrđeno da su osnovni nosioci informacija o stanju objekta posmatranja u trenutku  $t$ ,  $t=1,2,\dots,T$ . Ove veličine će se nazivati promenljive stanja. Svakoj veličini  $X_t^n$  dodeliće se po jedna fazna koordinata u  $N$  - dimenzionom prostoru stanja  $E^N$ . Na taj način omogućeno je prikazivanje stanja objekta u svakom trenutku  $t$  kao tačke  $X_t$  prostora stanja čije su koordinate, kako se navodi u (Kostić, 2008a):

$$X_t = (X_t^1, X_t^2, \dots, X_t^N), \quad t = 1, 2, \dots, T \quad (3.1)$$

Druga grupa izdvojenih veličina označiće se kao promenljive  $p_t^s$ , kojih će biti  $S$  ( $s=1,2,\dots,S$ ), gde je  $S$  fiksiran prirodni broj. To su veličine za koje je istraživanjem utvrđeno da su bitni nosioci promena u objektu, ali čije su aktiviranje i dinamika izvan neposrednog uticaja upravljača proizvodnjom. Te veličine se zovu promenljive okolnosti u kojima se odvija proizvodnja ili kratko promenljive okolnosti. Svakoj veličini  $p_t^s$  dodeliće se po jedna fazna koordinata u  $S$  - dimenzionom faznom prostoru  $E^S$ . Na taj način se omogućava prikazivanje okolnosti u svakom trenutku  $t$  kao tačke  $p_t$  u  $S$ -dimenzionom faznom prostoru  $E^S$ , u obliku:

$$p_t = (p_t^1, p_t^1, \dots, p_t^S), \quad t = 1, 2, \dots, T \quad (3.2)$$

Treća grupa izdvojenih veličina označava se kao promenljiva  $u_t^r$ , kojih će biti  $R$  ( $r=1,2,\dots,R$ ), gde je  $R$  fiksiran prirodan broj. To su veličine za koje je istraživanjem utvrđeno da su osnovni nosioci promena u objektu koje aktivira upravljač svojim upravljačkim akcijama i nazivaju se promenljive upravljanja. Svakoj veličini  $u_t^r$  dodelićemo po jednu faznu koordinatu u  $R$ -dimenzionom faznom prostoru  $E^R$ . Na taj način se omogućava prikazivanje upravljanja u svakom trenutku  $t$  kao tačke  $u_t$  u faznom prostoru  $E^R$  sa koordinatama:

$$u_t = (u_t^1, u_t^2, \dots, u_t^R), \quad t = 1, 2, \dots, T \quad (3.3)$$

Istražujući veze među ovim izdvojenim veličinama utvrdićemo da je svaka promenljiva  $X_t$  složena funkcija svih promenljivih sistema, a da je funkcionalna zavisnost data kao:

$$\begin{aligned} X_0 &= \text{poznato} \\ X_t &= f(X_{t-1}, p_t, u_t), \quad t = 1, 2, \dots, T \end{aligned} \quad (3.4)$$

gde su:

- $f(X, p, u) = (f^1(X, p, u), f^2(X, p, u), \dots, f^N(X, p, u))$  – zadata vektor funkcija definisana na prostoru  $E^{N+S+R}$ , čije su vrednosti realni brojevi;
- $X_t$  i  $X_{t-1}$  – vektor stanja funkcije objekta upravljanja u trenutku  $t$ , odnosno  $t-1$ ;
- $p_t$  – vektor okolnosti objekta upravljanja u trenutku  $t$ ;
- $u_t$  – vektor upravljanja objektom trenutku  $t$ .

Relacija (3.4) predstavlja zakon ponašanja diskretnog objekta upravljanja. Očigledno je da vrednost vektora stanja  $X_t$  u trenutku  $t$  zavisi od vrednosti stanja u prethodnom trenutku  $t-1$  ( $X_{t-1}$ ), kao i vrednosti vektora okolnosti  $p_t$  i vektora upravljanja  $u_t$  u trenutku  $t$ .

Za svako  $t=1,2,\dots,T$  zadat je neki konačni podskup  $M_t$  prostora  $E^N$ , pri čemu treba da važi  $X_t \in M_t$ . Za svako  $X \in E^N$ , svako  $p \in E^S$  i svako  $t=1,2,\dots,T$ , treba odrediti u prostoru  $E^R$  neki neprazan skup  $U_t(X, p)$ , koji će predstavljati dopustivu oblast upravljanja u trenutku  $t$ . Za svako  $t$  upravljanje  $u_t$  treba da uzima vrednosti samo iz dopustive oblasti upravljanja:

$$u_t \in U_t(X_{t-1}, p_t), \quad t = 1, 2, \dots, T \quad (3.5)$$



Relacije (3.4), (3.5) i uslov  $X_t \in M_t$  određuju ponašanje objekta diskretnog upravljanja u vremenu. Drugim rečima, relacije (3.4) i (3.5) jesu simulacioni model objekta posmatranja.

### 3.2.2 Optimalno upravljanje diskretnim objektom

Saglasno zakonima po kojima se odvija dinamika okolnosti proizvodnje u vremenu (Kostić, 2008a), veličina  $p_t$  će uzimati zadate vrednosti u svakom trenutku  $t, t=1,2,\dots,T$ ,

$$p_1, p_2, \dots, p_T \tag{3.6}$$

koje će se zvati okolnosti, prikazano Slikom 3-2 u sredini.

Na osnovu upravljačkih odluka preduzimaju se upravljačke akcije kojima se menja stanje objekta ili sistema. Matematički rečeno, veličini  $u_t$  u određenim trenucima  $t=1,2,\dots,T$ , zadaju se odgovarajuće vrednosti. Tako se dobija logičan niz tačaka u prostoru promenljivih (Kostić, 2008a):

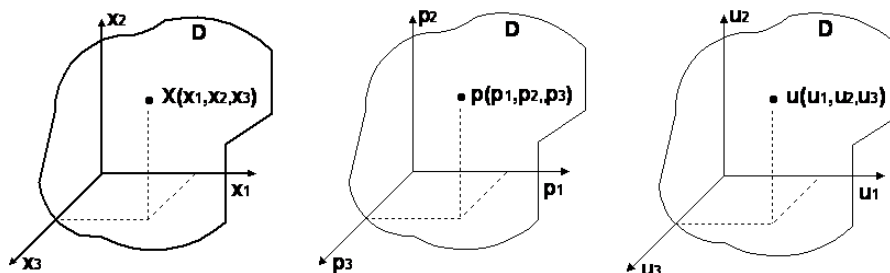
$$u_1, u_2, \dots, u_T \tag{3.7}$$

koje će se zvati upravljanje, prikazano Slikom 3-2, desno.

Kao posledica stanja objekta u prethodnom trenutku  $t-1$  i delovanja okolnosti i upravljanja u trenutku  $t (t=1,2,\dots, T)$ , stanje objekta  $X_t$  se menja. Tako se dobija niz tačaka (Kostić, 2008a):

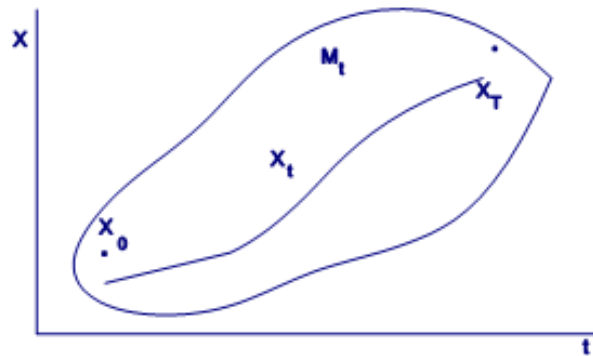
$$X_0, X_1, X_2, \dots, X_T \tag{3.8}$$

koje će se zvati trajektorija objekta upravljanja ili kratko trajektorija koja odgovara upravljanju  $u_1, u_2, \dots, u_T$ , prikazana Slikom 3-2, levo.



**Slika 3-2.** Dopustivi prostori stanja, okolnosti i upravljanja, (Kostić, 2008a)

Uređena trojka  $(p_t, u_t, X_t)$ , koju obrazuju okolnosti, upravljanje i odgovarajuća trajektorija, naziva se diskretni proces. Kod diskretnih modela objekata upravljanja bitno je unapred zadati, odnosno ustanoviti početnu vrednost promenljive  $X_t$  u trenutku  $t=0$ . Kasnije ponašanje objekta se određuje, pri pretpostavljenim okolnostima  $p_t$  i odabranom dopustivom upravljanju  $u_t$  pomoću relacije (3.4). Dobijena trajektorija  $X_0, X_1, X_2, \dots, X_T$  jeste odgovarajuća trajektorija za dato upravljanje. Međutim, ovako dobijeno upravljanje diskretnim objektom ne mora da je i optimalno za neki zadati kriterijum, (Kostić, 2008a).



**Slika 3-3.** Prevođenje diskretnog objekta upravljanja iz početnog u krajnje stanje po odgovarajućoj trajektoriji, (Kostić, 2008a)

Za utvrđivanje optimalnosti jednog procesa potrebno je utvrditi cilj, a zatim i kriterijum, kao meru postizanja datog cilja. Optimalno upravljanje će biti ono koje maksimizira ili minimizira odgovarajuću funkciju cilja. Kao kriterijum efikasnosti u opštem obliku posmatra se zbirni funkcional (Kostić, 2008a):

$$J = f^0(X_0, p_1, u_1) + f^0(X_1, p_2, u_2) + \dots + f^0(X_{T-1}, p_T, u_T) = \sum_{t=1}^T f^0(X_{t-1}, p_t, u_t) \quad (3.9)$$

gde je  $f^0$  neka zadata funkcija na prostoru  $E^{N+S+R}$ , čije su vrednosti realni brojevi.

Zadatak optimalnog upravljanja diskretnim objektom definiše se na sledeći način:

- Poznato je početno stanje  $X_0$  i zadate su okolnosti  $p_1, p_2, \dots, p_T$ ,
- U faznom prostoru  $E^N$  za svako  $t=0, 1, 2, \dots, T$  zadat je neki skup  $M_t$ .
- Treba naći takvo dopustivo upravljanje  $u_1, u_2, \dots, u_T$  za objekat

$$\begin{aligned} X_t &= f(X_{t-1}, p_t, u_t) \\ u_t &\in U_t(X_{t-1}, p_t) \end{aligned} \quad (3.10)$$

koje pri zadatim okolnostima  $p_1, p_2, \dots, p_T$ , prevodi diskretni objekat upravljanja iz zadanog početnog stanja  $X_0$  u zadati konačni skup  $M_T$ , a da za svako  $t=1, 2, \dots, T$  bude ispunjen uslov  $X_t \in M_t$  i da funkcional

$$J = \sum_{t=1}^T f^0(X_{t-1}, p_t, u_t) \quad (3.11)$$

dobije maksimalnu ili minimalnu vrednost, usklađenu sa postavljenim ciljevima upravljanja.

### 3.2.3 Osnovni elementi modela objekta diskretnog upravljanja

Vreme posmatranja odvijanja nekog poslovnog događanja može se podeliti na niz uzastopnih vremenskih perioda određene dužine (trajanja). Svakom vremenskom periodu dodeljuje se jedan prirodan broj  $t$ , tako da se celokupan vremenski period posmatra kao niz diskretnih vremenskih jedinica  $t, t=1, 2, \dots, T$  kako se navodi u (Kostić, 2008a). Sve procese u biznisu treba shvatiti kao materijalne ili nematerijalne tokove. Tokovi se sastoje iz niza faza kroz koje prolazi subjekat toka. Broj ovih faza može biti bilo koji prirodan broj  $(1, 2, \dots, n)$ . Razlikuju se kao akcije i akumulacije. Akcijama se povećava ili smanjuje količina subjekta toka u akumulaciji. Akumulacije u modelu dobijaju značenje elemenata stanja sistema i označavaju se promenljivom  $X_t$ , a grafički se predstavljaju pravougaonikom. Ove promenljive  $X_t$  predstavljaju količinu subjekta toka u akumulaciji na kraju vremenskog perioda  $t$ . Elementi stanja su zavisne promenljive u modelu. Akcijama na toku se dodeljuju regulatori protoka, koji se označavaju promenljivom  $Y_t$ , a grafički se prikazuju simbolom ventila. Ove promenljive  $Y_t$  pokazuju količinu subjekta toka koja je ušla ili izašla iz akumulacije tokom jednog vremenskog perioda  $t$ , kako je navedeno u (Kostić, 2008a).

Materijalni i nematerijalni tokovi se grafički prikazuju dvostrukom linijom koja je orijentisana ka akumulaciji (ulazni tok) ili od akumulacije (izlazni tok). Jedna akumulacija može da ima više prilivnih i više odlivnih grana. Svaka od tih grana ima svoj regulator protoka. Količina subjekta toka u akumulaciji na kraju tekućeg vremenskog perioda  $t, (t=1, 2, \dots, T)$  određuje se na osnovu količine subjekta toka u toj akumulaciji na

kraju prethodnog vremenskog perioda  $t-1$  i količina subjekta toka koje su ušle u akumulaciju ( $Y_t^I$ ) i izašle iz akumulacije ( $Y_t^O$ ) u toku tekućeg vremenskog perioda  $t$ .

$$X_t = X_{t-1} + \sum Y_t^I - \sum Y_t^O, t = 1, 2, \dots, T \quad (3.12)$$

Regulatori protoka su složene funkcije svih elemenata u modelu. Svi elementi u modelu dele se na elemente stanja, elemente promene stanja i pomoćne elemente. Elementi promene stanja su nezavisne promenljive modela. Oni se klasifikuju kao okolnosti i upravljanje. Promenljive okolnosti predstavljaju one veličine čije su vrednosti unapred zadate postavljanjem problema. Označavaju se slovom  $p$ . Promenljive upravljanja predstavljaju nepoznate u modelu čije vrednosti treba odrediti. Označavaju se slovom  $u$ . U modelu postoje i pomoćne promenljive koje se označavaju slovom  $S$  ili nekim drugim slovom. To su one promenljive u modelu koje predstavljaju neki međurezultat formiranja složene funkcije regulatora protoka  $Y$ . Svi elementi u modelu povezani su informacionim vezama (tokovima). Ove veze se grafički predstavljaju orijentisanim punim ili isprekidanim linijama. One pokazuju kako se informacija o vrednosti promenljive – davaoca informacije – uključuje u izračunavanje vrednosti promenljive – primaoca informacije, te kako se ove vrednosti povezuju u jednu složenu funkciju regulatora protoka  $Y$ . Diskretizacija vremena posmatranja kontinualnih procesa u biznisu ima za posledicu da se kontinualni procesi posmatraju kao niz diskretnih količina koje se kreću kroz faze toka. Bitna pretpostavka modela je da se jedna diskretna količina formira ravnomerno (linearno) u vremenu dužine jednog vremenskog perioda  $t$ . U fazama toka, koje su definisane kao akcije, izvršavaju se neke radnje nad subjektom toka. Izvršavanje tih radnji zahteva određeno vreme. Tako se na toku formira određeno kašnjenje koje ima za posledicu da se formiranje jedne diskretne količine započinje u jednom vremenskom periodu, a završava u narednom vremenskom periodu. Bitna pretpostavka modela je da zbir kašnjenja na toku nije veći od dužine vremenskog perioda  $t$ . Za iskazivanje uticaja kašnjenja na količinu protoka u jednom vremenskom periodu  $t$  ( $t=1, 2, \dots, T$ ) koriste se faktori kašnjenja  $K$  u relacijama za regulatore protoka  $Y$  (Kostić, 2008a).

$$Y_t = c_{t-1} \cdot K + c_t \cdot (1 - K), t = 1, 2, \dots, T \quad (3.13)$$

gde su:

- $Y_t$  - količina subjekta toka koja prođe kroz fazu akcije u tekućem vremenskom periodu  $t$ , ( $t=1,2,\dots,T$ );
- $c_{t-1}$  - diskretna količina subjekta toka čije je formiranje započeto u prethodnom vremenskom periodu  $t-1$ ;
- $c_t$  - diskretna količina subjekta toka čije je formiranje započeto u tekućem vremenskom periodu  $t$ ;
- $K$  - vrednost faktora kašnjenja.

Faktor kašnjenja  $0 \leq K \leq 1$  pokazuje koliki deo diskretne količine prođe kroz granu toka u vremenskom periodu u kome je počelo formiranje te diskretne količine. Ostatak diskretne količine  $0 \leq (1-K) \leq 1$  protekne u narednom vremenskom periodu. Relacija za izračunavanje vrednosti faktora kašnjenja zavisi od tipa protoka. Protok može da bude kontinualan ili etapni. Bitna pretpostavka za izračunavanje faktora kašnjenja  $K$  je da se diskretna količina formira ravnomerno tokom celog vremenskog perioda  $t$ . Faktor kašnjenja za kontinualni protok ima sledeći oblik:

$$K = \frac{LT}{tb} \quad (3.14)$$

Faktor kašnjenja za etapni protok, sa etapama na počecima delova vremenskog perioda, ima sledeći oblik:

$$K = \frac{1}{dbe} \operatorname{int}\left(\frac{LT}{tb} dbe\right) \quad (3.15)$$

pri čemu su:

- $LT$  - suma vremena koja prethode odskoku funkcije formiranja diskretne količine protoka;
- $tb$  - broj vremenskih jedinica u jednom vremenskom periodu  $t$ ;
- $dbe$  - dogovoreni broj etapa protoka;
- $\operatorname{int}()$  - funkcija koja određuje celobrojni deo argumenta.

Da bi se obezbedilo da izlazna količina iz akumulacije u jednom vremenskom periodu ne izazove negativnu vrednost akumulacije, uveden je uslov nenegativnosti protoka, kako je navedeno u (Kostić, 2008a):

$$\begin{aligned}
 \text{za } LT^0 \geq LT^I & \quad c_t^O \leq X_{t-1} + c_{t-1}^I \cdot K - c_{t-1}^O + c_t^I, \quad t = 1, 2, \dots, T \\
 \text{za } LT^0 < LT^I & \quad c_t^O \leq X_{t-1} + c_{t-1}^I \cdot K - c_{t-1}^O, \quad t = 1, 2, \dots, T
 \end{aligned}
 \tag{3.16}$$

gde su:

- $LT^I$  - suma vremena koja prethode odskoku funkcije formiranja diskretne količine priliva;
- $LT^O$  - suma vremena koja prethode odskoku funkcije formiranja diskretne količine odliva;
- $c_t^O$  - diskretna količina izlaza iz akumulacije  $X$ , čije je formiranje započeto u tekućem vremenskom periodu  $t$ ;
- $c_{t-1}^O$  - diskretna količina izlaza iz akumulacije  $X$ , čije je formiranje započeto u prethodnom vremenskom periodu  $t-1$ ;
- $c_t^I$  - diskretna količina ulaza u akumulaciju  $X$ , čije je formiranje započeto u tekućem vremenskom periodu  $t$ ;
- $c_{t-1}^I$  - diskretna količina ulaza u akumulaciju  $X$ , čije je formiranje započeto u prethodnom vremenskom periodu  $t-1$ ;
- $K^I$  - faktor kašnjenja za ulaznu diskretnu količinu;
- $K^O$  - faktor kašnjenja za izlaznu diskretnu količinu;
- $X_{t-1}$  - količina subjekta toka u akumulaciji na kraju prethodnog vremenskog perioda  $t-1$ .

Očigledno, ukoliko se uzme da je zbir prethodnih kašnjenja jednak nuli, dobija se da je vrednost koeficijenta kašnjenja  $K$  jednaka nuli. U tom slučaju, relacije za regulatore protoka i uslove nenegativnosti nemaju članove koji se pozivaju na diskretne količine  $c$ , čije formiranje je započeto u prethodnim vremenskom periodu. Polazeći do prethodnih razmatranja moguće je definisati različite tipove ulazno-izlaznih tokova i njihov uticaj na dinamiku akumulacije. Ovi tokovi se mogu modelirati kao objekti diskretnog upravljanja (u skladu sa opštim modelom iznetim u odeljcima 3.2.1. i 3.2.2.) i na njima se mogu definisati odgovarajući zadaci optimalnog upravljanja (Kostić, 2008a).

## 4. Modelovanje upravljačkih procesa u spredšitovima

U ovom poglavlju doktorske disertacije opisan je istorijat razvoja spredšitova, karakteristike kvaliteta analitičkih modela upravljačkih problema u spredšitovima, kao i koncept spredšit simulacije, uz navođenje mogućnosti i prednosti koje pruža ovaj koncept.

### 4.1. Istorijat razvoja spredšitova

Iako se elektronski spredšitovi prvi put spominju šezdesetih godina dvadesetog veka, ideja predstavljanja brojeva u formi mreže ili tabelarnoj formi datira još iz vremena Vavilona (Hermans, 2013). Plimpton 322, vavilonska ploča iz 1800 godine p.n.e. predstavlja Pitagorine trojke u formi veoma sličnoj spredšitu, što je prikazano Slikom 4-1. Prve dve kolone mogu se smatrati ulazima, dok treća kolona predstavlja rezultat Pitagorine teoreme (koja je bila poznata mnogo godina pre nego što je Pitagora rođen 570 godine p.n.e.).



**Slika 4-1.** Plimpton 322 predstavlja Pitagorine trojke:  
Forma spredšita iz 1800 godine p.n.e., (Hermans, 2013)

Matematičke tabele, kao što je Plimpton 322, korišćene su vekovima u matematičke svrhe, za kalkulacije, učenje, kao i u administrativne svrhe. Ipak, prikaz podataka u formi sličnoj spredšitovima postao je konvencionalan u petnaestom veku, kada je matematičar *Luca Pacioli* prvi put predstavio sistem dvojnog knjigovodstva u svojoj poznatoj knjizi „*Summa de arithmetica, geometria, proportioni et proportionalita*“. Svaki sistem se sastoji od dve strane: dugovne i potražne. Svaka transakcija se beleži dvostruko, po jednom na svakoj strani, a obe strane imaju formu sličnu spredšitu. Ovaj računovodstveni sistem, koji se i danas koristi u ne mnogo promenjenoj formi, učinio je

mrežni pregled podataka najčešće korišćenom formom za predstavljanje finansijskih podataka.

Izraz *spreadsheet* potiče iz oblasti računovodstva i odnosi se na velike listove papira sa redovima i kolonama, u kojima su se organizovali podaci o poslovnim transakcijama, kako se navodi u (Power, 2004). U njima su se „širili“ (*spread*-širiti), odnosno prikazivali podaci o troškovima, prihodima, porezima i drugi povezani podaci, na jednom listu papira (*sheet*), koje je trebalo analizirati u cilju donošenja odluke. Savremeni elektronski spredšitovi koriste se za organizaciju informacija u softverski definisanim redovima i kolonama. Podaci sa različitih izvora (lokacija) se mogu obrađivati različitim funkcijama (finansijskim, matematičkim, inženjerskim itd.) i prikazivati u formatu koji donosiocu odluke omogućava da sagleda „sliku“ celokupnog poslovanja kompanije na jednom mestu.

Prema (Kohler, 1963) *spreadsheet* predstavlja radni list koji korisniku omogućava dvodimenzionalnu analizu računovodstvenih podataka. Prvi kompijuterizovani spredšit razvio je *Richard Mattessich* 1961. godine. Njegov rad dokumentovan je u (Mattessich, 1964a) i (Mattessich, 1964b). *Daniel Bricklin*, student Harvard poslovne škole je 1978. godine prvi predstavio ideju interaktivnog digitrona. Zajedno sa *Bob Frankston*, Briklin je realizovao prvi poznati spredšit softver *VisiCalc*, Slika 4-2. Briklin i Frankston su 1979. godine osnovali kompaniju *Software Arts Corporation* (Evans, 2000). *VisiCalc* je bio prvi komercijalno uspešan spredšit program, koji se prodavao u preko 20.000 kopija mesečno do 1983 (Slater, 1989). *VisiCalc* je imao mogućnost automatskog proračuna vrednosti u ćelijama u skladu sa zavisnostima u odnosu na druge ćelije.

The screenshot shows a spreadsheet window with the following data:

	A	B	C	D	E	F	G	H
1								
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								

The formula bar at the top shows: B9 <U> @SUM<B2, B3, B4, B5, B6, B7>

**Slika 4-2.** Prikaz ekrana *VisiCalc* (1979)



Bilo je omogućeno kopiranje ćelija i opsega, korišćenje apsolutnih i relativnih adresa, kao i kreiranje formula referenciranjem ćelija. Prva verzija *VisiCalc*-a radila je na *Apple II* računaru. Interesantno je da je *VisiCalc* imao značajan uticaj na uspeh *Apple II*. Kompanije su kupovale *Apple II* računare samo da bi koristile *VisiCalc*.

Usled ograničenja *VisiCalc*-a i širenja interesovanja istraživača u ovoj oblasti, na tržištu spredšitova pojavljuje novi vodeći softverski paket se *Lotus 1-2-3*, razvijen od strane IBM korporacije 1983. *Lotus 1-2-3* učinio je korišćenje spredšitova lakšim dodavanjem rada sa grafikonima, mogućnosti štampe i boljih opcija za rad sa bazom podataka, bio je brz, koristio veću memoriju i mogao da radi sa većim spredšitovima. Prema (Phan, 2003) ključne prednosti *Lotus 1-2-3* aplikacije su: imenovanje ćelija, opsezi polja, makroi za automatizaciju ponavljajućih operacija i drugo.

*Microsoft* korporacija je 1982. godine kreirala svoj prvi spredšit program *Multiplan*. Osnovna razlika između *Multiplan*-a i sličnih programa je bila u R1C1 načinu adresiranja ćelija u odnosu na A1 načina adresiranja korišćenog u *VisiCalc*-u. A1 oblik adresiranja ćelija koristio je slova za označavanje kolona i brojeve za označavanje redova. Adrese redova i kolona mogu biti relativne i apsolutne u odnosu na nameru da li će se menjati pri kopiranju formule ili neće. R1C1 označavanje definiše redove slovom R i brojem reda, spojenim sa slovom C za oznaku kolone i brojem kolone. Stavljanje brojeva u uglaste zagrade označava trenutno aktivnu ćeliju. *Multiplan* nikada nije zaživeo na MS-DOS sistemu. Iako je *Microsoft* kompanija planirala novu verziju *Multiplan*-a za *Windows*, ona nikada nije realizovana pošto je *Lotus 1-2-3* zadržao prednost nad *Multiplan*-om. Umesto nove verzije *Multiplan*-a *Microsoft* je realizovao *Excel* za *Macintosh* računare.

Popularnost *Lotus 1-2-3* bila je neosporna sve do 1984. godine, kada je *Microsoft* korporacija predstavila *Excel*, kao deo *Microsoft Office* paketa. Izgled prozora u *Excel*-u 1.0 prikazan je slikom 4-3. *Microsoft* je zadržao R1C1 notaciju uz dodatak A1 tipa adresiranja. *Excel* spredšitovi su prvi u svojoj vrsti koji su koristili grafički interfejs sa padajućim menijima i pokazivačem, kojim se upravlja mišem. Popularnosti *MS Excel*-a doprinela je i mogućnost promene stilova slova. Kalkulacije u *Excel*-u su se izvršavale

brže, usled optimizovane rekalkulacije ćelija gde se ažuriraju samo zavisne ćelije. Obzirom da *Windows* nije bio popularan operativni sistem u tom periodu, *Microsoft* je davao besplatno kopiju *Windows*-a uz *Excel* 2.0. Ovaj spredšit paket je preuzeo je tržišno vodstvo 1988. godine i zadržao ga do danas.

The screenshot shows the Excel 1.0 interface with a menu bar (File, Edit, Formula, Format, Data, Options, Macro, Window) and a formula bar containing the formula  $=West!$B$5:$D$5+East!$B$5:$D$5$ . The main window displays a spreadsheet with the following data:

Sales Consolidation				
	A	B	C	D
1				
2		1982	1983	1984
3	Net sales	\$263,537	\$277,545	\$326,266
4	Total expenses	\$245,366	\$242,616	\$270,466

West				
	A	B	C	D
1	Sales:	1982	1983	1984
2	Model TD328-W	\$58,635	\$58,993	\$71,000
3	Model NH619-P	\$45,645	\$38,456	\$58,500
4	Model PH521-A	\$23,980	\$38,922	\$30,300
5	Total sales	\$128,260	\$136,371	\$159,800

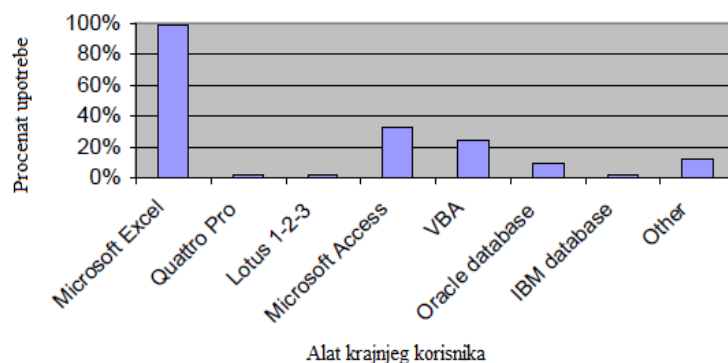
  

	A	B	C	D
1	Sales:	1982	1983	1984
2	Model TD328-W	\$65,000	\$29,766	\$34,555
3	Model NH619-P	\$56,777	\$67,400	\$74,098
4	Model PH521-A	\$13,500	\$44,008	\$57,815
5	Total sales	\$135,277	\$141,174	\$166,468

Slika 4-3. Izgled prozora u *Excel* 1.0 (1985. godine)

U današnjem poslovnom i naučnom kontekstu spredšitovi se mogu definisati kao aplikacioni paketi namenjeni tabelarnim proračunima. Oni izvršavaju jednostavne programe, ali mogu uključiti petlje, manipulirati matricama i koristiti se za rešavanje mnogobrojnih kompleksnih programerskih problema (Jackson, 1993a), (Jackson, 1993b), (Jackson & Callender, 1994). Spredšit aplikacioni programi su jeftini za korišćenje, nisu zahtevni sa aspekta specifikacija računara za rad, široko su raspoloživi ne samo profesionalnim inženjerima već i krajnjim korisnicima i izvršiocima sa nižim nivoom znanja. Sa druge strane, korišćenje u inženjerske svrhe omogućava efikasnije izvršenje zadataka i ostvarivanje višeg nivoa tačnosti (Oke, 2004), (Đorđević & Vasiljević, 2013). Spredšit sistemi ili aplikacije se učestalo koriste u svakodnevnom poslovanju, najčešće za praćenje informacija, poslovnih procesa i tokova na nivou organizacije, te su veoma popularne kod krajnjih korisnika sistema, kako za implementaciju tako i za korišćenje. Spredšit aplikacije se mogu koristiti za poslovne zadatke od matematičkog modeliranja, računanja, preko tabelarnog i grafičkog prikazivanje podataka, do analiza podataka i donošenje odluka. Numerički proračuni predstavljaju primarnu oblast za spredšit, uprkos činjenici da spredšit aplikacije imaju mogućnost vizualizacije informacija (Chi, 1999), (Abraham et al., 2005), (Hermans, 2013), zatim, istovremenog izračunavanja više indikatora, generisanja izveštaja (Yoder

& Cohn, 1994) i definisanja korisničkog interfejsa (Nardi & Miller, 1990). Osnova rada u spredšitovima se zasniva na aktivnostima rada sa sadržajem ćelija (bilo da se radi o numeričkom ili tekstualnom sadržaju), kroz procese unosa, menjanja, formatiranja, izračunavanja, sortiranja i prikazivanja podataka. Povezivanjem i unakrsnim ukrštanjem sadržaja ćelija iz različitih tabela smeštenih unutar radnih listova, formiraju se radne knjige koje predstavljaju osnovu spredšit aplikacije. Definisane kompleksnih modela i automatizacija u radu sa tabelama i podacima unutar njih, omogućava se vezom sa *Visual Basic for Application* (VBA). Spredšit predstavlja vizuelnu kompjutersku primenu matematičkog modela, gde model, koji je ugrađen u spredšit, na osnovu zadatog skupa ulaza i izračunava skup izlaza. Spredšitovi predstavljaju efikasan poslovni alat, jer omogućavaju istovremeno programiranje i modeliranje poslovnih funkcija, procesa i tokova, dok proces modeliranja često podržava i dizajniranje samih tabela i grafičko-korisničkog interfejsa. Zbog mogućnosti brze izmene modela, čak i od strane krajnjeg korisnika, spredšit predstavlja značajan alat za istraživačko modelovanje, ali i za sticanje konkurentske prednosti na organizacionom nivou.



**Slika 4-4.** Upotreba alata krajnjih korisnika, (SERP, 2006)

Spredšitovi predstavljaju najuspešnije programersku paradigmu u istoriji računara. Programeri-krajnji korisnici su značajno brojniji od programera softverskog inženjerstva. U SAD broj krajnjih korisnika procenjen na 11 miliona u odnosu na samo 2.75 miliona profesionalnih programera (Scaffidi, Shaw, & Myers, 2005). Prema (Bradley & McDaid, 2009), procenjeno je da 90% računara danas ima instaliran *MS Excel*. Razvoj od strane krajnjih korisnika (*engl. End User Development-EUD*) predstavlja aktivnost krajnjih korisnika, gde oni kreiraju aplikacije i informacione sisteme korišćenjem softvera za krajnje korisnike.

Softveri za krajnje korisnike uključuju: softvere za obradu teksta, spredšit softvere, softvere za rad sa bazama podataka i za prezentaciju. Alat krajnjih korisnika koji je najproduktivniji u svojoj primeni je spredšit, kao što se može videti u (SERP, 2006), što je prikazano na Slici 4-4.

## 4.2. Analitički modeli upravljačkih problema u spredšitovima

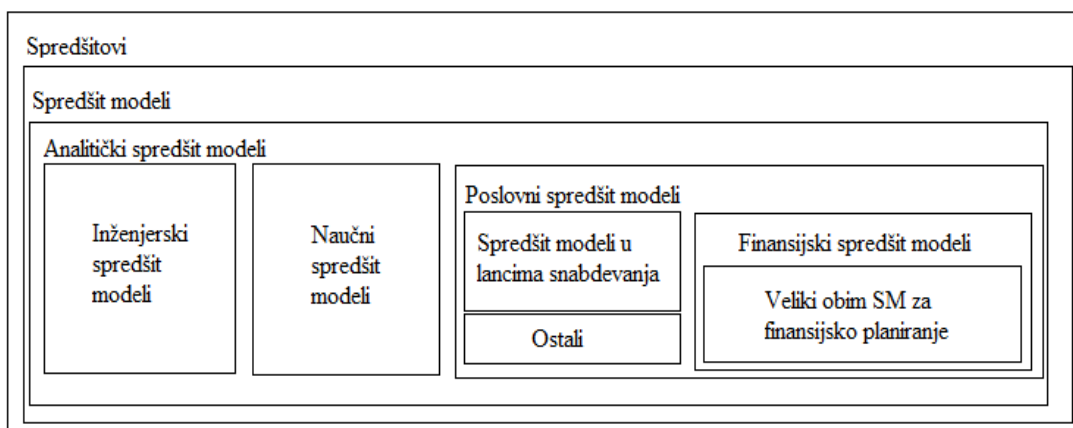
Oblast primene spredšitova je veoma raznovrsna, od izrade poslovnih modela i aplikacija, preko kreiranja obrazaca, do dekorativnog formatiranja. O širini upotrebe spredšitova svedoče brojni radovi, što je prikazano u (Jelen, 2005), (Grossman, Mehrotra, & Özlük, 2007), (Powell, Baker, & Lawson, 2009). Iz tog razloga se javljaju poteškoće prilikom postavljanja opštih smernica za rad sa spredšitovima. Specifičnost i upotrebljivost smernica se povećava proporcionalno opadanju opsega spredšitova koji se razmatraju. Bilo koja preporuka za rad sa spredšitovima je korisna samo za ograničeni skup ili domen spredšitova (Đorđević & Antić, 2014). Usled toga, neophodno je definisanje domena pri kreiranju smernica.

Domen analitičkih spredšit modela (ASM) se može definisati na osnovu iskustva u radu sa spredšit modelima zasnovanim na disciplinama nauke o menadžmentu, kao i modelima u finansijama, marketingu, lancima snabdevanja i drugim poslovnim disciplinama. Kako se navodi u (Grossman, Mehrotra, & Sander, 2011), analitički spredšit model je računarski spredšit program koji implementira matematički model u cilju analize, koja se koristi kao organizaciono sredstvo u širem poslovnom kontekstu. Analitički spredšit model poseduje sledeća svojstva:

1. Kao spredšit računarski program, analitički spredšit model je kreiran u nekom od spredšit jezika, kao što je *MS Excel*.
2. Analitički spredšit model implementira konceptualni matematički model, kojim se formuliše domen ekspertize (Powell & Baker, 2010), (Grossman & Özlük, 2001).
3. Analitički spredšit model, prema (Grossman, 2008), (Powell & Baker, 2010), (Spreadsheet Analytics, 2013), treba da omogući analizu promenom ulaznih vrednosti i posmatranjem izlaznih.

4. Kao organizaciono sredstvo, analitički spredšit model je namenjen upotrebi od strane kreatora, ali i njegovih kolega i naslednika.
5. Analize se izvršavaju u širokom poslovnom kontekstu, sa svrhom ostvarenja organizacionih ciljeva. U skladu sa tim izlazi spredšita su projektovani da zadovolje potrebe zaposlenih.

Značajno je naglasiti razliku između analitičkih spredšit modela i spredšit modela zasnovanih na podacima, koji podrazumevaju velike skupove numeričkih vrednosti. Spredšitovi koji podrazumevaju samo alate za pretragu, pregled donošenje odluka kao što su sortiranje, filtriranje i pivot tabele ne pripadaju grupi analitičkih spredšit modela. ASM primarno su zasnovani na poslovnoj logici ugrađenoj u model, a ne samo na podacima, sa velikim skupovima vrednosti i malim brojem jednostavnih formula. ASM se koriste u različitim oblastima poslovanja: upravljanje lancima snabdevanja, istraživanje tržišta, raspoređivanje, finansije, ali i u inženjerstvu. Oni se obimno razmatraju kroz obrazovanje studenata u oblastima nauke o menadžmentu, kvantitativne analize i u poslovnim školama. Primeri ovih modela mogu se naći u brojnim radovima kao što su: (Grossman, Mehrotra, & Sander, 2011), (Grossman, Mehrotra, & Özlük, 2007), (Powell & Baker, 2010), (Winston & Albright, 2008), (Ragsdale, 2010). Domen analitičkih spredšit modela, prema (Grossman, Mehrotra, & Johncharl, 2011) prikazan je Slikom 4-5.



**Slika 4-5.** Domen analitičkih spredšit modela,  
(Grossman, Mehrotra, & Johncharl, 2011)

Dimenzije kvaliteta analitičkog spredšit modela veoma zavise od domena istog. Prema (Grossman, Mehrotra, & Johncharl, 2011), na najopštijem nivou ASM visokog kvaliteta treba da poseduje sledeće karakteristike:

1. Pogodnost za efikasne analize;
2. Čitljivost (od strane korisnika);
3. Transferabilnost;
4. Tačnost;
5. Mogućnost ponovnog korišćenja (sa drugim ulaznim podacima);
6. Mogućnost modifikovanja (u skladu sa novim okolnostima).

Navedene dimenzije kvaliteta direktno su uslovljene definicijom ASM. Prilikom ocene kvaliteta ASM, bez obzira na domen istog, potrebno je uzeti u obzir preporuke dovoljno specifične sa jedne strane, ali dovoljne širine za obuhvatanje preporuka postojećih metodologija kreiranih na osnovu iskustva iz prakse, nauke o menadžmentu, modeliranja, razvoja softvera i spredšit inženjerstva (Đorđević & Antić, 2014). U skladu sa tim, svaki spredšit model treba da sadrži sledeće elemente:

1. Modularni dizajn;
2. Strukturni dizajn;
3. Informacioni tokovi;
4. Opseg polja za ulazne i izlazne podatke;
5. Deo modela za proračun;
6. Deo modela za izveštaje.

Modul predstavlja skup sličnih elemenata. Primer modula predstavlja nekoliko ćelija radnog lista, veći deo radnog lista, ceo radni list, skup radnih listova ili cela radna knjiga. Modul najvišeg nivoa je modul koji nije podmodul nekog drugog. Moduli nižeg nivoa ne sadrže podmodule. Svaki modul ima svoju jasno definisanu namenu. Modularni dizajn spredšita podrazumeva tri modula najvišeg nivoa: ulazni, za proračune i izlazni. Ulazni modul namenjen je skladištenju svih ulaznih podataka u redovima spredšita i njihovo prilagođavanje modulu za proračun. Modul za proračun koristi se za implementaciju matematičkog modela. Modul za izveštaje (izlazni modul) predstavlja rezultate u formi koja je odgovarajuća za korisnika. Strukturni dizajn može se definisati

kao smisljena organizacija ASM, sastavljenog od skupova povezanih blokova tzv. modula. Strukturni dizajn je standardna tehnika preuzeta iz softverskog inženjerstva. Njegova primena značajno utiče na čitljivost, tačnost, mogućnost ponovnog korišćenja i modifikovanja. Ovakva organizacija modela u skladu je sa logikom ulaznih i izlaznih modula i uređenih informacionih tokova. U cilju efikasne analize korišćenjem ASM, ulazne i izlazne vrednosti modela moraju biti jasno razdvojene i lako uočljive, bez “ometanja” izazvanih proračunom. Analitičar treba da može da koristi model na isti način kao sisteme za podršku odlučivanja ili softvere za *web* analitiku. Spređit mora biti organizovan tako da:

- analize ne uključuju logiku modela;
- ulazne vrednosti budu grupisane odvojeno od proračuna;
- izlazne vrednosti budu grupisane odvojeno od proračuna.

Ulazno-izlazna struktura ASM omogućava transferabilnost i olakšava analize. Informacioni tokovi modela treba da budu direktni, bez ukrštanja. Na najvišem nivou informacioni tokovi počinju od ulaznih podataka, nastavljaju se kroz deo za proračun i završavaju sa izlaznim vrednostima. Uređeni informacioni tokovi omogućavaju modifikaciju modela promenom dela za proračune, pri čemu su ulazni i izlazni deo izolovani. Ulazni modul predstavlja deo modela namenjen skladištenju ulaznih podataka. Projektovanje i dizajn ulaznog modula zahteva poznavanje domena problema koji se predstavlja modelom. Ovaj modul se najčešće sastoji iz više podmodula, kao što su:

- promenljive okolnosti (ulazne promenljive na koje organizacija nema uticaja);
- pretpostavke;
- upravljačke promenljive (promenljive koje predstavljaju upravljačku odluku organizacije);
- ulazna obrada (prevođenje prethodnih ulaznih podataka u potrebnu formu za dalje proračune).

Modul za proračune namenjen je implementaciji matematičkog modela od strane spređit inženjera. On preuzima informacije iz ulaznog modula, izvodi proračune i generiše informacije potrebne za izlazni deo ili modul izveštaja. Jedna od preporuka za

kreiranje ovog dela modela je korišćenje više podmodula za različite blokove proračuna, koji se pozivaju na radne listove. Organizacija ulaznih podataka, takođe, značajno utiče na modul proračuna. Modul izveštaja sadrži sve izlaze modela. Modul i podmoduli treba da budu formatirani u skladu sa potrebama i okruženjem korisnika.

Prilikom ocene kvaliteta ASM u obzir treba uzeti prethodno navedene preporuke. U skladu sa tim, spredšit model se analizira kroz elemente modularni dizajn, strukturni dizajn, informacioni tokovi, opseg polja za ulazne i izlazne podatke, deo modela za proračun, deo modela za izveštaje. Karakteristike kvaliteta elemenata ASM, prema (Đorđević & Antić, 2014), date su Tabelom 4-1.

**Tabela 4-1.** Karakteristike kvaliteta elemenata ASM, (Đorđević & Antić, 2014)

<b>Karakteristike kvaliteta elemenata ASM</b>	
Modularni dizajn	Tri modula najvišeg nivoa sa podmodulima po potrebi.
Strukturni dizajn	U skladu je sa logikom ulaznih i izlaznih modula.
Informacioni tokovi	Direktni, bez ukrštanja. Na najvišem nivou počinju od ulaznih podataka, nastavljaju se kroz deo za proračun i završavaju sa izlaznim vrednostima.
Ulazni/izlazni moduli	Razdvojeni po blokovima
Modul za proračun	Razdvojen po blokovima
Modul izveštaje	Razdvojen po blokovima

Koncept modeliranja različitih upravljačkih problema iz oblasti operacionog menadžmenta, kao problema upravljanja diskretnim sistemima, odnosno upravljanja diskretnim objektom, opisan u prethodnim poglavljima, izuzetno je pogodan za razvijanje u spredšitovima. Logika razvoja ovakvog modela u spredšitu je jednostavna. Svaka ćelija spredšita sadrži određeni tip podataka:

- Ulazi u model:

Ćelije mogu sadržati vrednosti parametara okolnosti. Takođe, ćelije se mogu odnositi na uzorkovanja vrednosti slučajne promenljive, koja predstavlja npr. tražnju ili cenu ili izbor iz opsega vrednosti, ali i upravljačkih promenljivih.



- Međurezultati:

Ćelije sadrže vrednosti promenljivih stanja, regulatora protoka, oblasti upravljanja i pomoćnih promenljivih, koje su potrebne za izračunavanje nekih od izlaznih vrednosti modela.

- Izlazi iz modela:

Ćelije sadrže tražene vrednosti zbog kojih je model i razvijen, tj. vrednosti ciljnog funkcionala.

Jedna od najboljih organizacija radnog lista za razvoj modela diskretnog objekta upravljanja sadrži tabele za prikaz: promenljivih okolnosti, formula za pomoćne promenljive, regulatora protoka, promenljivih stanja i ciljnog funkcionala. Neophodno je formatirati tabele za prikaz relacija oblasti upravljanja kroz sve vremenske periode  $t=1, 2, \dots, T$ . Zasebne ćelije treba nameniti upisu upravljačkih promenljivih. Kolone se koriste za prikaz vremenskih perioda horizonta posmatranja od  $t=0$  do  $t=T$ .

Opisani ASM karakteriše modularni dizajn, predstavljen kroz tri modula najvišeg nivoa: ulazni, za proračune i izlazni. Ulazni modul namenjen je skladištenju ulaznih podataka promenljivih okolnosti i ulaznih podataka promenljivih upravljanja, koje unosi korisnik. Modul za proračun koristi se za implementaciju matematičkog modela, obuhvata pomoćne promenljive, kao i oblast upravljanja. Modul za izveštaje (izlazni modul) predstavlja rezultate u formi ciljnog funkcionala kojim se lako određuje kvalitet upravljanja. Primena strukturnog dizajna utiče na čitljivost, tačnost, mogućnost ponovnog korišćenja i modifikovanja modela. Ovakva organizacija modela u skladu je sa logikom ulaznih i izlaznih modula i uređenih informacionih tokova. Ulazne i izlazne vrednosti modela su jasno razdvojene i lako uočljive, bez "ometanja" izazvanih proračunom. Model je organizovan tako da su ulazne i izlazne vrednosti grupisane odvojeno od proračuna i predstavljaju skupove povezanih blokova. Informacioni tokovi modela su direktni i bez ukrštanja. Na najvišem nivou informacioni tokovi počinju od ulaznih podataka, nastavljaju se kroz deo za proračun i završavaju sa izlaznim vrednostima. Uređeni informacioni tokovi omogućavaju modifikaciju modela promenom dela za proračune, pri čemu su ulazni i izlazni deo izolovani.

**Tabela 4-2.** Karakteristike kvaliteta ASM na opštem nivou, (Đorđević & Antić, 2014)

<b>Karakteristike kvaliteta ASM na opštem nivou</b>	
Pogodnost za efikasne analize	Pogodan za efikasne analize usled koncepcije objekta diskretnog upravljanja gde je svaka od celina: zakon ponašanja, oblast upravljanja i ciljni funkcional zasebno postavljena. Ulazne i izlazne vrednosti modela su jasno razdvojene i lako uočljive, bez “ometanja” izazvanih proračunom
Čitljivost	Na visokom nivou usled razdvojenosti ulaznih/izlaznih i modula za proračun, formatiranja ćelija i uslovnog formatiranja.
Transferabilnost	Podržana karakteristikama spredšit okruženja i primenom strukturnog dizajna
Tačnost	Obezbeđena je prevođenjem verbalnog modela u matematički, a matematičkog u spredšit model, kao i primenom uslovnog formatiranja i validacije podataka u oblasti upravljanja
Mogućnost ponovnog korišćenja	Svaka celina se može menjati u skladu sa novosaznatim činjenicama, a da se pri tome ne ugroze druge celine primenom strukturnog dizajna
Mogućnost modifikovanja	Svaka celina se može menjati u skladu sa novosaznatim činjenicama, a da se pri tome ne ugroze druge celine primenom strukturnog dizajna

U skladu sa definicijom ASM, opisani model je pogodan za efikasne analize i lako čitljiv od strane korisnika, usled koncepcije objekta diskretnog upravljanja, gde je svaka od celina: zakon ponašanja, oblast upravljanja i ciljni funkcional zasebno postavljena. Formatiranje ćelija upravljačkih promenljivih određenom bojom, jasno upućuje korisnika na deo modela koji je raspoloživ za unos podataka. Na taj način su, u cilju efikasne analize korišćenjem ASM, ulazne i izlazne vrednosti modela jasno razdvojene i lako uočljive, bez “ometanja” izazvanih proračunom. Svaka celina se može menjati u skladu sa novosaznatim činjenicama, a da se pri tome ne ugroze druge dve celine, što

podržava aspekt mogućnosti ponovnog korišćenja sa drugim ulaznim podacima i mogućnost modifikovanja u skladu sa novim okolnostima. Tačnost modela obezbeđena je prevođenjem verbalnog modela u matematički, a matematičkog u spredšit model, kao i primenom uslovnog formatiranja i validacije podataka u oblasti upravljanja. Uslovno formatiranje i validacija podataka oblasti upravljanja sprečavaju i daju jasan znak za unos vrednosti kojima bi se probila ograničenja modela. Karakteristike kvaliteta ASM na opštem nivou prikazane su Tabelom 4-2.

