Prepoznavanje i rješavanje anomalija u modelima procesa uporabom uzoraka

Vidačić, Tomislav

Doctoral thesis / Disertacija

2015

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Organization and Informatics Varaždin / Sveučilište u Zagrebu, Fakultet organizacije i informatike Varaždin**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:211:408989

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-29**

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN

Repository / Repozitorij:

Faculty of Organization and Informatics - Digital Repository

DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

Sveučilište u Zagrebu

Fakultet organizacije i informatike

Tomislav Vidačić

PREPOZNAVANJE I RJEŠAVANJE ANOMALIJA U MODELIMA PROCESA UPORABOM UZORAKA

DOKTORSKI RAD

Varaždin, 2015.

PODACI O DOKTORSKOM RADU

I. AUTOR

Ime i prezime	Tomislav Vidačić
Datum i mjesto rođenja	29.12.1973.
Naziv fakulteta i datum diplomiranja na VII/I stupnju	Fakultet organizacije i informatike Varaždin
Naziv fakulteta i datum diplomiranja na VII/II stupnju	Fakultet organizacije i informatike Varaždin
Sadašnje zaposlenje	Hrvatska poljoprivredna agencija

II. DOKTORSKI RAD

Naslov	Prepoznavanje i rješavanje anomalija u modelima procesa uporabom uzoraka
Broj stranica, slika, tabela, priloga, bibliografskih podataka	233 stranica, 128 slika, 40 tablica, 71 bibliografskih podataka
Znanstveno područje i polje iz kojeg je postignut doktorat znanosti	Društvene znanosti / Informacijske i komunikacijske znanosti / Informacijski sustavi i informatologija
Mentori ili voditelji rada	prof. dr.sc. Vjeran Strahonja
Fakultet na kojem je obranjen doktorski rad	Sveučilište u Zagrebu, Fakultet organizacije i informatike u Varaždinu, Hrvatska
Oznaka i redni broj rada	

III. OCJENA I OBRANA

Datum sjednice Fakultetskog vijeća na kojoj je prihvaćena tema	15.07.2014.
Datum predaje rada	01.04.2015.
Datum sjednice Fakultetskog vijeća na kojoj je prihvaćena pozitivna ocjena rada	26.05.2015.
Sastav povjerenstva koje je rad ocijenilo	Prof. dr.sc. Neven Vrček Prof.dr.sc. Vesna Bosilj Vukšić Prof.dr.sc. Vjeran Strahonja Izv.prof.dr.sc. Danijel Radošević Doc.dr.sc. Katarina Tomičić Pupek
Datum obrane doktorskog rada	17.06.2015.
Sastav povjerenstva pred kojim je rad obranjen	Prof. dr.sc. Neven Vrček Prof.dr.sc. Vesna Bosilj Vukšić Prof.dr.sc. Vjeran Strahonja Izv.prof.dr.sc. Danijel Radošević Doc.dr.sc. Katarina Tomičić Pupek
Datum promocije	

Sveučilište u Zagrebu

Fakultet organizacije i informatike

TOMISLAV VIDAČIĆ

PREPOZNAVANJE I RJEŠAVANJE ANOMALIJA U MODELIMA PROCESA UPORABOM UZORAKA

DOKTORSKI RAD

Mentor(i): prof.dr.sc. VJERAN STRAHONJA

Varaždin, 2015.

University of Zagreb

Faculty of Organization and Informatics

Tomislav Vidačić

IDENTIFICATION AND RESOLUTION OF ANOMALIES IN PROCESS MODELS USING PATTERNS

DOCTORAL THESIS

Varaždin, 2015.

Predgovor

Tijekom prošlog desetljeća došlo je do intenzivnog razvoja modeliranja poslovnih procesa, kako metoda modeliranja, tako i metoda validacije i verifikacije. Uz metode modeliranja, anomalije modela i metode njihovog otkrivanja i prevencije postale su novo područje istraživanja. S obzirom na moguće posljedice anomalija potrebno je osigurati metode njihovog prepoznavanja i rješavanja. U ovome radu daje se pregled mogućih anomalija i načine njihovog mogućeg prepoznavanja i rješavanja pomoću uzoraka. Motiviranost za razradu ove teme seže do 2009. godine. Na obrani znanstvenog magistarskog rada pod naslovom *Referentni model procesa visokogradnje uz intenzivnu ICT potporu* [64], profesor Strahonja kao član komisije, postavio mi je pitanje da li je moguće BPMN notaciju pretvoriti u neku drugu notaciju. Naravno, odgovorio sam potvrdno, ne razmišljajući koje sve probleme bi takva transformacija notacije mogla prouzročiti. Godinu dana kasnije zamolio sam profesora Strahonju da mi bude mentor kod odabira teme i izrade disertacije na doktorskom studiju. Kao tema, odmah se je nametnula ideja da bi trebalo razraditi upravo pitanje transformacije modela iz jedne notacije u drugu. Odmah se je tu pojavilo niz pitanja, od problema smjera pretvorbe, preko određivanja zajedničkih točaka dvaju modela, preko pronalaženja grešaka i načine rješavanja grešaka u modelima. S vremenom pojedina pitanja su se isprofilirala, pa smo za temu odabrali razradu mehanizama prepoznavanja i rješavanja anomalija u modelima procesa uporabom uzoraka. Ovom radu prethodila su istraživanja provedena tijekom 2013. i 2014. godine u kojima je razrađena temeljna struktura taksonomije anomalija sa podgrupama, njihovim osnovnim karakteristikama i primjerima, te su naznačene temeljne smjernice rješavanja pojedinih tipova anomalija. Istraživanje je rezultiralo radom autora Strahonja i Vidačić pod naslovom *Taxonomy of anomalies in business process models* [65] predstavljenog na konferenciji ISD2013. Tijekom izrade rada pojavljivali su se razni istraživački i znanstveni izazovi, pa koristim ovu priliku zahvaliti svima koji su doprinijeli kvaliteti rada posebno mentoru prof.dr.sc.Vjeranu Strahonji za strpljenje, poticaj, diskusije i dobronamjerne savjete. Također, zahvaljujem prof.dr.sc. Vesni Bosilj Vukšić, prof.dr.sc. Nevenu Vrčeku, doc.dr.sc. Katarini Tomičić Pupek i izv.prof.dr.sc. Danijelu Radoševiću na konstruktivnim primjedbama. Moram zahvaliti i svojoj obitelji, supruzi Jeleni, majci i ocu na nesebičnoj podršci i razumijevanju. Iskreno, hvala svima.

U Varaždinu, 2015. godine Autor

Sažetak

Modeliranje procesa i metodološka istraživanja modeliranja funkcionalnosti (funkcija, procesa, aktivnosti, operacija, itd.) provode se od početaka razvoja i primjene računala. Pritom su predmet interesa metode i tehnike modeliranja, ali i anomalije modela te njihovo otkrivanje i otklanjanje. Suvremene servisno orijentirane arhitekture, razvoj vođen transformacijom modela i općenito težnja prema automatiziranom razvoju, daju novu važnost istraživanjima anomalija modela procesa i drugih modela funkcionalnosti.

Rad se bavi istraživanjem anomalija u modelima procesa, njihovim osobinama, zajedničkim obilježjima, podjelom, metodama i metodikama prepoznavanja, te rješavanja uz primjenu uzoraka. Temeljni pristup otkrivanja i rješavanja anomalija u radu je formiranje taksonomije anomalija, te opis njihovih svojstava i uzoraka. Namjera je da se za svaku klasu anomalija odrede metode otkrivanja i rješavanja, s ciljem automatizacije prepoznavanja i rješavanja anomalija.

Ključne riječi: modeli procesa, anomalije, BPMN, uzorci, metamodel, OCL, pseudokod

Prošireni sažetak

Anomalije u modelima poslovnih procesa predstavljaju situacije u kojima dolazi do odstupanja od pravila modeliranja i prikaza modela. Anomalije su odstupanja od očekivane ili uobičajene strukture, ponašanja, funkcionalnosti, rezultata i sl. S obzirom da anomalije imaju određena zajednička svojstva i karakteristike po kojima ih se može grupirati i klasificirati, može se reći, da su anomalije zapravo obrasci koji uzrokuju odstupanja modela od normalnog ponašanja, a koje treba ukloniti s ciljem ispravke modela.

Model u kojem postoji anomalija je neispravan i zahtjeva dodatnu intervenciju kojom se taj model ispravlja. Anomalije u modelima poslovnih procesa mogu se pojaviti tijekom: modeliranja, modificiranja modela i pretvorbe modela. Problemi s anomalijama pogotovo dolaze do izražaja u slučaju pretvorbe modela iz jedne notacije u drugu. U tom slučaju različite namjene pojedinih notacija obično diktiraju i naglasak detalja na pojedinim elementima modela. Iz tog razloga može doći do gubitka informacija tijekom te transformacije.

Ovaj rad bavi se anomalijama u modelima poslovnih procesa, njihovim svojstvima, karakteristikama i obilježjima, razradom njihove taksonomije, načinima njihovog prepoznavanja i načinima njihovog rješavanja.

U radu je izložen pregled anomalija, koje su svrstane prema određenim karakteristikama. Za svaku anomaliju naveden je opis i primjer, te je predložen njihov način prepoznavanja i rješavanja, u obliku OCL pravila ili u pseudokodu. Na kraju, razrađen je prijedlog automatizacije prepoznavanja i rješavanja anomalija koji se temelji na prepoznavanju uzoraka, te je predloženi model isproban na nekoliko primjera.

Ključne riječi: modeli procesa, anomalije, BPMN, uzorci, metamodel, OCL, pseudokod

Extended abstract

Anomalies in business process models represent situations in which there is a deviation from the rules of modeling and presentation models. Anomalies are departures from the expected or normal structure, behavior, functionality, performance and etc. Since anomalies have certain common properties and characteristics by which they can be grouped and classified, we can say that the anomalies are actually patterns that cause deviations from normal behavior model and need to be cleared.

A model in which exists an anomaly is defective and requires additional intervention which corrects this model. Anomalies in the models of business processes can occur during: modeling, modify models and conversion models. Problems with anomalies especially come to the fore in the case of conversion models from one notation to another. In this case, different uses of specific notation usually dictate and emphasis details on individual elements of the model. For this reason, may cause loss of information during this transformation.

This paper deals with anomalies in the models of business processes, their properties, characteristics and features, by developing their taxonomy, the ways of their recognition and ways of their solution.

This paper presents an overview of anomalies, which are classified according to certain characteristics. For each anomaly is given a description and example.

He also suggested a way to identify and solve, in the form of OCL rules or in pseudo-code. Finally worked out a proposal automation to identify and solve anomalies based on pattern recognition, and our model is tested on several examples.

Keywords: process model, anomalies, BPMN, patterns, metamodel, OCL, pseudocode.

Sadržaj

1. UVOD ... 1
1.1. Područje, predmeti i objekti istraživanja ... 2
1.2. Temeljni pojmovi i definicije ... 3
1.3. Pregled dosadašnjih istraživanja i istraživačkih problema ... 8
1.4. Hipoteze i očekivani znanstveni doprinos ... 14
1.5. Metodologija i plan istraživanja ... 16
2. METODE ISTRAŽIVANJA ANOMALIJA U MODELIMA POSLOVNIH PROCESA ... 19
2.1. Metodologija oblikovanja taksonomije ... 20
2.2. Metode prepoznavanja i rješavanja anomalija u modelima poslovnih procesa ... 23
2.2.1. Metamodel ... 24
Hijerarhija metamodela ... 25
2.2.2. MOF 2.0 standard ... 26
Arhitektura MOF-a ... 27
UML biblioteke infrastrukture ... 27
2.2.3. OCL ... 31
OCL konstruktori ... 31
OCL izrazi ... 32
OCL standardne biblioteke ... 33
Nedostaci OCL jezika ... 36
2.2.4. Pseudokod ... 36
Pravila pisanja pseudokoda ... 37
Prednosti i nedostaci pseudokoda ... 39
Imenski prostor ... 40
3. TAKSONOMIJA ANOMALIJA U MODELIMA POSLOVNIH PROCESA ... 41
3.1. Sintaktičke anomalije ... 45
3.1.1. Nepravilna uporaba objekata toka ... 45
Više početnih događaja u modelu ... 46
Nepovezanost početnog događaja s ostalim elementima modela ... 47
Početni događaj ima više izlaznih sekvenci ... 48
Nepostojanje završnog događaja ... 49
Nepovezanost završnog događaja s ostalim elementima modela ... 50
Završni događaj ima više od jedne ulazne sekvence ... 51
Nepostojanje početnog događaja ... 52
3.1.2. Nepravilna uporaba veza ... 53
Ulazna sekvenca s početnim događajem ... 53
Izlazna sekvenca s završnim događajem ... 55
3.1.3. Nepravilna uporaba toka poruka ... 57
Početni događaj poruke vezan je tokom poruke s elementom početnog događaja ... 58
Početni događaj poruke vezan je tokom poruke s elementom bazena ... 59
Početni događaj poruke vezan je tokom poruke s elementom potprocesa ... 59
Početni događaj poruke vezan je tokom poruke s elementom aktivnosti ... 60
Početni događaj vezan je tokom poruke s elementom stop događaja poruke ... 60
Početni događaj vezan je tokom poruke s elementom završnog događaja poruke ... 61
Početni događaj poruke vezan je tokom poruke s završnim događajem ... 61
Bazen je vezan tokom poruke s završnim događajem ... 62

Potproces je vezan tokom poruke s završnim događajem ... 62
Aktivnost je vezana tokom poruke s završnim događajem ... 63
Stop događaj poruke je vezan tokom poruke s završnim događajem ... 64
Element završnog događaja poruke vezan je tokom poruke s elementom završnog događaja ... 64
3.1.4. Nepravilna uporaba spremnika ... 66
Nepravilna uporaba bazena ... 66
Bazen bez elemenata ... 67
Nepravilna uporaba staza ... 68
Staza bez elemenata ... 69
3.2. Strukturne anomalije u BPMN 2.0 notaciji ... 71
3.2.1. Kontrola toka ... 71
Nepotrebna kontrola toka ... 71
Redundancija modela ... 74
Nedefinirani uvjeti vrata ... 77
Kontradiktorni uvjeti vrata ... 78
Nedostatak sinkronizacije ... 80
Deterministički nedostatak sinkronizacije ... 80
Nedeterministički nedostatak sinkronizacije ... 82
Viseća stanja ... 83
Netolerancija na promjene poretka ... 85
Nekonzistentnost kod inserta paralelnih grana ... 86
Nekonzistentnost promjene iz prošlosti ... 87
Nedostatak tolerancije u petlji ... 89
Zapinjanje toka ... 91
Zastoj toka ... 92
3.2.2. Anomalije pravila ... 94
Redundantna pravila ... 94
Generalna pravila ... 96
Izravna kontradikcija ... 97
Kontradikcija u ulazu ... 99
Kontradikcija u zaključku ... 100
Zapinjanje temeljeno na pravilima ... 102
Zastoj temeljen na pravilima ... 103
Izolirana pravila ... 104
Beskorisni zaključci ... 105
Viseći uvjeti ... 107
Nepostojeća pravila ... 108
Redundantni podaci ... 109
Izgubljeni podaci ... 111
Nepostojeći podaci ... 113
Nekonzistentni podaci ... 115
Nikad izbrisani podaci ... 117
Dva puta izbrisani podaci ... 119
Podaci neizbrisani na vrijeme ... 120
3.3. Anomalije resursa ... 122
Klasifikacija resursa ... 123
3.3.1. Anomalije obrazaca kreiranja ... 128
Direktna alokacija u slučaju da više resursa ima isti naziv ... 128

Direktna alokacija resursa bez mehanizma eskalacije 129
Alokacija temeljena na ulogama kad odabrani resurs nije pripadnik određene uloge/role 131
Alokacija temeljena na ulogama kad resurs pripada u više od jedne grupe 131
Alokacija temeljena na ulogama kod grupe sa jednim resursom 133
Automatsko izvođenje bez mehanizma provjere izvođenja 134
3.3.2. Anomalije samopokretajućih obrazaca 135
Početak kreiranja sa alokacijom resursa 135
Djelomično lančano izvršenje u poslovnom procesu 137
3.3.3. Anomalije višestrukih obrazaca resursa 138
Istovremeno izvođenje kod ograničenih kapaciteta resursa 138
3.4. Semantičke anomalije 140
3.4.1. Nepravilno nazivlje elemenata 140
Nekonzistentnost nazivlja elemenata 143
Sinonimi u nazivlju elemenata modela 145
Homonimi u nazivlju elemenata modela 147
Nejedinstvenost nazivlja elemenata 150
Neimenovanje elemenata 151
3.4.2. Nepravilne vrijednosti atributa 152
Ne postoji vrijednost atributa 152
Nejedinstvenost vrijednosti atributa 153
Neispravan tip podataka vrijednosti atributa 154
Nedozvoljeni znakovi u vrijednosti atributa 155
3.5. Anomalije prikaza modela 157
3.5.1. Anomalije organizacije percepcije modela 157
3.5.1.1. Anomalije jednostavnosti 157
Previše elemenata u modelu 158
Neselektivni prikaz elemenata u modelu 159
3.5.1.2. Anomalije sličnosti 160
Slični elementi nisu grupirani zajedno 161
Slični elementi nemaju iste boje 162
Slični elementi nemaju istu veličinu 163
Slični elementi nemaju isti oblik 166
3.5.1.3. Anomalija kontinuiranosti 168
Križanje i savijanje veza između elemenata 168
3.5.1.4. Anomalije grupiranja 170
Nekompaktni i razvučeni model 170
3.5.1.5. Anomalije povezanosti 172
Preklapanje elemenata modela 172
3.5.1.6. Anomalije prisnosti 175
Nepoštovanje pozicije hijerarhije elemenata 175
3.5.2. Anomalije percepcije razlikovanja (izdvajanja) 177
3.5.2.1. Anomalija simetrije 177
Necentrirani i nesimetrični model 178
3.5.2.2. Anomalija orijentacije 181
Modeli orijentirani s desna na lijevo 181
Modeli orijentirani od dolje prema gore 183
Vertikalni nazivi elemenata 185
Vertikalni opisi unutar modela 186

3.5.2.3. Anomalija obrisa (konture) 187
Elementi modela se međusobno dodiruju 188
4. SMJERNICE AUTOMATIZACIJE PREPOZNAVANJA I RJEŠAVANJA ANOMALIJA U MODELIMA POSLOVNIH PROCESA 191
5. PRIMJERI PREPOZNAVANJA I RJEŠAVANJA ANOMALIJA NA REFERENTNOM SKUPU MODELA POSLOVNIH PROCESA 202
Primjer modela poslovnog procesa *Pratiti realizaciju po fazama* 209
Primjer modela poslovnog procesa *Nadzirati gradilište* 219
6. ZAKLJUČAK 225
LITERATURA 230

Popis slika

Slika 1. Područje istraživanja ovog rada 3
Slika 2. Tri temeljna elementa u razvoju taksonomije Izvor: Morville, P., Rosenfeld, L.: *Information Architecture for the World Wide Web: Designing Large-Scale Web Sites*, 3nd edition, O'Reilly Media, 2006. 21
Slika 3. Proces razvoja taksonomije 22
Slika 4. Metoda razvoja taksonomije (Izvor: Nickerson, R., Varshney, U., Isaac, H.: *Taxomomy development in information systems: developing a taxonomy of mobile applications*) 23
Slika 5. Odnos modela, jezika i metamodela (Izvor: Overbeek, J.,F.: *Meta Object Facility (MOF) investigation of the state of the art*, Software Engineering, Electrical Engineering, Mathematics and Computer Science, University of Twente, 2006.) 24
Slika 6. OMG hijerarhija metamodela 25
Slika 7. Međuzavisnost MOF i UML infrastrukturnih biblioteka (Izvor: Overbeek, J.,F.: *Meta Object Facility (MOF) investigation of the state of the art*, Software Engineering, Electrical Engineering, Mathematics and Computer Science, University of Twente, 2006.) 27
Slika 8. Paket jezgre sa međuzavisnostima (Izvor: Overbeek, J.,F.: *Meta Object Facility (MOF) investigation of the state of the art*, Software Engineering, Electrical Engineering, Mathematics and Computer Science, University of Twente, 2006.) 28
Slika 9. Virtualni sloj MOF specifikacije 29
Slika 10. Statična struktura elemenata i atributa BPMN metamodela (Izvor: OMG.org) 30
Slika 11. Anomalija ***Više početnih događaja u modelu*** 46
Slika 12. Primjer anomalije ***Više početnih događaja u modelu*** 46
Slika 13. Anomalija ***Nepovezanost početnog događaja*** 47
Slika 14. Primjer anomalije ***Nepovezanost početnog događaja*** 47
Slika 15. Anomalija ***Početni događaj ima više izlaznih sekvenci*** 48
Slika 16. Anomalija ***Nepostojanje završnog događaja*** 49
Slika 17. Primjer anomalije ***Nepostojanje završnog događaja*** 49
Slika 18. Anomalija ***Nepovezanost završnog događaja s ostalim elementima modela*** 50
Slika 19. Primjer anomalije ***Nepovezanosti završnog događaja s ostalim elementima modela*** 50
Slika 20. Anomalija ***Završni događaj ima više od jedne ulazne sekvence*** 51
Slika 21. Anomalija ***Nepostojanje početnog događaja*** 52
Slika 22. Anomalija ***Nepostojanje početnog događaja*** 52
Slika 23. Anomalija ***Ulazne sekvence sa početnim događajem*** 54
Slika 24. Primjer anomalije ***Ulazna sekvenca sa početnim događajem*** 54
Slika 25. Anomalija ***Izlazna sekvenca sa završnim događajem*** 55
Slika 26. Primjer anomalije ***Izlazne sekvenca sa završnim događajem*** 56
Slika 27. Anomalija ***Nepravilna uporaba bazena*** 66
Slika 28. Primjer anomalije ***Nepravilne uporabe bazena*** 66
Slika 29. Anomalija ***Bazen bez elemenata*** 67
Slika 30. Primjer anomalije ***Bazen bez elemenata*** 67
Slika 31. Anomalija ***Nepravilna uporaba staza*** 68
Slika 32. Primjer anomalije ***Nepravilna uporaba staza*** 68
Slika 33. Anomalija ***Staza bez elemenata*** 69
Slika 34. Primjer anomalije ***Staza bez elemenata*** 70
Slika 35. Anomalija ***Nepotrebna kontrola toka*** 71
Slika 36. Primjer anomalije ***Nepotrebna kontrole toka*** 72

Slika 37. Poslovni proces ***Kreirati ugovor*** ... 72
Slika 38. Poslovni proces ***Kreirati ugovor*** sa anomalijom ***Nepotrebna kontrola toka*** ... 73
Slika 39. Anomalija ***Redundancija modela*** ... 75
Slika 40. Primjer poslovnog procesa s anomalijom ***Redundancija modela*** ... 75
Slika 41. Primjer poslovnog procesa koji sliči anomaliji ***Redundancija modela*** ... 76
Slika 42. Anomalija ***Nedefinirani uvjeti vrata*** ... 77
Slika 43. Primjer anomalije ***Nedefinirani uvjeti vrata*** ... 77
Slika 44. Anomalija ***Kontradiktorni uvjeti vrata*** ... 78
Slika 45. Primjer anomalije ***Kontradiktorni uvjeti vrata*** ... 79
Slika 46. Anomalija ***Deterministički nedostatak sinkronizacije*** ... 80
Slika 47. Primjer anomalije ***Deterministički nedostatak sinkronizacije*** ... 81
Slika 48. Anomalija ***Nedeterministički nedostatak sinkronizacije*** ... 83
Slika 49. Anomalija ***Viseća stanja*** ... 84
Slika 50. Primjer anomalije ***Viseća stanja*** ... 84
Slika 51. Anomalija ***Netolerancija na promjene poretka*** ... 85
Slika 52. Anomalija ***Nekonzistentnost kod inserta paralelnih grana*** ... 86
Slika 53. Anomalija ***Nekonzistentnost promjene iz prošlosti*** ... 87
Slika 54. Primjer anomalije ***Nekonzistentnost promjene iz prošlosti*** ... 88
Slika 55. Anomalija ***Nedostatak tolerancije petlji*** ... 89
Slika 56. Primjer anomalije ***Nedostatak tolerancije u petlji*** ... 90
Slika 57. Anomalija ***Zapinjanje toka*** ... 91
Slika 58. Primjer anomalije ***Zapinjanje toka*** ... 91
Slika 59. Anomalija ***Zastoj toka*** ... 93
Slika 60. Primjer anomalije ***Zastoj toka*** ... 93
Slika 61. Anomalija ***Redundantna pravila*** ... 95
Slika 62. Primjer anomalije ***Redundantna pravila*** ... 95
Slika 63. Anomalija ***Generalna pravila*** ... 96
Slika 64. Anomalija ***Kontradikcija na ulazu*** ... 99
Slika 65. Primjer anomalije ***Kontradikcija na ulazu*** ... 99
Slika 66. Anomalija ***Kontradikcija u zaključku*** ... 100
Slika 67. Primjer anomalije ***Kontradikcija u zaključku*** ... 101
Slika 68. Anomalije ***Zapinjanje temeljeno na pravilima*** ... 102
Slika 69. Anomalija ***Zastoj temeljen na pravilima*** ... 103
Slika 70. Primjer anomalije ***Izolirana pravila*** ... 104
Slika 71. Anomalija ***Nepostojeća pravila*** ... 108
Slika 72. Anomalija ***Redundantni podaci*** ... 109
Slika 73. Primjer anomalije ***Redundantni podaci*** ... 110
Slika 74. Anomalija ***Izgubljeni podaci*** ... 112
Slika 75. Poslovni proces sa mogućom anomalijom ***Izgubljeni podaci*** ... 112
Slika 76. Anomalija ***Nepostojeći podaci*** ... 114
Slika 77. Primjer modela poslovnog procesa sa mogućom anomalijom ***Nepostojeći podaci*** 114
Slika 78. Anomalija ***Nekonzistentni podaci*** ... 116
Slika 79. Anomalija ***Nikad izbrisani podaci*** ... 117
Slika 80. Primjer anomalije ***Nikad izbrisani podaci*** ... 118
Slika 81. Anomalija ***Dva puta izbrisani podaci*** ... 119
Slika 82. Anomalija ***Podaci neizbrisani na vrijeme*** ... 120
Slika 83. BPMN 2.0 metamodel resursa ... 122

Slika 84. Odnos između obrazaca alokacija resursa (Izvor: Awad, A; Grosskopf, A, Meyer, A, Weske, M: *Enabling Resource Assigment Contraints in BPMN*, Hasso Plattner Institute, Potsdam, Germany) ... 126
Slika 85. Proširen BPMN metamodel sa podrškom za alokaciju resursa (Izvor: Awad, A; Grosskopf, A, Meyer, A, Weske, M: *Enabling Resource Assigment Contraints in BPMN*, Hasso Plattner Institute, Potsdam, Germany) ... 127
Slika 86. Primjer anomalije ***Direktna alokacija u slučaju da više resursa ima isti naziv*** ... 128
Slika 87. Anomalija ***Direktna alokacija resursa bez mehanizma eskalacije*** ... 130
Slika 88. Anomalija ***Alokacija temeljena na ulogama kad resurs pripada u više od jedne grupe*** ... 132
Slika 89. Primjer ***Alokacije temeljene na ulogama kada pojedina grupa ima samo jedan resurs*** ... 133
Slika 90. Anomalija ***Početak kreiranja sa alokacijom resursa*** ... 136
Slika 91. Anomalija ***Djelomično lančano izvršenje u poslovnom procesu i rješenje*** ... 137
Slika 92. Anomalija ***Istovremeno izvođenje kod ograničenih kapaciteta resursa*** ... 139
Slika 93. Primjer semantičke anomalije ***Nejedinstvenost nazivlja elemenata*** ... 150
Slika 94. Semantička anomalija ***Nejedinstvenost nazivlja aktivnosti*** ... 150
Slika 95. Primjer semantičke anomalije ***Neimenovanje elemenata*** ... 151
Slika 96. Primjer semantičke anomalije ***Neimenovanje elemenata*** ... 151
Slika 97. Primjer anomalije ***Previše elemenata u modelu*** ... 158
Slika 98. Elementi sa potencijalnom ***Anomalijom sličnosti*** ... 160
Slika 99. Anomalija ***Slični elementi nemaju iste boje*** ... 162
Slika 100. Anomalija ***Slični elementi nemaju istu veličinu*** ... 164
Slika 101. Primjer anomalije ***Slični elementi nemaju istu veličinu*** ... 164
Slika 102. Anomalija ***Slični objekti nemaju isti oblik*** ... 166
Slika 103. Primjer anomalije ***Slični objekti nemaju isti oblik*** ... 166
Slika 104. Anomalija ***Križanje i savijanje veza između elemenata*** ... 168
Slika 105. Primjer anomalije ***Križanje i savijanje veza između elemenata*** ... 169
Slika 106. Primjer anomalije ***Nekompaktni i razvučeni model*** ... 170
Slika 107. Primjer anomalije ***Nekompaktni i razvučeni model*** ... 171
Slika 108. Anomalija ***Preklapanje elemenata modela*** ... 172
Slika 109. Primjer ***Anomalije preklapanja elemenata*** ... 173
Slika 110. Shematski prikaz anomalije ***Preklapanja elemenata modela*** ... 173
Slika 111. Anomalija ***Nepoštovanja pozicije hijerarhije elemenata*** ... 175
Slika 112. Primjer anomalije ***Nepoštovanje pozicije hijerarhije elemenata*** ... 175
Slika 113. Shematski prikaz prepoznavanja anomalije ***Nepoštovanje pozicije hijerarhije elemenata*** ... 176
Slika 114. Anomalija ***Necentrirani i nesimetrični model*** ... 178
Slika 115. Primjer ***Necentrirani i nesimetrični model*** ... 178
Slika 116. Shematski prikaz pozicije elemenata kod razrade pravila prepoznavanja anomalije ***Necentrirani i nesimetrični model*** ... 179
Slika 117. Anomalija ***Modeli orijentirani od desna na lijevo*** ... 182
Slika 118. Shematski prikaz prepoznavanja anomalije ***Modeli orijentirani od desna na lijevo*** ... 182
Slika 119. Anomalija ***Modeli orijentirani od dolje prema gore*** ... 183
Slika 120. Shematski prikaz anomalije ***Modeli orijentirani od dolje prema gore*** ... 184
Slika 121. Anomalija ***Vertikalni nazivi elemenata*** ... 185
Slika 122. Anomalija ***Vertikalni opisi unutar modela*** ... 186
Slika 123. Primjer anomalije ***Vertikalni opisi unutar modela*** ... 186

Slika 124. Anomalija obrisa (konture) – ***Elementi modela se međusobno dodiruju*** 188
Slika 125. Primjer anomalije ***Elementi modela se međusobno dodiruju*** 188
Slika 126. Shematski prikaz anomalije ***Elementi modela se međusobno dodiruju*** 188
Slika 127. ***Pratiti realizaciju po fazama*** - po poslovnim funkcijama 212
Slika 128. ***Nadzirati gradilište*** u slobodnoj formi 219

Popis tablica

Tabela 1. Lista OCL osnovnih tipova podataka 33
Tabela 2. Lista OCL operacija s tipovima podataka 34
Tabela 3. Lista OCL operacija 34
Tabela 4. Lista osnovnih OCL funkcija operacija 34
Tabela 5. Lista OCL standardnih operacija 35
Tabela 6. Lista OCL iteracija operacija 35
Tabela 7. Matematičke operacije u pseudokodu 38
Tabela 8. Tabela poretka izvršenja operacija 38
Tabela 9. Operacije uvjeta pseudokoda 38
Tabela 10. Lista operacija i suprotnosti tih operacija u pseudokodu 38
Tabela 11. Lista logičkih operatora u pseudokodu 39
Tabela 12. Klasifikacija anomalija 43
Tabela 13. Pravila dozvoljenih veza sekvence toka (Izvor: OMG.org) 55
Tabela 14. Pravila veza toka poruka (izvor: OMG.org) 57
Tabela 15. Primjer usporedbe atributa objekta 142
Tabela 16. Izgled tablice sa sinonimima 145
Tabela 17. Tablica homonima ***tblHomonymes*** 148
Tabela 18. Redoslijed prepoznavanja i rješavanja anomalija i pripadajućih procedura ***Anomalija prikaza modela*** 193
Tabela 19. Redoslijed prepoznavanja i rješavanja anomalija i pripadajućih procedura ***Sintaktičkih anomalija*** 195
Tabela 20. Redoslijed prepoznavanja i rješavanja ***Semantičkih anomalija*** 197
Tabela 21. Redoslijed prepoznavanja i rješavanja ***Strukturnih anomalija*** 200
Tabela 22. Redoslijed prepoznavanja i rješavanja anomalija i pripadajućih procedura ***Anomalija resursa*** 201
Tabela 23. Primjer obrasca anomalije ***Nepostojanje početnog događaja*** s pravilima prepoznavanja. 202
Tabela 24. Tabela sa redoslijedom i pravilima prepoznavanja ***Anomalija prikaza modela*** .. 203
Tabela 25. Tabela sa redoslijedom i pravilima prepoznavanja ***Sintaktičkih anomalija*** 204
Tabela 26. Tabela sa redoslijedom i pravilima prepoznavanja ***Semantičkih anomalija*** 205
Tabela 27. Tabela sa redoslijedom i pravilima prepoznavanja ***Strukturnih anomalija*** 205
Tabela 28. Tabela sa redoslijedom i pravilima prepoznavanja ***Anomalija resursa*** 207
Tabela 29. Tabela objekata modela poslovnog procesa ***Pratiti realizaciju po fazama*** 210
Tabela 30. Tabela sa redoslijedom i pravilima prepoznavanja ***Anomalija prikaza modela*** za proces ***Pratiti realizaciju po fazama*** 214
Tabela 31. Tabela sa redoslijedom i pravilima prepoznavanja ***Sintaktičkih anomalija*** za model poslovnog procesa ***Pratiti realizaciju po fazama*** 215
Tabela 32. Tabela sa redoslijedom i pravilima prepoznavanja ***Semantičkih anomalija*** za model poslovnog procesa ***Pratiti realizaciju po fazama*** 216
Tabela 33. Tabela sa redoslijedom i pravilima prepoznavanja ***Strukturnih anomalija*** za poslovni proces ***Pratiti realizaciju po fazama*** 216
Tabela 34. Tabela sa redoslijedom i pravilima prepoznavanja ***Anomalija resursa*** za poslovni proces ***Pratiti realizaciju po fazama*** 218
Tabela 35. Detalji poslovnog procesa ***Nadzirati gradilište*** 219
Tabela 36. Tabela sa redoslijedom i pravilima prepoznavanja ***Anomalija prikaza modela*** za proces ***Pratiti realizaciju po fazama*** 220

Tabela 37. Tabela sa redoslijedom i pravilima prepoznavanja ***Sintaktičkih anomalija*** za model poslovnog procesa ***Pratiti realizaciju po fazama*** 221
Tabela 38. Tabela sa redoslijedom i pravilima prepoznavanja ***Semantičkih anomalija*** za model poslovnog procesa ***Pratiti realizaciju po fazama*** 222
Tabela 39. Tabela sa redoslijedom i pravilima prepoznavanja ***Strukturnih anomalija*** za poslovni proces ***Pratiti realizaciju po fazama*** 222
Tabela 40. Tabela sa redoslijedom i pravilima prepoznavanja ***Anomalija resursa*** za poslovni proces ***Pratiti realizaciju po fazama*** 224

Popis kôdova

Kôd 1. Način prepoznavanja anomalije ***Više početnih događaja u modelu*** 46
Kôd 2. Način rješavanja anomalije ***Više početnih događaja u modelu*** 47
Kôd 3. Način prepoznavanja anomalije ***Nepovezanost početnog događaja s ostalim elementima modela*** 48
Kôd 4. Način rješavanja anomalije ***Nepovezanost početnog događaja s ostalim elementima modela*** 48
Kôd 5. Način prepoznavanja anomalije ***Početni događaj ima više izlaznih sekvenci*** 48
Kôd 6. Način rješavanja anomalije ***Početni događaj ima više izlaznih sekvenci*** 49
Kôd 7. Način prepoznavanja anomalije ***Nepostojanje završnog događaja*** 50
Kôd 8. Način rješavanja anomalije ***Nepostojanje završnog događaja*** 50
Kôd 9. Način prepoznavanja anomalije ***Nepovezanost završnog događaja s ostalim elementima modela*** 51
Kôd 10. Način rješavanja anomalije ***Nepovezanost završnog događaja s ostalim elementima modela*** 51
Kôd 11. Način prepoznavanja anomalije ***Završni događaj ima više od jedne ulazne sekvence*** 52
Kôd 12. Način rješavanja anomalije ***Završni događaj ima više od jedne ulazne sekvence*** 52
Kôd 13. Način prepoznavanja anomalije ***Nepostojanje početnog događaja*** 53
Kôd 14. Način rješavanja anomalije ***Nepostojanje početnog događaja*** 53
Kôd 15. Način prepoznavanja anomalije ***Ulazna sekvenca sa početnim događajem*** 54
Kôd 16. Način rješavanja anomalije ***Ulazna sekvenca sa početnim događajem*** 54
Kôd 17. Način prepoznavanja anomalije ***Izlazna sekvenca sa završnim događajem*** 56
Kôd 18. Način rješavanja anomalije ***Izlazna sekvenca sa završnim događajem*** 56
Kôd 19. Način prepoznavanja anomalije ***Početni događaj poruke vezan je tokom poruke na element početnog događaja*** 58
Kôd 20. Način rješavanja anomalije ***Početni događaj poruke vezan je tokom poruke na element početnog događaja*** 58
Kôd 21. Način prepoznavanja anomalije ***Početni događaj poruke vezan je tokom poruke na element bazena*** 59
Kôd 22 Način rješavanja anomalije ***Početni događaj poruke vezan je tokom poruke na element bazena*** 59
Kôd 23. Način prepoznavanja anomalije ***Početni događaj poruke vezan je tokom poruke na element potprocesa*** 59
Kôd 24. Način rješavanja anomalije ***Početni događaj poruke vezan je tokom poruke na element potprocesa*** 60
Kôd 25. Način prepoznavanja anomalije ***Početni događaj poruke vezan je tokom poruke na element aktivnosti*** 60
Kôd 26. Način rješavanja anomalije ***Početni događaj poruke vezan je tokom poruke na element aktivnosti*** 60
Kôd 27. Način prepoznavanja anomalije ***Početni događaj poruke vezan je tokom poruke na element stop događaja poruke*** 60
Kôd 28. Način rješavanja anomalije ***Početni događaj poruke vezan je tokom poruke na element stop događaja poruke*** 61
Kôd 29. Način prepoznavanja anomalije ***Početni pogađaj poruke vezan je tokom poruke na završni element događanja poruke*** 61