### **4.3. Simulacija i simulacioni model procesa upravljanja u spredšitovima**

Spredšit simulacija podrazumeva korišćenje spredšita za predstavljanje modela, izvršavanje proračuna i generisanje izlaznih izveštaja. Da bi se jedna softverska platforma koristila za simulaciju, prema (Kostić, 2008a), treba da poseduje sledeće mogućnosti:

1. Mogućnost predstavljanja matematičkih i logičkih relacija među promenljivama i dodeljivanje vrednosti promenljivama i odgovarajuće obračune, kao i algoritme koji opisuju kako se izvršavaju serije izračunavanja.
2. Način da se generišu slučajni brojevi i koriste u uzorkovanju opažanja iz različitih distribucija.
3. Način da se ponavljaju serije izračunavanja.

Najveći broj spredšitova ima ove mogućnosti, kao i više od toga (Kostić, 2008a):

1. Veliki broj funkcija za izvršenje matematičkih, statističkih, vremenskih, finansijskih i drugih izračunavanja.
2. Predstavljanje i rad sa bazama podataka, kao i pristup eksternim bazama podataka.
3. Prikaze dijagrama i grafika.
4. Prikaze fontova, boja, geometrijskih oblika u cilju unapređenja prezentacije.
5. Automatizaciju rada sa spredšitovima korišćenjem programskog jezika (*Visual Basic for Applications*).

Tabelarna struktura spredšita omogućava kreatoru da organizuje izračunavanja i rezultate na prirodan i intuitivan način. Spredšitovi su široko rasprostranjeni. Formati fajlova su standardizovani, tako da fajl koji je razvijen u jednom spredšitu, obično mogu da koriste i drugi. Rezultat toga je da kreatori i korisnici spredšitova mogu da ih razmenjuju međusobno. To je jedan od razloga zašto su spredšitovi pogodna platforma za simulaciju. Bilo koji skup izračunavanja u spredšitu može se smatrati modelom. Ti modeli uključuju parametre čije su vrednosti poznate, kao i one čije se vrednosti moraju pretpostaviti. U tom smislu se govori o determinističkim i stohastičkim modelima. Mnogi modeli iz oblasti finansija (trgovina nekretninama i osiguranje), logistike, proizvodnje i inženjeringa mogu se pogodno razviti u odgovarajući simulacioni model. Na primer, aktuari često koriste spredšit da bi procenili metode stopa osiguranja. Modeli upravljanja zalihama se koriste u cilju procene određenih politika popunjavanja zaliha.

Spredšit simulacija je veoma korisna kod primene analiza tipa „Šta će biti ako...?“. Često je kreatoru modela potrebno da utvrdi koliko je model osetljiv na variranje vrednosti određenih parametara modela. Na primer, u modelu za ocenu posledica lizinga ili kupovine nekretnine, u momentu potpisivanja ugovora, interesna stopa otplate nije poznat parametar. A sadašnja vrednost svake odluke (kupiti ili zakupiti) zavisiće od vrednosti tog parametra. Logika razvijanja simulacionog modela u spredšitu je veoma jednostavna. U opštem slučaju, svaka ćelija u spredšitu sadrži jedan od tri tipa vrednosti, (Kostić, 2008a): ulaze u model, međurezultate ili izlaze.

Sistem povezanih formula za izračunavanje vrednosti u spredšitu jeste simulacioni model, a nalaženje dopustivih vrednosti promenljivih upravljanja na način „korak po korak“, gde svaki korak predstavlja vremenski period, naziva se simulacija. Za prezentaciju rezultata simulacije mogu se koristiti tabele i grafici, koji su spredšitovima raspoloživi u različitim formama. Proces razvoja i izvršenja simulacije u spredšitu može se pojednostaviti korišćenjem nekog od raspoloživih komercijalnih *Add-In* paketa za *MS Excel*, kao što su:

- *@RISK* (<http://www.palisade.com>),
- *Crystal Ball* (<http://www.decisioneering.com>),
- *Pop-Tools* (<http://www.cse.csiro.au/poptools/>),
- *SIMTOOLS.XLA* (<http://home.uchicago.edu/>).

Ovi paketi sadrže neke mogućnosti kojih nema u osnovnim spredšitovima:

- Generisanje slučajnih brojeva korišćenjem dokumentovanih algoritama;
- Ekstenzivne funkcije za generisanje slučajnih raspodela;
- Mogućnosti da se automatizuju postavke i aktiviranja simulacionih eksperimenata;
- Mogućnosti da se automatizuju prezentacije izlaznih rezultata simulacionih eksperimenata;
- Optimizaciju procedure modela.

Za izvođenje optimizacije može se koristiti i optimizacioni alat *Solver*, kompanije *Frontline Systems Inc.* *Excel Solver* primenjuje tehnike operacionih istraživanja u cilju određivanja optimalnih rešenja, različitih tipova problema. Osim *Excel Solver*-a, koji predstavlja standardni dodatak *MS Excel*-a raspoloživi su i slični komercijalni alati za optimizaciju u spredšitovima. Kao neke od mana ovakvih optimizacioni alata mogu se navesti uopštenost primenjenih metoda za određivanja optimalnog rešenja, cena alata koji nisu standardni dodatak spredšita, ali i ograničenost sagledavanja promena stanja i ostalih elemenata modela po iteracijama.

Spredšitovi predstavljaju korisnu platformu za mnoge simulacione modele. Atraktivnost ove platforme zasniva se na raspoloživosti, intuitivnom interfejsu, lakoći korišćenja i moćnim alatima. Nažalost, izuzev situacija gde simulacija obezbeđuje korisne informacije za poslovno odlučivanje, ona se u stvarnosti koristi u malom broju slučajeva. Razlozi za nedovoljnu primenu ovog pristupa su mnogobrojni. Spredšit je moćan i pogodan alat za simulaciona modeliranja, ali ima četiri glavna ograničenja:

- moguće su samo jednostavne strukture podataka,
- teško je primeniti kompleksne algoritme,
- spredšit je sporiji nego alternativna rešenja,
- prostor za smeštanje podataka je ograničen.

Ova četiri ograničenja značajno umanjuju opseg modela koji se mogu implementirati u spredšitu. Međutim, mnogi modeli ne podležu tim restrikcijama i takvi se modeli koriste za dobijanje „brzih i prljavih“ rešenja. To je prostor gde spredšit neophodan. Prototipovi

se mogu brzo razviti i izvršiti u spreadšitu. Tek ako prototip pokaže da simulacija ne radi dobro u spreadšitu, onda ima razloga da se ona razvije u mnogo prigodnijoj platformi. Mnoge simulacije ne zahtevaju ekstenzivnost. One su projektovane da obezbede okvirne ocene i da pokažu opšte ponašanje sistema. To najviše važi sa finansijske modele. Ti modeli se najuspešnije mogu razviti u spreadšitu.

Simulacioni modeli, za koje je spreadšit korisna platforma, uključuju modele-prototipove koji su relativno mali i koriste se da se dokaže ispravnost koncepta. Komercijalni i besplatni spreadšitovi se stalno razvijaju. Buduće verzije će, bez sumnje, dopustiti veće spreadšitove i izvršavaće izračunavanja mnogo uspešnije. Za sada *MS Excel* dolazi sa alatom za optimizaciju (*Solver*).

Kao što je u radu već spomenuto, spreadšitovi su veoma pogodni za modelovanje, predstavljanje i simulaciju diskretnog objekta upravljanja. Struktura spreadšita je odgovarajuća za predstavljanje različitih tipova i vrednosti promenljivih u vremenskim periodima. Određivanje vrednosti promenljivih upravljanja izvodi se postepeno, za svaki vremenski period. Naredni period se ispituje kada su vrednosti oblasti upravljanja nenegativne za tekući period. Na taj način, sve vrednosti oblasti upravljanja iz prethodnog perioda sigurno obezbeđuju nenegativnost. Određivanje zadovoljavajućih vrednosti upravljačkih promenljivih u mnogome zavisi od definicije oblasti upravljanja, kao i drugih relevantnih formula u modelu. Sistem povezanih formula, kreiran u skladu sa elementima diskretnog objekta upravljanja, predstavlja simulacioni model. Simulacija se izvodi izborom vrednosti upravljačkih promenljivih iz jednog u naredni vremenski period, pri čemu su vrednosti svih elemenata modela u svakom periodu povezane. Implementacija koncepta diskretnog objekta upravljanja čini spreadšitove još prikladnijim alatom za simulaciju, pogodnim za razumevanje i modelovanje poslovne dinamike. Takođe, pristup je u skladu sa preporukama za dizajn i projektovanje spreadšit modela i aplikacija.

Proces izrade simulacionog spredšit modela diskretnog objekta upravljanja može se prema (Kostić, 2008b), opisati sledećim koracima:

1. Postavka problema (verbalni opis problema)
2. Analiza problema
  - a. Definisanje vremenskog horizonta i vremenskih intervala razmatranja
  - b. Identifikacija tokova u sistemu (akcije i akumulacije)
  - c. Identifikacija uticajnih faktora (okolnosti i upravljanje)
  - d. Razvijanje relacija za regulatore protoka
  - e. Ograničenja oblasti upravljanja
    - Ograničenja nenegativnosti protoka
    - Ograničenja raspoloživog kapaciteta
    - Ograničenja skladišnog prostora
    - Ostala ograničenja
  - f. Formulisanje ciljnog funkcionala
  - g. Kombinovani blok dijagrama tokova
3. Zadatak optimalnog upravljanja
  - a. Prevođenje na osnovni oblik zadatka
  - b. Definisanje zadatka optimalnog upravljanja diskretnim objektom
4. Simulacija ponašanja u spredšitu
  - a. Prevođenje matematičkih relacija u spredšit formule i funkcije
  - b. Organizacija tabela, menija, pregleda i grafika
  - c. Nalaženje jednog dopustivog rešenja
  - d. Nalaženje jednog boljeg rešenja
5. Korišćenje programa za pretraživanje oblasti upravljanja (Spredšit optimizacija)
6. Zaključci

## 5. Model upravljanja zalihama kao dinamički diskretni proces razvijen u spredšitu

Kao što je već pomenuto u početnim poglavljima ovog rada, operacioni menadžment, u užem smislu, obuhvata veoma širok dijapazon poslovnih i naučnih disciplina i oblasti. Obzirom da je osnovna tema ovog rada detekcija i analiza grešaka u dinamičkim diskretnim modelima upravljanja zalihama, u ovom poglavlju biće razmatrana kontrola zaliha, kao jedan od problema u operacionom menadžmentu, konkretno model ekonomske količine naručivanja za više proizvoda sa ograničenim prostorom skladištenja. Problemi upravljanja zalihama se mogu modelirati kao dinamički diskretni modeli upravljački procesi u spredšitu, prema (Antić, 2012). Dinamika modela ogleda se u posmatranju više vremenskih perioda, u kome se razmatraju posledice odluka iz jednog vremenskog perioda na buduće vremenske periode. Osnovna kategorija u ovom pristupu modeliranja diskretnih sistema, odnosno diskretnog objekta upravljanja, je diskretno vreme  $t$ , koje može da uzima vrednosti samo u diskretnim vremenskim tačkama  $t=0,1,2,\dots,T$ , koje će predstavljati unapred definisan vremenski interval (čas, dan, kvartal, itd.).

### 5.1. Klasični deterministički model ekonomske količine naručivanja zaliha (EOQ model)

Najpoznatiji matematički model za određivanje optimalne veličine porudžbine po najnižim troškovima je EOQ model ili model ekonomske količine naručivanja zaliha. Kada je tražnja konstantna i poznata, klasični model za određivanje ekonomske količine naručivanja zaliha (EOQ model) predstavlja osnovni model za razumevanje teorije zaliha, kao što se može videti u mnogim radovima iz oblasti operacionog menadžmenta: (Barlow, 2003), (Chase, Jacobs, & Aquilano, 2006), (Anderson, Sweeney, & Williams, 2003), (Hesse, 2005), (Wild, 2002), (Ballou, 2004), (Chopra & Meindl, 2004) i drugih. Model ekonomske količine naručivanja zaliha je poznat kao i Wilson-ov EOQ model, koji određuje optimalnu količinu za naručivanje koja minimizira ukupne troškove držanja zaliha. Kao tvorac ovog modela je *Ford W. Harris*, (Harris, 1913) i (Harris, 1915), koji je prvi napravio matematičku formulu za optimalnu



količinu poručivanja (proizvodnje). To je bila prva primena kvantitativnih metoda u oblasti upravljanja zalihama. Razvoj ovog modela početkom prošlog veka u ranim analizama naučnog pristupa teoriji zaliha, prikazao je *Wilson* (Wilson, 1934).

**Tabela 5-1.** Karakteristike klasičnog EOQ modela, (Antić, 2012)

Opšte karakteristike modela	Opis	Engleski naziv
<b>Naziv modela:</b>	Model ekonomske količine naručivanja	<i>Economic Order Quantity Model (EOQ)</i>
<b>Vrsta modela:</b>	Model zaliha	<i>Inventory Model</i>
<b>Zavisnost tražnje:</b>	Model sa nezavisnom tražnjom	<i>Independent Demand Model</i>
<b>Dinamika tražnje:</b>	Statička tražnja, konstantna	<i>Static Demand</i>
<b>Okruženje odlučivanja</b>	Deterministički model	<i>Deterministic Model</i>
<b>Vremenski horizont planiranja</b>	Beskonačan vremenski period	<i>Infinite Time Period</i>
<b>Tip modela</b>	Model fiksne količine naručivanja	<i>Fixed Order Quantity Model</i>
<b>Interval popunjavanja zaliha</b>	Fiksno	<i>Fixed</i>
<b>Tip ulaznih/izlaznih tokova</b>	Etapni priliv – Kontinualni odliv	<i>Discrete Inflow – Continuous Outflow</i>
<b>Broj jedinica proizvoda za poručivanje:</b>	Jedan proizvod (artikal)	<i>Single Order Model</i>
<b>Vreme kašnjenja:</b>	Ne postoji	<i>Lead Time = 0</i>
<b>Nedostatak (proboj) zaliha</b>	Ne postoji	<i>No Shortages</i>
<b>Troškovi zaliha:</b>	Poznati	<i>Known</i>

EOQ model je jedan od najstarijih i najpoznatijih modela za pronalaženje optimalne količine naručivanja uz minimalne troškove. EOQ model predstavlja statički model u upravljanju zalihama, jer se dinamika akumulacije subjekta toka posmatra samo za jedan period vremena. Kao rezultat izlaza iz modela za jedan vremenski period (jedna godina) dobijaju se podaci o optimalnoj količini naručivanja, o broju narudžbina tokom

godine i iznos ukupnih troškova. Osnovne opšte pretpostavke modela, prema (Antić, 2012) predstavljene su Tabelom 5-1.

Osnovni cilj EOQ modela je minimizacija ukupnih troškova. Ukupni troškovi se mogu predstaviti sledećom relacijom (Anderson, Sweeney, & Williams, 2003):

$$TC = Ch + Cs + Cu \quad (5.1)$$

gde je:

- $TC$  – Ukupan godišnji trošak držanja zaliha (*engl. Total cost*),
- $Cs$  – Ukupan godišnji trošak naručivanja jedinice proizvoda (*engl. Total Setup/Ordering Cost*),
- $Ch$  – Ukupan godišnji trošak skladištenja jedinice proizvoda (*engl. Total Holding Cost per Year*)
- $Cu$  – Ukupan godišnji trošak za nabavku jedinice proizvoda (*engl. Total Unit Cost*).

Ako se posmatra ukupan trošak skladištenja jedinice proizvoda godišnje  $Ch$ , može se reći da manje količine robe za naručivanje rezultuju nižim zalihama u skladištu, a time i nižim troškovima skladištenja zaliha.

$$Ch = \frac{Q}{2} * H \quad (5.2)$$

Iz relacije (5.3), može se videti da jedinični trošak narudžbine  $S$ , ne zavisi od količine proizvoda za naručivanje. Za poznatu godišnju tražnju za proizvodima ukupan godišnji trošak naručivanja  $Cs$ , može se smanjiti nabavkom većih količina proizvoda prilikom naručivanja.

$$Cs = \frac{D}{Q} * S \quad (5.3)$$

Iako trošak koštanja nabavljene robe nije relevantan trošak za EOQ model, on se može uključiti u osnovnu relaciju ukupnih troškova držanja zaliha.

$$Cu = D * C \quad (5.4)$$

Razvijanjem jednačine (5.1) ukupan trošak držanja zaliha se može predstaviti sledećom matematičkom relacijom:

$$TC = \frac{Q}{2} H + \frac{D}{Q} S + DC \quad (5.5)$$

gde je:

- $Q$  – količina za naručivanje (*engl. Order Quantity*),
- $S$  – jedinični trošak narudžbine (*engl. Ordering Cost per Order*),
- $D$  – godišnja tražnja za artiklima (*engl. Annual Demand per Item*),
- $H$  – jedinični trošak držanja zalihe proizvoda godišnje (*engl. Unit Holding Cost per Year*),
- $C$  – jedinična cena nabavke proizvoda (*engl. Unit Cost*).

Minimalni troškovi se dobijaju kao minimum funkcije troškova  $TC$ . Minimum ukupnih funkcije troškova  $TC$  se dobija za vrednost  $Q_0$ , koja se naziva EOQ.

$$Q_0 = \sqrt{\frac{2DS}{H}} \quad (5.6)$$

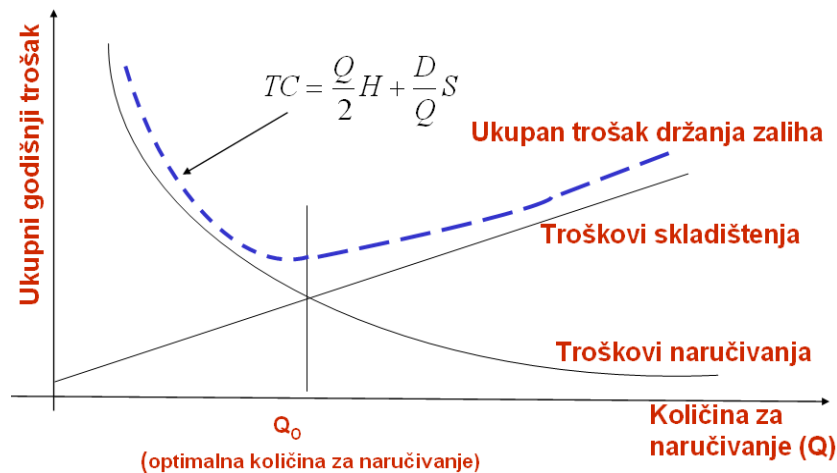
gde je:

- $Q_0$  – optimalna količina naručivanja (*engl. Optimal Order Quantity*),

Relacija za izračunavanje ukupnog troška držanja zaliha pri naručivanju optimalne količine zaliha, može se predstaviti:

$$TC = \frac{Q_0}{2} H + \frac{D}{Q_0} S + DC \quad (5.7)$$

Može se reći da kriva ukupnih troškova dostiže svoj minimum u preseku krivih ukupnih godišnji troškovi skladištenja jedinice proizvoda  $Ch$  i ukupnih godišnjih troškova naručivanja jedinice proizvoda  $Cs$ , kao što je prikazano Slikom 5-1.



**Slika 5-1.** Kriva ukupnih troškova u EOQ modelu,  
(Anderson, Sweeney, & Williams, 2003)

## 5.2. EOQ model kao dinamički diskretni upravljački model u spredšitu

Prevođenje statičkog, vremenski kontinualnog EOQ modela za upravljanje zalihama u dinamički, vremenski diskretni model za upravljanje zalihama, vrši se definisanjem osnovnih elemenata modela objekta diskretnog upravljanja (Kostić, 2009). Ako se razmatra vremenski period od godinu dana, onda se vremenski horizont može podeliti na 365 dana, što znači da je  $T=365$  dana ( $t=1,2,\dots,365$ ). Ako se model koristi za period kraći od godinu dana, onda je  $T \leq 365$ . Pošto se radi o jednom proizvodu, razmatra se jedan materijalni tok sa fazama, gde su:

- $X_t$  – količina subjekta toka u akumulaciji na kraju jednog vremenskog perioda  $t$ ,
- $Y_t^I$  ili  $(Y_t^1)$  - količinu subjekta toka koja je ušla u akumulaciju tokom jednog vremenskog perioda  $t$ .
- $Y_t^O$  ili  $(Y_t^2)$ - količinu subjekta toka koja je izašla iz akumulacije tokom jednog vremenskog perioda  $t$ .
- $u_t = N_0^I$  - broj isporuka u vremenskom horizontu  $T$ , odnosno upravljačka promenljiva  $u_t = N_0^I \forall t = 1,2,\dots,T$

Zakon ponašanja objekta diskretnog upravljanja se može predstaviti sledećim relacijama:

$$\begin{aligned} X_0 &= \text{poznato} \\ X_t &= X_{t-1} + Y_t^1 - Y_t^2, \quad t = 1, 2, \dots, T \end{aligned} \quad (5.8)$$

U skladu sa osnovnom jednačinom zakona ponašanja objekta diskretnog upravljanja, stanje akumulacije na kraju tekućeg vremenskog perioda jednako je stanju akumulacije iz prethodnog vremenskog perioda, uvećanom za sve prilive u akumulaciju u tekućem periodu i umanjenom za sve odlive iz akumulacije u tekućem vremenskom periodu. Svi troškovi su konstantni osim troška skladištenja koji raste, kako raste količina proizvoda u akumulaciji.

Pretpostavlja se da je priliv etapni sa jednom etapom na početku pojedinog vremenskog perioda  $t$ . Količina primljenih artikala u jednom vremenskom periodu može da bude nula (nema priliva) ili u-ti deo ukupne tražnje  $D$ .

$$Y_1^1 = \begin{cases} D/u_t, & X_{t-1} + 0,0001 < D/T \\ 0, & \text{u suprotnom} \end{cases}, \quad t = 1 \quad (5.9)$$

odnosno za  $t = 2, \dots, T$  može se definisati sledeća jednačina regulatora protoka:

$$Y_t^1 = \begin{cases} \min(D/u_t, D - \sum_{i=2}^T y_{i-1}^1), & X_{t-1} + 0,0001 < D/T \\ 0, & \text{u suprotnom} \end{cases}, \quad t = 2, \dots, T \quad (5.10)$$

Povećanje subjekta toka u akumulaciji je etapno, dok je odliv iz akumulacije kontinualan. Priliv i odliv u akumulaciji odvijaju se na samim počecima vremenskih perioda, pa je prethodno vreme potrebno za formiranje određene diskretne količine subjekta toka jednako nuli ( $LT=0$ ), iz čega sledi da je kašnjenje regulatora protoka jednako nuli.

$$Y_1^2 = D/T, \quad t = 1 \quad (5.11)$$

odnosno za  $t = 2, \dots, T$  može se definisati sledeća jednačinu regulatora protoka:

$$Y_t^2 = \min(D/T, D - \sum_{i=2}^T y_{i-1}^2) \quad t = 2, \dots, T \quad (5.12)$$

Kako je svakodnevna tražnja konstantna, odliv zaliha će biti  $D/T$  po danu, dok celokupna tražnja ne bude zadovoljena. Količina priliva  $D/N_0^I$  će biti raspoloživa u trenutku kada količina zaliha padne ispod količine dnevne tražnje da ne bi došlo do proboja zaliha (negativne zalihe). Artikal će se poručivati dok tražnja ne bude u potpunosti zadovoljena. Popunjavanje zaliha će se izvršiti  $N_0^I$  puta u jednakim količinama  $D/N_0^I$ .

Oblast upravljanja definiše se relacijama koje obezbeđuju nenegativnost zaliha i relacijom koja obezbeđuje nenegativnost promenjive upravljanja, kao i njenu celobrojnost.

$$\begin{aligned}
 0 &\leq u_t = N_0^I \leq T \\
 u_t &= \text{celobrojno} \\
 0 &\leq X_{t-1} + Y_t^1 - Y_t^2 \\
 t &= 1, 2, \dots, T
 \end{aligned} \tag{5.13}$$

Prvo ograničenje predstavlja prirodni uslov nenegativnosti upravljačke promenjive, jer broj isporuka može biti samo pozitivan broj, dok drugo ograničenje podrazumeva da stanje zaliha na stanju (akumulaciji) ne sme biti negativno.

Relacije 5.8 do 5.13. definišu objekat diskretnog upravljanja, odnosno simulacioni model dinamike zaliha. Zadržavajući osnovne postavke klasičnog EOQ modela, prema (Antić, 2012) moguće je definisati sledeće pretpostavke modela:

$$\begin{aligned}
 D &= \text{const}, \\
 S &= \text{const}, \\
 C &= \text{const}, \\
 H &= \text{const},
 \end{aligned} \tag{5.14}$$

Ukupni troškovi zaliha na kraju vremenskog horizonta  $T$ , mogu se prikazati ciljnim funkcionalom čiju vrednost treba minimizirati:

$$(\min) J = \sum_{t=1}^T \left( S \cdot \begin{cases} 1, & Y_t^1 > 0 \\ 0, & Y_t^1 = 0 \end{cases} \right) + H \cdot (X_{t-1} + Y_t^1 - Y_t^2 / 2) + C \cdot Y_t^1 \tag{5.15}$$

Pri čemu su:

- $T$  - Broj dana obuhvaćenih vremenskim horizontom ( $T \leq 365$ ),
- $C$  - jedinična cena nabavke artikla, koja je konstantna za svaki  $t$ ,
- $S$  - troškovi dopreme jedne etape isporuke u periodu  $t$ , ako se ostvaruje isporuka. Može biti konstanta za svaku isporuku. Ako nema isporuke, onda su troškovi dopreme jednaki nuli,
- $H$  - troškovi držanja jedinice količine artikla na zalihi u jednom danu su jednaki prosečnoj količini zaliha na skladištu, odnosno nivou  $Q/2$ .

Kako je definisan objekat diskretnog upravljanja svojim zakonom ponašanja, poznatim početnim stanjem  $X_{t=0}$ , oblašću upravljanja i ciljnim funkcionalom, zadatak optimalnog upravljanja diskretnim objektom glasi:

Za zadati vremenski period ( $T$ ) i zadatim okolnostima (navedenim u relaciji 5.14), odnosno tražnje ( $D$ ), cene nabavke robe ( $C$ ), troškova narudžbina ( $S$ ) i troškova skladištenja jedinice proizvoda ( $H$ ), potrebno je naći takvo upravljanje  $u_t$ , koje za zadato vreme ( $T$ ) prevodi diskretni objekat upravljanja iz početnog stanja  $X_0$  u neki skup krajnjih stanja  $X_T$ , a da pri tome budu zadovoljeni uslovi ograničenja i da ciljni funkcional (5.15) dobije minimalnu vrednost.

Za rešavanje ovog zadatka treba jednostavno ispitati vrednost ciljnog funkcionala za moguće vrednosti upravljačke promenljive  $u_t$ . Upravljačka promenljiva predstavlja nepoznat broj isporuka  $N_0^I$ , tako da  $u_t = N_0^I$ , za  $t = 1, 2, 3, 4, \dots, T$ ,  $T \leq 365$ . Razvijeni matematički model objekta diskretnog upravljanja je pogodan za razvijanje u spredšitu.

Na osnovu matematičkog modela objekta diskretnog upravljanja mogu se razviti formule u spredšitu, predstavljene Tabelom 5-2. Kreirani spredšit model je pogodan za izvođenje simulacije i analizu „Šta ako...“. Za različite vrednosti upravljačkih promenljivih, moguće je definisati niz različitih scenarija odlučivanja u spredšitu. Kao primer će biti uzet EOQ model prikazan u (Barlow, 2003).

**Tabela 5-2.** Formule za predstavljanje dinamičkog EOQ modela u spredšitu,  
(Antić, 2012)

Polje	Formula ili polazni podatak	Opseg kopiranja
D14:	=D12/D\$13	Nema
D17:	=F6/D\$13	Nema
D22:	=D\$12/D\$24	Nema
D27:	=M26+H27-I27	D27:D391
H27:	=IF(M26+0,0001<D\$14;D\$22;0)	Nema
H28:	=IF(M27+0,0001<D\$14;MIN(D\$22;D\$12-SUM(\$H\$27:H27));0)	H28:H391
I27:	=D\$14	Nema
I28:	=MIN(D\$14;F\$3-SUM(\$I\$27:I27))	I28:I391
M27:	=M26+H27-I27	M27:M391
Q27:	=Q26+R27+S27+T27	Q27:Q391
R27:	=IF(\$H27>0;D\$15;0)	R27:R391
S27:	=D\$17*(M26+\$H27-\$I27/2)	S27:S391
T27:	=D\$16*\$H28	T27:T391

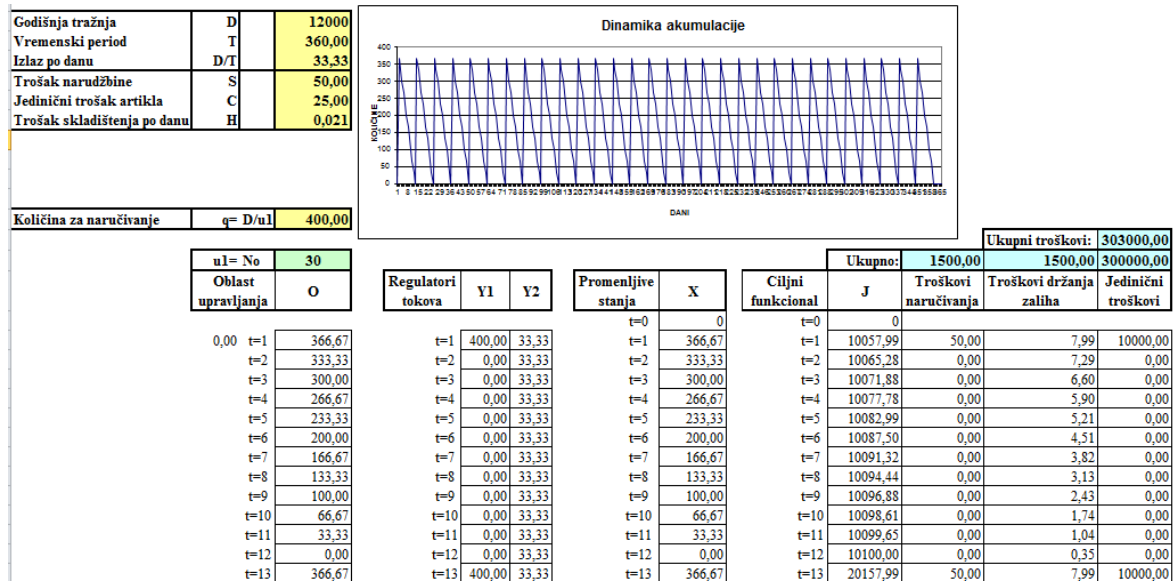
Prikazani statički EOQ model, pri definisanim ograničenjima pokazuje da pri nabavci optimalne količine proizvoda (EOQ=400 kom.), se ostvaruje minimalan ukupan godišnji trošak od  $TC=303.000,00$  n.j. sa ukupnim brojem od  $No=30$  porudžbina godišnje.

Economic Order Quantity (EOQ) Model		
<b>Input</b>	Annual Dimand D	12000
	Ordering Cost S	50
	Unit cost C	25
	Unit Holding cost per year H	7,5
<b>Output</b>	EOQ	400,00
	Total Costs TC	303.000,00
	Number of orders $N_0$	30

**Slika 5-2.** EOQ model u kompaniji BEDROCK, (Barlow, 2003)

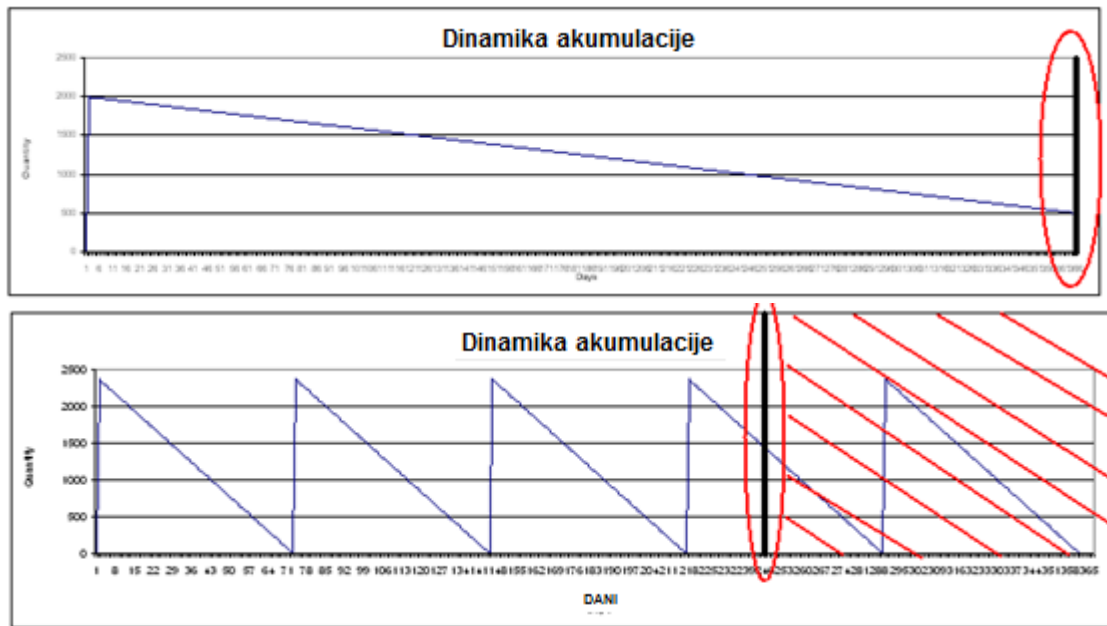


Vrednost broja isporuka se dobija iz relacije kada se ukupna tražnja podeli sa optimalnom količinom naručivanja. Postavljanjem istih početnih vrednosti u formirani spredšit model i upravljačke promenjive  $u_t=30$  isporuka, dobija se ista vrednost ciljnog funkcionala na kraju vremenskog perioda  $T$ . Opisani spredšit model prikazan je Slikom 5-3.



Slika 5-3. Diskretni EOQ model u spredšitu, (Antić, 2012)

Većina autora statičkih EOQ modela i njihovih ekstenzija posmatra zalihe u beskonačnom vremenskom horizontu, dok se izračunavanja promenljivih veličina modela rade za konačan vremenski horizont, na primer godinu dana. U cilju izračunavanja numeričkih vrednosti promenljivih veličina modela, u statičkim modelima se pravi presek beskonačnog vremenskog horizonta, a okolnosti koje se koriste u modelu uzimaju vrednost srazmerno veličini preseka beskonačnog vremenskog horizonta (Slika 5-4). Usled opisanog, dolazi do razlike između vrednosti ciljnog funkcionala dinamičkog i statičkog modela. Dinamički model podrazumeva potpuno zadovoljavanje okolnosti u određenom vremenskom periodu i tačnu vrednost ciljnog funkcionala modela, uzimajući u obzir u potpunosti troškove koji prate dinamiku priliva i odliva u definisanom vremenskom intervalu.



**Slika 5-4.** Dinamika akumulacije za izračunavanje promenljivih veličina modela, (Antić, 2012)

Na ovaj način može se pokazati da je simulacionim modelom objekta diskretnog upravljanja u spređšitu moguće na verodostojan način predstaviti i prikazati statički EOQ model. Rešenja koja daje primena upravljanja diskretnim sistemom i klasični EOQ model su identična u slučajevima kada i EOQ model i diskretni model daju celobrojnu vrednost promenjive  $u_t$ , odnosno broj isporuka je celobrojan broj, kao se navodi u (Kostić, 2009).

Kod primene klasičnog statičkog EOQ modela neophodno je razmotriti odgovore na više pitanja, koja značajno mogu uticati na ispravnost rezultata modela, koji se može koristiti u praksi. Potrebno je uzeti u obzir situaciju nastaje kada EOQ količina nije celobrojna vrednost, odnosno da li se naručuje količina koju daje EOQ formula ili neka druga količina koja daje celobrojan broj. Treba uzeti u obzir i koji broj isporuka se smatra relevantnim, ukoliko broj isporuka nije celobrojan broj, jer se vrednost broja isporuka dobija iz odnosa ukupne godišnje tražnje i EOQ količine za naručivanje (da li je moguće u posmatranom vremenskom periodu isporučiti 31,5 isporuka?). Još neki značajni aspekti su i na koji način se u statičkom EOQ modelu može pratiti trenutno stanje zaliha, u pojedinačnom periodu  $t$ , kao i na koji način se u statičkom EOQ modelu

mogu simulirati različite odluke menadžmenta pre donošenja konačne odluke o količini koja se nabavlja.

Simulacija EOQ modela, kao objekta diskretnog upravljanja u spredšitu, kako se navodi u (Antić, 2012), (Antić, 2015) omogućava:

- jednostavnije sagledavanje i razumevanje kompleksnosti procesa nabavke i upravljanja zalihama (npr. naručivanje optimalnih količina prema pakovanjima proizvođača);
- jednostavnije sagledavanje i razumevanje troškova zaliha u EOQ modelu (moguće je pratiti troškove zaliha u svakom vremenskom periodu);
- jednostavno sagledavanje visine zaliha u svakom pojedinom vremenskom periodu, odnosno dinamike akumulacije;
- pravljenje više različitih scenarija upravljanja, jer upravljačku promenljivu (koja treba da bude optimalna) ne izračunava formula, već je definisana proizvoljnom promenjivom upravljanja;
- grafički prikaz materijalnih tokova u svakom periodu vremena.

Pored klasičnog EOQ modela, postoji mnogo varijacija i ekstenzija ovog problema kao što su: model ekonomske količine naručivanja sa popustima na nabavljenu količinu i promenjivim troškovima skladištenja zaliha (DEOQ model), model ekonomske količine naručivanja sa dozvoljenim nedostatkom (probojem) zaliha, model ekonomske količine naručivanja za više proizvoda sa ograničenim prostorom skladištenja (Barlow, 2003), model zaliha sa tražnjom zavisnom od nivoa zaliha i varijabilnim troškovima držanja zaliha (Alfares, 2007), deterministički model zaliha za proizvode sa promenjivom tražnjom, deterioracijom zaliha (prema *weibull*-ovoj raspodeli) i nedostatkom zaliha, prikazan u (Wu, 2002), model zaliha sa deterioracijom zaliha i popustom na naručenu količinu, cenu i delimičnim zadovoljenjem neisporučenih narudžbina (Wee, 1999), deterministički EOQ model sa kašnjenjem u plaćanju (odloženim plaćanjem) i popustima u nabavnoj ceni, predstavljen u (Sana & Chaudhuri, 2007), deterministički EOQ model zaliha sa nedostatkom zaliha pri promenljivoj tražnji i promenjivim jediničnim troškovima nabavke, prikazan u (Teng, Chern, & Chan, 2005) i

drugi. Svi ovi modeli mogu se, na način opisan u ovom poglavlju, predstaviti kao dinamički diskretni modeli upravljanja u spredšitu.

### **5.3. Model ekonomske količine naručivanja za više proizvoda sa ograničenim prostorom skladištenja**

Model ekonomske količine naručivanja zaliha za više proizvoda sa ograničenjem skladišnog prostora (*engl. Multiple-Product EOQ Model with Storage Space Constraint*) je jedna od varijacija osnovnog modela za određivanje optimalne količine naručivanja (EOQ modela) uz minimalne troškove i ima sve karakteristike klasičnog EOQ modela. EOQ model za naručivanje više proizvoda sa ograničenjem skladišnog prostora pripada grupi modela fiksne količine poručivanja, odnosno modela sa fiksnim intervalima popunjavanja zaliha. Takođe, predstavlja statički model u upravljanju zalihama, jer se dinamika akumulacije subjekta toka posmatra samo za jedan period vremena. Kao rezultat izlaza iz modela za jedan vremenski period (godinu dana) dobijaju se podaci o optimalnoj količini naručivanja, o broju narudžbina tokom godine i iznos ukupnih troškova za jedan proizvod. Osnovni cilj EOQ modela za naručivanje više proizvoda sa ograničenjem skladišnog prostora je minimizacija ukupnih troškova.

U ovom radu biće prikazana i korišćena interpretacija statičkog EOQ modela za više artikala, sa ograničenim skladišnim prostorom kao dinamičkog diskretnog modela višestapnog upravljanja u spredšitu. Model će biti korišćen u cilju evaluacije pristupa za detekciju i analizu grešaka, zasnovanog na karakteristikama problema i načinu modeliranja.

#### **5.3.1 Dinamički diskretni EOQ model za više proizvoda sa ograničenim skladišnim prostorom**

U ovom potpoglavlju biće predstavljen dinamički diskretni EOQ model za više proizvoda sa ograničenim skladišnim prostorom. Problem se može definisati na sledeći način (Antić, 2014), (Đorđević et al., 2016).

Zadat je ukupan broj proizvoda  $m$  i za svaki proizvod  $i$ ,  $i=1,\dots,m$ , ukupna deterministička tražnja  $D_i$  koja treba da se zadovolji u okviru konačnog vremenskog horizont  $T$  na sledeći način: Ista količina  $Q_i$  proizvoda  $i$  naručuje se  $u_i$  puta, sa konstantnim vremenom  $t_i$  između dve narudžbine. Cela narudžbina  $Q_i$  dolazi na zalihe istovremeno i kada je potrebno, dok se sa zaliha troši kontinualno sa konstantnom brzinom  $D_i/T$ . Nedostaci proizvoda na zalihama (*enlg. Shortages*) nisu dozvoljeni. Za svaki proizvod  $i$  razmatraju se troškovi naručivanja  $S_i$ , troškovi nabavke po jedinici proizvoda  $C_i$  i troškovi skladištenja jedinice proizvoda u jedinici vremena  $H_i$ . Ukupni troškovi držanja zaliha u vremenskom periodu  $t_i$  se izračunavaju u skladu sa prosečnim nivoom zaliha  $Q_i/2$ .

Naručene količine različitih proizvoda dele isti skladišni prostor ukupne raspoložive površine  $G$ , koja je unapred poznata, dok je za svaki proizvod  $i$  zadat skladišni prostor  $P_i$ , koji zauzima jedinica tog proizvoda. Nivo zaliha u posmatranom vremenskom periodu  $[0,T]$  treba da zadovolji ovo ograničenje skladišnog prostora, tj. u svakom trenutku iz tog perioda ukupni prostor skladišta zauzet proizvodima ne sme da prekorači prostornu granicu  $G$ .

U skladu sa klasičnim EOQ modelom, ukupni troškovi  $TC$  za opisan problem upravljanja zalihama mogu se predstaviti jednačinom

$$TC = \sum_{i=1}^m (S_i + C_i Q_i + H_i \frac{Q_i}{2} t_i) u_i$$

Obzirom da je  $u_i = D_i / Q_i$  i  $t_i = T / u_i = T Q_i / D_i$ , ukupni troškovi  $TC$  mogu se predstaviti jednačinom

$$TC = \sum_{i=1}^m S_i \frac{D_i}{Q_i} + \sum_{i=1}^m C_i D_i + \sum_{i=1}^m H_i \frac{Q_i}{2} T \quad (5.16)$$

Problem upravljanja zalihama može se definisati kao: Odrediti količine  $Q_i$ ,  $i=1,\dots,m$ , koje zadovoljavaju ograničenje skladišnog prostora i minimiziraju ukupne troškove (5.16). U cilju približnog rešavanja ovog problema, posmatrani problem je modelovan

kao problem kombinatorne optimizacije za odgovarajući dinamički, vremenski diskretni proces upravljanja zalihama.

Umesto kontinualnog vremenskog perioda za naručivanje proizvoda, ceo vremenski period  $[0, T]$  je podeljen na  $n$  perioda  $t$  iste dužine  $T/n$ , gde je  $t = 1, \dots, n$ . Pretpostavka je da se naručivanje nekog proizvoda može realizovati samo na početku perioda  $t$ .

U vremenskom periodu  $T/n$  proizvodi se troše sa zaliha kontinualno, sa konstantnom brzinom  $D_i/T$ , a tražnja za proizvodima koja treba da bude zadovoljena u toku perioda je  $D_i/n$ . Umesto  $Q_i, i = 1, \dots, m$ , razmatramo  $u_i \in \{1, 2, \dots, n\}, i = 1, 2, \dots, m$ , kao promenljive upravljanja, dok je  $Q_i = D_i/u_i$ .

Za svaki proizvod  $i$  promena nivoa zaliha tokom celog vremenskog perioda može se predstaviti kao diskretni upravljački proces sa sledećim elementima:

- $X_t^i, t = 1, 2, \dots, n$  - ukupna količina proizvoda  $i$  koja ostaje na zalihama na kraju vremenskog perioda  $t$ ;
- $Y_t^i, t = 1, 2, \dots, n$  - količina proizvoda  $i$  naručena na početku perioda  $t$ .

Ako  $X_t^i$  predstavlja stanje procesa u vremenskom periodu  $t$  onda jednačina stanja koja opisuje ponašanje procesa može da se definiše na sledeći način:

$$X_0^i = 0 \quad (5.17)$$

$$X_t^i = X_{t-1}^i + Y_t^i - D_i/n, \quad t = 1, 2, \dots, n \quad (5.18)$$

Naručivanje količine  $Q_i = D_i/u_i$  se realizuje na početku perioda  $t$  samo u slučaju kada količina proizvoda  $i$  na zalihama, koja je preostala na kraju prethodnog vremenskog perioda  $t-1$ , nije dovoljna da bi se zadovoljila tražnja  $D_i/n$  u periodu  $t$ . Prema tome, vrednost  $Y_t^i$  zavisi od  $u_i$  i može se formalno predstaviti:

$$Y_t^i = \left\{ \begin{array}{ll} D_i/u_i, & X_{t-1}^i < D_i/n \\ 0, & \text{u suprotnom} \end{array} \right\}, t = 1, 2, \dots, n \quad (5.19)$$

Kako je  $X_0^i = 0$  onda, posledično  $X_n^i = 0$ .

Proces opisan izrazima (5.17) do (5.19) ne predstavlja tipični proces diskretnog upravljanja, gde u svakom vremenskom periodu stanje procesa zavisi od prethodnog stanja i izabranih vrednosti promenljivih upravljanja. Primenom izraza (5.17) do (5.19) za fiksiranu vrednost  $u_i$  količine  $X_t^i$  i  $Y_t^i$  mogu se izračunati za svako  $t$ ,  $t=1,2,\dots,n$ . Dakle  $Y_t^i$  nije promenljiva upravljanja, a jedina promenljiva upravljanja procesa je  $u_i$ .

Ograničenja skladišnog prostora se razmatraju samo na počecima vremenskih perioda  $t$  i mogu se formalno definisati:

$$\sum_{i=1}^m (X_{t-1}^i + Y_t^i) P_i \leq G \quad t = 1, 2, \dots, n \quad (5.20)$$

Ukupni troškovi  $J(u_1, u_2, \dots, u_m)$  za sistem upravljanja zalihama opisan sa (5.17) do (5.19) predstavljaju sumu ukupnih troškova za svaki proizvod  $i$  u svakom periodu  $t$ , pri čemu se ukupni troškovi držanja zaliha proizvoda  $i$  u periodu  $t$  računaju u skladu sa prosečnim nivoom zaliha u vremenskom periodu dužine  $T/n$ . Ukupni troškovi  $J$  predstavljeni su relacijom (5.21).

$$J(u_1, u_2, \dots, u_m) = \sum_{i=1}^m J_i(u_i)$$

$$J_i(u_i) = \sum_{t=1}^n \left( (S_i + C_i \cdot Y_t^i) \cdot \delta_t^i + H_i \cdot \left( X_{t-1}^i + Y_t^i - \frac{D_i}{2n} \right) \cdot \frac{T}{n} \right)$$

$$\delta_t^i = \begin{cases} 1, & Y_t^i > 0 \\ 0, & Y_t^i = 0 \end{cases} \quad (5.21)$$

$$J(u_1, u_2, \dots, u_m) = \sum_{i=1}^m S_i u_i + \sum_{i=1}^m \frac{H_i T}{n} \sum_{t=0}^{n-1} X_t^i + \sum_{i=1}^m \left( C_i D_i + H_i D_i \frac{T}{2n} \right)$$

uključujući da je  $\sum_{t=1}^n Y_t^i = D_i$ .

Sada se problem kombinatorne optimizacije za dinamički, vremenski diskretni proces upravljanja zalihama, definisan preko (5.17) do (5.19), može definisati na sledeći način: Za promenljive upravljanja  $u_1, u_2, \dots, u_m$  procesa (5.17) do (5.19) treba naći vrednosti iz skupa  $1, 2, \dots, n$  koje zadovoljavaju ograničenja prostora (5.20) i minimiziraju ukupne troškove (5.21).

### 5.3.2 Dinamički diskretni EOQ model za više proizvoda sa ograničenim skladišnim prostorom razvijen u spredšitu

Matematički model objekta diskretnog upravljanja, predstavljen u prethodnom potpoglavlju, pogodan je za razvoj modela u spredšitu. Na osnovu matematičkog modela objekta diskretnog upravljanja i njegovog ciljnog funkcionala kao primer je razvijen spredšit model problema za tri proizvoda, prema (Barlow, 2003), Slika 5-5. Osnovna jednačina zakona ponašanja objekta diskretnog upravljanja (5.17) i (5.18) pokazuje da je stanje akumulacije na kraju tekućeg vremenskog perioda jednako stanju akumulacije iz prethodnog vremenskog perioda, uvećanom za sve prilive u akumulaciju u tekućem periodu i umanjenom za sve odlive iz akumulacije u tekućem vremenskom periodu. Pretpostavka je da je priliv etapni sa jednom etapom na početku pojedinog vremenskog perioda  $t$ .

Količina primljenih artikala u skladištu jednom vremenskom periodu je nula ukoliko nema priliva ili  $u-t$  deo ukupne tražnje  $D$ . Ako razmatramo vremenski period od godinu dana, onda vremenski horizont možemo da podelimo na 365 dana, što znači da je  $T=365$  dana ( $t=1,2,\dots,365$ )<sup>2</sup>.

U Tabeli 5-3 prikazane su formule spredšit modela dinamičkog diskretnog EOQ problema za tri proizvoda sa ograničenim skladišnim prostorom, kao i oblast ćelija na koju se formule kopiraju.

Kreirani spredšit model je pogodan za izvođenje simulacije i analizu "Šta - ako...?". Za različite vrednosti upravljačke promenjive, moguće je definisati niz različitih scenarija odlučivanja u spredšitu.

U daljem radu, kao primer, će biti korišćen EOQ model prema (Barlow, 2003, str. 259). Poređenjem vrednosti ciljnih funkcija, za iste ulazne podatke, može se zaključiti da su troškovi statičkog i dinamičkog diskretnog spredšit modela približno isti. Kao izlaz statičkog modela dobijaju se niži troškovi (Antić, 2012), .

<sup>2</sup> Ako se model koristi za period kraći od godinu dana, onda je  $T \leq 365$ ,



**Tabela 5-3.** Spredšit formule dinamičkog diskretnog EOQ modela za više proizvoda sa ograničenim skladišnim prostorom, (Antić, 2012)

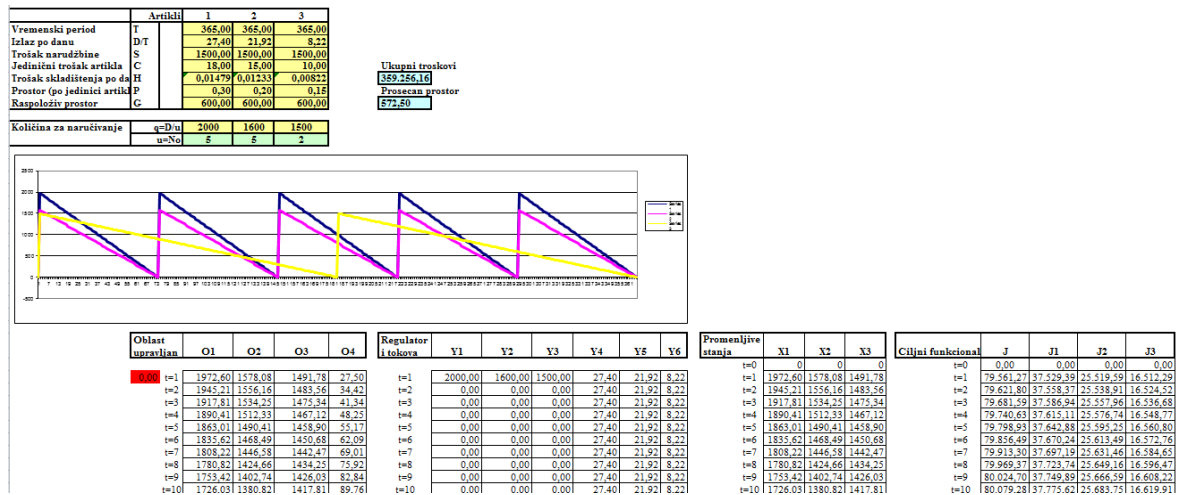
Polje	Formula ili polazni podatak	Opseg kopiranja
D15:	=F3/D14	D15:F15
D16:	=F4	D16:F16
D17:	=F5	D17:F17
D18:	=F6/D14	D18:F18
D19:	=F7	D19:F19
D23:	=F3/D24	D23:F23
D42:	=S41+J42-M42	D42:F406
G42:	=D\$20-((S41+J42)*D\$19/2+(T41+K42)*E\$19/2+(U41+L42)*F\$19/2)	G42:G406
J42:	=IF(S41+0,0001<D\$15;D\$23;0)	J42:L42
J43:	=IF(S42+0,0001<D\$15;MIN(D\$23;F\$3-SUM(J\$42:J42));0)	J43:L406
M42:	=D\$15	M42:O42
M43:	=MIN(D\$15;F\$3-SUM(M\$42:M42))	M43:O406
S42:	=S41+J42-M42	S42:U406
Y42:	=SUM(Z42:AB42)	J42:J406
Z42:	=Z41+IF(J42>0;D\$16;0)+D\$18*(S41+J42-M42/2)+D\$17*J42	Z42:AB406
J10:	=SUM(F10:H10)	Nema
K10:	=Y406	Nema
J12:	=SUMPRODUCT(F7:H7;F9:H9)/2	Nema
K12:	=SUMPRODUCT(F7:H7;D23:F23)/2	Nema

Zanemarljiva razlika u troškovima statičkog i dinamičkog modela je posledica (Antić, 2015):

- Razlike u posmatranju dužine vremenskog perioda u kome se računaju promenjive modela, a time i troškova u dinamičkom modelu u odnosu na statički model;
- Broj porudžbina ne može biti decimalan broj, već samo ceo broj što dovodi do pojave razlike u izračunavanju ukupnih troškova;

- Optimalna rešenja statičkog modela ne zadovoljavaju u potpunosti uslov zauzetosti skladišta, dok rešenja diskretnog dinamičkog simulacionog modela u potpunosti zadovoljavaju postavljene uslove raspoloživosti prostora.

Na ovaj način dokazuje se da je simulacionim modelom objekta diskretnog upravljanja u spredu moguće na verodostojan način predstaviti i prikazati statički EOQ model za više proizvoda sa ograničenjem skladišnog prostora.



Slika 5-5. Dinamički diskretni višetapni EOQ spredu model za više proizvoda sa ograničenim skladišnim prostorom, (Antić, 2012)

## 6. Pristupi za obezbeđenje kvaliteta modela u spredšitovima

U prethodnim poglavljima rada opisani su koncepti operacionog menadžmenta, upravljanja zalihama, dinamičkih diskretnih upravljačkih modela, kao i modeliranja u spredšitovima. Cilj svakog od poglavlja bio je definisanje elementa iz određenih oblasti, potrebnih za kreiranje modela problema operacionog menadžmenta, kao dinamičkih diskretnih procesa upravljanja u spredšitu. Ovi modeli su se pokazali kao veoma praktični i primenljivi, ali i veoma kompleksni za kreiranje, zbog složene strukture i procesa prevođenja u spredšit. Usled navedenog, ali i činjenice da su krajnji korisnici stručnjaci u svojim oblastima, a ne profesionalni, obučeni programeri, jasno je da je mogućnost nastanka greške pri kreiranju i korišćenju modela velika. U nastavku rada biće opisan pojam greške u spredšitovima, kao i različiti pristupi za predupređenje, detekciju i analizu spredšit grešaka, koji treba da posluže kao putokaz za razvoj metode za detekciju i analizu grešaka u dinamičkim diskretnim spredšit modelima upravljačkih problema operacionog menadžmenta.

### 6.1. Greške u spredšit modelima

Spredšit aplikacije, danas najčešće bazirane na *MS Excel* softverskom alatu, koriste se na mnogobrojnim radnim mestima, na skoro svim nivoima poslovne ili proizvodne organizacije. Ove interaktivne računarske aplikacije često razvijaju korisnici koji nisu programeri, već stručnjaci u svojim oblastima poslovanja, kao što su finansije, planiranje, predviđanje, donošenje odluka itd. Spredšit sistemi su postali popularni 80-ih godina dvadesetog veka, kao najpoznatija paradigma programiranja od strane krajnjih korisnika. Njihova glavna prednost je mogućnost, koju pružaju ekspertima iz različitih oblasti poslovanja, da kreiraju sopstvene alate za specifične probleme, u koje će ugraditi svoje ekspertsko znanje. Vreme potrebno za kreiranje takvih aplikacija je najčešće značajno kraće nego za izradu klasičnih poslovnih aplikacija, koje se prave u okviru IT odeljenja preduzeća i u skladu sa standardima kompanije o obezbeđenju kvaliteta. Jasno je da opisana karakteristika spredšitova, da omogućavaju krajnjim korisnicima brzu i jednostavnu izradu modela i aplikacija, uzrokuje čest nastanak grešaka.

Kada je u pitanju pojam greške, postoji mnogo definicija datih od strane različitih autora. Prema IEEE standardima softverskog inženjerstva, greška (*engl. Error*) se odnosi na nerazumevanje od strane kreatora spredšita, uzrokovano nesporazumom ili pogrešnim razumevanjem problema, koje se javlja u procesu ljudskog razmišljanja. Neispravnost (*engl. Fault*) predstavlja manifestaciju greške u spredšitu, koja može uzrokovati otkaz. Otkaz (*engl. Failure*) se definiše kao devijacija ili odstupanje realizovanog ponašanja spredšita od očekivanog. Ipak, u mnogim radovima ovi termini se koriste kao sinonimi.

Istraživanja na temu grešaka u različitim oblastima pokazala su da ne postoji skoro nijedna aktivnost koju čovek može ponoviti hiljadu puta, a da ne napravi neprimećenu grešku, kako se navodi u (Panko, 2000a). Nedetektovane greške se javljaju prilikom izvođenja jednostavnih aktivnosti, kao što su kucanje ili selektovanje u proseku od 0,2% do 0,5%. Procenat pojave grešaka kod složenijih kognitivnih aktivnosti čoveka, kao što je pisanje kompleksnih formula ili linija koda, kreće se od 2% do 5%. Na sajtu posvećenom istraživanjima o spredšitovima (Panko, 2000b), postoje podaci o brojnim eksperimentima koji uključuju više od hiljadu subjekata, od početnika do iskusnih profesionalaca. U svim ovim eksperimentima barem 1% ćelija sadrži greške.

Usled lakoće učenja o radu u spredšitovima sa jedne strane, i mogućnosti izvođenja kompleksnih analiza sa druge, oni su prihvaćeni od strane velikog broja korisnika od početnika do eksperata. Fleksibilnost spredšitova omogućava korišćenje bez primene striktnih pravila. Upravo ta karakteristika čini spredšitove sklonim razvoju grešaka i lošem dizajnu modela i aplikacija. Greške se, usled nedovoljne obučenosti korisnika, lako prave, ali teško uočavaju. Eksperimenti prikazani u (Powell, Baker, & Lawson, 2008) pokazali su da su spredšitovi podložniji greškama nego ostali softveri. U okviru Evropske interesne grupe za rizik u spredšitovima (EuSpRiG<sup>3</sup>) prikupljeni su i predočeni značajni dokazi o slučajevima grešaka u spredšitovima koje su prouzrokovale velike finansijske gubitke u kompanijama. Ovi slučajevi ističu uticaj koji greške u spredšitovima mogu imati na kompanije. Međutim, za razliku od procesa ili softvera koji su u organizacijama validirani i dobro kontrolisani, za spredšitove ne postoje

---

<sup>3</sup> <http://eusprig.org/horror-stories.htm>

odgovarajuće procedure, ni standardi. Rizik pojave grešaka u spreadšitovima izaziva potrebu za unapređenjem metoda i alata za detekciju istih. Svest o riziku ovog tipa u mnogome se razvila u poslednje dve decenije. U skladu da tim, naučnici i istraživači su predlagali više metodologija, tehnika i automatizovanih alata za otkrivanje grešaka, koje bi krajnji korisnici primenjivali. Pregled najnovije literature na ovu temu pokazuje da je razvijeno veliki broj klasifikacija pristupa za obezbeđenje kvaliteta spreadšitova, od vizualizacije spreadšitova, preko statičkih analiza i izveštaja, testiranja, do podrške zasnovane na modelima razvoja.

Spreadšit greške se javljaju u makar 30% svih spreadšit modela, kako se navodi u (Panko, 1998). Jedan od primer uticaja grešaka u spreadšitovima na poslovanje predstavlja gubitak od 24 miliona dolara u kompaniji *Trans Atlanta Corporation*, koji je nastao usled greške kopiranja prilikom pravljenja energetskih ugovora u Njujorku, (Thorne, Ball, & Lawson, 2004 ). Ukoliko se posmatraju manje organizacije, na osnovu istraživanja prikazanih u (Powell, Baker, & Lawson, 2009), zaključeno je da je u 50 spreadšitova iz 5 organizacija stopa grešaka između 1-5%. Uticaj ovih grešaka pokazao se veoma promenljivim, od malih novčanih vrednosti do preko 100 000 dolara. Na osnovu navedenih podataka može se zaključiti da, bez obzira na veličinu kompanije, greške u spreadšitovima predstavljaju veliki rizik za poslovanje. Istraživanja na temu spreadšit grešaka obuhvataju eksperimente, taksonomije, zapažanja iz prakse, tehnička rešenja za smanjenje grešaka, teorije upravljanja greškama u spreadšitovima, ručne metode ispitivanja i softverske alate.

Pristupi testiranju spreadšitova su veoma aktuelna tema, usled značajne potrebe da se stopa grešaka u spreadšitovima smanji. Kako se predstavlja u (Panko, 2005), (Panko, 2006):

- Spreadšitovi se koriste u mnogim kompanijama, pri finansijskom izveštavanju i drugim ključnim poslovnim aktivnostima u različitim oblastima;
- U skladu sa istraživanja o stopi ljudskih greška u realnim spreadšitovima, može se tvrditi da kreatori spreadšitova prave neuočene greške u 2% do 5% formula;
- Skoro svi veliki spreadšitovi imaju više grešaka;

- Ispitivanje po oblastima pokazalo je da se najveći broj grešaka pojavljuje u spreadšitovima finansijske prirode ili značajno utiče na donošenje odluka.

Većina radova posvećena je smanjenju grešaka pri razvoju spreadšitova, dok je značajno manji broj radova napisan o samom testiranju npr. (PriceWaterhouseCoopers, 2004). Ovaj rad predstavlja različite tipove testiranja koje spreadšit kreatori mogu da primenjuju.

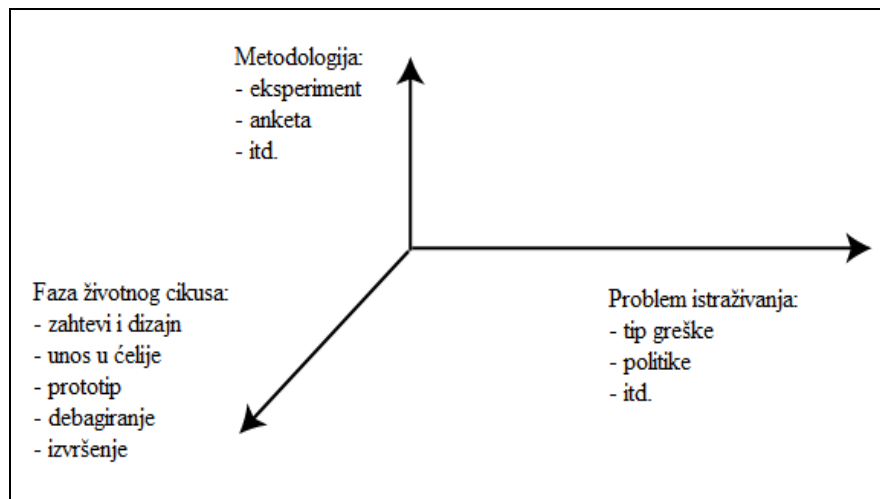
## **6.2. Taksonomija spreadšit grešaka**

Klasifikaciona šema spreadšit grešaka treba da obuhvata najčešće i najvažnije greške. Kako se navodi u (Ayalew, 2001), efikasnost prevencije grešaka i tehnike detekcije mogu se proceniti na osnovu taksonomije grešaka koja obuhvata tipove, učestalost i moguće uzroke. Ipak, u skladu sa brojnim klasifikacijama definisanim od različitih autora, može se zaključiti da ne postoji univerzalni način za kategorizaciju grešaka. Greška može pripadati različitim kategorijama, u zavisnosti od stava programera i uzroka greške. U skladu sa važnošću spreadšit grešaka ova tema je često razmatrana, kako se može videti u brojnim radovima: (Howe & Simkin, 2006), (Rajalingham, Chadwick, & Knight, 2000), (Powell, Baker, & Lawson, 2009), (Panko & Aurigemma, 2010), (Powell, Baker, & Lawson, 2008), (Panko, 2000a), (Ayalew, Clermont, & Mittermeir, 2000), (Caulkins, Morrison, & Weidemann, 2006) i drugi.

Jedna od prvih klasifikacija različitih tipova grešaka, bazirana na dotadašnjem pregledu literature, prikazana je u (Ronen, Palley, & Halverson, 1989). U radu se prikazuju kategorije koje se odnose na sam model, kao što su greške u logici i one koje se odnose na izvršenje modela, kao što je pogrešna referenca ćelije.

U (Panko & Halverson, 1996) prvi put se teži kompletnom modelu spreadšit grešaka i razdvajanju kvalitativnih i kvantitativnih grešaka. Kvantitativne greške su numeričke greške, koje odmah uzrokuju pogrešan rezultat. Kvalitativne greške ne vode odmah pojavljivanju greške, ali je mogu prouzrokovati kada se spreadšit menja. Predstavljaju se tri glavne kategorije grešaka (Panko & Halverson, 1996):

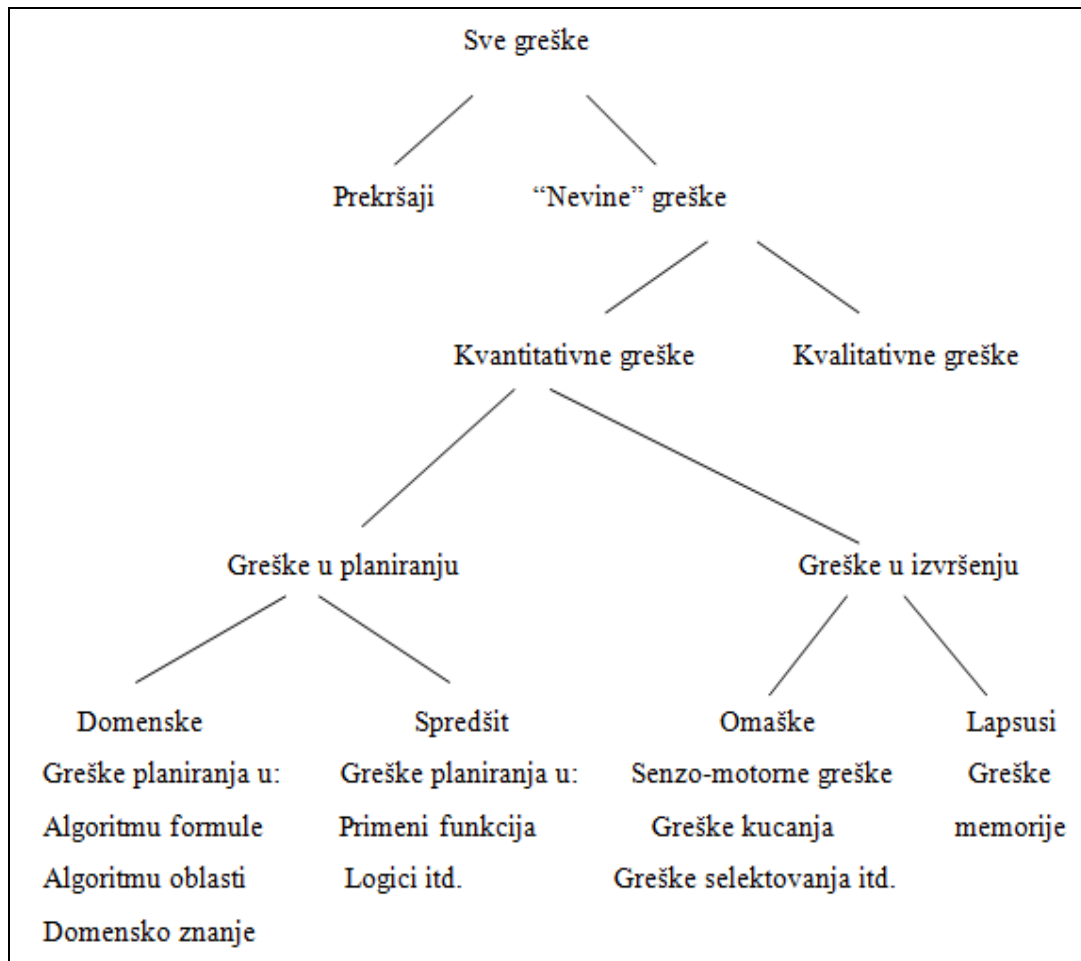
- **Mehaničke greške** - nastaju kao greške kucanja, selektovanja ili druge jednostavne omaške;
- **Logičke greške** - nastaju kao rezultat izbora pogrešnog algoritma ili kreiranja pogrešne formule;
- **Greške propusta** - rezultat su izostavljanja dela modela i često vode pogrešnoj interpretaciji rezultata modela.



**Slika 6-1.** Metrike za merenje spredšit grešaka, (Panko i Halverson, 1996)

Isti izvor predstavlja istraživanje o greškama u razvoju i inspekciji spredšitova. Autori su kreirali taksonomiju istraživanja spredšit rizika u formi trodimenzionalne kocke, prikazane Slikom 6-1. Tri strane kocke odnose se na problem istraživanja, fazu životnog ciklusa i metodologiju (eksperiment, anketa itd.) za istraživanje. Problem istraživanja odnosi se na strukturu, stvarne greške, iskustvo korisnika, pretpostavke i karakteristike spredšit modela (veličinu, procenat ćelija sa formulama ili podacima, kompleksnost formula, frekvenciju upotrebe spredšita (jednom ili više puta), broj ljudi koji ga koristi, namenu) i kontrolne politike.

Panko i Halverson taksonomija je revidirana i proširena u (Panko & Aurigemma, 2010). Svrha rane taksonomije je bila da podrži kvantitativna istraživanja i pokaže da su kvantitativne greške u spredšitovima česte, teške za detektovanje, kao i da su spredšitovi sa greškama značajni.



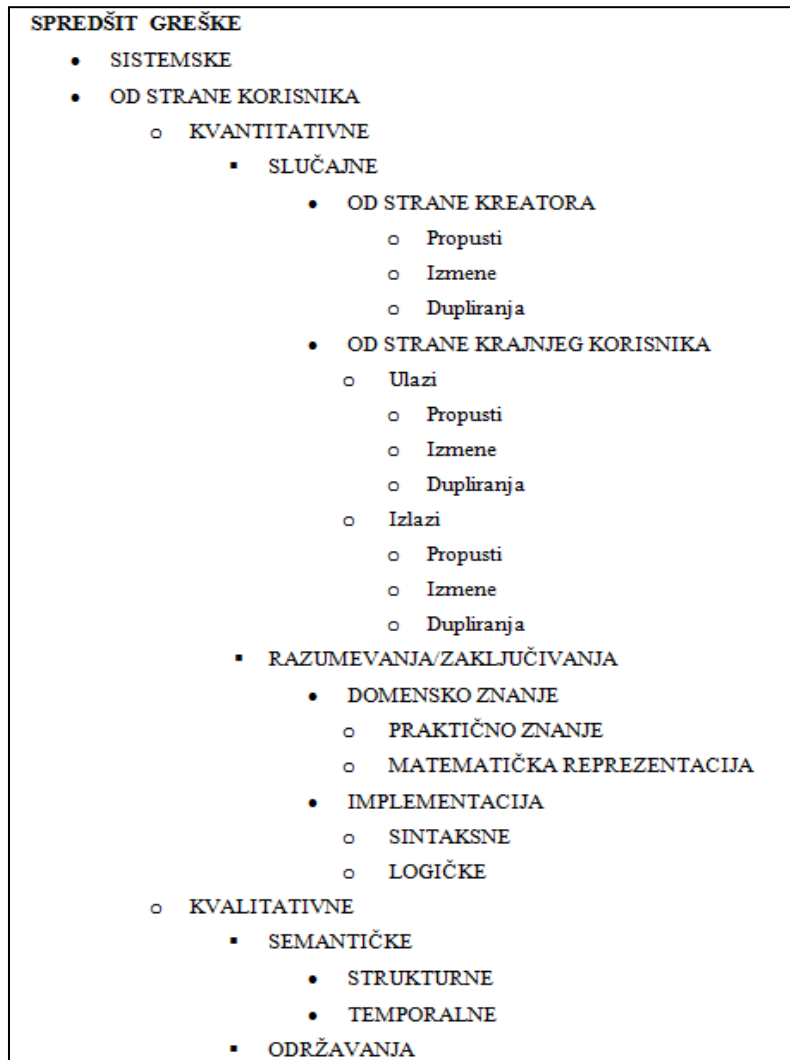
**Slika 6-2.** Revidirana taksonomija spređšit grešaka, (Panko and Aurigemma, 2010)

Slika 6-2. prikazuje revidiranu taksonomiju u kojoj ima dosta razlika u odnosu na prethodnu:

- Postoji razlika između nevinih grešaka i prekršaja korporativne prakse;
- Razlika između logičkih, mehaničkih i greška propusta zamenjena je češće korišćenom razlikom između domenskih i grešaka planiranja izraza u spređšitu sa jedne strane i greška implementacije (omaški i lapsusa) sa druge. Greške planiranja predstavljaju pogrešnu implementaciju planova;
- Domenske greške planiranja pojavljuju se kada kreator naprave greške u domenskom poznavanju modela (finansije, ekologija, fizika itd.). Greške planiranja izraza u spređšitu dešavaju se kada kreator planira pogrešan izraz za domenski algoritam;



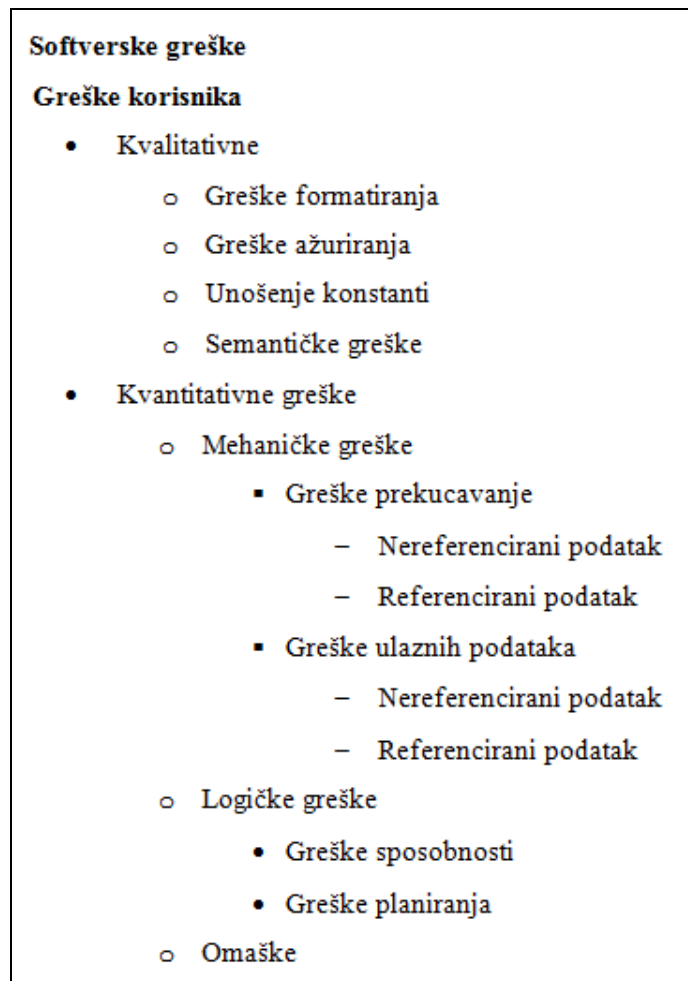
- Logičke greške postaju greške zamene, a mehaničke se dele na omaške i lapsuse. Omaške su senzomotorne greške, kao što su greške u kucanju ili selektovanju. Lapsusi su greške pamćenja.



**Slika 6-3.** Taksonomija spredšit grešaka, (Rajalingham et al., 2000)

U radu (Rajalingham, Chadwick, & Knight, 2000) predstavlja se taksonomija spredšit grešaka koja se zasniva na razlici između kvalitativnih i kvantitativnih grešaka. Za razliku od taksonomije prikazane u (Panko & Halverson, 1996), ovde se u daljoj distinkciji pojavljuju slučajne i greške rasuđivanja. Značajan element u ovoj taksonomiji grešaka je razlika između kreatora spredšit modela i krajnjih korisnika. Ovi autori u razmatranje uzimaju i greške koje nastaju u interpretaciji rezultata spredšita. Ukoliko model daje tačan rezultat, ali se on pogrešno interpretira, na primer usled loše

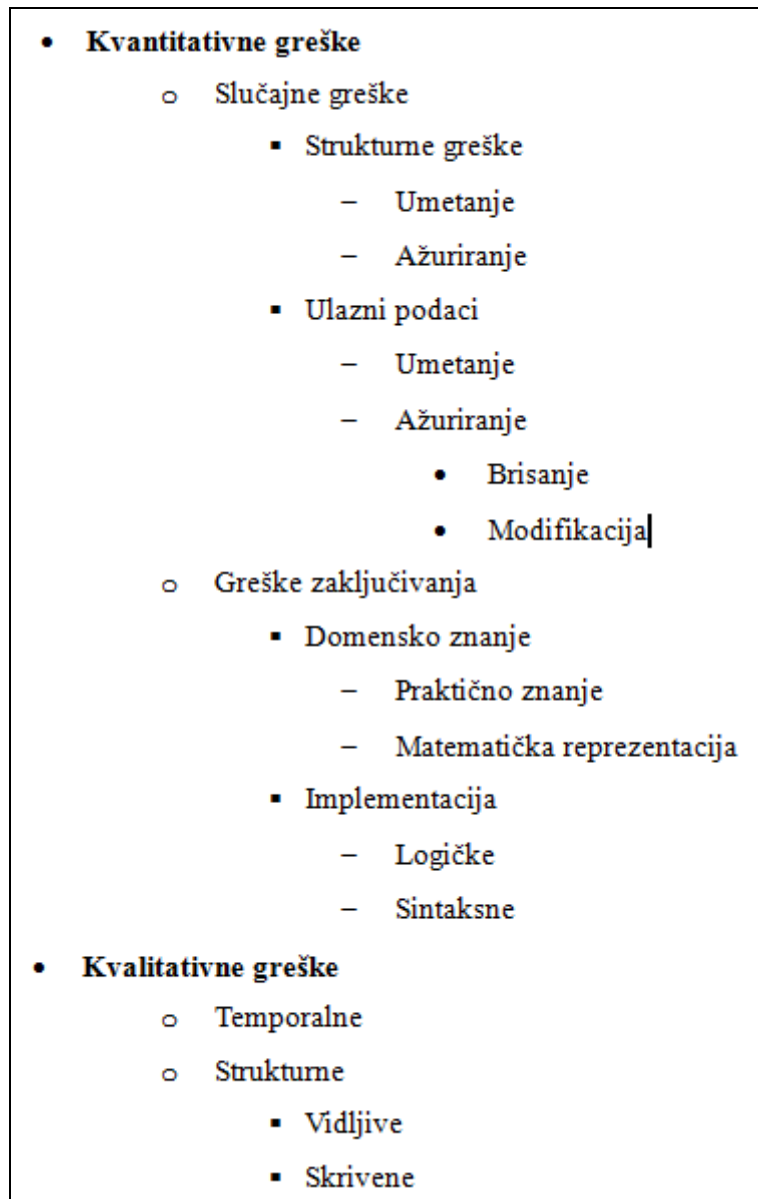
postavljenih zaglavlja i formatiranja, javlja se podjednako važna greška koliko i greška kreatora. Za razliku od *Panko* i *Halverson* taksonomije, koja se zasniva na uzroku greške, ova taksonomija je opštija i opisna kao hijerarhijski sistem kategorija spredšit greška, baziran na zajedničkim karakteristikama i vezama. Slika 6-3. prikazuje klasifikaciju spredšit grešaka predstavljenu u (Rajalingham, Chadwick, & Knight, 2000).



**Slika 6-4.** Žbunasta taksonomija spredšit grešaka, (Rajalingham, 2005)

U (Rajalingham, 2005) predstavljena je njihova revidirana taksonomija. Na Slici 6-4. prikazana je revidirana tzv. žbunasta (*engl. Bushy*) taksonomija, koja se često grana na tri ili više alternative. Mana ove klasifikacije je poteškoća da se odredi gde treba svrstati grešku, a može se desiti i da se greška može svrstati u dve ili više klasa. *Rajalingham* u

istom radu predstavlja i “binarnu” taksonomiju. Binarni izbor u svakom koraku olakšava korišćenje klasifikacije (Slika 6-5.).



**Slika 6-5.** Binarna taksonomija spredšit grešaka, (Rajalingham, 2005)

### 6.3. Pristupi za detekciju i analizu grešaka u spredšitovima

Pristupi za detekciju i analizu grešaka u spredšitovima vrlo često se zasnivaju na primeni metoda i alata iz oblasti softverskog inženjerstva. Ipak, zahtevi koji se postavljaju za pristupe obezbeđenja kvaliteta spredšitova se razlikuju od onih

namenjenih tipičnim imperativnim jezicima. Neke od glavnih razlika, u kontekstu testiranja spreadšitova, opisane su u radu (Rothermel, Dupuis, & Burnett, 1998). Interakcija između korisnika i spreadšit okruženja se značajno razlikuje od načina na koji su programi imperativnih jezika kreirani. Na primer, korisnici često započinju razvoj spreadšita nestruktuiranim inkrementalnim procesom, koristeći podatke za testiranje. Za date podatke korisnici odmah dobijaju povratnu vizualnu informaciju. Na osnovu navedenog, jasno je da alat koji se razvija za spreadšit korisnike mora biti kreiran tako da podrži inkrementalni proces razvoja, i u isto vreme obezbedi trenutne povratne informacije. Osim već spomenute razlike, paradigma proračuna u spreadšitovima je značajno drugačija od imperativnih jezika. Osnovna karakteristika spreadšitova je da se proračuni ili evaluacija programa, zasniva na zavisnostima podataka u ćelijama, dok su pravila sadržana u formulama. Ova činjenica se mora uzeti u obzir pri definisanju mehanizama i alata za obezbeđenje kvaliteta. Na primer, kada se usvajaju i prilagođavaju postojeći pristupi imperativnih programa, moraju se razmatrati različite karakteristike i mere kvaliteta, a zavisnosti podataka su relevantnije od toka podataka. Takođe, konceptualni model se zasniva više na zavisnosti podataka i formula, nego izvršenju naredbi. Još jedna značajna razlika je što se spreadšit programi ne zasnivaju samo na jednostavnijim modelima proračuna, tj. fizičkom rasporedu, već je prostorni raspored labela i formula značajno određen semantikom proračuna. Ove prostorne informacije mogu se koristiti za detekciju inkonzistentnosti susednih ćelija i određivanje verovatnoće da je formula semantički tačna, za automatsko zaključivanje na osnovu labela ili rangiranje preporuka za ispravku, kod pristupa detekcije grešaka zasnovanih na cilju (Erwig, 2009).

U obzir se mora uzeti i da kreatori spreadšita uglavnom nisu profesionalni programeri. Kreatori imperativnih programa pohađaju formalne treninge i edukovani su u oblasti razvoja softvera, i svesni su važnosti sistemskih procesa obezbeđenja kvaliteta. Korisnici koji razvijaju spreadšitove uglavnom imaju ograničeno interesovanje i svesnost o potrebama za dodatnim aktivnostima obezbeđenje kvaliteta spreadšitova, kao što je testiranje. Zbog toga, metodologije i alati moraju biti jednostavni za razumevanje i primenu, bez potrebe za dugim treninzima i učenjem novih tehnika.

Prema (Jannach et al., 2014), pristupi obezbeđenja kvaliteta spredšitova mogu se klasifikovati u dve osnovne kategorije, u zavisnosti od njihove uloge i upotrebe u životnom ciklusu razvoja:

- “Pronalaženje i ispravljanje grešaka” - odnosi se na tehnike i alate koji su dizajnirani da pomognu korisnicima da detektuju greške i utvrde uzrok nastanka. Ove alate najčešće koriste kreatori, revizori ili recenzenti, u toku ili nakon izrade spredšit modela ili aplikacija.
- „Izbegavanje grešaka“ - obuhvata tehnike i alate koji podržavaju kreiranje spredšitova koji nemaju greške. Oni se primenjuju u toku procesa razvoja spredšit modela ili aplikacija.

Detaljnija kategorizacija postojećih pristupa automatizovanom obezbeđenju kvaliteta spredšitova, prema (Jannach et al., 2014), prikazana je u Tabeli 6-1.

**Tabela 6-1.** Osnovne kategorije pristupa automatizovanog obezbeđenja kvaliteta spredšitova, (Jannach et al., 2014)

Rb.	Pristupi automatizovanog obezbeđenja kvaliteta spredšitova	Pronalaženje grešaka	Izbegavanje grešaka
1.	Pristupi zasnovani na vizualizaciji	x	x
2.	Statičke analize koda i izveštaji	x	x
3.	Pristupi testiranja	x	
4.	Automatska lokalizacija i ispravka grešaka	x	
5.	Pristupi razvoju zasnovani na modelima		x
6.	Projektovanje i podrška održavanju		x

Pristupi 1 i 2 mogu da se koriste za otkrivanje i izbegavanje greška. Na primer, dobra vizualizacija zavisnosti ćelija podstiče uočavanje problema. U isto vreme, vizualizacija se može koristiti za isticanje ćelija i oblasti koje imaju veliku verovatnoću da će se greška u njima javiti u budućnosti, na primer, kada postoje ponavljajuće strukture u spredšitu. Statističke analize mogu identifikovati već postojeće probleme, kao što je referenciranje praznih ćelija, ali i služiti kao indikatori potencijalnih problema, na primer, izlistavanjem formula koje su kompleksne. Pristupi iz kategorija 3 i 4 uglavnom

se odnose na problem nalaženja i ispravljanja grešaka, s obzirom da ih korisnici primenjuju da identifikuju postojanje problema ili da lokalizuju uzrok greške. Pristupi kategorija 5 i 6 namenjeni su izbegavanju grešaka, na primer, procesom refaktorisanje ili dodavanjem nivoa apstrakcije.

Pristupi zasnovani na vizualizaciji obezbeđuju korisniku poboljšano predstavljanje određenih aspekata spredšitova, na osnovu kojih se lakše sagledavaju i razumeju odnosi i zavisnosti ćelija ili blokova spredšita. Vizualizacija omogućava korisniku bržu detekciju anomalija i neregularnosti u spredšitu. Statičke analize i izveštaji bazirani su na statičkoj analizi koda, sa ciljem da ukažu kreatoru na potencijalno problematične oblasti spredšita. Primeri ovih tehnika uključuju “miris koda” ili detekciju klonova podataka, ali i grupe tipičnih tehnika korišćenih u komercijalnim alatima, koji detektuju povratne veze ili prave izveštaje o nereferenciranim ćelijama. Pristupi zasnovani na testiranju imaju za cilj da stimulišu i podrže kreatore da sistematično testiraju spredšit aplikacije u toku i nakon razvoja. Ovi alati uključuju mehanizme za testiranje slučajeva, automatsko generisanje slučajeva za testiranje ili analize pokrivenosti različitih slučajeva. Pristup automatske lokalizacije i ispravke grešaka zasniva se na računarskim analizama mogućih uzroka grešaka ili neočekivanog ponašanja (algoritamsko otklanjanje grešaka). Oni obuhvataju uključivanje dodatnih ulaza od strane kreatora, u obliku slučajeva za testiranje ili proveru tačnosti ćelija. Neki od ovih alata omogućavaju ispravku na osnovu “sugestija”. Pristupi razvoju zasnovani na modelima, usvajaju ideju primene objektno-orijentisanih konceptualnih modela, kao i tehnike razvoja vođene modelom, koje su danas opšte poznate i primenjene u softverskoj industriji. Prednosti ovih pristupa su uvođenje dodatnih nivoa apstrakcije ili upotreba mehanizama za generisanje koda. Projektovanje i podrška održavanju predstavlja pristup koji kreatoru olakšava izradu spredšit strukture bez grešaka. Pristup uključuje automatizovane alate za refaktorisanje, metode za izbegavanje pogrešne reference ćelija i rukovanje izuzecima. U Tabeli 6-2. prikazani su različiti pristupi u okviru svake kategorije automatizovanog obezbeđenja kvaliteta spredšitova (Jannach et al., 2014).

**Tabela 6-2.** Pristupi automatizovanog obezbeđenja kvaliteta spredšitova,  
(Jannach et al., 2014)

1.	Pristupi zasnovani na vizualizaciji	<ul style="list-style-type: none"> <li>1.1 Vizualizacija tokova podataka i zavisnosti</li> <li>1.2 Vizualizacija povezanih oblasti</li> <li>1.3 Vizualizacija zasnovana na semantici</li> <li>1.4 Pristupi vizualizacije informacija</li> </ul>
2.	Statičke analize koda i izveštaji	<ul style="list-style-type: none"> <li>2.1 Zaključivanje na osnovu jedinica i tipova</li> <li>2.2 Spredšitovi sumnjive strukture</li> <li>2.3 Statičke analize i komercijalni alati</li> </ul>
3.	Pristupi testiranja	<ul style="list-style-type: none"> <li>3.1 Testiranje adekvatnosti i slučajeva</li> <li>3.2 Automatsko generisanje test slučajeva</li> <li>3.3 Testiranje na osnovu uslova</li> <li>3.4 Razvoj spredšitova vođen testiranjem</li> </ul>
4.	Automatska lokalizacija i ispravka grešaka	<ul style="list-style-type: none"> <li>4.1 Rangiranje kandidata na osnovu tragova</li> <li>4.2 Lokalizacija grešaka na osnovu ograničenja</li> <li>4.3 Pristupi popravke</li> </ul>
5.	Pristupi razvoja vođeni modelom	<ul style="list-style-type: none"> <li>5.1 Deklarativni spredšit modeli</li> <li>5.2 Spredšit šabloni</li> <li>5.3 Objektno-orijentisani vizualni modeli</li> <li>5.4 Relacioni spredšit modeli</li> </ul>
6.	Projektovanje i održavanje	<ul style="list-style-type: none"> <li>6.1 Upravljanje referenciranjem</li> <li>6.2 Rukovanje izuzecima</li> <li>6.3 Promene i evolucija spredšita</li> <li>6.4 Refaktorisanje</li> <li>6.5 Ponovna upotreba</li> </ul>

### 6.3.1 Vizualizacija spredšitova

Slaba vidljivost zavisnosti ćelija predstavlja jednu od značajnih poteškoća u razumevanju i analizi spredšit programa. Pristupi vizualizacije za cilj imaju obezbeđenje vidljive grafičke predstave strukture proračuna i zavisnosti formula u spredšitu. Grafičko predstavljanje može uključiti strelice, boje, senčenje i slično, što prikazuje grupe povezanih ćelija i njihove interakcije. Pored prikazivanja mogućih grešaka, alati za vizualizaciju imaju potencijal da spredšit program učine lakšim za razumevanje.

Iako se pristupi vizualizacije spredšitova baziraju na predstavljanju sakrivene strukture toka podataka u grafičkoj formi, oni variraju kada su u pitanju procedure i načini ispitivanja prethodnika i sledbenika. U nekim pristupima se koriste odvojeni grafički prikazi, dok se drugi baziraju na prikazivanju celog spredšita. Razlike u pristupima vizualizacije mogu se istraživati iz nekoliko perspektiva, kao što su mogućnost primene koncepta oblasti, veza formula, navigacije i zumiranja različitih nivoa pregleda.

#### 6.3.1.1 Vizualizacija tokova podataka i zavisnosti

Pristup vizualizacije tokova podataka i odgovarajućih zavisnosti formula u ćelijama spredšita razmatran je u mnogim radovima (Hodnigg & Mittermeir, 2008), (Kankuzi & Ayalew, 2008), (Ayalew, 2009), (Hermans, Pinzger, & Deursen, 2011a), (Hermans, Pinzger, & Deursen, 2011b) i drugi. U mnogim slučajevima koriste se strelice za predstavljanje ćelija prethodnika i sledbenika određene formule. Ovo je standardna opcija komercijalnog spredšit okruženja.

Većina pristupa vizualizacije zasniva se na ekstrakciji zavisnosti ćelija i prikazivanju istih u grafičkoj formi. Jedan od mogućih pristupa, prikazivanje grafa zavisnosti podataka u trodimenzionalnom prostoru, predstavljen je u radu (Shiozawa, Okada, & Matsushita, 1999). Iako su spredšitovi sami po sebi dvodimenzionalni, treća dimenzija se koristi za predstavljanje međućelijske zavisnosti. Željena formula ćelije se podiže na z-osu. Kako je ova ćelija na višem nivou, ćelije od kojih zavisi odabrana ćelija ili one koje zavise od nje, će biti jedan nivo ispod. Na ovaj način zavisnosti ćelija se mogu ispitivati bez dodatnih strelica i bojenja.



Pored grafova tokova podataka, za bolje razumevanje spredšitova, može se koristiti mehanizam dekompozicije na različite funkcionalne delove. U radu (Chen & Chan, 2000) koriste se dodatne tehnike za uključivanje funkcionalnih delova (modula podataka) u vizualizaciju spredšita. Vizualizuju se zavisnosti većih blokova formula iz susednih ćelija, umesto prikazivanja strelica između individualnih ćelija. Prva identifikacija se pravi između ulaznih, međurezultata i izlaznih ćelija na osnovu ćelija prethodnika i sledbenika. Na ovaj način lako se detektuju ćelije sa anomalijom, koje nemaju ni prethodnike ni sledbenike. Veze između blokova se takođe ističu, u cilju identifikacije između različitih grupa ćelija.

Najnoviji pristup vizualizaciji spredšitova predstavljen je u radu (Hermans, 2013) U ovom pristupu razmatra se podrška korisnicima spredšita za bolje razumevanje i prilagođavanje dizajna spredšita. Osnovna istraživačka pitanja odnose se primenljivost metoda i tehnika softverskog inženjerstva na spredšitove, odnosno kako se može vizualizovati dizajn spredšita u cilju boljeg razumevanja istog, kao i u kojoj meri vizualizacija i analiza dizajna spredšita pomaže korisnicima u oceni kvaliteta spredšita. Odgovori na ova pitanja obuhvataju različite delove od kojih se dizajn spredšita sastoji. Prvo, spredšitovi sadrže podatke. Zatim, postoji aspekt metapodatka, podatka koji je opis drugih podataka. Iako spredšit sistem ne pravi razliku između ova dva tipa podataka, korisnici ih razlikuju. Treća, centralna, komponenta su formule na osnovu kojih se vrši proračun u spredšitovima. Četvrta je organizacija spredšita, koja se odnosi na način na koji su podaci podeljeni u radnom listu u različite tabele. Svaki od ova četiri aspekta se analizira i vizualizuje različitim metodama i tehnikama.

U istom radu (Hermans, 2013), predstavlja se tehnika automatskog izvlačenja metapodataka iz spredšita, i transformacija ovih podataka u dijagram klasa. Dijagram klasa može se koristiti za razumevanje, preradu ili reimplementaciju funkcionalnosti spredšita. Da bi se ova transformacija omogućila, kreirana je biblioteka najčešće korišćenih šablona u spredšitovima. Šabloni su lokalizovani upotrebom dvodimenzionalnog algoritma raščlanjivanja. Dobijeni raščlanjeni algoritam je transformisan i dopunjen informacijama iz biblioteke.

Isti autor (Hermans, 2013), opisuje rezultate studije o problemima i informacijama profesionalnih spreadšit korisnika, koja je sprovedena u jednoj velikoj finansijskoj kompaniji. Na spreadšitovima koji se koriste u kompaniji primenjen je pristup ekstrakcije potrebnih informacija iz spreadšita i njihovo prezentovanje u obliku kompaktnom i lakom za razumevanje - dijagramom toka podataka po nivoima. Pristup predstavlja tri različita pogleda na tok podataka, koji omogućavaju korisniku da analizira dijagram toka odozgo-nadole. Da bi se pristup ocenio, sprovedena je serija intervjuja i devet studija slučaja u industrijskom okruženju. Rezultati su pokazali da ovaj pristup olakšava razumevanje spreadšitova. Na osnovu sprovedenih istraživanja pokazano je da je moguće primeniti metode softverskog inženjerstva na spreadšitove. Analizirani spreadšitovi nisu prethodno pripremani ili menjani, niti su njihovi korisnici dodatno obučavani.

### **6.3.1.2 Vizualizacija povezanih oblasti**

Različite metode i alati za identifikaciju i vizualizaciju semantički povezanih ili strukturno sličnih blokova ćelija prikazane su u radovima: (Mittermeir & Clermont, 2002), (Hipfl, 2004), (Clermont, 2003), (Clermont, 2008). Ovi blokovi mogu se naglasiti korišćenjem različitih boja, kako bi korisnici lakše razumeli logičku strukturu spreadšita ili identifikovali neregularnosti (Clermont, 2005).