Kôd 30. Način rješavanja anomalije ***Početni događaj poruke vezan je tokom poruke na završni element događaja poruke*** 61
Kôd 31. Način prepoznavanja anomalije ***Početni događaj poruke vezan tokom poruke s završnim događajem*** 62
Kôd 32. Način rješavanja anomalije ***Početni događaj poruke vezan je tokom poruke s završnim događajem*** 62
Kôd 33. Način prepoznavanja anomalije ***Bazen je vezan tokom poruke sa završnim događajem*** 62
Kôd 34. Način rješavanja anomalije ***Bazen je vezan tokom poruke sa završnim događajem*** 62
Kôd 35. Način prepoznavanja anomalije ***Potproces je vezan tokom poruke sa završnim događajem*** 63
Kôd 36. Način rješavanja anomalije ***Potproces je vezan tokom poruke sa završnim događajem*** 63
Kôd 37. Način prepoznavanja anomalije ***Aktivnost je vezana tokom poruke sa završnim događajem*** 63
Kôd 38. Način rješavanja anomalije ***Aktivnost je vezana tokom poruke sa završnim događajem*** 63
Kôd 39. Način prepoznavanja anomalije ***Stop događaj je vezan tokom poruke sa završnim događajem*** 64
Kôd 40. Način rješavanja anomalije ***Stop događaj vezan je tokom poruke sa završnim događajem*** 64
Kôd 41. Način prepoznavanja anomalije ***Element završnog događaja spojen je tokom poruke sa završnim događajem*** 64
Kôd 42. Način rješavanja anomalije ***Element završnog događaja spojen je tokom poruke sa završnim događaja*** 65
Kôd 43. Način prepoznavanja anomalije ***Nepravilna uporaba bazena*** 67
Kôd 44. Način rješavanja anomalije ***Nepravilna uporaba bazena*** 67
Kôd 45. Način prepoznavanja anomalije ***Bazen bez elemenata*** 68
Kôd 46. Način rješavanja anomalije ***Bazen bez elemenata*** 68
Kôd 47. Način prepoznavanja anomalije ***Nepravilna uporaba staza*** 69
Kôd 48. Način rješavanja anomalije ***Nepravilna uporaba staza*** 69
Kôd 49. Način prepoznavanja anomalije ***Staza bez elementa*** 70
Kôd 50. Način rješavanja anomalije ***Staza bez elemenata*** 70
Kôd 51. Način prepoznavanja anomalije ***Nepotrebna kontrola toka*** 73
Kôd 52. Način prepoznavanja anomalije ***Nepotrebna kontrola toka*** u *jBoss Drools* verifikatoru pravila (Izvor: Dohring, M., Heublein, S.: *Anomalies in Rule-Adopted Workflows – A Taxonomy and solutuions for vBPMN*, 6th European Conference on Software Maintenance and Reengineering, pages 117-126. Szeged, Hungary : IEEE.) 74
Kôd 53. Način rješavanja anomalije ***Nepotrebna kontrola toka*** 74
Kôd 54. Način prepoznavanja anomalije ***Redundancija modela*** 76
Kôd 55. Način rješavanja anomalije ***Redundancija modela*** 76
Kôd 56. Način prepoznavanja anomalije ***Nedefinirani uvjeti vrata*** 78
Kôd 57. Način rješavanja anomalije ***Nedefinirani uvjeti vrata*** 78
Kôd 58. Način prepoznavanja anomalije ***Kontradiktorni uvjeti vrata*** 79
Kôd 59. Način rješavanja anomalije ***Kontradiktorni uvjeti vrata*** 80
Kôd 60. Način prepoznavanja anomalije ***Deterministički nedostatak sinkronizacije*** 81
Kôd 61. Način rješavanja anomalije ***Deterministički nedostatak sinkronizacije*** (Izvor: Dohring, M., Heublein, S.: *Anomalies in Rule-Adopted Workflows – A Taxonomy*

and solutuions for vBPMN, 6th European Conference on Software Maintenance and Reengineering, pages 117-126. Szeged, Hungary : IEEE.) 82
Kôd 62. Način rješavanja anomalije ***Deterministički nedostatak sinkronizacije*** 82
Kôd 63. Način prepoznavanja anomalije ***Nedeterministički nedostatak sinkronizacije*** 83
Kôd 64. Način rješavanja anomalije ***Nedeterministički nedostatak sinkronizacije*** 83
Kôd 65. Način prepoznavanja anomalije ***Viseća stanja*** 84
Kôd 66. Način rješavanja anomalije ***Viseća stanja*** 85
Kôd 67. Način prepoznavanja anomalije ***Netolerancija na promjene poretka*** 85
Kôd 68. Način rješavanja anomalije ***Netolerancija na promjene poretka*** 86
Kôd 69. Način prepoznavanja anomalije ***Nekonzistentnost kod inserta paralelnih grana*** 87
Kôd 70. Način rješavanja anomalije ***Nekonzistentnost kod inserta paralelnih grana*** 87
Kôd 71. Način prepoznavanja anomalije ***Nekonzistentnost promjene iz prošlosti*** 89
Kôd 72. Način rješavanja anomalije ***Nekonzistentnost promjene iz prošlosti*** 89
Kôd 73. Način prepoznavanja anomalije ***Nedostatak tolerancije u petlji*** 90
Kôd 74. Način rješavanja anomalije ***Nedostatak tolerancije u petlji*** 90
Kôd 75. Način prepoznavanja anomalije ***Zapinjanje toka*** 92
Kôd 76. Način rješavanja anomalije ***Zapinjanje toka*** 92
Kôd 77. Način prepoznavanja anomalije ***Zastoj toka*** 94
Kôd 78. Način rješavanja anomalije ***Zastoj toka*** 94
Kôd 79. Način prepoznavanja anomalije ***Redundantna pravila*** 95
Kôd 80. Način rješavanja anomalije ***Redundantna pravila*** 96
Kôd 81. Način prepoznavanja anomalije ***Generalna pravila*** 97
Kôd 82. Način rješavanja anomalije ***Generalna pravila*** 97
Kôd 83. Način prepoznavanja anomalije ***Izravna kontradikcija*** 98
Kôd 84. Način rješavanja anomalije ***Izravna kontradikcija*** 98
Kôd 85. Način prepoznavanja anomalije ***Kontradikcija na ulazu*** 100
Kôd 86. Način rješavanja anomalije ***Kontradikcija na ulazu*** 100
Kôd 87. Način prepoznavanja anomalije ***Kontradikcija u zaključku*** 101
Kôd 88. Način rješavanja anomalije ***Kontradikcija u zaključku*** 101
Kôd 89. Način prepoznavanja anomalije ***Zapinjanje temeljeno na pravilima*** 102
Kôd 90. Način rješavanja anomalije ***Zapinjanje temeljeno na pravilima*** 102
Kôd 91. Način prepoznavanja anomalije ***Zastoj temeljen na pravilima*** 103
Kôd 92. Način rješavanja anomalije ***Zastoj temeljen na pravilima*** 103
Kôd 93. Način prepoznavanja anomalije ***Izolirana pravila*** 105
Kôd 94. Način rješavanja anomalije ***Izolirana pravila*** 105
Kôd 95. Način prepoznavanja anomalije ***Beskorisni zaključci*** (Izvor: Dohring, M., Heublein, S.: *Anomalies in Rule-Adopted Workflows – A Taxonomy and solutions for vBPMN*, 6th European Conference on Software Maintenance and Reengineering, pages 117-126. Szeged, Hungary : IEEE.) 106
Kôd 96. Način prepoznavanja anomalije ***Beskorisni zaključci*** 106
Kôd 97. Način rješavanja anomalije ***Beskorisni zaključci*** 106
Kôd 98. Način prepoznavanja anomalije ***Viseća pravila*** 107
Kôd 99. Način rješavanja anomalije ***Viseća pravila*** 107
Kôd 100. Način prepoznavanja anomalije ***Nepostojeća pravila*** 108
Kôd 101. Način rješavanja anomalije ***Nepostojeća pravila*** 108
Kôd 102. Način prepoznavanja anomalije ***Redundantni podaci*** 110
Kôd 103. Način rješavanja anomalije ***Redundantni podaci*** 111
Kôd 104. Način prepoznavanja anomalije ***Izgubljeni podaci*** 113
Kôd 105. Način rješavanja anomalije ***Izgubljeni podaci*** 113

Kôd 106. Način prepoznavanja anomalije ***Nepostojeći podaci*** 115
Kôd 107. Način rješavanja anomalije ***Nepostojeći podaci*** 115
Kôd 108. Način prepoznavanja anomalije ***Nekonzistentni podaci*** 116
Kôd 109. Način rješavanja anomalije ***Nekonzistentni podaci*** 116
Kôd 110. Način prepoznavanja anomalije ***Nikad izbrisani podaci*** 118
Kôd 111. Način rješavanja anomalije ***Nikad izbrisani podaci*** 118
Kôd 112. Način prepoznavanja anomalije ***Dva puta izbrisani podaci*** 119
Kôd 113. Način rješavanja anomalije ***Dva puta izbrisani podaci*** 120
Kôd 114. Način prepoznavanja anomalije ***Podaci neizbrisani na vrijeme*** 121
Kôd 115. Način rješavanja anomalije ***Podaci neizbrisani na vrijeme*** 121
Kôd 116. Način prepoznavanja anomalije ***Direktna alokacija u slučaju da više resursa ima isti naziv*** 129
Kôd 117. Način rješavanja anomalije ***Direktna alokacija u slučaju da više resursa ima isti naziv*** 129
Kôd 118. Način prepoznavanja anomalije ***Direktna alokacija resursa bez mehanizma eskalacije*** 130
Kôd 119. Način rješavanja anomalije ***Direktna alokacija resursa bez mehanizma eskalacije*** 130
Kôd 120. Način prepoznavanja anomalije ***Alokacija temeljene na ulogama kad odabrani resurs nije pripadnik određene uloge/role*** 131
Kôd 121. Način rješavanja anomalije ***Alokacija temeljene na ulogama kad odabrani resurs nije pripadnik određene uloge/role*** 131
Kôd 122. Način prepoznavanja anomalije ***Alokacija temeljena na ulogama kad resurs pripada u više od jedne grupe*** 132
Kôd 123. Način rješavanja anomalije***Aalokacije temeljene na ulogama kad odabrani resurs nije pripadnik određene uloge/role*** 132
Kôd 124. Način prepoznavanja anomalije ***Alokacija temeljene na ulogama kada pojedina grupa ima samo jedan resurs*** 133
Kôd 125. Način rješavanja anomalije ***Alokacije temeljene na ulogama kada pojedina grupa ima samo jedan resurs*** 134
Kôd 126. Način prepoznavanja anomalije ***Početak kreiranja sa alokacijom resursa*** 136
Kôd 127. Način prepoznavanja anomalije ***Početak kreiranja sa alokacijom resursa*** 136
Kôd 128. Način prepoznavanja anomalije ***Djelomično lančano izvršenje u poslovnom procesu*** 138
Kôd 129. Način rješavanja anomalije ***Djelomično lančano izvršenje u poslovnom procesu*** 138
Kôd 130. Način prepoznavanja anomalije ***Istovremeno izvođenje kod ograničenih kapaciteta resursa*** 139
Kôd 131. Način prepoznavanja anomalije ***Istovremeno izvođenje kod ograničenih kapaciteta resursa*** 139
Kôd 132. Prepoznavanje anomalije ***Nekonzistentnost nazivlja elemenata*** 144
Kôd 133. Procedura rješavanja anomalije ***Nekonzistentnost nazivlja elemenata*** 145
Kôd 134. Algoritam prepoznavanja anomalije ***Sinonimi u nazivlju elemenata modela*** 146
Kôd 135. Funkcija dohvaćanja sinonima ***GetSynonymes*** 146
Kôd 136. Način rješavanja anomalije ***Sinonimi u nazivlju elemenata modela*** 147
Kôd 137. Način prepoznavanja anomalije ***Nejedinstvenosti nazivlja elemenata*** 147
Kôd 138. Algoritam prepoznavanja anomalije ***Homonimi u nazivlju elemenata modela*** 148
Kôd 139. Funkcija dohvaćanja sinonima ***GetHomonymes*** 149
Kôd 140. Način rješavanja u slučaju duplog objekta u modelu 149

Kôd 141. Način rješavanja anomalije ***Homonimi u nazivlju elemenata modela*** 149
Kôd 142. Način prepoznavanja anomalije ***Nejedinstvenost nazivlja elemenata*** 150
Kôd 143. Način rješavanja anomalije ***Nejedinstvenost nazivlja elemenata*** 151
Kôd 144. Način prepoznavanja anomalije ***Neimenovanje elementa modela*** 152
Kôd 145. Način rješavanja anomalije ***Neimenovanje elementa modela*** 152
Kôd 146. Način prepoznavanja anomalije ***Ne postoji vrijednost atributa*** 153
Kôd 147. Način prepoznavanja anomalije ***Ne postoji vrijednost atributa*** 153
Kôd 148. Način prepoznavanja anomalije ***Nejedinstvenost vrijednosti atributa*** 154
Kôd 149. Način rješavanja anomalije ***Nejedinstvenost vrijednosti atributa*** 154
Kôd 150. Prepoznavanje anomalije ***Neispravan tip podataka vrijednosti atributa*** 154
Kôd 151. Način rješavanja anomalije ***Neispravan tip podataka vrijednosti atributa*** 155
Kôd 152. Prepoznavanje anomalije ***Nedozvoljeni znakovi u vrijednosti atributa*** 155
Kôd 153. Način rješavanja anomalije ***Nedozvoljeni znakovi u vrijednosti atributa*** 155
Kôd 154. Način prepoznavanja anomalije ***Previše elemenata u modelu*** 159
Kôd 155. Način rješavanja anomalije ***Previše elemenata u modelu*** 159
Kôd 156. Način prepoznavanja anomalije ***Neselektivni prikaz elemenata u modelu*** 160
Kôd 157. Način rješavanja anomalije ***Neselektivni prikaz elemenata u modelu*** 160
Kôd 158. Način prepoznavanja anomalije ***Slični elementi nisu grupirani zajedno*** 161
Kôd 159. Način rješavanja anomalije ***Slični elementi nisu grupirani zajedno*** 162
Kôd 160. Način prepoznavanja anomalije ***Slični elementi nemaju iste boje*** 163
Kôd 161. Način rješavanja anomalije ***Slični elementi nemaju iste boje*** 163
Kôd 162. Način prepoznavanja anomalije ***Slični elementi nemaju istu veličinu*** 165
Kôd 163. Način rješavanja anomalije ***Slični elementi nemaju istu veličinu*** 165
Kôd 164. Način prepoznavanja anomalije ***Slični objekti nemaju isti oblik*** 167
Kôd 165. Način rješavanja anomalije ***Slični objekti nemaju isti oblik*** 168
Kôd 166. Način prepoznavanja anomalije ***Križanje i savijanje veza između elemenata*** 169
Kôd 167. Način rješavanja anomalije ***Križanje i savijanje veza između elemenata*** 170
Kôd 168. Način prepoznavanja anomalije ***Nekompaktni i razvučeni model*** 171
Kôd 169. Način rješavanja anomalije ***Nekompaktni i razvučeni model*** 172
Kôd 170. Način prepoznavanja ***Anomalije preklapanje elemenata modela*** 173
Kôd 171. Način rješavanja anomalije ***Preklapanje elemenata modela*** 174
Kôd 172. Način prepoznavanja anomalije ***Nepoštovanje pozicije hijerarhije elemenata*** 176
Kôd 173. Način rješavanja anomalije ***Nepoštivanje pozicije hijerarhije elemenata*** 177
Kôd 174. Način prepoznavanja anomalije ***Horizontalne asimetrije*** 179
Kôd 175. Način prepoznavanja anomalije ***Vertikalne asimetrije*** 180
Kôd 176. Način rješavanja ***Necentrirani i nesimetrični model*** 181
Kôd 177. Način prepoznavanja anomalije ***Modeli orijentirani od desna na lijevo*** 182
Kôd 178. Način rješavanja anomalije ***Modeli orijentirani od desna na lijevo*** 183
Kôd 179. Način prepoznavanja anomalije ***Modeli orijentirani od dolje prema gore*** 184
Kôd 180. Način rješavanja anomalije ***Modeli orijentirani od dolje prema gore*** 184
Kôd 181. Način prepoznavanja anomalije ***Vertikalni nazivi elemenata*** 185
Kôd 182. Način rješavanja anomalije ***Vertikalni nazivi elemenata*** 186
Kôd 183. Način prepoznavanja anomalije ***Vertikalni opisi unutar modela*** 187
Kôd 184. Način rješavanja anomalije ***Vertikalni opisi unutar modela*** 187
Kôd 185. Način prepoznavanja anomalije ***Elementi modela se međusobno dodiruju*** 189
Kôd 186. Način rješavanja anomalije gdje se ***Elementi modela se međusobno dodiruju*** 190
Kôd 187. Prepoznavanje i rješavanje anomalije ***Elementi modela se međusobno dodiruju*** 190

Popis kratica

BPMN	Business Process Model and Notation
BPML	Business Process Modeling Language
DSS	Decision support system
EPC	Event Driven Process Chain
EPM	Enterprise Project Management
GME	Generic Modeling Environment
JWT	Java Workflow Tooling
MDA	Model Driven Architecture
MDD	Model Driven Development
MOF	Meta-Object Facility
MPP	Model poslovnih procesa
NIST	National Institute of Standards and Technology
OCL	Object Constraint Language
OMG	Object Management Group
OMA	Object Management Architecture
PIF	Process Interchange Format
PSL	Process Specification Language
UML	Unified Modeling Language
UPM	Unified Process Model
VDM	Vienna Development Method
XML	EXtensible Markup Language
Z	Formal specification Language
WfMC	Workflow Management Coalition

1. UVOD

U poslovnom svijetu razlozi za izradu funkcionalnih modela mogu biti razni, na primjer: izrada referentnih modela, izrada podloga za unapređenje postojećih poslovnih procesa i organizacijske strukture (eng. *AS-IS*), izrada podloga za planiranje, provjera budućih modela (eng. *TO-BE*), izrada podloga za izradu sustava za odlučivanje (eng. *DSS*), itd. Za modeliranje se koriste razne metodologije i notacije na temelju kojih se razvijaju alati za modeliranje.

Iako su modeli izrađeni u raznim notacijama na prvi pogled različiti, ipak ti modeli imaju neke zajedničke točke kao što su koncepti, elementi, struktura, veze, itd. Dakle, ti različiti modeli su slični po notaciji, sadržaju i strukturi, isto tako ti različiti modeli slični su i po pogreškama tj. anomalijama koje su u njima mogući. Samim po sebi se nameće pitanje na koji način prepoznati anomalije u modelima i kako ih riješiti? Da li postoje anomalije sličnih karakteristika? Da li postoje obrasci njihovog prepoznavanja? Da li postoje metode njihovog rješavanja?

U ovome radu će se koristiti sljedeća definicija: *Anomalije u modelima poslovnih procesa su odstupanja od njihove očekivanje strukture, funkcionalnosti, ponašanja, semantike, uporabe koncepta i njihovog izražavanja između ostalog. Mogu se pojaviti u najrazličitijim fazama oblikovanja modela. Mogu nastati tijekom modeliranja, kada je najčešći uzrok anomalija ljudski faktor ili mehanizmom pretvorbe kod pretvaranja jednog modela u drugi.*

Istraživanje opisano u ovome radu fokusira se na anomalije u modelima poslovnih procesa s ciljem sistematizacije prepoznavanja anomalija i načina njihovog rješavanja. Anomalije su podijeljene u temeljne kategorije i potkategorije na razini detalja koje omogućuju njihovo razlikovanje, opis njihovih specifičnih uzroka i rješenja, kao i njihov izraz u formatu uzoraka.

Ovaj rad bavi se istraživanjem uzroka i vrstama anomalija u modelima poslovnih procesa sa posebnim naglaskom na BPMN 2.0 notaciju. Cilj istraživanja je razrada taksonomije anomalija u svrhu određivanja načina izbjegavanja anomalija i njihovog rješavanja kad se dogode.

1.1. Područje, predmeti i objekti istraživanja

U ovom poglavlju određeno je područje izraživanja i temeljni pojmovi kojima se rad bavi. Fokus ovog rada zalazi u nekoliko različitih područja koje se međusobno isprepliću, kao što su; razvoj temeljen na modelima, modeliranje, programsko inženjerstvo, projektiranje informacijskih sustava, operativno istraživanje, mrežno planiranje, tehničko normiranje, itd.

Razvoj temeljen na modelima (eng. *Model driven development*) oslanja se na ideju pretvorbe modela u programski kod. Takav razvoj automatizirao bi način razvoja računalnih aplikacija i time ubrzao i pojeftinio proces osmišljavanja, razvoja, implementacije i održavanje programskog rješenja u organizacijskoj okolini. Ovakav pristup omogućava ljudima različitih struka i područja djelovanja da zajednički razmotre potrebe i zahtjeve rješavanja pojedinog problema, te da na temelju izgrađenog modela kreiraju programsko rješenje koje omogućava prikupljanje, pohranjivanje, obradu i isporuku informacija potrebnih za određeni problem u organizaciji.

Modeliranje je postupak apstrakcije pojma, situacije ili funkcionalnosti iz realnog svijeta u oblik pogodan za proučavanje. Modeliranje je uvijek vezano uz kontekst koji se modelira, u ovome radu postupak modeliranja vezan je uz oblikovanje modela koji opisuju određeni rad u sustavu tj. organizaciji.

Programsko inženjerstvo je isto područje ovog rada, jer se ovaj rad bavi i pitanjima otkrivanja i rješavanja anomalija u funkcionalnim modelima, a koji zahtjeva primjenu određenih znanja, alata i metoda vezanih uz dizajn i izgradnju aplikacija.

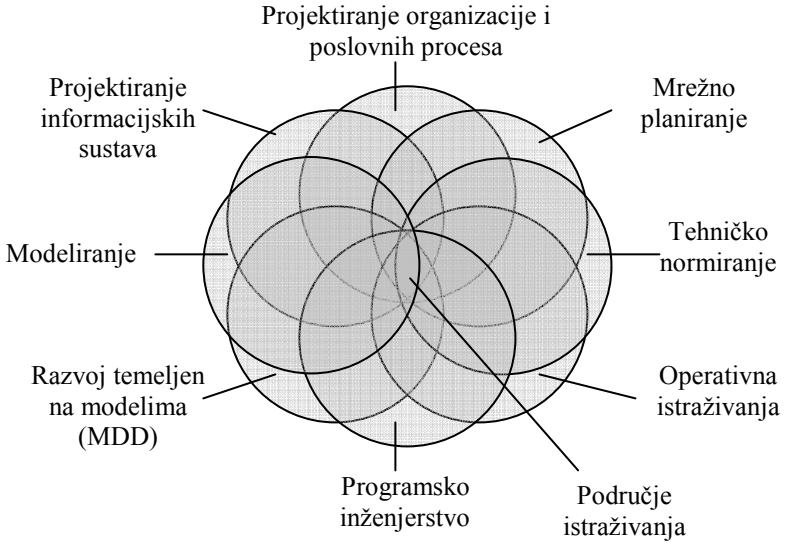
Metode koje se koriste kod **projektiranja informacijskih sustava** također predstavljaju područje istraživanja ovog rada. Neki dijelovi ovog rada bave se područjima operativnih istraživanja, pojedini dijelovi modeliranja poslovnih procesa uključuju i tehničko normiranje i studij rada, mrežno planiranje, planiranje procesa i projektiranje organizacije.

Operativno istraživanje je skup znanstvenih metoda i tehnika kojim se istražuju operacije cijele organizacije s ciljem optimiziranja operacija u skladu s ciljem organizacije.

Mrežno planiranje je tehnika koja omogućava pravovremeno sagledavanje rizika u poslovnom poduzimanju planiranjem i predviđanjem.

Tehničko normiranje je razrada teorije i prakse određivanja tehnički zasnovanih normi vremena i otkrivanja rezervi radnog vremena na osnovu proučavanja i iskorištavanja naprednih oblika organizacije rada, koji osiguravaju najracionalnije iskorištavanje radnog vremena, radne snage, sredstava i predmeta rada [68].

Sve navedeno usmjereno je prema pravilnom razumijevanju i analizi funkcionalnih modela te mogućih anomalija u njima, kao što je prikazano na Slici 1.



Slika 1. Područje istraživanja ovog rada

U ovome poglavlju prikazan je fokus interesa ovog istraživanja. Kao što se iz prethodnih rečenica može vidjeti radi se o vrlo širokom i kompleksnom području gdje istraživači svakodnevno izbacuju nove radove vezane uz najrazličitija istraživanja vezana uz promatrana područja. Istraživanje opisano u ovome radu dati će određeni doprinos upravo na tom promatranom području.

1.2. Temeljni pojmovi i definicije

U prethodnom poglavlju određeno je područje istraživanja ovog rada. U ovome poglavlju opisani su temeljni pojmovi i definicije tih pojmova, a koje su bitne za ovaj rad. Već iz naslova ovog rada vidljivo je da su model i modeliranje temeljni pojmovi ovoga rada.

Model je približni prikaz sustava ili procesa, koji se koristi za pojednostavljivanje sustava u svrhu njegovog mijenjanja ili upravljanja. Model predstavlja pojednostavljenje, odnosno aproksimaciju stvarnosti uzimajući u obzir samo najvažnije dijelove čime stvarnost čini razumljivijim. Model je produkt procesa modeliranja.

Prema Šešiću [59] modeliranje je sistematski istraživački postupak pomoću kojeg se izgrađuje neki stvarni ili idealni znakovni sustav (model), a koji je sposoban:

- zamijeniti predmet koji se istražuje,
- zamijeniti predmet koji daje određenu informaciju o njemu (modelu) i
- zahvaljujući navedenim svojstvima, moguće eksperimentalno istraživati proračune ili logičku analizu.

Kako bi se na taj način dobiveni podaci proširili na pojavu koja se istražuje, kako bi se o njoj stekla pouzdana spoznaja.

Prema Zeleniki [61] osnovni cilj metode modeliranja je da se što točnije i što potpunije spoznaju stvari, pojave i postupci, odnosno njihove strukture, funkcije i ponašanja. Da bi se to postiglo potrebno je u što većoj ili manjoj mjeri ispuniti određene uvjete i to:

- Model mora biti sličan predmetu tj. originalu u materijalu ili strukturi ili u ponašanju u rezultatu funkcioniranja.
- Model mora predstavljati određeni teorijsko-spoznajni ili praktično-realni odraz odnosno original.
- Model mora na temelju prethodna dva uvjeta pružiti određenu informaciju o predmetu odnosno originalu.

Šešić tvrdi da modeliranje odnosno modeli mogu biti [59]:

- **Model stvarnog u idealnom** – misaoni modeli realnih stvari.
- **Model konkretnog u apstraktnom** – predstavlja svaka generalizacija konkretnih pojmova.
- **Model idealnog u realnom** – realne modele predstavljaju ilustracije i interpretacije pojmova, stavova, stvarnih funkcija i algoritama u čulno-opažajnim i fizičkim predmetima, sustavima, odnosima.
- **Model apstraktnog u konkretnom** - predstavljaju sve konkretizacije u specijalizacije apstraktnih pojmova i stavova.

Šešić [59] razlikuje sljedeće vrsta modela:

- **Teorijski ili apstraktni modeli** – logički i matematički modeli izraženi formulama.
- **Praktični ili konkretni modeli** – predstavljaju oruđa ili rezultate određene praktične aktivnosti.
- **Realni modeli** – predstavljaju realne sustave.

- **Idealni modeli** – na idealizirani način predstavljaju originalne predmete.
- **Jednostavni modeli** – imaju jednostavnu strukturu i jednostavne funkcije.
- **Složeni modeli** – imaju složenu strukturu i funkciju.
- **Modeli strukture** – opisuju strukturu originala.
- **Funkcionalni modeli** – opisuju funkcije dinamičkih sustava originala.
- **Djelomični modeli** – opisuju samo neka obilježja originala.
- **Globalni modeli** – opisuju cjelinu predmeta originala.
- **Analitički modeli** – sastoje se od skupa analitičkih relacija u obliku jednadžbi i nejednadžbi kojima se matematički modeliraju dinamički procesi ili ponašanje složenih dinamičkih sustava.
- **Topološki i mrežni modeli** – su modeli u obliku prostornih shema i predstavljaju proizvodne i drugi procese.
- **Deterministički modeli** – imaju strogo određen način funkcioniranja.
- **Stohastički i statistički modeli** – su modelu slučajnih i vjerojatnih događaja, određeni relacijama vjerojatnosti.

Osim navedene podjele, vrlo česta je podjela po **tipu modela** i tada se modeli dijele na [59]:

- mentalne,
- materijalne,
- matematičke,
- konceptualne i
- računalne modele.

Mentalni modeli su konstrukti namijenjeni povezivanju činjenica, a služe predviđanju događaja. **Materijalni modeli** su konkretni fizički modeli koji služe ilustraciji struktura. **Matematički modeli** koriste matematičke koncepte i jezik za opis složenih struktura. Koriste se u prirodnim znanostima, inženjerstvu, ekonomiji, itd. **Konceptualni modeli** sadrže dijagrame struktura i logike rada sustava uz korištenje grafičkih simbola. Konceptualni modeli koji sadrže opise funkcionalnosti sustava nazivaju se funkcionalnim modelima. **Računalni modeli** omogućuju dobivanje kvantitativnih rezultata o ponašanju sustava.

Prema sadržaju, modeli se dijele na: **objektne** i **funkcijske**. **Objektni modeli** prikazuju sastavne dijelove određenih složenih struktura i njihovu hijerarhiju i odnose. **Funkcionalni modeli** su strukturne reprezentacije funkcija, koje mogu biti aktivnosti, radni

koraci, akcije, procesi, operacije unutar sustava. Svrha takvih modela je opisivanje funkcija i procesa, istraživanje, identificiranje i unapređivanje složenih dinamičkih sustava.

Šešić [59] kaže da metoda modeliranja ima tri temeljne funkcije i to:

1. **Primijenjeno-praktičnu funkciju** – gdje se modeli upotrebljavaju u rješavanju praktičnih problema.
2. **Demonstrativna funkcija** – gdje se modeli upotrebljavaju u odgojno obrazovne svrhe.
3. **Znanstveno-istraživačka funkcija** - gdje se metoda modeliranja koristi u znanstveno-istraživačkom radu.

Modeliranje poslovnih procesa i funkcijsko modeliranje koristi se u različitim kontekstima i s različitim ciljevima. Razlozi za modeliranje u poslovnom svijetu mogu biti različiti, najčešće kod razrade poslovnih procesa unutar organizacije, reinženjeringa poslovnih procesa, razrade poslovne tehnologije ili razrade nekog projektnog plana i slično. Tehnike modeliranja, opet mogu biti najrazličitije, na primjer, dijagram toka podataka, EPC, EPM, Petrijeve mreže, BPMN, dijagrami aktivnosti i slično. Sve te tehnike obično su teško spojive, jer svaka od njih nosi neku posebnost koja je primjenjiva za opis točno onih procesa za koje je primarno namijenjena.

Općenito **anomalije** su odstupanja od očekivane ili uobičajene strukture, ponašanja funkcionalnosti, rezultata i sl. Autori Chandola, Banerjee i Kumar u svojem radu *Anomaly Detection: A Survey* definiraju anomalije kao obrasce koji ne zadovoljavaju definiciju normalnog ponašanja [13].

Autori Teng, Chen i Lu u radu *Adaptive real-time anomaly detection using inductively generated sequential patterns* navode da je prepoznavanje anomalija povezano s uklanjanjem šuma iz podataka [60].

Anomalije u modelima predstavljaju obrasce koji uzrokuju odstupanja modela od normalnog ponašanja i koje treba ukloniti.

Svako modeliranje započinje određenom ljudskom aktivnošću u kojem se ručno postavljaju elementi modela i njihove međusobne veze. Kako se radi o ručnom radu vjerojatnost pogreške je poprilično visoka, uzroci tim pogreškama mogu biti najrazličitiji, kao što je nepoznavanje notacije modela, nepreglednost modela, slučajnost, itd. Otkrivanje anomalije sastavni je dio provjere valjanosti modela.

Uzorci dizajna su opće prihvaćena rješenja koja se mogu primijeniti na više različitih mjesta unutar određenog konteksta. To nisu rješenje primjenjivo direktno na određenom

mjestu, već predstavljaju opis ili obrazac koji govori o tome kako riješiti određeni problem koji se može pojaviti u najrazličitijim situacijama. Oni obično predstavljaju najbolju praksu rješavanja određenog problema.

Antiuzorci su obrnuti slučaj, oni su uobičajeni odgovor na određene probleme koji se javljaju u određenom kontekstu, a koji ne daje dobre rezultate i u biti su kontraproduktivni.

Prema autorima uzorka dizajna Gamma, Helm, Johnson, Vlissides. Booch, moraju postojati barem dva ključna elementa prisutna da se razluči antiuzorak od loše navike, loše prakse ili loše ideje, a to su:

- uobičajeno korištena struktura, proces ili aktivnost koja usprkos tome da izgleda dobro rješenje za određeni problem, zapravo daje loše rezultate s lošim posljedicama i
- postoji dobro alternativno rješenje koje je dokumentirano, koje se već koristi i koje se već dokazalo u praksi.

Funkcionalni modeli su strukturne reprezentacije funkcija koje mogu biti aktivnosti, radni koraci, akcije, procesi, operacije unutar sustava. Svrha takvih modela je opisivanje funkcija i procesa, istraživanje, identificiranje i unapređivanje složenih dinamičkih sustava. Modeliranje je temelj inženjerskog projektiranja i predstavlja kreativnu aktivnost koja je najčešće podržana pomagalima za projektiranje kao što su alati, no i dalje postoji velika vjerojatnost nastanka pogreške.

Pojmovi opisani u ovome poglavlju određeni su područjem istraživanja ovog rada. Cilj ovog poglavlja je opisati i definirati temeljne pojmove bitne za istraživanje opisano u ovom radu.

1.3. Pregled dosadašnjih istraživanja i istraživačkih problema

Prethodno poglavlje sadrži pregled i definicije temeljnih pojmova kojima se rad bavi. U ovome poglavlju detaljnije se izlaže problematika istraživanja ovoga rada i pregled dosadašnjih istraživanja i problema na koje su istraživači naišli tijekom svojih istraživanja.

Tijekom analize literature izdvojeno je nekoliko radova koji se bave klasifikacijom anomalija u modelima poslovnih procesa. Solti-Rogge i suradnici u svojem radu *Business process configuration wizard and consistency checker for BPMN 2.0.* [56] istražuju anomalije s ciljem dizajna konfiguracijskog čarobnjaka za poslovne procese i provjeru konzistentnosti modela u BPMN 2.0 notaciji. Oni dijele anomalije u tri temelje kategorije: **nedostatak podataka**, **redundantni podaci** i **konfliktni podaci**. Anomalija pod nazivom **nedostatak podataka** nastaje kad postoji vjerojatnost da se određeni podatak pokušava pročitati u određenoj aktivnosti, a da nikad nije bio spremljen u bazu podataka. Anomalija **redundantni podaci** nastaje kad se pojedini podaci spremaju, ali se nikad ne koriste, odnosno čitaju u nekoj drugoj aktivnosti. **Konflikti** se događaju kad se podatak sprema od strane dvije konkurentne aktivnosti. Iz navedenog može se zaključiti da anomalije povezane sa podacima, generalno ne pokrivaju nepravilnu uporabu koncepata, semantičku nekonzistenciju ili pogrešno izvršenje.

Autori Awad, Decker i Lohmann u svojem radu *Diagnosing and Repairing Data Anomalies in Process Models* [5] predlažu sljedeće kategorije anomalija: **anomalije nastale zbog previše restriktivnih uvjeta**, **anomalije nastale zbog implicitnog preusmjeravanja** i **anomalije nastale zbog implicitnog ograničenja u poretku izvođenja**. **Anomalije nastale zbog previše restriktivnih uvjeta** generalno predstavljaju anomalije podataka. Nastaju kad je pojedina aktivnost spremna na izvršenje, no podatak koji se koristi u toj aktivnosti nije u očekivanom stanju. **Anomalije nastale zbog implicitnog preusmjeravanja** nastaju u slučaju kad stanje podataka u podatkovnim objektima određuje koja grana sa aktivnostima će biti izvedena, a istovremeno svaka aktivnost ima određena zasebna podatkovna ograničenja. **Anomalija nastala zbog implicitnog ograničenja** nastaje kad dvije kompetitivne aktivnosti dijele isti preduvjet, a problem nastaje kad pojedina aktivnost promijeni stanje podataka.

Autori Kim, Lee i Soon u radu *Classification and analyses of business process anomalies* predlažu dvije temeljne kategorije anomalija poslovnih procesa i to: **sintaktičke** i **strukturne anomalije** [29]. **Sintaktičke anomalije** su vezane uz nepravilnu uporabu koncepata i pravila modeliranja. **Strukturne anomalije** nastaju kad poslovni procesi ne mogu

kontrolirati interakciju tokena zbog nepravilne uporabe vrata. Autori dodaju da većina verifikatora modela rade isključivo sa I i ILI vratima.

Autori Dohring i Heublein u radu *Anomalies in Rule-Adopted Workflows – A Taxonomy and solutions for vBPMN* spominju tri temeljne kategorije anomalija i to: **anomalije kontrole toka, anomalije pravila i anomalije podataka** [16].

Iz navedenog može se zaključiti da postoje različiti pristupi proučavanju anomalija u modelu procesa. Možemo se zaključiti sljedeće:

- **Teorijski pristup** - temelji se na potrazi konzistentnih i jasnih pravila, metoda, tehnika i drugih teorija u modelima poslovnih procesa. Ako ispravna primjena rezultira pojavom anomalija, onda je nužna promjena ili ispravka pravila ili nekog drugog dijela teorije modeliranja.
- **Empirijski pristup** - temelji se na primjeni analitičkih i dijagnostičkih metoda na realnim projektima i modelima. Ovaj pristup traži identifikaciju uzroka anomalija i odstupanja između modela i realnosti. Krajnji cilj je određivanje najbolje prakse koja rezultira modelima bez anomalija.
- **Pristup dizajnom** – temelji se na primjeni teorije uzoraka tijekom oblikovanja modela. Teži definiranju neprihvatljivih uzoraka dizajna ili antiuzoraka. Prisustvo antiuzoraka indicira anomalije. Uzorci postaju elementi dizajna, a antiuzorci predstavljaju zabranjene cjeline ili cjeline koje treba koristiti uz posebni oprez.

Kada se govori o anomalijama u funkcionalnim modelima, postoji nekoliko radova koje treba spomenuti, a koje će se kasnije detaljnije obraditi. Pa tako autori Solti-Rogge, Kunze, Awad, Weske u svojem radu *Business process configuration wizard and consistency checker for BPMN 2.0.* [56] navode temeljne kategorije anomalije. Autori Awad, Decker i Lohmann u svojem radu *Diagnosing and Repairing Data Anomalies in Process Models* [5] daju daljnju kategorizaciju anomalija. Autori Kim, Lee i Soon u svojem radu *Classification and analyses of business process anomalies* [29] navode dvije temeljne kategorije anomalija u poslovnim procesima. Autori Dohring i Heublein u radu *Anomalies in Rule Adopted workflows – A Taxonomy and solutions for vBPMN* [16] navode tri temeljne kategorije anomalija. Svi navedeni radovi tvore temelj na kojem će se u ovome radu pokušati predložiti detaljnija taksonomija anomalija u funkcionalnim modelima.

Osim toga, s obzirom da će se u anomalije funkcionalnih modela uvrstiti i anomalije resursa, bitan rad je rad autora Awad, Grosskopf, Meyer i Weske-a pod nazivom *Enabling*

Resource Assignment Constraints in BPMN u kojem autori predlažu prošireni BPMN metamodel elementima bitnim za određivanje pravila, kojim bi se izbjegli problemi s alokacijom resursa u BPMN modelima [6].

Kao što je prije navedeno, postupak pretvorbe modela jedan je od uzroka nastanka anomalija. Uporaba jednog modela kao izvorište za kreiranje drugog modela nije nova. Postoji cijeli niz znanstvenih radova u kojima se razrađuje tema pretvorbe modela. Tako se primjerice autor Dumas u radu *Case Study: BPMN to BPEL Model Transformation* [18] bavi pretvaranjima modela između dvaju jezika različite strukture i primjene. Ovdje se radi o vertikalnom pretvaranju modela. U razmatranje uzima četiri bitna kriterija važna za razvoj, a to su: kompletnost, ispravnost, čitljivost i reverzibilnost. Ta transformacija je jedna od temeljnih za uporabu BPMN koncepta i jedna je od prvih nastalih uopće sa BPMN notacijom.

Jedan on načina prethodno definiranog problema je primjena metamodela. Prema Seidewitzu [54] metamodel je model koji opisuje jezik kojim se neki model modelira. Metamodel opisuje moguće modele koji mogu biti jezikom koji definira metamodel. Upotrebu metamodela u postupku pretvorbe jednog modela u drugi opisao je Eessaar u svojem radu *Using Metamodeling in order to Evaluate Data Models* [20]. Autor navodi da je uporaba metamodela prilično raširena u inženjeringu aplikativnih rješenja. Metamodel je „model modela“ koji daje pravila, odnosno gramatiku samog jezika modeliranja. Prednosti uporabe metamodela koje autor navodi su:

- Kreiranjem metamodela zapravo se kreira specifikacija modela.
- Metamodel vizualizira skrivene koncepte modela.
- Izradom metamodela u vizualnim jezicima kao što su UML ili MOF, olakšava se razumijevanje metamodela u širim profesionalnim krugovima.
- Metamodel se može koristit u svrhu učenja.
- Metamodele je moguće uspoređivati (mapiranjem, metrikom).
- Metamodel može pripomoći unapređenju modela, jer sam postupak metamodeliranja zahtjeva proučavanje postojećeg modela, što obično rezultira pronalaženjem grešaka i mana postojećeg modela.

Uporabu MOF (eng. *Meta Object Facility*) notacije za izradu metamodela opisuju Emerson i Sztipanovits u radu *Implementing a MOF-Based Metamodeling Environment Using Graph Transformations* [21] koji se bavi ulogom grafičkih transformacija u novim standardima modeliranja. U svojem radu daju primjer pretvorbe MOF i GME (eng. *Generic Modeling*

Environment) metamodela. Principi usporedbe dobro su opisani od strane autora Breton i Bézivin u radu *An Overview of Industrial Process Meta-Models* [11] u kojem uspoređuju više različitih koncepata kao što su: PIF, NIST, PSL, UPM, WfMC sa UML i MOF konceptom. Kretschmer u članku *JWT metamodel compared to BPMN metamodel* [30] uspoređuje dva koncepta i to JWT (eng. *Java Workflow Tooling*) i BPMN na razini metamodela. Upravo navedeni radovi dobro opisuju pristup izgradnje i usporedbe više koncepata na razini metamodela.