Koncept logičkih opsega prikazan je u radu (Mittermeir & Clermont, 2002). Opsezi se automatski identifikuju traženjem strukturno sličnih (istih) formula u različitim delovima spreadšita. Takvi opsezi su, na primer, rezultat kopiranja formula tokom procesa konstrukcije spreadšita. Eksperimenti bazirani na ovoj ideji pokazali su da postoje ograničenja koncepta za veće spreadšitove, zbog čega je uveden koncept semantičkih klasa. Korisnici manuelno specificiraju povezane opsege u spreadšitu. Na osnovu ovako određenog ulaza i informacija o prostornom rasporedu potencijalno povezanih ćelija, može se izvršiti dalje zaključivanje o opsezima sa velikom sličnošću. Ova razmatranje su unapređena u radu (Hipfl, 2004), gde se semantičke klase identifikuju na osnovu informacija sadržanih u labelama ćelija i pomoću skupa heuristika. Alternativna metoda dekompozicije spreadšita, u cilju vizualizacije, predstavljena je u (Clermont, 2003) i (Clermont, 2008). Identifikacija opsega zasnovana

je na osobinama toka podataka u spredšitu. U radu (Sajaniemi, 2000) predložena su dva pristupa vizualizacije, nazvana S2 i S3. Osnovna ideja je detekcija ekvivalentnih formula u blokovima ćelija i vizualizacija zavisnosti između individualnih blokova.

### **6.3.1.3 Vizualizacija zasnovana na semantici**

Neispravna semantika u formulama u spredšitovima, koji je uzrokovan upotrebom velikog broja referenci ćelija, umesto imena koja nose informacije, je veoma poznat problem i razlog nastanka velikog broja grešaka u spredšitovima.

Pristup rešavanja ovog problema prikazan u radu (Chadwick, Knight, & Rajalingham, 2001) bazira se na zapažanju dva tipična tipa grešaka, koja često nastaju pri kreiranju formula: referenciranje pogrešne ćelije kao ulaza, i pogrešno kopiranje formula. Za rešavanje prvog problema predlaže se tehnika kojom se olakšava i čini intuitivnim razumevanje formula. Za definisanje formula se mogu koristiti imenovane oblasti, zasnovane na labelama ili se kompleksne formule mogu predstaviti u vizualnoj formi. Da bi se formule vizualizovale, potrebno ih je dekomponovati, reference ćelija zameniti imenima, a operatore prevesti u prirodne jezike, tako da formula bude lakše razumljiva. Za problem pogrešno kopiranih opsega, isti izvor preporučuje primenu vizualnih indikatora i označavanje kopiranih ćelija i njihovih izvora istom bojom.

### **6.3.1.4 Pristupi vizualizacije informacija**

Pristupi vizualizacije informacija mogu se primeniti na analizu spredšitova, što je prikazano u (Brath & Peters, 2006). Ovakva vizualizacija treba da pomogne kreatoru u procesu detekcije anomalija u spredšitu. Za razliku od prethodno opisanih koncepata, cilj nije vizualizacija toka podataka ili strukture spredšita, već samih podataka. U tu svrhu predlaže se 3D reprezentacija, gde se vrednosti ćelija, na primer, prikazuju visinom stubića umesto brojem. Posmatrajući ovako predstavljen spredšit, korisnik može uočiti vrednosti koje odstupaju u odnosu na ostale ćelije.

Posmatrajući radove, koji se bave temom vizualizacije informacija u spredšitovima, (Rao & Card, 1994), (Ballinger, Biddle, & Noble, 2003), može se zaključiti da postoji

veliki broj tehnika iz ove oblasti koji se mogu primeniti na velike spreadšitove, u cilju inspekcije.

### **6.3.2 Statičke analize i izveštaji**

Pristupi statičke analize koda ili izveštavanje, analiziraju formule spreadšita i prikazuju moguće greške ili loš dizajn spreadšita, koji može voditi ka nastanku grešaka. Pristupi ove kategorije ne koriste vrednosti iz ćelija ili informacije test slučajeva za otkrivanje grešaka. Umesto toga, oni analiziraju formule i njihove zavisnosti, uzimaju u obzir statičke labele i određuju druge strukturne karakteristike spreadšitova.

#### **6.3.2.1 Zaključivanje na osnovu jedinica i tipova**

Pristupi zaključivanja na osnovu jedinica i tipova predstavljaju vrlo aktuelnu temu, obrađivanu u brojnim radovima: (Erwig & Burnett, 2002), (Burnett & Erwig, 2002), (Ahmad et al., 2003), (Abraham & Erwig, 2004), (Antoniou et al., 2004), (Abraham & Erwig, 2006), (Abraham & Erwig, 2007), (Chambers & Erwig, 2009), (Chambers & Erwig, 2010a). Osnovna ideja ovih pristupa je izvođenje informacija o jedinicama ulaznih ćelija i upotreba ovih informacija za procenu da li se proračuni u formulama verodostojni, u odnosu na jedinice uključenih ćelija. Za određivanje informacija o jedinicama ćelija koriste se njihova zaglavlja. Na primer, podaci iz kolone koja se odnosi na jabuke ne mogu se sabirati sa podacima koji se odnose na narandže. Uz pomoć ovakvog mehanizma zaključivanja na osnovu jedinica, semantika kalkulacija može se proveriti u cilju detekcije grešaka. Proces određivanja informacija o jedinicama iz zaglavlja naziva se zaključivanje na osnovu zaglavlja (*engl. Header Inference*). Ideja upotrebe sistema zaključivanja na osnovu jedinica, za identifikovanje potencijalnih grešaka uvedena je u radu (Erwig & Burnett, 2002). U ovom pristupu ćelije su mogle imati više od jedne jedinice. Ipak, u radu nisu date konkretne smernice, ni procedura kako sprovesti zaključivanje na osnovu zaglavlja. Pristup je kasnije unapređen u (Abraham & Erwig, 2004), tako da sistem može da funkcioniše bez ili sa ograničenom interakcijom korisnika. Druga unapređenja osnovnog pristupa uključuju različite oblike sofisticiranijeg zaključivanja i razmatrana su u radovima (Burnett & Erwig, 2002), (Abraham & Erwig, 2004), (Abraham & Erwig, 2006), (Abraham & Erwig, 2007),

(Chambers & Erwig, 2009), (Chambers & Erwig, 2010a). U (Chambers & Erwig, 2009) i (Chambers & Erwig, 2010a), izvodi se semantička analiza zaglavlja u cilju mapiranja zaglavlja i određivanja mernih jedinica. Na osnovu ovih informacija omogućava se preciznije zaključivanje o tačnosti proračuna.

### 6.3.2.2 Spredšitovi sumnjive strukture

Koncept spredšitova sumnjive strukture (*engl. Spreadsheet Smells*) izveden je iz sumnjive strukture koda u održavanju softvera (Fowler, 1999), a odnosi se na lošu strukturu koda (*engl. Code Design*). Ovakva struktura ne mora sadržati grešku, ali može voditi ka nastanku greške u budućem razvoju softvera, na primer, kada softver treba da se nadogradi ili refaktoriše (*engl. Refactoring*). Tipičan primer sumnjive strukture koda je dupliranje fragmenata koda. Ukoliko se isti deo koda nekoliko puta ponavlja u programu, obično je bolje napraviti funkciju tako da se eventualne promene dela koda sprovode samo jednom. Ponavljanje delova koda čini ga težim za čitanje.

Spredšitovi sumnjive strukture su u poslednjih nekoliko godina postali aktuelna tema u spredšit istraživanjima, kao što se može videti u radovima: (Hermans, Pinzger, & Deursen, 2012b), (Hermans et al., 2013), (Cunha et al., 2012). Slična ideja predložena je u kontekstu vizualizacije, gde se heuristike koriste za identifikaciju neregularnosti u spredšitovima (Clermont, 2005), (Brath & Peters, 2006).

Spredšitovi sumnjive strukture mogu se definisati kao heuristike koje opisuju lošu strukturu, koja vodi ka nastanku grešaka, prilikom uvođenja promena u spredšit ili kreiranja nove instance sa novim ulaznim podacima. U radu (Hermans, Pinzger, & Deursen, 2012a) predlaže se koncept sumnjive strukture unutar radnog lista. Ove sumnjive strukture ukazuju na lošu strukturu ili dizajn spredšita i zasnovane su na analizi zavisnosti između različitih radnih listova. Na primer, ukoliko formula ima mnogo referenci koje se odnose na drugi radni list, u njemu treba i da se nalazi. U radu (Hermans, Pinzger, & Deursen, 2012b) razmatra se sumnjiva struktura formule, koja se odnosi na lošu strukturu formule, kada je, na primer, formula veoma kompleksna. Evaluacija navedenih koncepata pokazala je da detekcija sumnjivih struktura uspešno

ukazuje na potencijalno „slabe tačke“ strukture spredšita. Metoda detekcije klonova u spredšitovima predlaže se u (Hermans et al., 2013). Na osnovu algoritama za detekciju klonova za tekstualne podatke, razvijen je algoritam koji detektuje klonove u spredšitu, odnosno formule čije su vrednosti kopirane kao tekst na drugu lokaciju. Nakon istraživanja ovog koncepta zaključeno je da se klonovi često javljaju, predstavljaju pretnju za kvalitet spredšita, kao i da predloženi koncept olakšava pronalaženje i rešavanje problema klonova.

### 6.3.2.3 Statičke analize i komercijalni alati

Tehnike statičke analize često predstavljaju deo komercijalnih spredšit okruženja i alata za detekciju grešaka. Na primer, *MS Excel* omogućava vizualizaciju „sumnjivih“ formula, na osnovu predefinisanih skupa pravila. Jedan od poznatih alata za detekciju grešaka je „*Spreadsheet Detective*“ (Nixon & O’Hara, 2001). Ovaj alat baziran je na identifikaciji „sumnjivih“ formula korišćenjem statičkih analiza i konstruisanjem odgovarajućih izveštaja. Za identifikaciju ovih formula upotrebljavaju se heuristike, koje u obzir uzimaju složenost formule, na primer, proverom višestrukih *IF* petlji. Još neki od popularnih komercijalnih alata za detekciju i analizu greška su:

- *Excel's Built-In Auditing Functions*;
- *The Excel Auditor*;
- *The Operis Analysis Kit*;
- *Spreadsheet Auditing for Customs and Excise (SpACE)* i drugi.

### 6.3.3 Testiranje spredšitova

U procesu razvoja softvera testiranje predstavlja važnu aktivnost, koja se sprovodi u više faza, a za cilj ima dostizanje visokog nivoa kvaliteta softvera. Kada su u pitanju spredšitovi, njihovi kreatori uglavnom nisu profesionalni programeri, upoznati sa metodama testiranja.

U skladu sa prirodom spredšitova i trenutnom povratnom informacijom, testiranje se može vrlo jednostavno sprovesti unošenjem ulaznih podataka, proverom međurezultata i

izlaznih vrednosti. Komercijalni alati ne obezbeđuju mehanizme za čuvanje test slučajeva ili izvođenje regresionih testova. Takođe, alati ne pružaju kreatoru mogućnost procene i upoređivanja rezultata više različitih testova. Usled navedenih problema javila se potreba za primenom i prilagođavanjem ideja, koncepata, alata i najbolje prakse iz oblasti testiranja softvera na razvoj spredšitova. Važan kriterijum za izbor metode testiranja je sposobnost metode da smanji greške u formulama sa 2% - 5%, na značajno manju vrednost, detektujući 60% - 80% svih grešaka u svakoj iteraciji testiranja.

### **6.3.3.1 Testiranje adekvatnosti i upravljanje test slučajevima**

Predstavnik ovog tipa testiranja je metodologija “*What You See Is What You Test*” (WYSIWYT), odnosno, ono što vidiš je ono što testiraš. WYSIWYT metodologija bazira se na pristupu formalnog testiranja i lokalizacije grešaka, originalno razvijenim za tradicionalne jezike bazirane na formama (*engl. Form-Based*). WYSIWYT metodologija koristi vizualne aparate za obezbeđenje povratne informacije o regularnosti spredšita u odnosu na kriterijum adekvatnosti toka podataka. Pristup je prvi put predstavljen u radu (Rothermel, Lixin, & Burnett, 1997) i naknadno potvrđen u (Rothermel et al., 1998). Evaluacija ovog pristupa pokazala je da on omogućava detekciju 81% grešaka. Primenljivost ove metodologije potvrđena je i u drugim radovima (Rothermel et al., 2000), (Kruck, 2006), (Carver, Fisher II, & Rothermel, 2006) i drugi. WYSIWYT metodologija zahteva da korisnici eksplicitno ukažu na ćelije koje su tačne, na osnovu čega sistem proširuje testiranost na ćelije sledbenike. Na primer: ćelije sa sličnim formulama se grupišu obeležavanjem istom bojom. Testirane ćelije se boje plavom, netestirane crvenom bojom. Čekboksovi se koriste za indicaciju mesta gde se donosi odluka. Ukoliko se ćelija označi kao tačna, u skladu sa ulazima, sve ćelije povezanog toka podataka koje utiču na tu ćeliju će biti označene kao pokrivene, povećavajući testiranost ćelija koje utiču na vrednost ispitane ćelije. Ukoliko je ćelija označena kao netačna, ćelije koje utiču na pogrešnu vrednost će biti obojene u crveno ili roze. Ćelija koja više puta utiče na vrednosti koje su pogrešne će biti tamno crvena i označena kao najverovatnije pogrešna.

Metodologija je kasnije razvijena kao integrisana familija tehnika za automatizovano generisanje test slučajeva u radovima: (Fisher et al., 2002), lokalizaciju grešaka (Ruthruff et al., 2003), (Ruthruff, Burnett, & Rothermel, 2005); generisanje test slučajeva, ponovnu upotrebe testova u: (Fisher II et al., 2002); mehanizama odgovora, kao i primenu tvrdnji u (Beckwith, Burnett, & Cook, 2002), (Burnett et al., 2003).

### **6.3.3.2 Automatsko generisanje test slučajeva**

Primenom osnovnog oblika WYSIWYT pristupa zahteva se da kreator spredšita odredi slučajeve za testiranje, na osnovu kojih će dobiti informaciju o tome koliko je dobro testiran spredšit. Kako bi se proces testiranja olakšao, u radovima: (Fisher et al., 2002) i (Fisher et al., 2006) predlažu se tehnike za automatizovano generisanje slučajeva koji će se testirati. U radovima su ispitane dve metode za generisanje vrednosti test slučajeva, „slučajna” i „lančana“. „Slučajna” metoda na slučajan način generiše vrednosti i proverava da li njihovo izvršenje izaziva upotrebu nevalidiranih elemenata. Druga, „lančana” metoda, ispituje nevalidirane parove ulaza i izlaza i pokušava da modifikuje ulazne vrednosti. Ukoliko je generisani ulaz za novi test slučaj uspešan, korisnik treba samo da validira izlaz, kako bi se dobio kompletan test slučaj. Na osnovu istraživanja prikazanog u (Fisher et al., 2006) dokazano je da je „lančana” metoda efikasnija od „slučajne“.

Alat *AutoTest*, predstavljen je u radu (Abraham & Erwig, 2006), implementira različite strategije za automatsku detekciju test slučajeva i programiranje ograničenja, u cilju određivanja vrednosti test slučajeva. Metoda određuje test slučajeve za sve dopustive parove ulaz-izlaz. Istraživanja su pokazala da *AutoTest* alat brže i efikasnije određuje vrednosti test slučajeva, u odnosu na prethodno opisane metode.

### **6.3.3.3 Testiranje na osnovu uslova**

Koncept upozorenja na osnovu uslova (*engl. Assertions*), koji se inicijalno vezuje za imperativne jezike, prilagođen je testiranju i obezbeđenju validnosti spredšitova u radu (Burnett et al., 2003). Pojam upozorenja na osnovu uslova koji se ovde naziva i “čuvari” (*engl. Guards*) se u domenu spredšitova odnosi na uslove o dopustivim vrednostima



ćelija u formi bulovskih izraza. Upozorenja na osnovu uslova određuju krajnji korisnici kroz odgovarajući alat. One se automatski proveravaju i delimično prenose kroz spredšit u pravcu toka podataka. Kada se detektuje konflikt između uslova i vrednosti ćelije ili između uslova definisanih od strane korisnika i prosleđenih, korisnik dobija vizuelnu informaciju. Evaluacija pristupa pokazala je da upozorenja na osnovu uslova pomažu korisnicima u detekciji različitih tipova grešaka, efikasnije i efektivnije od većine drugih pristupa.

#### **6.3.3.4 Razvoj spredšitova vođen testiranjem**

Princip razvoja programa vođen testiranjem potiče iz oblasti softverskog inženjerstva. Primenljivost ovog koncepta na razvoj spredšitova razmatra se u radu (McDaid, Rust, & Bishop, 2008). U skladu sa ovim principom, korisnici definišu test slučajeve koji se odnose na planiranu funkcionalnost spredšita, a onda unose ili menjaju postojeće formule, kako bi izvršili testiranje. Ovakva ponavljajuća i sistematična forma testiranja ima za cilj minimizaciju velikog broja grešaka u konačnoj verziji spredšita. Nakon evaluacije ovog pristupa, zaključeno je da se on lako primenjuje i da smanjuje procenat grešaka u spredšita, iako je vreme razvoja duže nego kod klasičnog načina kreiranja spredšita.

#### **6.3.4 Automatska detekcija i ispravka grešaka u spredšitovima**

Pristupi ove kategorije odnose se na scenarije procesa razvoja, u kojima kreatori unose podatke test slučajeve u spredšit i razmatraju neočekivane rezultate kalkulacija u jednoj ili više ćelija. Ovakve situacije nastaju tokom inicijalnog razvoja ili primenom neke od metodologija testiranja. Za spredšitove srednje veličine, skup mogućih kandidata ili ćelija, uzroka neočekivanog ponašanja, može biti veliki. Kada se spredšit sastoji od dužih lanaca povezanih proračuna, to uključuje preveliki skup mogućih uzroka nastanka greške, koji se ne može manuelno pretražiti u razumnom vremenu. Bez odgovarajućeg alata, korisnici bi morali da ispituju sve formule ili ćelije od kojih pogrešna vrednost zavisi, kao i proveriti njihovu tačnost. U skladu sa navedenim, cilj većine pristupa iz ove kategorije je podrška korisnicima pri lociranju pravog uzroka problema, u većini

slučajeva rangiranjem mogućih kandidata. Neki od pristupa su usmereni i na pokušaj da se odredi skup mogućih ispravki, odnosno izmena formula, kako bi se dobio željeni izlaz. Za razliku od pristupa inspekcije iz grupe statičkih analiza, osnova ovih proračuna obično obuhvata specifikaciju ulaznih i očekivanih izlaznih vrednosti ili test slučajeva.

#### **6.3.4.1 Rangiranje kandidata na osnovu tragova**

Jedan od prvih pristupa rangiranja kandidata, uzroka greške u spredšitu, koji ima određene sličnosti sa metodom *spectrum-based* lokalizacije grešaka za imperativne programe, predstavljen je u radu (Ruthruff et al., 2005). Metodom se predlaže primena koncepta odsecanja programa (*engl. Program Slicing*) na spredšitove, u cilju eliminisanja nedopustivih kandidata u inicijalnom koraku. Pristup koristi informacije o tačnosti i netačnosti vrednosti ćelije koje obezbeđuje korisnik, i u razmatranje uzima one ćelije koje teoretski utiču na netačnost vrednosti u ćeliji. Formula u ćeliji ima veće tendencije da bude netačna, ukoliko utiče na više ćelija koje su označene kao netačne, i obrnuto. Ukoliko ćelija utiče na netačnu vrednost, ali je “put” do nje “blokiran” ćelijom sa tačnom vrednošću, može se pretpostaviti da se greška nalazi negde između. Kasnija razrada ovog pristupa prikazana je u radu (Ruthruff, Burnett, & Rothermel, 2006). Ovi autori predstavljaju dve heuristike za detekciju grešaka i detaljniju analizu efikasnosti metode, koja se naziva tehnika blokiranja. Slična tehnika opisana je i u radu (Ayalew & Mittermeir, 2003), gde se za pogrešnu vrednost ćelije označavaju one ćelije koje imaju najveći uticaj na nju. U radu (Hofer et al., 2013) se takođe predlaže prilagođavanje *spectrum-based* metode lokalizacije grešaka iz oblasti tradicionalnog programiranja na spredšitove. Ovde se koristi formalizovaniji pristup sa koeficijentima sličnosti za proračun verovatnoće greške.

#### **6.3.4.2 Lokalizacija grešaka na osnovu ograničenja**

Pristupi lokalizacije grešaka na osnovu ograničenja prevode spredšitove u ograničenjima zasnovanu prezentaciju, na osnovu kojih se može izvršiti dodatno zaključivanje o mogućim razlozima neočekivanih vrednosti ćelija.

Jedan od prvih primera prevođenja spredšita u problem zadovoljenja ograničenja (*engl. Constraint Satisfaction Problem-CSP*, (Tsang, 1993)) predstavljen je u radu (Jannach & Engler, 2010). U skladu sa test slučajevima definisanim od strane korisnika i informacijama o neočekivanim vrednostima ćelija, primenjuje je princip dijagnosticiranja na osnovu modela (*engl. Model-Based Diagnosis-MBD*) za određivanje ćelije koja teoretski može biti pravi uzrok uočenog i neočekivanog izlaza proračuna. Ideja primene MBD principa originalno je razvijena za otkrivanje problema hardvera, a kasnije je prilagođena otkrivanju grešaka softvera, kao što se može videti u (Mateis et al., 2000), (Felfernig et al., 2004). Pristup prikazan u (Felfernig et al., 2004) prilagođen je i spredšitovima, i u obzir može da uzima više “pozitivnih” i “negativnih” test slučajeva i uključuje upozorenja na osnovu uslova.

Prvim evaluacijama pristupa, na relativno malim spredšitovima, pokazalo se da efikasnost u mnogome zavisi od korišćenih test slučajeva. Nakon toga, metod je unapređen i optimizovan, što je prikazano u (Jannach, Baharloo, & Williamson, 2013). Takođe, korišćeni su različiti algoritmi, u cilju poboljšanja skalabilnosti pristupa, kao što se može videti u (Jannach & Schmitz, 2014). Slični pristupi za određivanje uzroka neočekivanih vrednosti primenom CSP reprezentacije i MBD principa predstavljeni su i u radu (Abreu, Ribeiro, & Wotawa, 2012). Iako je ideja slična prikazanoj u radu (Jannach & Engler, 2010), realizacija je sprovedena na relativno drugačiji način. Umesto primene Algoritma ciljnog skupa (*engl. Hitting-Set Algorithm*, (Reiter, 1987)), rezonovanje tačnosti formula je izvršeno direktno predstavljanjem kroz ograničenja. Tačnost formule u ćeliji se predstavlja dodatnim bulovskim promenljivima. Još jedna specifičnost ovog pristupa je što se koristi samo jedan test slučaj. Istraživanja su pokazala da je pristup pogodan za određivanje uzroka grešaka u spredšitovima srednje veličine, koji sadrže realne brojeve, u realnom vremenu. U radu (Hofer et al., 2013) predlaže se kombinovana tehnika *spectrum-based* pristupa detekcije grešaka i MBSD (*engl. Model-Based Software Debugging*). Autori koriste koeficiente dobijene SFL (*engl. Spectrum-based Fault Localization*) tehnikom, kao inicijalne verovatnoće za MBD procedure. Iako se ovakav hibridni pristup pokazao pogodnim za rangiranje potencijalno pogrešnih ćelija, vreme realizacije je duže nego kod SFL metode.

### 6.3.4.3 Pristupi popravke

Pristupi popravke, ne samo da ukazuje na potencijalno problematične formule, već za cilj imaju i predlog moguće korekcije formule, tako da neočekivana vrednost bude izmenjena u očekivanu. Prva metoda automatskog određivanja sugestije za promenu formule, tzv. ciljno usmerena detekcija grešaka (*engl. Goal-Directed Debugging*) predstavljena je u (Abraham & Erwig, 2005). Korisnici daju svoju sugestiju o očekivanoj vrednosti za netačnu ćeliju, a metoda preračunava i predlaže izmenu, rekursivnom promenom individualnih formula, i širi unazad predloge promena za ćelije prethodnike koje sadrže formule. Predložene izmene koje obezbeđuju željeni rezultat se rangiraju heuristikom. Kasnije je metoda unapređena kombinovanjem sa pristupom automatskog generisanja test slučajeva, u cilju unapređenja rezultata debugiranja uključivanjem više test slučajeva i drugih informacija značajnih za testiranje, (Abraham & Erwig, 2008).

### 6.3.5 Pristupi razvoja spredšitova vođeni modelom

Za razliku od prethodno opisanih pristupa, pristupi razvoja zasnovani na modelu nemaju za primarni cilj podršku korisnicima u detektovanju grešaka, već poboljšanje strukture spredšita i prevenciju grešaka. Slično pristupima razvoja baziranim na modelu u oblasti razvoja softvera, osnovna ideja je uvođenje dodatnih nivoa apstrakcije u proces razvoja. Spredšit modeli na međunivou uvode apstraktniju konceptualizaciju problema i predstavljaju vezu između ideje koju je kreator imao i stvarne implementacije. Na ovaj način jaz između ideje i implementacije može biti prevaziđen.

Apstraktni spredšit modeli koriste se u dve faze procesa razvoja. Mogu se koristiti u formi generatora koda, pri čemu se delovi spredšita automatski generišu iz modela, smanjujući rizik nastanka mehaničkih grešaka. Druga faza se odnosi na korišćenje konceptualne strukture spredšita, na sličan način kao kod pristupa reverznog inženjeringa za razvoj softvera.

### 6.3.5.1 Deklarativni i objektno-orijentisani spreadšit modeli

Iako se spreadšitovi često razmatraju kao programi namenjeni tabelarnim proračunima, oni se mogu posmatrati iz perspektive modeliranja, kao što se navodi u ranim radovima (Isakowitz, Schocken, & Lucas, 1995). Prema ovom izvoru spreadšitovi se posmatraju sa fizičkog i logičkog aspekta. Fizički aspekt spreadšita odnosi se na formule i vrednosti u ćelijama, a logički na skup veza kojima se opisuje funkcionalnost spreadšita. U skladu sa ovim pristupom, logika programa predstavlja se “šemom”, dok se vrednosti ulaznih podataka posmatraju kao “podaci”. Primenom odgovarajućih alata, logika spreadšita može se automatski ekstrahovati iz spreadšita i predstaviti nekim drugim programskim jezikom. Ovakav sistem ima mogućnost sintetizovanja spreadšita iz specifikacije.

Slična objektno-orijentisana konceptualizacija spreadšit programa predstavljena je i u radovima (Paine, 1997) i (Paine, 2001). Prema *Model Master* pristupu spreadšitovi se specificiraju na deklarativan način, kao tekstualni programi. Oni se mogu proslediti kompajleru, koji generiše spreadšit iz specifikacije. Logika spreadšita je organizovana u kontekstu objektno-orijentisanog programiranja, u formi klasa koje obuhvataju attribute i logiku kalkulacije. Osim automatskog generisanja spreadšita iz modela, sistem omogućava ekstrakciju modela iz spreadšita. Za ekstrahovane modele može se proveriti postojanje grešaka ili se mogu koristiti kao standard za deljenje (razmenu) spreadšitova. Aspekti otkrivanja strukture spreadšita razmatrani su u radu (Paine, 2004).

Drugačiji pristup primene deklarativnog jezika za modelovanje predstavljen je u (Paine, 2005). *Excelsior* je sistem za razvoj spreadšitova, koji obuhvata programski jezik zasnovan na *Prolog*-u i dizajniran je za modularnu specifikaciju *MS Excel* spreadšita. Osim standardnih funkcionalnosti *Prolog*a, programski jezik se sastoji i od specifičnih konstrukcija i operatora za modelovanje logike spreadšita u modularnoj formi. Na osnovu takvog dizajna, raspored u spreadšitu može se odvojiti od njegove funkcionalnosti i kompajler može automatski generisati instance spreadšita iz specifikacija.

### 6.3.5.2 Spredšit šabloni

Određeni aspekti strukture modela spredšita mogu se ekstrahovati metodama baziranim na šablonima i vizualizaciji, kako je opisano u radovima: (Erwig et al., 2005), (Abraham et al., 2005), (Erwig et al., 2006). “Šablon” pristupa *Gencel* može se koristiti za specificiranje oblasti koje se ponavljaju u spredšitu. Dizajn šablona može se kreirati primenom vizualizacije, slične tipičnom korisničkom interfejsu sistema, kao što je *MS Excel*. Instance spredšita mogu se automatski generisati iz modela. Generisani spredšitovi mogu se dalje menjati na predefinisani način. Grupe oblasti koje se ponavljaju mogu se dodavati ili brisati, a vrednosti se mogu ažurirati. Pristup zasnovan na šablonima je u skladu sa idejom reverznog inženjeringa i automatske rekonstrukcije šablona iz spredšita, primenom određenih heuristika, kao što se može videti u (Abraham & Erwig, 2006). Evaluacija pristupa pokazala je da su automatski generisani šabloni značajno višeg kvaliteta nego ručno kreirani.

### 6.3.5.3 Objektno-orijentisani vizualni modeli

Kao nastavak i ekstenzija pristupa baziranog na šablonima, u cilju adresiranja šireg opsega tipova grešaka, u radu (Engels & Erwig, 2005) predstavlja se koncept *ClassSheets*, kao primena principa objektno-orijentisanog programiranja na spredšitove. Različite klase se vizualno razdvajaju pravougaonicima u bojama i predstavljaju semantički povezane ćelije. Za razliku od klasičnih šablona, klase nisu samo sintaksne strukture, već predstavljaju objekte, u smislu objektno-orijentisanog razvoja softvera. Pored vizualne notacije, ovaj pristup modeliranja obuhvata mehanizme za adresiranje modeliranih objekata, kroz simbolična imena klasa, umesto direktnog referenciranja ćelija. Prvi prototipovi alata zasnovanih na šablonima razvijeni su za automatsko generisanje spredšitova iz modela i ekstrakciju *ClassSheet* modela iz postojećih spredšitova.

Osnovni *ClassSheet* pristup je razmatran od strane više autora. Problem evolucije modela i kako ta unapređenja automatski transferisati u već generisani spredšit razmatran je u radu (Luckey, Erwig, & Engels, 2012). Isti problem evolucije modela i

koevolucije modela i spredšit instanci predstavljen je u (Cunha, Visser, Alves, & Saraiva, 2011) i (Cunha et al., 2011). U radu (Cunha et al., 2012) predlaže se prilaz koji se odnosi na suprotan pravac ažuriranja, odnosno automatski transfer promena u spredšitu na spredšit model. Dalja unapređenja pristupa *ClassSheet* obuhvataju primarne i sekundarne ključeve, generisanje *UML* dijagrama iz *ClassSheet* modela, u cilju validacije modela ili mehanizma za predstavljanje ograničenja ćelija (Cunha, Fernandes, Mendes, & Saraiva, 2012), (Cunha, Fernandes, & Saraiva, 2012). Evaluacija ovih ekstenzija osnovnog pristupa nije izvršena.

Pristup rekonstrukcije objektno-orijentisanog modela iz spredšita, baziran na vizualizaciji, predstavljen je u (Hermans, Pinzger, & Deursen, 2010). Ovaj pristup je zasnovan na biblioteci šablona, koji se pokušavaju locirati u spredšitu uz pomoć dvodimenzionalnog raščlanjivanja i algoritma za traženje šablona. Pronađeni šabloni se transformišu u *UML* dijagrame klase, koji se mogu koristiti za bolje razumevanje ili poboljšanje postojećeg spredšita. Rezultati evaluacije pristupa su se pokazali dobrim.

#### **6.3.5.4 Relacioni spredšit modeli**

Jedan od osnovnih principa spredšitova je organizacija podataka u tabelarnoj formi. Logičan način za definisanje apstraktnog modela strukture spredšita zasniva se na pristupima i principima dizajna relacionih baza podataka. U cilju postizanja boljeg kvaliteta spredšitova, bez grešaka, u radu (Cunha, Saraiva, & Visser, 2009) predlaže se ekstrakcija šema relacionih baza podataka iz spredšita, što bi pomoglo korisnicima da bolje razumeju spredšit i poboljšaju dizajn. Izlaz takvog procesa bi trebao biti modularni spredšit dizajn, bez redundanse podataka. Kasnije je ovaj pristup unapređen dodacima za automatsko kompletiranje vrednosti i bezbedno brisanje redova.

#### **6.3.6 Projektovanje i održavanje spredšitova**

Pristupi projektovanja i održavanja spredšitova obuhvataju alate koji za cilj imaju izbegavanje pogrešnih referenci, rukovanje izuzecima, dugoročno korišćenje spredšita (izmene, refaktorisanje, ponovna upotreba formula). Ovi alati imaju važnu ulogu u

obezbeđenju kvaliteta spredšita, s obzirom da je njihov cilj izbegavanje grešaka, na različite načine.

### 6.3.6.1 Upravljanje referenciranjem

Jedna od najvećih mana komercijalnih spredšit alata je ograničena sposobnost za obezbeđenje tačnosti reference ćelija, usled imena referenciranih ćelija koja ne nose semantičku informaciju o sadržaju. Korisnici često referenciraju pogrešnu ćeliju, ili koriste relativne umesto apsolutnih adresa, i obrnuto. Detekcija pogrešnih referenci je veoma zahtevan zadatak. Iako u *MS Excel*-u postoji opcija imenovanih oblasti, većina korisnika primenjuje apstraktna imena ćelija, koja se sastoje od oznaka redova i kolona. Prvi pristupi koji su se odnosili na ovaj problem *NO PumpG* (Wilde & Lewis, 1990), (Lewis, 1990) i *Action Graphics* (Hughes & Moshell, 1990), uključuju princip koordinatne mreže spredšita i inicijativu da korisnici dodeljuju eksplicitna imena ćelijama. WYSIWYC pristup, prema (Wilde, 1993), je alternativni pristup koji zadržava paradigmu koordinatne mreže i dodaje novi vizualni jezik za spredšitove. Ovakav pristup imao je za cilj da poboljša vizualizaciju strukture spredšita, proračuna i referenciranja, i na taj način uskladi vizualnu i logičku strukturu spredšita. Tako bi se olakšalo izbegavanje grešaka uzrokovano pogrešnom referencom ćelija. Nijedna od opisanih tehnika nije ispitana na konkretnim primerima, tako da njihova primenljivost nije dokazana.

### 6.3.6.2 Rukovanje „izuzecima“

Termin upravljanja „izuzecima“ se odnosi na kolekciju mehanizama za detekciju, signaliziranje i naknadno rukovanje izuzecima, kako se navodi u (Burnett, Agrawal, & Zee, 2000). „Izuzeci“ se definišu kao neočekivani događaji koji mogu zahtevati specijalnu obradu, prema (Sebesta, 1999). U (Burnett, Agrawal, & Zee, 2000) predstavlja se model greške vrednosti koji se može koristiti za jednostavno i adekvatno rukovanje greškama u spredšitu. U modelu grešaka vrednosti, poruke grešaka (npr. #DIV/0 u *MS Excel*-u) se prikazuju umesto očekivanih vrednosti. Prednosti pristupa primene vrednosti grešaka je što nema promena u generalnom modelu evaluacije, za



razliku od pristupa rukovanja greškama kod imperativnih programa, gde se izvršenje sekvenci obično menja. Spredšit kreatori ne moraju sticati dodatne veštine za rukovanje greškama, jer se mogu koristiti standardni operatori (npr. konstrukcija *if-then-else*).

### 6.3.6.3 Promene i evolucija spredšita

Spredšitovi su podložni menjaju, i prilikom menjanja često nastaju nove greške. *FormulaDataSleuth* alat, prikazan u (Bekenn & Hooper, 2008), namenjen je podršci kreatorima spredšita pri detekciji grešaka koje se javljaju promenom spredšita. Kada kreator specificira koje oblasti i ćelije treba da se prate ovim alatom, sistem automatski detektuje potencijalne probleme. Za definisane oblasti, na primer, mogu se detektovati prazne ćelije, ili ulazne vrednosti koje sadrže pogrešan tip podatka, ili prelaze van definisanog opsega dozvoljenih vrednosti. Ukoliko se formule u ćelijama koje se prate slučajno obrišu, ili prepisu drugim vrednostima, ili izmene u opsezima, vode ka pogrešnim referencama, alat će to detektovati.

Razumevanje evolucije spredšita i praćenje promena između različitih verzija spredšita je veoma važno, kada se spredšit ponovo koristi u različitim projektima. U radu (Chambers, Erwig, & Luckey, 2010b) predlaže se *SheetDiff* algoritam za detekciju i vizualizaciju određenih tipova netrivialnih razlika između dve verzije spredšita. Evaluacija pristupa je pokazala značajne prednosti u odnosu na postojeće pristupe.

Algoritam baziran na dinamičkom programiranju, pod imenom *RowColAlign*, namenjen detekciji razlika, koji rešava probleme „proždrljive“ *SheetDiff* procedure, prikazan je u (Harutyunyan et al., 2012). Umesto manuelne selekcije ili modifikacije spredšita, koristi se tehnika parametrizovanog generisanja test slučajeva, na sistematičniji način.

### 6.3.6.4 Refaktorisanje

Refaktorisanje (*engl. Refactoring*) se definiše kao proces promene interne strukture programa bez izmene funkcionalnosti, kako se navodi u (O’Beirne, 2010). Refaktorisanje utiče na kvalitet spredšitova na različite načine, na primer,

uprošćavanjem formula one postaju lakše za razumevanje, eliminisanjem duplikata kodova omogućava se lakše održavanje, sa manje grešaka. Refaktorisanje u kontekstu spredšitova često podrazumeva rearanžiranje redova i kolona, tj. transformisanje dizajna spredšita. Ručno izvođenje ovakve transformacije je vremenski zahtevno i sklono nastanku grešaka. U skladu sa ovom činjenicom, razvijeni su pristupi automatizacije zadatka poboljšanja kvaliteta održavanja i prevencije nastanka novih grešaka. U radu (Badame & Dig, 2012), identifikovano je sedam mera refaktorisanja za spredšitove i odgovarajući dodaci (*plug-in*) za *MS Excel*, pod nazivom *Ref-Book*. Ovi dodaci automatski detektuju lokacije koje zahtevaju refaktorisanje i omogućavaju izvođenje samog procesa. Pristup se pokazao kao dobar i pogodan za korišćenje.

Pristup namenjen složenim transformacija, primenom primera definisanih od strane korisnika, predstavljen je u radu (Harris & Gulwani, 2011). Pristup koji se zasniva na jeziku za opisivanje transformacija u tabelama, nazvan je *TableProg*, a algoritam *ProgFromEx*. Algoritam kao ulaze uzima male primere postojećeg spredšita i željene izlaze, na osnovu kojih automatski implementira transformacija. U evaluaciji ovog pristupa pokazalo se da je u nekim slučajevima neophodno obezbediti detaljniji primer nego što su oni koje korisnici obično koriste. Za jednostavnije slučajeve spredšitova pristup se pokazao uspešnim.

Alat *Excelsior*, pomenut u poglavlju 6.2.5.1 može se koristiti i za restrukturiranje spredšitova, kako se navodi u (Paine, Tek, & Williamson, 2006). *Excelsior* podržava operacije izmene pozicije i promene veličine opsega u spredšitu. Dodatno, korisnici mogu napraviti nekoliko verzija istog spredšita. Eksperimenti su pokazali da je pristup primenljiv.

### **6.3.6.5 Ponovna upotreba**

Ponovna upotreba postojećih i već validiranih softvera skraćuje vreme razvoja, smanjuje rizik nastanka grešaka i olakšava održavanja, kako se navodi u (Ye & Fischer, 2005). Sve navedeno odnosi se i na projekat razvoja spredšita. Individualni spredšitovi ili njihovi delovi, često se mogu koristiti u drugim projektima. Na mikronivou, čak i

individualne formule se često koriste više puta u spredšitu. Najčešći način primene ponovne upotrebe je jednostavno kopiranje formula. Ipak, menjanje originalnih formula, ne znači promenu njihovih kopija, zbog čega dolazi do čestog nastanka grešaka.

Problem ponovne upotrebe spredšit programa razmatra se u radovima: (Djang & Burnett, 1998), (Montigel, 2002). U radu (Djang & Burnett, 1998) ponovna upotreba se ostvaruje kroz koncept nasleđivanja, tipičan za objektno-orijentisano programiranje. Korisniku se omogućava da specificira zavisnosti između kopiranih ćelija spredšita u formi višestrukog i uzajamnog nasleđivanja, na nivou individualnih ćelija i viših nivoa.

Spredšit jezik *Wizcell* prikazan je u radu (Montigel, 2002). *Wizcell* ima za cilj pojednostavljenje ponovne upotrebe primenom semantičkih akcija kopiranja i premeštanja. Mogući izlazi ovakvih akcija su: kopija formule kao duplikat ili referenca ka originalnoj formuli, formule u kopiranim ćelijama pozivaju se na originalne ćelije ili se reference menjaju u skladu sa relativnim odstojanjima kopije i originala. *Wizcell* jezik omogućava korisniku da specificira željenu semantiku, na taj način smanjujući verovatnoću nastanka greške.

#### **6.4. Uporedni prikaz pristupa za detekciju i analizu grešaka u spredšitovima**

U kontekstu vizualizacije spredšitova predloženo je više pristupa, od strane različitih autora, koji bi omogućili predstavljanje određenih aspekata spredšita u vizualnoj formi, kao što je prikazano u poglavlju 6.2.1. Svrha vizualizacije je razumevanje spredšita sa aspekta ispitivanja anomalija i detekcije grešaka. Kao što se može zaključiti iz navedenih primera, samo nekoliko radova prikazuje sistematičnu evaluaciju predložene metode. U većini slučajeva validacija je ograničena na kvalitativne intervjuje ili male skupove učesnika ispitivanja prototipa. Zbog toga je primenljivost pristupa za krajnje korisnike nedovoljno potvrđena. Moguća evaluacija tehnika vizualizacije uključuje konstrukciju spredšita, inspekciju ili vežbe, ali i pristupe posmatranja zasnovane na razmišljanju „na glas“. Sa aspekta primenjenih alata, *MS Excel* uključuje veoma mali skup jednostavnih opcija za vizualizaciju, u cilju analize i detekcije grešaka u spredšitu.

Mogu se vizualizovati zavisnosti ćelija strelicama, ili pravougaonicima u boji koji se odnose na ćelije referencirane u formuli. Takođe, u slučaju da je narušeno neko od pravila pisanja formula ugrađenih u *MS Excel*, u uglu ćelije se pojavljuje zeleni trougao. Još jedna od opcija koja se nudi korisnicima *MS Excel*-a je podržana idejom „semantičkih“ imena promenljivih umesto adresa ćelija, čime je omogućeno malo povećanje razumljivosti spredšita.

Cilj pristupa statičke analize, prikazanih u poglavlju 6.2.2, je identifikacija formula ili strukturalnih karakteristika spredšita, koje se smatraju indikatorima potencijalnih problema. Tačnost ovih metoda zavisi od kvaliteta heuristike ili metrike za detekciju grešaka koje se koriste za definisanje sumnjivih delova spredšita. Uopšteno govoreći, alati za statičke analize predstavljaju familiju metoda za detekciju grešaka, koje se mogu naći kod komercijalnih alata. Sistemi zasnovani na jedinicama ili tipovima usmereni su na zaključivanje o dodatnim tipovima potencijalnih problema i mogu se smatrati jednostavnijim semantičkim pristupima. Takve tehnike zaključivanja imaju potencijal da identifikuju različite klase grešaka, ali postoji mogućnost da detektuju veliki broj „lažnih uzbuna“. Evaluacije pristupa su izvedene korišćenjem postojećih korpusa i spredšitova kreiranih od strane studenata i profesora. Potencijalna ograničenja primene EUSES<sup>4</sup> korpusa su nepoznavanja semantike formula za koje smatra da su pogrešne, kao rezultat primene određene tehnike. Zbog toga se ne može sa sigurnošću tvrditi da je formula zaista netačna, odnosno da je tehnika uspešno primenjena. Sa aspekta krajnjeg korisnika, rezultati statičkih analiza su jednostavni za razumevanje.

Jedan od najvećih problema programa krajnjih korisnika je što, obično, nisu rigorozno testirani. Iz tog razloga bi alati za podršku procesa razvoja, prikazani u poglavlju 6.2.3, trebali da omoguće kreiranje spredšitova sa što manje grešaka. Ipak, komercijalni spredšit alati imaju ograničene funkcionalnosti u ovom kontekstu. *MS Excel* obezbeđuje samo osnovne alate za validaciju, koji se odnose na definisanje tipova i vrednosti individualnih ćelija. Problem povezan sa test alatima odnosi se na dizajn korisničkog interfejsa pogodnog za korisnike. Efikasnost generisanja test slučajeva i njihove adekvatnosti još uvek nije dovoljno ispitana. Još jedno od nedovoljno obrađenih pitanja

---

<sup>4</sup> <http://eusesconsortium.org/resources.php>

odnosi se na ograničenu svesnost krajnjih korisnika o važnosti testiranja i preterano poverenje u tačnost programa. Potrebno je analizirati na koji način spredšit kreatori zaista testiraju spredšitove i da li bi mogli, barem delimično, da usvoje principe razvoja zasnovane na testiranju. Primena tehnike testiranja mutacijom razmatrana je u (Abraham & Erwig, 2009). Testiranje mutacijom sastoji se od uvođenja malih promena u program i provere koliko takvih mutanata se može eliminisati određenim skupom testova. Mutacije se mogu koristiti za testiranje manuelno i automatski kreiranih test slučajeva. U širem kontekstu detekcije i eliminacije grešaka, mogu se koristiti za evaluaciju pristupa debugiranja.

Efikasnost tehnika debugiranja koje se zasnivaju automatskoj lokalizaciji grešaka i popravci, prikazane u poglavlju 6.2.4, ocenjene su korišćenjem protokola evaluacije kod kojih su greške veštački ugrađene u spredšit koji se razmatra. Evaluacija je pokazala da predložene tehnike vode ka uspešnom rangiranju kandidata za greške ili popravku grešaka, ili je moguće određivanje skupa potencijalnih uzroka grešaka. Ipak, spredšitovi korišćeni za eksperimente su uglavnom veoma mali, tako da je skalabilnost pristupa nedokazana. Takođe, pristupi zasnovani na ograničenjima često su ograničeni na male dimenzije ili celobrojne vrednosti. Za sve pristupe ovog tipa, spredšit kreatori moraju da odrede željene izlazne vrednosti ili ćelije koje proizvode tačan/netačan izlaz. Samim tim, pristupi podrazumevaju da korisnik unapred zna koja vrednost rezultata treba da se dobije. To se javlja kao problem primene ovog pristupa, jer kreatori često nemaju takve informacije ili zadaju pogrešne vrednosti.

Modelom vođeni pristupi razvoja, prikazani u poglavlju 6.2.5, uvode dodatne sintaksne ili semantičke nivoe apstrakcije u proces razvoja spredšita. Ovi dodatni mehanizmi i konceptualizacija treba da pomognu smanjenju jaza između gotovog spredšit modela i realnog problema. Takođe, ovi pristupi imaju za cilj i podizanje nivoa kvaliteta dizajna spredšit modela, smanjenje grešaka i lakše održavanje. Primena pristupa zasnovanih na razvoju modelu donosi i izazove koji se mogu naći kod standardnih procesa razvoja softvera, kao što je problem koevolucije modela i programa. Dizajn i jezik za modelovanje, takođe, su značajni, jer se često mora praviti kompromis između ekspresivnosti i razumljivosti. Dodatno, kod spredšitova se javlja problem kreatora koji

uglavnom nisu stručnjaci u oblasti informacionih tehnologija. Kod njih se javlja problem razumevanja alata i dugoročne prednosti apstrakcije i struktuiranosti. Zapravo, jedan od glavnih razloga popularnosti spredšitova je činjenica da ne zahtevaju struktuirani, formalni proces razvoja, već je razvoj *ad-hoc*, interaktivan proces inkrementalnog razvoja prototipa. Sa istraživačkog aspekta, većina pristupa nije evaluirana ili je izvršena preliminarna evaluacija. Stoga, ovi pristupi moraju biti sistematičnije evaluirani, kao i ispitana njihova primenljivost i prihvatljivost za korisnike.

Pristupi projektovanja i održavanja, prikazani u 6.2.6, podrazumevaju primenu postojećih tehnika softverskog inženjerstva na spredšitove. Ipak, neophodno je zadržati karakteristike razvoja spredšita, kao i razumljivost za krajnje korisnike. Nasuprot tome, neki pristupi zahtevaju određeno poznavanja netrivialnih programerskih koncepata. Pošto krajnji korisnici obično nisu profesionalni programeri, primenljivost ovih pristupa u praksi je diskutabilna.