Kao bitan izvor nastanka anomalija je pretvorba modela, a bitan dio pretvorbe jednog koncepta u drugi je pronalaženje strukturnih i semantičkih sličnosti i razlika dvaju koncepata. Upravo na tom području postoji cijeli niz radova na temelju kojih su razvijeni postupci usporedbe i mapiranja dvaju modela. Autori Awad, Decker i Lohmann u svojem radu *Diagnosing and Repairing Data in Process Models* [5] bave se otkrivanjem i ispravkom podataka u modelima poslovnih procesa. Kao primjer uzimaju mapiranje BPMN notacije i Petrijevih mreža, te daju pregled pretvorbenih obrazaca za navedene koncepte. S obzirom da će se i u ovome radu koristiti mapiranje metamodela, ovaj rad je bitan da se vidi uporaba mapiranja u potvrdi određenog koncepta. Mapiranje metamodela je transformacija, odnosno pretvorba preslikavanjem koncepata između različitih komponenata pojedinih jezika, zatim između sintakse i koncepta modela, pa između modela i pojedine instance komponente. U svrhu boljeg shvaćanja mapiranja modela u obzir je uzet i rad autora Bitsaki i suradnika koji u radu pod naslovom *Model Transformations to Leverage Service Networks* [9] predstavljaju model transformacije za kreiranje poslovnih procesa koji bi se izvodili između poslovnih partnera, a koji uključuju servisne mreže. Vrlo bitan rad za izradu okvira pretvorbe je i rad autora Diskin i Dingeu koji se u svojem radu *A metamodel independent framework for model transformation: Towards generic model management patterns in reverse engineering* [14] bave izradom integralnog okvira u kojem se model pretvaranja temelji na generičkom pristupu u kojem su metamodeli u biti parametri procedure pretvaranja. Temelj te procedure je model pretvaranja baziran na mapiranju dviju metamodela. Dobro je uvidjeti i pitanje proširenja jezika metamodeliranja, pa su autori Hausmann i Kent u svojem radu *Metamodeling Relations - Relating metamodels?* [23] predložili proširenje jezika metamodeliranja specificirajući mapiranje između dva metamodela. Cilj tog proširenja je pouzdanija i konzistentnija pretvorba između metamodela radi lakšeg održavanja. Dok autori Celms, Kalnins i Lace u radu *Diagram definition facilities based on metamodel mappings* [12] predlažu novu tehniku za definiranje dijagrama u generičkom alatu za modeliranje koji

dozvoljava izradu nekoliko različitih grafičkih prezentacija iste domene radi nedvosmislenog opisa određenog procesa. Prilikom mapiranja dvaju modela dolazi i do određenih nepredviđenih situacija kojima se bave autori: Wang i Wang u svojem radu *Detecting artifact anomalies in business process specifications with a formal model* [67] u kojem se bave problemima koji nastaju zbog neprecizne specifikacije elemenata u notacijama za modeliranje poslovnih procesa. U tome radu bave se postavljanjem modela za opis ulaza i izlaza procesnih tokova i analize uporabe grafičkih elemenata u modelu poslovnih procesa.

Istraživanje u okviru ove disertacije pokazalo je da postoje informatička rješenja koja omogućavaju pretvorbu jednog funkcijskog modela u drugi. Spomenimo autora Hammoudi i suradnike koji su u članku *Metamodel Matching: Experiments and Comparison* [24] opisali principe rada svojeg alata za usporedbu i pretvaranje dvaju funkcijskih modela. Sličan pristup imaju i autori Mork, Bernstein i Melnik u radu *Translating Schemas and Data between Metamodels* [39] koji predstavljaju alat *ModelGen* koji automatski prevodi izvorni metamodel u odredišni metamodel. Model metamodela mora biti definiran u XML shemi, a alat može generirati poglede modela u oba smjera. Kod razvoja alata za automatizaciju pretvorbe modela bitno je odrediti verifikacijska pravila i metode. Autori Lohmann, Verbeek i Dijkman u svojem radu *Petri Net Transformations for Business Processes – A Survey* [34] bave se međusobnom pretvorbom grafičkih jezika za modeliranje. Jezike za modeliranje poslovnih procesa dijele na dvije kategorije: akademske i poslovne. Akademske jezike preferiraju znanstvenici u njih spadaju Petrijeve mreže i procesna algebra. Posebne su po tome da imaju detaljnu formalnu semantiku koja omogućava formalnu verifikaciju modela. Poslovni jezici za modeliranje poslovnih procesa imaju primjenu u poslovnim okolinama, a znanstvenici ih obično izbjegavaju, jer imaju slabiju semantiku što može dovesti do dvosmislenosti modela. U tu grupu grafičkih jezika za modeliranje poslovnih procesa spadaju BPMN, EPC, itd. Cilj njihovog rada je primijeniti verifikacijske tehnike iz Petrijevih mreža na modele izvedene u jezicima za modeliranje poslovnih procesa koji imaju slabiju semantiku.

Za verifikaciju bitan je doprinos autora Dijkman, Dumas i Ouyang koji se u svojem radu *Formal Semantics and Analysis of BPMN Process Models* [15] bave problemima nastalih zbog manjkavosti semantike u BPMN modelima, gdje predlažu formalnu semantiku temeljenu na Petrijevim mrežama. Verifikacijom se bavi i autor van Aalst, koji je jedan od najznačajnijih autora u području teorije uzoraka, a u radu *Verification of workflow nets* primjenjuje teoriju Petrijevih mreža za verifikaciju mrežnih tokova [1]. Osim za pretvorbu BPMN koncepta u koncept mrežnog planiranja, bitan je i pogled na stanje u semantici

dijagrama aktivnosti čime se posebno bave autori Vitolins i Kalnins u svojem radu *Semantics of UML 2.0 Activity Diagram for Business Modelling by Means of Virtual Machine* [66], a koji predlažu formalnu definiciju semantike UML 2.0 dijagrama aktivnosti. Svoj rad temelje na metodologiji kretanja tokena kroz dijagram. Token, odnosno značka predstavlja poslovni slučaj koji služi za opis dinamike poslovnog procesa. Broj tokena u određenoj aktivnosti pokazuju količinu zauzeća resursa.

O pretvaranju jednog koncepta u drugi, te sljedivosti i problemima koji pri tome nastaju piše i Hidaka iz Japanskog nacionalnog instituta za informatiku u svojem radu *On the traceability in graph roundtrip transformation system GRoundTram* [26] razrađuje dvosmjerni model transformacije, te predlaže kompozitni okvir za dvosmjernu transformaciju modela koji se temelji na metamodeliranju. Njegov rad uključuje round-trip ili povratni inženjering čime bi se omogućilo dvosmjerno pretvaranje modela. U radu *Round-Trip Engineering for Maintaining Conceptual-Relational Mappings* [3] autori An, Hu i Song bave se problemima koji nastaju kod održavanja mapiranih modela i njihovih pretvaranja jedan u drugi. U tome radu oni predlažu rješenje temeljeno na round-trip inženjeringu koje bi omogućavalo sinkronizaciju modela držeći ih konzistentnima za održavanje mapiranjem semantički kompatibilnim instancama. Autori u radu navode nekoliko primjera koji ilustriraju način rada predloženog rješenja. Zanimljiva je i doktorska disertacija autora Hettel naslova *Model Round-Trip engineering* [25] u kojem razlaže temeljne pojmove *round-trip* inženjeringa.

Za sve notacije i jezike koje se spominju u ovome radu postoje javno objavljeni standardi i specifikacije koje će se koristiti radi preciznih i nedvosmislenih definicija pojedinih metamodela. Dakle, bitne su specifikacije za BPMN 2.0 [8] [43] notaciju i UML specifikacija za metamodele [63]. Nadalje, osim specifikacija za koncepte modela bitni su i detalji vezani uz standarde metodologije koji će se koristiti za dokazivanje hipoteza rada. Za prikaz metamodela koristiti će se MOF standard [37], čija specifikacija se može naći na stranicama OMG konzorcija. Osim MOF-a koristiti će se OCL jezik [44] koji će služiti za definiranje pravila i ograničenja pretvorbi.

Iz prethodnih rečenica vidljivo je da postoji dosta radova na temu problematike istraživanja ovog rada. Područje istraživanja ovog rada dosta je aktualno i istraživači svakodnevno izbacuju nove radove čime se proširuje i produbljuje znanje vezano uz ovo područje istraživanja. Ovaj rad upravo u tom smislu i doprinosi proširivanju i produbljivanju znanja iz navedenog područja istraživanja.

1.4. Hipoteze i očekivani znanstveni doprinos

U prethodnom poglavlju detaljnije su izložena dostupna istraživanja brojnih autora na temu problematike ovog rada. U ovome poglavlju postavljaju se hipoteze, definiraju se ciljevi istraživanja i izlaže se znanstveni doprinos ovog rada.

Istraživački problem i osnovni cilj ovog istraživanja je određivanje načina prepoznavanja anomalija u modelima poslovnih procesa. Prepoznavanjem anomalije može se navedenu anomaliju smjestiti u određenu kategoriju odnosno vrstu anomalija. Na temelju vrste, odnosno kategorije anomalije može se predložiti način njenog rješavanja. U ovome radu posebno će se se baviti BPMN 2.0 notacijom i prepoznavanjem anomalija u modelima poslovnih procesa s ciljem njihovog rješavanja. Kako bi se odgovorilo na osnovni problem istraživanja, postavljene su odgovarajuće hipoteze, koje će biti testirane u empirijskom dijelu istraživanja.

H1: Nova taksonomija anomalija, koja se temelji na uzorcima dizajna, omogućava prepoznavanje i razvrstavanje svih anomalija na referentnom skupu modela poslovnih procesa.

H2: Redoslijed provjere uzoraka anomalija u modelima poslovnih procesa utječe na uspješnost i učinkovitost postupaka otklanjanja anomalija.

Ciljevi istraživanja su:

- istraživanje i sistematizacija literature o anomalijama u procesnim modelima,
- definiranje osnovnih pojmova vezanih uz anomalije procesnih modela,
- razrada taksonomije anomalija na temelju metode koja se temelji na teoriji uzoraka dizajna,
- određivanje pravila prepoznavanja i otklanjanja anomalija,
- prijedlog koncepta automatizacije prepoznavanja i otklanjanja anomalija.

Znanstveni doprinosi ovog istraživanja su:

- Cjelovita taksonomija anomalija u modelima procesa sa posebnim naglaskom na BPMN 2.0 notaciju.

- Uzorak svake od anomalija u taksonomiji.
- Konsolidirana pravila, metode prepoznavanja i procesni model rješavanja anomalija, na temelju ekspertne procjene, teorije uzoraka i metamodela.
- Smjernice za automatizaciju prepoznavanja i rješavanja anomalija u modelima procesa, na temelju definirane taksonomije, te metoda prepoznavanja i rješavanja anomalija.

Iz gore navedenoga može se zaključiti, da je znanstveni doprinos ovog istraživanja u izgradnji koncepta prepoznavanja anomalija u modelima poslovnih procesa, njihovo svrstavanje u određenu kategoriju koja ima određena obilježja pomoću kojeg se može predložiti i način njihovog rješavanja. Ovakav pristup rješavanju anomalija u modelima poslovnih procesa do sad nije prikazan u dostupnoj literaturi. Za postizanje ciljeva istraživanja i potvrđivanje hipoteza potrebno je razraditi cijelu taksonomiju anomalija sa katalogom mogućih anomalija i sa razrađenim pravilima prepoznavanja. Posebni znanstveni doprinos je u formiranju kataloga anomalija resursa, anomalija sekvenci i anomalija prikaza modela koje do sad nisu obrađivane na ovakav način. Također, znanstveni doprinos ovog istraživanja je u izgradnji koncepta prepoznavanja i rješavanja anomalija u modelima poslovnih procesa kojeg je jednostavno ugraditi u alate za modeliranje s ciljem poboljšanja postojećih mehanizama validacije i verifikacije modela poslovnih procesa.

1.5. Metodologija i plan istraživanja

U prethodnom poglavlju postavljene su hipoteze, ciljevi i predstavljen je doprinos ovog rada. U ovome poglavlju biti će izložena metodologija i plan istraživanja s ciljem realizacije istraživanja opisanog u ovome radu.

Plan istraživanja korišten u ovome radu sastoji se od niza istraživačkih aktivnosti, od kojih su najvažnije:

- odabir područja istraživanja, istraživačkog problema i ciljeva istraživanja,
- prikupljanje i analiza literature o dosadašnjim istraživanjima u zadanom području, te provjera relevantnosti istraživačkog problema,
- definiranje kriterija za razvrstavanje anomalija i početne taksonomije metodom analize literature i komparacije postojećih taksonomija,
- izrada uzoraka pojedinih anomalija metodom modeliranja,
- određivanje pravila prepoznavanja anomalija primjenom teorije uzoraka, upotpunjenom drugim metodama (ekspertna procjena, metamodeliranje),
- provjera taksonomije, uzoraka i drugih specifikacija anomalija na odabranom skupu modela poslovnih procesa,
- poboljšanje taksonomije, uzoraka i drugih specifikacija anomalija u skladu s rezultatima provjere,
- definiranje, provjera i poboljšanje pravila rješavanja za svaku pojedinu anomaliju,
- istraživanje utjecaja redoslijeda provjere uzoraka anomalija u modelima poslovnih procesa na uspješnost i učinkovitost postupaka otklanjanja anomalija,
- izrada smjernica za automatizaciju prepoznavanja i rješavanja anomalija u modelima procesa, na temelju definirane taksonomije te metoda prepoznavanja i rješavanja anomalija i
- sređivanje i dokumentiranje rezultata istraživanja.

Odabir područja istraživanja, istraživačkog problema i ciljeva istraživanja odnosi se na fazu istraživanja prije prijave teme doktorske disertacije. Područje istraživanja je multidisciplinarno, jer istraživanje ulazi u domenu modeliranja, projektiranja informacijskih sustava, projektiranja organizacije i poslovnih procesa, mrežnog planiranja, tehničkog normiranja, operativnih istraživanja, programskog inženjerstva i razvoja temeljenih na

modelima. Navedeno istraživanje predstavlja nastavak i daljnje proširenje dosadašnjeg rada autora na temu anomalija u modelima poslovnih procesa.

Tijekom istraživanja koja su prethodila prijavi teme najvećim je dijelom završena i aktivnost **prikupljanja i analize literature o dosadašnjim istraživanjima u zadanom području te provjera relevantnosti istraživačkog problema.** Prikupljeni su i analizirani dostupni znanstveni radovi o anomalijama i njihovim karakteristikama te o načinu njihovog rješavanja. Cilj ovog koraka je sakupiti građu bitnu za izgradnju taksonomije anomalija i koncepta njihovog rješavanja.

Pri **definiranju kriterija za razvrstavanje anomalija i izradi početne taksonomije metodom analize literature i komparacije postojećih taksonomija** koriste se postojeće metode izgradnje taksonomija, kod kojih se određuju kriteriji razvrstavanja predmeta taksonomije, u ovom slučaju anomalija u modelima poslovnih procesa, zatim se određuju zajednička obilježja pojedinih skupina, rješava se problem višestrukih obilježja i gradi hijerarhijska klasifikacija. Cilj ove aktivnosti je dobiti taksonomiju pomoću koje je moguća daljnja razrada razreda anomalija ovisno o kriterijima i zajedničkim karakteristikama. Ova se taksonomija upotpunjava i poboljšava kroz više iteracija.

Cilj **izrade uzoraka anomalija metodom modeliranja** je razrada jednog ili više uzoraka za svaku anomaliju iz taksonomije. Dakle, svaka se anomalija modelira u BPMN 2.0 notaciji koristeći odabrani alat za modeliranje poslovnih procesa. Ovim se postupkom teorijski modeli upotpunjuju grafičkim modelima u zadanoj notaciji. Pritom se provjerava jednoznačnost pridruživanja tekstualnog opisa anomalije i pratećih specifikacija uzorka. U radu će se koristiti standardizirane notacije i jezici modeliranja, eventualno uz primjenu mehanizama proširenja dozvoljenih notacijom odnosno jezikom. Uz BPMN 2.0 notaciju za modeliranje poslovnih procesa, za razradu metamodela koristit će se MOF [37] te jezici UML [63] i OCL [44], čije se specifikacije mogu naći na stranicama OMG konzorcija.

Nakon izrade uzoraka, slijedi **aktivnost određivanja pravila prepoznavanja anomalija primjenom teorije uzoraka, upotpunjenom drugim metodama** (ekspertna procjena, metamodeliranje). U ovoj aktivnosti, ovisno o karakteristikama pojedine anomalije, određuje se skup pravila za prepoznavanje anomalije u modelu poslovnih procesa. Pravila mogu biti definirana kao skup ograničenja u metamodelu ili neformalno, kao skup pravila za koja se primjenjuju tijekom modeliranja poslovnih procesa. Za razradu pojedinih pravila koristit će se metoda metamodeliranja, odnosno modeliranja koncepata odabrane metode

modeliranja, te izrađeni uzorci anomalija. Osim toga za opis pojedinih pravila koristiti će se opisi razrađeni u pseudokodu.

Slijedi **provjera taksonomije, uzoraka i drugih specifikacija anomalija na odabranom skupu modela poslovnih procesa**. Odabrani skup modela poslovnih procesa sadrži sve primjere procesnih modela kojima se u prikupljenoj literaturi ilustriraju pojedine anomalije. Sekundarni skup modela poslovnih procesa mogu biti modeli iz stvarnih projekata, koje za potrebe provjere prilagodi ekspert različit od istraživača, uvodeći anomalije u model.

Poboljšanje taksonomije, uzoraka i drugih specifikacija anomalija u skladu s rezultatima provjere obavit će se kroz nekoliko iteracija, na temelju rezultata prethodnih aktivnosti.

Za **definiranje, provjeru i poboljšanja pravila rješavanja anomalija** koristit će se metoda metamodeliranja i specificiranja ograničenja u metamodelima, utemeljena na MOF-u i UML-u. Krajnji rezultat ove aktivnosti je jedna ili više provjerenih metoda rješavanja svake pojedine anomalije. Pojedina složenija pravila rješavanja anomalija biti će opisana u pseudokodu.

Sljedeća aktivnost je **istraživanje utjecaja redoslijeda provjere uzoraka anomalija u modelima poslovnih procesa na uspješnost i učinkovitost postupaka otklanjanja anomalija**, čime će se potvrditi ili opovrgnuti hipoteza da takav utjecaj postoji. Kako bi se ova aktivnost provela, nužno je izraditi prijedlog procesnog modela prepoznavanja i otklanjanja anomalija.

Osnovni rezultat prethodnih aktivnosti je cjelovita taksonomija anomalija procesnih modela, s pratećim uzorcima i drugim specifikacijama, pogodna za prepoznavanje anomalije i njihovo rješavanje. Drugi važan rezultat su pravila i procesni model prepoznavanja i otklanjanja anomalija, na temelju kojih će se **izraditi smjernica za automatizaciju prepoznavanja i rješavanja anomalija**.

Sređivanje i dokumentiranje rezultata istraživanja je završna faza istraživanja, čiji je cilj izrada konačnog teksta disertacije s pripadajućim prilozima.

U predhodnim rečenicama izložena je metodologija i koraci plana istraživanja s ciljem razrade realizacije istraživanja opisanog u ovome radu.

2. METODE ISTRAŽIVANJA ANOMALIJA U MODELIMA POSLOVNIH PROCESA

U prethodnim poglavljima izloženi su područje, predmeti i objekti istraživanja, zatim prikazane su definicije temeljnih pojmova ovog rada, zatim izložen je pregled dosadašnjih istraživanja, postavljene su hipoteze i ciljevi ovog rada i razrađena je metodologija i plan istraživanja. U ovome poglavlju fokus je na metodama istraživanja anomalija u modelima poslovnih procesa. Posebno se razrađuje metodologija oblikovanja taksonomije, zatim metode, pravila i postupci koje ćemo koristiti za prepoznavanje i rješavanje anomalija u modelima poslovnih procesa.

Anomalije u modelima poslovnih procesa mogu se pojaviti tijekom: modeliranja, modificiranja i pretvorbe modela. **Modeliranje** predstavlja proces u kojem se objekti ili pojave iz realnog svijeta prikazuju na način pogodan za promatranje i analizu. To je proces koji zahtjeva manualni rad i gdje su greške česte. Nakon što je model kreiran obično je model potrebno modificirati kako bi se što bolje naglasile promatrane funkcionalnosti. Kad je dobiven model dovoljno detaljan da obuhvaća sve promatrane funkcionalnosti obično slijedi **modificiranje** tog temeljnog modela s ciljem unapređenja postojećeg modela. Obično se tu radi o eksperimentiranju u kojem se odvajaju nove funkcionalnost, oduzimaju postojeće, mijenjaju određeni radni koraci, dodaju elementi odlučivanja, itd. U toj fazi isto ima dosta grešaka koje mogu prouzročiti anomalije funkcionalnih modela. Naime, modificiranje modela obično predstavlja manualni rad koji zahtjeva promišljanje osoba uključenih u modeliranje, poštujući notaciju kojom grade model. Kod automatiziranog modificiranja modela obično se radi o nekakvim spajanjima više modela u jedan ili o reduciranju modela u kojem se određeni dijelovi modela miču iz osnovnog modela. Nadalje, postupak u kojem mogu nastati anomalije je **pretvorba** modela. Greške kod pretvorbe modela moguće su kod pretvaranja jedne grafičke notacije u drugu, pretvorbe jednog jezika u drugi i jedne perspektive promatranja u drugu.

Obično su postupci pretvaranja modela automatizirani u nekim od dostupnih alata. Problemi s anomalijama pogotovo dolaze do izražaja u slučaju pretvorbe modela iz jedne notacije u drugu. U tom slučaju različite namjene pojedinih notacija obično diktiraju i naglasak detalja na pojedinim elementima modela. Iz tog razloga može doći do gubitka informacija tijekom te transformacije.

2.1. Metodologija oblikovanja taksonomije

Cilj ovog poglavlja je opisati pojmove i metodologiju oblikovanja taksonomije. Ovo poglavlje mora dati odgovor na koji način treba oblikovati taksonomiju anomalija, te koji su koraci oblikovanja taksonomije anomalija.

Taksonomija dolazi od grčke riječi *tassein* što znači svrstati i riječi *nomos* što znači zakon [40]. Taksonomija je znanstvena disciplina koja na temelju sličnosti i razlika taksonomske jedinice, kategorizira i razvrstava u skupine. One su najčešće hijerarhijski strukturirane. Taksonomija je uža znanost od ontologije, koja u informacijskoj znanosti predstavlja model znanja koji opisuje područje, te se koristi za zaključivanje o entitetima u prostoru i relacijama između njih [40]. Najpoznatija taksonomija je švedskog znanstvenika Linnaeusa iz 18 st. koji taksonomiju koristi kod klasifikacije u biologiji i koja se koristi i dan danas [41].

Taksonomija, se po definiciji bavi određivanjem skupina na temelju zajedničkih obilježja i davanjem naziva tim skupinama. Svaka skupina dobiva kategoriju i skupine predmetne kategorije mogu se sakupiti kako bi tvorile nadskupinu više kategorije, te time stvaraju hijerarhijsku klasifikaciju. Ona je izvorno akademska disciplina grupiranja i organizacije bioloških organizama. Metoda grupiranja i organiziranja u hijerarhije uobičajeno se koristi u lingvistici s porodicama riječi, narječjima i naglascima.

Razvoj taksonomije je prilično složen zadatak, koji se obično obavlja brzopleto. Biologija je vrlo rano razradila taksonomiju živih bića. Tradicionalna Linnaeova taksonomija koja se može pronaći u biologiji, klasificira organizme na temelju predefinirane hijerarhije kategorija. Cilj izgradnje taksonomije je implementacija strukture na nestrukturirani svijet informacija.

Autori Morville i Rosenfeld u knjizi *Information Architecture for the World Wide Web: Designing Large-Scale Web Sites* [36] ilustriraju arhitekturu razvoja taksonomije pomoću tri elementa (Slika 2.) i to:

- poslovni kontekst,
- sadržaj i
- korisnici.

Slika 2. Tri temeljna elementa u razvoju taksonomije
Izvor: Morville, P., Rosenfeld, L.: *Information Architecture for the World Wide Web: Designing Large-Scale Web Sites*, 3nd edition, O'Reilly Media, 2006.

Ti elementi su temeljni za izgradnju većine taksonomija, ista slika pojavljuje se i kod autora Morrison-a u članku *How to create effective taxonomy* [40] koji predlaže tri temeljna elementa koji su bitnu za izgradnju:

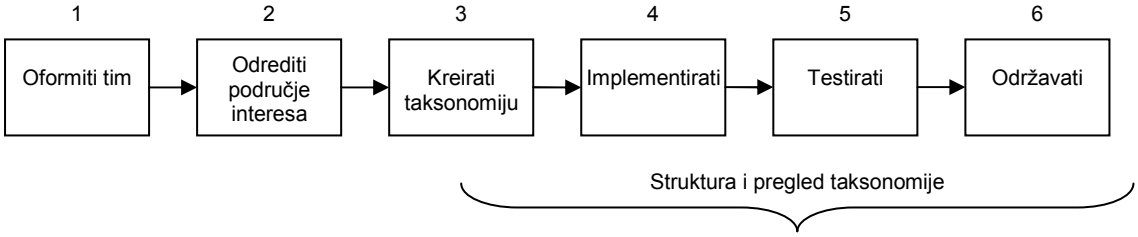
- **Poslovni kontekst** - odgovara na pitanja: Koja je svrha taksonomije? Gdje će se taksonomija koristiti? Koji su ulazni izvori za oblikovanje taksonomije?
- **Sadržaj** – odgovara na pitanja: Što je sadržaj taksonomije? Na temelju kojih izvora sadržaja će se taksonomija izgraditi?
- **Korisnici** – odgovara na pitanja: Tko će koristiti taksonomiju? Koji je profil korisnika koji će koristiti taksonomiju?

Na navedena pitanja može se odgovoriti u kontekstu izgradnje taksonomije anomalija. Pa s obzirom da **poslovni kontekst** može se reći da će se taksonomija anomalija koristiti kao koncept koji će omogućavati prepoznavanje i rješavanje anomalija uz pretpostavku da anomalije sličnih karakteristika imaju slične principe prepoznavanja i rješavanja.

Kad se govori o **sadržaju** taksonomije, tada se može reći da je cilj razviti taksonomiju koja bi se sastojala od pojedinih jediničnih anomalija prepoznatih u BPMN 2.0 modelima koje bi bile grupiranje po kategorijama koje će nastati na temelju nekih zajedničkih obilježja anomalija.

Korisnici koji bi koristili taksonomiju anomalija u početku bile osobe koje se bave modeliranjem poslovnih procesa, stoga taksonomija mora biti čitljiva i pregledna za korisnike. U kasnijoj fazi izgradnje automatiziranog koncepta prepoznavanja i rješavanja anomalija taksonomiju treba razraditi u oblik pogodan za ugradnju u alate za modeliranje.

Isti autor prikazuje i proces razvoja taksonomije, koja je prikazana na sljedećoj slici.

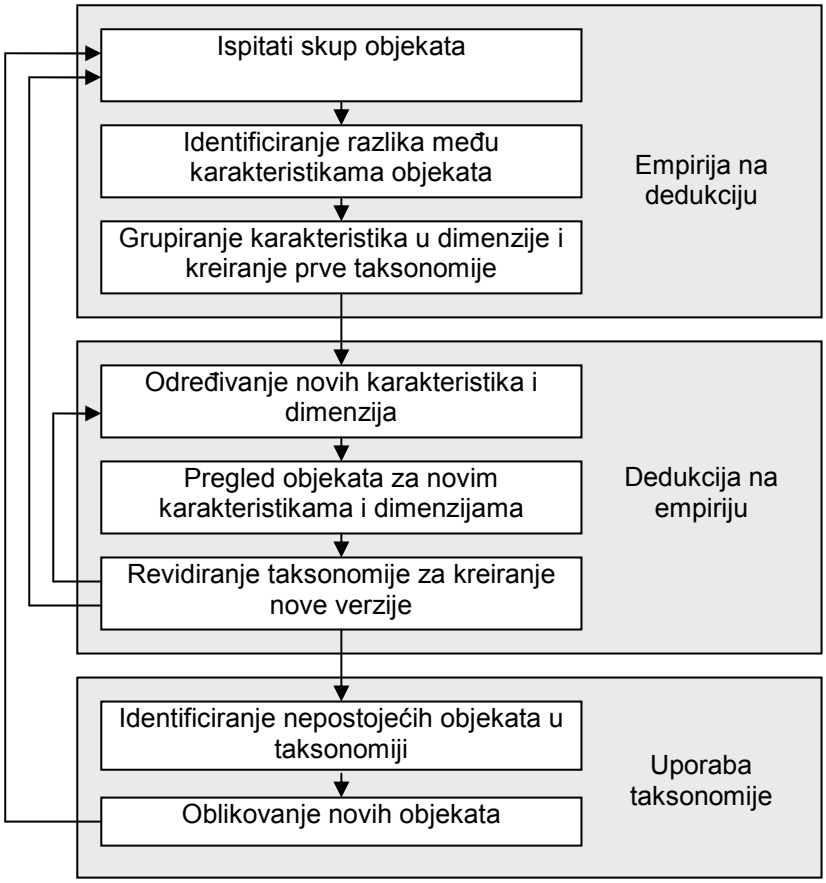


Slika 3. Proces razvoja taksonomije

Proces izgradnje taksonomije sastoji se od 6 temeljnih koraka [41]:

- **Oformiti tim** - krajnji rezultat svakog oblikovanja taksonomije uvelike može ovisiti o ekipi koja ju osmišljava.
- **Odrediti područje interesa** - odgovara na pitanja: Koja je svrha taksonomije? Gdje će se taksonomija koristiti? Koji su ulazni izvori za oblikovanje taksonomije?
- **Kreirati taksonomiju** – u skladu s prethodno definiranim koracima.
- **Implementirati** - implementacija se izvodi na najrazličitije načine ovisno o namjeni taksonomije. Implementacija je najčešće u obliku alata za upravljanje sadržaja.
- **Testirati** - cilj testiranja je identificirati greške i nepravilnosti. Testni rezultati se koriste za daljnje rafiniranje dizajna taksonomije.
- **Održavati** - s obzirom da je promjena sastavni dio svake organizacije, tako i promjene moraju biti predviđene u fazi izgradnje taksonomije.

Slika 4. prikazuje metodu razvoja taksonomije iz rada autora Nickerson, Varshney, Isaac: *Taxonomy development in information systems: developing a taxonomy of mobile application* [41]. Njihova metoda oslanja se na Bailey-ev pristup opisan u radu *A Three-Level Measurement Model. Quality and Quantity* [7]. Taj pristup ponavlja postupke "*empirija na dedukciju*" i "*dedukciju na empiriju*" tako dugo dok istraživač nije zadovoljan razvijenom taksonomijom. Ova metoda dozvoljava da se koristi i nakon što je taksonomija oblikovana.



Slika 4. Metoda razvoja taksonomije
(Izvor: Nickerson, R., Varshney, U., Isaac, H.: *Taxomomy development in information systems: developing a taxonomy of mobile applications*)

U ovom poglavlju opisani su temeljni pojmovi vezani uz metodologiju oblikovanja taksonomije. U kratkim crtama izložena je povijest razvoja pojma taksonomije, te je predstavljena metoda razvoja taksonomije koja će se koristiti u ovome radu.

2.2. Metode prepoznavanja i rješavanja anomalija u modelima poslovnih procesa

U ovome poglavlju opisati će se temeljni pojmovi i mehanizmi kojim će se prepoznati i riješiti anomalije u BPMN 2.0 notaciji. Dakle, odgovoriti će se na pitanja, što je to metamodel i kakva je njegova arhitektura, što je to MOF i u kakvom je on odnosu na metamodel. Također, ovo poglavlje će se se osvrnuti na OCL i pseudokod i u kakvom su odnosu ti jezici s metamodelom, te na koji način će nam svi ti mehanizmi pomoći u prepoznavanju i rješavanju anomalija u modelima poslovnog procesa.

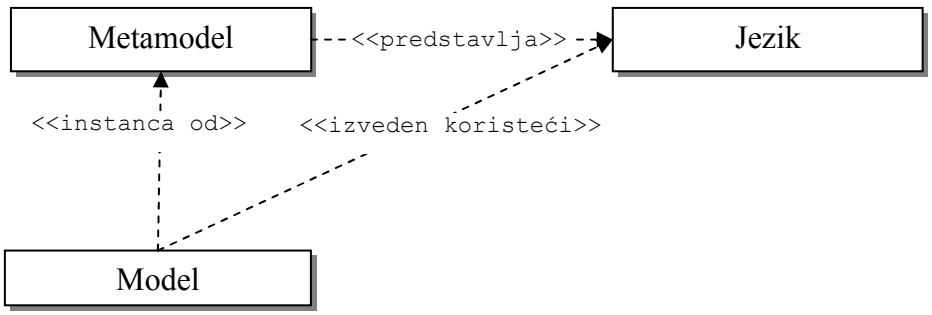
2.2.1. Metamodel

Temeljni alat pomoću kojega će se riješiti prepoznavanje i rješavanje anomalija je metamodel. Metamodeli olakšavaju pronalaženje svih potrebnih elemenata i atributa na koje treba obratiti pažnju prilikom izrade taksonomije koja se oslanja na karakteristike pojedinih anomalija. Metamodel je model modela i na njega je moguće primijeniti isti skup metoda koji se koriste za izgradnju ostalih modela. UML dijagram klasa ima navedene osobine, pa zapravo i on predstavlja osnovu za MOF (eng. *Meta-Object Facility*) standard za izradu metamodela. Dakle, MOF je standardni jezik za specifikaciju metamodela i sastoji od dva dijela i to:

- Apstraktni model generičkih objekata i njihovih asocijacija (vrlo slično UML-u).
- Skup pravila za preslikavanje MOF modela u jezično nezavisna sučelja.

Metamodel BPMN 2.0 je u ovome radu preuzet od OMG organizacije koja je i definirala BPMN 2.0 standard notacije i koji je izgrađen prema MOF standardu. S obzirom da je metamodel BPMN 2.0 notacije vrlo velik, tijekom rada će se naglasiti pojedini dijelovi metamodela, kako bi se lakše uočavali pojedini dijelovi bitni za istraživanje u ovome radu.

Prema Seidewitzu [54] metamodel je model koji opisuje jezik kojim se neki model modelira. Metamodel opisuje moguće modele koji mogu biti jezikom koji definira metamodel. Drugim riječima metamodel je instanca temeljena na opisu metamodela. Slika koja najbolje opisuje odnos modela, jezika i metamodela je slika autora Overbeak-a iz rada *Meta Object Facility (MOF) investigation of the state of the art* [48]:

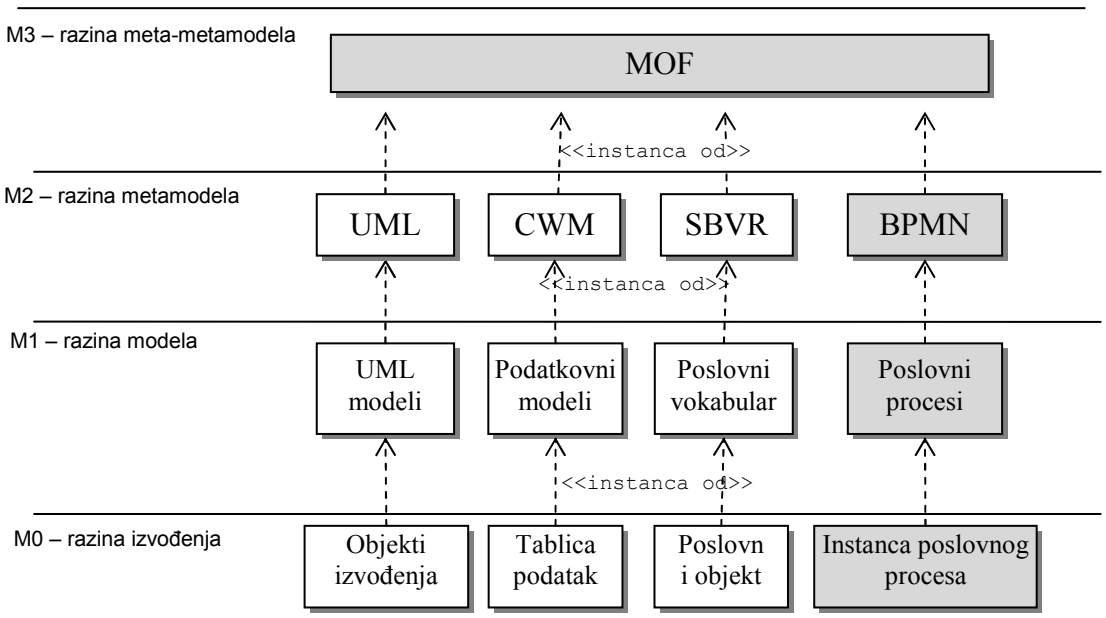


Slika 5. Odnos modela, jezika i metamodela
(Izvor: Overbeek, J.,F.: *Meta Object Facility (MOF) investigation of the state of the art*, Software Engineering, Electrical Engineering, Mathematics and Computer Science, University of Twente, 2006.)

Hijerarhija metamodela

OMG hijerarhija metamodela ima četiri sloja. Četiri-razinska arhitektura rafinira semantičke konstrukcije modela, zatim osigurava daljnje proširivanje UML metamodela i omogućava usporedbu UML metamodela s drugim standardima.

OMG hijerarhija metamodela ima četiri sloja i to: M3, M2, M1 i M0. Razine M3 i M2 su razine određene jezikom. **M3 razina** određuje druge metamodele, njezin primarni zadatak je definirati druge metamodele, dakle sadrži MOF. **M3 razina** daje arhitekturu metamodeliranja. Jedan meta-metamodel može definirati više različitih metamodela, dok više meta-metamodela može biti pridruženo svakom modelu. **M2 razina** je razina metamodela koji uvažava M3 razinu, dakle meta-metamodel, a opet s druge strane opisuje jezik modela. Ovaj sloj može sadržavati više metamodela. **Razina M1** je razina koji sadrži konkretne specifikacije podataka korisnika. **M0 razina** izvođenja sadrži objekte instancirane na temelju M1 razine modela koji se kreiraju tijekom izvođenja.



Slika 6. OMG hijerarhija metamodela

Iz navedenog se vidi da se pojedina instanca poslovnog procesa može opisati definiranim i opisanim poslovnim procesom, kojeg se može opisati BPMN 2.0 notacijom, dok je BPMN 2.0 notacija definirana M3 razinom u kojoj je MOF okvir pomoću kojeg se definira jezik metamodela BPMN 2.0 notacije.

Metamodel koji se koristi u ovome radu objavljen je od strane OMG organizacije i u njega su uključeni samo statični dijelovi BPMN 2.0 notacije, znači elementi, veze, kontejneri, dok elementi koji se bave dinamičkim aspektom BPMN 2.0 notacije (simulacije, vremenski aspekti, performanse, itd.) ovime metamodelom nije obuhvaćen. Stoga se za metamodel resursa BPMN 2.0 notacije koristi drugi metamodel, koji je isto pronađen na OMG stranicama.

2.2.2. MOF 2.0 standard

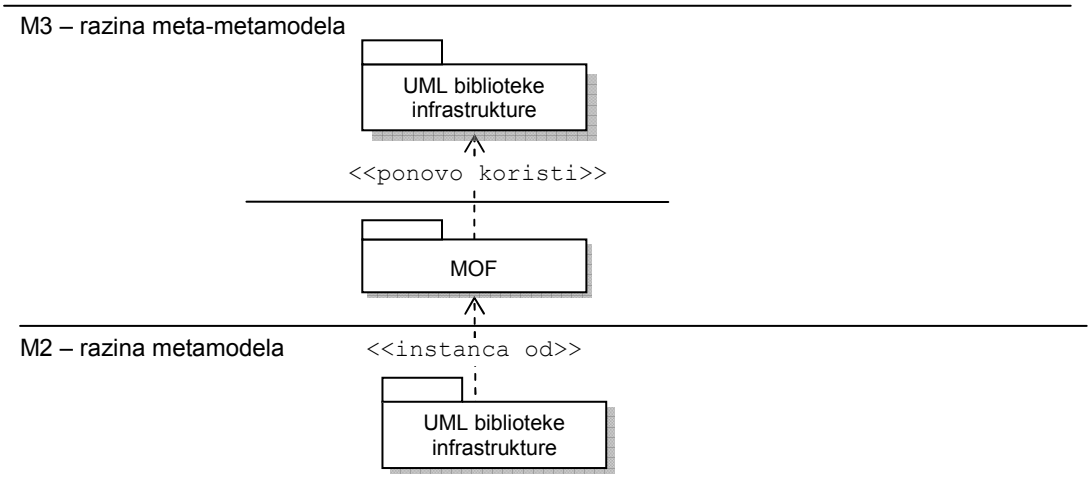
Kao što je vidljivo iz modela OMG četveroslojne hijerarhije metamodela, MOF definira na koji način će se opisati pojedini metamodel neke notacije, metode, modela i slično. Stoga će se u ovome poglavlju dati pregled osnovnih pojmova vezanih za MOF specifikaciju.