Na osnovu prikazanih pristupa može se zaključiti da su oni evaluirani na različite načine. To se može povezati sa metodama koje potiču iz podoblasti računarskih nauka i informacionih sistema, i prilagođene su spredšitovima. Svaka od oblasti podrazumeva standarde i protokole za evaluaciju. Za evaluaciju su korišćena ispitivanja sa korisnicima u laboratorijskim uslovima, empirijske studije bez korisnika ili teorijska analiza. U nekim slučajevima, efikasnost pristupa uopšte nije ispitana. U novijim radovima: (Jannach & Schmitz, 2014), (Abraham, Erwig, & Andrew, 2007) i drugi, može se zapaziti primena simulacije u laboratorijskim uslovima, kao najčešće korišćen način evaluacije pristupa. Za evaluaciju su često korišćeni slučajno izabrani spredšitovi iz EUSES korpusa, dok su spredšitovi iz prakse, koji se odnose na realne probleme, značajno ređe evaluirani. Kriterijum za izbor spredšita za analizu uglavnom nije definisan ili ne postoji. U cilju izvođenja empirijske evaluacije, greške su veštački ugrađivane u spredšit, bez definisanih pravila o tipu i lokaciji greške. Uz sve navedeno, eksperimenti uglavnom nisu detaljno ili nisu uopšte dokumentovani. Takođe, skalabilnost pristupa često nije analizirana.

## **7. Pristup za detekciju i analizu grešaka u dinamičkim diskretnim modelima upravljanja zalihama razvijenim u spredšitovima**

Nakon prikaza šest kategorija pristupa automatizovanog obezbeđenja kvaliteta spredšitova i njihovih podkategorija, u najvećem broju slučajeva zasnovanih na primeni metoda i alata iz oblasti softverskog inženjerstva, u ovom poglavlju se predstavlja novi pristup za detekciju i analizu spredšit grešaka zasnovan na karakteristikama problema i načinu modeliranja. Pristup se razmatra u kontekstu detekcije i analize grešaka u modelu ekonomske količine naručivanja zaliha za više proizvoda, sa ograničenim prostorom skladištenja, razvijenom u spredšitu.

### **7.1. Osnovni koncept pristupa za detekciju i analizu grešaka u DDMUZRS**

Problemi upravljanja zalihama modelirani kao dinamički diskretni procesi, implementirani u spredšitovima, podložni su nastanku grešaka usled kompleksne strukture samog modela, ali i procesa razvoja, koji uključuje prevođenje matematičkih relacija u spredšit formule i funkcije, kao što je već navedeno u prethodnim poglavljima. Takođe, greške se vrlo često javljaju i u toku korišćenja, zbog nerazumevanja modela od strane krajnjeg korisnika ili sasvim slučajno. Detekcija i analiza grešaka predstavljaju zahtevne aktivnosti, s obzirom da skup potencijalnih grešaka i njihovih uzroka može biti veoma širok.

U prethodnom poglavlju opisani su postojeći pristupi za detekciju i analizu grešaka u spredšit modelima. Ovi pristupi su razmatrani sa aspekta primenljivosti, vremena odziva i skalabilnosti. Za njihovu evaluaciju korišćeni su veštački kreirani spredšitovi ili oni iz EUSES baze, različite veličine i kompleksnosti. Jednostavnost prikazanih primera omogućava lakše predstavljanje, razumevanje i implementaciju pristupa u različitim oblastima primene spredšitova. Međutim, prikazani pristupi u većini slučajeva nisu primenjeni na inženjerske, poslovne ili modele problema iz prakse. Opisani pristupi su namenjeni detekciji ili analizi spredšit grešaka u opštem slučaju, i zasnovani na primeni koncepata razvijenih u drugim oblastima. Primenljivost ovih pristupa na kompleksne probleme, u najvećem broju slučajeva, nije poznata.

U ovom poglavlju, predstavlja se novi pristup za detekciju i analizu grešaka u DDMUZRS, zasnovan na karakteristikama problema i načinu modeliranja. Pristup je prikazan UML 2.0 dijagramom aktivnosti (Slika 1). Svaka aktivnost pristupa opisana je zasebnim algoritmom. Za označavanje iteracije koristi se klauzula *For-Each-Next*.

Oznake koje će biti korišćene su:

$m$ - ukupni broj proizvoda ( $i=1, \dots, m$ );

$T$  - konačni vremenski horizont ( $t=1, \dots, T$ );

$D_i$  - ukupna deterministička tražnja ( $\sum_{t=1}^m Y_t^i = D_i$ );

$Q_i$  - količina proizvoda  $i$  za naručivanje ( $i=1, \dots, m$ );

$u_i$  - broj narudžbina za proizvod  $i$  ( $u_i \in 1, 2, \dots, T$ ), ( $i=1, \dots, m$ );

$G$  - raspoloživi skladišni prostor;

$P_i$  - skladišni prostor koji zauzima jedinica proizvoda  $i$ , ( $i=1, \dots, m$ );

$X_{t,i}$  - promenljiva stanja ( $t=1, \dots, T$ ;  $i=1, \dots, m$ );

(ukupna količina proizvoda  $i$  koja ostaje na zalihama na kraju vremenskog perioda  $t$ );

$Y_{t,i}$  - regulator protoka ( $t=1, \dots, T$ ;  $i=1, \dots, m$ );

(količina proizvoda  $i$  naručena na početku perioda  $t$ );

$J_{t,i}$  - ciljna funkcija ( $t=1, \dots, T$ ;  $i=1, \dots, m$ ) definisana relacijom (5.21);

$\sum_{i=1}^m J_{t,i}$  – ukupna ciljna funkcija  $t=1, \dots, T$  definisana relacijom 5.21 ;

*LMG* - Lista ćelija sa mogućom greškom;

*LM* - Matrica pikova;

*LPPF* - Lista ćelija sa potencijalno pogrešnom formulom;

*Ispravljene\_formule* - pomoćna promenljiva tipa *Boolean*;

*TS* - test slučaj;

$gdg_i$  - globalna donja granica ciljne funkcije za proizvod  $i$  ( $i=1, \dots, m$ );

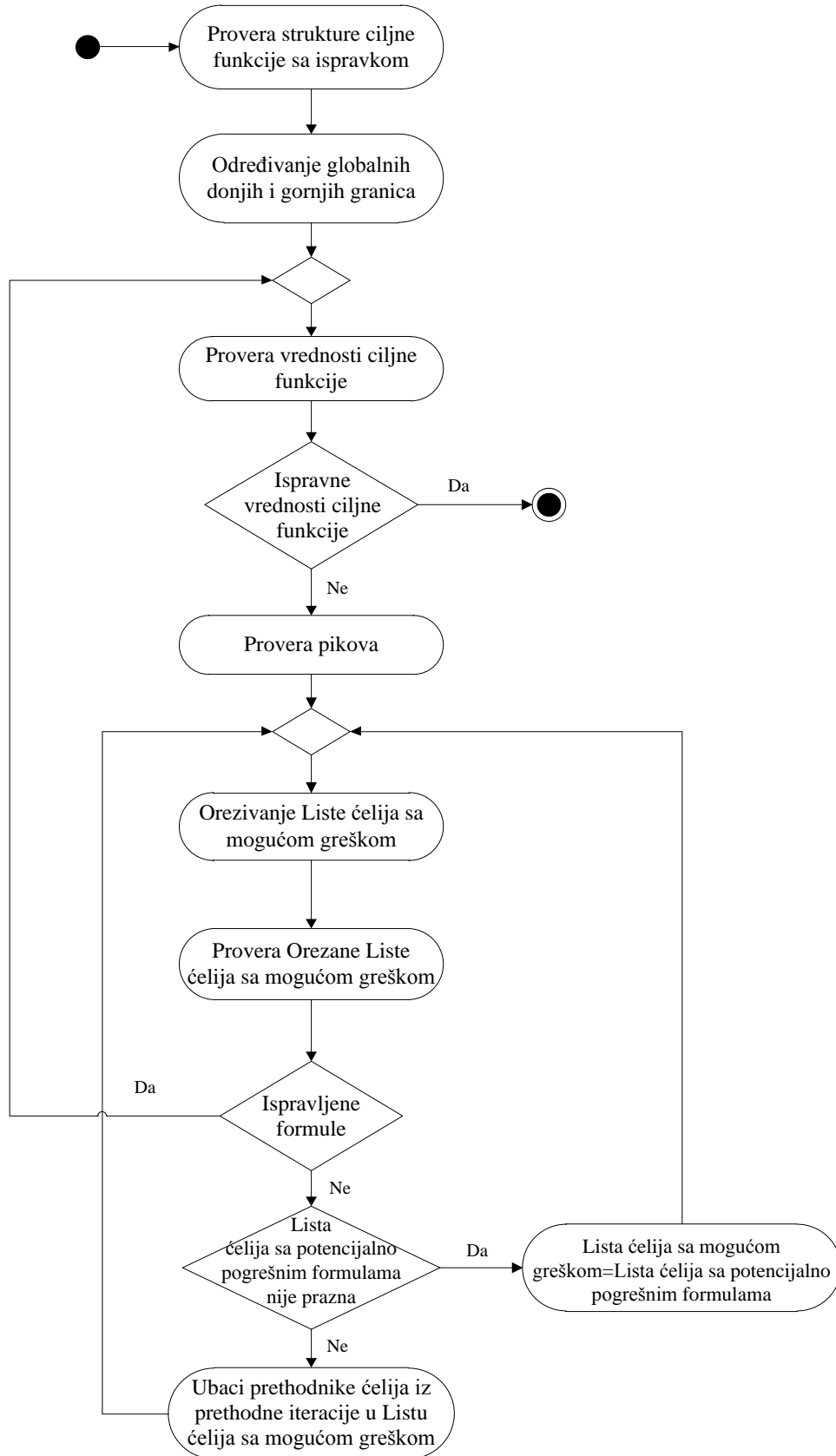
$ggg_i$  - globalna gornja granica ciljne funkcije za proizvod  $i$  ( $i=1, \dots, m$ );

$sdg_i(TS)$  - specifična donja granica ciljne funkcije za proizvod  $i$  ( $i=1, \dots, m$ ) za test slučaj *TS*;

$sgg_i(TS)$ -specifična gornja granica ciljne funkcije za proizvod  $i$  ( $i=1, \dots, m$ ) za test slučaj *TS*;

$pik_{t,i}$ -odskok zaliha proizvoda  $i$  u vremenskom periodu  $t$  kada je narudžbina realizovana ( $t=1, \dots, T$ ;  $i=1, \dots, m$ );





**Slika 7-1.** Pristup za detekciju i analizu grešaka u DDMUZRS predstavljen UML 2.0 dijagramom aktivnosti

## 7.2. Algoritmi pristupa za detekciju i analizu grešaka u DDMUZRS

Pristup za detekciju i analizu grešaka u DDMUZRS, koji se razvija u ovoj disertaciji, započinje proverom strukture formula, koje odgovaraju ciljnoj funkciji  $J_{t,i}$  za svaki proizvod  $i$ ,  $i=1, \dots, m$  i ukupnoj ciljnoj funkciji  $\sum_{i=1}^m J_{t,i}$ , u svakom periodu  $t$  horizonta posmatranja  $T$ . Preduslov za realizaciju ovog koraka, predstavljenog Algoritmom 1, je poznata ispravna struktura formula pojedinačnih ciljnih funkcija  $J_{t,i}$  i ukupne ciljne funkcije  $\sum_{i=1}^m J_{t,i}$ , definisanih u skladu sa relacijom (5.21).

Ovaj jednostavan korak ima za cilj eliminisanje grešaka iz strukture formula ciljnih funkcija, pre dalje provere modela. Provera strukture formula ciljnih funkcija je zasnovana na *RICI* stilu referenciranja u spredšitovima. Adresiranje tipa *RICI* podrazumeva određivanje relativnih odstojanja elemenata formule od aktivne ćelije u kojoj se formula nalazi. Ukoliko se detektuje odstupanje u strukturi formule, neophodno je sprovesti ispravke iste, pre nego što se nastavi ispitivanje modela. Ispravka formule bazira se na poznatoj ispravnoj strukturi ciljne funkcije  $J_{t,i}$  i ukupne ciljne funkcije  $\sum_{i=1}^m J_{t,i}$ . Sve ispravke, ukoliko postoje, se prikazuju u žurnalu (listi ustanovljenih grešaka sa utvrđenim uzrocima). Žurnal se sastoji iz opisa grešaka. Struktura opisa greške sadrži adresu ćelije sa greškom, formulu u ćeliji i ispravku formule. Na osnovu opisa greške jednostavno je odrediti uzrok greške, odnosno očitati iz spredšit modela koji je element i vremenski period u kome je greška nastala, kao i tip greške. Tip greške se određuje poređenjem neispravne i ispravljene strukture formule.

Provera strukture formula koje odgovaraju ciljnoj funkciji predstavlja prvi korak detekcije grešaka, ali njihova ispravnost ne podrazumeva tačnost celog modela. Greške mogu postojati kod pomoćnih promenljivih, regulatora protoka ili promenljivih stanja.

Kao posledica povezanosti elemenata modela, postojanje grešaka se može uočiti na osnovu odstupanja vrednosti ciljne funkcije od njene očekivane vrednosti. Kako ciljna funkcija predstavlja kumulativnu vrednost ciljne funkcije u prethodnom vremenskom

periodu i vrednosti ciljne funkcije u posmatranom vremenskom periodu, neophodno je odrediti vremenski period u kome se greška pojavila. Osnovni problem koji se ovde javlja je činjenica da kreator ili korisnik modela u realnoj situaciji ne mogu znati tačne vrednosti izlaznih promenljivih za poslednji vremenski period posmatranja ( $T$ ), kao ni međurezultate prethodnih perioda ( $t=1, \dots, T$ ). Svaka kombinacija promenljivih upravljanja ( $u_i$ ) rezultuje različitim vrednostima ciljne funkcije za  $t$  vremenskih perioda ( $t=1, \dots, T$ ) horizonta posmatranja, što dodatno otežava definisanje tačnih vrednosti rezultata. U cilju detekcije grešaka, u skladu sa pristupom koji se razvija u ovom radu, koriste se donje i gornje granice ciljne funkcije (Algoritam 2 i Algoritam 3) i pravila zasnovana na karakteristikama modela (Algoritam 4).

*Ulaz:* spredšit model, broj proizvoda ( $i=1, \dots, m$ ), broj vremenskih perioda

( $t=1, \dots, T$ ), poznata ispravna struktura formula  $J_{t,i}$  i  $\sum_{i=1}^m J_{t,i}$

*Izlaz:* žurnal, ispravljen model

For each  $i$

For each  $t$

If struktura formule  $J_{t,i} \neq$  ispravna struktura formule  $J_{t,i}$   
 ispravka strukture formule  
 ubaci opis greške za  $J_{t,i}$  u žurnal

End if

Next  $t$

Next  $i$

For each  $t$

If struktura formule  $\sum_{i=1}^m J_{t,i} \neq$  ispravna struktura formule  $\sum_{i=1}^m J_{t,i}$

ispravka strukture formule

ubaci opis greške za  $\sum_{i=1}^m J_{t,i}$  u žurnal

End if

Next  $t$

End

### Algoritam 1. Provera strukture ciljne funkcije sa ispravkom

Donja granica određuje se za vremenski period  $t=1$ , a gornja za  $t=T$ . Ukoliko bi se svi proizvodi naručivali u svakom periodu ( $t$ ) horizonta posmatranja, troškovi za prvi dan bi bili najniži. Nasuprot tome, svakodnevno naručivanje prouzrokuje najviše troškove u

poslednjem vremenskom periodu ( $T$ ), s obzirom da ukupni troškovi zaliha predstavljaju sumu troškova za svaki proizvod ( $i$ ) u svakom vremenskom periodu ( $t$ ). Kako bi ovo tvrđenje bilo istinito, uvodi se pretpostavka da su troškovi držanja zaliha značajno niži od troškova naručivanja.

Naručivanje svih proizvoda na početku horizonta posmatranja inicira najviše troškove prvog dana i najniže ukupne troškove. Izuzetak od ovog pravila se može javiti ako je ukupni broj perioda posmatranja veoma veliki, pa troškovi držanja zaliha značajno rastu. Ukoliko je horizont posmatranja veoma dugačak (duži od godinu dana), upravljačke promenljive koje će se koristiti za definisanje gornjih granica se mogu eksperimentalno odrediti. Upravljačke promenljive ( $u_i$ ) odnose se na broj naručivanja po svakom proizvodu  $i$ . Prilikom definisanja maksimalne gornje i minimalne donje granice ograničenje skladišnog prostora (5.20) se zanemaruje. Relaksacija ovog ograničenja garantuje da nijedna vrednost ciljne funkcije dobijena za dopustivi test slučaj, za koji važi ograničenje skladišnog prostora, ne može biti manja, odnosno veća od granica dobijenih na opisani način. Određivanje gornjih i donjih granica predstavljeno je Algoritmom 2 i Algoritmom 3.

*Ulaz:* spredšit model, broj proizvoda ( $i=1, \dots, m$ ), broj vremenskih perioda ( $t=1, \dots, T$ ),  $TS (u=T)$

*Izlaz:*  $\min gdg_i$  i  $\max ggg_i$

For each  $i$   
      $\min gdg_i = J_{1, i}$   
      $\max ggg_i = J_{T, i}$

Next  $i$   
 End

**Algoritam 2.** Određivanje minimalne donje i maksimalne gornje granice ciljne funkcije

Donja ( $gdg_i$ ) i gornja granica ( $ggg_i$ ), određene na opisani način, predstavljaju globalne granice i određuju dopustivu oblast svih specifičnih granica. Specifične granice su vrednosti ciljne funkcije koje ona dobija za vremenske periode  $t=1$ ,  $t=T$  i konkretni test slučaj ( $TS$ ). Test slučajevi se kreiraju kao kombinacije dopustivih vrednosti promenljivih upravljanja ( $u_i$ ). Odstupanje specifične donje ( $sdg_i (TS)$ ) i specifične gornje

granice ( $sgg_i$ ) za konkretni test slučaj od globalnih granica ukazuje na postojanje greške. Postojanje greške može se uočiti i ukoliko vrednost ciljne funkcije u vremenskom periodu  $t$ , za  $1 < t < T$ , nije u opsegu definisanom specifičnim granicama.

*Ulaz:* spredšit model, broj proizvoda ( $i=1, \dots, m$ ), broj vremenskih perioda ( $t=1, \dots, T$ ),  
 $TS (u=1)$

*Izlaz:* max  $gdg_i$  i min  $ggg_i$

For each  $i$   
     max  $gdg_i = J_{1, i}$   
     min  $ggg_i = J_{T, i}$

Next  $i$   
 End

**Algoritam 3.** Određivanje maksimalne donje i minimalne gornje granice ciljne funkcije

Dodatna ograničenja, zasnovana na osobinama ciljne funkcije, mogu se primeniti u procesu detekcije grešaka. Ciljna funkcija je kumulativna vrednost, koja se sastoji od vrednosti ciljne funkcije u prethodnom vremenskom periodu i njene vrednosti u periodu posmatranja. U skladu sa ovom karakteristikom, vrednost ciljne funkcije u vremenskom periodu  $t$  mora da bude niža od vrednosti ciljne funkcije u narednom vremenskom periodu  $t+1$ .

Svaki put kada je neko od opisanih ograničenja narušeno i na taj način greška detektovana, odgovarajuća ćelija postaje kandidat za dalju analizu, odnosno puni se Lista ćelija sa mogućom greškom (*LMG*). *LMG* odnosi se na ćelije u kojima je greška detektovana i njihove direktne prethodnike. Poseban slučaj predstavlja greška zamene formule konstantom. U tom slučaju se ne analiziraju prethodnici, već je uzrok greške odmah otkriven. Vrednosti svih elementa modela, osim ulaznih podataka, test slučaja i promenljivih u nultom vremenskom periodu, izračunavaju se formulama. Ulazne vrednosti modela se smatraju tačnim i ne razmatraju u kontekstu grešaka. Ulazni podaci uključuju ukupnu tražnju, troškove naručivanja, jedinične troškove nabavke proizvoda, troškove držanja zaliha i raspoloživi prostor za skladištenje.

```

    Ulaz: spredšit model, broj proizvoda ( $i=1, \dots, m$ ), broj vremenskih perioda ( $t=1, \dots, T$ ),
            $gdg_i, ggg_i, TS$ 
    Izlaz:  $LMG, sdg_i(TS), sgg_i(TS)$ 
    For each  $i$ 
        If  $J_{1,i} \geq \min gdg_i$  and  $J_{1,i} \leq \max gdg_i$ 
             $sdg_i(TS) = J_{1,i}$ 
        Else
            ubaci ćeliju  $J_{1,i}$  u  $LMG$ 
        End if
        If  $J_{T,i} \geq \min ggg_i$  and  $J_{T,i} \leq \max ggg_i$ 
             $sgg_i(TS) = J_{T,i}$ 
        Else
            ubaci ćeliju  $J_{T,i}$  u  $LMG$ 
        End if
    Next  $i$ 
    For each  $i$ 
        If exist  $sdg_i(TS)$  and exist  $sgg_i(TS)$  Then
            For  $t = 2$  to  $T-1$ 
                If  $sdg_i(TS) > J_{t,i}$  or  $J_{t,i} > sgg_i(TS)$ 
                    ubaci ćeliju  $J_{t,i}$  u  $LMG$ 
                End if
            Next  $t$ 
        End if
    Next  $i$ 
    For each  $i$ 
        For  $t=1$  to  $T-1$ 
            If  $J_{t,i} > J_{t+1,i}$  and ćelija  $J_{t,i}$  not in  $LMG$ 
                ubaci ćeliju  $J_{t,i}$  u  $LMG$ 
            End if
        Next  $t$ 
    Next  $i$ 
    End
    
```

#### Algoritam 4. Provera vrednosti ciljne funkcije

Pred globalnih i specifičnih granica, dodatna ograničenja mogu se odrediti u odnosu na pikove (Algoritam 5). Pik ( $pik_{t,i}$ ) predstavlja odskok zaliha u vremenskom periodu  $t$ , kada je narudžbina za proizvod  $i$  realizovana. Količnik naručenih proizvoda ( $D_i/u_i$ ) i dnevne tražnje ( $D_i/T_i$ ) mora da bude jednak broju dana između dve uzastopne narudžbine. Odstupanje između ovih vrednosti ukazuje na grešku u oblasti promenljivih stanja ( $X_{t,i}$ ). Naručene količine proizvoda  $i$  ( $D_i/u_i$ ) su iste u svim vremenskim periodima u kojima pikovi postoje. Ulazne količine koje se pojavljuju u vremenskim periodima

kada pikovi postoje ulaze u Matricu pikova ( $MP$ ). Razlika u vrednosti ulaznih promenljivih podrazumeva grešku u oblasti regulatora protoka ( $Y_{t,i}$ ). Navedena pravila slede iz modela prikazanog u poglavlju 5.1, relacije (5.17) do (5.19).

```

Ulaz: spredšit model, naručena količina ( $D_i/u_i$ ), dnevna tražnja ( $D_i/T_i$ ), broj
dana između uzastopnih narudžbina, broj proizvoda ( $i=1, \dots, m$ ), broj
vremenskih perioda ( $t=1, \dots, T$ )

Izlaz:  $LMG, MP$ 

For each  $i$ 
   $k=0$ 
  For each  $t$ 
    If  $pik_{t,i} > 0$  Then
       $k=k+1$ 
       $pik_{k,i} = pik_{t,i}$ 
      ubaci  $pik_{k,i}$  u  $MP$ 
       $z_{k,i} = t$ 
    End if
  Next  $t$ 
 $k=1$ 
  For each  $pik_{k,i}$  in  $MP$ 
    //  $z_{k+1,i} - z_{k,i}$  je broj dana između uzastopnih narudžbina proizvoda  $i$ 
    //  $D_i/u_i / D_i/T_i = T_i/u_i$ 
    If  $z_{k+1,i} - z_{k,i} \neq T_i/u_i$  Then
      ubaci ćeliju( $z_{k,i}, i$ ) u  $LMG$ 
    End if
    If  $pik_{k,i} \neq pik_{k+1,i}$  Then
      ubaci ćeliju ( $z_{k,i}, i$ ) u  $LMG$ 
    End if
  Next  $pik_{k,i}$ 
Next  $i$ 
    
```

#### Algoritam 5. Provera pikova

Kada se greška u modelu detektuje, sledeći korak predstavlja određivanje uzroka greške. Određivanje uzroka greške podrazumeva utvrđivanje elementa i vremenskog perioda u kome je greška nastala, adresu ćelije, kao i tip greške. Najjednostavniji, ali i najređi, slučaj je da se uzrok greške otkriva u ćeliji u kojoj je greška detektovana. Značajno češći slučaj je da se uzrok greške nalazi u nekoj od ćelija prethodnika i posledično se manifestuje u ćeliji u kojoj je greška detektovana. Dodatno ukoliko se greška detektuje u ćelijama koje se odnose na ciljne funkcije, za koje je ispravnost

strukture formula ispitana Algoritmom 1, jasno je da je uzrok greške u nekoj od ćelija prethodnika. Kompleksne zavisnosti elemenata DDMUZRS impliciraju širok skup prethodnika za ćelije u kojima je greška detektovana. Stablo prethodnika mora biti orezano, kako bi se uzrok nastanka greške odredio u prihvatljivom vremenu. Pravila orezivanja i procedura provere potencijalno pogrešnih ćelija prikazani su Algoritmom 6 i Algoritmom 7.

Pravila orezivanja stabla ćelija u kojima je detektovana greška i njihovih prethodnika su sledeća:

- Ukoliko ćelija iz *LMG* predstavlja prethodnika neke druge ćelije iz ove liste i obe ćelije se odnose na promenljive istog tipa (npr. promenljive stanja  $X_{t,i}$ ), onda druga detektovana ćelija i njeni prethodnici nisu kandidati za ispitivanje u aktuelnoj iteraciji.
- Ukoliko ćelija iz *LMG* predstavlja prethodnika neke druge ćelije iz iste liste, ali se one ne odnose na promenljive istog tipa, onda prethodnici ćelije nisu kandidati za ispitivanje u aktuelnoj iteraciji.

Prvo pravilo zasniva se na karakteristikama načina modeliranja u skladu sa konceptom diskretnog objekta upravljanja. Ciljna funkcija i promenljive stanja direktno su zavisne od svojih vrednosti u prethodnim vremenskim periodima. U slučaju ispitivanja promenljivih istog tipa, prioritet ima promenljiva koja se odnosi na raniji vremenski period. Prioritet ispitivanja je posledica uticaja promenljivih iz ranijih vremenskih perioda na promenljive istog tipa u kasnijim vremenskim periodima.

Drugo pravilo se odnosi na prioritet ispitivanja ćelija koje ne predstavljaju promenljive istog tipa. Ukoliko je potencijalno pogrešna ćelija prethodnik druge potencijalno pogrešne ćelije, prva se ispituje roditeljska ćelija. Prioritet roditeljske ćelije određuje hijerarhijska struktura, u kojoj viši nivoi imaju prednost u procesu provere. Ovo pravilo je u skladu sa algoritmom pretrage po širini (engl. *breadth-first search*).



```

Ulaz: LMG
Izlaz: Orezana LMG
For each c in LMG
  If c is not type of (ulazni podatak) Then
    Lista prethodnika = predecessors (c)
    For each p in Lista prethodnika
      If p is type of (ulazni podatak) Then
        izbaci p iz LMG
      End if
    Next p
  Else
    izbaci c iz LMG
  End If
Next c
// c1 i c2 su ćelije koje se ispituju
For each c1 in LMG
  Lista sledbenika = successors (c1)
  For each c2 in Lista sledbenika
    If c1 and c2 promenljive istog tipa Then
      izbaci prethodnike c2 iz LMG
    Else
      izbaci prethodnike c1 iz LMG
    End if
  Next c2
Next c1
    
```

### Algoritam 6. Orezivanje Liste ćelija sa mogućom greškom

Kada se stablo ćelija sa mogućom greškom oreže (*LMG*), sledeći korak podrazumeva analizu preostalih elemenata za ispitivanje (Algoritam 7). Analiza se realizuje poređenjem strukture formula u ćelijama iz orezane *LMG* sa strukturom formula promenljivih istog tipa u prethodnoj i narednoj ćeliji. Prethodna i naredna ćelija se odnose na promenljive u prethodnom ( $t-1$ ) i narednom ( $t+1$ ) vremenskom periodu. Ako se struktura formule razlikuje od struktura formula u prethodnoj i narednoj ćeliji, a u strukturi formula prethodne i naredne ćelije nema razlike, onda ćelija koja se analizira verovatno predstavlja uzrok greške u modelu. Struktura formule u ćeliji koja se analizira se menja strukturom formula iz ćelije iznad i/ili ispod, dok razlika u strukturi formula otkriva grešku. Opis greške ulazi u žurnal, a pomoćna promenljiva *Ispravljene\_formule* dobija vrednost *TRUE*.

Analiza orezane *LMG* se dodatno usložnjava u situaciji kada formule koje se odnose na promenljive istog tipa u vremenskom periodu pre ( $t-1$ ) i nakon ( $t+1$ ) posmatranog, nisu iste. Različita struktura ovih formula ukazuje na ćeliju sa greškom, ali je potrebno proveriti koja od dve potencijalno neispravne strukture formula je uzrokovala grešku. U ovom koraku testiranje se nastavlja upoređivanjem strukture novih potencijalno pogrešnih ćelija sa ćelijama promenljivih istog tipa u prethodnom ili narednom vremenskom periodu. Za ćeliju ispod analizirane ( $t+1$ ) kao uporedna struktura formule uzima se naredna formula, odnosno naredni vremenski period promenljive ( $t+2$ ). Ukoliko je ta ćelija po strukturi formule podudarna promenljivoj istog tipa u narednom vremenskom periodu, njena struktura se smatra ispravnom. Prvobitno analizirana formula se ispravlja u skladu sa ispravnom strukturom, opis greške ulazi u žurnal, a pomoćna promenljiva *Ispravljene\_formule* dobija vrednost *TRUE*. Pošto se formula u ćeliji koja se odnosi na promenljivu u vremenskom periodu  $t+1$  smatra ispravnom, to ukazuje na mogućnost da je formula u ćeliji koja se odnosi na promenljivu u vremenskom periodu  $t-1$  neispravna. Ćelija koja se odnosi na promenljivu u vremenskom periodu  $t-1$  ulazi u Listu ćelija sa potencijalno pogrešnom formulom (*LPPF*). U slučaju da strukture formula u vremenskim periodima  $t+1$  i  $t+2$  nisu iste, ćelije koje se odnose na promenljive u ovim vremenskim periodima ulaze u *LPPF*. Obe ćelije ulaze u *LPPF* jer u ovom koraku nije moguće odrediti koja od njih ima neispravnu strukturu formule. Analiza se nastavlja nad ćelijom iznad prvobitno analizirane, odnosno za period  $t-1$ . Za ćeliju iznad analizirane ( $t-1$ ), kao uporedna struktura formule uzima se prethodna formula, odnosno prethodni vremenski period promenljive ( $t-2$ ). Dalji postupak analize je analogan slučaju ispitivanja strukture formule ispod analizirane.

Ukoliko je rezultat Algoritma 7 barem jedna ispravljena ćelija, odnosno pomoćna promenljiva *Ispravljene\_formule* ima vrednost *TRUE*, procedura se vraća na Algoritam 4. Opisi ustanovljenih grešaka ulaze u žurnal. U slučaju da pomoćna promenljiva *Ispravljene\_formule* ima vrednost *FALSE*, odnosno nije bilo ispravki, proverava se da li je *LPPF* prazna. Ako *LPPF* nije prazna, elementi ove liste će biti prekopirani u *LMG*, a procedura se se vraća na Algoritam 6. Prazna *LPPF* upućuje na analizu prethodnika

sledećeg nivoa. *LMG* se puni prethodnicima ćelija koje su analizirane u realizovanoj iteraciji i njihovim direktnim prethodnicima, a procedura se vraća na Algoritam 6.

```

Ulaz: orezana LMG, spredšit model
Izlaz: ispravljen spredšit model, žurnal, LPPF
For each c in orezana LMG
    If formula  $c(t) \neq$  formula  $c(t-1)$  or formula  $c(t) \neq$  formula  $c(t+1)$  Then
        If formula  $c(t-1) =$  formula  $c(t+1)$  Then
            formula  $c(t) =$  formula  $c(t+1)$ 
            ubaci opis greške za  $c(t)$  u žurnal
            Ispravljene_formule = TRUE
        Else
            If formula  $c(t+1) =$  formula  $c(t+2)$  Then
                formula  $c(t) =$  formula  $c(t+1)$ 
                ubaci opis greške za  $c(t)$  u žurnal
                Ispravljene_formule = TRUE
                ubaci ćeliju  $c(t-1)$  u LPPF
            Else
                ubaci ćeliju  $c(t+1)$  and ćeliju  $c(t+2)$  u LPPF
                If formula  $c(t-1) =$  formula  $c(t-2)$  Then
                    formula  $c(t) =$  formula  $c(t-1)$ 
                    ubaci opis greške za  $c(t)$  u žurnal
                    Ispravljene_formule = TRUE
                Else
                    ubaci ćeliju  $c(t-1)$  and ćeliju  $c(t-2)$  u LPPF
                End if
            End if
        End if
    End if
Next c
    
```

**Algoritam 7.** Provera orezane Liste ćelija sa mogućom greškom

Specijalan slučaj predstavlja poređenje strukture formula u ćelijama iz orezane *LMG* sa strukturama formula promenljivih istog tipa u prethodnom ( $t-1$ ) i narednom ( $t+1$ ) vremenskom periodu koje imaju istu neispravnu strukturu. Struktura formule u ćeliji koja se analizira će biti izmenja u skladu sa strukturom formula iz ćelije iznad i/ili ispod. Ispravka se beleži u žurnalu, a pomoćna promenljiva *Ispravljene\_formule* dobija vrednost *TRUE*, dok se procedura vraća na Algoritam 4. Ispravka sprovedena u ovom slučaju inicira novu grešku koja može biti detektovana Algoritmom 4 ili Algoritmom 5. Pravila orezivanja *LMG* omogućavaju detekciju greške u strukturi formule koja se

odnosi na promenljivu u vremenskom periodu  $t-1$ . Nakon ispravke strukture ove formule analiza se sprovodi na već opisan način, kada formule koje se odnose na promenljive istog tipa u vremenskom periodu  $t-1$  i  $t+1$  nisu iste.

Opisani koraci pristupa se ponavljaju sve dok izlaz Algoritma 4 nije prazna *LMG*. Kako ovaj algoritam uključuje izlaz pristupa za detekciju i analizu grešaka u DDMUZRS, koji se opisuje u disertaciji, krajnji rezultat primene Algoritma 4 je ispravljen spredšit model. Pristup može da uključi i kriterijum zaustavljanja. Kriterijum zaustavljanja podrazumeva maksimalni broj iteracija ili vreme izvršenja.

### 7.3. Generisanje test slučajeva

U skladu sa definisanim DDMUZRS, test slučajevi za detekciju i analizu grešaka predstavljaju kombinacije dopustivih vrednosti promenljivih upravljanja. Dopustivost test slučajeva podrazumeva nenegativnost oblasti upravljanja, određene relacijama 5.13 u opštem slučaju i 5.20 za konkretan slučaj. Kreiranje test slučajeva se, u najvećem broju koraka, može poistovetiti sa traženjem rešenja zadatka optimalnog upravljanja diskretnim objektom. Kao rešenje ovog zadatka, potrebno je pronaći onaj diskretni proces  $(p, u, X_t)$  koji ciljnoj funkciji daje vrednost ne manju (ne veću) od vrednosti koju ciljnoj funkciji daje bilo koji drugi dopustivi proces. Kada je u pitanju kreiranje test slučajeva, cilj je kreiranje različitih dopustivih procesa, bez uslova postizanja optimalne vrednosti ciljne funkcije. U skladu sa navedenim, metode koje se koriste za pretraživanja oblasti upravljanja objekta diskretnog upravljanja (Đorđević, 2010), mogu se koristiti i za kreiranje test slučajeva za pristup detekcije i analize grešaka u DDMUZRS. U cilju skraćivanja vremena za generisanje test slučajeva, koraci koji se odnose na određivanje vrednosti ciljne funkcije se mogu eliminisati iz metoda pretraživanja oblasti upravljanja objekta diskretnog upravljanja.

Određivanje vrednosti promenljivih upravljanja može se izvoditi simulacijom za svaki vremenski period. Moguće vrednosti upravljačkih promenljivih za naredni vremenski period se ispituju kada su vrednosti oblasti upravljanja nenegativne za tekući period. Na taj način, sve vrednosti oblasti upravljanja za prethodne periode sigurno obezbeđuju

nenegativnost. Određivanje zadovoljavajućih vrednosti upravljačkih promenljivih umnogome zavisi od razumevanja formula oblasti upravljanja, kao i drugih relevantnih formula u modelu. Sistem povezanih formula, kreiran u skladu sa elementima diskretnog objekta upravljanja, predstavlja simulacioni model. Simulacija se izvodi izborom vrednosti upravljačkih promenljivih iz jednog u naredni vremenski period, pri čemu su vrednosti svih elemenata modela u svakom periodu povezane. Pažljivim izborom vrednosti promenljivih upravljanja može se pronaći dopustivo rešenje. To je ono rešenje koje obezbeđuje nenegativnost vrednosti koje daju formule oblasti upravljanja. Negativne vrednosti odgovarajućih relacija oblasti upravljanja treba da budu putokaz ka pronalazenju dopustivih vrednosti promenljivih upravljanja.

Za kreiranje test slučajeva može se koristiti i logika metode potpunog pretraživanja. Koncepcijska postavka metode potpunog pretraživanja oblasti dopustivog upravljanja suštinski je veoma jednostavna i primenljiva za utvrđivanje svih test slučajeva. Potrebno je ispitati sve moguće tačke unapred utvrđene oblasti, za koju se zna da sadrži optimalnu tačku. Metoda potpunog pretraživanja se može prikazati sledećim koracima (Đorđević, 2010):

1. Utvrditi upravljačku tačku koja će se ispitivati (ovo se obično radi utvrđivanjem kombinacija vrednosti po datim koordinatama prostora upravljanja za svaki vremenski period  $t=1,2,\dots,T$ );
2. Proveriti dopustivost tačke kroz sistem relacija oblasti upravljanja;
3. Ako tačka nije dopustiva, vratiti se na korak 1; ako je dopustiva, nastaviti proces;
4. Izračunati vrednost ciljne funkcije za odabranu upravljačku tačku;
5. Ako je nova vrednost ciljne funkcije bolja od ranije registrovane najbolje, zapamtiti odabranu upravljačku tačku i vrednost ciljne funkcije;
6. Vratiti se na korak 1 ukoliko postoji još mogućih tačaka ispitivanja.

Tehnika potpunog pretraživanja ima svoja praktična ograničenja u broju tačaka koje se mogu analizirati u prihvatljivom vremenu (Kostić, 2008a). Već preko milion tačaka predstavlja problem, a svaki ozbiljniji model zahtevaće pretraživanje oblasti koja sadrži mnogo više od milion tačaka. Broj nedopustivih tačaka, koje se izdvajaju na ovaj način,

moguće je smanjiti različitim pristupima. Jedan takav pristup je metoda pretraživanja oblasti upravljanja sa izbegavanjem nedopustivih grana ili metoda implicitnog pretraživanja, koja se može povezati sa pogodnostima koje pruža koncepcija objekta diskretnog upravljanja. Upravljanje  $u_1, u_2, \dots, u_t$  je dopustivo ukoliko je u svakom vremenskom periodu  $t$  ( $t=1, 2, \dots, T$ ) izabrana dopustiva upravljačka tačka  $u_t$ . Polazeći od činjenice da se proces upravljanja diskretnim objektom odvija u  $t$  vremenskih etapa,  $R \cdot T$ -dimenziona oblast pretraživanja se može podeliti u  $T$   $R$ -dimenzionih oblasti, pri čemu postoji  $R$  promenljivih upravljanja. U etapi  $t=1$  pretražuje se  $R$ -dimenziona oblast pretraživanja, sa mnogo manjim brojem tačaka od broja koji ima  $R \cdot T$ -dimenziona oblast pretraživanja. Svaka otkrivena nedopustiva tačka u ovoj oblasti pretraživanja eliminiše mogućnost izbora kombinacije ove nedopustive tačke sa nekom od  $(T-1)^{BT}$  kombinacija tačaka ( $BT=m^R$ ), koliko se nalazi u preostalim  $T-1$  oblasti pretraživanja. Sa  $m$  se označava broj diskretnih vrednosti na nekoj koordinati prostora upravljanja, a sa  $R$  broj dimenzija prostora upravljanja. Kada se otkrije dopustiva tačka  $u_1$ , prelazi se na oblast pretraživanja u vremenu  $t=2$ . U vremenu  $t=2$  pretražuje se  $R$ -dimenziona oblast pretraživanja, sa mnogo manjim brojem tačaka od broja koji ima  $R \cdot T$ -dimenziona oblast pretraživanja. Svaka otkrivena nedopustiva tačka u ovoj oblasti pretraživanja eliminiše mogućnost izbora kombinacije ove nedopustive tačke sa nekom od  $(T-2)^{BT}$  kombinacija tačaka, koliko se nalazi u preostalim  $T-2$  oblasti pretraživanja. Kada se otkrije dopustiva tačka  $u_2$  prelazi se na oblast pretraživanja u vremenu  $t=3$ . Ovakav postupak se sprovodi sve dok se ne nađe dopustiva tačka upravljanja u poslednjem vremenskom periodu  $t=T$ . Na taj način je pronađen jedan dopustivi proces  $(p_t, u_t, X_t)$ . Preostaje da se pronađe i dovoljan broj drugih dopustivih procesa, pa da se upoređivanjem vrednosti odgovarajućih ciljnih funkcionala izabere najbolji proces od svih ispitanih.

Za kreiranje test slučajeva može se koristiti i *Monte Carlo* metoda. Ova metoda podrazumeva slučajni izbor tačaka iz oblasti upravljanja. Broj tačaka koje se analiziraju su u skladu sa raspoloživim vremenom za rešavanje problema i brzinom rada računara. Oblast pretraživanja (utvrđena kao višedimenzioni poliedar ili neka druga hiperpovrš) sadrži konačan broj diskretnih tačaka. Suština metode *Monte Carlo* je da se iz prostora dopustivih rešenja, na slučajan način, bira tačka koja predstavlja novo rešenje. Ovaj postupak se ponavlja sve dok se ne zadovolji neki, unapred zadati, izlazni kriterijum.

Prema (Đorđević, 2010) Metoda *Monte Carlo* se, u kontekstu pretraživanja oblasti upravljanja objekta diskretnog upravljanja, može prikazati i sledećim koracima:

1. Utvrditi slučajnu upravljačku tačku koja će se ispitivati;
2. Proveriti dopustivost tačke kroz sistem relacija oblasti upravljanja;
3. Ako tačka nije dopustiva, vratiti se na korak 1; ako je dopustiva, nastaviti proces;
4. Izračunati vrednost ciljne funkcije za odabranu upravljačku tačku;
5. Ako je nova vrednost ciljne funkcije bolja od ranije registrovane kao najbolje, zapamtiti odabranu upravljačku tačku i vrednost ciljnog funkcionala;
6. Vratiti se na korak 1 ukoliko postoji još mogućih tačaka ispitivanja i kriterijum zaustavljanja nije zadovoljen.

Nedostaci *Monte Carlo* metode ne odražavaju se pri kreiranju test slučajeva za pristup detekcije i analize grešaka u DDMUZRS, koji je opisan u ovom radu. Ova metoda veoma haotično pretražuje prostor dopustivih rešenja, te je stoga lako objasniti njenu neefikasnost prilikom traženja globalnog minimuma funkcije cilja, naročito u slučajevima problema velikih dimenzija ili problema kod kojih je prostor dopustivih rešenja izuzetno veliki. S obzirom da cilj pristupa detekcije i analize grešaka nije traženje globalnog optimuma funkcije cilja, metoda *Monte Carlo* se može smatrati pogodnom za definisanje test slučajeva.

#### **7.4. Implementacija pristupa za detekciju i analizu grešaka u DDMUZRS i numerički rezultati izvedenih eksperimenata**

Implementacija pristupa za detekciju i analizu grešaka u DDMUZRS, prikazanog u prethodnim potpoglavljima, nad modelom problema prikazanog u poglavlju 5.3, realizovana je korišćenjem programskog jezika *Visual Basic for Application*. Kao spredšit alat za razvoj modela korišćen je *MS Excel*. Algoritmi pristupa koriste međurezultate dobijene proračunima u diskretnom dinamičkom modelu, razvijenom u spredšitu. Svi eksperimenti su izvršeni na *Windows 7 - Ultimate Operating System, Pentium (R) Dual-Core CPU T4200 procesor sa 4.00 GB RAM i 2.00 GHz*. Kao

napomena, neophodno je da se ukaže da performanse računara mogu značajno da utiču na vreme izvršenja algoritama ovog pristupa.

Za evaluaciju pristupa koristi se mutacija spredšit modela opisanog u poglavlju 5.3. Mutacija podrazumeva veštačko umetanje grešaka, odnosno spredšit orjentisanu mutaciju operatora (Abraham & Erwig, 2009). Primeri mutacije operatora podrazumevaju zamenu matematičke operacije oduzimanja operacijama množenja ili sabiranja, operacije množenja zamenom operacijama deljenja ili sabiranja i slično. Pored mutacije operatora, za umetanje grešaka mutirani su i elementi modela. Na primer: u formulu se uvodi tip promenljive koji ona ne podrazumeva, tip promenljive koji pripada određenoj formuli se uvodi u neodgovarajućem vremenskom periodu, delovi formule se izostavljaju, formula se menja konstantom itd.

Kao metrike evaluacije pristupa koji je opisan u ovom radu, razmatrani su procenat detektovanih i ispravljenih grešaka, kao i vreme potrebno za izvršenje algoritama. Kriterijum zaustavljanja nije korišćen u eksperimentima, kako bi bilo ispitano vreme potrebno za izvršenje algoritama. U cilju provere skalabilnosti pristupa, razmatrani su modeli različitih dimenzija. Broj ćelija spredšit modela, odnosno formula matematičkog modela, povećavan je dodavanjem proizvoda koji se naručuju i povećavanjem vremenskih perioda u kojima se naručivanje sprovodi. Algoritmi pristupa su testirani na DDMUZRS za 3, 20 i 100 proizvoda. Proces upravljanja zalihama razmatran je u vremenskim periodima od  $T=1$  nedelje, podeljene na 7 dana;  $T=4$  nedelje, podeljene na 28 dana i  $T=1$  godine, podeljene na 365 dana. Efikasnost pristupa ispitana je za slučajno generisane ulazne podatke, za isti broj proizvoda i posmatranih vremenskih perioda. Opsezi ulaznih parametara predstavljeni su Tabelom 7-1.

**Tabela 7-1.** Opseg vrednosti ulaznih parametara

Ulazni podaci	$D_i$ [jed.]	$P_i$ [m <sup>2</sup> /jed.]	$S_i$ [\$/narudžbina]	$C_i$ [\$/jed.]	$H_i$ [\$/jed./god.]
Opseg	[200; 10000]	[0.3; 2.00]	[200;1000]	[10; 20]	[1.5; 5]



### 7.4.1 Detekcija i analiza grešaka u DDMUZRS za tri proizvoda

Pristup detekcije i analize grešaka zasnovan na karakteristikama problema i načina modeliranja evaluiran je nad DDMUZRS, prikazanim u poglavlju 5.3 i njegovim potpoglavljima, gde je u razmatranje uzeto naručivanje tri vrste proizvoda. Proces upravljanja zalihama posmatran je u vremenskim periodima:  $T=1$  nedelja, podeljena na 7 dana;  $T=4$  nedelje, podeljene na 28 dana i  $T=1$  godina, podeljena na 365 dana. Opisani model za tri proizvoda i period posmatranja od 7 dana prikazan je na Slici 7-1.

Ispitivanje efikasnosti algoritama predstavljenih u potpoglavljju 7.2, sprovedeno je variranjem broja i tipa veštački ugrađenih grešaka. Sve greške u strukturi ciljnih funkcija su 100% detektovane u zanemarljivo kratkom vremenu, pa zbog toga nisu analizirane u daljem tekstu.

Ulazni		Proizvod		
Ukupna trazenja D		200	150	60
Tr. narucivanja S		1500	1500	1500
Jed. cena C		18	15	10
Ukupni tr. drzanja zalih (H)		2,7	2,25	1,5
Zauzece prostora (po jed.)		0,8	0,4	0,3

Proizvod	Proizv	1	2	3
Broj dana T		7,00	7,00	7,00
Dnevna trazenja D/T		28,57	21,43	8,57
Tr. narucivanja S		1500,00	1500,00	1500,00
Jednaci cena C		18,00	15,00	10,00
Tr. drzanja zalih H		0,39	0,32	0,21429
Zauzece prostora (jed) P		0,80	0,40	0,30
Raspolozivi prostor G		100,00	100,00	100,00

Quantity ordered	q=D/u	50	50	20
	u=N <sub>o</sub>	4	3	3

Oblast upravljanja				Regulatori protoka						Promenljive stanja			Ciljna funkcija			
O1	O2	O3	O4	Y1	Y2	Y3	Y4	Y5	Y6	X1	X2	X3	J	J1	J2	J3
t=1	21,43	28,57	11,43	67,00	t=1	50,00	50,00	20,00	28,57	21,43	8,57	t=0	0,00	0,00	0,00	0,00
t=2	42,86	7,14	2,86	64,00	t=2	50,00	0,00	0,00	28,57	21,43	8,57	t=1	6,379,77	2,413,78	2,262,63	1,703,37
t=3	14,29	35,71	14,29	68,00	t=3	0,00	50,00	20,00	28,57	21,43	8,57	t=2	8,809,08	4,835,82	2,268,37	1,704,90
t=4	35,71	14,29	5,71	65,00	t=4	50,00	0,00	0,00	28,57	21,43	8,57	t=3	12,789,01	4,846,84	4,533,29	3,408,88
t=5	7,14	42,86	17,14	69,00	t=5	0,00	50,00	20,00	28,57	21,43	8,57	t=4	15,218,47	7,266,12	4,541,33	3,411,02
t=6	28,57	21,43	8,57	66,00	t=6	50,00	0,00	0,00	28,57	21,43	8,57	t=5	19,198,55	7,274,39	6,808,55	5,115,61
t=7	0,00	0,00	0,00	83,00	t=7	0,00	0,00	0,00	28,57	21,43	8,57	t=6	21,628,16	9,690,92	6,818,88	5,118,37
												t=7	21,638,04	9,696,43	6,822,32	5,119,29

Slika 7-2. DDMUZRS za tri proizvoda i horizont posmatranja od sedam dana

Rezultati evaluacije pristupa nad DDMUZRS za tri proizvoda i sedam dana, sa jednom ugrađenom greškom, prikazani su u Tabeli 7-2. Eksperimenti su sprovedeni sa pet različitih tipova grešaka, pri čemu je pozicija ugrađene greške u modelu menjanja, po pet puta za svaku. Svaki eksperiment ponovljen je 10 puta, u cilju eliminacije “šuma“ na rezultatima. Različiti tipovi grešaka uključuju: pogrešnu dimenziju promenljive u strukturi formule, pogrešan tip promenljive u strukturi formule, eliminisanje elementa

formule, mutaciju operatora i zamenu formule konstantom. Na osnovu rezultata eksperimenata može se zaključiti da je pristup zasnovan na karakteristikama problema i načinu modeliranja primenljiv nad DDMUZRS manjih dimenzija i da u 100% slučajeva detektuje postojanje greške i utvrđuje njen uzrok u prihvatljivom vremenskom periodu. Mala odstupanja u vremenu izvršenja impliciraju da tip i pozicija greške nemaju veliki uticaj na detekciju i analizu grešaka, kada su u pitanju modeli manjih dimenzija. Za implementaciju pristupa nad modelom manjih dimenzija uspešno je korišćen jedan test slučaj, generisan na slučajan način.

**Tabela 7-2.** Rezultati evaluacije pristupa nad DDMUZRS za tri proizvoda,  $T=7$  dana, sa jednom ugrađenom greškom

veličina modela [broj ćelija]	broj formula u modelu	tip greške	detektovane i ispravljene greške [%]	prosečno vreme (sec)
171	120	Pogrešna dimenzija promenljive u strukturi formule stanja	100	6
171	120	Pogrešan tip promenljive u strukturi formule	100	5
171	120	Eliminisanje elementa formule	100	5
171	120	Mutacija operatora	100	6
171	120	Zamena formule konstantom	100	4

Opisani eksperimenti ponovljeni su nad istim modelom sa ugrađene dve greške (Tabela 7-3). Razmatrane greške postavljanje su u jednu ili više ćelija. Zaključeno je da se više grešaka u istoj ćeliji može poistovetiti sa jednom greškom, s obzirom da se nakon prvobitne detekcije greške sprovodi analiza cele formule u odnosu na tačnu strukturu prethodne ili naredne ćelije. Rezultati eksperimenata pokazuju da vreme detekcije i analize za dve greške u značajnoj meri ne odstupa od vremena potrebnog za detekciju i analizu jedne ugrađene greške, kao i da tip i pozicija greške nemaju veliki uticaj. Osim implementacije dve greške istog tipa, razmatrane su istovremeno ugrađene dve greške različitog tipa, što nije bitno uticalo na rezultat. Kao osnovna razlika u evaluaciji pristupa ustanovljen je broj test slučajeva. Detekcija dve ili više grešaka u najvećem

broju slučajeva zahteva više od jednog test slučaja. Test slučajevi se mogu generisati u skladu sa nekim od pristupa opisanih u potpoglavlju 7.3 ili na slučajan način. U ovom slučaju skup od 5 test slučajeva generisan je simulacijom.

**Tabela 7-3.** Rezultati evaluacije pristupa nad DDMUZRS za tri proizvoda,  $T=7$  dana, sa dve ugrađene greške

veličina modela [broj ćelija]	broj formula u modelu	tip greške	detektovane i ispravljene greške [%]	prosečno vreme (sec)
171	120	Pogrešna dimenzija promenljive u strukturi formule stanja	100	6
171	120	Pogrešan tip promenljive u strukturi formule	100	6
171	120	Eliminisanje elementa formule	100	6,5
171	120	Mutacija operatora	100	6
171	120	Zamena formule konstantom	100	4

Eksperimenti su ponovljeni za više ugrađenih grešaka, pri čemu se pokazalo da detekcija i analiza grešaka u modelima manjih dimenzija ima iste performanse do pet ugrađenih grešaka. Pristup se pokazao u potpunosti uspešnim i za veći broj ugrađenih grešaka, ali vreme realizacije pristupa umnogome zavisi od izbora test slučajeva u odnosu na poziciju greške i tip greške.

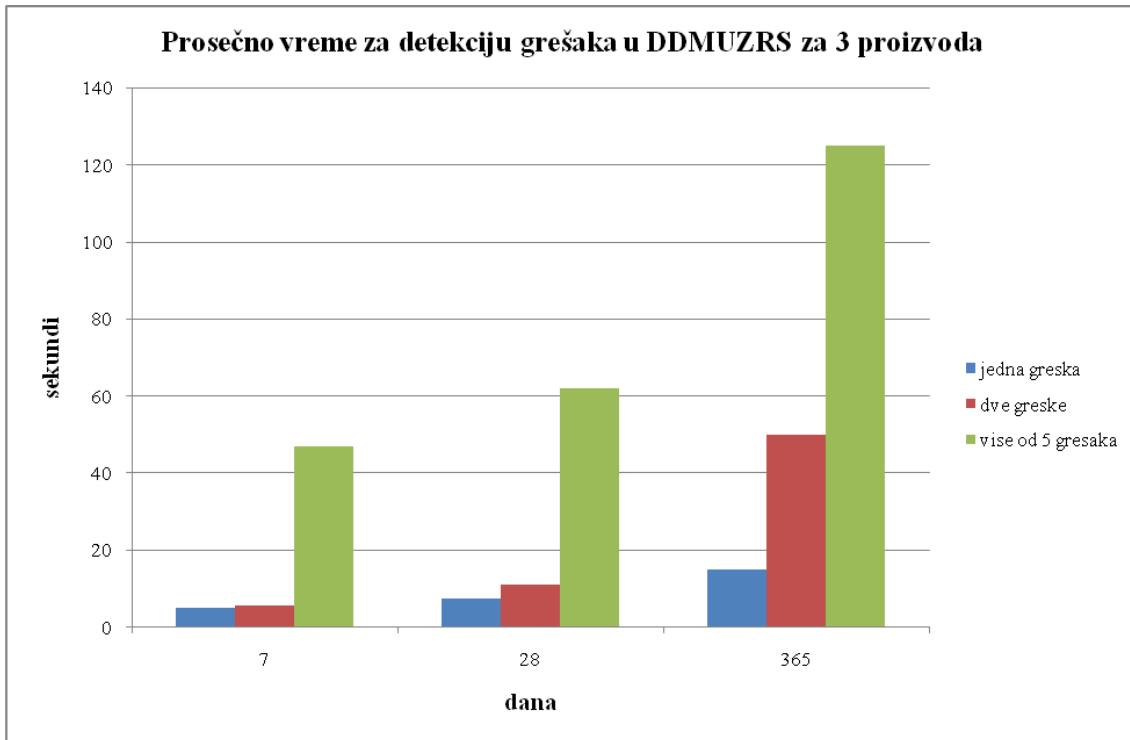
U cilju provere skalabilnosti pristupa opisanog u ovom radu, veličina modela je varirana povećanjem broja posmatranih vremenskih perioda, pri čemu je broj proizvoda ostao isti. Rezultati eksperimenata prikazani su Tabelom 7-4. Uspešnost detekcije i analize grešaka je visoka, bez obzira na veličinu modela koji se analizira. Ipak, vreme izvršenja se povećava sa produženjem horizonta posmatranja. Uzrok se može naći u činjenici da veći broj perioda posmatranja podrazumeva širi skup dopustivih test slučajeva. Kako bi se odredio uzrok greške u velikim modelima, neophodno je koristiti više test slučajeva. Takođe, verovatnoća slučajnog izbora „dobrih“ test slučajeva, koji će omogućiti detekciju grešaka i otkrivanje uzroka, se smanjuje sa povećanjem broja posmatranih

vremenskih perioda. Procenat uspešno detektovanih grešaka direktno zavisi od broja ispitanih test slučajeva. Ukoliko bi se ispitali svi dopustivi test slučajevi, moglo bi se pretpostaviti da su sve greške i njihovi uzroci otkriveni. Sa druge strane, povećanje broja ispitanih test slučajeva direktno utiče na vreme izvršenja procedura. U eksperimentima prikazanim Tabelom 7-4, modeli manjih i srednjih dimenzija analizirani su na osnovu pet test slučajeva, dok je za veći model razmatrano 10 test slučajeva, po svakom eksperimentu.

**Tabela 7-4.** Upporedni prikaz rezultata evaluacije pristupa nad DDMUZRS za tri proizvoda

veličina modela [broj ćelija]	broj formula u modelu	T [dana]	broj ugrađenih grešaka	detektovane i ispravljene greške [%]	prosečno vreme (sec)
171	120	7	1	100	5,2
534	483	28	1	100	7,5
6260	6209	365	1	100	15
171	120	7	2	100	5,7
534	483	28	2	100	11
6260	6209	365	2	100	50
171	120	7	>5	100	47
534	483	28	>5	90	62
6260	6209	365	>5	80	125

Poređenjem rezultata evaluacije ovog pristupa, može se zaključiti da veći broj vremenskih perioda posmatranja značajno utiče na vreme potrebno za detekciju i analizu grešaka, posebno kod modela većih dimenzija. Pored broja vremenskih perioda, na vreme realizacije pristupa značajno utiče i broj ugrađenih grešaka (Slika 7-2). Duže vreme potrebno za detekciju i analizu grešaka, u slučaju većeg broja ugrađenih grešaka, posledica je potrebe uključivanja većeg broja test slučajeva, u cilju detekcije različitih grešaka.



**Slika 7-3.** Prosečno vreme potrebno za detekciju grešaka u DDMUZRS za tri proizvoda

#### 7.4.2 Detekcija i analiza grešaka u DDMUZRS za 20 i 100 proizvoda

Pored povećanja broja posmatranih vremenskih perioda, skalabilnost pristupa detekcije i analize grešaka može se analizirati povećanjem broja proizvoda koji se naručuju. Proces upravljanja zalihama posmatran je u vremenskim periodima:  $T=1$  nedelja, podeljena na 7 dana;  $T=4$  nedelje, podeljene na 28 dana i  $T=1$  godina, podeljena na 365 dana, kao i u slučaju razmatranja tri proizvoda, ali je u obzir uzet veći broj proizvoda. Povećanje broja proizvoda odgovara realnoj situaciji promene broja artikala na zalihama. U nastavku rada biće prikazani eksperimenti koji podrazumevaju 20 i 100 proizvoda.

Eksperimenti su izvedeni na način opisan u potpoglavlju 7.4.1. Ispitivanja efikasnosti algoritama predstavljenih u potpoglavlju 7.2, sprovedeno je variranjem broja i tipa veštački ugrađenih grešaka. Rezultati evaluacije pristupa nad DDMUZRS za 20 i 100 proizvoda, različiti broj perioda posmatranja i ugrađenih greška prikazani su Tabelom 7-5 i Tabelom 7-6. Eksperimenti su sprovedeni sa različitim tipovima greška, pri čemu je pozicija ugrađene greške u modelu menjanja. Svaki eksperiment ponavljen je 10 puta,

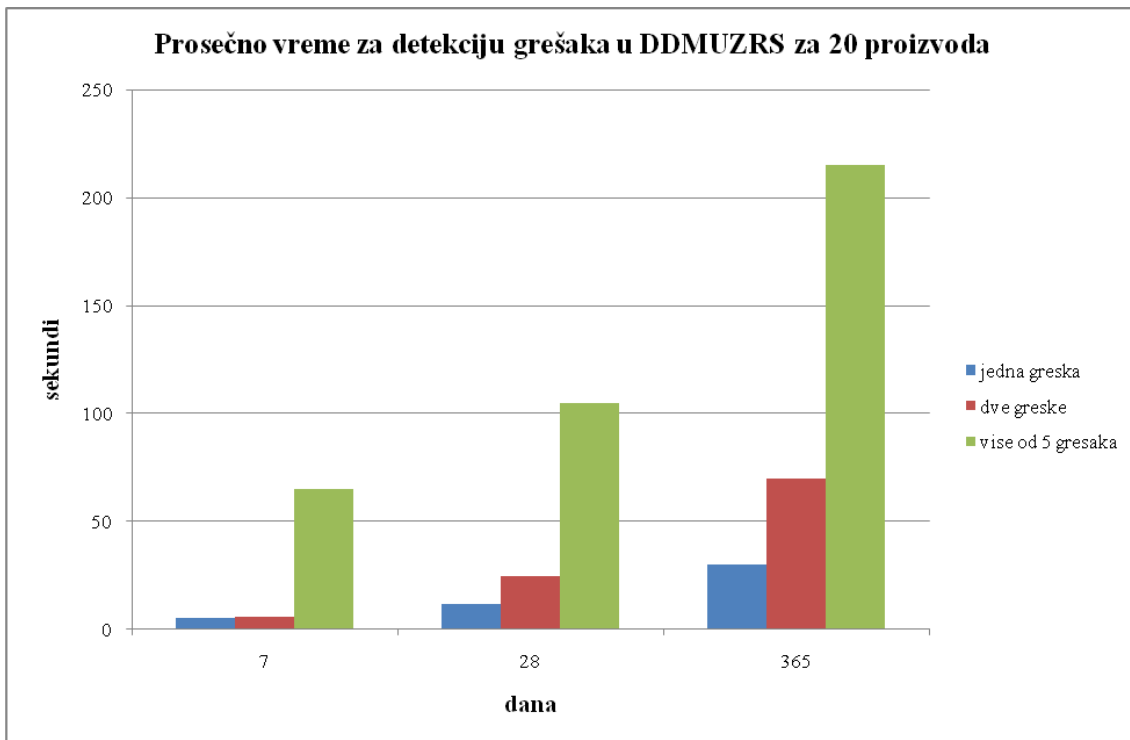
u cilju eliminacije „šuma“ na rezultatima. Različiti tipovi grešaka uključuju: pogrešnu dimenziju promenljive u strukturi formule stanja, pogrešan tip promenljive u strukturi formule, eliminisanje elementa formule, mutaciju operatora i zamenu formule konstantom.

**Tabela 7-5.** Upporedni prikaz rezultata evaluacije pristupa nad DDMUZRS za 20 proizvoda

veličina modela [broj ćelija]	broj formula u modelu	T [dana]	broj ugrađenih grešaka	detektovane i ispravljene greške [%]	prosečno vreme (sec)
973	616	7	1	100	5,5
3375	3018	28	1	100	12
39413	39056	365	1	100	30
973	616	7	2	100	6
3375	3018	28	2	100	25
39413	39056	365	2	100	70
973	616	7	>5	90	65
3375	3018	28	>5	90	105
39413	39056	365	>5	80	215

Na osnovu rezultata eksperimenata može se zaključiti da je pristup zasnovan na karakteristikama problema i načinu modeliranja primenljiv nad DDMUZRS različitih dimenzija. U slučaju modela manjih dimenzija, detekcija postojanja grešaka i utvrđivanje njihovih uzroka u kratkom vremenskom periodu je 100%. Primena na modelima koji podrazumevaju veći broj ćelija i formula pokazala se uspešnom, ali sa smanjenim procentom uspešnosti. Iako smanjen, procenat uspešnosti je na zadovoljavajućem nivou. Smanjena uspešnost pristupa posledica je značajno povećanog broja mogućih test slučajeva koje treba ispitati kako bi se greška detektovala. Dužina horizonta posmatranja direktno utiče na skup svih test slučajeva. Kao što je prethodno spomenuto, procenat pronađenih grešaka zavisi od test slučajeva, odnosno za sve ispitane

test slučajevne greške bi bile 100% detektovane. Ipak, ispitivanje celog skupa dopustivih test slučajeva je za velike primere vremenski zahtevno i neprihvatljivo. Duže vreme potrebno za detekciju grešaka uzrokovano je i veoma širokim skupom prethodnika ćelija u kojima se greška detektuje. Prosečno vreme za detekciju i analizu grešaka u DDMUZRS za 20 proizvoda prikazano je Slikom 7-3. Za implementaciju pristupa nad modelima većih dimenzija korišćeno je od dva do deset test slučajeva, srazmerno veličini modela.



**Slika 7-4.** Prosečno vreme potrebno za detekciju i analizu grešaka u DDMUZRS za 20 proizvoda

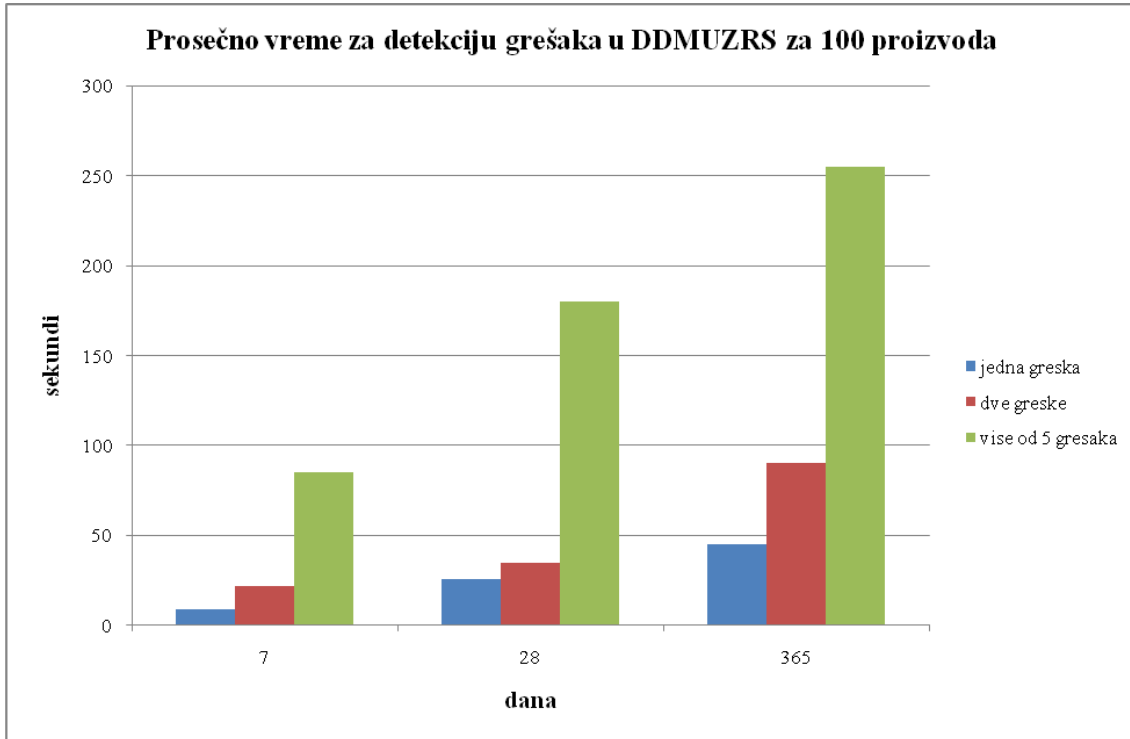
Eksperimenti koji odgovaraju DDMUZRS za 100 proizvoda sprovedeni su na prethodno opisan način za 20 proizvoda. Modeli koji se odnose na 100 proizvoda spadaju u grupu velikih i izuzetno velikih. Na osnovu rezultata, prikazanih Tabelom 7-6. i Slikom 7-4., dokazana je primenljivost i skalabilnost pristupa za detekciju i analizu grešaka zasnovanog na karakteristikama problema i načinu modeliranja nad DDMUZRS.

**Tabela 7-6.** Uporedni prikaz rezultata evaluacije pristupa nad DDMUZRS za 100 proizvoda

<b>veličina modela [broj ćelija]</b>	<b>broj formula u modelu</b>	<b>T [dana]</b>	<b>broj ugrađenih grešaka</b>	<b>detektovane i ispravljene greške [%]</b>	<b>prosečno vreme (sec)</b>
5027	3599	7	1	100	9
15923	14495	28	1	100	26
189243	187815	365	1	100	45
5027	3599	7	2	100	22
15923	14495	28	2	100	35
189243	187815	365	2	100	90
5027	3599	7	>5	90	85
15923	14495	28	>5	90	180
189243	187815	365	>5	80	255

Vreme potrebno za detekciju i analizu grešaka u velikim modelima, iako duže nego kod modela manjih i srednjih dimenzija, može se smatrati prihvatljivim u odnosu na broj ćelija posmatranih modela. Procenat detektovanih grešaka, iako smanjen, ostaje na visokom nivou. Uzrok ovog smanjenja objašnjen je na primeru sa 20 razmatranih proizvoda. Povećanje broja ugrađenih grešaka se značajno odražava na izuzetno velike modele. Uticaj broja grešaka na vreme potrebno za detekciju i analizu u velikim modelima, može se objasniti i širokim skupovima prethodnika, čiji se broj dodatno povećava sa svakom novougrađenom greškom. Drugi aspekt razmatranja veze između broja grešaka i vremena potrebnog za sprovođenje koraka pristupa podrazumeva veći broj test slučajeva koje potrebno ispitati kako bi se greške detektovale.





**Slika 7-5.** Prosečno vreme potrebno za detekciju grešaka u DDMUZRS za 100 proizvoda

## 7.5. Analiza rezultata

Pristup zasnovan na načinu modeliranja i karakteristikama problema, čija implementacija je prikazana u prethodnom poglavlju, pokazao se kao primenljiv i efikasan za detekciju i analizu grešaka u DDMUZRS. U cilju evaluacije i provere skalabilnosti pristupa, analiziran je proces upravljanja zalihama za: 3, 20 i 100 proizvoda. Kao horizont posmatranja razmatrano je: 7, 28 i 365 dana, odnosno nedelja, mesec dana i godina. Kombinacijom različitog broja proizvoda i perioda posmatranja sprovedeni su eksperimenti nad modelima različitih veličina. Broj veštački ugrađenih grešaka je variran između jedne, dve i više od pet. Svaki eksperiment je ponovljen po 10 puta. Uporedni prikaz rezultata sprovedenih eksperimenata predstavljen je u Tabeli 7-7.

U Tabeli 7-7. prikazane su dimenzije modela kroz ukupni broj ćelija, ali i broj ćelija koje sadrže formule. Razlika između ukupnog broja ćelija i ćelija sa formulama odnosi se na ulazne podatke i test slučajeve, koji se ne razmatraju kao kandidati za detekciju i

analizu grešaka, ali utiču na veličinu modela razvijenog u spredšitu. Test slučajevi su kreirani simulacijom.

**Tabela 7-7.** Uporedni prikaz rezultata evaluacije pristupa nad DDMUZRS

broj proizvođa	T [dan]	veličina modela [broj ćelija]	broj formula u modelu	broj ugrađenih grešaka	detektovane i ispravljene greške [%]	prosečno vreme (sec)
3	7	171	120	1	100	5,2
		171	120	2	100	5,7
		171	120	>5	90	47
	28	534	483	1	100	7,5
		534	483	2	100	11
		534	483	>5	90	62
	365	6260	6209	1	100	15
		6260	6209	2	100	50
		6260	6209	>5	90	125
20	7	973	616	1	100	5,5
		973	616	2	100	6
		973	616	>5	90	65
	28	3375	3018	1	100	12
		3375	3018	2	100	25
		3375	3018	>5	90	105
	365	39413	39056	1	100	30
		39413	39056	2	100	70
		39413	39056	>5	80	215
100	7	5027	3599	1	100	9
		5027	3599	2	100	22
		5027	3599	>5	90	85
	28	15923	14495	1	100	26
		15923	14495	2	100	35
		15923	14495	>5	90	180
	365	189243	187815	1	100	45
		189243	187815	2	100	90
		189243	187815	>5	80	255

Na osnovu rezultata eksperimenata zaključeno je da je uspešnost razvijenog pristupa, opisanog u ovoj disertaciji, 100% u slučajevima detekcije u utvrđivanja uzroka jedne i dve greške. Rezultati koji se odnose na više od pet grešaka pokazali su da je uspešnost pristupa ne manja od 80%, čak i za modele veoma velikih dimenzija. Vreme potrebno

za detekciju i analizu grešaka kreće se u intervalu od 5,2 do 255 sekundi, u zavisnosti od veličine modela i broja ugrađenih grešaka.

Detaljna analiza dobijenih rezultata pokazala je da uticaj veličine modela na vreme rada algoritma zavisi od toga da li je povećan broj posmatranih proizvoda ili vremenskih perioda. Povećanje dimenzija modela usled većeg broja proizvoda manje utiče na vreme potrebno za detekciju i analizu grešaka, nego razmatranje većeg broja vremenskih perioda, kao što se može videti u poslednjoj koloni tabele 7-7.

**Tabela 7-8.** Uporedni prikaz rezultata evaluacije pristupa nad DDMUZRS za  $T=7$  dana

<b>T [dana]</b>	<b>broj proiz voda</b>	<b>veličina modela [broj ćelija]</b>	<b>broj formula u modelu</b>	<b>broj ugrađenih grešaka</b>	<b>detektovane i ispravljene greške [%]</b>	<b>prosečno vreme (sec)</b>
7	3	171	120	1	100	5,2
	20	973	616	1	100	5,5
	100	5027	3599	1	100	9
7	3	171	120	2	100	5,7
	20	973	616	2	100	6
	100	5027	3599	2	100	22
7	3	171	120	>5	90	47
	20	973	616	>5	90	65
	100	5027	3599	>5	90	85

Rezultati eksperimenata grupisani na osnovu horizonta posmatranja prikazani su Tabelom 7-8, Tabelom 7-9 i Tabelom 7-10. Iako veličina modela u svakom slučaju utiče na brzinu rada, s obzirom da je model razvijen u *MS Excel*-u, navedene tabele pokazuju da povećanje broja proizvoda ne povećava vreme srazmerno povećanju dimenzija, već manje. Upoređivanjem rezultata detekcije i analize grešaka za različite horizonte posmatranja (Tabela 7-7, Tabela 7-8, Tabela 7-9, Tabela 7-10) uočljivo je da se vreme izvršenja povećava sa povećanjem broja posmatranih perioda.

**Tabela 7-9.** Uporedni prikaz rezultata evaluacije pristupa nad DDMUZRS za  $T=28$  dana

T [dana]	broj proizvoda	veličina modela [broj ćelija]	broj formula u modelu	broj ugrađenih grešaka	detektovane i ispravljene greške [%]	prosečno vreme (sec)
28	3	534	483	1	100	7,5
	20	3375	3018	1	100	12
	100	15923	14495	1	100	26
28	3	534	483	2	100	11
	20	3375	3018	2	100	25
	100	15923	14495	2	100	35
28	3	534	483	>5	90	62
	20	3375	3018	>5	90	105
	100	15923	14495	>5	90	180

Razlika u vremenu realizacije pristupa, usled povećanja broja proizvoda i horizonta posmatranja, može se objasniti funkcionalnim karakterom spređit formula i koncepcijom diskretnog objekta upravljanja. S obzirom da su kandidati za ispitivanje prethodnici potencijalno pogrešnih ćelija, povećanje broja proizvoda ne utiče na povećanje broja prethodnika, već samo na dimenzije modela. Povećanje broja posmatranih vremenskih perioda direktno utiče na povećanje broja prethodnika potencijalno pogrešnih ćelija, jer se vrednosti velikog broja elemenata modela izračunavaju na osnovu vrednosti svih prethodnih perioda. Na osnovu navedenog, može se zaključiti da veći broj perioda posmatranja proširuje skup prethodnika za ispitivanje. Ovaj zaključak se može smatrati važnim sa aspekta primene u realnim situacijama. Kako je broj proizvoda jednog preduzeća promenljiva vrednost, opisana karakteristika se može smatrati pozitivnom stranom pristupa.

**Tabela 7-10.** Uporedni prikaz rezultata evaluacije pristupa nad DDMUZRS za  $T=365$  dana

T [dana]	broj proiz voda	veličina modela [broj ćelija]	broj formula u modelu	broj ugrađenih grešaka	detektovane i ispravljene greške [%]	prosečno vreme (sec)
365	3	6260	6209	1	100	15
	20	39413	39056	1	100	30
	100	189243	187815	1	100	45
365	3	6260	6209	2	100	50
	20	39413	39056	2	100	70
	100	189243	187815	2	100	90
365	3	6260	6209	>5	90	125
	20	39413	39056	>5	80	215
	100	189243	187815	>5	80	255

Horizont posmatranja od godinu dana se uvek može podeliti na kraće vremenske periode, čime se može rešiti problem uticaja broja posmatranih perioda na brzinu rada algoritma.

**Tabela 7-11.** Uporedni prikaz rezultata evaluacije pristupa nad DDMUZRS sa jednom ugrađenom greškom

T[dan]	veličina modela [broj ćelija]	broj formula u modelu	detektovane i ispravljene greške [%]	prosečno vreme (sec)
7	171	120	100	5,2
28	534	483	100	7,5
7	973	616	100	5,5
28	3375	3018	100	12
7	5027	3599	100	9
365	6260	6209	100	15
28	15923	14495	100	26
365	39413	39056	100	30
365	189243	187815	100	45

U cilju analize vremena potrebnog za detekciju i analizu grešaka, ali i procenta detektovanih grešaka neophodno je razmotriti i broj ugrađenih grešaka u modelu. U Tabelama 7-11, 7-12, 7-13 prikazana su prosečna vremena i udeo ispravljenih grešaka u modelima sa veštački ugrađenom jednom, dve i više od pet grešaka. Na osnovu rezultata modela sa jednom i dve veštački ugrađene greške zaključuje se da je uspešnost pristupa potpuna u prihvatljivom vremenu. U primerima sa više od pet ugrađenih grešaka uspešnost pristupa se smanjuje do 80%, dok je vreme sve duže u skladu sa povećanjem dimenzija modela.

**Tabela 7-12.** Uporedni prikaz rezultata evaluacije pristupa nad DDMUZRS sa dve ugrađene greške

<b>T[dan]</b>	<b>veličina modela [broj ćelija]</b>	<b>broj formula u modelu</b>	<b>detektovane i ispravljene greške [%]</b>	<b>prosečno vreme (sec)</b>
7	171	120	100	5,7
28	534	483	100	11
7	973	616	100	6
28	3375	3018	100	25
7	5027	3599	100	22
365	6260	6209	100	50
28	15923	14495	100	35
365	39413	39056	100	70
365	189243	187815	100	90

Na osnovu rezultata primene pristupa za detekciju i analizu grešaka zasnovanog na karakteristikama problema i načinu modeliranja na modele sa različitim brojem ugrađenih grešaka, prikazane Tabelama 7-11, 7-12, 7-13, potvrđena je i prethodno opisana osobina, da broj posmatranih vremenskih perioda značajno utiče na vreme potrebno za izvršenje koraka svih algoritma. U spomenutim tabelama može se zapaziti da je u nekim slučajevima za veće dimenzije modela potrebno vreme za detekciju i analizu grešaka kraće nego u manjim modelima. To je posledica većeg broja perioda posmatranja u manjim modelima u odnosu na modele koji su veći zbog broja posmatranih proizvoda.

Problem preklapanja dve ili više grešaka, odnosno ometanje detekcije greške zbog postojanja druge greške koja je “maskira”, rešena je uključivanjem više test slučajeva. S obzirom da se u ovom radu razmatraju diskretni dinamički simulacioni modeli, promena ulaznih vrednosti i test slučajeva utiče na promenu vrednosti promenljivih modela.

**Tabela 7-13.** Upporedni prikaz rezultata evaluacije pristupa nad DDMUZRS sa više od pet ugrađenih greška

<b>T[dan]</b>	<b>veličina modela [broj ćelija]</b>	<b>broj formula u modelu</b>	<b>detektovane i ispravljene greške [%]</b>	<b>prosečno vreme (sec)</b>
7	171	120	90	47
28	534	483	90	62
7	973	616	90	65
28	3375	3018	90	105
7	5027	3599	90	85
365	6260	6209	90	125
28	15923	14495	90	180
365	39413	39056	80	215
365	189243	187815	80	255

## 7.6. Upporedna analiza postojećih i novog pristupa za detekciju i analizu grešaka u spredšitovima

U cilju uporedne analize postojećih pristupa za detekciju i analizu grešaka u spredšitovima sa pristupom koji je prikazan u ovoj doktorskoj disertaciji biće korišćena kategorizacija pristupa automatizovanog obezbeđenja kvaliteta spredšitova, prikazana u potpoglavljju 6.3.

Pristupi vizualizacije (potpoglavlje 6.3.1) uglavnom su orijentisani ka unapređenju prikaza određenih aspekata spredšitova. Bolja vizualizacija elemenata spredšita olakšava korisniku razumevanje međuzavisnosti između ćelija ili većih blokova podataka. Primenljivost ovih pristupa za krajnje korisnike često je nedovoljno ispitana. Statičke analize i izveštaji (potpoglavlje 6.3.2) zasnovane su na analizi koda, sa ciljem ustanovljenja potencijalno problematičnih delova spredšita. Ovi pristupi uključuju zaključivanje na osnovu jedinica i tipova koji prikupljaju informacije o elementima

modela na osnovu zaglavlja, spredšitove sumnjive strukture orijentisane ka detekciji loše strukture koda i komercijalne alate. Osnovna ideje spomenutih metoda se značajno razlikuju od ideje razmatrane u disertaciji, zbog čega poređenje nije relevantno.

Testiranje spredšitova (potpoglavlje 6.3.3) obuhvata testiranje adekvatnosti i upravljanje test slučajevima, testiranje na osnovu uslova i spredšit razvoj vođen testiranjem. Krajnji korisnici u najvećem broju slučajeva nisu zainteresovani za kreiranje test slučajeva, dok je dodavanje upozorenja na osnovu uslova dobro prihvaćeno. Upozorenja na osnovu uslova su se pokazala kao veoma korisna za detekciju grešaka različitog tipa. Upozorenja, u kontekstu spredšit aplikacija, odgovaraju preduslovima i postuslovima dozvoljenih vrednosti promenljivih u ćelijama, u formi bulovskih izraza. Uslove definišu krajnji korisnici upotrebom odgovarajućih korisnički orijentisanih alata. Postavljeni uslovi se automatski proveravaju i delimično propagiraju kroz spredšit model, u pravcu toka podataka. Detekcija grešaka koja je razmatrana u disertaciji zasnovana je na sličnom konceptu uslova koje treba da zadovolje elementi modela. Osnovna razlika novorazvijenog pristupa je u tome što se ne razmatraju samo proračunate vrednosti, već i struktura formula. Još jedna značajna razlika pristupa su predefinisana pravila, koja proističu iz načina modeliranja u skladu sa konceptom diskretnog objekta upravljanja. Na ovaj način smanjena je mogućnost pogrešnog postavljanja uslova, od strane krajnjeg korisnika.

Automatska detekcija i ispravka grešaka (potpoglavlje 6.3.4) u spredšitovima obuhvata više pristupa opisanih u radu. Pristupi ove grupe podrazumevaju različite informacije koje korisnik treba da obezbedi, kako bi proces detekcije i analize grešaka bio sproveden. Primeri takvih informacija su očekivana vrednost izlaza modela, manuelna provera tačnosti izlaznih vrednosti, indikacije ćelija koje su netačne i slično. Većina ovih metoda je evaluirana pod pretpostavkom poznatih rezultata, odnosno vrednosti izlaza modela. Praksa korišćenja spredšit modela pokazuje da ni kreatori, ni korisnici, veoma često ne mogu obezbediti potrebne informacije ili mogu uneti pogrešne vrednosti. Usled navedenog, pristup opisan u disertaciji zasnovan je na pravilima izvedenim iz karakteristika modela i ograničenja problema.



Pristupi razvoja spredšitova vođeni modelom (potpoglavlje 6.3.5) namenjeni su unapređenju kvaliteta i strukture spredšita u fazi kreiranja. Projektovanje i održavanje spredšitova (potpoglavlje 6.3.6) usmereno je na procese pre i nakon upotrebe modela. Ove dve grupe pristupa nisu pogodne za poređenje sa pristupom prikazanim u disertaciji jer im primarni cilj nije podrška korisnicima u detektovanju grešaka i analizi uzroka nastanka istih, već poboljšanje strukture spredšita, prevencija grešaka, izbegavanje pogrešnih referenci, rukovanje izuzecima i dugoročno korišćenje spredšita (izmene, refaktorisanje, ponovna upotreba formula).

## 8. Zaključak

Na osnovu analize dostupne literature, postavljenog predmeta i ciljeva istraživanja, prikazanih algoritama pristupa za detekciju i analizu grešaka u dinamičkim diskretnim modelima upravljanja zalihama razvijenim u spredšitovima, zasnovanog na karakteristikama problema i načinu modeliranja, i rezultata eksperimenta, u nastavku rada dat je pregled istraživanja, naučni i stručni doprinosi, potvrda postavljenih hipoteza i pravci budućih istraživanja.

### 8.1. Pregled istraživanja, naučni i stručni doprinosi, hipoteze

Na osnovu istraživanja prikazanih u disertaciji može se zaključiti da je jedna od najznačajnijih tema u oblasti operacionog menadžmenta upravljanje zalihama. Neki od najčešće korišćenih metodoloških pristupa za rešavanje ovog problema su matematičko modelovanje i simulacija. Kako operacioni menadžment, u užem smislu, obuhvata veoma širok dijapazon poslovnih i naučnih disciplina i oblasti, zbog fizičkog obima, ovaj rad je usmeren samo na probleme upravljanja zalihama u operacionom menadžmentu.

Jedan od veoma pogodnih načina za modelovanje i rešavanje realnih problema operacionog menadžmenta, između ostalog i problema upravljanja zalihama, je koncept diskretnog objekta upravljanja. Ovaj koncept omogućava jasno definisanje zakona ponašanja, oblasti upravljanja, ciljne funkcije i metode rešavanja modeliranog problema upravljanja, u cilju dobijanja optimalnog rešenja. Matematički aparat koji se koristi za rešavanje problema upravljanja je optimalno upravljanje diskretnim sistemima. Problemi koji se razmatraju u ovom radu mogu se modelirati i predstaviti u spredšitovima, kao simulacioni modeli objekta diskretnog upravljanja. Simulacionim modelom problema upravljanja predstavlja se objekat diskretnog upravljanja, odnosno njegov zakon ponašanja, oblast upravljanja i kriterijum efikasnosti upravljanja. Diskretnim dinamičkim simulacionim spredšit modelom se, na verodostojan način, mogu predstaviti i rešavati problemi operacionog menadžmenta sa kompleksnim

matematičkim aparatima, i oni se, na taj način, mogu relativno jednostavno primeniti u realnim sistemima.

Spredštovi su veoma pogodni za modelovanje, predstavljanje i simulaciju diskretnog objekta upravljanja. Struktura spredšita je odgovarajuća za predstavljanje različitih tipova i vrednosti promenljivih u vremenskim periodima. Sistem povezanih formula, kreiran u skladu sa elementima diskretnog objekta upravljanja, predstavlja simulacioni model. Simulacija se izvodi izborom vrednosti upravljačkih promenljivih iz jednog u naredni vremenski period, pri čemu su vrednosti svih elemenata modela u svakom periodu povezane. Implementacija koncepta diskretnog objekta upravljanja čini spredšitove još prikladnijim alatom za simulaciju, pogodnim za razumevanje i modelovanje poslovne dinamike. Takođe, pristup je u skladu sa preporukama za dizajn i projektovanje spredšit modela i aplikacija. Ipak, spredšit modeli kreirani na ovaj način su podložni nastanku grešaka, usled procesa razvoja koji uključuje prevođenje verbalnih i matematičkih izraza u spredšit formule i funkcije.

Problemi upravljanja zalihama, predstavljeni kao dinamički diskretni procesi, mogu se uspešno modelovati u spredšitovima i rešavati primenom različitih heurističkih, metaheurističkih i hibridnih metoda (Antić, 2014), (Đorđević et al., 2016). Kako spredšitovi predstavljaju alat veoma jednostavan i prigodan za upotrebu od strane krajnjih korisnika, oni se mogu posmatrati kao izuzetna veza između složenih metoda operacionih istraživanja i tehnika razvijenih u oblasti informacionih sistema, koje bi se mogle primeniti za rešavanje realnih problema u praksi.

Najpopularniji spredšit današnjice, *MS Excel*, uključuje izuzetno malo alata koji bi kreatorima omogućili izradu spredšitova bez grešaka i olakšali detekciju i analizu grešaka. Usled toga, u toku poslednje decenije, razvijeno je više pristupa povezanih sa različitim oblastima računarskih nauka i informacionih sistema, koji za cilj imaju kreiranje metoda i alata za unapređenje kvaliteta spredšitova. Kategorizacija i opis postojećih pristupa prikazan je ovoj doktorskoj disertaciji.

Postojeći pristupi za detekciju i analizu grešaka u spredšitovima nisu prilagođeni kompleksnim problemima, sa velikim brojem zavisnosti, kao što je problem upravljanja

zalihama. Takođe, pristupi koji bi mogli biti primenjeni na problem razmatran u ovom radu podrazumevaju da su izlazne vrednosti modela unapred poznate ili da korisnik može da obezbedi sve informacije o strukturi i ograničenima modela, što veoma često nije realna situacija. U skladu sa navedenim, opravdan je i neophodan razvoj algoritama, koji bi omogućili unapređenje kvaliteta spredšit modela za upravljanje zalihama, odnosno detekciju i analizu grešaka u njima.

Predmet istraživanja ove doktorske disertacije je pristup za detekciju i analizu grešaka u dinamičkim diskretnim spredšit modelima upravljanja zalihama, zasnovan na karakteristikama problema i načinu modeliranja. Pristup je razvijen i evaluiran nad modelom ekonomske količine naručivanja zaliha za više proizvoda sa ograničenim prostorom skladištenja.

U praksi, većina kompanija istovremeno upravlja zalihama više različitih proizvoda, pri čemu se od jednog ili više dobavljača poručuje više različitih proizvoda odjednom. Ovakav način poručivanja više artikala odjednom ili sinhronizacija više narudžbina, dovodi do smanjenja troškova poručivanja robe. Upravljanje zalihama sa jednim artiklom je relativno jednostavno i može se rešiti iskustvenim metodama, na osnovu podataka vremenskih serija ili metodom potpunog pretraživanja. Problem postaje značajno kompleksniji ukoliko se vrši uvođenje novih artikala, odnosno pojavljuje se veći broj dimenzija upravljanja. Takvi problemi mogu da se rešavaju samo heurističkim metodama pretraživanja i spadaju u domen NP-teških problema kombinatorne optimizacije. Na osnovu toga, kao osnovni primer za evaluaciju pristupa detekcije i analize grešaka razmatranog u ovoj doktorskoj disertaciji korišćen je problem upravljanja zalihama za više artikala sa ograničenim skladišnim prostorom, odnosno određivanja broja narudžbina u dinamičkom spredšit modelu ekonomske količine naručivanja zaliha za više artikala, primenom upravljanja diskretnim sistemima.

U cilju analize efikasnosti i skalabilnosti razvijenog pristupa korišćeni su modeli za 3, 20 i 100 proizvoda. Kao horizont posmatranja za proces upravljanja razmatrano je 7, 28 i 365 dana. Broj ćelija modela varirao je od 171 do 189243, kako bi u obzir bili uzeti spredšitovi malih, srednjih i velikih dimenzija. U modele su veštačkim putem

ugrađivane jedne, dve i više od pet grešaka različitog tipa. Korišćen je jedan ili više test slučajeva, kreiranih simulacijom ili na slučajan način. Test slučajevi koji se koriste za lokalizaciju grešaka, podržani su konceptom diskretnog objekta upravljanja, kroz izdvojenu oblast upravljanja, namenjenu simulaciji, sa domenom određenim ograničenjima modeliranog problema. Na osnovu rezultata eksperimenata zaključeno je da je uspešnost pristupa 100% u slučajevima detekcije u utvrđivanja uzroka jedne i dve greške. Rezultati koji se odnose na više od pet grešaka pokazali su da je uspešnost pristupa ne manja od 80%, čak i za modele veoma velikih dimenzija. Vreme potrebno za detekciju i analizu grešaka kreće se u intervalu od 5,2 do 255 sekundi, u zavisnosti od veličine modela i broja ugrađenih grešaka. U skladu sa rezultatima eksperimenata postojećih pristupa (Jannach, Baharloo & Williamson, 2013), (Jannach & Engler, 2010), (Jannach & Schmitz, 2014), (Hofer & Wotawa, 2014) i drugih, može se zaključiti da je pristup za detekciju i analizu grešaka zasnovan na karakteristikama problema i načinu modeliranja, prikazan u ovoj disertaciji, veoma efikasan. Skalabilnost pristupa je na visokom nivou, a vreme realizacije pristupa prihvatljivo. Specifičnost pristupa prikazanog u disertaciji, ogleda se u mogućnosti primene bez unapred poznatih tačnih vrednosti izlaznih promenljivih. Pretpostavka poznavanja izlaznih vrednosti modela, na kojoj se bazira većina postojećih pristupa za analizu i detekciju grešaka u spredšitovima, onemogućava primenu modela u praksi, s obzirom da se modeli upravo koriste za određivanje nepoznatih izlaznih vrednosti. Pristup detekcije i analize grešaka zasnovan na karakteristikama problema i načinu modeliranja, pored ispitivanja vrednosti uključuje i proveru strukture formula.

U odnosu na slučajno izabrane spredšitove, obično korišćene za evaluaciju postojećih pristupa, koji nisu detaljno opisani, niti je struktura i namena modela jasna, ova doktorska disertacija razmatra konkretne spredšit modele sa jasnom specifikacijom. Problemi upravljanja zalihama predstavljaju realne probleme, za koje se može odrediti opseg očekivanih izlaznih vrednosti, što se često javlja kao problem kod veštački kreiranih test slučajeva, gde se očekivani izlaz vrlo često ne može definisati. Pošto su opisani modeli zasnovani na matematičkim modelima, izlazna vrednost, kao uporedna vrednost za testiranje nekog novog pristupa detekcije i analize grešaka, može se odrediti u jednostavnijim primerima, kako manuelno, tako i automatski, određenim programom.

Proces izvođenja i rezultati eksperimenata su dokumentovani, u cilju kreiranja *banchmark*-a za buduća istraživanja. Novorazvijeni pristup je korisnički orijentisan, u skladu sa činjenicom da kreatori i korisnici spredšitova nisu profesionalni programeri.

U toku izrade ove doktorske disertacije korišćen je veći broj naučnih metoda istraživanja. Na početku disertacije prikazani su naučni rezultati u oblasti klasifikacije problema upravljanja zalihama, načina njihovog modeliranja i rešavanja, kao i pristupa automatizovanog obezbeđenja kvaliteta spredšitova, odnosno pristupa za detekciju i analizu grešaka u spredšitovima, primenom naučnih metoda sistematizacije i klasifikacije naučnih saznanja. U toj fazi korišćene su i metode prikupljanja, dedukcije, analize i upoređivanja. Na osnovu kritičke analize postojećih rezultata dedukcijom se došlo do uočavanja ključnih problema za koje postojeći pristupi ne nude odgovarajuća rešenja. U sledećoj fazi istraživanja učinjen je pokušaj da se uočeni problem formalizuje i da se prepoznaju metode koje se mogu prilagoditi za rešavanje tako definisanog problema. Na osnovu karakteristika problema i prepoznatih metoda kreiran je originarni pristup za detekciju i analizu grešaka u dinamičkim diskretnim modelima upravljanja zalihama. U ovoj fazi korišćene su metode modelovanja, metode i tehnike savremenih pristupa za detekciju i analizu grešaka iz oblasti softverskog inženjerstva i spredšit inženjerstva, kao i teorije algoritama. U trećoj fazi istraživanja razvijeno je prototipsko rešenje kojim se omogućava implementacija predloženih metoda. Implementacija je realizovana korišćenjem programskog jezika *Visual Basic for Application*. Kao spredšit alat za razvoj modela korišćen je *MS Excel*. Konačno, u završnoj fazi istraživanja, izvršena je eksperimentalna provera predloženog pristupa simulacijom u laboratorijskim uslovima, nad poznatim spredšit modelima problema upravljanja zalihama, u koji su greške veštački ugrađene. Greške su ugrađene u spredšit model metodom testiranja mutacijom. Kao metrike za evaluaciju pristupa, koji je opisan u ovom radu, razmatrani su procenat detektovanih i ispravljenih grešaka, kao i vreme potrebno za izvršenje algoritama. U cilju provere skalabilnosti pristupa, razmatrani su modeli različitih dimenzija. Broj ćelija spredšit modela, odnosno formula matematičkog modela, povećavan je, dodavanjem proizvoda koji se naručuju i povećavanjem vremenskih perioda u kojima se naručivanje sprovodi.

Na osnovu pregleda literature, implementacije razvijenog pristupa, sprovedenih eksperimenata i njihovih rezultata prikazanih u ovoj doktorskoj disertaciji, može se zaključiti da su postavljeni ciljevi, naučni i opšti, ostvareni:

- Kreiran je novi pristup za obezbeđenje višeg kvaliteta dinamičkih diskretnih modela upravljačkih problema operacionog menadžmenta, konkretno upravljanja zalihama, razvijenih u spredšitovima, razvojem algoritama za detekciju i analizu grešaka u navedenim modelima.
- Primenom ovog pristupa omogućava se podizanje kvaliteta spredšit modela za upravljanje zalihama, jer je kreirani alat, koji uključuje metode operacionih istraživanja, tehnike razvijene u oblasti informacionih sistema i spredšit inženjerstva, pogodan za upotrebu od strane krajnjih korisnika, i može se primeniti za rešavanje realnih problema u praksi.