MOF 2.0 je specifikacija koja opisuje semantiku i sintaksu MOF-a. OMG organizacija osnovana je 1989. godine. Zahtjev za izradu i standardizaciju MOF-a je prvi puta iniciran 1996. godine. Opis što je MOF koji je u tom slučaju naveden glasi: *MOF je sučelje i sekvencijalna semantika potrebna za kreiranje, spremanje i upravljanje shemama objekata koji definira strukturu, značenje i ponašanje ostalih objekata u OMG arhitekturi upravljanja objektima.*

Negdje u isto vrijeme OMG je izdao koncepciju i pristup modeliranju. Taj koncept je započeo u isto vrijeme kad i zahtjev za UML. 2001. godine OMG lansira MDA (eng. *Model Driven Architecture*) kao novi pristup razvoju programskih proizvoda uporabom modela i tehnika modeliranja. Cilj MDA je postići visoku razinu apstrakcije koja je usmjerena na definiciju struktura i ponašanja sustava neovisno od programskih i sistemskih rješenja. Prema Milleru i Mukerji. MDA je prilagođena već definiranoj tehnologiji i omogućava pristup temeljen na modeliranju kao UML i MOF [36]. Zahvaljujući prilagodbi MOF-a od strane MDA, MOF je napravio značajan pomak s OMA (eng. *Object Management Architecture*) na MDA (eng. *Model Driven Architecture*). Navedeni pomak je zapravo promijenio ulogu MOF-a, jer MOF više nije okvir za upravljanje shemama objekata, već je postao meta metamodel koji se koristi za definiranje metamodela. 2004. godine arhitektura UML-a je dobila inačicu 2.0, iste godine izdan je MOF 2.0 sa ciljem da bolje naliježe uz UML 2.0.

Arhitektura MOF-a

Na Slici 7. vidi se međuzavisnost MOF-a i infrastrukturnih biblioteka po razinama meta-metamodela. Navedena slika temelj je za stvaranje slike uloge MOF-a i UML biblioteka u BPMN metamodelu.



Slika 7. Međuzavisnost MOF i UML infrastrukturnih biblioteka
(Izvor: Overbeek, J.,F.: *Meta Object Facility (MOF) investigation of the state of the art*, Software Engineering, Electrical Engineering, Mathematics and Computer Science, University of Twente, 2006.)

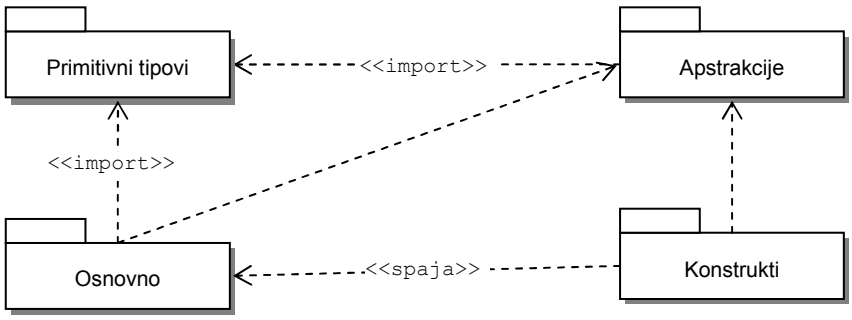
UML biblioteke infrastrukture

Svrha UML biblioteke infrastrukture je definiranje uobičajenih elemenata meta-jezika, koji se mogu koristiti za definiranje drugih metamodela. Prednost ponovne uporabe iste infrastrukture između više metamodela je u biti održavanje arhitektonski usklađenih metamodela. Arhitektonska usklađenost se postiže, jer i MOF i UML koriste isti jezik elemenata što osigurava da su obje arhitekture iste.

UML biblioteke infrastrukture se sastoje od dva temeljna paketa i to:

- Jezgre i
- Profila.

Paket jezgre - sastoji se od paketa primitivnih tipova, osnovnog paketa, konstrukta i apstrakcije [44] kao što prikazje Slika 8.



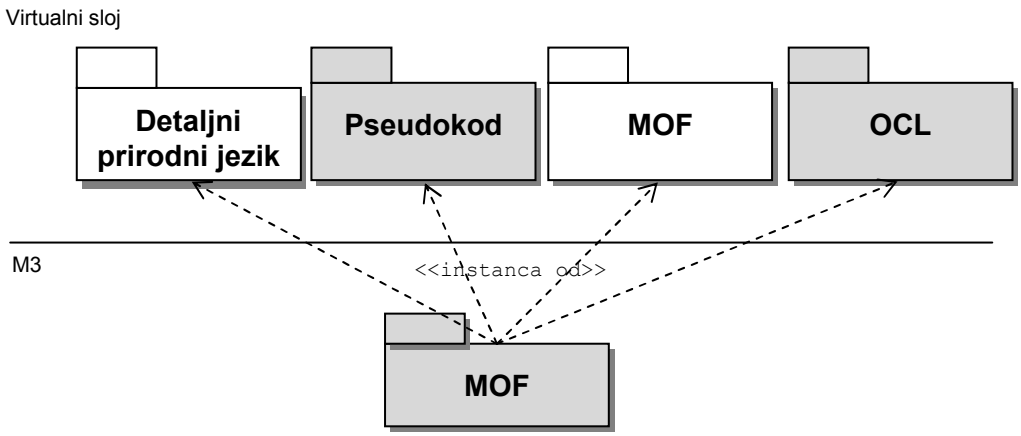
Slika 8. Paket jezgre sa međuzavisnostima
(Izvor: Overbeek, J.,F.: *Meta Object Facility (MOF) investigation of the state of the art*, Software Engineering, Electrical Engineering, Mathematics and Computer Science, University of Twente, 2006.)

- **Paket apstrakcija** – sadrži skup uobičajenih elemenata meta-jezika. Paket apstrakcija direktno koristi predefinirane tipove u paketu primitivnih tipova.
- **Osnovni paket** – osigurava minimalni set potreban za meta-jezik temeljen na klasama. Namjena mu je da u potpunosti bude višestruko korišten.
- **Konstrukti** – sadrži elemente meta-jezika koji se koriste u objektno orijentiranom modeliranju.
- **Primitivni tipovi** – sadrži uobičajen skup tipova (integer, real, string, itd.)

Paket profila – koristi se kao mehanizam koji omogućava prilagodbu postojećih modela određenim specifičnim platformama ili područjima primjene. Ovaj paket ovisi o konstruktima iz paketa jezgre i paketima proširenja iz MOF sloja. Paket profila koristi pristup koji određenim ograničenjima omogućava implementaciju pojedinih rješenja za razne primjene i razne platforme.

Kao što je prije navedeno u ovome radu koristit će se metamodel objavljen od strane OMG organizacije koji je usklađen s MOF standardom. Metamodel će prikazivati samo statičnu strukturu BPMN notacije. Plavi elementi imaju grafičku reprezentaciju u modelu.

Kao što je vidljivo po arhitekturi MOF-a, MOF sloj se nalazi na najvišoj razini. Pa prema tome u hijerarhiji modela ne postoji razina iznad M3 sloja. No, da bi se prikazala međuzavisnost MOF-a sa drugim tehnikama, koristi se virtualna razina koja sadrži sve ostale tehnike koje se mogu nadovezati na MOF s ciljem što detaljne izgradnje modela.

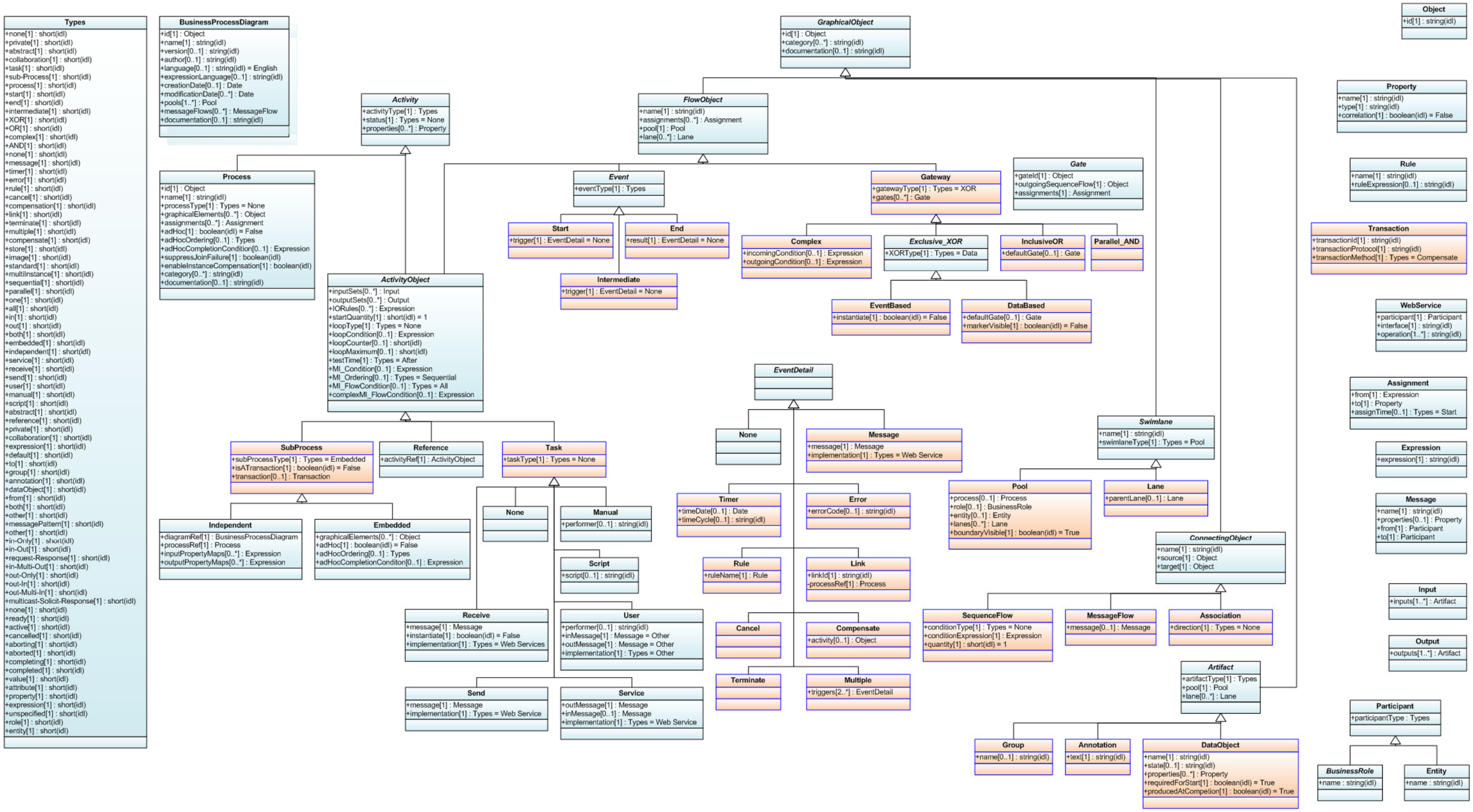


Slika 9. Virtualni sloj MOF specifikacije

Ovdje posebno treba istaknuti pseudokod i OCL koji će se u ovome radu upotrijebiti za opisivanje mehanizma pronalaženja i rješavanja anomalija u BPMN notaciji. OCL jezikom se opisuju izrazi kojim se definiraju uvjeti koji se moraju zadovoljiti da bi se neki sustav na pravilan način preslikao u model. OCL jezikom opisat će se prepoznavanje i rješavanje sintaktičkih anomalija.

Osim OCL jezika koristiti će se i pseudokod, kojim se daju smjernice prepoznavanja i rješavanja anomalija.

Na sljedećoj slici prikazan je metamodel BPMN 2.0 notacije koji je temelj za izgradnju rješenja za prepoznavanje i rješavanje anomalija. Na temelju tog metamodela izgraditi će se OCL pravila i pripadajući imenski prostor (eng. *namespace*) u procedurama pseudokoda za prepoznavanje i rješavanje anomalija u modelima poslovnih procesa.



Slika 10. Statična struktura elemenata i atributa BPMN metamodela (Izvor: OMG.org)

2.2.3. OCL

OCL (eng. *Object Constraint Language*) je jezik MDA arhitekture definiran od strane OMG organizacije [44]. OCL je jezik za definiranje ograničenja nad objektima. Također, omogućuje selektiranje djela modela ili računanje vrijednosti izraza na osnovu vrijednosti atributa razreda u modelu. U ovome radu OCL će poslužiti za rješavanje dva pitanja.

Prvo pitanje odnosi se na mogućnost OCL jezika da na temelju pravila postavljenih u metamodel BPMN 2.0 notacije moguće prepoznati određenu anomaliju.

Drugo pitanje odnosi se da je pomoću OCL jezika moguće BPMN 2.0 metamodel razraditi na takvu razinu da je tada vrlo lako moguće odrediti da li bilo koji BPMN 2.0 model procesa zadovoljava pravila BPMN 2.0 notacije.

OCL je dio UML standarda. Koji ima preciznu sintaksu i koji je jednostavan za čitanje. OCL je nastao, jer UML nije dovoljan za detaljni opis pojedinih programskih funkcija. OCL se koristi za određivanje detalja u objektima i operativne pretpostavke i uvjete u UML dijagramima klasa. Pri tome se koristi rječnikom UML dijagrama klasa, a atributi koriste elemente dijagrama klasa. S obzirom da će se OCL jezik koristiti intenzivnije za opise pravila prepoznavanja i rješavanja pojedinih anomalija modela poslovnih procesa, slijedi detaljniji opis OCL konstruktora, izraza, standardnih biblioteka i operacija. Te će se na kraju opisati temeljni nedostaci OCL jezika.

OCL konstruktori

- **`context`** – određuje kontekst OCL izraza.

```
context Account
```

- **`inv`** – postavlja uvjet koji mora biti ispunjen za sve instance navedenog konteksta.

```
context Account
inv: balance >= 0
```

- **`pre`** – postavlja uvjete koje je potrebno ispuniti prije nego se neka operacija počinje izvoditi.

```
context Account::deposit(amt: Integer): void
pre: amt > 0
```

- **post –** postavlja uvjete koje je potrebno zadovoljiti u trenutku kad operacija završava izvođenje.

```
context Account::deposit(amt: Integer): void
post: balance = balance@pre + amt
```

- **init** – određuje početnu vrijednost nekog atributa ili pridružuje ulogu.

```
context Account::balance: Integer
init: 0
```

- **derive** – određuje vrijednost atributa ili uloge nakon izvođenja.

```
context Account::interest: Real
derive: balance * .03
```

- **body** – određuje rezultat operacije upita.

```
context Account::getBalance(): Integer
body: balance
```

- **def** – definira novi atribut ili operaciju upita.

```
context Account
def: getBalance(): Integer = balance
```

- **package** – eksplicitno određuje kojem paketu pripadaju OCL izrazi.

```
package BankSystem::Accounting
context Account
      inv: balance >= 0
endpackage
```

OCL izrazi

- **self** – označava trenutnu instancu.

```
context Account
inv: self.balance >= 0
```

- **result** – označava rezultat operacije u izlaznim uvjetima.

```
context Account::getBalance(): Integer
post: result = balance
```

- **@pre** – u izlaznim uvjetima označava vrijednost određenog svojstva na početku operacije.

```
context Account::deposit(amt: Integer): void
post: balance = balance@pre + amt
```

- **`navigation`** – omogućuje navigaciju kroz atribute, povezane krajeve, povezane klase kvalifikacije.

```
context Account
inv: self.balance >= 0 -- dot notation
-- collection operator (->)
inv: owners->size() > 0
-- association class, TransInfo
inv: transactions.TransInfo->forall(amount > 0)
-- qualified association, owners
inv: not owners['primary'].isOclUndefined()
```

- **`if-then-else expression`** – uvjetni izraz sa uvjetom i dva izraza.

```
context Account::interestRate: Real
derive: if balance > 5000 then .03 else .02 endif
```

- **`let-in expression`** – izraz sa lokalnim varijablama.

```
context Account::canWithdraw(amt: Integer): boolean
def:  let newBalance: Integer = balance - amt
      in newBalance > minimumBalance
```

- **`Messaging (^)`** – pokazuje na početak razmjene poruka.

```
context Account::deposit(s: Sequence(Integer)): void
pre: s->forAll(amt: Integer | amt > 0)
post: balance = balance@pre + s->sum()
post: s->forAll(amt: Integer | self^deposit(amt))
```

OCL standardne biblioteke

Osnovni tipovi podataka

Tabela 1. Lista OCL osnovnih tipova podataka

Tipovi	Vrijednosti	Operacije
Boolean	false, true	or, and, xor, not, =, <>, implies
Integer	-10, 0, 10, …	=, <>, <, >, <=, >=, +, -, *, /, mod(), div(), abs(), max(), min(), round(), floor()
Real	-1.5, 3.14, …	
String	'Carmen'	=, <>, concat(), size(), toLower(), toUpper(), substring()

Izvor: Object Constraint Language, Version 2.3.1, Release Date: January 2012., <http://www.omg.org/spec/OCL/2.3.1/>, (1.9.2014.)

- **`OclAny`** - supertip svih UML i OCL tipova podataka

Tabela 2. Lista OCL operacija s tipovima podataka

Operacija	Opis
`=`	Istina ako su *self* instanca i argument isti
`<>`	Istina ako su *self* instanca i argument nisu isti
`oclIsNew()`	Istina ako je *self* instanca kreirana tijekom operacije
`oclIsUndefined()`	Istina ako je *self* instanca nedefinirana
`oclAsType(type)`	Instanca *self* ista kao i zadani tip
`oclIsTypeOf(type)`	Istina, ako je *self* instanca zadanog tipa
`oclIsKindOf(type)`	Istina ako instanca *self* odgovara zadanom tipu
`oclisInState(state)`	Istina ako je instanca self u određenom stanju
`T::allInstance()`	Postavlja sve instance tipa T.

Izvor: Object Constraint Language, Version 2.3.1, Release Date: January 2012., <http://www.omg.org/spec/OCL/2.3.1/>, (1.9.2014.)

- **`OclVoid`** – tip sa jednom jedninom instancom koji odgovara svim drugim tipovima.

Tabela 3. Lista OCL operacija

Operacija	Opis
`oclIsUndefined()`	Always true

Izvor: Object Constraint Language, Version 2.3.1, Release Date: January 2012., <http://www.omg.org/spec/OCL/2.3.1/>, (1.9.2014.)

- **`OclMessage`** – poruke koje se mogu slati i primati od strane objekata.

Tabela 4. Lista osnovnih OCL funkcija operacija

Operacija	Opis
`hasReturned()`	Da li je operacija pozvana i vrijednost vraćena.
`result()`	Rezultat operacije ili nedefinirano.
`isSignalSent()`	Da li self instanca šalje UML signal?
`isOperationCall()`	Da li je self instanca UML poziv operacije?

Izvor: Object Constraint Language, Version 2.3.1, Release Date: January 2012., <http://www.omg.org/spec/OCL/2.3.1/>, (1.9.2014.)

- **`Tuple`** – sastoji se od imenovanih dijelova od kojih svaki može biti različiti tip.

```
-- Tuple(name: String, age: Integer)
Tuple {name: String = 'John, age: Integer = 20}
```

Tipovi kolekcija

Podržava četiri tipa kolekcija (Set, OrderedSet, Bag, i Sequence) sa **Collection** kao supertipom.

Ograničenja kolekcija

```
Set {1, 2, 3} -- Set(Integer)
OrderedSet {„apple, „pear?, „orange?} -- OrderedSet(String)
Bag {1, 1, 2, 2} -- Bag(Integer)
Sequence {1..(4 + 6), 15} - Sequence(Integer)
```

Standardne operacije

Tabela 5. Lista OCL standardnih operacija

Operacija	Opis
`count(o)`	Broj ponavljanja elementa O u kolekciji instance self?
`execludes(o)`	Da li o nije element kolekcije?
`excludesAll(c)`	Da li svi elementi iz c nisu prisutni u kolekciji?
`includes(o)`	Da li je o element kolekcije?
`includesAll(c)`	Da li su svi elementi c sadržani u kolekciji?
`isEmpty()`	Da li kolekcija e sadrži elemente?
`notEmpty()`	Da li kolekcija sadrži elemente?
`size()`	Broj elemenata u kolekciji
`sum()`	Suma svih elemenata u kolekciji

Izvor: Object Constraint Language, Version 2.3.1, Release Date: January 2012., <http://www.omg.org/spec/OCL/2.3.1/>, (1.9.2014.)

Iteracije operacija

Tabela 6. Lista OCL iteracija operacija

Operacija	Opis
any(expr)	Vraća sve elemente za koje vrijedi za je izraz expr istinit.
collect(expr)	Vraća kolekciju koja je rezultat provjere izraza expr za svaki element u instanci *self*.
collectNested(expr)	Vraća kolekciju koja je rezultat izraza expr za svaki element u instanci *self*.
exists(expr)	Ima najmanje jedan element za koje vrijedi izraz expr.
forAll(expr)	Da li je izraz istinit za sve elemente?
isUnique(expr)	Da li izraz expr ima jedinstvenu vrijednost za sve elemente?
iterate(x: S; y: T| expr)	Iteracija po svim elementima
one(expr)	Da li postoji samo jedan element za koji vrijedi da je expr istinit?
reject(expr)	Vraća kolekciju koji sadrži sve elemente z koje ne vrijedi izraz expr.
select(expr)	Vraća kolekciju koja sadrži sve elemente z koje vrijedi da je izraz expr istinit.
sortedBy(expr)	Vraća kolekciju sortiranu po izrazu expr.

Izvor: Object Constraint Language, Version 2.3.1, Release Date: January 2012., <http://www.omg.org/spec/OCL/2.3.1/>, (1.9.2014.)

Nedostaci OCL jezika

OCL jezik ima tri obličja. To je jezik izraza, jezik modeliranja i formalni jezik.

To što je jezik izraza znači da OCL izrazi nikad ne mijenjaju model, nego ga samo nadograđuju. Dakle ni jedan OCL izraz ne mijenja vrijednost bilo kojeg atributa objekta. Sve vrijednosti u modelu zadržavaju svoju vrijednost i nikad se ne mijenjaju. OCL izrazi samo ocjenjuju vrijednosti pojedinih atributa i upravo to daje novu vrijednost modelu.

Kao jezik modeliranja OCL nije programski jezik, sa njime nije moguće napisati programsku logiku ili kontrolu toka. OCL jezikom nije moguće pozvati neke druge procese ili aktivirati neke operacije drugih procesa. Pitanje implementacije je izvan područja koje OCL pokriva i ne može biti izraženo u OCL jeziku.

Kao formalni jezik u OCL-u svaki izraz mora imati formalno značenje. Svi OCL izrazi su atomarni. Nadalje OCL ipak nije jednostavan za čitanje, koliko got je oblikovan da bude što lakši za čitanje korisnicima bez programskog znanja, no opet nije zahtjevan kao drugi formalni jezici (VDM, Z,.. itd.). OCL jezik je strog što se tiče tipova, u OCL-u nije moguće usporediti Integer i String.

Iz svega navedenog proizlazi da se OCL ne može koristiti u baš svakoj situaciji koja omogućava pronalaženje i rješavanje svih vrsta anomalija. Da bi mogli dodatno proširiti mehanizme prepoznavanja i rješavanja anomalija u BPMN 2.0 modelima poslovnih procesa, osim OCL jezika koristi će se **pseudokod**.

2.2.4. Pseudokod

Pseudokod služi za opisivanje operativnih principa algoritama ili računalnog programa na visokoj razini. Koristi strukturne konvencije programskih jezika, no prvenstvena mu je namjena ljudsko čitanje. Pseudokod je namijenjen programerima, dizajnerima i arhitektima sustava da se detaljno, korak po korak izraze i opišu pojedinu funkcionalnost koju pokušavaju riješiti. Pseudokod može biti različito detaljan, ovisno o tome što opisuje i kome je namijenjen. Recimo, ako je namijenjen programeru, onda će biti više detaljan nego ako je namijenjen projektantu informacijskog sustava kojem je u pojedinim slučajevima dovoljan okvir ili nacrt pojedinog rješenja. Smisao pseudokoda je olakšati ljudima razumijevanje pojedinog programskog rješenja s ciljem učinkovitijeg opisa određenog mehanizma u algoritmu. Prema tome prednost uporabe pseudokoda je da omogućava programeru da se koncentrira na algoritam bez potrebe da se fokusira na sintaktičke detalje. Pseudokod u biti

predstavlja model programskog koda, koji služi kao smjernica kako bi se pojedini programski mehanizmi trebali riješiti. Pseudokod nije programski jezik već način opisivanja programa ili algoritma. Ne zahtjeva strogu sintaksu, već služi kao reprezentacija programskih funkcionalnosti. Pseudokod je neformalni jezik koji se koristi za stvaranje konture, odnosno nacrta izvršnog programa. Ne postoje strogo definirani standardi sintakse pseudokoda, jer se pseudokod ne kompajlira u izvršnu datoteku. S obzirom da razni programski jezici imaju različitu sintaksu i strukturu razumijevanje različitih jezika može biti problematično, pseudokod je doskočio tom problemu na način da problem uporabe uobičajene sintakse zamjenjuje s temeljnim frazama jezika komunikacije kome je pseudokod namijenjen. Sa opisivanjem programskih funkcija i algoritama pseudokodom, programeri koji barataju različitim programskim jezicima mogu razumjeti opisanu programsku logiku. Dakle, pseudokod ne sadrži detalje kao što su deklaracije varijabli, sistemske detalje ili detalje pojedinih rutina.

Pravila pisanja pseudokoda

U ovom poglavlju definirana su opće poznata pravila pisanja pseudokoda, jer će neke od procedura za otkrivanje i rješavanje anomalija biti će opisani pseudokodom. U pojedinim procedurama pseudokod će biti precizniji i prilagođeniji suvremenim programerskim alatima upravo da se posebno naglasi pojedini detalj bitan za funkcioniranje pojedine procedure. Na temelju navedenog mogu se definirati sljedeća pravila:

- **Pisati samo jedan izraz po liniji pseudokoda –** svaki izraz u pseudokodu mora definirati jednu aktivnost računala. Ako je pseudokod pravilno napisan, tada svaka linija koda predstavlja jedan zadatak koji računalo mora obaviti.
- **Početnu ključnu riječ treba pisati velikim slovima** – na primjer: `READ, WRITE, IF, ELSE, ENDIF, WHILE, ENDWHILE, REPRAT, UNTIL.`
- **Za prikazivanje hijerarhije treba koristiti uvlačenje linija koda –**
    - Linije naredbi koje se odvijaju jedna za drugom, dakle u sekvenci trebaju početi na istoj poziciji.
    - Naredbe koje se odvijaju unutar strukture odlučivanja moraju biti uvučene.
    - Naredbe koje se odvijaju unutar petlje moraju biti uvučene.
- **Strukture od više linija moraju imati zatvorenu strukturu sa „END“ izrazom** – npr. `IF/ELSE/ENDIF`, itd.

- **Svaki izraz treba biti pisan u neutralnom jeziku** – treba se oduprijeti izazovu da se pseudokod piše u jeziku koji je najpoznatiji nekom autoru, jer dugoročno to štedi vrijeme. To posebno dolazi do izražaja u slučajevima gdje neke specifičnosti pojedinog jezika mogu sakriti neki detalj u razradi algoritma u pseudokodu.
- **Matematičke operacije** koje je moguće koristiti u pseudokodu prikazane su u Tabeli 6.
- **Poredak izvršavanja** u pseudokodu prikazan je u Tabeli 7. Operacije jednakih prioriteta biti će izvedene od lijepa prema desno.
- **Operacije uvjeta** koje se koriste u pseudokodu prikazane su u Tabeli 8.
- **Operacije i njihove suprotnosti** u pseudokodu prikazane su u Tabeli 9.
- **Logički operatori** koji se koriste u pseudokodu prikazane su u Tabeli 10.

Tabela 7. Matematičke operacije u pseudokodu

Operacija	Opis
`()`	Grupiranje
`**`	Eksponent
`*`	Množenje
`/`	Dijeljenje
`+`	Zbrajanje
`-`	Oduzimanje

Tabela 8. Tabela poretka izvršenja operacija

Operacija	Opis
`()`	Izrazi u zagradama
`**`	Potenciranje
`/*`	Dijeljenje i množenje
`+-`	Zbrajanje i oduzimanje

Tabela 9. Operacije uvjeta pseudokoda

Operacija	Opis
>	Veći od
<	Manji od
>=	Veći ili jednak
<=	Manji ili jednak nego
=	Jednak
<>	različit

Tabela 10. Lista operacija i suprotnosti tih operacija u pseudokodu

Operacija	Suprotnost
>	<=
<	>=
=	<>

Tabela 11. Lista logičkih operatora u pseudokodu

Operacija	Suprotnost
AND	Ako je bilo koji uvjet laž, cijeli izraz je laž
OR	Ako je bilo koji uvjet istina, cijel izraz je istina.
NOT	Inverzira izraz, istina postaje laž, laž postaje istina.

Prednosti i nedostaci pseudokoda

Prednosti uporabe pseudokoda su:

- Može se napisati u bilo kojem tekst procesoru.
- Može se jednostavno preoblikovati u dijagram toka.
- Vrlo je koristan kod strukturiranih dizajnerskih elemenata.
- Lako se piše.
- Razrješava algoritme.
- Lako se čita i razumljiva je.
- Pretvaranje pseudokoda u programski jezik je jednostavnije u usporedbi prema pretvorbi dijagrama toka u programski jezik.
- Prisiljava da se disciplinirano vodi dokumentacija dizajna pojedinog rješenja.
- Osigurava dodatnu razinu provjere rješenja.
- Pomaže u uočavanju grašaka prije nego se rješenje pretvori u programski kod.
- Povećava pouzdanost programskog proizvoda.
- Smanjuje ukupne troškove rješenja.

Navedimo i neke nedostatke pseudojezika:

- Pseudokod nije grafički jezik, pa nema vizualizacije.
- Krajnji rezultat realiziran u pseudokodu nije slika.
- Ne postoji standardiziran stil ili format, pa se svaki pseudokod može razlikovati jedan od drugog.
- Početnici teško mogu pratiti logiku pseudokoda u usporedbu s dijagramom toka.
- Kreira jedan dodatni sloj dokumentacije koju treba održavati.
- Može dovesti do pogrešaka kod pretvaranja pseudokoda u programski kod.
- Kod kompliciranih projekata može zahtijevati posebni alat za izdvajanje pseudokoda i olakšavanje dijagrama toka.

Imenski prostor

Pseudokod će se kao i OCL naslanjati na metamodel koji će nam pomoći definirati imenski prostor (eng. *Namespace*). U OCL-u i pseudokodu imenski prostor omogućava identifikatoru definiciju konteksta. Na taj način u OCL-u i pseudokodu moguće je raditi s klasama i atributima istog naziva, ali u različitom kontekstu. Imenski prostor dio je objektnog pristupa programiranju (eng. OOP – O*bject Oriented Programming*) koji omogućava da klase i atributi istog nazivlja mogu koegzistirati u istoj proceduri u različitom kontekstu. Na taj način izbjegavaju se problemi s jedinstvenošću nazivlja. Mehanizam imenskog prostora omogućava grupiranje atributa i klasa koje pripadaju istom kontekstu. Pomoću imenskog prostora moguće je odrediti i hijerarhiju pojedinih programskih struktura. Pojedine razine klasa i atributa odvajaju se posebnim znakovima kao što su točka (.), kosa crta (\), donja vodoravna crta (_), dvije dvotočke (::).

Uzmimo primjer atributa **`name`** u metamodelu BPMN 2.0 notacije, iz modela je vidljivo, da više klasa ima atribut **`name`** i to istog tipa **`string`**. Da bi mogli koristiti taj atribut u nekoj proceduri, mora se ispred naziva atributa **`name`** dodati kontekst za koju klasu se taj atribut odnosi. Dakle **`Process.name`** ako se atribut odnosi na klasu **`Process`** ili **`FlowObject.name`** ako se atribut odnosi na klasu **`FlowObject`**. Osim ovog načina definiranja konteksta koji dozvoljava da za svaki atribut postoji definicija atributa, moguće je kontekst definirati i za cijelu proceduru, u tom slučaju na početku procedure navodi se kontekst, npr. `Context Process` ili **`Context FlowObject`**.

Problem koji se javlja uporabom imenskog prostora je povećanje količine znakova u programskom kodu što dovodi do većeg zauzeća memorije i u krajnjem slučaju sporijeg izvršenja programa.

U prethodnim poglavljima prikazani su temeljni pojmovi i mehanizmi kojim će se prepoznati i riješiti anomalije u BPMN 2.0 notaciji. Dakle, opisan je metamodel, njegova arhitektura, MOF, OCL, pseudokod. Navedeni mehanizmi pomoći se u prepoznavanju i rješavanju anomalija u modelima poslovnih procesa.

3. TAKSONOMIJA ANOMALIJA U MODELIMA POSLOVNIH PROCESA

Već u poglavlju 1.3. navedeni su neki radovi koji se bave anomalijama i njihovom kategorizacijom. U ovome poglavlju detaljnije su prikazane postojeće taksonomije autora, s ciljem analize i sinteze anomalija i izgradnje nove taksonomije anomalija koja bi obuhvatila sve kategorije anomalija. Autori Solti-Rogge, Kunze, Awad i Weske u svojem radu *Business process configuration wizard and consistency checker for BPMN 2.0.* [56] kategoriziraju anomalije na tri temeljne kategorije:

- nedostatak podataka,
- redundantni podaci i
- konfliktni podaci.

Autori navode da anomalije prouzročene nedostatkom podataka nastaju kada postoji vjerojatnost da se neka aktivnost izvede kad čita podatke koji nisu nikad bili zapisani. Dok anomalije nastale redundantnim podacima nastaju kada su podaci objekata zapisani, no nikad nisu pročitani od strane neke aktivnosti. Anomalije nastale zbog konflikata nastaju kad su podaci zapisani od strane dvije konkurentne aktivnosti. Iz ovoga se može zaključiti da se anomalije, koje autori navode, bave isključivo problemima podataka, dok u obzir ne uzimaju probleme tokova i probleme koji nastaju zbog loše uporabe elemenata modela. Autori Awad, Decker i Lohmann u svojem radu *Diagnosing and Repairing Data Anomalies in Process Models* [5] dijele anomalije na:

- **Anomalije nastale zbog previše ograničavajućih preduvjeta** – nastaju kad aktivnost u modelu ima preduvjet na nekim podatkom čije stanje nije ispunjeno.
- **Anomalije nastale implicitnim usmjeravanjem** – predstavljaju poseban slučaj anomalije nastale previše ograničavajućim preduvjetom, gdje nedostaje neki podatak iz preduvjeta nužan za neku aktivnost, a koji je nastao zbog nepravilnog odabira putanje.
- **Anomalije nastale implicitnim ograničenjem kod izvršenja** - događaju se kad postoje dvije konkurentne aktivnosti koje dijele neki preduvjet, a anomalija nastaje kad jedna aktivnost promijeni stanje podatka preduvjeta.

Autori Kim, Lee i Soon u svojem radu *Classification and analyses of business process anomalies* [29] navode dvije temeljne kategorije anomalija u poslovnim procesima, i to:

- sintaktičke i
- strukturne anomalije.

Sintaktične anomalije odnose se na pogrešnu uporabu koncepata i pravila modeliranja. **Strukturne anomalije** nastaju kad poslovni procesi ne kontroliraju interakciju tokena zbog krive uporabe vrata. Autori navode da se većina alata za verifikaciju modela bavi isključivo I i ILI vratima.

Autori Dohring i Heublein u radu *Anomalies in Rule Adopted workflows – A Taxonomy and solutions for vBPMN* [16] navode tri temeljne kategorije anomalija.

- **Anomalije kontrole toka** (eng. *Control flow anomalies*),
    - Redundancija kontrola toka (eng. *Control flow redundancy*)
        - Nepotrebna kontrola toka (eng. *unnecessary control flow*) – opisuje problem višestruke semantički iste veze između dva objekta toka (te veze ne dodaju novu vrijednost toku, već su redundantne).
        - Redundancija modela (eng. *modeling redundancy*).
    - Vrata (eng. *Gateway*)
        - Nedefinirani uvjeti vrata.
        - Kontradiktorni uvjeti vrata (eng. *contradictive gateway conditions*).
    - Nedostatak sinkronizacije (eng. *Lack of synchronization*)
        - Zastoj toka (eng. *Flow deadlock*)
        - Zapinjanje toka (eng. *Flow livelock*)
    - Nedostatak tolerancije petlji (eng. *missing loop tolerance*) – nastaje kad podaci potrebni za aktivnosti u petlji.
    - Nekonzistentna promjena prošlosti (eng. *inconsistent change of past*) – nastaje kad određena aktivnost daje podatke koje utječu na aktivnosti u budućnosti a koji uzrokuju nekonzistentnost.
- **Anomalije temeljene na pravilima** (eng. *Rule based anomalies*) i
- **Anomalije toka podataka** (eng. *Data flow anomalies*).

S obzirom da se sve četiri klasifikacije u pojedinim detaljima preklapaju, može se pokušati objediniti sve četiri klasifikacije u jedinstvenu taksonomiju. Ako se sve četiri kvalifikacije uzmu u obzir, tada se dobiva sljedeća tablica u kojoj su objedinjene anomalije koje se mogu pojaviti u modelima poslovnih procesa.

Tabela 12. Klasifikacija anomalija

Anomalije	Sintaktičke anomalije	Nepravilna uporaba objekata toka			***1. Više početnih događaja u modelu***
					2. Nepovezanost početnog događaja s ostalim elementima modela
					3. Početni događaj ima više izlaznih sekvenci
					4. Nepostojanje završnog događaja
					5. Nepovezanost završnog događaja s ostalim elementima modela
					6. Završni događaj ima više ulaznih sekvenci
					7. Nepostojanje početnog događaja
		Nepravilna uporaba veza			***8. Ulazna sekvenca s početnim događajem***
					9. Izlazna sekvenca s završnim događajem
		Nepravilna uporaba toka poruka			***10. Početni događaj poruke vezan je tokom poruke s elementom početnog događaja***
					11. Početni događaj poruke vezan je tokom poruke s elementom baze
					12. Početni događaj poruke vezan je tokom poruke s elementom potprocesa
					13. Početni događaj poruke vezan je tokom poruke s elementom aktivnosti
					14. Početni događaj vezan je tokom poruke s elementom stop događaja poruke
					15. Početni događaj vezan je tokom poruke s elementom završnog događaja poruke
					16. Početni događaj poruke vezan je tokom poruke s završnim događajem
					17. Bazen je vezan tokom poruke s završnim događajem
					18. Potproces je vezan tokom poruke s završnim događajem
					19. Aktivnost je vezana tokom poruke s završnim događajem
					20. Stop događaj poruke je vezan tokom poruke s završnim događajem
					21. Element završnog događaja poruke vezan je tokom poruke s elementom završnog događaja
		Nepravilna uporaba spremnika			***22. Nepravilna uporaba bazena***
					23. Bazen bez elemenata
					24. Nepravilna uporaba staza
					25. Staza bez elemenata
	Strukturne anomalije	Kontrola toka	Redundancija kontrole toka		***26. Nepotrebna kontrola toka***
					27. Redundancija modela
			Nedefinirani uvjeti vrata		***28. Nedefinirani uvjeti vrata***
					29. Kontradiktorni uvjeti vrata
					30. Deterministički nedostatak sinkronizacije
					31. Nedeterministički nedostatak sinkronizacije
			Specijalni slučajevi toka		***32. Viseća stanja***
					33. Netolerancija na promjene poretka
					34. Nekonzistentnost kod inserta paralelnih grana
					35. Nekonzistentnost promjene iz prošlosti
					36. Nedostatak tolerancije u petlji
			Narušavanje ispravnosti		***37. Zapinjanje toka***
					38. Zastoj toka
		Anomalije pravila	Konzistentnost	Redundancija pravila	***39. Redundantna pravila***
					40. Generalna pravila
				Konfliktna pravila	***41. Izravna kontradikcija***
					42. Kontradikcija u ulazu
					43. Kontradikcija u zaključku
					44. Zapinjanje temeljeno na pravilima
					45. Zastoj temeljen na pravilima
			Obuhvat pravila		***46. Izolirana pravila***
					47. Beskorisni zaključci (Beskorisna pravila)
					48. Viseći uvjeti
					49. Nepostojeća pravila
			Tok podataka		***50. Redundantni podaci***
					51. Izgubljeni podaci
					52. Nepostojeći podaci
					53. Nekonzistentni podaci
					54. Nikad izbrisani podaci
					55. Dva puta izbrisani podaci
					56. Podaci neizbrisani na vrijeme
	Anomalije resursa	Anomalije obrazaca kreiranja			***57. Direktna alokacija u slučaju da više resursa ima isti naziv***
					58. Direktna alokacija resursa bez mehanizma eskalacije
					59. Alokacija temeljena na ulogama kad odabrani resurs nije pripadnik određene uloge/role
					60. Alokacija temeljena na ulogama kad resurs pripada u više od jedne grupe
					61. Alokacija temeljena na ulogama kod grupe sa jednim resursom
					62. Automatsko izvođenje bez mehanizma provjere izvođenja
		Anomalije samopokretajućih obrazaca			***63. Početak kreiranja sa alokacijom resursa***
					64. Djelomično lančano izvršenje u poslovnom procesu
		Anomalije višestrukih obrazaca resursa			***65. Istovremeno izvođenje kod ograničenih kapaciteta resursa***
	Semantičke anomalije	Nepravilno nazivlje elemenata			***66. Nekonzistentnost nazivlja elemenata***
					67. Sinonimi u nazivlju modela
					68. Homonimi u nazivlju elemenata
					69. Nejedinstvenost nazivlja elemenata
					70. Neimenovanje elemenata
		Nepravilne vrijednosti atributa			***71. Ne postoji vrijednost atributa***
					72. Nejedinstvenost vrijednosti atributa
					73. Neispravan tip podataka vrijednosti atributa
					74. Nedozvoljeni znakovi u vrijednosti atributa
	Anomalije prikaza modela	Anomalije organizacije percepcije	Anomalije jednostavnosti		***75. Previše elemenata u modelu***
					76. Neselektivni prikaz elemenata u modelu
			Anomalije sličnosti		***77. Slični elementi nisu grupirani zajedno***
					78. Slični elementi nemaju iste boje
					79. Slični elementi nemaju istu veličinu
					80. Slični elementi nemaju isti oblik
			Anomalija kontinuiranosti		***81. Križanje i savijanje veza između elemenata***
			Anomalije grupiranja		***82. Nekompaktni i razvučeni model***
			Anomalije povezanosti		***83. Preklapanje elemenata modela***
			Anomalije prisnosti		***84. Nepoštovanje pozicije hijerarhije elemenata***
		Anomalije percepcije razlikovanja	Anomalija simetrije		***85. Necentrirani i nesimetrični model***
			Anomalija orijentacije		***86. Modeli orijentirani s desna na lijevo***
					87. Modeli orijentirani od dolje prema gore
					88. Vertikalni nazivi elemenata
					89. Vertikalni opisi unutar modela
			Anomalija obrisa (konture)		***90.Elementi modela se međusobno dodiruju***

Sukladno prethodnoj tablici, anomalije u modelima poslovnih procesa mogu se podijeliti u pet temeljnih kategorija:

- sintaktičke anomalije,
- strukturne anomalije,
- anomalije resursa,
- semantičke anomalije i
- anomalije prikaza modela.

Svaka od navedenih anomalija imaju svoje specifične uzroke i načine rješavanja. Osim toga pojedini oblici anomalija učestaliji su u određenim notacijama.

Sintaktičke anomalije su anomalije koje u modelima nastaju zbog nepravilne uporabe grafičkih elemenata (simbola i notacije). Ove anomalije usko su vezane uz grafičku notaciju funkcionalnih modela. U ovoj klasifikaciji prikazanoj na tabeli taksonomije anomalija sintaktičke anomalije se odnose na BPMN 2.0 notaciju. Za UML dijagram aktivnosti struktura sintaktičkih anomalija bi bila drugačija. Rješavanje sintaktičnih anomalija usko su vezane uz pravila koja se odnose na notaciju.

Strukturne anomalije su strukturno-logičkog karaktera te su moguće i u drugim oblicima modela kao što su: pseudokodovi, modeli case alata, programski modeli, itd. Rješavanje ovakvih vrsta anomalija zahtjeva uporabu složenih metoda prepoznavanja mogućih problema prije nego do problema u stvari dođe ili ugradnju preventivnih mehanizama koji ne dozvoljavaju da model procesa bude u bezizlaznom stanju.

Anomalije resursa u modelima poslovnih procesa nastaju kod pogrešno definiranog resursa za pojedinu aktivnost, krive alokacije ili pogrešne procjene zahtjevnosti određenog radnog koraka. BPMN 2.0 notacija ima vrlo slabu podršku resursa, pa su iz obrazaca poslovnih procesa izvučene samo one anomalije koje su moguće u navedenoj notaciji.

Semantičke anomalije nastaju u slučaju nepravilno dodijeljenog nazivlja ili nepravilne dodijeljene vrijednosti atributa. Neimenovanje pojedinih objekata isto su uzrok pojave semantičkih anomalija. Semantičke anomalije nastaju i kod višestrukog ponavljanja nazivlja pojedinih objekata ili atributa.

Anomalije prikaza modela su anomalije zbog kojih modeli postaju nepregledni, nečitljivi i vizualno kompleksni. Modeli gube svoju primarnu funkciju pojednostavljivanja promatranog poslovnog procesa.

U nastavku ovog poglavlja biti će detaljnije razrađena i objašnjena svaka od navedenih skupina anomalija.

3.1. Sintaktičke anomalije

Nepravilna uporaba grafičkih elemenata u modelu predstavljaju sintaktičku anomaliju. One se mogu podijeliti na tri temeljne kategorije koje se odnose na vrstu elemenata od kojih su modeli izgrađeni, a to su [29]:

- nepravilna uporaba objekata toka,
- nepravilna uporaba veza i
- nepravilna uporaba kontejnera.

3.1.1. Nepravilna uporaba objekata toka

Kod nepravilne uporabe objekta toka anomalije nastaju najčešće zbog nepravilne uporabe [29]:

- događaja,
- radnih koraka i
- vrata.

Nepravilna uporaba događaja moguća je u tri slučaja kod:

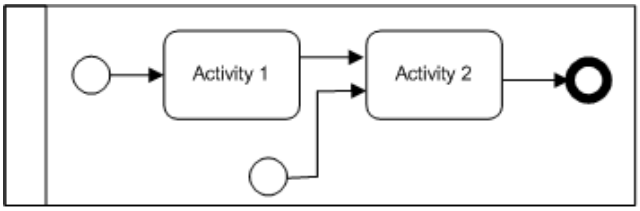
- nepravilne uporabe početnog događaja,
- nepravilne uporabe završnog događaja i
- nepravilne uporabe međudogađaja.

Početni događaj je opcionalan objekt u modelima poslovnih procesa, ako u modelu postoji početni događaj onda mora postojati i završni događaj. Najčešći uzroci anomalija kod nepravilne uporabe početnog događaja su:

- više početnih događaja u modelu,
- nepovezanost početnog događaja s ostalim elementima modela,
- početni događaj ima više od jedne izlazne sekvence i
- nepostojanje završnog događaja.

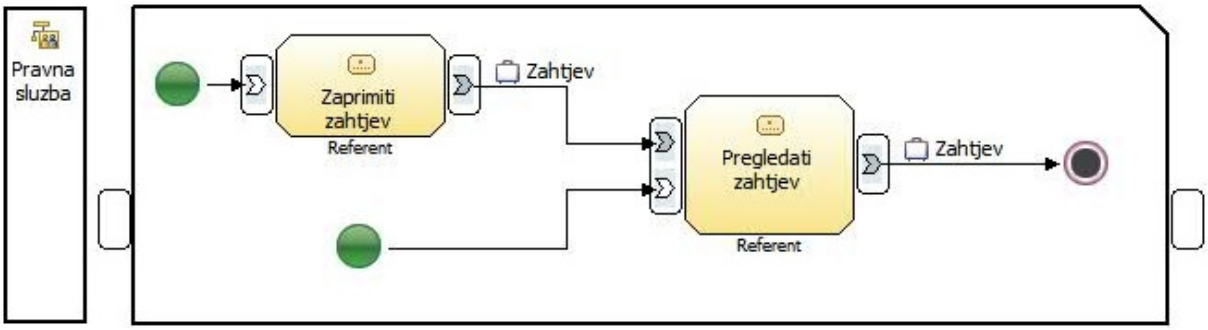
Više početnih događaja u modelu

Opis – ova anomalija nastaje u slučaju kada u modelu poslovnog procesa postoji više od jednog početnog događaja. Navedena anomalija prvi puta spominje se u radu grupe autora Kim, Lee, Son pod nazivom *Classification and Analyses of Business Process Anomalies* 2009. godine [29]. Ovdje valja napomenuti da BPMN 2.0 specifikacija dozvoljava više početnih događaja [10] u modelu, no zbog brojnih dijalekata BPMN notacije u alatima za modeliranja, ovu anomaliju treba promatrati isključivo prema alatu modeliranja.



Slika 11. Anomalija ***Više početnih događaja u modelu***

Primjer – prikazuje navedenu anomaliju izvedenu u *IBM Websphere* alatu za modeliranje poslovnih procesa. Na Slici 12. vide se dva početka događaja u modelu, što je uzrok nastanka anomalije ***Više početnih događaja u modelu***.



Slika 12. Primjer anomalije ***Više početnih događaja u modelu***

Način prepoznavanja anomalije – ova anomalija može se prepoznati na temelju sljedećeg pravila (za *IBM Websphere*):

- u modelu poslovnog procesa postoji više od jednog nedefiniranog početnog događaja.

Ovo pravilo izraženo u OCL jeziku izgleda ovako:

```
1  context Process
2  inv MultipleStartEvents:
3    (self.events.Start.Count()>1)
4    implies
5    MultipleStartEventsAnomaly()
```

Kôd 1. Način prepoznavanja anomalije ***Više početnih događaja u modelu***

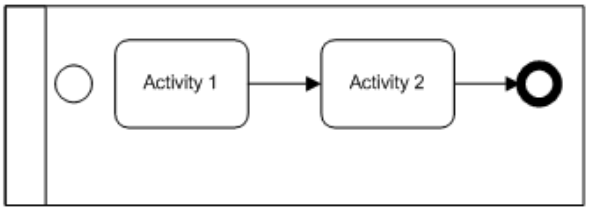
Način rješavanja anomalije – ovu anomaliju najbolje je spriječiti ugradnjom OCL pravila u BPMN 2.0 metamodel, gdje se provjerava da li je broj početnih događaja jedan ili manje od jedan.

```
1  context Process
2  inv MultipleStartEventsAnomaly:
3     self.Events.Start.Count()<=1
```

Kôd 2. Način rješavanja anomalije ***Više početnih događaja u modelu***

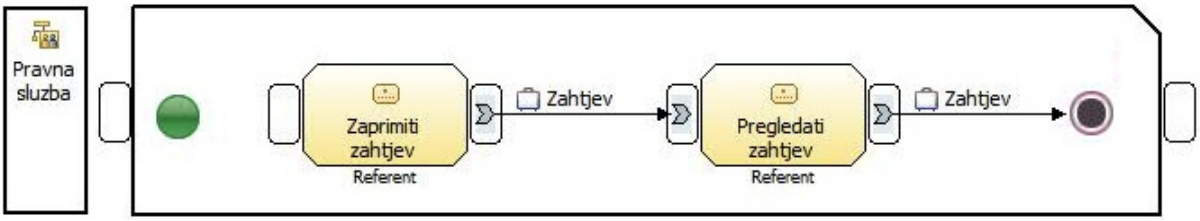
Nepovezanost početnog događaja s ostalim elementima modela

Opis – ova anomalija nastaje ako početni događaj nema izlaznu sekvencu, odnosno početni događaj nije spojen s ostalim elementima u modelu poslovnog procesa. Anomalija ***Nepovezanost početnog događaja s ostalim elementima modela*** poznata je još od rada grupe autora Kim, Lee, Son pod nazivom *Classification and Analyses of Business Process Anomalies,* 2009. godine [29].



Slika 13. Anomalija ***Nepovezanost početnog događaja***

Primjer – na sljedećoj slici prikazan je primjer modela u kojem početni događaj nije spojen s ostalim elementima u modelu, tj. početni događaj nema izlaznu sekvencu.



Slika 14. Primjer anomalije ***Nepovezanost početnog događaja***

Način prepoznavanja anomalije – navedena anomalija može se otkriti primjenom sljedećeg pravila:

- u modelu poslovnog procesa početni događaj nema izlaznu sekvencu.

Navedeno pravilo izvedeno u OCL jeziku izgleda ovako:

```
1 context StartEvent
2 inv StartEventHaveNotOutputSequence:
3    (self.outputSequenceFlows()->IsEmpty())
4    implies
5    StartEventHaveNotOutputSequenceAnomaly()
```

Kôd 3. Način prepoznavanja anomalije ***Nepovezanost početnog događaja s ostalim elementima modela***

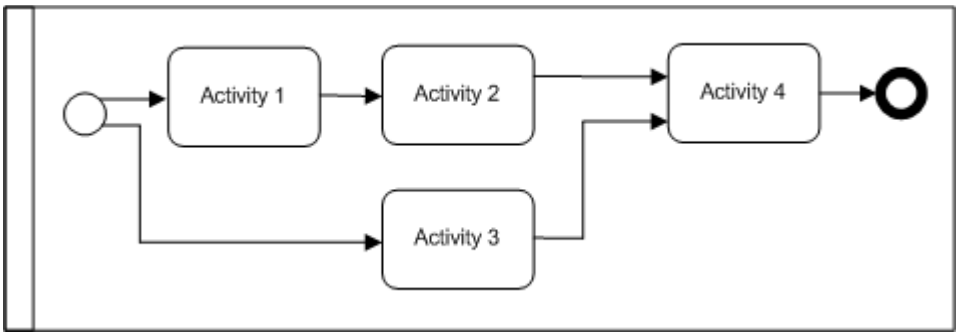
Način rješavanja anomalije – ovu anomaliju treba riješiti definicijom OCL pravila koje neće dozvoliti da početni objekt nema izlaznu sekvencu.

```
1 context StartEvent
2 inv StartEventHaveNotOutputSequenceAnomaly:
3    self.outputSequenceFlows()->notEmpty()
```

Kôd 4. Način rješavanja anomalije ***Nepovezanost početnog događaja s ostalim elementima modela***

Početni događaj ima više izlaznih sekvenci

Opis – početni događaj može imati samo jednu izlaznu sekvencu, dakle ne može se dogoditi da početni događaj ima dvije izlazne sekvence na dvije aktivnosti koje se zatim mogu početi izvoditi. Ovdje valja napomenuti da BPMN 2.0 specifikacija dozvoljava više izlaznih sekvenci iz početnog događaja [10], no zbog brojnih dijalekata BPMN notacije u alatima za modeliranja, ovu anomaliju treba promatrati isključivo prema alatu modeliranja.



Slika 15. Anomalija ***Početni događaj ima više izlaznih sekvenci***

Način prepoznavanja anomalije – temelji se na sljedećim pravilima (za *IBM Websphere*):

- u modelu poslovnog procesa mora postojati barem jedan početni događaj i
- početni događaj ima više od jedne izlazne sekvence.

Ako se navedena pravila izvedu u OCL jeziku onda to izgleda ovako:

```
1 context StartEvent
2 inv StartEventHaveMoreThanOneOutputSequence:
3    Count(Process.StartEvents) > 0
4    AND Count(self.outputSequenceFlows()) > 1
5    implies
6    StartEventHaveMoreThanOneOutputSequenceAnomaly()
```

Kôd 5. Način prepoznavanja anomalije ***Početni događaj ima više izlaznih sekvenci***

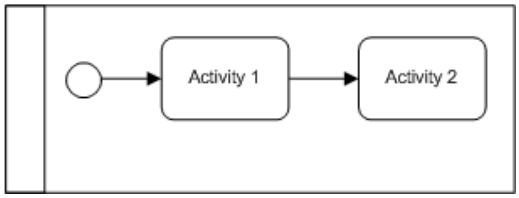
Način rješavanja anomalije – temelji se na OCL pravilu koje neće dozvoliti da početni događaj nema ni jednu ni više od jedne izlazne sekvence.

```
1  context StartEvent
2  inv StartEventHaveMoreThanOneOutputSequenceAnomaly:
3     self.outputSequenceFlows()->notEmpty()
4     AND count(self.outputSequenceFlows())=1
```

Kôd 6. Način rješavanja anomalije ***Početni događaj ima više izlaznih sekvenci***

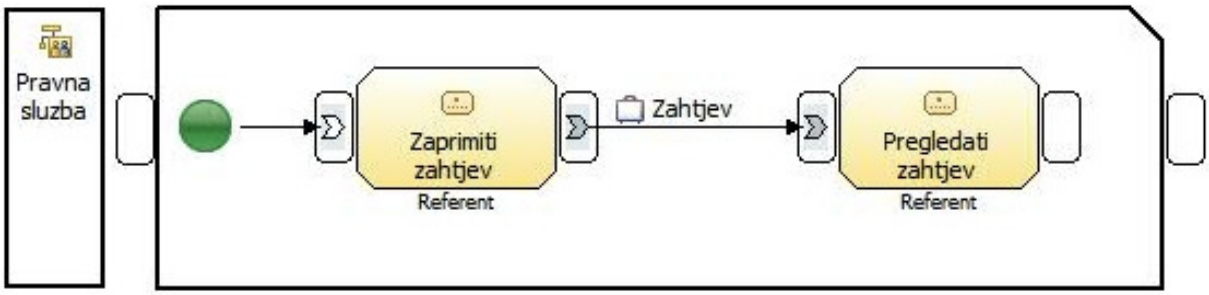
Nepostojanje završnog događaja

Opis – u BPMN notaciji početni događaj može biti opcionalan, dakle model može, ali i ne mora imati početni događaj. S druge strane završni događaj je obavezan u svakom modelu poslovnog procesa koji je izveden u BPMN notaciji. Navedenu anomaliju većina alata za modeliranje poslovnih procesa provjerava već tijekom aktivnosti modeliranja.



Slika 16. Anomalija ***Nepostojanje završnog događaja***

Primjer – sljedeća slika pokazuje model poslovnog procesa u kojem nije definiran završni događaj.



Slika 17. Primjer anomalije ***Nepostojanje završnog događaja***

Način prepoznavanja anomalije – izvodi se primjenom sljedećeg pravila:

- u modelu poslovnog procesa ne postoji završni događaj.

Navedeno pravilo opisano u OCL jeziku izgleda ovako:

```
1  context Process
2  inv ProcessDontHaveEndEvent:
3     (self.Events.End.Count()= 0)
4     implies
5     ProcessDontHaveEndEventAnomaly()
```

Kôd 7. Način prepoznavanja anomalije ***Nepostojanje završnog događaja***

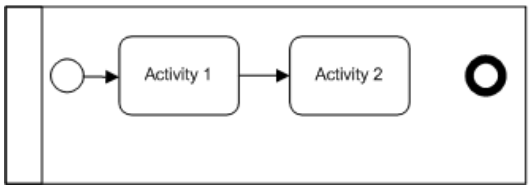
Način rješavanja anomalije – ovu anomaliju trebalo bi riješiti sljedeće OCL pravilo u kojem se ne dozvoljava da model procesa ima završnih događaja manje od jedan:

```
1  context Process
2  inv ProcessDontHaveEndEventAnomaly:
3     count(self.Events.End())=>1
```

Kôd 8. Način rješavanja anomalije ***Nepostojanje završnog događaja***

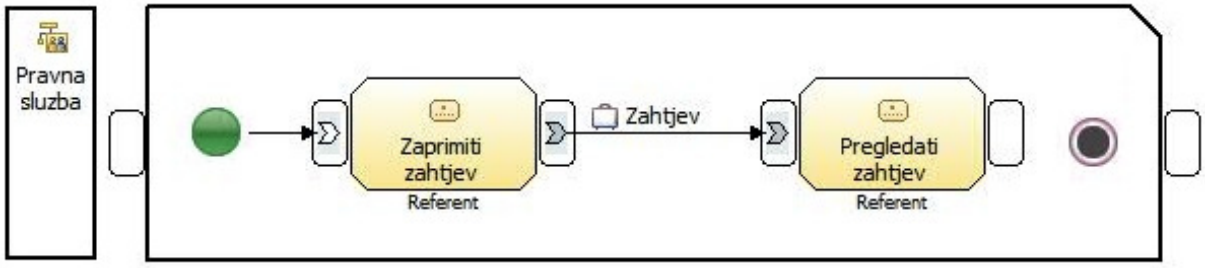
Nepovezanost završnog događaja s ostalim elementima modela

Opis - završni događaj je opcionalan objekt u modelima poslovnih procesa, no u cijelom poslovnom procesu mora biti najmanje jedan završni događaj. Iz ovog proizlazi da su nepravilnosti uporabe završnog događaja jednaka nepravilnoj uporabi početnog događaja. No završni događaj se koristi iako nema početnog događaja. Završni događaj mora biti spojen s ostalim elementima modela, kako bi se model izvršio do kraja.



Slika 18. Anomalija ***Nepovezanost završnog događaja s ostalim elementima modela***

Primjer – na sljedećem primjeru može se vidjeti anomalija ***Nepovezanosti završnog događaja s ostalim elementima modela***.



Slika 19. Primjer anomalije ***Nepovezanosti završnog događaja s ostalim elementima modela***

Način prepoznavanja anomalije – izvodi se na temelju sljedećeg pravila:

- u modelu poslovnog procesa ulazna sekvenca završnog događaja je prazan skup.

Navedeno pravilo opisano u OCL jeziku izgleda na sljedeći način:

```
1  context Event.End
2  inv EndEventHaveNotInputSequence:
3     (self.inputSequenceFlows()->IsEmpty())
4     implies
5     EndEventHaveNotInputSequenceAnomaly()
```

Kôd 9. Način prepoznavanja anomalije ***Nepovezanost završnog događaja s ostalim elementima modela***

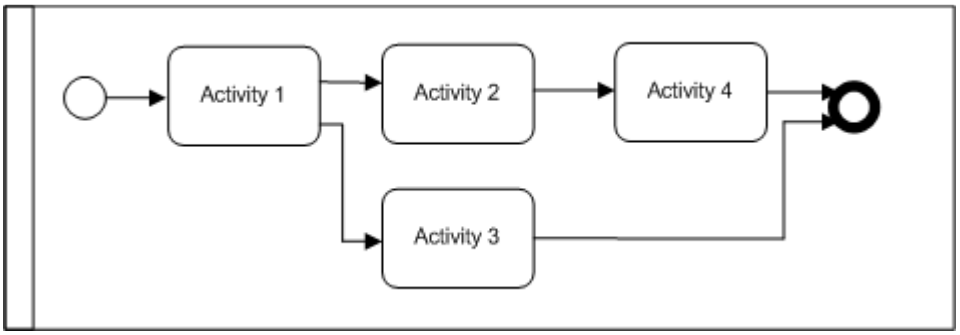
Način rješavanja anomalije – ovu anomaliju trebalo bi riješiti sljedeće OCL pravilo u kojem se ne dozvoljava da model procesa ima završnih događaja bez ulazne sekvence:

```
1  context Event.End
2  inv EndEventHaveNotInputSequenceAnomaly:
3     self.inputSequenceFlows()->notEmpty()
```

Kôd 10. Način rješavanja anomalije ***Nepovezanost završnog događaja s ostalim elementima modela***

Završni događaj ima više od jedne ulazne sekvence

Opis – svaki završni događaj mora imati maksimalno jednu ulaznu sekvencu. Ukoliko završni događaj ima više ulazni sekvenci takav model poslovnog procesa je neispravan. Ovdje valja napomenuti da BPMN 2.0 specifikacija dozvoljava više ulaznih sekvenci u završni događaj [10] u modelu, no zbog brojnih dijalekata BPMN notacije u alatima za modeliranja, ovu anomaliju treba promatrati isključivo prema alatu modeliranja.



Slika 20. Anomalija ***Završni događaj ima više od jedne ulazne sekvence***

Način prepoznavanja anomalije – temelji se na sljedećim pravilima (za *IBM Websphere*):

- u modelu poslovnog procesa mora biti barem jedan završni događaj i
- završni događaj ne smije imati više od jedne ulazne sekvence.

Ako se navedena pravila opišu pomoću OCL jezika, onda to izgleda ovako:

```
1  context Event.End
2  inv EndEventHaveMoreThanOneInputSequence:
3     Count(Process.Event.End)> 0
4     AND Count(self.inputSequenceFlows()=1
5     implies
6     EndEventHaveMoreThanOneInputSequenceAnomaly()
```

Kôd 11. Način prepoznavanja anomalije ***Završni događaj ima više od jedne ulazne sekvence***

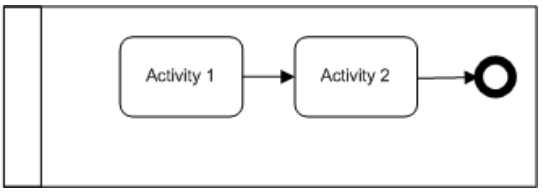
Način rješavanja anomalije – temelji se na OCL pravilu, koje će onemogućiti da završni događaj ima praznu ulaznu sekvencu, te više od jedne ulazne sekvence.

```
1  context Event.End
2  inv EndEventHaveMoreThanOneInputSequenceAnomaly:
3     self.inputSequenceFlows()->notEmpty()
4  AND Count(self.inputSequenceFlows) = 1
```

Kôd 12. Način rješavanja anomalije ***Završni događaj ima više od jedne ulazne sekvence***

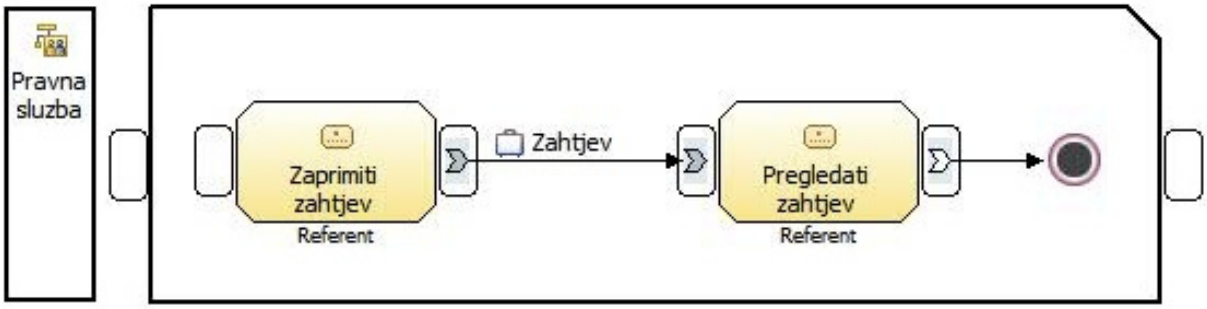
Nepostojanje početnog događaja

Opis – ako je model nije iniciran izvan poslovnog procesa, onda mora postojati početni događaj. Ili ako unutar procesa ne postoji početni događaj, onda mora postojati ulazni tok sekvence u poslovni proces.



Slika 21. Anomalija ***Nepostojanje početnog događaja***

Primjer – sljedeći primjer pokazuje model poslovnog procesa u kojem ne postoji početni događaj, a ni ulazna sekvenca poslovnog procesa.



Slika 22. Anomalija ***Nepostojanje početnog događaja***

Način prepoznavanja anomalije – uključuje sljedeća pravila prepoznavanja:

- u modelu poslovnog procesa ne postoji ulazna sekvenca i
- u modelu poslovnog procesa broj početnih događaja jednak je nuli.

Ako se navedena pravila prikažu u OCL jeziku, onda to izgleda ovako:

```
1  context Process
2  inv ProcessDontHaveStartEvent:
3     ((self.inputSequenceFlows()->IsEmpty())AND (self.StartEvents.Count()=0)
4     implies
5     ProcessDontHaveStartEventAnomaly()
```

Kôd 13. Način prepoznavanja anomalije ***Nepostojanje početnog događaja***

Način rješavanja anomalije – temelji se na OCL pravilu koje bi onemogućilo da u procesu ne postoji početni događaj i ulazna sekvenca toka.

```
1  context Process
2  inv ProcessDontHaveStartEventAnomaly:
3     count(self.StartEvents())=1 OR
4     (self.inputSequenceFlows()->IsNotEmpty()
```

Kôd 14. Način rješavanja anomalije ***Nepostojanje početnog događaja***

3.1.2. Nepravilna uporaba veza

Veze ili tok sekvence predstavljaju grafičke linije koje se koriste za prikaz slijeda aktivnosti koje će se izvesti unutar procesa. Svaki tok može imati samo jedan izvor i samo jedno odredište. Tok sekvence unutar procesa povezuje sve elemente unutar dijagrama, od početnog događaja, do završnog događaja.

Nepravilna uporaba veza najčešće se manifestira kod slučajeva kad u modelu postoji [29]:

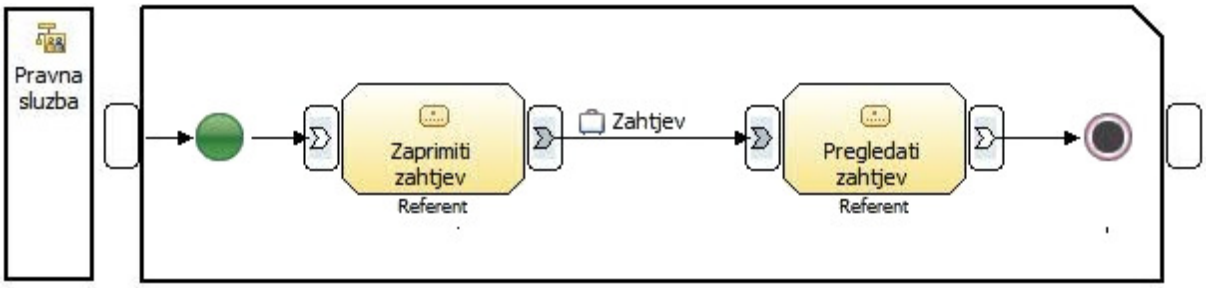
- ulazna sekvenca sa početnim događajem i
- izlazna sekvenca sa završnim događajem.

Ulazna sekvenca s početnim događajem

Opis – početni događaj ne smije imati ulaznu sekvencu. U slučaju da se takva situacija dogodi, početni događaj nije ni potreban, jer je poslovni proces u tom slučaju iniciran izvan poslovnog procesa. Početni događaj ne smije biti odredište toka sekvence. Što znači da početni događaj ne smije imati ulaz, ako ulazi postoje u tom slučaju ne radi se o početnom događaju.

Slika 23. Anomalija ***Ulazne sekvence sa početnim događajem***

Primjer – na sljedećoj slici može se vidjeti kao izgleda kad početni događaj ima ulaznu sekvencu. Takav slučaj se kosi sa sintaksom BPMN 2.0 grafičke notacije.



Slika 24. Primjer anomalije ***Ulazna sekvenca sa početnim događajem***

Način prepoznavanja anomalije - izvodi se na temelju sljedećeg pravila:

- u modelu poslovnog procesa početni događaj ima ulaznu sekvencu.

Ako se navedeno pravilo prikaže pomoću OCL jezika onda bi on izgledao ovako:

```
1  context Event.Start
2  inv StartEventHasInputSequence:
3     (self.inputSequenceFlows()->notEmpty())
4     implies
5     StartEventHasInputSequenceAnomaly()
```

Kôd 15. Način prepoznavanja anomalije ***Ulazna sekvenca sa početnim događajem***

Način rješavanja anomalije - kao način rješavanja predlaže se dodatno definiranje OCL pravila u kojem stoji da početni događaj mora imati definiranu izlaznu sekvencu, dok ulaz mora biti prazan. U OCL jeziku to se može napisati na sljedeći način:

```
1  context Event.Start
2  inv StartEventHasInputSequenceAnomaly:
3     self.inputSequenceFlows()->isEmpty()
4     AND self.outputSequenceFlows()->notEmpty()
```

Kôd 16. Način rješavanja anomalije ***Ulazna sekvenca sa početnim događajem***

Posebni slučajevi anomalije – mogu se prepoznati u tablici dozvoljenih veza sekvence toka.

Tabela 13. Pravila dozvoljenih veza sekvence toka (Izvor: OMG.org)

Od/Prema	Početni događaj	Potproces	Aktivnost	Vrata	Stop događaj	Završni događaj
Početni događaj		•	•	•	•	•
Potproces		•	•	•	•	•
Aktivnost		•	•	•	•	•
Vrata		•	•	•	•	•
Stop događaj		•	•	•	•	•
Završni događaj						

Iz navedene tabele mogu se izvući sljedeće anomalije:

- početni događaj spojen na element početnog događaja,
- potproces spojen na element početnog događaja,
- aktivnost spojena na element početnog događaja,
- vrata spojena na element početnog događaja,
- stop događaj spojen na element početnog događaja i
- završnog događaj spojen na element početnog događaja.

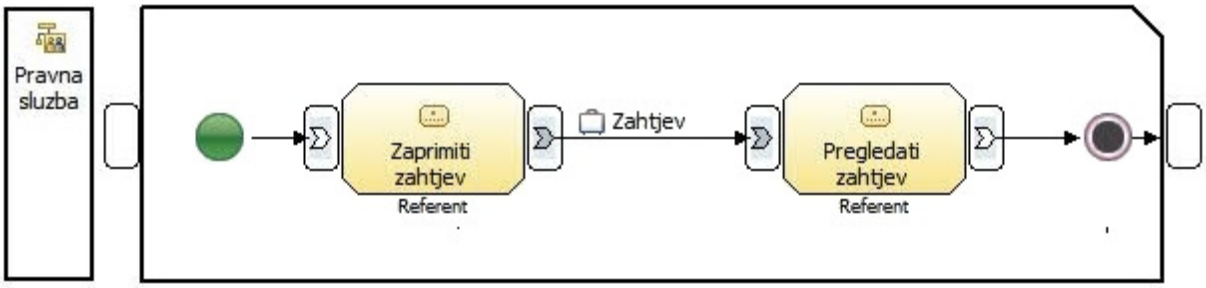
Sve navedene anomalije rješavaju se na isti način kao i anomalija ulazna sekvenca sa početnim događajem. Iz tog razloga navedene anomalije nisu posebno obrađivane.

Izlazna sekvenca s završnim događajem

Opis – Kod izlazne sekvence sa završnim događajem nepravilna uporaba nastaje u slučaju da završni događaj ima izlaznu sekvencu. Odnosno ne smije postojati objekt koji ima ulazni tok iz završnog događaja.

Slika 25. Anomalija ***Izlazna sekvenca sa završnim događajem***

Primjer – na Slici 26. Vidi se primjer anomalije gdje završni događaj ima izlaznu sekvencu, čime ovaj primjer krši pravila BPMN 2.0 notacije.



Slika 26. Primjer anomalije ***Izlazne sekvenca sa završnim događajem***

Način prepoznavanja anomalije - izvodi se na temelju sljedećeg pravila:

- u modelu poslovnog procesa završni događaj ima izlaznu sekvencu.

Ako se navedeno pravilo prikaže pomoću OCL jezika onda to izgleda ovako:

```
1  context Event.End
2  inv EvenEndtHasOutputSequence:
3     (self.outputSequenceFlows()-> notEmpty())
4     implies
5     EventEndHasOutputSequenceAnomaly()
```

Kôd 17. Način prepoznavanja anomalije ***Izlazna sekvenca sa završnim događajem***

Način rješavanja anomalije - za rješavanje ove anomalije potrebno je definirati OCL pravilo u BPMN 2.0 metamodelu kojim će se onemogućiti izlazne sekvence iz završnog događaja.

```
1  context Event.End
2  inv EventEndHasOutputSequenceAnomaly:
3     self.outputSequenceFlows()-> isEmpty()
4     AND self.inputSequenceFlows()-> notEmpty()
```

Kôd 18. Način rješavanja anomalije ***Izlazna sekvenca sa završnim događajem***

Ako je međudogađaj spojen na granice aktivnosti ono ne smije biti odredište toka sekvence. Međudogađaj ne smije imati ulazni tok sekvence.

Do nepravilne uporabe višestruke izlazne sekvence sa međudogađajem pridruženim granici aktivnosti dolazi ako se ne poštuje pravilo da ako je međudogađaj pridružen granici aktivnosti ono može imati samo jednu izlaznu sekvencu osim u slučaju da je međudogađaj okidač tipa kompenzacija. Postoje četiri uvjeta koja moraju biti zadovoljena za uporabu uvjetnog toka sekvence i to:

- izvor ne smije biti događaj,
- ako su izvor vrata, ne smije se koristiti uvjetni tok sekvence,
- izvorna vrata ne smiju biti I (eng. *AND*) vrata i
- ako se uvjetni tok sekvence koristi iz aktivnosti, tada mora postojati barem jedan izlazni tok sekvence iz aktivnosti.

Ako nisu ispunjeni navedeni uvjeti, tada postoji nepravilna uporaba toka sekvence. Kod nepravilne uporabe toka poruke u jednom spremniku (eng. *Pool*) dolazi ako se prekrši pravilo da se tokovi poruka koriste za spajanje dviju spremnika ili unutar spremnika ili objekte toka sa porukama spremnika.

Nepravilna uporaba toka poruka sa početnim događajem i vratima događa se u slučaju ako je tok poruke povezan sa početnim događajem ili vratima unutar spremnika.

Posebni slučajevi anomalije – mogu se prepoznati u tablici dozvoljenih veza sekvence toka.

Posebni slučajevi anomalija izlazne sekvence sa završnim događajem izvedeni iz tabele dozvoljenih veza sekvence toka:

- završni događaj spojen na element početnog događaja,
- završni događaj spojen na element potprocesa,
- završni događaj spojen na element aktivnosti,
- završni događaj spojen na element vrata,
- završni događaj spojen na element stop događaja i
- završni događaj spojen na element završnog događaja.

Navedene anomalije rješavaju se na isti način kao i anomalija izlazna sekvenca sa završnim događajem, pa iz toga razloga svaka od navedenih anomalija nije posebno obrađivana.

3.1.3. Nepravilna uporaba toka poruka

Temeljne anomalije nepravilne uporabe toka poruka ukratko se mogu prikazati sljedećom tablicom:

Tabela 14. Pravila veza toka poruka (izvor: OMG.org)

Od/Prema	Početni događaj poruke	Bazen	Potproces	Aktivnost	Stop događaj poruke	Završni događaj poruke
Početni događaj poruke						
Bazen	•	•	•	•	•	
Potproces	•	•	•	•	•	
Aktivnost	•	•	•	•	•	
Stop događaj poruke	•	•	•	•	•	
Završni događaj poruke	•	•	•	•	•	

Iz gore navedene tabele mogu se izvesti sljedeća pravila:

- početni događaj poruke vezan je tokom poruke na element početnog događaja,
- početni događaj poruke vezan je tokom poruke sa elementom bazena,
- početni događaj poruke vezan je tokom poruke na element potprocesa,
- početni događaj poruke vezan je tokom poruke na element aktivnosti,
- početni događaj vezan je tokom poruke sa elementom stop događanja poruke i
- početni događaj vezan je tokom poruke sa elementom završnog događaja poruke.

Kao što je vidljivo iz prethodne tabele, navedena pravila su složenija od tablice pravila dozvoljenih sekvenci veze toka. Navedene anomalije rješavaju se na način da se korisnik obavijesti o nedozvoljenoj sintaktičkoj grešci. S obzirom da je prepoznavanje navedenih anomalija ugrađeno u sve važnije alate za BPMN modeliranje, traženje takvog primjera u postojećim alatima se je pokazalo nemogućim. Iz tog razloga uz svaku on navedenih anomalija biti će opis, način prepoznavanja i način rješavanja, dok će se izostaviti primjeri iz postojećih alata za modeliranje.

Početni događaj poruke vezan je tokom poruke s elementom početnog događaja

Opis - kao što je vidljivo iz tabele 13. početni događaj ne smije biti vezan tokom poruke sa elementom početnog događaja.

Način prepoznavanja anomalije - anomalija postoji ako je zadovoljeno sljedeće pravilo:

- u modelu poslovnog procesa početni događaj poruke vezan je tokom poruke sa elementom početnog događaja.

Navedeno pravilo opisano u pseudokodu izgleda ovako:

```
RULE: EventStartConnectedToEventStart
IF: (Mesage-Flow EXIST BETWEEN Event.Start AND Event.Start)
THEN: EventStartConnectedToEventStartAnomaly
ENDIF
```

Kôd 19. Način prepoznavanja anomalije ***Početni događaj poruke vezan je tokom poruke na element početnog događaja***

Način rješavanja anomalije - temelji se na generiranju poruke korisniku da navedena veza nije po pravilu BPMN sintakse.

```
FIX: EventStartConnectedToEventStartAnomaly
     strMessage = "Start event can not be connected with message flow to Start event"
```

Kôd 20. Način rješavanja anomalije ***Početni događaj poruke vezan je tokom poruke na element početnog događaja***

Početni događaj poruke vezan je tokom poruke s elementom bazena

Opis - početni događaj ne smije biti vezan tokom poruke sa elementom bazena. Time se krši pravilo BPMN 2.0 sintakse.

Način prepoznavanja anomalije - izvodi se pomoću sljedećeg pravila koje kaže da:

- u modelu poslovnog procesa početni događaj poruke vezan je tokom poruke sa elementom bazena.

Navedeno pravilo opisano u pseudokodu izgleda ovako:

```
RULE: EventStartConnectedToPool
IF: (Mesage-Flow EXIST BETWEEN Event.Start AND Pool)
THEN: EventStartConnectedToPoolAnomaly
ENDIF
```

Kôd 21. Način prepoznavanja anomalije ***Početni događaj poruke vezan je tokom poruke na element bazena***

Način rješavanja anomalije - temelji se na generiranju poruke korisniku da navedena veza nije po pravilu BPMN 2.0 sintakse.

```
FIX: EventStartConnectedToPoolAnomaly
     strMessage = "Start event can not be connected with message flow to Pool"
```

Kôd 22 Način rješavanja anomalije ***Početni događaj poruke vezan je tokom poruke na element bazena***

Početni događaj poruke vezan je tokom poruke s elementom potprocesa

Opis - početni događaj ne smije biti vezan tokom poruke sa elementom potprocesa. Time se krši pravilo sintakse BPMN 2.0 notacije.

Način prepoznavanja anomalije - izvodi se pomoću pravila koje kaže da:

- u modelu poslovnog procesa početni događaj poruke vezan je tokom poruke sa elementom potprocesa.

Navedeno pravilo opisano u pseudokodu izgleda ovako:

```
RULE: EventStartConnectedToSubProcess
IF: (Mesage-Flow EXIST BETWEEN Event.Start AND SubProcess)
THEN: EventStartConnectedToSubProcessAnomaly
ENDIF
```

Kôd 23. Način prepoznavanja anomalije ***Početni događaj poruke vezan je tokom poruke na element potprocesa***

Način rješavanja anomalije - temelji se na generiranju poruke korisniku da navedena veza nije po pravilu BPMN 2.0 sintakse.

```
FIX: EventStartConnectedToSubProcessAnomaly
     strMessage = "Start event can not be connected with message flow to SubProcess"
```

Kôd 24. Način rješavanja anomalije ***Početni događaj poruke vezan je tokom poruke na element potprocesa***

Početni događaj poruke vezan je tokom poruke s elementom aktivnosti

Opis - početni događaj ne smije biti vezan tokom poruke sa elementom aktivnosti, jer time se krši pravilo sintakse BPMN 2.0 notacije.