Hipoteze, postavljene u doktorskoj disertaciji, su testirane na osnovu pregleda literature, implementacije razvijenog pristupa, sprovedenih eksperimenata i njihovih rezultata.

H1: *Problemi upravljanja zalihama modelovani kao dinamički diskretni procesi upravljanja u spredšitu podložni su nastanku grešaka, zbog kompleksnog procesa razvoja strukture i međuzavisnosti elemenata modela.*

Tvrđnja da je probleme operacionog menadžmenta moguće modelovati kao dinamičke diskretne procese upravljanja u spredšitu, potvrđena je u petom poglavlju, gde su problemi određivanja ekonomične količine naručivanja zaliha modelovani kao objekti diskretnog upravljanja u spredšitu. Na osnovu analize modela jasno je da ovakvo modelovanje problema podrazumeva kompleksni proces razvoja strukture i međuzavisnosti elemenata modela, kao i prevođenje složenih matematičkih modela u spredšit formule i funkcije, usled čega je mogućnost nastanka grešaka izuzetno velika. Takođe, kao što je prikazano u šestom poglavlju, brojna istraživanja svedoče o mogućnosti nastanka grešaka u spredšit modelima i aplikacijama, i posledično velikim finansijskim gubicima u kompanijama. Dodatno, krajnji korisnici spredšit modela su stručnjaci u svojim oblastima poslovanja, ali ne i profesionalni, obučeni programeri, što povećava

mogućnost nastanka greške pri kreiranju i korišćenju modela. Greške se vrlo često javljaju u toku korišćenja, zbog nerazumevanja modela od strane krajnjeg korisnika ili slučajno, u skladu sa ljudskom prirodom. Sve napred rečeno predstavlja dokaze da je *prva posebna hipoteza potvrđena*.

H2: *Postoji širok skup potencijalnih uzroka grešaka u problemima upravljanja zalihama, modelovanim kao dinamički diskretni procesi upravljanja u spređšitu, iz koga je potrebno izdvojiti prave uzroke.*

U skladu sa velikim brojem međusobno povezanih promenljivih, ograničenja problema koje treba modelirati u spređšitu, ali i činjenice da stručnjaci iz različitih poslovnih oblasti nisu profesionalni programeri, da im rad sa složenim softverskim paketima nije blizak, a često im ni metodologije i postupci kreiranja matematičkih modela i primene matematičkih metoda nisu poznati, jasno je da postoji širok skup potencijalnih uzroka grešaka u problemima operacionog menadžmenta, modelovanim kao dinamički diskretni procesi upravljanja u spređšitu, iz koga je potrebno izdvojiti prave uzroke, što je i pretpostavljeno drugom posebnom hipotezom. Kako je prikazano u sedmom poglavlju, liste prethodnika ili skupovi prethodnika, potencijalni uzroci grešaka, određuju se kao svi elementi formule u ćeliji u kojoj je detektovana greška. Kada se greška u modelu detektuje, sledeći korak predstavlja određivanje uzroka greške. Određivanje uzroka greške podrazumeva utvrđivanje elementa i vremenskog perioda u kome je greška nastala, kao i tipa greške. Najjednostavniji ali i najređi slučaj je da se uzrok greške otkriva u ćeliji u kojoj je greška i detektovana. Značajno češći slučaj je da se uzrok greške nalazi u nekoj od ćelija prethodnika i prenosi do ćelije u kojoj je greška detektovana. Kompleksne zavisnosti elemenata DDMUZRS impliciraju širok skup prethodnika za ćelije u kojima je greška detektovana. Stablo prethodnika mora biti orezano, kako bi se uzrok nastanka greške odredio u prihvatljivom vremenu. Pravila orezivanja i procedura provere potencijalno pogrešnih ćelija prikazani su u ovoj disertaciji u potpoglavlju 7.2 . Ovime je *druga posebna hipoteza potvrđena*.



H3: *Primenom postojećih, opštih pristupa za detekciju i analizu grešaka u spreadšitovima, ne može se pouzdano i računarski efikasno identifikovati podskup pravih uzroka grešaka u problemima modelovanim kao dinamički diskretni procesi upravljanja.*

Postojeći pristupi za obezbeđenje kvaliteta modela u spreadšitovima, prikazani u šestom poglavlju rada, pokazuju da se njihovom primenom na probleme upravljanja zalihama, modelovane kao dinamičke diskretne procese upravljanja u spreadšitu, ne može pouzdano i računarski efikasno identifikovati podskup pravih uzroka grešaka. Nijedan od pristupa nije primenjen na ovakav tip složenih problema kompleksno modelovanih u spreadšitovima. Postojeći pristupi podrazumevaju različite informacije koje korisnik treba da obezbedi, kako bi proces detekcije i analize grešaka bio sproveden. Primeri takvih informacija su očekivana vrednost izlaza modela, manuelna provera tačnosti izlaznih vrednosti, indikacije ćelija koje su netačne i slično. Većina metoda je evaluirana pod pretpostavkom poznatih rezultata, odnosno poznatih vrednosti izlaza iz modela. Praksa korišćenja spreadšit modela pokazuje da ni kreatori, ni korisnici veoma često ne mogu da obezbede potrebne informacije ili mogu da unesu pogrešne vrednosti. Pretpostavka poznavanja izlaznih vrednosti modela, na kojoj se bazira većina postojećih pristupa za analizu i detekciju grešaka, onemogućava primenu u praksi, s obzirom da se modeli upravo koriste za određivanje nepoznatih izlaznih vrednosti. Iznete činjenice ukazuju da je *treća posebna hipoteza potvrđena.*

H4: *Pristup zasnovan na karakteristikama problema i načinu modeliranja može da unapredi detekciju i analizu grešaka spreadšit modela.*

Na osnovu rezultata eksperimenta, koji su izvedeni u cilju evaluacije pristupa za detekciju i analizu grešaka zasnovanog da karakteristikama problema i načinu modeliranja, može se zaključiti da je ovaj pristup veoma efikasan. Skalabilnost pristupa je na visokom nivou, a vreme realizacije pristupa prihvatljivo. Specifičnost pristupa prikazanog u disertaciji, ogleda se u mogućnosti primene

bez unapred poznatih tačnih vrednosti izlaznih promenljivih. Pretpostavka poznavanja izlaznih vrednosti modela, na kojoj se bazira većina postojećih pristupa za analizu i detekciju grešaka, onemogućava primenu modela u praksi, obzirom da se modeli upravo koriste za određivanje nepoznatih izlaznih vrednosti. Pristup detekcije i analize grešaka zasnovan na karakteristikama problema i načinu modeliranja, pored ispitivanja vrednosti, uključuje i proveru strukture formula. Dodatno, ukoliko je postojanje greške u modelu detektovano, ali se njen uzrok ne može odrediti, to upućuje na grešku u postavci problema. Stoga, može se zaključiti da je *četvrta posebna hipoteza potvrđena*.

H5: *Moguće je odrediti intervale za vrednosti elemenata problema upravljanja zalihama modelovanih kao dinamički diskretni procesi upravljanja u spređitu, koji omogućavaju detekciju grešaka.*

Kao posledica povezanosti elemenata modela, postojanje grešaka se može uočiti na osnovu odstupanja vrednosti ciljne funkcije od njene očekivane vrednosti. Kako ciljna funkcija predstavlja kumulativnu vrednost ciljne funkcije u prethodnom vremenskom periodu i vrednosti ciljne funkcije u posmatranom vremenskom periodu, neophodno je odrediti vremenski period u kome se greška pojavila. Osnovni problem koji se javlja prilikom detekcije grešaka, na osnovu odstupanja vrednosti ciljne funkcije, je činjenica da kreator ili korisnik modela u realnoj situaciji ne mogu znati tačne vrednosti izlaznih promenljivih za poslednji vremenski period posmatranja ( $T$ ), kao ni međurezultate prethodnih perioda ( $t=1, \dots, T$ ). Svaka kombinacija promenljivih upravljanja ( $u_i$ ) rezultuje različitim vrednostima ciljne funkcije za  $t$  vremenskih perioda ( $t=1, \dots, T$ ) horizonta posmatranja, što dodatno otežava definisanje tačnih vrednosti rezultata. U cilju detekcije grešaka, u skladu sa pristupom koji se razvija u ovom radu, koriste se donje i gornje granice ciljne funkcije (Algoritam 2 i Algoritam 3) i pravila zasnovana na karakteristikama modela (Algoritam 4) u potpoglavlju 7.2.

Donja granica određuje se za vremenski period  $t=1$ , a gornja za  $t=T$ . Ukoliko bi se svi proizvodi naručivali u svakom periodu ( $t$ ) horizonta posmatranja, troškovi

za prvi dan bi bili najniži. Nasuprot tome, svakodnevno naručivanje prouzrokuje najviše troškove u poslednjem vremenskom periodu ( $T$ ), s obzirom da ukupni troškovi zaliha predstavljaju sumu troškova za svaki proizvod ( $i$ ) u svakom vremenskom periodu ( $t$ ). Kako bi ovo tvrđenje bilo istinito, uvodi se pretpostavka da su troškovi držanja zaliha značajno niži od troškova naručivanja.

Ukoliko je horizont posmatranja dugačak (duži od godinu dana), upravljačke promenljive koje će se koristiti za definisanje gornjih granica se mogu eksperimentalno odrediti. Upravljačke promenljive ( $u_i$ ) odnose se na broj naručivanja po svakom proizvodu  $i$ . Prilikom definisanja maksimalne gornje i minimalne donje granice ograničenje skladišnog prostora (5.20) se zanemaruje. Relaksacija ovog ograničenja garantuje da nijedna vrednost ciljne funkcije dobijena za dopustivi test slučaj, za koji važi ograničenje skladišnog prostora, ne može biti manja, odnosno veća od granica dobijenih na opisani način. Određivanje gornjih i donjih granica predstavljeno je Algoritmom 2 i Algoritmom 3 u potpoglavlju 7.2.

Donja i gornja granica, određene na opisani način, predstavljaju opšte granice i određuju dopustivu oblast svih specifičnih granica. Specifične granice su vrednosti ciljne funkcije koje ona dobija za vremenski period  $t=1$  i  $t=T$  i konkretni test slučaj. Test slučajevi se kreiraju kao kombinacije dopustivih vrednosti promenljivih upravljanja. Odstupanje specifične donje i specifične gornje granice za konkretni test slučaj od opštih granica ukazuje na postojanje greške. Postojanje greške može se uočiti i ukoliko vrednost ciljne funkcije u vremenskom periodu  $t$ , za  $1 < t < T$ , nije u opsegu definisanom specifičnim granicama.

Dodatna ograničenja, zasnovana na osobinama ciljne funkcije, mogu se primeniti u procesu detekcije grešaka. Ciljna funkcija je kumulativna vrednost, koja se sastoji od vrednosti ciljne funkcije u prethodnom vremenskom periodu i njene vrednosti u periodu posmatranja. U skladu sa ovom karakteristikom, vrednost ciljne funkcije u vremenskom periodu  $t$  mora da bude niža od vrednosti

ciljne funkcije u narednom vremenskom periodu  $t+1$ . Svaki put kada je neko od opisanih ograničenja narušeno i na taj način greška detektovana, odgovarajuća ćelija postaje kandidat za dalju analizu, odnosno puni se Lista ćelija sa mogućom greškom. Lista ćelija sa mogućom greškom odnosi se na ćelije u kojima je greška detektovana i njihove direktne prethodnike. Pored globalnih i specifičnih granica, dodatna ograničenja mogu se odrediti u odnosu na pikove (Algoritam 5, potpoglavljje 7.2). Pik predstavlja odskok zaliha u vremenskom periodu kada je narudžbina realizovana. Količnik naručenih proizvoda i dnevne tražnje mora biti jednak broju dana između dve uzastopne narudžbine. Odstupanje između ovih vrednosti ukazuje na grešku u oblasti promenljivih stanja. Naručene količine proizvoda su iste u svim vremenskim periodima u kojima pikovi postoje. Razlika u vrednosti ulaznih promenljivih podrazumeva grešku u oblasti regulatora protoka. Navedena pravila slede iz modela prikazanog u poglavlju 5.1, relacije (5.17) do (5.19). Ovime je **potvrđena** *peta posebna hipoteza*.

**H6:** *Uvođenjem dodatnih pravila o osobinama problema upravljanja zalihama, širok skup potencijalnih uzroka grešaka u spreadit modelu može se suziti na redukovani skup.*

Uvođenje dodatnih pravila o osobinama problema upravljanja zalihama omogućava detekciju grešaka u dinamičkom diskretnom spreadit modelu upravljanja zalihama. Na osnovu osobina problema, mogu se odrediti granice u kojima se moraju naći vrednosti elemenata modela. Koraci koji se realizuju u određivanju gornjih i donjih granica predstavljeni su Algoritmom 2 i Algoritmom 3, u šestom poglavlju. Specifične granice su vrednosti ciljne funkcije koje ona dobija za vremenski period  $t=1$  i  $t=T$  i konkretni test slučaj. Test slučajevi se kreiraju kao kombinacije dopustivih vrednosti promenljivih upravljanja. Odstupanje donje i gornje granice za konkretni test slučaj od opštih granica ukazuje na postojanje greške. Postojanje greške može se uočiti i ukoliko vrednost ciljne funkcije u vremenskom periodu  $t$ , za  $1 < t < T$ , nije u opsegu definisanom specifičnim granicama.

Pred globalnih i specifičnih granica, dodatna ograničenja mogu se odrediti u odnosu na pikove (Algoritam 5, potpoglavlje 7.2). Pik predstavlja odskok zaliha u vremenskom periodu kada je narudžbina realizovana. Količnik naručenih proizvoda i dnevne tražnje mora biti jednak broju dana između dve uzastopne narudžbine. Odstupanje između ovih vrednosti ukazuje na grešku u oblasti promenljivih stanja. Naručene (ulazne) količine proizvoda su iste u svim vremenskim periodima u kojima pikovi postoje. Razlika u vrednosti ulaznih promenljivih podrazumeva grešku u oblasti regulatora protoka. Dokaz pravila sledi iz modela prikazanog u poglavlju 5.1.

Na osnovu navedenih dokaza može se zaključiti da je *šesta posebna hipoteza delimično potvrđena*. Pravila o osobinama problema omogućavaju detekciju grešaka, dok se analiza, odnosno određivanja pravog uzroka iz širokog skupa potencijalnih uzroka, u velikom delu zasnovana na načinu modeliranja. Međutim, kako model predstavlja matematički prikaz problema, ne može se napraviti striktna razlika između karakteristika problema i modela, odnosno model i problem su usko povezani.

H7: *Dinamički diskretni spredšit modeli upravljanja zalihama sadrže pravila kojima se dodatno može redukovati početni širok skup potencijalnih uzroka grešaka modela.*

Ukoliko ćelija iz Liste ćelija sa mogućom greškom predstavlja prethodnika neke druge ćelije iz ove liste i obe ćelije se odnose na promenljive istog tipa (npr. promenljive stanja  $X_{t,i}$ ), onda druga detektovana ćelija i njeni prethodnici nisu kandidati za ispitivanje u aktuelnoj iteraciji. Ovo pravilo zasniva se na karakteristikama načina modeliranja u skladu sa konceptom diskretnog objekta upravljanja. Ciljna funkcija i promenljive stanja direktno su zavisne od svojih vrednosti u prethodnim vremenskim periodima. U slučaju ispitivanja promenljivih istog tipa, prioritet ima promenljiva koja se odnosi na raniji vremenski period. Prioritet ispitivanja je posledica uticaja promenljivih iz ranijih vremenskih perioda na promenljive istog tipa u kasnijim vremenskim periodima.

Ukoliko ćelija iz Liste ćelija sa mogućom greškom predstavlja prethodnika neke druge ćelije iz iste liste, ali se one ne odnose na promenljive istog tipa, onda prethodnici ćelije nisu kandidati za ispitivanje u aktuelnoj iteraciji. Ovo pravilo se odnosi na prioritet ispitivanja ćelija koje ne predstavljaju promenljive istog tipa. Ukoliko je potencijalno pogrešna ćelija prethodnik druge potencijalno pogrešne ćelije prva se ispituje roditeljska ćelija. Prioritet roditeljske ćelije određuje hijerarhijska struktura u kojoj viši nivoi imaju prednost u procesu provere. Pravilo je u skladu sa algoritmom pretrage po širini (*breadth-first search*). Dakle, može se zaključiti sa je *sedma posebna hipoteza potvrđena*.

Na osnovu svega navedenog, analize dostupne literature, postavljenog predmeta i ciljeva istraživanja, prikazanih algoritama pristupa i rezultata eksperimenta, može se zaključiti da su sve posebne hipoteze potvrđene, a samim tim i opšta hipoteza ovog rada:

H0: *Pristup detekcije i analize grešaka u dinamičkim diskretnim spredšit modelima upravljanja zalihama, zasnovan na karakteristikama problema i načina modeliranja, kao i primeni metoda operacionih istraživanja, tehnika razvijenih u oblasti informacionih sistema i spredšit inženjerstva, može da unapredi kvalitet ovih modela, odnosno da omogući detekciju grešaka i utvrđivanje njihovih uzroka na brz i efikasan način.*

potvrđena, a cilj doktorske disertacije ispunjen.

U okviru ove disertacije kao najvažniji naučni doprinosi mogu se izdvojiti:

- Sistematizacija postojećih naučnih saznanja iz oblasti upravljačkih problema operacionog menadžmenta, konkretno upravljanja zalihama i dinamičkih diskretnih upravljačkih modela.
- Prikaz procesa modeliranja upravljačkih procesa u spredšitovima, konkretno upravljanja zalihama, koji su modelirani kao dinamički diskretni modeli.

- Razvijen simulacioni model odlučivanja u spredšitu, u kome je moguće, na jednostavan i verodostojan način, prikazati matematički model diskretnih procesa operacionog menadžmenta, konkretno upravljanja zalihama.
- Sistematizacija naučnih saznanja, pregled i analiza dosadašnjih istraživanja iz oblasti informacionih sistema, računarskih nauka i spredšit inženjerstva, koja se odnose na testiranje i automatizovano obezbeđenje kvaliteta spredšit aplikacija.
- Sistematizacija postojećih naučnih saznanja u oblasti detekcije i analize grešaka u spredšitovima, uporednim prikazom i komparacijom pristupa za obezbeđenje kvaliteta spredšit modela, od vizualizacije spredšitova, preko statičkih analiza i izveštaja, testiranja, automatske detekcije i ispravke grešaka, do pristupa razvoja zasnovanih na modelima i projektovanja i podrške održavanju spredšit modela.
- Razvijen novi pristup za detekciju i analizu grešaka u dinamičkim diskretnim modelima upravljanja zalihama, razvijenim u spredšitovima, zasnovan na karakteristikama problema i načina modeliranja.

Stručni doprinos doktorske disertacije predstavlja:

- Razvijen alat za detekciju i analizu grešaka u dinamičkim diskretnim modelima upravljanja zalihama, implementacijom predloženog pristupa u spredšitu, čijom primenom se znatno olakšava korišćenje ovog pristupa i približava postojećoj stručnoj zajednici. Takođe, ovim alatom omogućen je okvir za dalju nadogradnju alata i proširenje, prouzrokovano implementacijom novih vektora ograničenja i troškova, kao i povezivanjem ili ugrađivanjem metoda i tehnika postojećih pristupa za testiranje, automatsku detekciju i ispravku grešaka u spredšit modelima i generisanje test slučajeva.

Na osnovu svega izloženog, može se zaključiti da dobijeni rezultati doktorske disertacije predstavljaju originalni naučni doprinos u odnosu na postojeće stanje i otvaraju prostor za dalja istraživanja.

## 8.2. Pravci budućih istraživanja

Budući pravci istraživanja, koji proističu iz ove doktorske disertacije, usmereni su na primenu razvijenog pristupa za detekciju i analizu grešaka, zasnovanog na karakteristikama problema i načina modeliranja, na varijacije problema upravljanja zalihama, ali i širi skup problema operacionog menadžmenta koji se mogu kvantifikovati. Proširenje postojećeg dinamičkog diskretnog spredšit modela upravljanja zalihama za više artikala može se realizovati uvođenjem novih vektora ograničenja i troškova. U ciljnu funkciju moguće je uključiti troškove transporta, troškove smanjenja upotrebne vrednosti zaliha, troškove škarta i slično, dok se u ograničenja mogu uključiti ograničenja broja dana vezanih za rokove upotrebe artikala, potrebna vremena manipulacije zalihama i drugo.

Pored proširenja skupa problema, neophodno je razmotriti i način projektovanja interaktivnog i integrisanog okruženja za detekciju i analizu grešaka, koje bi bilo prilagođeno korisnicima svih nivoa znanja. Akcenat projektovanja interaktivnog i integrisanog okruženja treba da se stavi na korisnički interfejs.

Interesantno istraživačko pitanje odnosi se na automatsko generisanje jedinstvenih i višestrukih test slučajeva, kao i negativne test slučajeve. Kako test slučajevi predstavljaju kombinacije dopustivih vrednosti upravljačkih promenljivih, specijalna, metaheuristika ili hibridna heuristika može biti razvijena za njihovo definisanje. Za kreiranje test slučajeva može se iskoristiti i koncept nekog od postojećih pristupa automatskog generisanje test slučajeva. Dodatno, novorazvijeni pristup se može unaprediti povezivanjem ili ugrađivanjem delova postojećih koncepata iz kategorija testiranja spredšitova ili automatske detekcije i ispravke grešaka u spredšitovima.

Kao napomenu, neophodno je navesti da performanse računara mogu značajno uticati na vreme izvršenja razvijenog algoritama za ovaj pristupa. Stoga, u okviru budućih istraživanja, potrebno je sprovesti eksperimente na računaru boljih performansi u odnosu na one koje su prikazane u radu.



## Literatura

1. Abraham, R., & Erwig, M. (2004). Header and Unit Inference for Spreadsheets Through Spatial Analyses. *IEEE Symposium on Visual Languages and Human Centric Computing (VL/HCC 2004)*, (pp. 165–172). Rome, Italy.
2. Abraham, R., & Erwig, M. (2005). Goal-Directed Debugging of Spreadsheets. *IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC 2005)*, (pp. 37–44). Dallas, TX, USA.
3. Abraham, R., Erwig, M., Kollmansberger, S., & Seifert, E. (2005). Visual specifications of correct spreadsheets. *IEEE Symposium on Visual Languages and Human Centric Computing (VL/HCC 2005)*, (pp. 189-196). Dallas, TX, USA.
4. Abraham, R., & Erwig, M. (2006). AutoTest: A Tool for Automatic Test Case Generation in Spreadsheets. *IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC 2006)*, (pp. 43–50). Brighton, United Kingdom.
5. Abraham, R., & Erwig, M. (2006). Inferring Templates from Spreadsheets. *28th International Conference on Software Engineering (ICSE 2006)*, (pp. 182–191). Shanghai, China.
6. Abraham, R., & Erwig, M. (2006). Type Inference for Spreadsheets. *8th ACM SIGPLAN International Conference on Principles and Practice of Declarative Programming (PPDP 2006)*, (pp. 73–84). Venice, Italy.
7. Abraham, R., & Erwig, M. (2007). UCheck: A Spreadsheet Type Checker for End Users. *Journal of Visual Languages & Computing*, *18(1)*, pp. 71–95.
8. Abraham, R., Erwig, M., & Andrew, S. (2007). A type system based on end-user vocabulary. *IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC 2007)*, (pp. 215–222). Coeur d'Alene, Idaho, USA.
9. Abraham, R., & Erwig, M. (2008). Test-driven goal-directed debugging in spreadsheets. *IEEE Symposium on Visual Languages and Human Centric Computing (VL/HCC 2008)*, (pp. 131–138). Herrsching am Ammersee, Germany.
10. Abraham, R., & Erwig, M. (2009). Mutation Operators for Spreadsheets. *IEEE Transactions on Software Engineering*, *35(1)*, pp. 94–108.
11. Abreu, R., Riboira, A., & Wotawa, F. (2012). A CSP based Approach. *23rd IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW 2012)*, (pp. 159–164). Dallas, TX, USA.

12. Abreu, R., Ribeiro, A., & Wotawa, F. (2012). Constraint-based debugging of spreadsheets. *XV Ibero-American Conference on Software Engineering (CibSE 2012)*, (pp. 1-14). Buenos Aires, Argentina.
13. Ahmad, Y., Antoniu, T., Goldwater, S., & Krishnamurthi, S. (2003). A Type System for Statically Detecting Spreadsheet Errors. *18th IEEE/ACM International Conference on Automated Software Engineering (ASE 2003)*, (pp. 174–183). Montreal, Canada.
14. Alfares, H. K. (2007). Inventory model with stock-level dependent demand rate and variable holding cost. *International Journal of Production Economics*, *108*(1-2), pp. 259–265.
15. Amoako-Gyampah, K., & Meredith, J. R. (1989). The operations management research agenda: An update. *Journal of Operations Management*, *8*(3), pp. 250-262.
16. Anderson, D., Sweeney, D., & Williams, T. (2003). *An introduction to management science - Quantitative approaches to decision making, Tenth edition*. Thomson Learning.
17. Antić, S. (2014). *Modeli i metode upravljanja zalihama zasnovani na metaheuristikama*, Doktorska disertacija, Fakultet organizacionih nauka, Beograd.
18. Antić, S. (2012). *Modeliranje procesa operacionog menadžmenta u spreadšitovima*, Magistarska teza, Fakultet organizacionih nauka, Beograd.
19. Antić, S., Đorđević, L., Kostić, K., & Lisec, A. (2015). Dynamic discrete simulation model of an inventory control with or without allowed shortages. *Scientific Bulletin Series A, Applied Mathematics and Physics*, *77*(1), pp. 163-176.
20. Antoniu, T., Steckler, P., Krishnamurthi, S., Neuwirth, E., & Felleisen, M. (2004). Validating the Unit Correctness of Spreadsheet. *26th International Conference on Software Engineering (ICSE 2004)*, (pp. 439–448). Edinburgh, United Kingdom.
21. Aurigemma, S., & Panko, R. R. (2010). The detection of human spreadsheet errors by humans versus inspection (auditing) software. *European Spreadsheet Risks Interest Group 11th Annual Conference (EuSpRIG 2010)*, Greenwich, United Kingdom. *arXiv:0807.2997v1*.
22. Ayalew, Y. (2009). A Visualization-based Approach for Improving Spreadsheet Quality. *Warm Up Workshop for ACM/IEEE ICSE 2010 (WUP 2009)*, (pp. 13–16). Cape Town, South Africa.

23. Ayalew, Y. (2001). *Spreadsheet Testing Using Interval Analysis*. PhD thesis. Austria: Klagenfurt University.
24. Ayalew, Y., & Mittermeir, R. (2003). Spreadsheet Debugging. *European Spreadsheet Risks Interest Group 4th Annual Conference (EuSpRIG 2003)*. Dublin, Ireland.
25. Ayalew, Y., Clermont, M., & Mittermeir, R. (2000). Detecting errors in spreadsheets. *European Spreadsheet Risks Interest Group Annual Conference, In Spreadsheet Risks, Audit and Development Methods, volume 1*, (pp. 51–62). University of Greenwich, London.
26. Badame, S., & Dig, D. (2012). Refactoring meets Spreadsheet Formulas. *28th IEEE International Conference on Software Maintenance (ICSM 2012)*, (pp. 399–409). Riva del Garda, Trento, Italy.
27. Ballinger, D., Biddle, R., & Noble, J. (2003). Spreadsheet visualisation to improve end-user understanding. *Proceedings of the Asia-Pacific Symposium on Information Visualisation - Volume 24 (APVIS 2003)*, (pp. 99–109). Adelaide, Australia.
28. Ballou, R. (2004). *Business Logistics & Supply Chain management, Fifth Edition*. Pearson Education Inc. New Jersey.
29. Barlow, J. (2003). *Excel models for business and Operations Management*. John Willey & Sons Ltd. West Sussex. England.
30. Beckwith, L., Burnett, M., & Cook, C. (2002). Reasoning About Many-to-Many Requirement Relationships in Spreadsheets. *IEEE Symposium on Human Centric Computing Languages and Environments*, (pp. 149-157). Arlington, VA USA.
31. Bekenn, B., & Hooper, R. (2008). Reducing Spreadsheet Risk with Formula-DataSleuth. *European Spreadsheet Risks Interest Group 9th Annual Conference (EuSpRIG 2008)*. London, United Kingdom.
32. Boltianski, V. G. (1978). *Optimal Control of Discrete Systems*. Wiley, New York.
33. Bradley, L., & McDaid, K. (2009). Using bayesian statistical methods to determine the level of error in large spreadsheets. *In Proc. of ICSE '09, Companion Volume*, (pp. 351-354). Vancouver, Canada.
34. Brath, R., & Peters, M. (2006). Excel Visualizer: One Click WYSIWYG Spreadsheet Visualization. *10th International Conference on Information Visualisation (IV 2006)*, (pp. 68–73). London, United Kingdom.

35. Burnett, M. M., Agrawal, A., & Zee, P. (2000). Exception Handling in the Spreadsheet Paradigm. *IEEE Transactions on Software Engineering* 26 (10) , pp. 923–942.
36. Burnett, M., & Erwig, M. (2002). Visually Customizing Inference Rules About Apples and Oranges,. *IEEE Symposia on Human Centric Computing Languages and Environments (HCC 2002)*, (pp. 140–148). Arlington, VA, USA.
37. Burnett, M., Cook, C., Pendse, O., Rothermel, G., Summet, J., & Wallace, C. (2003). End-User Software Engineering with Assertions in the Spreadsheet Paradigm. *25th International Conference on Software Engineering (ICSE 2003)*, (pp. 93-103). Portland, Oregon.
38. Carver, J., Fisher II, M., & Rothermel, G. (2006). An Empirical Evaluation of a Testing and Debugging Methodology for Excel. *5th ACM-IEEE International Symposium on Empirical Software Engineering*, (pp. 278-287). Rio de Janeiro, Brazil.
39. Caulkins, J., Morrison, E., & Weidemann, T. (2006). Spreadsheet errors and decision making: evidence from field interviews. *Journal of End User Computing*, 19(3), pp. 1–23.
40. Chadwick, D., Knight, B., & Rajalingham, K. (2001). Quality Control in Spreadsheets:A Visual Approach using Color Codings to Reduce Errors in Formulae. *Software Quality Control*, 9(2), pp. 133–143.
41. Chambers, C., & Erwig, M. (2009). Automatic Detection of Dimension Errors in Spreadsheets. *Journal of Visual Languages & Computing*, 20(4), pp. 269–283.
42. Chambers, C., & Erwig, M. (2010a). Reasoning About Spreadsheets with Labels and Dimensions. *Journal of Visual Languages & Computing*, 21(5), pp. 249–262.
43. Chambers, C., Erwig, M., & Luckey, M. (2010b). SheetDiff: A Tool for Identifying Changes in Spreadsheets. *IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC 2010)*, (pp. 85–92). Madrid, Spain.
44. Chang, H. (2004). A note on the EPQ Model with Shortages and Variable Lead Time. *Information and Management Sciences*, 15(1), pp. 61-67.
45. Chase, R., Jacobs, F. R., & Aquilano, N. J. (2006). *Operations Management for Competitive Advantage, 11th ed.* McGraw-Hill Irwin.

46. Chen, Y., & Chan, H. C. (2000). Visual Checking of Spreadsheets. *European Spreadsheet Risks Interest Group 1st Annual Conference (EuSpRIG 2000)*. London, United Kingdom.
47. Chi, E. H. (1999). *A Framework for Information Visualization Spreadsheets*, PhD thesis, University of Minnesota, Minnesota.
48. Chopra, S., & Meindl, P. (2004). *Supply Chain Management, Strategy, Planning, and Operations*. Pearson Education, Inc., Upper Saddle River, New Jersey.
49. Clermont, M. (2008). A Toolkit for Scalable Spreadsheet Visualization. *European Spreadsheet Risks Interest Group 5th Annual Conference (EuSpRIG 2008)*. Klagenfurt, Austria.
50. Clermont, M. (2003). Analyzing Large Spreadsheet Programs. *10th Working Conference on Reverse Engineering (WCRE 2003)*, (pp. 306–315). Victoria, BC, Canada.
51. Clermont, M. (2005). Heuristics for the Automatic Identification of Irregularities in Spreadsheets. *1st Workshop on End-User Software Engineering (WEUSE 2005)*, (pp. 1–6). St. Louis, MO, USA.
52. Cunha, J., Fernandes, J., Ribeiro, H., & Saraiva, J. (2012). Towards a Catalog of Spreadsheet Smells. *12th International Conference on Computational Science and Its Applications (ICCSA)*, (pp. 202–216). Salvador de Bahia, Brazil.
53. Cunha, J., Fernandes, J. P., & Saraiva, J. (2012). From Relational ClassSheets to UML+OCL. *27th Annual ACM Symposium on Applied Computing (SAC 2012)*, (pp. 1151–1158). Trento, Italy.
54. Cunha, J., Fernandes, J., Mendes, J., & Saraiva, J. (2012). Extension and Implementation of ClassSheet Models. *IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC 2012)*, (pp. 19–22). Innsbruck, Austria.
55. Cunha, J., Fernandes, J., Mendes, J., Pacheco, H., & Saraiva, J. (2012). Bidirectional Transformation of Model-Driven Spreadsheets. *5th International Conference on Theory and Practice of Model Transformations (ICMT 2012)*, *Springer Lecture Notes in Computer Science*, (pp. 105–120). Prague, Czech Republic.
56. Cunha, J., Mendes, J., Saraiva, J., & Fernandes, J. (2011). Embedding and Evolution of Spreadsheet Models in Spreadsheet Systems. *IEEE Symposium on*

- Visual Languages and Human-Centric Computing (VL/HCC 2011)*, (pp. 179–186). Pittsburgh, PA, USA.
57. Cunha, J., Saraiva, J., & Visser, J. (2009). From Spreadsheets to Relational Databases and Back. *ACM SIGPLAN Workshop on Partial Evaluation and Program Manipulation (PEPM 2009)*, (pp. 179–188). Savannah, GA, USA.
  58. Cunha, J., Visser, J., Alves, T., & Saraiva, J. (2011). Type-Safe Evolution of Spreadsheets. *14th International Conference on Fundamental Approaches to Software Engineering: Part of the Joint European Conferences on Theory and Practice of Software (FASE 2011/ETAPS 2011)*, (pp. 186–201). Saarbrücken, Germany.
  59. Darwish, M. A. (2008). EPQ models with varying setup cost. *International Journal of Production Economics*, 113(1), pp. 297–306.
  60. Djang, R. W., & Burnett, M. M. (1998). Similarity Inheritance: A New Model of Inheritance for Spreadsheet VPLs. *The IEEE Symposium on Visual Languages (VL 1998)*, (pp. 134–141). Halifax, NS, Canada.
  61. Đorđević, L., & Antić, S. (2014). Ocena kvaliteta analitičkog spredšit modela. *YU INFO 2014, Zbornik radova XX naučnostručne i biznis konferencije*. Kopaonik.
  62. Đorđević, L., Antić, S., Čangalović, M., & Lisec, A. (2016). A metaheuristic approach to solving a multiproduct EOQ-based inventory problem with storage space constraints. *Optimization letters*, DOI: 10.1007/s11590-016-1009-5 (First online: 18 February 2016, <http://link.springer.com/article/10.1007/s11590-016-1009-5>)
  63. Đorđević, L. (2010) Upporedni prikaz nekoliko metoda pretraživanja oblasti upravljanja objekta diskretnog upravljanja. Master rad, Fakultet organizacionih nauka, Univerzitet u Beogradu.
  64. Đorđević, L., & Vasiljević, D. (2013). Spreadsheets in education of logistics managers at Faculty of organizational sciences: an example of inventory dynamics simulation. *Proceedings of the 7th International Technology Education and Development Conference INTED 2013*, ISBN: 978-84-616-2661-8, pp. 640-649, 4-6. III 2013. Valencia, Spain.
  65. Engels, G., & Erwig, M. (2005). ClassSheets: Automatic Generation of Spreadsheet Applications from Object-Oriented Specifications. *20th IEEE/ACM International*

- Conference on Automated Software Engineering (ASE 2005)*, (pp. 124–133). Long Beach, CA, USA.
66. Erwig, M. (2009). Software engineering for spreadsheets. *IEEE Software*, 26(5), 25–30.
  67. Erwig, M., & Burnett, M. (2002). Adding Apples and Oranges. *International Symposium on Practical Aspects of Declarative Languages (PADL 2002)*, (pp. 173–191). Portland, OR, USA.
  68. Erwig, M., Abraham, R., Cooperstein, I., & Kollmansberger, S. (2005). Automatic Generation and Maintenance of Correct Spreadsheets. *27th International Conference on Software Engineering (ICSE 2005)*, (pp. 136–145). St. Louis, MO, USA.
  69. Erwig, M., Abraham, R., Kollmansberger, S., & Cooperstein, I. (2006). Gencil: A Program Generator for Correct Spreadsheets. *Journal of Functional Programming*, 16(3), pp. 293–325.
  70. Evans, J. R. (2000). Spreadsheets as a tool for teaching simulation. *INFORMS Transactions on Education*, 1(1), pp. 27–37.
  71. Felfernig, A., Friedrich, G., Jannach, D., & Stumptner, M. (2004). Consistency-based diagnosis of configuration knowledge bases. *Artificial Intelligence* 152 (2), pp. 213–234.
  72. Fisher II, M., Jin, D., Rothermel, G., & Burnett, M. (2002). Test Reuse in the Spreadsheet Paradigm. *International Symposium on Software Reliability Engineering*, (pp. 257–268). Annapolis, Maryland.
  73. Fisher, M. I., Cao, M., Rothermel, G., Cook, B. C., & Burnett, M. (2002). Automated Test Case Generation for Spreadsheets. *24th International Conference on Software Engineering (ICSE 2002)*, (pp. 141–151). Orlando, FL, USA.
  74. Fisher, M. I., Rothermel, G., Brown, D., Cao, M., Cook, C., & Burnett, M. (2006). Integrating Automated Test Generation into the WYSIWYT Spreadsheet Testing Methodology. *ACM Transactions on Software Engineering and Methodology*, 15(2), pp. 150–194.
  75. Fowler, M. (1999). *Refactoring: Improving the Design of Existing Code*. Addison-Wesley Professional.

76. Grossman, T. A. (2008). A Primer on Spreadsheet Analytics. *European Spreadsheet Risks Interest Group 9<sup>th</sup> Annual Symposium (EuSpRIG 2008)*. Greenwich, England.
77. Grossman, T. A., & Özlük, Ö. (2001). A Paradigm for Spreadsheet Engineering Methodologies. *European Spreadsheet Risks Interest Group 5th Annual Symposium (EuSpRIG 2001)*. Klagenfurt, Austria.
78. Grossman, T. A., Mehrotra, V., & Johncharl, S. (2011). Towards Evaluating the Quality of a Spreadsheet: The Case of the Analytical Spreadsheet Model. *arXiv preprint arXiv:1111.6907*.
79. Grossman, T. A., Mehrotra, V., & Özlük, Ö. (2007). Lessons from Mission-Critical Spreadsheets. *Communications of the Association for Information Systems*. Vol. 20, Article 60.
80. Grossman, T. A., Mehrotra, V., & Sander, J. C. (2011). Towards Evaluating the Quality of a Spreadsheet: The Case of the Analytical Spreadsheet Model. *European Spreadsheet Risks Interest Group 12<sup>th</sup> Annual Symposium (EuSpRIG 2011)*. London, United Kingdom.
81. Harris, F. W. (1913). How many parts to make at once» Factory. *The Magazine of Management*, 10(2), pp. 135–136.
82. Harris, F. W. (1915). *Operations and costs*. New York.
83. Harris, W. R., & Gulwani, S. (2011). Spreadsheet Table Transformations from Examples. *The 32nd ACM SIGPLAN conference on Programming language design and implementation (PLDI 2011)*, (pp. 317–328). San Jose, CA, USA.
84. Harutyunyan, A., Borradaile, G., Chambers, C., & Scaffidi, C. (2012). Planted model evaluation of algorithms for identifying differences between spreadsheets. *IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC 2012)*, (pp. 7–14). Innsbruck, Austria.
85. Hermans, F. F. (2013). *Analyzing and visualizing spreadsheets*. Netherlands: Software Engineering Research Group, Delft University of Technology.
86. Hermans, F., Pinzger, M., & Deursen, A. (2010). Automatically Extracting Class Diagrams from Spreadsheets. *24th European Conference on Object-Oriented Programming (ECOOP 2010)*, (pp. 52–75). Maribor, Slovenia.
87. Hermans, F., Pinzger, M., & Deursen, A. (2011a). Supporting Professional Spreadsheet Users by Generating Leveled Dataflow Diagrams. *33rd International*



- Conference on Software Engineering (ICSE 2011)*, (pp. 451–460). Waikiki, Honolulu, HI, USA.
88. Hermans, F., Pinzger, M., & Deursen, A. (2011b). Breviz: Visualizing Spreadsheets using Dataflow Diagrams. *Proceedings of the European Spreadsheet Risks Interest Group 12th Annual Conference (EuSpRIG 2011)*. London, United Kingdom.
  89. Hermans, F., Pinzger, M., & Deursen, A. (2012a). Detecting and Visualizing Inter-Worksheet Smells in Spreadsheets. *34th International Conference on Software Engineering (ICSE 2012)*, (pp. 441–451). Zurich, Switzerland.
  90. Hermans, F., Pinzger, M., & Deursen, A. (2012b). Detecting Code Smells in Spreadsheet Formulas. *28th IEEE International Conference on Software Maintenance (ICSM 2012)*, (pp. 409–418). Riva del Garda, Trento, Italy.
  91. Hermans, F., Sedee, B., Pinzger, M., & Deursen, A. (2013). Data Clone Detection and Visualization in Spreadsheets. *35th International Conference on Software Engineering (ICSE 2013)*, (pp. 292–301). San Francisco, CA, USA.
  92. Hesse, R. (2005). *Managerial Spreadsheet Modeling and Analysis*. Richard D. Irwin.
  93. Hipfl, S. (2004). Using Layout Information for Spreadsheet Visualization. *European Spreadsheet Risks Interest Group 5<sup>th</sup> Annual Conference (EuSpRIG 2004)*. Klagenfurt, Austria.
  94. Hodnigg, K., & Mittermeir, R. T. (2008). Metrics-Based Spreadsheet Visualization: Support for Focused Maintenance. *European Spreadsheet Risks Interest Group 9<sup>th</sup> Annual Conference, EuSpRIG 2008*, (pp. 79–94). London, United Kingdom.
  95. Hofer, B., & Wotawa, F. (2014). On the Usage of Dependency-based Models for Spreadsheet Debugging. *First workshop on Software Engineering methods in Spreadsheets (SEmS '14)*, paper 5. Delft, Holandija.
  96. Hofer, B., Ribeiro, A., Wotawa, F., Abreu, R., & Getzner, E. (2013). On the empirical evaluation of fault localization techniques for spreadsheets. *16th International Conference, FASE 2013, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2013* (pp. 68-82). Rome, Italy: Springer Berlin Heidelberg.
  97. Hopp, W., & Spearman, M. (1996). *Factory Physics*. Irwin. Ohio.

98. Howe, H., & Simkin, M. G. (2006). Factors affecting the ability to detect spreadsheet errors. *Decision Sciences Journal of Innovative Education*, 4(1), pp. 101–122.
99. Hughes, C., & Moshell, J. (1990). Action Graphics: A Spreadsheet-based Language for Animated Simulation. *Visual Languages and Applications*, (T. Ichikawa, E. Jungert, R. Korfhage, eds.). New York: Plenum Pub., pp. 203-235.
100. Isakowitz, T., Schocken, S., & Lucas, H. C. (1995). Toward a Logical/Physical Theory of Spreadsheet Modeling. *Trans.on Information Systems*, 13(1), pp. 1-37.
101. Jackson, R. (1993a). A spreadsheet approach to complex transformations. *Teaching Maths and Its Applications*, No.12, pp. 174.
102. Jackson, R. (1993b). Modelling population growth using a spreadsheet. *Teaching Maths and its Applications*, No.12, pp. 74.
103. Jackson, R., & Callender, J. T. (1994). Estimators: a spreadsheet investigation of bias. *Teaching Maths and Its Applications*, No.13, pp. 78.
104. Jannach, D., & Engler, U. (2010). Toward model-based debugging of spreadsheet programs. *9th Joint Conference on Knowledge-Based Software Engineering (JCKBSE 2010)*, (pp. 252–264). Kaunas, Lithuania.
105. Jannach, D., & Schmitz, T. (2014). Model-based diagnosis of spreadsheet programs: a constraint-based debugging approach. *Automated Software Engineering*, 23(1), pp. 105-144.
106. Jannach, D., Baharloo, A., & Williamson, D. (2013). Toward an integrated framework for declarative and interactive spreadsheet debugging. *8th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2013)*, (pp. 117–124). Angers, France.
107. Jannach, D., Schmitz, T., Hofer, B., & Wotawa, F. (2014). Avoiding, Finding and Fixing Spreadsheet Errors-A Survey of Automated Approaches for Spreadsheet QA. *Journal of Systems and Software*. Vol. 94, pp. 129-150.
108. Jelen, B. (2005). *The Spreadsheet at 25: 25 Amazing Excel Examples that Evolved from the Invention that Changed the World*. Holy Macro! Books.
109. Kankuzi, B., & Ayalew, Y. (2008). An End-User Oriented Graph-Based Visualization for Spreadsheets. *4th International Workshop on End-User Software Engineering (WEUSE 2008)*, (pp. 86–90). Leipzig, Germany.

110. Kohler, E. (1963). *Dictionary for Accountants, 3rd ed.* Prentice-Hall.
111. Kostić, K. (2009). Inventory control as a discrete system control for the fixed-order quantity system. *Applied Mathematical Modelling*, 33(11), pp. 4201-4214.
112. Kostić, K. (2008a). *Izrada i korišćenje poslovnih modela.* Fakultet organizacionih nauka, Beograd.
113. Kostić, K. (2008b). *Simulacija biznis situacija – Primeri iz prakse.* Fakultet organizacionih nauka, Beograd.
114. Krčevinac, S., Čangalović, M., Kovačević-Vujčić, V., Martić, M., & Vujošević, M. (2004). *Operaciona istraživanja.* Fakultet organizacionih nauka, Beograd.
115. Kruck, S. E. (2006). Testing spreadsheet accuracy theory. *Information & Software Technology*, 48(3), pp. 204-213.
116. Lawrence, J., & Pasternack, B. (2002). *Applied Management Science – Modeling, Spreadsheet Analysis, and Communication for Decision Making, Second Edition.* John Wiley & Sons. San Francisco.
117. Lawson, B. R., Baker, K. R., Powell, S. G., & Foster-Johnson, L. (2009). A comparison of spreadsheet users with different levels of experience. *Omega*, 37(3), pp. 579-590.
118. Lewis, C. (1990). NoPumpG: Creating Interactive Graphics with Spreadsheet Machinery. *Visual Programming Environments: Paradigms and Systems*, CA: IEEE Computer Society Press, (pp. 526–546). Los Alamitos.
119. Liao, J. J. (2007). On EPQ model for deteriorating items under permissible delay in payments. *Applied Mathematical Modelling*, 31(3), pp. 393-403.
120. Luckey, M., Erwig, M., & Engels, G. (2012). Systematic Evolution of Model-Based Spreadsheet Applications. *Journal of Visual Languages & Computing*, 23(5), pp. 267–286.
121. Mateis, C., Stumptner, M., Wieland, D., & Wotawa, F. (2000). Model-Based Debugging of Java Programs. *Fourth International Workshop on Automated Debugging (AADEBUG 2000)*. Munich, Germany.
122. Mattessich, R. (1964a). *Accounting and Analytical Methods.* Illinois: Irwin Inc. Homewood.
123. Mattessich, R. (1964b). *Simulation of the Firm Through a Budget Computer Program.* Illinois: Irwin Inc. Homewood.

124. McDaid, K., Rust, A., & Bishop, B. (2008). Test-Driven Development: Can it Work for Spreadsheets?. *4th International Workshop on End-User Software Engineering (WEUSE 2008)*, (pp. 25–29). Leipzig, Germany.
125. Mentzer, J. T., Stank, T. P., & Esper, T. L. (2008). Supply Chain Management and Its Relationship to Logistics, Marketing, Production, and Operations Management. *Journal of Business Logistics*, 29(1), pp. 31–46.
126. Mittermeir, R., & Clermont, M. (2002). Finding High-Level Structures in Spreadsheet Programs. *9th Working Conference on Reverse Engineering (WCRE 2002)*, (pp. 221–232). Richmond, VA, USA.
127. Montigel, M. (2002). Portability and Reuse of Components for Spreadsheet Languages. *The IEEE CS International Symposium on Human-Centric Computing Languages and Environments (HCC 2002)*, (pp. 77–79). Arlington, VA, USA.
128. Nardi, B. A., & Miller, J. R. (1990). *The Spreadsheet Interface: A Basis for End User Programming. Technical Report HPL-90-08*. Hewlett-Packard Software Technology Laboratory.
129. Nixon, D., & O'Hara, M. (2001). Spreadsheet Auditing Software. *European Spreadsheet Risks Interest Group 2nd Annual Conference (EuSpRIG 2001)*. Amsterdam, Netherlands.
130. O'Beirne, P. (2010). Spreadsheet Refactoring. *European Spreadsheet Risks Interest Group 11th Annual Conference (EuSpRIG 2010)*. London, UK.
131. Oke, S. A. (2004). Spreadsheet Applications in Engineering Education: A Review. *International Journal of Engineering Education* 20(6) , pp. 893±901.
132. Omerbegović-Bijelović, J. i drugi (2010). *Osnove Operacionog menadžmenta*. Fakultet organizacionih nauka, Beograd.
133. Paine, J. (2005). Bringing the Benefits of Modularisation to Excel. *European Spreadsheet Risks Interest Group 6th Annual Conference (EuSpRIG 2005)*. London, United Kingdom.
134. Paine, J. (2001). Ensuring Spreadsheet Integrity with Model Master. *European Spreadsheet Risks Interest Group 2nd Annual Conference (EuSpRIG 2001)*. Amsterdam, Netherlands.

135. Paine, J. (1997). Model Master: an object-oriented spreadsheet front-end. *CALECO Conference on Using Computer Technology in Economics and Business (CALECO 1997)*, (pp. 84–92). Bristol, United Kingdom.
136. Paine, J. (2004). Spreadsheet Structure Discovery with Logic Programming. *European Spreadsheet Risks Interest Group 5th Annual Conference (EuSpRIG 2004)*. Klagenfurt, Austria.
137. Paine, J., Tek, E., & Williamson, D. (2006). Rapid Spreadsheet Reshaping with Excelsior: multiple drastic changes to content and layout are easy when you represent enough structure. *European Spreadsheet Risks Interest Group 7th Annual Conference (EuSpRIG 2006)*. Cambridge, United Kingdom.
138. Panko, R. (2000a). A Human Error Website:  
<http://panko.shidler.hawaii.edu/HumanErr/Index.htm>. Preuzeto 19. 11. 2014.
139. Panko, R. (2000b). Spreadsheet Research Website:  
<http://panko.shidler.hawaii.edu/SSR/index.htm>. Preuzeto 19.11 2014.
140. Panko, R. R. (2005). Sarbanes–Oxley: What about All the Spreadsheets? Controlling for Errors and Fraud in Financial Reporting. *European Spreadsheet Research Information Group, EuSpRIG'05*. University of Greenwich. London, UK
141. Panko, R. R. (2006). Spreadsheets and Sarbanes–Oxley: Regulations, Risks, and Control Frameworks. *Communications of the AIS* .
142. Panko, R. R. (1998). What we know about spreadsheet errors. *Journal of End User Computing*, 10(2), pp. 15–21.
143. Panko, R. R., & Aurigemma, S. (2010). Revising the panko-halverson taxonomy of spreadsheet errors. *Decision Support System*, 49(2), pp. 235-244.
144. Panko, R. R., & Halverson, R. P. (1996). Spreadsheets on trial: A survey of research on spreadsheet risks. *Proceedings of the 29th Annual Hawaii International Conference on System Sciences*, (pp. 326-336). Hawaii, SAD.
145. Panko, R. (2005). Sarbanes–Oxley: What about All the Spreadsheets? Controlling for Errors and Fraud in Financial Reporting. *European Spreadsheet Risks Interest Group 6th Annual Conference (EuSpRIG 2005)*, University of Greenwich, London, UK.
146. Panko, R. (1998). What we know about spreadsheet errors. *Journal of End User Computing*, 10(2), pp. 15-21.

147. Panko, R. (2008). *What we know about spreadsheet errors*.  
<http://panko.shidler.hawaii.edu/MyPublications/Whatknow.htm>.  
 Preuzeto 12.11.2014.
148. Pawar, K. (2011). *Developments in operations management*. University of Nottingham.
149. Pennirselvam, G. P., Ferguson, L. A., & Ash, R. C. (1999). Operations management research: An update for the 1990s. *Journal of Operations Management*, 18(1), pp. 95-112.
150. Phan, T. T. (2003). Technical considerations for the evaluation of electronic spreadsheets for complying with 21 CFR Part II. *Pharmaceutical Technology*, no. January, pp. 50-62.
151. Powell, S. G., & Baker, K. R. (2010). *Management Science: The Art of Modeling with Spreadsheets 3e*. John Wiley&Sons.
152. Powell, S. G., Baker, K. R., & Lawson, B. (2008). A Critical Review of the Literature on Spreadsheet Errors. *Decision Support Systems*, 46(1), pp. 128-138.
153. Powell, S. G., Baker, K. R., & Lawson, B. (2009). Errors in Operational Spreadsheets. *Journal of Organizational and End User Computing*, 21(3), pp. 24-36.
154. Powell, S. G., Baker, K. R., & Lawson, B. (2009). Impact of errors in operational spreadsheets. *Decision Support Systems*, 47(2), pp. 126-132.
155. Power, D. J. (2004). *A Brief History of Spreadsheets*. DSSResources.COM, v3.6.
156. PriceWaterhouseCoopers. (2004). *The Use of Spreadsheets: Considerations for Section 404 of the Sarbanes-Oxley Act*. PriceWaterhouseCoopers, London.
157. Ragsdale, C. T. (2010). *Spreadsheet Modeling and Decision Analysis*. South-Western College Publishing.
158. Rajalingham, K. (2005). A Revised Classification of Spreadsheet Errors. *European Spreadsheet Risks Interest Group 6th Annual Conference (EuSpRIG 2005)*, (pp. 185-199). Greenwich, London.
159. Rajalingham, K., Chadwick, D., & Knight, B.(2000). Classification of spreadsheet errors. *European Spreadsheet Risks Interest Group Annual Conference*, (pp. 23–34). Greenwich, England.
160. Rao, R., & Card, S. K. (1994). The Table Lens: Merging Graphical and Symbolic Representations in an Interactive Focus+Context Visualization for Tabular

- Information. *SIGCHI Conference on Human Factors in Computing Systems (CHI 1994)*, (pp. 318–322). Boston, MA, USA.
161. Reiter, R. (1987). Theory of Diagnosis from First Principles. *Artificial Intelligence*, 32 (1), pp. 57–95.
  162. Robinson, P. E., & Sahin, F. (2001). Economic Production Lot Sizing with Periodic Costs and Overtime. *Decision Sciences*, 32(3), pp. 423-452.
  163. Robinson, P., Narayanan, A., & Sahin, F. (2009). Coordinated deterministic dynamic demand lot-sizing problem: A review of models and algorithms. *Omega*, 37(1), pp. 3-15.
  164. Ronen, B., Palley, M., & Halverson, L. (1989). Spreadsheet analysis and design. *Communication of the ACM*, 32(1), pp. 84-93.
  165. Rothermel, G., Dupuis, L. L., & Burnett, M. (1998). What You See Is What You Test: A Methodology for Testing Form-Based Visual programs. *20th International Conference on Software Engineering, ICSE 1998*, (pp. 198–207). Kyoto, Japan.
  166. Rothermel, G., Lixin, L., & Burnett, M. (1997). Testing strategies for form-based visual programs. *Eighth International Symposium on Software Reliability Engineering*, (pp. 96-107). Albuquerque, NM.
  167. Rothermel, K., Cook, C., Burnett, M., & Schonfeld, J. (2000). WYSIWYT testing in the spreadsheet paradigm: An empirical evaluation. *International Conference on Software Engineering (ICSE)*, (pp. 230–239). Limerick, Ireland.
  168. Ruthruff, J. R., Burnett, M., & Rothermel, G. (2006). Interactive Fault Localization Techniques in a Spreadsheet Environment. *IEEE Transactions on Software Engineering*, 32(4), pp. 213–239.
  169. Ruthruff, J. R., Prabhakararao, S., Reichwein, J., Cook, C., Creswick, E., & Burnett, M. (2005). Interactive, Visual Fault Localization Support for End-User Programmers. *Journal of Visual Languages & Computing*, 16(1), pp. 3–40.
  170. Ruthruff, J., Burnett, M., & Rothermel, G. (2005). An Empirical Study of Fault Localization for End-User Programmers. *27th International Conference on Software Engineering*, (pp. 352-361). St. Louis, MO USA.
  171. Ruthruff, J., Creswick, E., Burnett, M., Cook, C., Prabhakararao, S., Fisher, I., i drugi. (2003). End-User Software Visualizations for Fault Localization. *ACM Symposium on Software Visualization*. San Diego, CA USA.

172. Sajaniemi, J. (2000). Modeling Spreadsheet Audit: A Rigorous Approach to Automatic Visualization. *Journal of Visual Languages & Computing*, 11(1), 49-82.
173. Salvendy, G. (2001). *Handbook of industrial engineering: technology and operations management*. John Wiley & Sons.
174. Sana, S. S., & Chaudhuri, K. S. (2007). A deterministic EOQ model with delays in payments and price-discount offer. *European Journal of Operational Research*, 184(2), pp. 509–533.
175. Scaffidi, C., Shaw, M., & Myers, B. A. (2005). Estimating the numbers of end users and end user programmers. In *Proc. of IEEE Symp. on Visual Languages and Human-Centric Computing '05*, (pp. 207-214). Dallas, Texas.
176. Schroeder, R. G. (2007). *Operations management: Contemporary Concepts and cases, Third edition*. McGraw-Hill International Editions.
177. Sebesta, R. W. (1999). *Concepts of Programming Languages (4th ed.)*. Addison-Wesley-Longman.
178. SERP. (2006). *SERP (2006). Spreadsheet Engineering Research Project*. Tuck Business School, Dartmouth College, USA.  
<http://mba.tuck.dartmouth.edu/spreadsheet>. Preuzeto 16. 11. 2014.
179. Shiozawa, H., Okada, K., & Matsushita, Y. (1999). 3D Interactive Visualization for Inter-Cell Dependencies of Spreadsheets. *IEEE Symposium on Information Visualization (Info Vis 1999)*, (pp. 79-82, 148). San Francisco, CA, USA.
180. Shtub, A. (1999). *Enterprise Resource Planning, The Dynamics of Operations Management*. Hingham: Kluwer Academic Publishers.
181. Slater, R. (1989). *Portraits In Silicon*. The MIT Press.
182. *Spreadsheet Analytics*. (2013). <http://www.usfca.edu/bps/spreadsheet-analytic>  
Preuzeto 17. 11. 2014.
183. Teng, J. T., Chern, M. S., & Chan, Y. L. (2005). Deterministic inventory lot-size models with shortages for fluctuating demand and unit purchase cost. *Journal of International transaction in operational research* 12(1), pp. 83–100.
184. Teng, J. T., Ouyang, L. Y., & Chang, C. T. (2005). Deterministic economic production quantity models with time-varying demand and cost. *Applied Mathematical Modelling*, 29(10), pp. 987–1003.



185. Thonemann, U. (2006). *Introduction and Economic Order Quantity (EOQ) model, Lecture*. University of Keln, Germany.
186. Tomašević, I., Stojanović, D., & Simeunović, B. (2014). Operations management research- An update for 21st century. *XIV international symposium Symorg 2014: New business models and sustainable competitiveness (Symorg 2014)*, (pp. 1280-1287). Fakultet organizacionih nauka, Zlatibor-Beograd.
187. Tsang, E. (1993). *Foundations of Constraint Satisfaction*. Academic Press.
188. Vasiljević, D. (2001). *Računarski integrisana logistika: modeli i trendovi*. Fakultet organizacionih nauka, Beograd.
189. Vasiljević, D., & Jovanović, B. (2008). *Menadžment logistike i lanaca snabdevanja*. Fakultet organizacionih nauka, Beograd.
190. Wee, H. M. (1999). Deteriorating inventory model with quantity discount, pricing and partial backordering. *International Journal of Production Economics*, 59(1-3), pp. 511-518.
191. Wild, R. (2002). Best Practice in Inventory Management. *Elsevier Science, London*, pp. 112–148.
192. Wilde, N. P. (1993). A WYSIWYC (What You See Is What You Compute) Spreadsheet. *IEEE Symposium on Visual Languages (VL 1993)*, (pp. 72–76). Bergen, Norway.
193. Wilde, N., & Lewis, C. (1990). Spreadsheet-based interactive graphics: from prototype to tool. *SIGCHI Conference on Human Factors in Computing Systems (CHI 1990)*, (pp. 153–160). Seattle, WA, USA.
194. Wilson, R. H. (1934). A scientific routine for stock control. *Harvard Business Rev.* 13 , pp. 116–128.
195. Winston, W., & Albright, S. (2008). *Practical Management Science*. South-Western College Publishing.
196. Winston, W., Albright, S., & Broadie, M. (2001). *Practical management science*. Duxbury, Thomson Learning.
197. Wu, K. S. (2002). Deterministic inventory model for items with time varying demand, Weibull distribution, deterioration and shortages. *Yugoslav Journal of Operations Research*, 12(1), pp. 61-71.

198. Ye, Y., & Fischer, G. (2005). Reuse-Conducive Development Environments. *Automated Software Engineering*, 12(2), pp. 199–235.
199. Yoder, A. G., & Cohn, D. L. (1994). Real Spreadsheets for Real Programmers. *International Conference on Computer Languages*, (pp. 20–30). Toulouse, France.

## Biografija

Lena Đorđević je rođena 26.08.1986. godine u Beogradu, gde je završila osnovnu i srednju (Arhitektonska tehnička škola, 2005) školu. Studije na Fakultetu organizacionih nauka (FON) upisala je 2005/2006 školske godine, kao redovan budžetski student. Diplomirala je 12.10.2009. godine, na Odseku za Operacioni menadžment (OM), sa prosečnom ocenom 8,81 i ocenom 10 na diplomskom radu “Logistika u *lin (lean)* kontekstu”. Proglašena je za najboljeg studenta Odseka za OM 2009. godine. Iste (2009/2010) školske godine upisuje diplomske akademske (master) studije na FONu, studijski program *Inženjerski i operacioni menadžment (I&OM)*, studijska grupa *Računarski integrisana proizvodnja i logistika*. Sve ispite (1. Upravljački sistemi i modeli, 2. Računarski integrisani proizvodni sistemi, 3. Integrisani logistički sistemi, 4. Upravljanje lancima snabdevanja, 5. Metaheuristike + Stručna praksa, Pristupni rad i Master-rad) je položila sa ocenom 10. Završni (master) rad, na temu “Uporedni prikaz nekoliko metoda pretraživanja oblasti upravljanja objekta diskretnog upravljanja”, odbranila je 07.10.2010. godine, pod mentorstvom prof. dr Konstantina Kostića. Akademske 2011/12 upisala je doktorske studije na Fakultetu organizacionih nauka, studijski program: *Informacioni sistemi i menadžment (IS&M)*, izborno područje *Menadžment*. Položila je sve ispite sa prosečnom ocenom 10 (deset):

1. Nauka o menadžmentu
2. Spredšit inženjerstvo
3. Metodologija naučno – istraživačkog rada
4. Odlučivanje – izabrana poglavlja
5. Interakcija čoveka i računara - odabrana poglavlja
6. Marketing logistika
7. Metaheuristike
8. Upravljanje proizvodnjom i uslugama
9. Menadžment informacioni sistemi

Kandidat je 05.02.2015. odbranila pristupni rad pod nazivom „Detekcija i analiza grešaka u dinamičkim diskretnim modelima upravljačkih problema operacionog menadžmenta“, pod mentorstvom dr Danice Lečić-Cvetković, vanrednog profesora i

ostvario 30 ESPB bodova. Ukupno je ostvario 120 ESPB bodova sa prosečnom ocenom 10. Kandidatu je 08.04.2015. odobrena tema doktorske disertacije pod nazivom „Detekcija i analiza grešaka u implementaciji dinamičkih diskretnih modela upravljanja zalihama“.

Posebne teme interesovanja, na kojima kandidat aktivno radi su: spredšit inženjerstvo, spredšit menadžment, upravljanje materijalnim i nematerijalnim tokovima, upravljački i informacioni sistemi preduzeća.

Dosadašnji istraživački rad kandidata odnosi se na razvoj metoda pretraživanja oblasti upravljanja objekta diskretnog upravljanja, razvoj metoda za detekciju i analizu grešaka u spredšit modelima, razvoj i programiranje optimizacionih algoritama (*Visual Basic for Applications-VBA*), kao i na razvoj upravljačkih spredšit aplikacija i spredšit informacionih sistema.

U toku studija, izradila je softversku aplikaciju za finansijsko i računovodstveno praćenje i pretraživanje podataka za Zadružni savez Srbije, koju i danas održava.

Stručno osposobljavanje i usavršavanje, u toku master studija, obavila je u agenciji „*M&M consulting*“, u periodu od 01.03.2010-01.08.2010. Radila je na izradi softverskih aplikacija: o zaposlenima, za odnose sa kupcima i dobavljačima, za rad finansijsko-računovodstvenih službi i dr.

Kao student master studija (od oktobra 2009. do aprila 2011), bila je angažovana kao demonstrator na predmetima *Upravljački sistemi* i *Logistika* - na trećoj godini i na predmetima *Informacioni sistemi preduzeća* i *Upravljanje lancima snabdevanja* na četvrtoj godini FON-ovih osnovnih studija, studijske grupe Operacioni menadžment - OM (u okviru studijskog programa Menadžment i organizacija). Za to vreme, pored izvođenja laboratorijskih vežbi, učestvovala je i u organizaciji i održavanju predispitnih obaveza studenata OM-a.

U periodu 01.04.2011-31.05.2012. godine, na FON-u, angažovana je kao stručni saradnik u *Centru za Operacioni menadžment*. Radila je na poslovima:

- Pronalaženja, organizovanja i praćenja stručne prakse u proizvodnim i uslužnim preduzećima, za studente treće i četvrte godine studijske grupe OM, u okviru *Inkubatora za spredšit-usluge preduzećima*;
- Izrade aplikacija u okviru *Inkubatora za spredšit-usluge preduzećima*;
- Pripreme i realizacije *Naprednog kursa MS Excel-a* u okviru Centra za OM;
- Pripreme i izvođenja laboratorijskih vežbi na predmetu *Informacioni sistemi preduzeća*, na četvrtoj godini studijske grupe OM;
- Pripreme i izvođenja laboratorijskih vežbi na predmetu *Upravljački sistemi*, na trećoj godini studijske grupe OM;
- Pripreme i izvođenja laboratorijskih vežbi na predmetu *Logistika* na trećoj godini studijske grupe OM;
- Pripreme i izvođenja laboratorijskih vežbi na predmetu *Upravljanje lancima snabdevanja*, na četvrtoj godini studijske grupe OM;
- Pripreme i izvođenja laboratorijskih vežbi na predmetu *Integrirani logistički sistemi*, studijske grupe *Logistika i Računarski integrirana proizvodnja*, pri studijskom programu master-studija *Inženjerski i operacioni menadžment*;
- Pripreme i izvođenja laboratorijskih vežbi na predmetu *Marketing logistika*, studijske grupe *Menadžment*, na četvrtoj godini osnovnih studija;
- Organizacije promotivnih aktivnosti studijske grupe OM;
- Organizacije i rada na izradi Biltena za studente studijske grupe OM;
- Organizacije skupova i održavanja komunikacija u okviru Alumni asocijacije (bivših i sadašnjih) studenata OM.

Od juna 2012. godine je, na FON-u, angažovana kao saradnik u nastavi za užu naučnu oblast *Upravljanje proizvodnjom i uslugama*. Kandidat učestvuje u nastavi na predmetima: *Upravljački sistemi* (obavezni predmet III god. studijske grupe OM) i *Informacioni sistemi preduzeća* (obavezni predmet IV god. iste studijske grupe).

Od maja 2013. godine je, na FON-u, angažovana kao asistent u nastavi za užu naučnu oblast *Upravljanje proizvodnjom i uslugama*. Kandidat učestvuje u nastavi na predmetima: *Upravljački sistemi* (obavezni predmet III god. studijske grupe OM) i *Informacioni sistemi preduzeća* (obavezni predmet IV god. iste studijske grupe), *Upravljački sistemi i modeli* (izborni predmet na master studijama, studijski program

MiO) i *Informacioni sistemi MSP* (obavezni predmet na master studijama, studijski program MiO), *Spređsit menadžment* (izborni predmet na master studijama, studijski program M).

U anonimnim anketama studenata - za ocenu kvaliteta rada nastavnika i saradnika, od početka rada na FON-u do sprovođenja poslednje ankete, konstantno je ocenjivana visokim ocenama, preko 4.8 (na skali od 1 do 5).

Od 2011. godine učestvuje, u svojstvu predavača, u organizaciji i izvođenju kurseva naprednog *Excel*-a (koji se, po potrebi, održavaju na FON-u ili van FON-a).

Kao rukovodilac projekta, učestvovala je (2012.) u projektima izrade spređsit aplikacije za praćenje i upravljanje poslovanjem preduzeća SPZ Centroprojekt, Beograd i VS Vladimir, Beograd.

Učestvovala je (2013.), u svojstvu predavača, na seminaru “Životni ciklus biznis-plana”, u organizaciji FON-ovog Centra za preduzetništvo i upravljanje MSP, i nastavlja da radi u tom timu (koji ovu temu prezentuje studentima FON-a i drugih fakulteta Univerziteta u Beogradu).

Kao član projektnog tima, učestvovala je u projektu: Omerbegović-Bijelović, J., Lečić-Cvetković, D., Bečejski-Vujaklija, D., Antić, S., Đorđević, L., Rakićević, Z., Atanasov, N. (2013.), Unapređivanje kvaliteta upravljanja izabranim podsistemima preduzeća „Zelena pijaca“ iz Novog Sada – konceptualno rešenje (naučno-istraživački projekat, po ugovoru sa privrednim društvom, decembra 2013). „Zelena pijaca“ iz Novog Sada i FON iz Beograda.

Kandidat je aktivni učesnik u Timu saradnika za organizovanje i realizovanje brojnih aktivnosti promocije i rada studijske grupe OM:

- Član je Organizacionog odbora za pripremu “OM-Info-day-a” – prilike za informisanje potencijalnih (budućih) studenata studijske grupe OM;
- Član je Organizacionog odbora za pripremu sportskog dana studenata OM-a;

- Učestvovala je (2010.) u osnivanju alumni asocijacije studenata FON-ovog Odseka za OM, čiji je i danas član;
- Još iz perioda studiranja, angažovana je na poslovima podrške i organizacije događaja Skup naučnika i privrednika - SPIN ('07, '08, '09); potom, kao član Centra za OM, je i član Organizacionog odbora za pripremu SPIN (od 2009.); bila je član Organizacionog odbora i tehnički sekretar VIII Skupa privrednika i naučnika SPIN '11, u organizaciji Centra za OM; bila je član Organizacionog odbora i sekretar IX Skupa privrednika i naučnika SPIN '13, u organizaciji Centra za OM, član je Organizacionog X Skupa privrednika i naučnika SPIN'15, u organizaciji Centra za OM.
- Kao član tehničkog odbora uključena je u organizaciju XV Internationalnog Simpozijuma SymOrg 2016.
- Kandidat aktivno učestvovao u kreiranju i danas radi na održavanju sajta studijske grupe za Operacioni menadžment.

Kandidat aktivno podržava i učestvuje u vannastavnim aktivnostima studenata Operacionog menadžmenta FON-a, kao i u pripremi nastavne literature:

- U svojstvu člana žirija, učestvovala je u izboru najboljeg rešenja studije slučaja, na Lokalnom inženjerskom takmičenju *Belgrade LEC*, koji je organizovala studentska organizacija BEST, 2012. godine;
- U cilju lakšeg savladavanja gradiva i primene istog, sa kolegama sa *Katedre za upravljanje proizvodnjom i uslugama*, organizuje i održava dodatne sate konsultacija na kojima se sa studentima radi na rešavanju konkretnih problema iz prakse;
- Učestvovala u unapređivanju udžbeničke literature na obaveznom predmetu „Informacioni sistemi preduzeća“ na četvrtoj godini, studijske grupe Operacioni menadžment: Kostić K., „INFORMACIONI SISTEMI PREDUZEĆA U EXCEL-u“, 3. izmenjeno i dopunjeno izdanje, Privredni savetnik, Beograd, 2010, ISBN 978-86-732-065-8.

## Spisak objavljenih radova

Lena Đorđević je do sada objavila ukupno 42 rada, 1 udžbenik, 2 zbirke i 3 praktikuma, kao autor i koautor. Naučni rad ogleda se kroz sledeću bibliografiju radova:

### **Završni (diplomski) rad:**

Đorđević, L. (2009). Logistika u lin (*lean*) kontekstu. (mentor: prof. dr D. Vasiljević), Fakultet organizacionih nauka, Beograd.

### **Završni (master) rad:**

Đorđević, L. (2010). Uperedni prikaz nekoliko metoda pretraživanja oblasti upravljanja objekta diskretnog upravljanja. (mentor: prof. dr K. Kostić), Fakultet organizacionih nauka, Beograd.

### **Pristupni (doktorski) rad:**

Đorđević, L. (2015). Detekcija i analiza grešaka u dinamičkim diskretnim modelima upravljačkih problema operacionog menadžmenta. (mentor: dr D. Lečić-Cvetković, vanredni profesor), Fakultet organizacionih nauka, Univerzitet u Beogradu.

### **Nastavni materijali – skripte, zbirke i udžbenici**

1. Kostić, K., Antić, S., **Đorđević, L.** (2011). *Upravljački sistemi - praktikum*. Fakultet organizacionih nauka, Beograd, ISBN 978-86-7680-250-0 (obavezna udžbenička literatura na predmetu „Upravljački sistemi“, na trećoj godini studijske grupe za Operacioni menadžment).
2. Antić, S., **Đorđević, L.** (2011). *Praktikum: MS Excel - napredno korišćenje*. nastavni materijal u štampanoj formi za “Kurs naprednog Excel-a”, održan na Fakultetu organizacionih nauka, u Beogradu.
3. Kostić, K., Antić, S., **Đorđević, L.** (2014). *Informacioni sistemi preduzeća u Excel-u*. Fakultet organizacionih nauka, Beograd, ISBN 978-86-7680-302-6 (obavezna udžbenička literatura na predmetu „Informacioni sistemi preduzeća“, na četvrtoj godini studijske grupe Operacioni menadžment).
4. Kostić, K., Antić, S., **Đorđević, L.** (2014). *Informacioni sistemi preduzeća u Excel-u - zbirka primera*. Fakultet organizacionih nauka, Beograd, ISBN 978-86-7680-240-1



(obavezna udžbenička literatura na predmetu „Informacioni sistemi preduzeća“, na četvrtoj godini studijske grupe Operacioni menadžment).

5. Antić, S., **Đorđević, L.** (2015). *Upravljački sistemi - praktikum*. Fakultet organizacionih nauka, Beograd, ISBN 978-86-7680-312-5 (obavezna udžbenička literatura na predmetu „Upravljački sistemi“, na trećoj godini studijske grupe za Operacioni menadžment).
6. Kostić, K., Antić, S., **Đorđević, L.** (2015). *Informacioni sistemi preduzeća u Excel-u - zbirka primera*. Fakultet organizacionih nauka, Beograd, ISBN 978-86-7680-240-1 (obavezna udžbenička literatura na predmetu „Informacioni sistemi preduzeća“, na četvrtoj godini studijske grupe Operacioni menadžment).

#### ***Radovi objavljeni u naučnim časopisima međunarodnog značaja (M23)***

1. Šošević, U., **Đorđević, L.**, Milovanović, M. (2013). Impact of Screen Aspect Ratio on Reading Electronic Material. *International Journal of Engineering Education*, vol. 29, No. 3, pp. 602-609, ISSN 0949-149X (IF2013: 0.360)
2. Antić, S., **Đorđević, L.**, Kostić, K., Lisec, A. (2015). Dynamic discrete simulation model of an inventory control with or without allowed shortages. *Scientific Bulletin Series A, Applied Mathematics and Physics*, ISSN 1223-027, vol. 77, No. 1, pp. 163-176 (IF2014: 0.405)
3. **Đorđević, L.**, Antić, Čangalović, M., Lisec, A. (2016). A metaheuristic approach to solving a multiproduct EOQ-based inventory problem with storage space constraints, *Optimization Letters*, DOI 10.1007/s11590-016-1009-5 (IF2104: 0.934) (First online: 18 February 2016, <http://link.springer.com/article/10.1007/s11590-016-1009-5>)

#### ***Radovi objavljeni u časopisima nacionalnog značaja (M50)***

##### ***Radovi objavljeni u časopisu nacionalnog značaja (M52)***

1. **Đorđević, L.**, Antić, S. (2012). Primena softvera *Logware* u edukaciji menadžera logistike. *Info M 41/2012, časopis za informacione tehnologije i multimedijalne sisteme*, Broj: 41, Redni broj objavljenog rada – 5, pp. 26-32, ISSN 1451-4397

2. **Đorđević, L.** (2012). Primena metoda klasterovanja za definisanje programa spredšit obuke na osnovu atributa korisnika. *Računovodstvo 7-8/2012*, Redni broj objavljenog rada – 2, pp. 18-28, ISSN 0023-2394, ISSN 1450-6114
3. Antić, S., Lečić-Cvetković, D., Čangalović, M., **Đorđević, L.** (2014). Spredšit model metaheurističkog pristupa za rešavanje problema ekonomične količine naručivanja zaliha za više proizvoda sa ograničenim prostorom skladištenja. *Info M 52/2014, časopis za informacione tehnologije i multimedijalne sisteme*, Broj: 52, Redni broj objavljenog rada – 3, pp. 11-19, ISSN 1451-4397
4. **Đorđević, L.**, Lečić-Cvetković, D. (2015). Pregled i komparacija pristupa za detekciju i analizu grešaka u spredšitovima. *Info M 56/2015, časopis za informacione tehnologije i multimedijalne sisteme*, Broj: 56, Redni broj objavljenog rada – 2, pp. 11-16, ISSN 1451-4397

***Radovi objavljeni u zbornicima međunarodnih naučnih skupova (M30)***

*Saopštenja sa međunarodnih skupova štampana u celini (M33)*

1. Antić, S., **Đorđević, L.** (2011). Logistički spredšit-model za nagrađivanje vozača u distribuciji maloprodajnih lanaca. *III Međunarodni simpozijum „Novi horizonti saobraćaja i komunikacija“*, Zbornik radova – CD, ISBN: 978-99955-36-28-2, pp. 91-101, 24-25. XI 2011, Saobraćajni fakultet Doboj, Doboj, Republika Srpska.
2. **Đorđević, L.**, Antić, S., Kostić, K. (2012). Dynamic discrete spreadsheet EOQ model for educational purposes, *31st International Conference On Organizational Science Development, “Quality, Innovation, Future”*, CD with Papers, ISBN: 978-961-232-254-0, COBISS.SR-ID260819200, pp. 21-23. III 2012, Portorož, Slovenija.
3. Antić, S., **Đorđević, L.** (2012). Dynamic discrete inventory spreadsheet model in water-coolers distribution. *VI International conference on agricultural logistics, Symposium proceedings*, ISBN 978-961-6562-55-3, pp. 1-10, 8 - 9 November 2012, Novo mesto, Slovenia.
4. Marinović, M., Makajić-Nikolić, D., Stanojević, M., **Đorđević, L.** (2012). Optimization of electricity trading using linear programming. *3rd Student Conference on Operational Research*, OpenAccess Series in Informatics-OASICs, ISBN 978-3-939897-39-2, ISSN 2190-6807, pp. 94-102, Nottingham, UK, publisher: Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik Dagstuhl,

- Germany, URL <http://drops.dagstuhl.de/opus/volltexte/2012/3550>, urn:nbn:de:0030-drops-35505, doi:<http://dx.doi.org/10.4230/OASICS.SCOR.2012.94>, 20-22. IV 2012.
5. Antić, S., Kostić, K., **Dorđević, L.** (2012). Spreadsheet model of inventory control based on modern control theory. *XIII International Symposium SymOrg 2012, "Innovative management and business performance"*, Symposium proceedings, CD with Papers, Book of Abstracts, ISBN 978-86-7680-255-5, pp. 1365-1373, Faculty of Organizational Sciences, University of Belgrade, COBISS.SR-ID 191307276, 5-9. VI 2012. Zlatibor, Serbia.
  6. Antić, S., **Dorđević, L.** (2012). Case study: Dynamic discrete EOQ model for inventories in water coolers distribution, *Proceedings of the 17th International Symposium on Logistics and Supply chain management – ISL 2012*, ISBN: 978 085358 285 4, pp. 372-381, 8–11th July 2012, Cape Town, South Africa.
  7. **Dorđević, L.**, Vasiljević, D. (2013). Spreadsheets in education of logistics managers at Faculty of organizational sciences: an example of inventory dynamics simulation. *Proceedings of the 7th International Technology Education and Development Conference INTED 2013*, ISBN: 978-84-616-2661-8, pp. 640-649, 4-6. III 2013. Valencia, Spain.
  8. Antić, S., **Dorđević, L.** (2013). Spreadsheet usability in theory and application of operations management. *32nd International Conference on Organizational Science Development "SMART ORGANIZATION", High Potentials. Lean Organization. Internet of Things*, CD with Papers, ISBN 978-961-232-264-9, pp. 1267-1276, 20 – 22. III 2013. Portorož, Slovenia.
  9. Atanasov, N., Lečić-Cvetković, D., Rakićević, Z., Omerbegović-Bijelović, J., **Dorđević, L.** (2013). An approach to lean inventory management by balanced stock cover. *2nd International Scientific Conference on Lean Technologies (LeanTech '13)*, LeanTech'13 Proceedings, ISBN 978-86-7680-283-8, pp. 159-166, Chair of industrial and management engineering, Faculty of organizational sciences, University of Belgrade. Belgrade, Serbia.
  10. Antić, S., **Dorđević, L.**, Panić, S. (2013). Case Study: Implementation of Lean methodology for improvement of storage process in the company PINOLES. *2nd International Scientific Conference on Lean Technologies, LeanTech'13 Proceedings*, ISBN 978-86-7680-283-8, pp. 167-174, Chair of industrial and

- management engineering, Faculty of organizational sciences, University of Belgrade. Belgrade, Serbia.
11. Omerbegović-Bijelović, J., Antić, S., **Dorđević, L.**, Rakićević, Z., Atanasov, N. (2013). Planning tools for SMEs management quality improvement in supply chains: Case study in water distribution. *Proceedings of the 7th International Conference on Logistics in Agriculture*, ISBN 978-961-6562-85-0, pp. 125-140, University of Maribor, Faculty of Logistics, Center of university studies and Research, Novo mesto, Slovenia.
  12. Antić, S., **Dorđević, L.**, Čangalović, M., Kostić, K. (2013). A Metaheuristic Approach to Solving a Multiproduct EOQ-Based Inventory Problem with Storage Space Constraints. *BALCOR 2013*, Proceedings of the XI Balkan Conference on Operational Research, ISBN 978-86-7680-285-2, pp. 228-234, University of Belgrade, Faculty of Organizational Sciences. Belgrade & Zlatibor, Serbia.
  13. **Dorđević, L.**, Antić, S., Lečić-Cvetković, D. (2014). Spreadsheet application in operations management education. *XIV International Symposium SymOrg 2014 „New Business Models and Sustainable Competitiveness“*, Symposium proceedings, CD with Papers, ISBN 978-86-7680-295-1, pp. 1231-1238, Faculty of Organizational Sciences, University of Belgrade, COBISS.SR-ID 191307276, Zlatibor, Serbia.
  14. Liseć, A., Levičar S., Antić, S., **Dorđević, L.** (2014). Shelf ready packaging in retailing. *19th International Symposium on Logistics (ISL2014)*, Symposium proceedings, pp. 154-159, Ho Chi Minh, Vietnam.
  15. Antić, S., **Dorđević, L.** (2014). Case study: Using a special heuristics approach for solving a multiproduct EOQ-based inventory problem with storage space constraints in the company LA FANTANA. *Proceedings of the 8th International Conference on Logistics in Agriculture*, ISBN 978-961-6962-01-8, pp. 32-40, University of Maribor, Faculty of Logistics, Center of university studies and Research, Novo mesto, Slovenia.
  16. Antić, S., **Dorđević, L.**, Lečić-Cvetković, D., Liseć, A. (2015). Using a VNS methodology approach to solving a multiproduct EOQ-based inventory problem with storage space constraints in the company LaFantana. *20th International*

*Symposium on Logistics (ISL2015)*, Symposium proceedings, pp. 517-525 July 5-8, Bologna, Italy.

***Radovi objavljeni u zbornicima sa skupova nacionalnog značaja (M60)***

*Saopštenja sa skupova nacionalnog značaja štampana u celini (M63)*

1. **Đorđević, L.**, Tešić, D., Vasiljević, D. (2009). Logistika u ekološkom kontekstu. VII Skup privrednika i naučnika – SPIN '09, „Operacioni menadžment i globalna kriza“, Zbornik radova, ISBN 978-86-7680-202-9, pp. 337-344, Centar za operacioni menadžment, FON i Privredna komora Srbije, 5-6. XI 2009, Beograd.
2. **Đorđević, L.**, Jovanović, B., Vasiljević, D. (2010). Logističke igre u nastavnim programima strukovnih studija. Drugi naučno-stručni skup sa međunarodnim učešćem „Tendencije razvoja u tekstilnoj industriji – dizajn, tehnologija, menadžment“, DTM 2010, Zbornik radova, ISBN 978-86-87017-05-4, pp. 171-175, 04-05. VI 2010, Visoka tekstilna strukovna škola za dizajn, tehnologiju i menadžment, Beograd.
3. Antić, S., **Đorđević, L.** (2011). Projekat razvoja aplikacije za praćenje materijala u magacinu”, XV Internacionalni simpozijum iz projektnog menadžmenta – YUPMA 2011, zbornik radova, ISBN: 978-86-86385-08-6, pp. 432-436, Udruženje za upravljanje projektima Srbije, Beograd, 10-12. VI 2011. Zlatibor, Srbija.
4. Antić, S., Stanišić M., **Đorđević L.** (2011). Softverska aplikacija u spredšitovima za praćenje materijala i rezervnih delova u magacinu. Konferencija o računarskim naukama i informacionim tehnologijama - YU INFO 2011, Zbornik radova – CD, ISBN 978-86-85525-08-7, pp. 305-309, 06-09. III 2011, Kopaonik, Društvo za informacione sisteme i računarske mreže, Beograd.
5. **Đorđević, L.**, Čangalović, M., Kostić K. (2011). Metaheuristike za problem optimalnog upravljanja diskretnim sistemima. XXXVIII Simpozijum o operacionim istraživanjima – SYM-OP-IS 2011, Zbornik radova, ISBN 978-86-403-1168-7, pp. 345-348, Ekonomski fakultet u Beogradu, 04-07. X 2011, Zlatibor.
6. Antić, S., **Đorđević, L.** (2011). EOQ model za više proizvoda sa ograničenim skladišnim prostorom kao objekat diskretnog upravljanja u MS Excel-u. XXXVIII Simpozijum o operacionim istraživanjima – SYM-OP-IS 2011, Zbornik radova,

- ISBN 978-86-403-1168-7, pp. 727-731, Ekonomski fakultet u Beogradu, 04-07. X 2011, Zlatibor.
7. **Đorđević, L.**, Danilović, M., Vasiljević, D. (2011). Primena softvera *Logware* u edukaciji menadžera logistike. *VIII skup privrednika i naučnika – SPIN '11*, „Operacioni menadžment u funkciji održivog ekonomskog rasta i razvoja Srbije 2011-2020“, Zbornik radova, ISBN 978-86-7680-244-9, pp. 358-365, Centar za operacioni menadžment, FON i Privredna komora Srbije, 1-2. XI 2011, Beograd.
  8. Antić, S., **Đorđević, L.** (2011). Ključni logistički indikatori performansi u distribuciji maloprodajnih lanaca. *VIII Skup privrednika i naučnika – SPIN '11*, „Operacioni menadžment u funkciji održivog ekonomskog rasta i razvoja Srbije 2011-2020“, Zbornik radova, ISBN 978-86-7680-244-9, pp. 335-341, Centar za operacioni menadžment, FON i Privredna komora Srbije, 1-2. XI 2011, Beograd.
  9. Antić, S., Laskač, M., **Đorđević, L.** (2012). Softverska aplikacija za praćenje podataka o komitentima u turističkoj agenciji *Smart travel*. *Konferencija o računarskim naukama i informacionim tehnologijama - YU INFO 2012*, Zbornik radova – CD, ISBN 978-86-85525-09-4, pp. 317-321, 29. II - 03. III 2012, Kopaonik, Društvo za informacione sisteme i računarske mreže, Beograd.
  10. **Đorđević, L.**, Suknović, M., Jovanović, M., Antić, S. (2012). Primena metoda klasterovanja za definisanje programa obuka na osnovu atributa korisnika. *XXXIX Simpozijum o operacionim istraživanjima – SYM-OP-IS 2012*, Zbornik radova, pp. 11-15, Visoka građevinsko-geodetska škola, Beograd, 25-28. X 2012, Tara.
  11. **Đorđević, L.**, Antić, S., Đurić, M., (2013). Karakteristike i najbolja praksa spredšit korisnika sa različitim nivoom iskustva. *Konferencija o računarskim naukama i informacionim tehnologijama - YU INFO 2013*, Zbornik radova – CD, ISBN 978-86-85525-11-7, pp. 76-82, 03-06. III 2013, Kopaonik, Društvo za informacione sisteme i računarske mreže, Beograd.
  12. **Đorđević, L.**, Antić, S., Kostić, K. (2013). Upotrebljivost spredšitova u teoriji i praksi. *IX skup privrednika i naučnika SPIN'13*, Zbornik radova IX skupa privrednika i naučnika, ISBN 978-86-7680-288-3, pp. 307-316, Centar za operacioni menadžment – FON, Univerzitet u Beogradu, Srbija.
  13. **Đorđević, L.**, Antić, S. (2014). Ocena kvaliteta analitičkog spredšit modela. *XX naučno-stručna i biznis konferencija YU INFO 2014*, Zbornik radova XX naučno-

- stručne i biznis konferencije, pp. 206-211, Kopaonik, Društvo za informacione sisteme i računarske mreže, Beograd, Srbija.
14. Antić, S., **Đorđević, L.**, Čangalović, M., Lečić-Cvetković, D. (2014). Metaheuristički pristup rešavanju problema ekonomske količine naručivanja zaliha za više proizvoda sa ograničenim prostorom skladištenja. *XLI Simpozijum o operacionim istraživanjima – SYM-OP-IS 2014*, Zbornik radova, pp. 245-250, Saobraćajni fakultet u Beogradu, Srbija.
  15. **Đorđević, L.**, Ljubičić, M., Marjanović, Z., Antić, S. (2015). Primena tehnologije semantičkog web-a za detekciju grešaka u spredšit modelima diskretnih sistema, *XXI naučno-stručna i biznis konferencija YU INFO 2015*, Zbornik radova XXI naučno-stručne i biznis konferencije, ISBN 978-86-85525-15-5, pp. 121-126, Društvo za informacione sisteme i računarske mreže, Beograd.
  16. **Đorđević, L.**, Lečić-Cvetković, D. (2015). Uvodni prikaz pristupa za detekciju i analizu grešaka u spredšitovima. *X skup privrednika i naučnika SPIN'15*, Zbornik radova X skupa privrednika i naučnika, ISBN 978-86-7680-320-0, pp. 280-287, Centar za operacioni menadžment – FON, Univerzitet u Beogradu, Srbija.
  17. Bogdanović, J., Antić, S., **Đorđević, L.** (2015). Spredšit model kao efikasan alat za obračun zarada. *X skup privrednika i naučnika SPIN'15*, Zbornik radova X skupa privrednika i naučnika, ISBN 978-86-7680-320-0, pp. 272-279, Centar za operacioni menadžment – FON, Univerzitet u Beogradu, Srbija.
  18. Miletić, D., Antić, S., **Đorđević, L.** (2015). Spredšit model za upravljanje matičnim podacima artikala u maloprodaji. *X skup privrednika i naučnika SPIN'15*, Zbornik radova X skupa privrednika i naučnika, ISBN 978-86-7680-320-0, pp. 288-295, Centar za operacioni menadžment – FON, Univerzitet u Beogradu, Srbija.
  19. Popović, J., **Đorđević, L.**, Antić, S. (2015). Simulacioni spredšit model za merenje performansi prodaje. *X skup privrednika i naučnika SPIN'15*, Zbornik radova X skupa privrednika i naučnika, ISBN 978-86-7680-320-0, pp. 296-303, Centar za operacioni menadžment – FON, Univerzitet u Beogradu, Srbija.

## Prilog 1.

### Izjava o autorstvu

Potpisana Lena S. Đorđević

broj indeksa 5011/2011

### Izjavljujem

da je doktorska disertacija pod naslovom

Detekcija i analiza grešaka u implementaciji dinamičkih diskretnih modela upravljanja  
zalihama

- rezultat sopstvenog istraživačkog rada,
- da predložena disertacija u celini ni u delovima nije bila predložena za dobijanje bilo koje diplome prema studijskim programima drugih visokoškolskih ustanova,
- da su rezultati korektno navedeni i
- da nisam kršila autorska prava i koristio intelektualnu svojinu drugih lica.

**Potpis doktoranda**

U Beogradu, \_\_\_\_\_

\_\_\_\_\_



**Prilog 2.**

## **Izjava o istovetnosti štampane i elektronske verzije doktorskog rada**

Ime i prezime autora Lena S. Đorđević

Broj indeksa 5011/2011

Studijski program Informacioni sistemi i menadžment, izborno područje Menadžment

Naslov rada Detekcija i analiza grešaka u implementaciji dinamičkih diskretnih  
modela upravljanja zalihama

Mentor dr Danica Lečić-Cveković, vanredni profesor, Fakultet organizacionih nauka,  
Univerzitet u Beogradu

Potpisana Lena S. Đorđević

Izjavljujem da je štampana verzija mog doktorskog rada istovetna elektronskoj verziji koju sam predao/la za objavljivanje na portalu **Digitalnog repozitorijuma Univerziteta u Beogradu**.

Dozvoljavam da se objave moji lični podaci vezani za dobijanje akademskog zvanja doktora nauka, kao što su ime i prezime, godina i mesto rođenja i datum odbrane rada.

Ovi lični podaci mogu se objaviti na mrežnim stranicama digitalne biblioteke, u elektronskom katalogu i u publikacijama Univerziteta u Beogradu.

**Potpis doktoranda**

U Beogradu, \_\_\_\_\_

\_\_\_\_\_

### Prilog 3.

## Izjava o korišćenju

Ovlašćujem Univerzitetsku biblioteku „Svetozar Marković“ da u Digitalni repozitorijum Univerziteta u Beogradu unese moju doktorsku disertaciju pod naslovom:

Detekcija i analiza grešaka u implementaciji dinamičkih diskretnih modela upravljanja zalihama

koja je moje autorsko delo.

Disertaciju sa svim priložima predala sam u elektronskom formatu pogodnom za trajno arhiviranje.

Moju doktorsku disertaciju pohranjenu u Digitalni repozitorijum Univerziteta u Beogradu mogu da koriste svi koji poštuju odredbe sadržane u odabranom tipu licence Kreativne zajednice (Creative Commons) za koju sam se odlučila.

1. Autorstvo
2. Autorstvo - nekomercijalno
3. Autorstvo – nekomercijalno – bez prerade
4. Autorstvo – nekomercijalno – deliti pod istim uslovima
5. Autorstvo – bez prerade
6. Autorstvo – deliti pod istim uslovima

**Potpis doktoranda**

U Beogradu, \_\_\_\_\_

\_\_\_\_\_

1. Autorstvo - Dozvoljavate umnožavanje, distribuciju i javno saopštavanje dela, i prerade, ako se navede ime autora na način određen od strane autora ili davaoca licence, čak i u komercijalne svrhe. Ovo je najslobodnija od svih licenci.
2. Autorstvo – nekomercijalno. Dozvoljavate umnožavanje, distribuciju i javno saopštavanje dela, i prerade, ako se navede ime autora na način određen od strane autora ili davaoca licence. Ova licenca ne dozvoljava komercijalnu upotrebu dela.
3. Autorstvo - nekomercijalno – bez prerade. Dozvoljavate umnožavanje, distribuciju i javno saopštavanje dela, bez promena, preoblikovanja ili upotrebe dela u svom delu, ako se navede ime autora na način određen od strane autora ili davaoca licence. Ova licenca ne dozvoljava komercijalnu upotrebu dela. U odnosu na sve ostale licence, ovom licencom se ograničava najveći obim prava korišćenja dela.
4. Autorstvo - nekomercijalno – deliti pod istim uslovima. Dozvoljavate umnožavanje, distribuciju i javno saopštavanje dela, i prerade, ako se navede ime autora na način određen od strane autora ili davaoca licence i ako se prerada distribuira pod istom ili sličnom licencom. Ova licenca ne dozvoljava komercijalnu upotrebu dela i prerada.
5. Autorstvo – bez prerade. Dozvoljavate umnožavanje, distribuciju i javno saopštavanje dela, bez promena, preoblikovanja ili upotrebe dela u svom delu, ako se navede ime autora na način određen od strane autora ili davaoca licence. Ova licenca dozvoljava komercijalnu upotrebu dela.
6. Autorstvo - deliti pod istim uslovima. Dozvoljavate umnožavanje, distribuciju i javno saopštavanje dela, i prerade, ako se navede ime autora na način određen od strane autora ili davaoca licence i ako se prerada distribuira pod istom ili sličnom licencom. Ova licenca dozvoljava komercijalnu upotrebu dela i prerada. Slična je softverskim licencama, odnosno licencama otvorenog koda.