Način prepoznavanja anomalije - izvodi se pomoću pravila koje kaže da:

- u modelu poslovnog procesa početni događaj poruke vezan je tokom poruke sa elementom aktivnosti.

Navedeno pravilo opisano u pseudokodu izgleda ovako:

```
RULE: EventStartConnectedToActivity
IF: (Mesage-Flow EXIST BETWEEN Event.Start AND Activity)
THEN: EventStartConnectedToActivityAnomaly
ENDIF
```

Kôd 25. Način prepoznavanja anomalije ***Početni događaj poruke vezan je tokom poruke na element aktivnosti***

Način rješavanja anomalije - temelji se na generiranju poruke korisniku da navedena veza nije po pravilu BPMN 2.0 sintakse. Dakle, navedeno rješenje ne rješava problem ukoliko se anomalija pojavi, već obavještava korisnika da postoji i da je mora ukloniti.

```
FIX: EventStartConnectedToActivityAnomaly
     strMessage = "Start event can not be connected with message flow to Activity"
```

Kôd 26. Način rješavanja anomalije ***Početni događaj poruke vezan je tokom poruke na element aktivnosti***

Početni događaj vezan je tokom poruke s elementom stop događaja poruke

Opis - početni događaj ne smije biti vezan tokom poruke sa elementom aktivnosti, jer time se krši pravilo sintakse BPMN 2.0 notacije. Anomalija nastane ako su ta dva elementa spojena tim redoslijedom.

Način prepoznavanja anomalije - izvodi se pomoću pravila koje kaže da:

- u modelu poslovnog procesa početni događaj poruke vezan je tokom poruke sa elementom stop događaja.

Navedeno pravilo opisano u pseudokodu izgleda ovako:

```
RULE: EventStartConnectedToEventStop
IF: (Mesage-Flow EXIST BETWEEN Event.Start AND Event.Stop)
THEN: EventStartConnectedToEventStopAnomaly
ENDIF
```

Kôd 27. Način prepoznavanja anomalije ***Početni događaj poruke vezan je tokom poruke na element stop događaja poruke***

Način rješavanja anomalije - temelji se na generiranju poruke korisniku da navedena veza nije po pravilu BPMN 2.0 sintakse.

```
FIX: EventStartConnectedToEventStopAnomaly
     strMessage = "Start event can not be connected with message flow to Event stop"
```

Kôd 28. Način rješavanja anomalije ***Početni događaj poruke vezan je tokom poruke na element stop događaja poruke***

Početni događaj vezan je tokom poruke s elementom završnog događaja poruke

Opis - početni događaj ne smije biti vezan tokom poruke sa elementom završnog događaja poruke. Time se krši pravilo sintakse BPMN 2.0 notacije.

Način prepoznavanja anomalije - izvodi se pomoću pravila koje kaže da:

- u modelu poslovnog procesa početni događaj poruke vezan je tokom poruke sa elementom završnog događaja poruke.

Navedeno pravilo opisano u pseudokodu izgleda ovako:

```
RULE: EventStartConnectedToEventEnd
IF: (Mesage-Flow EXIST BETWEEN Event.Start AND Event.End)
THEN: EventStartConnectedToEventEndAnomaly
ENDIF
```

Kôd 29. Način prepoznavanja anomalije ***Početni pogađaj poruke vezan je tokom poruke na završni element događanja poruke***

Način rješavanja anomalije - temelji se na generiranju poruke korisniku da navedena veza nije po pravilu BPMN 2.0 sintakse.

```
FIX: EventStartConnectedToEventEndAnomaly
     strMessage = "Start event can not be connected with message flow to Event End"
```

Kôd 30. Način rješavanja anomalije ***Početni događaj poruke vezan je tokom poruke na završni element događaja poruke***

Početni događaj poruke vezan je tokom poruke s završnim događajem

Opis - početni događaj ne smije biti vezan tokom poruke sa elementom završnog događaja. Time se krši pravilo sintakse BPMN 2.0 notacije.

Način prepoznavanja anomalije - izvodi se pomoću pravila koje kaže da:

- u modelu poslovnog procesa početni događaj poruke vezan je tokom poruke sa elementom završnog događaja.

Navedeno pravilo opisano u pseudokodu izgleda ovako:

```
RULE: EventStartConnectedToEventEnd
IF: (Mesage-Flow EXIST BETWEEN Event.Start AND Event.End)
THEN: EventStartConnectedToEventEndAnomaly
ENDIF
```

Kôd 31. Način prepoznavanja anomalije ***Početni događaj poruke vezan tokom poruke s završnim događajem***

Način rješavanja anomalije - temelji se na generiranju poruke korisniku da navedena veza nije po pravilu BPMN 2.0 sintakse.

```
FIX: EventStartConnectedToEventEndAnomaly
     strMessage = "Start event can not be connected with message flow to Event End"
```

Kôd 32. Način rješavanja anomalije ***Početni događaj poruke vezan je tokom poruke s završnim događajem***

Bazen je vezan tokom poruke s završnim događajem

Opis - bazen ne smije biti vezan tokom poruke sa elementom završnog događaja. Time se krši pravilo sintakse BPMN 2.0 notacije.

Način prepoznavanja anomalije - izvodi se pomoću pravila koje kaže da:

- u modelu poslovnog procesa bazen je vezan tokom poruke sa elementom završnog događaja.

Navedeno pravilo opisano u pseudokodu izgleda ovako:

```
RULE: PoolConnectedToEventEnd
IF: (Mesage-Flow EXIST BETWEEN Pool AND Event.End)
THEN: PoolConnectedToEventEndAnomaly
ENDIF
```

Kôd 33. Način prepoznavanja anomalije ***Bazen je vezan tokom poruke sa završnim događajem***

Način rješavanja anomalije - temelji se na generiranju poruke korisniku da navedena veza nije po pravilu BPMN sintakse.

```
FIX: PoolConnectedToEventEndAnomaly
     strMessage = "Pool can not be connected with message flow to Event End"
```

Kôd 34. Način rješavanja anomalije ***Bazen je vezan tokom poruke sa završnim događajem***

Potproces je vezan tokom poruke s završnim događajem

Opis - potproces ne smije biti vezan tokom poruke sa elementom završnog događaja poruke. Time se krši pravilo sintakse BPMN 2.0 notacije.

Način prepoznavanja anomalije - izvodi se pomoću pravila koje kaže da:

- u modelu poslovnog procesa potproces vezan je tokom poruke sa elementom završnog događaja.

Navedeno pravilo opisano u pseudokodu izgleda ovako:

```
RULE: SubProcessConnectedToEventEnd
IF: (Mesage-Flow EXIST BETWEEN SubProcess AND Event.End)
THEN: SubProcessConnectedToEventEndAnomaly
ENDIF
```

Kôd 35. Način prepoznavanja anomalije ***Potproces je vezan tokom poruke sa završnim događajem***

Način rješavanja anomalije - temelji se na generiranju poruke korisniku da navedena veza nije po pravilu BPMN 2.0 sintakse.

```
FIX: SubProcessConnectedToEventEndAnomaly
     strMessage = "Subprocess can not be connected with message flow to Event End"
```

Kôd 36. Način rješavanja anomalije ***Potproces je vezan tokom poruke sa završnim događajem***

Aktivnost je vezana tokom poruke s završnim događajem

Opis - aktivnost ne smije biti vezan a tokom poruke sa elementom završnog događaja poruke. Time se krši pravilo sintakse BPMN 2.0 notacije.

Način prepoznavanja anomalije - izvodi se pomoću pravila koje kaže da:

- u modelu poslovnog procesa aktivnost je vezana tokom poruke sa elementom završnog događaja poruke.

Navedeno pravilo opisano u pseudokodu izgleda ovako:

```
RULE: ActivityConnectedToEventEnd
IF: (Mesage-Flow EXIST BETWEEN Activity AND Event.End)
THEN: ActivityConnectedToEventEndAnomaly
ENDIF
```

Kôd 37. Način prepoznavanja anomalije ***Aktivnost je vezana tokom poruke sa završnim događajem***

Način rješavanja anomalije - temelji se na generiranju poruke korisniku da navedena veza nije po pravilu BPMN 2.0 sintakse.

```
FIX: ActivityConnectedToEventEndAnomaly
     strMessage = "Activity can not be connected with message flow to Event End"
```

Kôd 38. Način rješavanja anomalije ***Aktivnost je vezana tokom poruke sa završnim događajem***

Stop događaj poruke je vezan tokom poruke s završnim događajem

Opis – stop događaj poruke ne smije biti vezan tokom poruke sa elementom završnog događaja. Time se krši pravilo sintakse BPMN 2.0 notacije.

Način prepoznavanja anomalije - izvodi se pomoću pravila koje kaže da:

- u modelu poslovnog procesa stop događaj poruke vezan je tokom poruke s elementom završnog događaja.

Navedeno pravilo opisano u pseudokodu izgleda ovako:

```
1  RULE: EventStopConnectedToEventEnd
2  IF: (Mesage-Flow EXIST BETWEEN Event.Stop AND Event.End)
3  THEN: EventStopConnectedToEventEndAnomaly
4  ENDIF
```

Kôd 39. Način prepoznavanja anomalije ***Stop događaj je vezan tokom poruke sa završnim događajem***

Način rješavanja anomalije - temelji se na generiranju poruke korisniku da navedena veza nije po pravilu BPMN sintakse.

```
1  FIX: EventStopConnectedToEventEndAnomaly
2       strMessage = "Event Stop can not be connected with message flow to Event End"
```

Kôd 40. Način rješavanja anomalije ***Stop događaj vezan je tokom poruke sa završnim događajem***

Element završnog događaja poruke vezan je tokom poruke s elementom završnog događaja

Opis – element završnog događanja poruke ne smije biti vezan a tokom poruke sa elementom završnog događaja poruke. Time se krši pravilo sintakse BPMN 2.0 notacije.

Način prepoznavanja anomalije - izvodi se pomoću pravila koje kaže da:

- u modelu poslovnog procesa element završnog događanja poruke vezan je tokom poruke sa elementom završnog događaja.

Navedeno pravilo opisano u pseudokodu izgleda ovako:

```
1  RULE: EventEndConnectedToEventEnd
2  IF: (Mesage-Flow EXIST BETWEEN Event.End AND Event.End)
3  THEN: EventEndConnectedToEventEndAnomaly
4  ENDIF
```

Kôd 41. Način prepoznavanja anomalije ***Element završnog događaja spojen je tokom poruke sa završnim događajem***

Način rješavanja anomalije - temelji se na generiranju poruke korisniku da navedena veza nije po pravilu BPMN 2.0 sintakse.

```
FIX: EventEndConnectedToEventEndAnomaly
     strMessage = "Event End can not be connected with message flow to Event End"
```

Kôd 42. Način rješavanja anomalije ***Element završnog događaja spojen je tokom poruke sa završnim događaja***

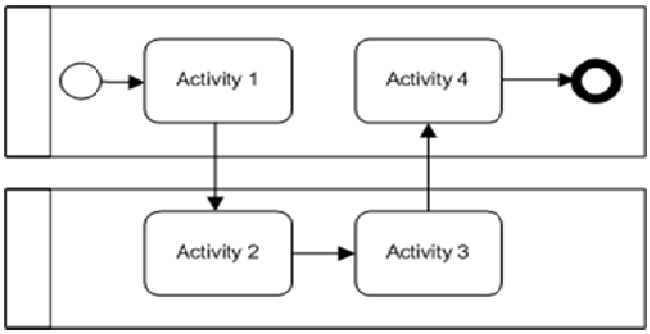
3.1.4. Nepravilna uporaba spremnika

Opis - nepravilna uporaba spremnika može se podijeliti na dvije temeljne kategorije [29]:

- nepravilna uporaba spremnika (eng. *pool*) i
- nepravilna uporaba staza (eng. *lines*).

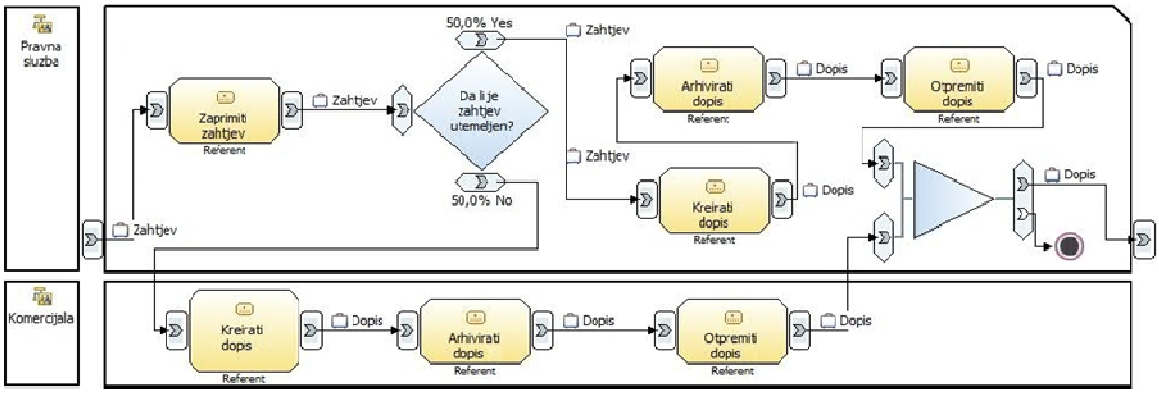
Nepravilna uporaba bazena

Opis - nepravilna uporaba bazena događa se u slučaju kad tok sekvence prelazi granice bazena. Obično granice bazena moraju obuhvatiti sve objekte toka i tokove sekvence. Tok sekvence ne smije prelaziti granice bazena, za to treba koristiti tok poruka.



Slika 27. Anomalija ***Nepravilna uporaba bazena***

Primjer – na sljedećem primjeru vidi se slučaj u kojem tok sekvence prelazi granice bazena.



Slika 28. Primjer anomalije ***Nepravilne uporabe bazena***

Način prepoznavana anomalije – temelji se na sljedećem pravilu:

- u modelu poslovnog procesa postoji bazen ima tok sekvence koji prelazi granice spremnika.

Navedeno pravilo može se riješiti uporabom pravila:

```
RULE: InvalidUseOfPool
IF: (Control-Flow EXIST BETWEEN Pools)
THEN: InvalidUseOfPoolAnomaly
ENDIF
```

Kôd 43. Način prepoznavanja anomalije ***Nepravilna uporaba bazena***

Način rješavanja anomalije - kao način rješavanja predlaže se dodatno definiranje OCL pravila u kojem stoji spremnici ne smiju biti povezani sekvencom. U OCL jeziku to se može napisati na sljedeći način:

```
FIX: InvalidUseOfPoolAnomaly:
   self.inputSequenceFlows()->isEmpty() AND
   self.outputSequenceFlows()->notEmpty()
```

Kôd 44. Način rješavanja anomalije ***Nepravilna uporaba bazena***

Bazen bez elemenata

Opis – anomalija ***Bazen bez elemenata*** nastaje ako u bazenu ne postoji ni jedan element. Dakle sam bazen predstavlja redundantan element. Sljedeća slika prikazuje takvu anomaliju, dakle vidi se prazan bazen bez ijednog elementa.

Slika 29. Anomalija ***Bazen bez elemenata***

Primjer – primjer pokazuje anomalije ***Bazen bez elemenata.***

Slika 30. Primjer anomalije ***Bazen bez elemenata***

Način prepoznavanja anomalije – kao način rješavanja koristit će se OCL pravilo. Temeljni kontekst pravila je bazen (eng. *Pool*), a pravilo je sljedeće:

- u modelu poslovnog procesa postoji bazen u kojem ne postoji ni jedan grafički element.

Ako se navedeno pravilo opiše OCL jezikom, onda to izgleda ovako:

```
1  context Pool
2  inv PoolWithoutElements:
3         (self.GraphicElements()->isEmpty())
4         implies
5         PoolWithoutElementsAnomaly()
```

Kôd 45. Način prepoznavanja anomalije ***Bazen bez elemenata***

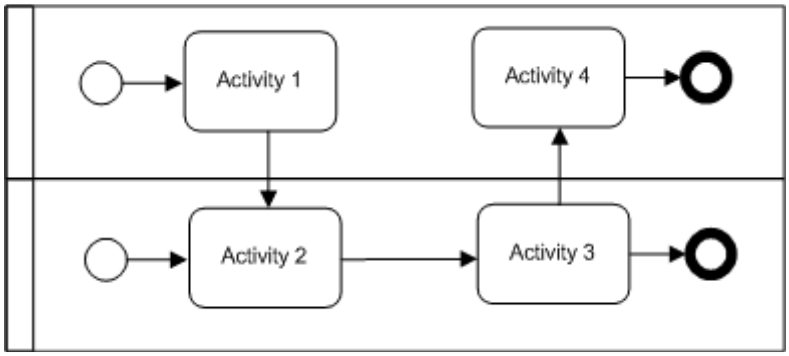
Način rješavanja anomalije - pomoću OCL pravila, ne smije se dozvoliti da model ima bazen bez elemenata.

```
1  context Pool
2  inv PoolWithoutElementsAnomaly:
3         self.GraphicElements()->isEmpty()
```

Kôd 46. Način rješavanja anomalije ***Bazen bez elemenata***

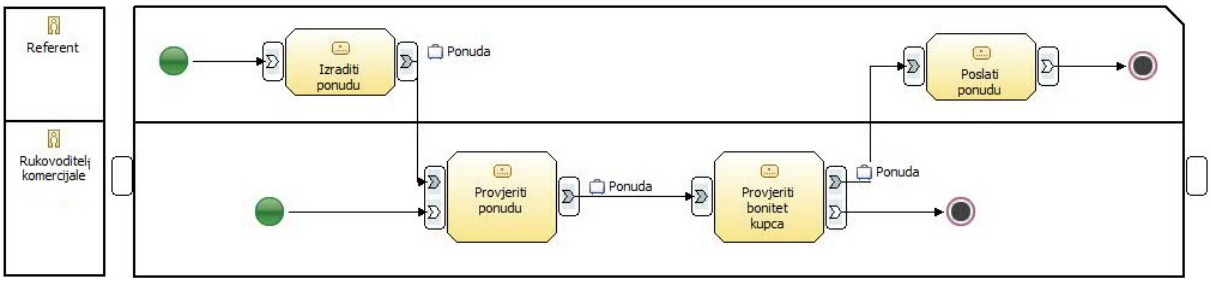
Nepravilna uporaba staza

Opis - anomalija ***Nepravilna uporaba staza*** nastaje ako se staze koriste prije spremnika, jer staze služe za kategoriziranje elemenata modela procesa unutar spremnika.



Slika 31. Anomalija ***Nepravilna uporaba staza***

Primjer – na sljedećem primjeru vidi se anomalija ***Nepravilna uporaba staza***, u ovom slučaju staze se koriste poput spremnika (bazena).



Slika 32. Primjer anomalije ***Nepravilna uporaba staza***

Način prepoznavana anomalije – temelji se na sljedećem pravilu:

- u modelu poslovnog procesa postoji staza, no ne postoji i bazen koji sadrži navedenu stazu.

Ako se navedeno pravilo prikaže kao OCL pravilo, onda to izgleda ovako:

```
1  Context Swimline
2  inv InvalidUseOfSwimline
3     (Swimline.parent() -> isEmpty()
4     implies
5     InvalidUseOfSwimlineAnomaly
```

Kôd 47. Način prepoznavanja anomalije ***Nepravilna uporaba staza***

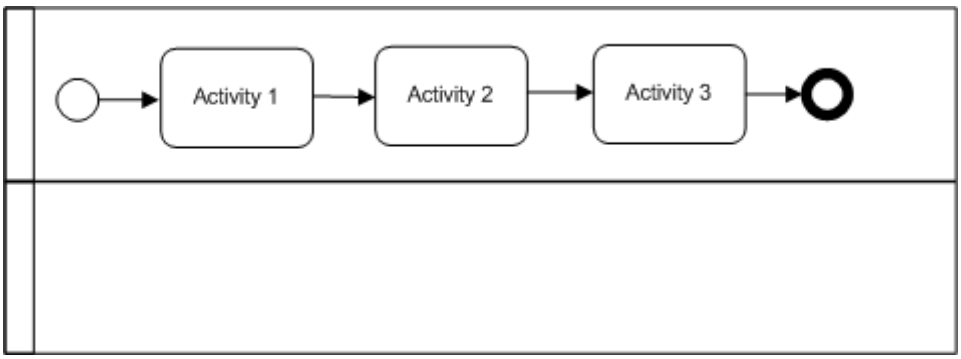
Način rješavanja anomalije – navedenu anomaliju može se riješiti postavljanjem OCL pravila koje neće dozvoliti da staza nema nadređeni bazen:

```
1  context Swimline
2  inv InvalidUseOfSwimlineAnomaly:
3         self.parent()-> isNotEmpty()
```

Kôd 48. Način rješavanja anomalije ***Nepravilna uporaba staza***

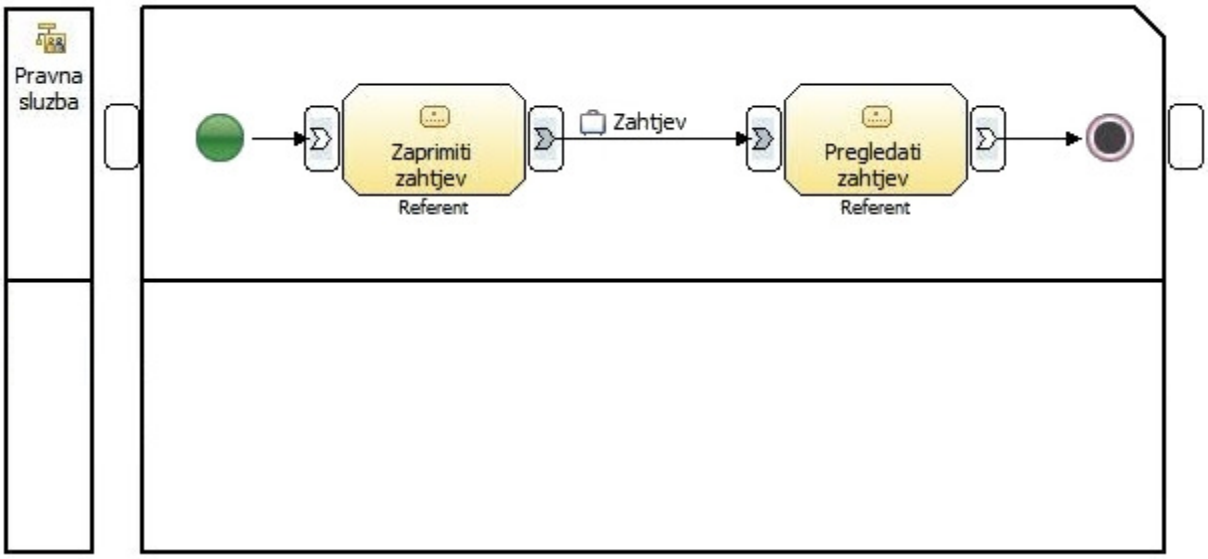
Staza bez elemenata

Opis – ako u modelu poslovnih procesa postoji staza u kojoj ne postoje elementi, tada se može govoriti o anomaliji ***Staza bez elemenata.***



Slika 33. Anomalija ***Staza bez elemenata***

Primjer – pokazuje anomaliju ***Staza bez elemenata*** u kojoj je vidljivo da model ima već jednu stazu u kojoj se nalazi proces, osim toga model ima još jednu stazu u kojoj se ne nalazi ni jedan element modela.



Slika 34. Primjer anomalije ***Staza bez elemenata***

Način prepoznavanja anomalije – obavlja se pomoću sljedećeg pravila prepoznavanja:

- u modelu poslovnog procesa postoji staza bez elemenata.

Ako se navedeno pravilo prikaže pomoću OCL pravila, onda to izgleda ovako:

```
1  context Swimline
2  inv SwimlineWithoutElements:
3         (self.GraphicElements()->isEmpty())
4         implies
5         SwimlineWithoutElementsAnomaly()
```

Kôd 49. Način prepoznavanja anomalije ***Staza bez elementa***

Način rješavanja anomalije - obavlja se preko OCL pravila u kojem se ne dozvoljava da postoji staza koja ne sadrži grafičke elemente.

```
1  context Swimline
2  inv SwimlineWithoutElementsAnomaly:
3         self.GraphicElements()->isEmpty()
```

Kôd 50. Način rješavanja anomalije ***Staza bez elemenata***

3.2. Strukturne anomalije u BPMN 2.0 notaciji

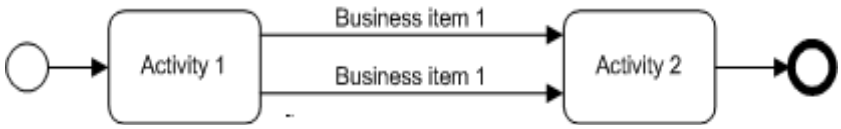
Strukturne anomalije su strukturno-logičkog karaktera. Zahtijevaju uporabu složenih metoda prepoznavanja i uporabu preventivnih mehanizama koji sprečavaju njihovo formiranje prilikom izvođenja. Mogu se podijeliti na **anomalije kontrole toka** i **anomalije pravila**. Pod anomalije kontrole toka spada redundancija kontrole toka, nedefinirani uvjeti vrata, specijalni slučajevi toka, te narušavanje ispravnosti. Anomalije pravila dijele se na anomalije konzistentnosti i anomalije obuhvata pravila. Osim toga pod anomalije pravila spajaju anomalije konzistentnosti u koje spada redundancija pravila, konfliktna pravila i anomalije petlji. Strukturne anomalije ovise o kontekstu i scenariju poslovnog procesa, jer te anomalije iako prepoznate u modelima poslovnih procesa, ne moraju biti problematične ako to scenarij poslovnog procesa zahtjeva.

3.2.1. Kontrola toka

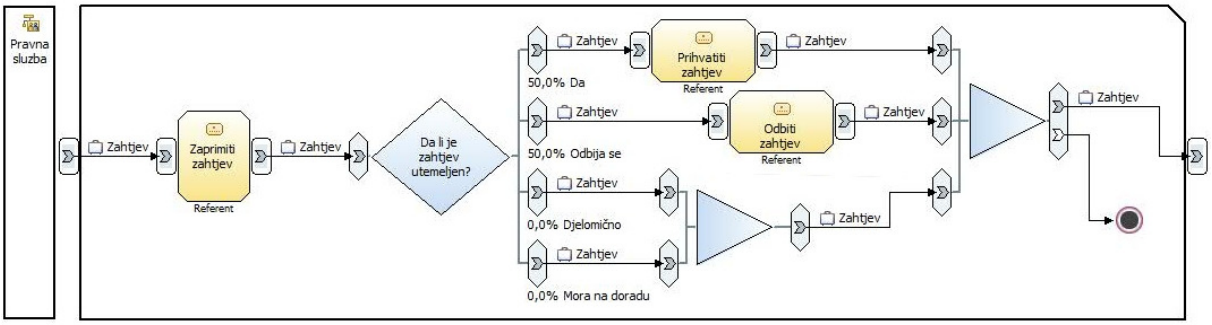
Kontrola toka predstavljaju grupu strukturnih anomalija u koju spadaju grupe anomalija kao što su Redundancija kontrole toka, Nedefinirani uvjeti vrata, Specijalni slučajevi toka i Narušavanje ispravnosti.

Nepotrebna kontrola toka
(eng. *Unnecessary control flow*)

Opis – Anomalija ***Nepotrebna kontrola toka*** (eng. *Unnecessary control flow*) opisuje problem koji nastaje kad postoji više semantički identičnih veza između dva elementa toka, npr. više sekvenci toka idu iz jedne aktivnosti u drugu. S obzirom da ne daju nikakvu dodatnu vrijednost modelu, oni su redundantni. Anomalija ***Nepotrebna kontrola toka*** spada u grupu anomalija ***Kontrola toka*** u podgrupu ***Redundancija kontrole toka***. Ova anomalija prvi puta se spominje se u radu Olkhovich-a pod nazivom *Semi-Automatic Business Process Performance Optimization Based On Redundant Control Flow Detection* [46], kasnije se ista anomalija opisuje i u radu Döhring-a i Heublein-a pod nazivom *Anomalies in Rule-Adapted Workflows - A Taxonomy and Solutions for vBPMN* [16].

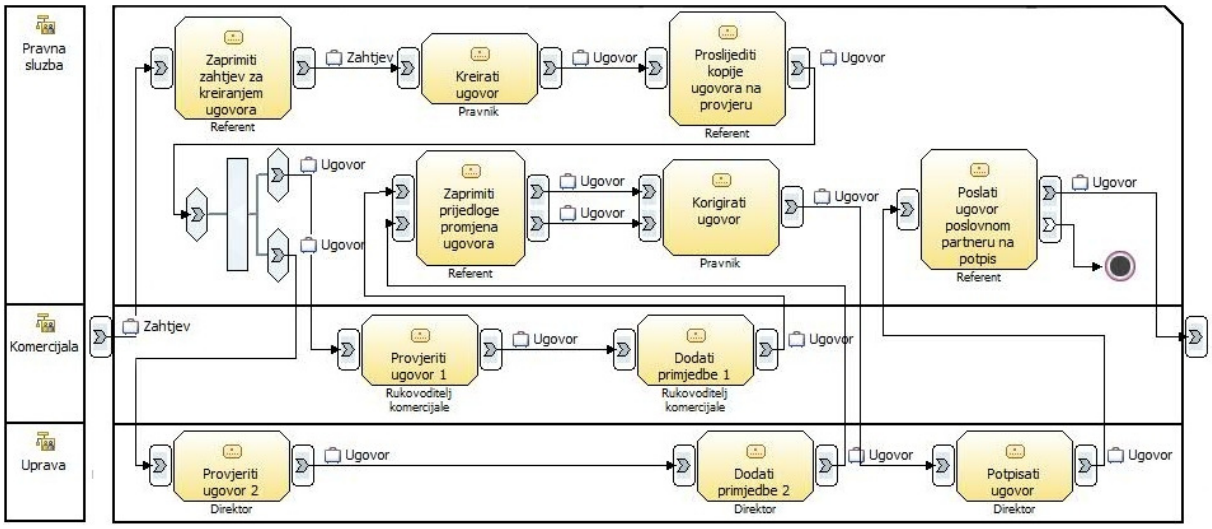


Slika 35. Anomalija ***Nepotrebna kontrola toka***



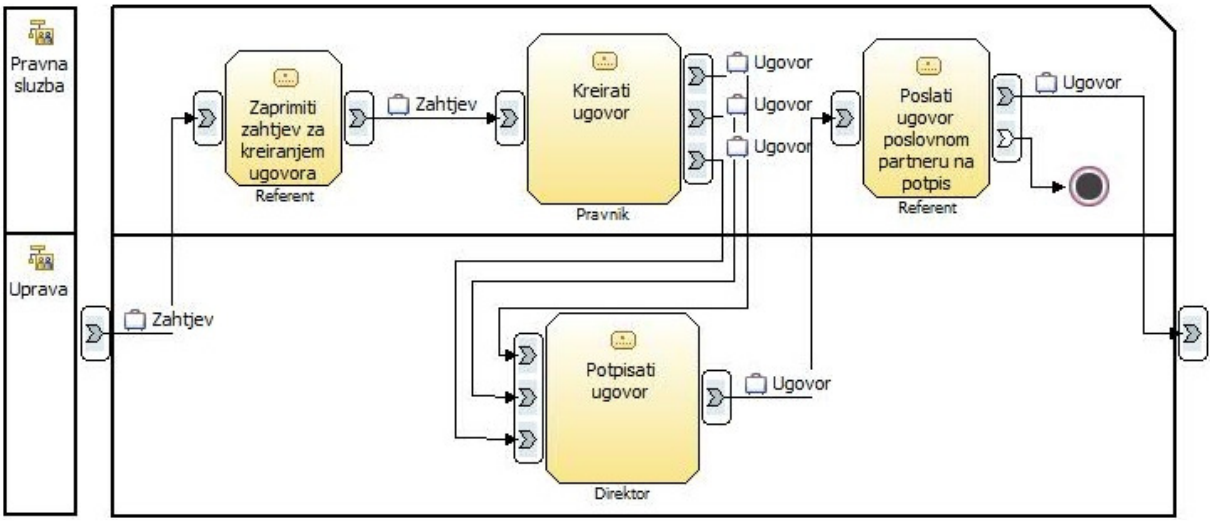
Slika 36. Primjer anomalije ***Nepotrebna kontrole toka***

Primjer – primjeri poslovnih procesa s ovakvom anomalijom nisu rijetke, čak i neki poslovi procesi mogu podržati sličan scenarij. Kao primjer uzet je poslovni proces ***Kreirati ugovor*** u kojem se vidi aktivnost *Zaprimiti prijedloge promjena ugovora*, koja na izlazu ima dvije poslovne stavke koje su iste. U ovome slučaju ne može se govoriti o anomaliji, jer poslovni proces ima definirane aktivnosti koje osiguravaju kreiranje više različitih instanci iste klase dokumenta. Osim toga, ovdje semantički jednake poslovne stavke donose dodatnu vrijednost modelu, a to je u ovome slučaju korekcija ugovora na temelju više različitih prijedloga iz različitih organizacijskih jedinica (Komercijala i Uprava).



Slika 37. Poslovni proces ***Kreirati ugovor***

S druge strane ako se promotri sljedeća slika u kojoj se vidi da iz aktivnosti *Kreirati ugovor* idu tri iste poslovne stavke u sljedeću aktivnost. U ovome slučaju ova redundancija potpuno je nepotrebna i ne pridonosi dodatnu vrijednost modelu. Ovaj slučaj može se nazvati anomalijom.



Slika 38. Poslovni proces ***Kreirati ugovor*** sa anomalijom ***Nepotrebna kontrola toka***

Način prepoznavanja anomalije - kod nailaska na slučaj da dva ili više aktivnosti imaju za svoje izlaze i ulaze isti poslovni stavak, tada obavezno treba provjeriti da li se radi o istim dokumentima.

Pravila koja opisuju anomaliju ***Nepotrebna kontrola toka*** nastaje ako:

- u modelu poslovnog procesa postoji aktivnost koja ima više od jednog izlaza s istom poslovnom stavkom,
- u modelu poslovnog procesa postoji aktivnost koja ima više od jednog ulaza s istom poslovnom stavkom,
- navedene poslovne stavke su po atributima potpuno jednake,
- navedene poslovne stavke imaju istu jedinstvenu oznaku (ID),
- navedene poslovne stavke su nastale u istoj aktivnosti,
- navedene poslovne stavke su nastale u isto vrijeme i
- navedene poslovne stavke su nastale od strane istog resursa.

Ako se navedena pravila opišu u pseudokodu, onda to izgleda ovako:

```
1  RULE: UnnecessaryControlFlow
2  IF: ((Control-Flow EXIST BETWEEN Tasks)
3     AND (COUNT(Control-Flow) > 1))
4     OR  ((Data-Flow EXIST BETWEEN Tasks)
5     AND (COUNT(Identical Data-Flow) > 1))
6  THEN: UnnecessaryControlFlowAnomaly
7  ENDIF
```

Kôd 51. Način prepoznavanja anomalije ***Nepotrebna kontrola toka***

Autori Dohring i Heublein u radu *Anomalies in Rule-Adopted Workflows – A Taxonomy and solutuions for vBPMN* [16] razradili su klasu koja prepoznaje navedenu anomaliju u *eCore/oclInEcore* objektima u *jBoss Drools* verifikatoru pravila.

```
operationarcsBetweenGW(t:FlowNode[1]):SequenceFlow[*]
{body:
 self.arcsDownstreamGW(t)->intersection(t.arcsUpstreamGW(self));
}
```

Kôd 52. Način prepoznavanja anomalije ***Nepotrebna kontrola toka*** u *jBoss Drools* verifikatoru pravila
(Izvor: Dohring, M., Heublein, S.: *Anomalies in Rule-Adopted Workflows – A Taxonomy and solutuions for vBPMN*, 6th European Conference on Software Maintenance and Reengineering, pages 117-126. Szeged, Hungary : IEEE.)

Način rješavanja - u slučaju da postoji više istih veza između pojedinih radnih koraka potrebno je stoga provjeriti da li su oba dokumenta nastala u isto vrijeme i da li imaju isti jedinstveni identifikator. Ukoliko se oba podatka podudaraju, u tom slučaju može se takav scenarij proglasiti anomalijom. Nadalje, ukoliko je jedan od elemenata složena aktivnost, potproces ili servis u tom slučaju potrebno je proanalizirati unutarnju strukturu potprocesa s ciljem otkrivanja da li se radi o redundantnosti poslovne stavke.

U slučaju da je anomalija prepoznata onda je potrebno upozoriti korisnika na moguću anomaliju, te da je potrebno pregledati model i ispraviti ga.

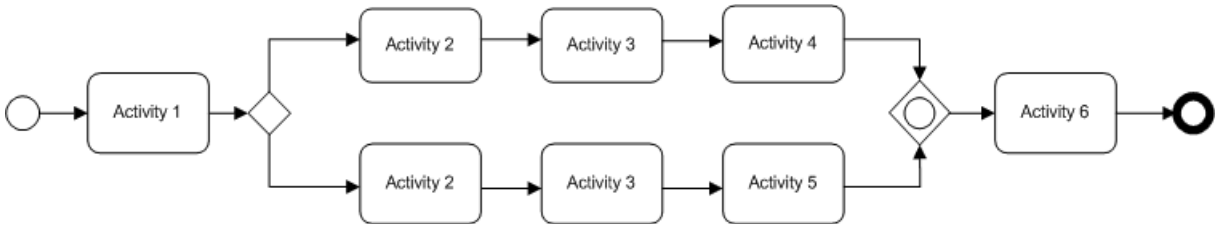
```
FIX: UnnecessaryControlFlowAnomaly()
     strMessage = ”There is unnecessary control anomaly in model”
```

Kôd 53. Način rješavanja anomalije ***Nepotrebna kontrola toka***

Redundancija modela

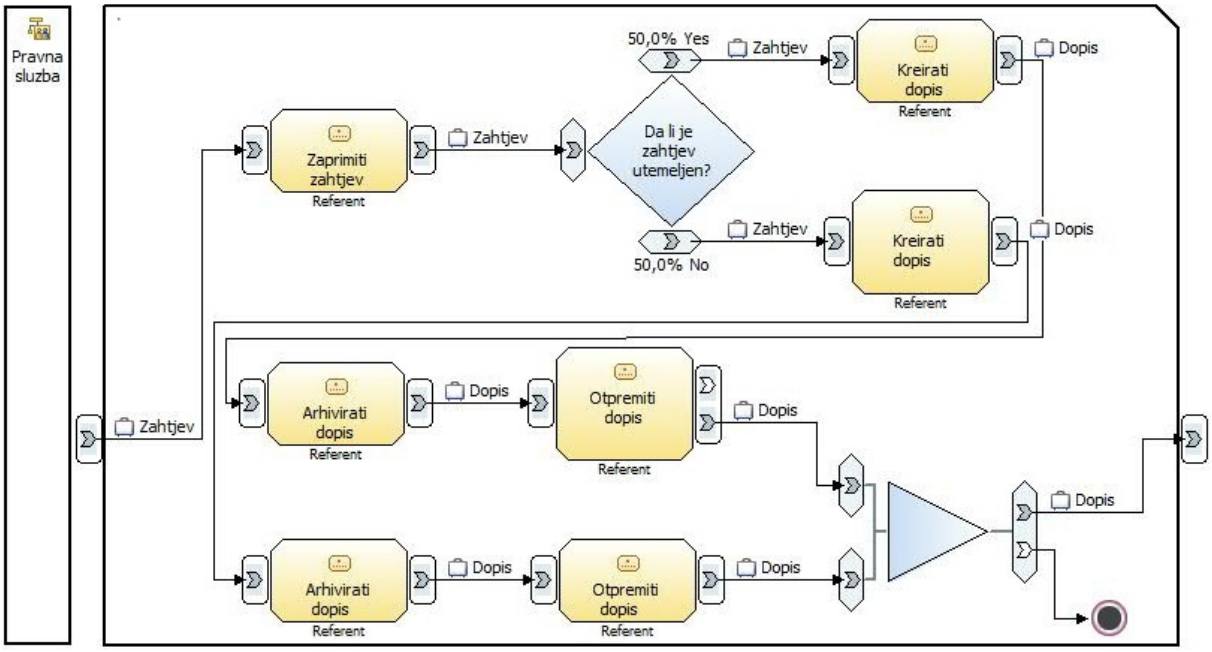
(eng. *Modeling redundancy*)

Opis - anomalija ***Redundancija modela*** česta je kod fleksibilnih notacija poput BPMN. Često se jedna grana od druge razlikuje samo po određenim aktivnostima. Svaka od navedenih grana mora biti definirana zasebno što predstavlja jedan oblik redundancije. Takav oblik redundancije komplicira daljnje održavanje i promjenu toka rada. Anomalija ***Redundancija modela*** spada u grupu anomalija ***Kontrola toka*** u podgrupu ***Redundancija kontrole toka***. Ova anomalija pod tim nazivom prvi puta se spominje u radu Döhring-a i Zimmermann-a, *vBPMN: Event-Aware Work-flow Variants byWeaving BPMN2 and Business Rules* [17], kasnije u razgradnji svoje taksonomije spominje je autori Döhring i Heublein u radu *Anomalies in Rule-Adapted Workflows - A Taxonomy and Solutions for vBPMN* [16].



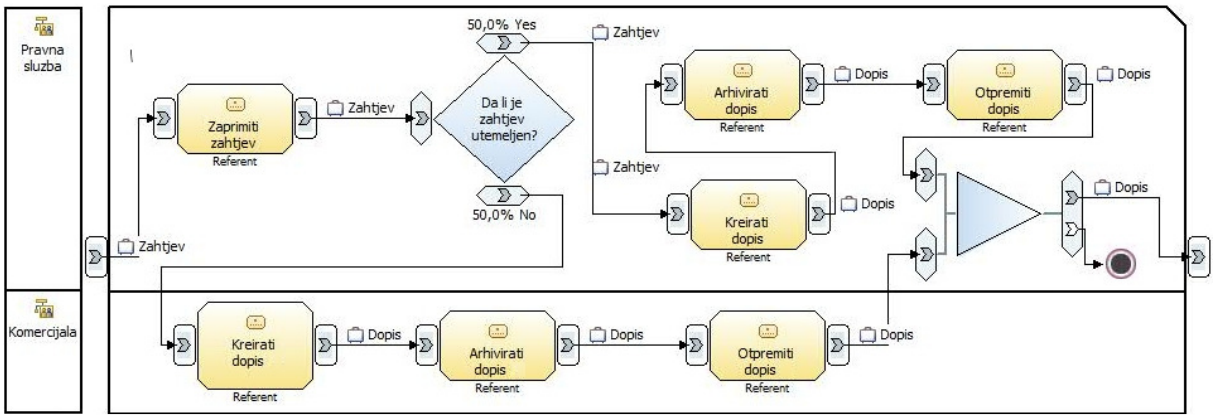
Slika 39. Anomalija ***Redundancija modela***

Primjer - primjer anomalije ***Redundancija modela*** pokazuje slučaj u kojem je vidljiva redundancija modela dakle u modelu postoji sekvenca sa aktivnostima koji nose isti naziv, kojeg obavljaju isti resursi i koji se obavljaju u istoj organizacijskoj jedinici. Model ne bi izgubio na vrijednosti ako se izbace duple aktivnosti i model reducira na jednu sekvencu.



Slika 40. Primjer poslovnog procesa s anomalijom ***Redundancija modela***

S druge strane ako se pogleda sljedeći primjer, koji isto ima dvije sekvence sa aktivnostima istog naziva, međutim kod toga primjera jedan niz aktivnosti odvija se u drugoj organizacijskoj jedinici. Iz tog razloga ne smije se takav primjer proglasiti redundancijom toka.



Slika 41. Primjer poslovnog procesa koji sliči anomaliji ***Redundancija modela***

Način prepoznavanja anomalije – izvodi se pomoću sljedećih pravila:

- u modelu poslovnog procesa u jednoj stazi postoji više istih aktivnosti koji obavljaju isti posao i
- u modelu poslovnog procesa u jednoj stazi postoji niz aktivnosti kojeg obavljaju isti resursi više puta.

Ako se u modelu niz aktivnosti ponavlja više od jednog puta i sve te aktivnosti se nalaze u istoj stazi, onda se to može smatrati redundancijom modela. Ako se pojedini radni koraci odvijaju u drugoj stazi ili se pojedini isti radni koraci odvijaju od strane različitih resursa onda se to više ne može smatrati anomalijom redundancije modela. Ako se navedena pravila prikažu u pseudokodu, onda to izgleda ovako:

```
RULE: ModelingRedundancy
IF: ((COUNT(Identical Task)>1)
AND (
       IF: ((Control-Flow EXIST BETWEEN Tasks)
       AND (COUNT(Control-Flow) > 1))
       OR  ((Data-Flow EXIST BETWEEN Tasks)
       AND (COUNT(Identical Data-Flow) > 1)))
THEN: ModelingRedundancyAnomaly
ENDIF
```

Kôd 54. Način prepoznavanja anomalije ***Redundancija modela***

Način rješavanja anomalije - ako sustav prepozna ovu anomaliju potrebno je upozoriti korisnika na postojanje ove anomalije.

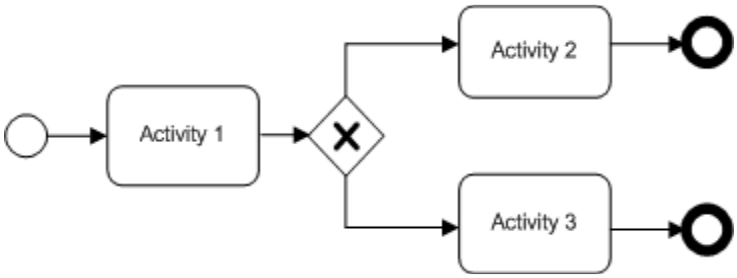
```
FIX: ModelingRedundancyAnomaly ()
     strMessage = ”There is modeling redundancy anomaly in model”
```

Kôd 55. Način rješavanja anomalije ***Redundancija modela***

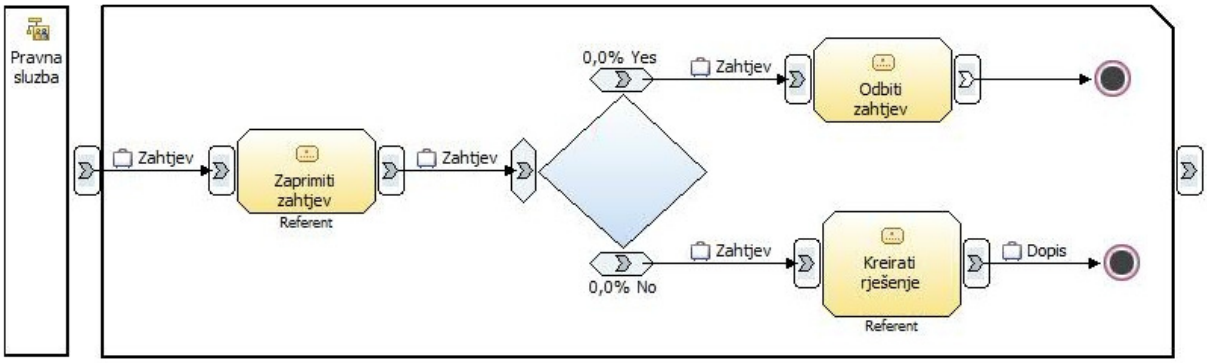
Nedefinirani uvjeti vrata

(eng. *Undefined gateway conditions*)

Opis - anomalija pod nazivom ***Nedefinirani uvjeti vrata*** nastaje, jer je prema BPMN 2.0 standardu moguće definirati vrata bez uvjeta, što u primjeni modela u realnim situacijama može predstavljati veliki problem. Anomalija ***Nedefinirani uvjeti vrata*** u predloženoj taksonomiji dio je grupe anomalija ***Kontrola toka*** i podgrupe ***Nedefinirani uvjeti vrata.*** Ova anomalija prvi puta se spominje u BPMN specifikaciji OMG grupe [10].



Slika 42. Anomalija ***Nedefinirani uvjeti vrata***



Slika 43. Primjer anomalije ***Nedefinirani uvjeti vrata***

Primjer: Na slici pokazan je slučaj u kojem nije definiran uvjet na vratima. Nakon zaprimanja zahtjeva ne postoji odluka na temelju koje se može odrediti da li poslovni proces treba nastaviti prema radnom koraku *Odbiti zahtjev* ili prema koraku *Kreirati rješenje*. Također na vratima nije definirana ni zadana vrijednost što bi modelu omogućilo da se uspješno završi. Ako u vratima postoji zadana vrijednost (eng. *default value*) u tom slučaju takav slučaj ne predstavlja anomaliju ***Nedefinirani uvjeti vrata***.

Način prepoznavanja anomalije– izvodi se pomoću sljedećih pravila:

- u modelu poslovnog procesa postoji barem jedan element odlučivanja,
- u modelu poslovnog procesa postoji element odlučivanja koji nema definiran uvjet i
- u modelu poslovnog procesa ne postoji početna vrijednost uvjeta.

Ako se navedena pravila prikažu pomoću pseudokoda, onda to izgleda ovako:

```
1  RULE: UndefinedGatewayConditions
2  IF: Count(Process.Gateways)>0
3  AND Gateway.Condition is null
4  AND Gateway.DefaultValue is null
5  THEN: UndefinedGatewayConditionsAnomaly
6  ENDIF
```

Kôd 56. Način prepoznavanja anomalije ***Nedefinirani uvjeti vrata***

Način rješavanja anomalije - ako sustav prepozna navedenu anomaliju, korisniku treba dati navedenu informaciju kako bi što ispravio pogrešku u modelu.

No, s ciljem automatizacije rješavanja ove anomalije, preporuča se postavljanje početnih vrijednosti uvjeta vrata, u ovome slučaju 70% Yes i 30% No. Ako se navedeno opiše u pseudokodu onda to izgleda ovako:

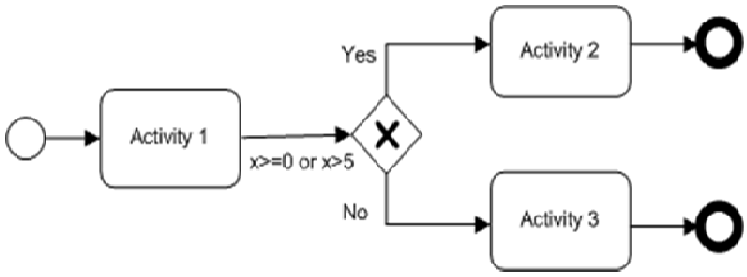
```
1  FIX: UndefinedGatewayConditionsAnomaly
2  IF: Count(Process.Gateways)>0
3  AND Gateway.Condition is null
4  AND Gateway.DefaultValue is null
5  THEN:
6      Gateway.Conditions.Add(Yes, 70%, Default)
7      Gateway.Conditions.Add(No, 30%)
8  ENDIF
```

Kôd 57. Način rješavanja anomalije ***Nedefinirani uvjeti vrata***

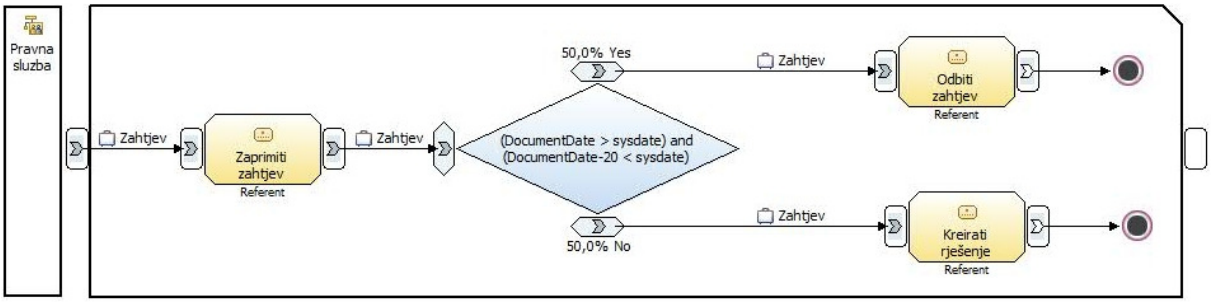
Kontradiktorni uvjeti vrata

(eng. *Contradictive gateway conditions*)

Opis - anomalija ***Kontradiktorni uvjeti vrata*** nastaje kad se uvjeti na vratima djelomično ili potpuno međusobno preklapaju. Posljedica toga je da daljnji tok izvršenja ne ovisi o uvjetima. Anomalija ***Kontradiktorni uvjeti vrata*** u predloženoj taksonomiji dio je grupe anomalija ***Kontrola toka*** i podgrupe ***Nedefinirani uvjeti vrata.*** Ova anomalija spominje se u specifikaciji BPMN notacije OMG grupe [10].



Slika 44. Anomalija ***Kontradiktorni uvjeti vrata***



Slika 45. Primjer anomalije ***Kontradiktorni uvjeti vrata***

Primjer - na slici prikazan je primjer u kojem je navedeno da se provjerava da li je *Zahtjev* datiran datumom koji je veći od današnjeg datuma i da li je veći od današnjeg datuma plus 5 dana. Oba uvjeta su istinita, pa sukladno tome taj uvjet je nepotreban i daljnji tok poslovnog procesa uopće ne ovisi o uvjetima vrata.

Način prepoznavanja anomalije – izvodi se pomoću sljedećeg pravila:

- u modelu poslovnog procesa postoji barem jedna vrata i
- u modelu poslovnog procesa svi uvjeti navedenih vrata odvojeni su veznikom AND uvijek su istiniti ili neistiniti.

Ako se navedeno pravilo opiše u pseudokodu onda to izgleda ovako:

```
RULE: ContradictiveGetewayConditions
FOR EACH Process.Gateway IN Process.Gateways
   intConditionCountYes = 0
   intConditionCountNo = 0
  FOR EACH Process.Gateway.Condition IN Process.Gateway.Conditions
      IF: Process.Gateway.Condition IS TRUE
      THEN
        intConditionCountYes = + intConditionCountYes
      ELSE
        intConditionCountNo = + intConditionCountNo
    ENDIF
  NEXT
  IF: intConditionCountYes = Process.Gateways.Count() OR
     intConditionCountNo = Process.Gateways.Count()
  THEN
     ContradictiveGatewayConditionsAnomaly(Process.Gateway.Name)
  ENDIF
NEXT
```

Kôd 58. Način prepoznavanja anomalije ***Kontradiktorni uvjeti vrata***

Način rješavanja anomalije - u slučaju da sustav prepozna postojanje ovakve anomalije, korisniku valja predočiti element uvjeta koji ima kontradiktorni uvjet kako bi korisnik mogao ispraviti uvjet.

```
FIX: ContradictiveGatewayConditionsAnomaly(strElement as string)
     strMessage = "Anomaly contradictive gateway conditions is detected in model in
                    element" + strElement
```

Kôd 59. Način rješavanja anomalije ***Kontradiktorni uvjeti vrata***

Nedostatak sinkronizacije

(eng. *Lack of synchronizaion*)

Nedostatak sinkronizacije je anomalija kontrola toka koja nastaje kad se struktura toka nenamjerno izvrši nekoliko puta [53]. Ovu anomaliju prepoznalo je više autora, pa je spominju autori Laue, Award u radu *Visualization of Business Process Modeling Anti Patterns* [32], osim u tome radu ovu anomaliju spominje Olkhovich u radu *Semi-Automatic Business Process Performance Optimization Based On Redundant Control Flow Detection* [46], a spominju je i autori Sadiq i Orlowska u radu *Applying graph reduction techniques for identifying structural conflicts in process models* [52].

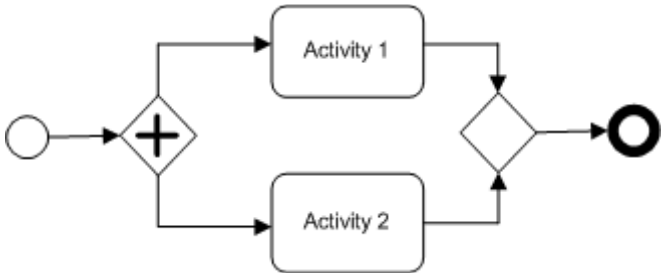
Nedostatak sinkronizacije je nenamjerna višestruka aktivacija aktivnosti koji slijede nakon XILI (eng. *XOR*) vrata spajanja. Nedostatak sinkronizacije može se podijeliti na dva temeljna tipa anomalija i to:

- deterministički nedostatak sinkronizacije i
- nedeterministički nedostatak sinkronizacije.

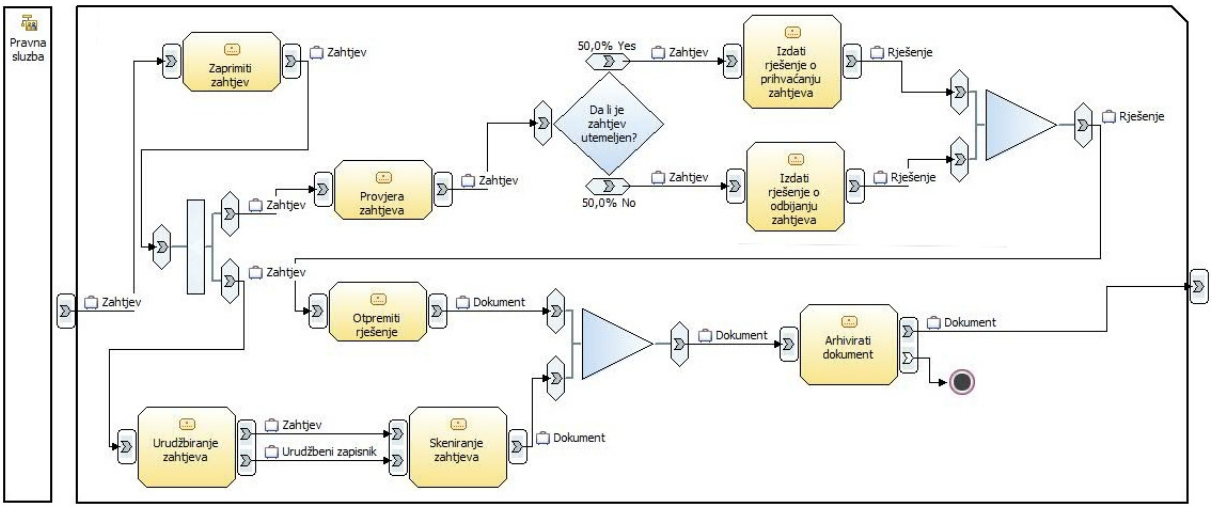
Deterministički nedostatak sinkronizacije

(eng. *Deterministic lack of synchronization*)

Opis - anomalija ***Deterministički nedostatak sinkronizacije*** definirana je kao jedna od konkurentnih aktivnosti koji su spojeni s XILI (eng. *XOR*) vratima spajanja i koji uvijek rezultira sa nenamjernim višestrukim izvođenjem [29]. Ova anomalija ovisi o kontekstu primjene navedene strukture, jer u nekim scenarijima se može pojaviti, a u nekima ne.



Slika 46. Anomalija ***Deterministički nedostatak sinkronizacije***



Slika 47. Primjer anomalije ***Deterministički nedostatak sinkronizacije***

Primjer - u ovome primjeru vidi se da elementi spajanja predstavljaju uzrok nastajanja više instanci radnih koraka i potprocesa, razlog tome je što element razdvajanja emitira tokene na sve svoje izlaze grane, dok element spajanja čeka samo jedan ulaz. To znači da svaka grana elementa spajanja koja dobije tokena na ulazu aktivira daljnje aktivnosti u modelu. U literaturi se ovaj slučaj naziva ***Deterministički nedostatak sinkronizacije***, jer izvršenje jednog segmenta procesa prethodi element spajanja, čime se dobije izvršenje do sljedećeg elementa spajanja.

Način prepoznavanja anomalije - ovakvo ponašanje se može namjerno izmodelirati, no obično je u pitanju greška tijekom modeliranja. Ovakva anomalija nije uvijek pogreška za razliku od recimo anomalije zastoja (eng. *Deadlock*).

Pravila pomoću kojih se može prepoznati navedena anomalija su:

- u modelu poslovnog procesa postoje elementi razdvajanja i XOR element spajanja i
- u modelu poslovnog procesa postoji element razdvajanja prethodi XOR element spajanja.

Ako se navedeno pravilo prikaže u pseudokodu, onda to izgleda ovako:

```
1  RULE: DeterministicLackOfSynchronization
2  IF: (((COUNT(ANDFork) ≥ 1 AND COUNT(XORMerge) ≥ 1)
3  AND (ANDFork PRECEDE XORMerge)
4  THEN: DeterministicLackOfSynchronizationAnomaly
5  ENDIF
```

Kôd 60. Način prepoznavanja anomalije ***Deterministički nedostatak sinkronizacije***

Autori Dohring i Heublein u radu *Anomalies in Rule-Adopted Workflows – A Taxonomy and solutuions for vBPMN* [16] razradili su klasu koja prepoznaje navedenu anomaliju u *eCore/oclInEcore* objektima u *jBoss Drools* verifikatoru pravila.

```
1  class ParallelGateway extends Gateway
2  operation hasMultipleExecutionPartner(): Boolean [1]
3  {body :
4  if self.gatewayDirection.oclAsType(ecore::EEnumLiteral).value>1
5      gatewayDirection: 1=converging; 2=diverging ; 3=mixed
6  then
7    let successorNodes : Set (FlowNode) = self.arcsDownstream(self).targetRef->
8    collect(oclAsType(bpmn2::FlowNode))->asSet() in
9    successorNodes->exists(x|x.oclIsKindOf(bpmn2::ExclusiveGateway) and
10   (x.oclAsType(bpmn2::ExclusiveGateway).gatewayDirection.oclAsType(ecore
11   ::EEnumLiteral).value=1 or x.oclAsType(bpmn2::ExclusiveGateway).
12   gatewayDirection.oclAsType(ecore::EEnumLiteral).value=3) and
13   self.StartIsSesePairWithExit(x))
14 else
15   false
16 endif;
17 }
```

Kôd 61. Način rješavanja anomalije ***Deterministički nedostatak sinkronizacije***
(Izvor: Dohring, M., Heublein, S.: *Anomalies in Rule-Adopted Workflows – A Taxonomy and solutuions for vBPMN*, 6th European Conference on Software Maintenance and Reengineering, pages 117-126. Szeged, Hungary : IEEE.)

Način rješavanja anomalije - u slučaju da model sadrži anomaliju ***Deterministički nedostatak sinkronizacije*** korisnika o tome treba obavijestiti, kako bi mogao ispraviti model.

```
1  FIX: DeterministicLackOfSynchronizationAnomaly
2       strMessage = "Anomaly Deterministic Lack of synchronisaton is detected in model"
```

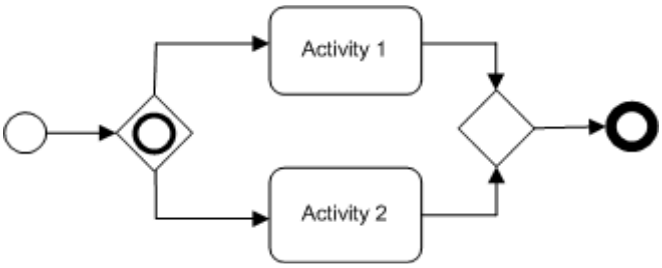
Kôd 62. Način rješavanja anomalije ***Deterministički nedostatak sinkronizacije***

Nedeterministički nedostatak sinkronizacije

(eng. *Undeterministic lack of synchronization*)

Opis - anomalija ***Nedeterministički nedostatak sinkronizacije*** definirana je XILI (eng. *XOR*) vratima spajanja u kojoj jedna od konkurentnih aktivnosti može rezultirati nenamjernim višestrukim izvođenjem [29].

Primjer – prikazuje najjednostavniji oblik anomalije ***Nedeterministički nedostatak sinkronizacije***.



Slika 48. Anomalija ***Nedeterministički nedostatak sinkronizacije***

Način prepoznavanja anomalije – temelje se na pravilima pomoću kojih se može prepoznati navedena anomalija, a to su:

- u modelu poslovnog procesa ima barem jedan element OR razdvajanja,
- u modelu poslovnog procesa ima barem jedan element XOR spajanja i
- u modelu poslovnog procesa OR element razdvajanja prethodi XOR elementu spajanja.

Ako se navedena pravila prikažu pseudokodom, onda to izgleda ovako:

```
1  RULE: UnDeterministicLackOfSynchronization
2  IF: (((COUNT(ORFork) ≥ 1 AND COUNT(XORMerge) ≥ 1)
3  AND (ORFork PRECEDE XORMerge)
4  THEN: UnDeterministicLackOfSynchronizationAnomaly
5  ENDIF
```

Kôd 63. Način prepoznavanja anomalije ***Nedeterministički nedostatak sinkronizacije***

Način rješavanja anomalije - u slučaju da sustav naleti na anomaliju ***Nedeterministički nedostatak sinkronizacije*** korisniku treba dati informaciju o postojanju anomalije kako bi je ispravio u modelu.

```
1  FIX: UnDeterministicLackOfSynchronizationAnomaly
2       strMessage = "Anomaly Cyclic Lack of synchronisaton is detected in model"
```

Kôd 64. Način rješavanja anomalije ***Nedeterministički nedostatak sinkronizacije***

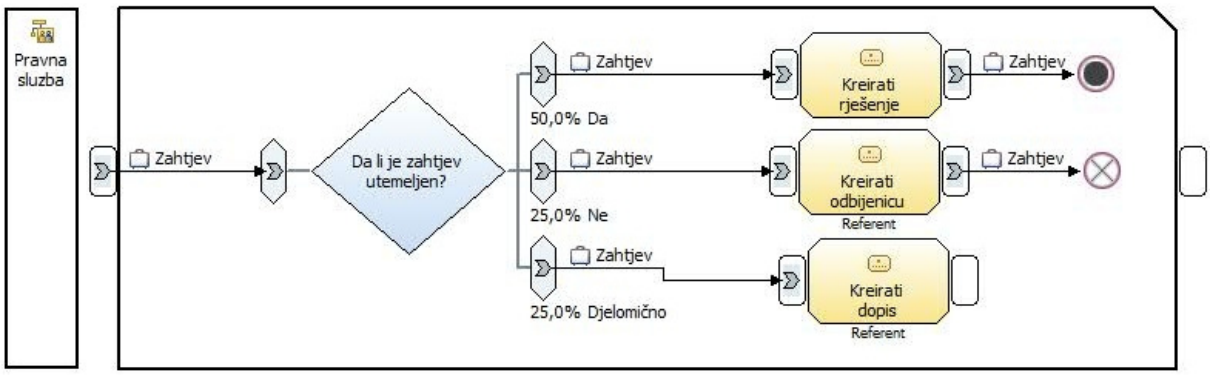
Viseća stanja

(eng. *Dangling states*)

Opis - anomalija ***Viseća stanja*** nastaje kad nije moguće razlučiti aktiviranu od započete aktivnosti. To često dovodi do gubitka rada, jer se time ili onemogućava brisanje aktiviranih aktivnosti ili onemogućava brisanje započetih aktivnosti [50].

Slika 49. Anomalija ***Viseća stanja***

Primjer - prikazuje model koji može doći u viseće stanje ako vrata uvjeta usmjere tok na drugu ili treću granu. U tom slučaju model poslovnog procesa ne završava već ostaje u visećem stanju. Ako u modelu postoje pojedine sekvence za koje ne postoji završni događaj, u tom slučaju postoji mogućnost da model ostane u visećem stanju. U primjeru se vidi da ako se tok preusmjeri u drugu (Ne) ili treću (Djelomično) granu, tada model ostane u visećem stanju i nikad ne završi. Ako se takav model kao potproces uključi u viši proces onda će takav model biti u pojedinim slučajevima neispravan.



Slika 50. Primjer anomalije ***Viseća stanja***

Pravila prepoznavanja – navedena anomalija u modelu poslovnog procesa može se prepoznati na temelju sljedećih pravila:

- u modelu poslovnog procesa postoji element koji ima izlaz iz elementa koji nije spojen na drugi element niti ne izlazi iz modela, niti ne završava završnim elementom i
- u modelu poslovnog procesa postoji element koji ima ulaz u element koji nije spojen na drugi element niti ne predstavlja ulaz u model.

Ako se navedena pravila napišu u pseudokodu, to onda izgleda ovako:

```
RULE: DanglingInputAndOutput
IF: ((InputVar DEFINED)
AND (COUNT(Connection(InputVar))=0))
OR ((OutputVar DEFINED)
AND (COUNT(Connection(OutputVar))= 0))
THEN: DanglingInputAndOutputAnomaly
ENDIF
```

Kôd 65. Način prepoznavanja anomalije ***Viseća stanja***

Način rješavanja anomalije - u slučaju da je navedena anomalija prepoznata u modelu poslovnog procesa, korisnika je potrebno obavijestiti o njenom postojanju kako bi korisnik mogao ispraviti model.

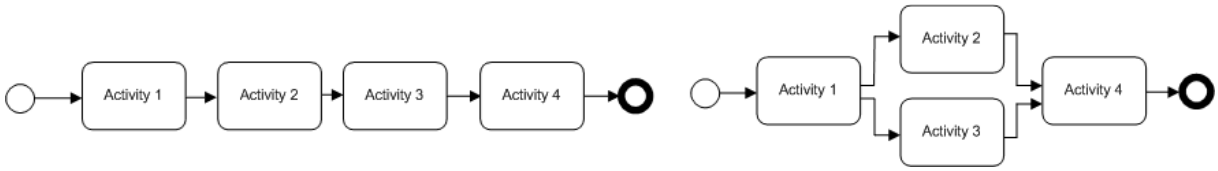
```
FIX: DanglingInputAndOutputAnomaly
     strMessage = "Anomaly Dangling States is detected in model"
```

Kôd 66. Način rješavanja anomalije ***Viseća stanja***

Netolerancija na promjene poretka

(eng. *No tolerance for change of order*)

Opis - anomalija pod nazivom ***Netolerancija na promjene poretka*** odnosi se na ispravnu zamjenu poretka primjene operacija kao što je paralelizam, slijednost ili zamjena aktivnosti [50]. Ako postoje dvije aktivnosti koje slijede jedna drugu, no obavljaju je različiti resursi i međusobno nisu zavisni, ne postoji ni jedan razlog zašto se navedene aktivnosti ne mogu odvijati paralelno.



Slika 51. Anomalija ***Netolerancija na promjene poretka***

Pravila prepoznavanja – navedena anomalija prepoznaje se na temelju sljedećih pravila:

- u modelu poslovnog procesa postoje minimalno dvije aktivnosti koje slijede jedna drugu,
- navedene aktivnosti imaju određeno vrijeme trajanja,
- navedene aktivnosti obavljaju različiti međusobno nezavisni resursi i
- u modelu poslovnog procesa poslovne stavke navedenih aktivnosti su jedinstvene.

Ako se navedeno pravilo prikaže u pseudokodu, onda bi to izgleda ovako:

```
RULE: NoToleranceOnChangeOfOrder:
IF: Count(Activites)> 1
  AND Activity PRECEDE Activity
  AND Activity.Duration > 0
  AND IsUnique(Activity.BusinessItems)=False
THEN:
  NoToleranceOnChangeOfOrderAnomaly()
ENDIF
```

Kôd 67. Način prepoznavanja anomalije ***Netolerancija na promjene poretka***

Način rješavanja anomalije - s obzirom da se u ovome slučaju radi o poprilično zahtjevnoj promjeni strukture poslovnog procesa, korisnik mora ručno intervenirati u modelu kako bi se navedena anomalija ispravila. Dakle, ovome slučaju korisnika treba obavijestiti da se anomalija nalazi u modelu i da ispravi model.

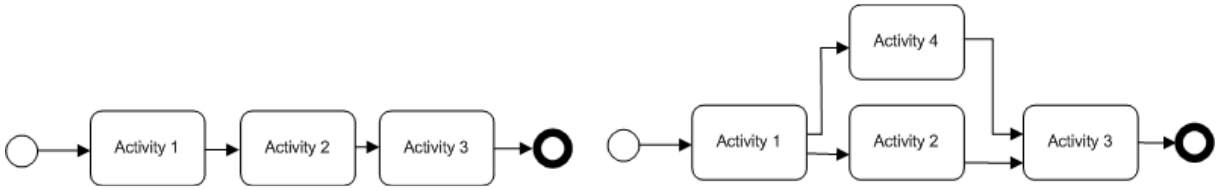
```
1  FIX: NoToleranceOnChangeOfOrderAnomaly
2       strMessage = "Anomaly No tolerance od change of order is detected in model"
```

Kôd 68. Način rješavanja anomalije ***Netolerancija na promjene poretka***

Nekonzistentnost kod inserta paralelnih grana

(eng. *Inconsistency when inserting parallel branches*)

Opis - anomalija pod nazivom ***Nekonzistentnost kod inserta paralelnih grana*** nastaje kod inserta nove paralelne grane. Anomalije ***Nekonzistentnost kod inserta paralelnih grana*** i anomalije ***Netolerancije na promjene poretka*** vezani su uz dinamičke promjene modela [50]. Navedene anomalije dio su grupe ***Specijalnih slučajeva anomalija toka*** unutar grupe anomalija ***Kontrola toka***.



Slika 52. Anomalija ***Nekonzistentnost kod inserta paralelnih grana***

Pravila prepoznavanja – navedena anomalija prepoznaje se na temelju sljedećih pravila koja su vrlo slična anomaliji ***Netolerancija na promjene poretka***, no u obrnutom smislu:

- u modelu poslovnog procesa postoje minimalno dvije aktivnosti koje se odvijaju paralelno,
- u modelu poslovnog procesa navedene aktivnosti imaju određeno vrijeme trajanja i
- u modelu poslovnog procesa navedene aktivnosti obavljaju isti resursi.

Ako se navedeno pravilo prikaže u pseudokodu, onda to izgleda ovako:

```
RULE: InconsistenclesWhenInsertingParallelBraches:
IF: Count(Fork) > 0
   AND Count(Merge) > 0
   AND Count(Activites)> 1
   AND Activity.Duration > 0
   AND IsUnique(Activity.Resources)=False
THEN:
    InconsistenclesWhenInsertingParallelBrachesAnomaly()
ENDIF
```

Kôd 69. Način prepoznavanja anomalije ***Nekonzistentnost kod inserta paralelnih grana***

Način rješavanja anomalije - s obzirom da se u ovome slučaju radi o poprilično zahtjevnoj promjeni strukture poslovnog procesa, korisniku treba izlistati sve aktivnosti koje su paralelne i koje zbog navedenih zavisnosti predstavljaju usko grlo poslovnog procesa.

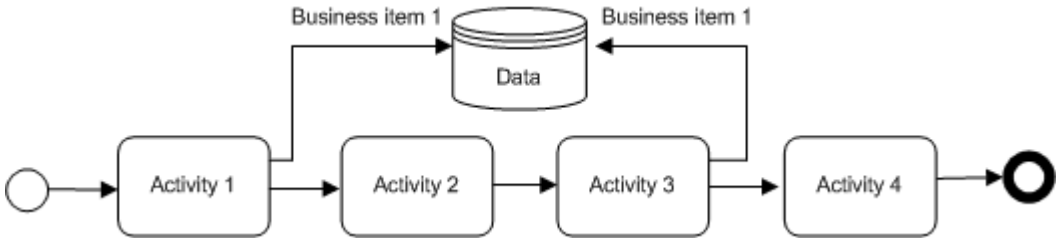
```
FIX: InconsistenclesWhenInsertingParallelBrachesAnomaly
     strMessage = "Anomaly Inconsistent Change Of Past is detected in model"
```

Kôd 70. Način rješavanja anomalije ***Nekonzistentnost kod inserta paralelnih grana***

Nekonzistentnost promjene iz prošlosti

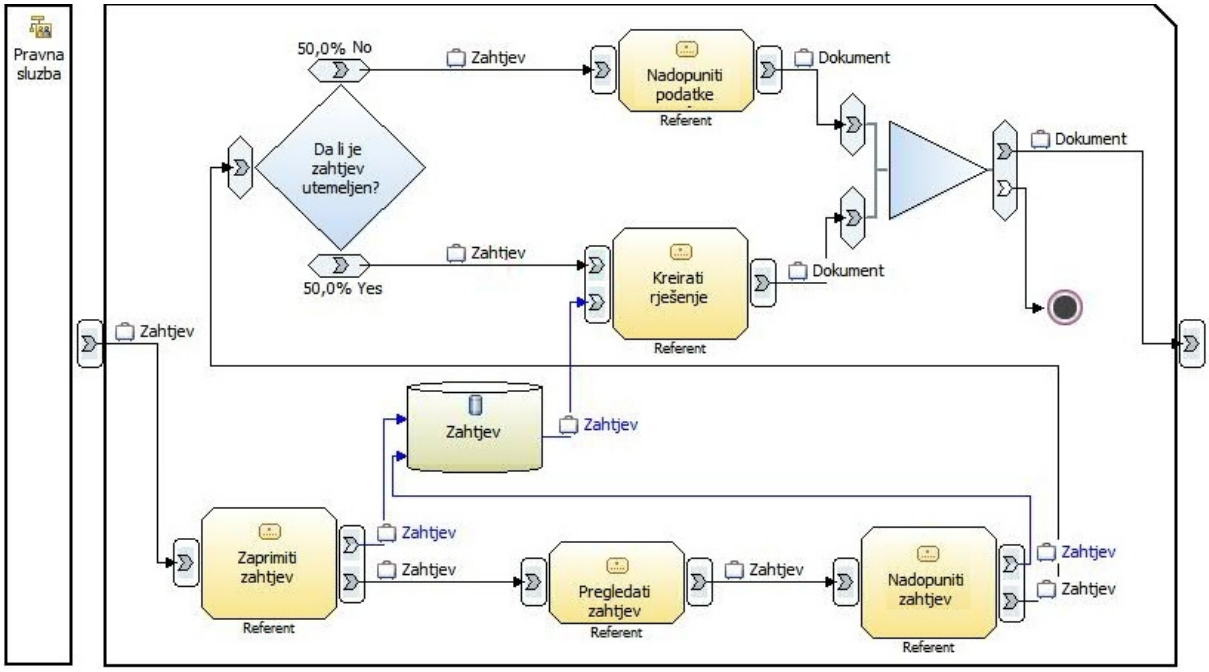
(eng. *Inconsistent change of past*)

Opis - anomalija se odnosi na pravilo «Ne mijenjaj prošlost instance». Ignoriranje ovog pravila može dovesti do nekonzistentnosti stanja instanci ili nestalih ulaznih podataka za naredne aktivnosti koje treba izvršiti [57]. Navedena anomalija dio su grupe ***Specijalnih slučajeva anomalija toka*** unutar grupe anomalija ***Kontrola toka***.



Slika 53. Anomalija ***Nekonzistentnost promjene iz prošlosti***

Primjer - ova anomalija se događa u slučajevima ako se u pojedinom poslovnom procesu u jednom radnom koraku zapisuju podaci u neku tablicu u bazi podataka, dok se u narednim aktivnostima čitaju podaci iz te iste tablice baze podataka. Ovakav princip rada može predstavljati problem ako prvi radni korak mijenja podatak bitan za daljnji tijek procesa. Recimo da se u prvoj aktivnosti ažurira status dokumenta u primjerice *Zatvoreno*, dok se u sljedećoj aktivnosti uzima taj dokument na daljnju obradu, iako dokument ima status *Zatvoreno*.



Slika 54. Primjer anomalije ***Nekonzistentnost promjene iz prošlosti***

Postoji bezbroj poslovnih procesa koji u jednom radnom koraku bilježe određeni podatak u bazu, dok već u drugome mijenjaju taj isti podatak, problem nastaje ako daljnji tijek poslovnog procesa ovisi o statusu pojedinog sloga u bazi podataka. Ovakva anomalija treba se rješavati poslovnim pravilom gdje se određuje koji radni korak ima pravo mijenjati status pojedinog dokumenta.

Način prepoznavanja anomalije – temelje se na sljedećim pravilima:

- u modelu poslovnog procesa postoji ima barem jedan element baze podataka,
- u modelu poslovnog procesa u navedenu bazu podataka u jednom koraku se piše, a u drugom se čita isti slog i
- u modelu poslovnog procesa navedena aktivnost koja čita slijedi aktivnost koja zapisuje slog.

Ako se navedeno pravilo napiše u pseudokodu, onda to izgleda ovako:

```
1  RULE InconsistentChangeOfPast:
2  IF:  Count(Database.Input)>0
3  AND Count(Database.Output)>0
4  AND Activity.Write PRECEDE Activity.Read
5  THEN:
6     InconsistentChangeOfPastAnomaly()
7  ENDIF
```

Kôd 71. Način prepoznavanja anomalije ***Nekonzistentnost promjene iz prošlosti***

Način rješavanja anomalije - u slučaju da sustav prepozna anomaliju ***Nekonzistentnost promjene iz prošlosti*** korisnika treba obavijestiti da je navedena anomalija pronađena u modelu poslovnog procesa i da model treba ispraviti.

```
1  FIX: InconsistentChangeOfPastAnomaly()
2       strMessage = "Anomaly Inconsistencles when inserting parallel branches is
3                     detected in model"
```

Kôd 72. Način rješavanja anomalije ***Nekonzistentnost promjene iz prošlosti***

Nedostatak tolerancije u petlji

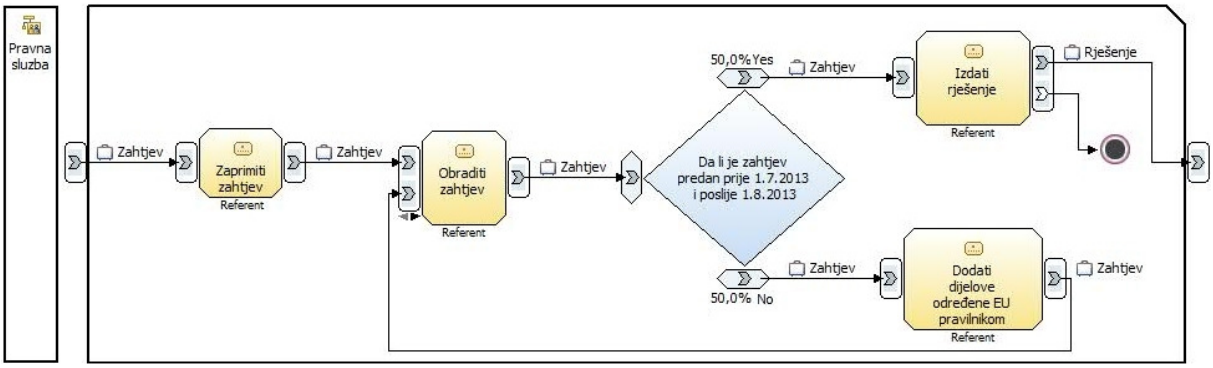
(eng. *Missing loop tolerance*)

Opis – anomalija ***Nedostatak tolerancije u petlji*** predstavlja specijalni oblik anomalije, koja nastaje zbog definicije uvjeta kojim se izlazi iz petlje (npr. do until 1=2;). Ova anomalija prvi puta se spominje u radu autora Rinderle-a, Reichert-a i Dadam-a pod nazivom *Correctness Criteria for Dynamic Changes inWorkflow Systems: A Survey* [50].

Anomalija ***Nedostatak tolerancije u petlji*** predstavlja problem koji se odnosi na korektno i razumno određivanje promjena u strukturi petlje. Konkretno, u pojedinim slučajevima nije potrebno isključiti instance promjena temeljenu na činjenici odgovarajućih izmjena u petlji [57].

Slika 55. Anomalija ***Nedostatak tolerancije petlji***

Primjer - kao što je navedeno u definiciji ove anomalije, primjer mora u petlji imati uvjet koji zbog svoje definicije ne može izaći iz petlje. U prikazanom primjeru vidi se scenarij u kojem poslovna stavka pod nazivom *Zahtjev* ulazi u element odluke u kojem je uvjet *Da li je zahtjev predan prije 1.7.2013. i poslije 1.8.2013.?* S obzirom da ne postoji takav *Zahtjev* nastaje petlja iz koje proces ne može izaći.



Slika 56. Primjer anomalije ***Nedostatak tolerancije u petlji***

Pravila prepoznavanja - pravila za otkrivanje ovakve anomalije izgledala bi ovako:

- u modelu poslovnog procesa poslije elementa odluke postoji povratna veza i
- u modelu poslovnog procesa postoji uvjet u elementu odluke koji ima vrlo malu toleranciju zadovoljenja.

Ako se navedeno pravilo prikaže pomoću pseudokoda, onda to izgleda ovako:

```
RULE: MissingLoopTolerance
IF: ((Activity PRECEDE Gateway)
AND ((Gateway BACKCONNECT Activity)
AND (Gateway.Output.Percentage > 90))
THEN: MissingLoopToleranceAnomaly
ENDIF
```

Kôd 73. Način prepoznavanja anomalije ***Nedostatak tolerancije u petlji***

Način rješavanja anomalije - u slučaju da se u modelu otkrije nedostatak tolerancije u petlji, tada korisniku treba prikazati koji element odluke sadrži nedostatak tolerancije u petlji.

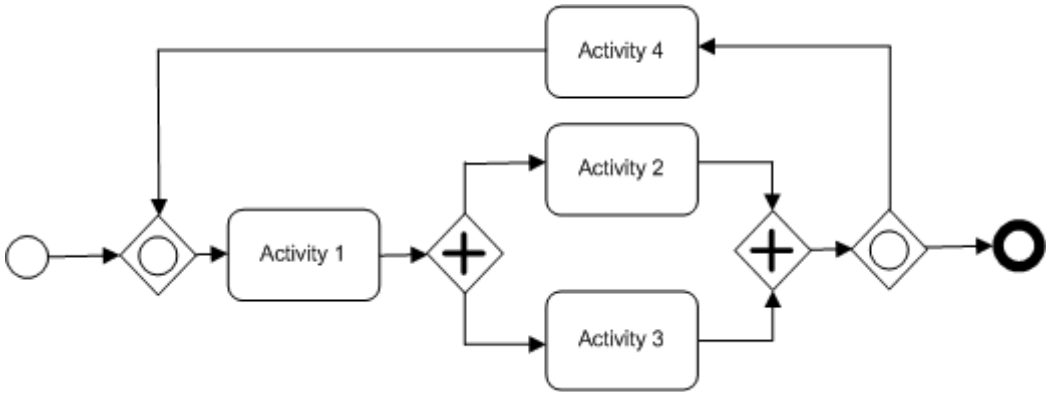
```
FIX: MissingLoopToleranceAnomaly()
     strMessage = "Anomaly Missing Loop Tolerance Anomaly is detected in model"
```

Kôd 74. Način rješavanja anomalije ***Nedostatak tolerancije u petlji***

Zapinjanje toka

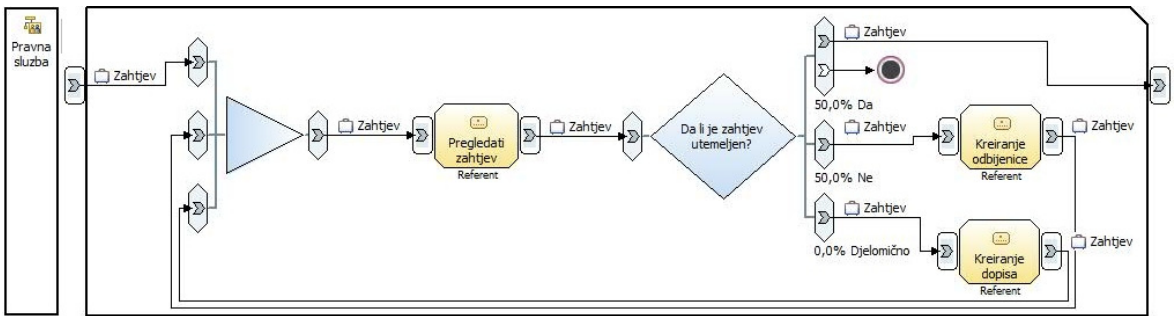
(eng. *Livelock flow*)

Opis - anomalija ***Zapinjanje toka*** obično se naziva beskonačna petlja. To se događa zbog loše modeliranog uvjeta petlje koji onemogućava izlaz iz petlje. Navedenu anomaliju u modelima poslovnih procesa prvi su opisali Laue i Awad u radu *Visualization of Business Process Modeling Anti Patterns* [32]. Anomalija ***Zapinjanje toka*** spada u grupu anomalija ***Kontrole toka*** i u podgrupu ***Narušavanje ispravnosti***.



Slika 57. Anomalija ***Zapinjanje toka***

Primjer - na primjeru se vidi model poslovnog procesa u kojem postoji element spajanja uparen s elementom odluke. Ako u elementu odluke tok krene u bilo kojem drugom smjeru osim da zadovoljava, u tom slučaju kreira se beskonačna petlja.



Slika 58. Primjer anomalije ***Zapinjanje toka***

Pravila prepoznavanja – navedena anomalija može se prepoznati na temelju sljedećih pravila:

- u modelu poslovnog procesa postoji minimalno dva elementa OR spajanja i
- u modelu poslovnog procesa jedan element OR spajanja slijedi drugi element OR spajanja,

- u modelu poslovnog procesa drugi element OR spajanja ima povratnu vezu s prvim elementom spajanja,
- u modelu poslovnog procesa postoji minimalno dva elementa AND spajanja,
- u modelu poslovnog procesa element OR spajanja slijedi element AND spajanja i
- u modelu poslovnog procesa jedan element AND spajanja slijedi drugi element AND spajanja.

Ako se navedeno pravilo prikaže u pseudokodu, onda to izgleda ovako:

```
1  RULE: LivelockFlow
2  IF: Count(ORMegre)=>2
3  AND Count(ANDGateway)=>2
4  AND ORMegre(i-1) Precede ORMerge(i)
5  AND ORMerge(i) BackConnect ORMerge(i-1)
6  AND ANDGateway(i-1) Precede ANDGateway(i)
7  AND ORMerge(i) Precede ANDGateway(i)
8  THEN: LivelockFlowAnomaly
9  ENDIF
```

Kôd 75. Način prepoznavanja anomalije ***Zapinjanje toka***

Način rješavanja ovakve anomalije obično zahtjeva kreiranje mehanizma koji neće dozvoliti da se petlja izvrši više od određenog broja puta. Znači dodati varijablu *i* koja će se povećati svaki puta kad se petlja izvede.

```
Npr: Do until i > 5
      ...
      loop
```

U slučaju pronalaska ovakve anomalije u modelu poslovnih procesa, korisnika valja obavijestiti o anomaliji kako bi mogao ispraviti model.

```
1  FIX: LivelockFlowAnomaly
2       strMessage = "Anomaly livelock flow is detected in model"
```

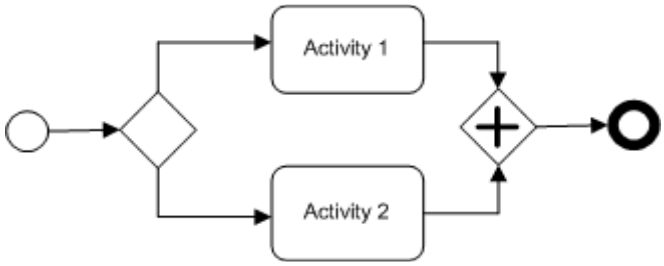
Kôd 76. Način rješavanja anomalije ***Zapinjanje toka***

Zastoj toka

(eng. *Deadlock flow*)

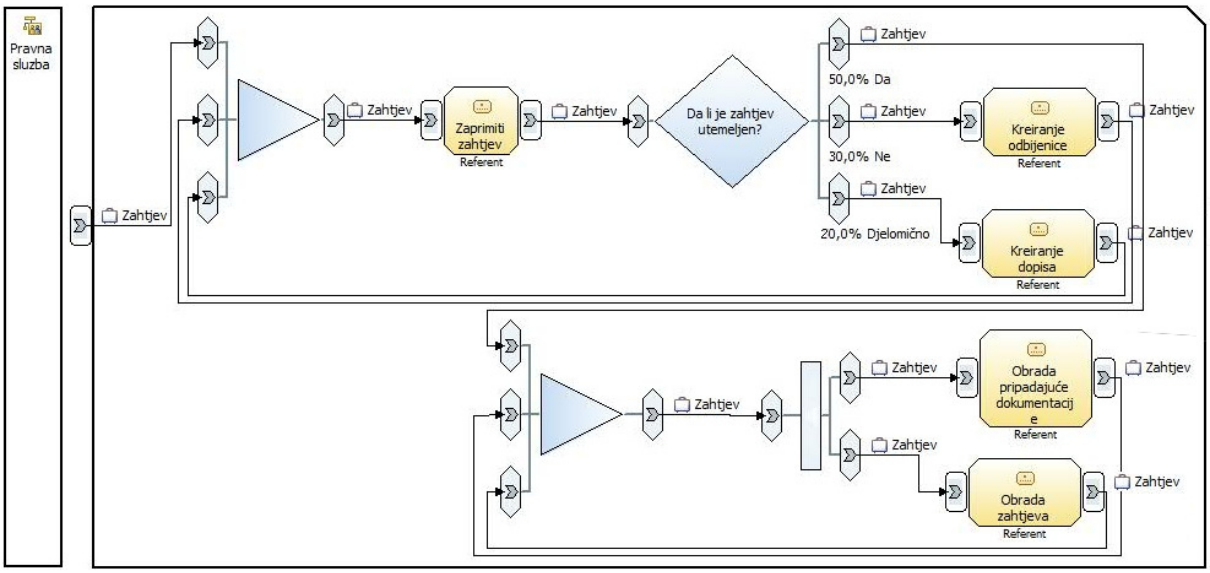
Opis - anomalija ***Zastoj toka*** opisuje situaciju gdje je tok sekvence zapeo na svojoj putanji i ne može biti završen. Zastoj toka se može dogoditi na primjer kad nije moguće aktivirati sve izlazne sekvence toka koji su potrebni za prolaz kroz paralelna vrata. U tom slučaju tok sekvence ne može biti nastavljen iza tih vrata. Anomalija ***Zastoj toka*** spominje se u cijelom nizu radova u kojima je detaljno opisana i razrađena, počevši od autora Laue i Awad-a u radu

Visualization of Business Process Modeling Anti Patterns [32], zatim autori Sadiq i Orlowska u radu *Applying graph reduction techniques for identifying structural conflicts in process models* [52], pa autori Liu i Kumar u radu *An analysis and taxonomy of unstructured workflows* [33] i autori Onoda, Ikkai, Kobayashi i Komoda u radu *Definition of deadlock patterns for business processes workflow models* [47] gdje su još 1999. godine spomenuli navedenu anomaliju u modelima poslovnih procesa.



Slika 59. Anomalija ***Zastoj toka***

Primjer – prikazuje primjer anomalije ***Zastoj toka***.



Slika 60. Primjer anomalije ***Zastoj toka***

Način prepoznavanja anomalija – navedenu anomaliju može se u modelu poslovnog procesa prepoznati po sljedećim pravilima:

- u modelu poslovnog procesa mora postojati jedan element razdvajanja i jedan element AND spajanja i
- u modelu poslovnog procesa element AND spajanja mora slijediti nakon elementa razdvajanja.

Ako se navedena pravila prikažu pseudokodom, onda to izgleda ovako:

```
1   RULE: DeadlockFlow
2   IF: Count(Fork)=>1
3   AND Count(ANDGateway)=>1
4   AND ANDGateway Precede Fork
5   THEN: DeadlockFlowAnomaly
6   ENDIF
```

Kôd 77. Način prepoznavanja anomalije ***Zastoj toka***

Način rješavanja anomalija – kod ovakve anomalije korisnika treba obavijestiti da u modelu postoji anomalija ***Zastoj toka*** kako bi mogao ispraviti model.

```
1   FIX: DeadlockFlowAnomaly
2        strMessage = "Anomaly Deadlock flow is detected in model"
```

Kôd 78. Način rješavanja anomalije ***Zastoj toka***

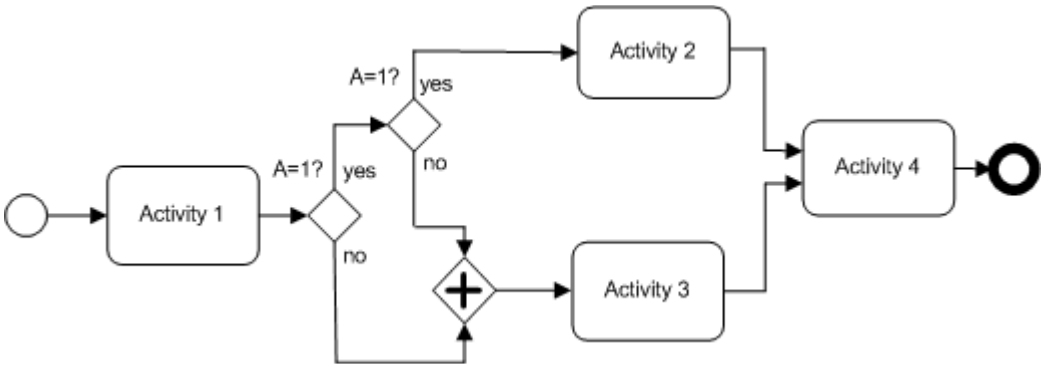
3.2.2. Anomalije pravila

Anomalije pravila predstavljaju grupu anomalija unutar ***Strukturnih anomalija***. ***Anomalije pravila*** dijele se na anomalije ***Konzistentnosti***, ***Obuhvat pravila*** i ***Tok podataka***. Anomalije ***Konzistentnosti*** dijele se na ***Redundancije pravila*** i ***Konfliktna pravila***.

Redundantna pravila

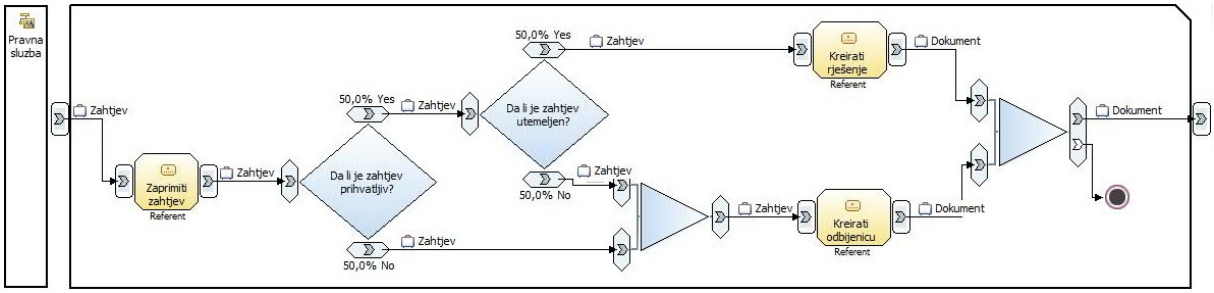
(eng. *Redundant rules*)

Opis - anomalija pod nazivom ***Redundantna pravila*** predstavljaju skup pravila koja se izvode u istom kontekstu i imaju isti efekt [69]. Anomaliju ***Redundantna pravila*** spominje u svojim radovima više autora. Pa tako autori Suwa, Scott i Shortliffe u radu *An approach to verifying completeness and consistency in a rule-based expert system* [58], spominju redundantna pravila kao anomaliju, autori Nguyen, Perkins, Laffey i Pecora u radu *Knowledge verification base* [42], spominju redundantna pravila, isto kao i autori Zhang i Nguyen u radu *Prepare: a tool for knowledge base verification* [71], pa autori Preece i Shinghal u radu *Foundation and application of knowledge base verification* [49], zatim Zaidi i Levis u radu *Validation & verification of decision making rules* [69]. Iz navedenog se vidi da je ova anomalija vrlo prepoznaljiva s obzirom da je popriličan broj autora opisalo detalje njenog prepoznavanja i uočavanja.



Slika 61. Anomalija ***Redundantna pravila***

Primjer - pokazuje model poslovnog procesa u kojem je vidljivo da oba elementa odluke imaju isti uvjet s istim brojem izlaza. Na kraju se izlazi iz elemenata odluke spajaju čime jedan od elementa odluke čine suvišnim. Provjera ove anomalije zahtjeva provjeru da li više elemenata odluke sadrže isti uvjet uz iste ulaze i isti broj izlaza. Ovakav tip anomalije u složenim modelima može nastati slučajno, a opet lako je moguće zamisliti scenarij gdje je potrebna ovakva redundancija iz recimo sigurnosnih razloga.



Slika 62. Primjer anomalije ***Redundantna pravila***

Pravila prepoznavanja – temelje se na sljedećim pravilima:

- u modelu poslovnog procesa dva ili više elemenata odluke unutar istog procesa imaju isti uvjet,
- u modelu poslovnog procesa dva ili više elemenata odluke istog procesa imaju iste ulaze i
- u modelu poslovnog procesa dva ili više elemenata odluke istog procesa imaju iste izlaze.

Ako se navedena pravila prikažu pomoću pseudokoda, onda to izgleda ovako:

```
RULE: RedundantRules
IF: Count(Gateway.Condition)>1
   AND Count(Gateway.Inputs)>1
   AND Count(Gateway.Outputs)>1
THEN: RedundantRulesAnomaly
ENDIF
```

Kôd 79. Način prepoznavanja anomalije ***Redundantna pravila***

Način rješavanja anomalije - u slučaju da se u modelu prepozna anomalija ***Redundantna pravila*** potrebno je korisniku dati do znanja da takva anomalija postoji u modelu i da je potrebno model ispraviti.

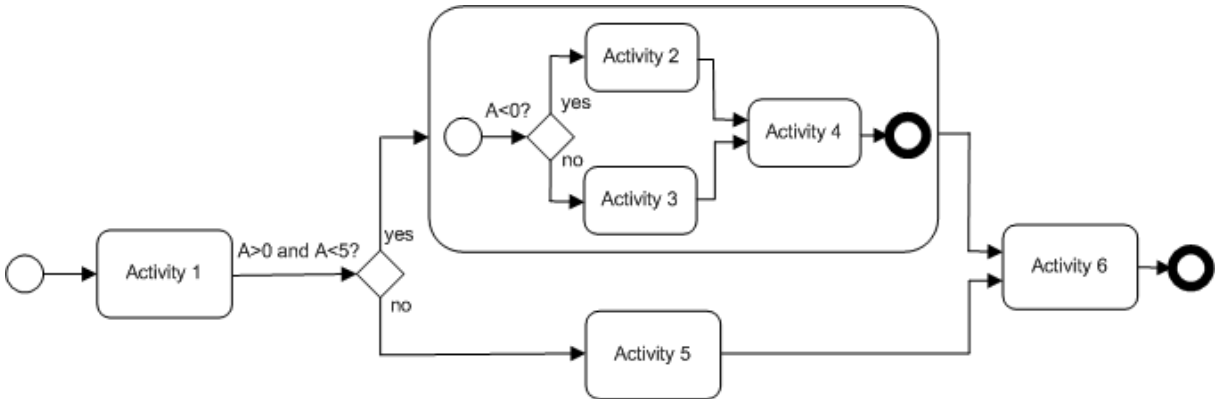
```
1    FIX: RedundantRulesAnomaly
2         strMessage = "Anomaly Redundant Rules is detected in model"
```

Kôd 80. Način rješavanja anomalije ***Redundantna pravila***

Generalna pravila

(eng. *General rules*)

Opis - anomalija ***Generalna pravila*** odnose se na skup pravila koje dijele isti zaključak, dok imaju uvjete gdje je jedno stanje generalizacija drugog tj. stanje jednog pravila vrijede i za drugi kontekst drugog skupa pravila [69]**.** Ova anomalija je obrađena od strane više autora, spomenimo neke, Suwa, Scott i Shortliffe u radu *An approach to verifying completeness and consistency in a rule-based expert system* [58], spominju je i autori Nguyen, Perkins, Laffey i Pecora u radu *Knowledge verification base* [42], zatim autori Zhang i Nguyen u radu *Prepare: a tool for knowledge base verification* [71], pa autori Preece i Shinghal u radu *Foundation and application of knowledge base verification* [49], kao i autori Zaidi i Levis u radu *Validation & verification of decision making rules* [69]. Anomalija ***Generalna pravila*** u predloženoj taksonomiji nalazi se u ***Strukturnim anomalijama*** u grupi ***Anomalija pravila*** u podgrupi ***Anomalija konzistentnosti*** u sekciji ***Redundancija pravila***.



Slika 63. Anomalija ***Generalna pravila***

Primjer - anomalija ***Generalna pravila*** vrlo je blizak anomaliji ***Redundantna pravila*** naime razlika između ove dvije anomalije je u činjenici da je drugi element odluke unutar potprocesa. Provjera ove anomalije mora uključivati i provjeru pripadajućih potprocesa, te

ako postoje elementi odluke sa istim uvjetima i istim ulazima i brojem izlaza tada se može govoriti o anomaliji ***Generalnih pravila***.

Pravila prepoznavanja anomalije – može se sagledati u sljedećoj tvrdnji jedno pravilo je generalizirana inačica drugog pravila, ako oba pravila daju isti zaključak, no jedno sadrži barem jedno ograničenje više od drugog pravila. Ako se navedena tvrdnja raspiše u pravila, onda to izgleda ovako:

- u modelu poslovnog procesa dva ili više elemenata odluke unutar procesa i potprocesa imaju uvjete s istim ili detaljnijim zahtjevima,
- u modelu poslovnog procesa dva ili više elemenata odluke istog procesa i potprocesa imaju isti ulaz i
- u modelu poslovnog procesa dva ili više elemenata odluke istog procesa i potprocesa imaju iste izlaze.

Ako se navedena pravila prikažu u pseudokodu, onda to izgleda ovako:

```
1  RULE: SubSubsumedRules
2  IF: Count(Process.Gateway)>1
3  AND Process.Gateway(i).Conditions.Rules > Process.Gateway(i+1).Conditions.Rules
4  AND (Count(Process.Gateway.Inputs)>1
5  AND Count(SubProcess.Gateway.Inputs)>1)
6  AND (Count(Process.Gateway.Outputs)>1
7  AND Count(SubProcess.Gateway.Outputs)>1)
8  THEN: SubSubsumedRulesAnomaly
9  ENDIF
```

Kôd 81. Način prepoznavanja anomalije ***Generalna pravila***

Način rješavanja anomalije - u slučaju da se u modelu prepozna ovakva anomalija, potrebno je korisnika obavijestiti o njoj da sukladno tome korisnik korigira model.

```
1  FIX: SubSubsumedRulesAnomaly
2       strMessage = "Anomaly Redundant Rules is detected in model"
```

Kôd 82. Način rješavanja anomalije ***Generalna pravila***

Izravna kontradikcija

(eng. *Direct contradiction*)

Opis - dva pravila su kontradiktorna ako uz iste uvjete daju različite rezultate. Ako navedeno pravilo stavimo u kontekst jednog uvjeta, tada znači da vrata imaju uvjet koji uz iste ulazne uvjete može dati različiti izlazni rezultat.

Skup pravila je nekonzistentno ako primjenom tih pravila jedno može obuhvatiti od predikata p tvrdnju q, gdje p i q pripadaju setu međusobno isključivih (eng. *exclusive*) predikata. Za ilustraciju uzmimo skup međusobno isključivih predikata označenih sa μ. Uopćena reprezentacija takve nekonzistentnosti je: $(\alpha \rightarrow \beta)$ gdje su $\alpha, \beta \in \mu$

Skup μ može se sintaktički definirati kao $\beta = \neg \alpha$. Definicija skupa μ čini nekonzistenciju većim problemom nego u formalnom logici [69]. Anomalija ***Izravna kontradikcija*** u predloženoj taksonomiji anomalija spada među ***Strukturne anomalije*** unutar grupe ***Anomalije pravila*** u podgrupu ***Anomalija konzistentnosti*** u sekciju ***Konfliktna pravila***.

Primjer:

IF a < 5 AND a < 100 then a =1

Način prepoznavanja anomalije – dva pravila su kontradiktorna ako uz iste uvjete daju različite rezultate, ako se navedeno pravilo stavi u kontekst jednog uvjeta, tada znači da vrata imaju uvjet koji uz iste ulazne uvjete može dati različiti izlazni rezultat. Pravila za prepoznavanje navedene anomalije su:

- u modelu poslovnog procesa postoji barem jedan element vrata i
- u modelu poslovnog procesa navedena vrata imaju uvjet koje je kontradiktorno samo po sebi, dakle bez obzira na ulazne parametre izlaz može biti različit.

Ako se navedena pravila prikažu u pseudokodu, onda to izgleda ovako:

```
1  RULE: DirectContradiction
2  IF: (Count(Process.Gateways)>0
3  AND IsContracition(Process.Gateway.Condition)=True
4  THEN: DirectContradictionAnomaly
5  ENDIF
```

Kôd 83. Način prepoznavanja anomalije ***Izravna kontradikcija***

Način rješavanja anomalije - u slučaju pronalaska anomalije ***Izravna kontradikcija*** u modelu poslovno procesa, korisnika je potrebno informirati da takva anomalija postoji u modelu i da je potrebno korigirati model.

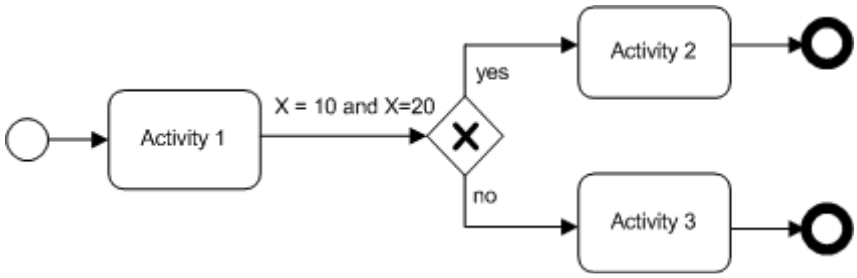
```
1  FIX: DirectContradictionAnomaly
2       strMessage = "Anomaly Direct Contradiction is detected in model"
```

Kôd 84. Način rješavanja anomalije ***Izravna kontradikcija***

Kontradikcija u ulazu

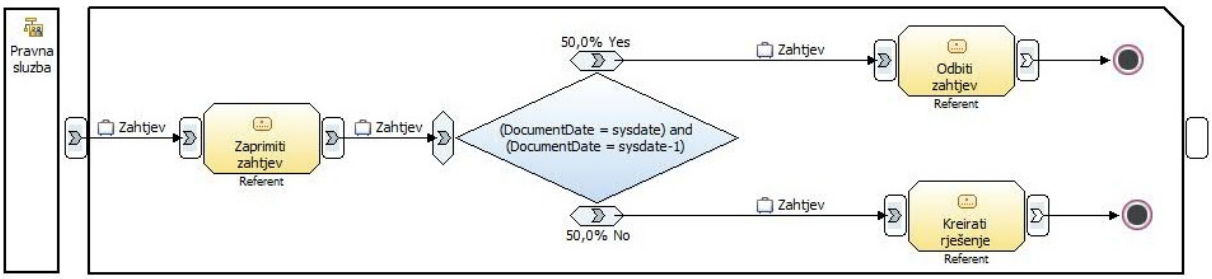
(eng. *Contradiction in input*)

Opis - anomalija ***Kontradikcija na ulazu*** odnosi se na pravila koja sadrže kontradiktorne uvjete, npr. pravilo koje vrijedi ako jedna varijabla ima dvije različite vrijednosti istovremeno, npr. IF x = 10 and x = 20 THEN B [69].



Slika 64. Anomalija ***Kontradikcija na ulazu***

Primjer - na primjeru se vidi model poslovnog procesa u kojem je u elementu uvjeta, uvjet koji je kontradiktoran. Uvjet provjerava da li je poslovna stavka *Zahtjev* datirana današnjim danom i kao jučerašnji dan. S obzirom da ni jedan dokument ne zadovoljava ovaj uvjet, tijek procesa će se uvijek odvijati prema drugoj grani uvjeta.



Slika 65. Primjer anomalije ***Kontradikcija na ulazu***

Način prepoznavanja anomalije – navedena anomalija može se prepoznati samo ispitivanjem uvjeta, ako uvjet elementa odluke nikad ne zadovoljava. Ako se navedene tvrdnje raspišu u pravila, onda to izgleda ovako:

- u modelu poslovnog procesa postoji jedan element odluke i
- u modelu poslovnog procesa navedeni uvjet odluke u svim slučajevima ne zadovoljava.

Ako se navedena pravila prikažu u pseudokodu, onda se za ispitivanje uvjeta koristi funkcija **AllScenario** koja vraća **True** ili **False**.

```
1  RULE: ContradictionInInput
2  IF: (Count(Process.Gateways)>0
3  AND AllScenario(Process.Gateway.Condition) IS FALSE
4  THEN: ContradictionInInputAnomaly
5  ENDIF
```

Kôd 85. Način prepoznavanja anomalije ***Kontradikcija na ulazu***

Način rješavanja anomalije – u slučaju da u modelu postoji anomalija ***Kontradikcija na ulazu*** potrebno je korisnika obavijesti da postoji u modelu kako bi ga mogao korigirati i ispraviti.

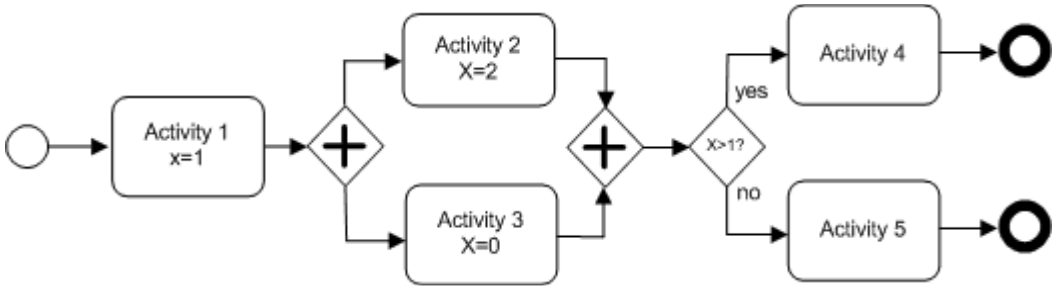
```
1    FIX: ContradictionInInputAnomaly
2         strMessage = "Anomaly Contradiction In Input is detected in model"
```

Kôd 86. Način rješavanja anomalije ***Kontradikcija na ulazu***

Kontradikcija u zaključku

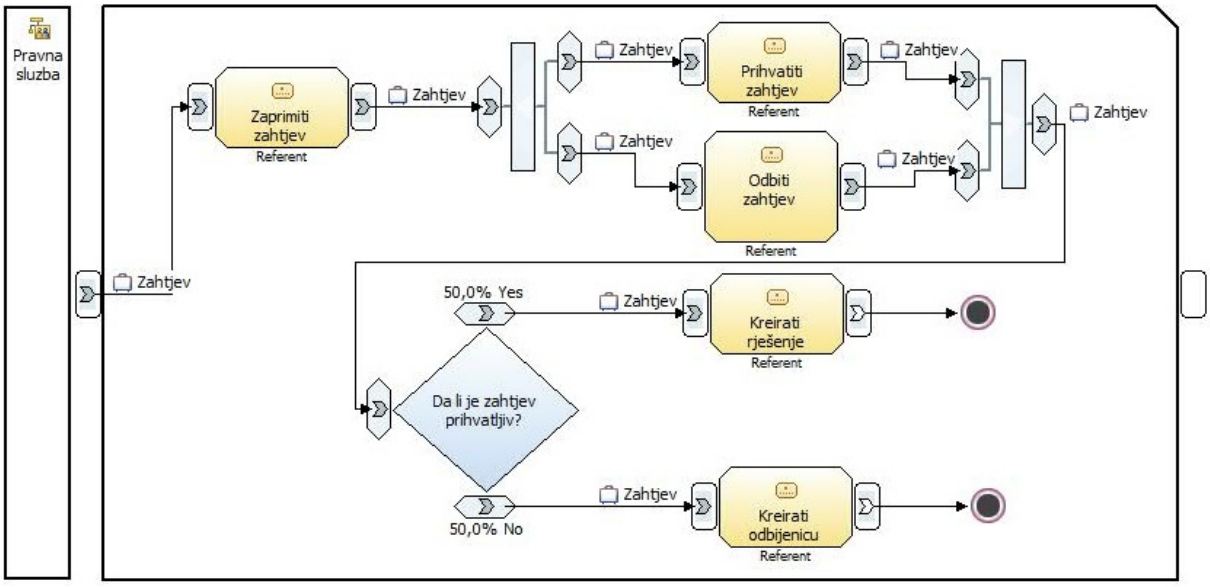
(eng. *Contradiction in conclusion*)

Opis - anomalija ***Kontradikcija u zaključku*** odnosi se na skup pravila koja imaju iste uvjete, ali čiji zaključci ne rade kad se primjene u isto vrijeme. Na primjer, dva pravila imaju iste uvjete i izvode se u isto vrijeme, ali svaki od njihovih zaključaka daje različitu vrijednost na istu vanjsku varijablu, pa nije u potpunosti jasno koja vrijednost varijable je ispravna [69].



Slika 66. Anomalija ***Kontradikcija u zaključku***

Primjer: Primjer pokazuje anomaliju ***Kontradikcija u zaključku***, gdje se provjerava da li je *Zahtjev* prihvatljiv iako su zaključne aktivnosti vezane uz stanje poslovnog stavka već izvedene.



Slika 67. Primjer anomalije ***Kontradikcija u zaključku***

Način prepoznavanja anomalije - navedenu anomaliju može se prepoznati po sljedećim pravilima:

- u modelu poslovnog procesa postoje elementi razdvajanja i element stapanja,
- u modelu poslovnog procesa nakon tih elemenata slijedi element odluke i
- u modelu poslovnog procesa poslovna stavka je ista kroz sve navedene elemente.

Ako se navedena pravila prikažu u pseudokodu, onda to izgleda ovako:

```
RULE: ContradictionInConclusion:
IF: (Fork PRECEDE Merge)
AND (Merge And PRECEDE Gateway)
AND (Fork.InputSequenceFlow.BusinessItems = Merge.OutputSequenceFlow.BusinessItems)
AND (Fork.InputSequenceFlow.BusinessItems = Gateway.InputSequenceFlow.BusinessITems)
THEN:
   ContradictionInConclusionAnomaly()
ENDIF
```

Kôd 87. Način prepoznavanja anomalije ***Kontradikcija u zaključku***

Način rješavanja anomalije - u slučaju da u modelu postoji anomalija ***Kontradikcija u zaključku*** potrebno je korisnika obavijesti da postoji u modelu kako bi ga mogao korigirati i ispraviti.

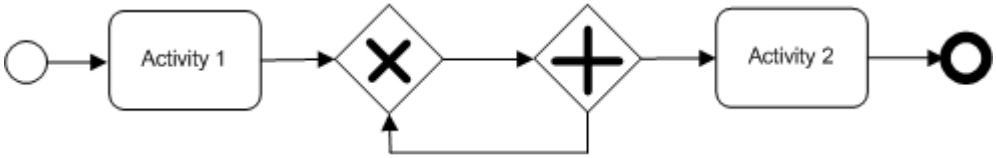
```
FIX: ContradictionInConclusionAnomaly()
     strMessage = = "There is anomaly Contradiction in conclusiuon in process model"
```

Kôd 88. Način rješavanja anomalije ***Kontradikcija u zaključku***

Zapinjanje temeljeno na pravilima

(eng. *Rule-base livelock*)

Opis - anomalija pod nazivom ***Zapinjanje temeljeno na pravilima*** odnosi se na probleme sa derivacijom pravila koji ovise jedan o drugom iako ne bi trebali. Do anomalije ***Zapinjanje temeljeno na pravilima*** dovodi kad izlazno pravilo petlje sadrži preklapajuće zaključke npr: `If A then B and if B then A` [69]. Anomalija ***Zapinjanje temeljeno na pravilima*** u predloženoj taksonomiji spada u ***Strukturne anomalije*** u grupu ***Anomalije pravila*** pod ***Anomalije konzistentnosti*** u podgrupu ***Anomalije petlje***.



Slika 68. Anomalije ***Zapinjanje temeljeno na pravilima***

Način prepoznavanja anomalije – navedena anomalija može se prepoznati na temelju sljedećih pravila:

- u modelu poslovnog procesa postoje element OR spajanja,
- u modelu poslovnog procesa postoje element AND razdvajanja,
- u modelu poslovnog procesa AND razdvajanje slijedi nakon OR spajanja i
- u modelu poslovnog procesa AND element razdvajanja ima povratnu vezu na element OR spajanja.

Ako se navedena pravila prikažu u pseudokodu, onda to izgleda ovako:

```
RULE: LivelockByRules:
IF: (ORJoin PRECEDE ANDFork)
AND (ANDFork BACKCONNECT ORJoin)
THEN:
   LivelockByRulesAnomaly()
ENDIF
```

Kôd 89. Način prepoznavanja anomalije ***Zapinjanje temeljeno na pravilima***

Način rješavanja anomalije – za navedenu anomaliju potrebno je korisnika obavijestiti da je prepoznata u modelu poslovnog procesa i da model treba korigirati.

```
FIX: LivelockByRulesAnomaly()
     strMessage = = "There is anomaly rule-based livelock in process model"
```

Kôd 90. Način rješavanja anomalije ***Zapinjanje temeljeno na pravilima***

Zastoj temeljen na pravilima

(eng. *Rule-base deadlock*)

Opis - anomalija ***Zastoj temeljen na pravilima*** nastaje kad dva ili više pravila onemogućavaju jedan drugog od izvršenja. Prvo pravilo se ne može izvršiti, jer su njegovi uvjeti zaključak drugog pravila čiji zaključci ovise o zaključcima prvog pravila. Npr. `If A and C then B, If B and C then A` [69].

Slika 69. Anomalija ***Zastoj temeljen na pravilima***

Način prepoznavanja anomalije – navedena anomalija može se otkriti sljedećim pravilima:

- u modelu poslovnog procesa postoji element OR razdvajanja,
- u modelu poslovnog procesa postoji element OR spajanja,
- u modeli poslovnog procesa element OR razdvajanja prati element OR spajanja i
- u modelu poslovnog procesa element OR spajanja povratno je spojen s elementom OR razdvajanja.

Ako se navedena pravila opišu pseudokodom onda to izgleda ovako:

```
RULE: RuleBasedDeadlock:
IF: Count(ORFork)>0
AND Count(ORJoin)>0
AND (ORJoin PRECEDE ORFork)
AND (ORFork BACKCONNECT ORJoin)
THEN:
   RuleBasedDeadlockAnomaly()
ENDIF
```

Kôd 91. Način prepoznavanja anomalije ***Zastoj temeljen na pravilima***

Način rješavanja anomalije – ako se navedena anomalija pojavi u modelu poslovnog procesa potrebno je korisnika obavijestiti o anomaliji u modelu kako bi model mogao ispraviti.

```
FIX: RuleBasedDeadlockAnomaly()
     strMessage = = "There is anomaly rule base deadlock in process model"
```

Kôd 92. Način rješavanja anomalije ***Zastoj temeljen na pravilima***

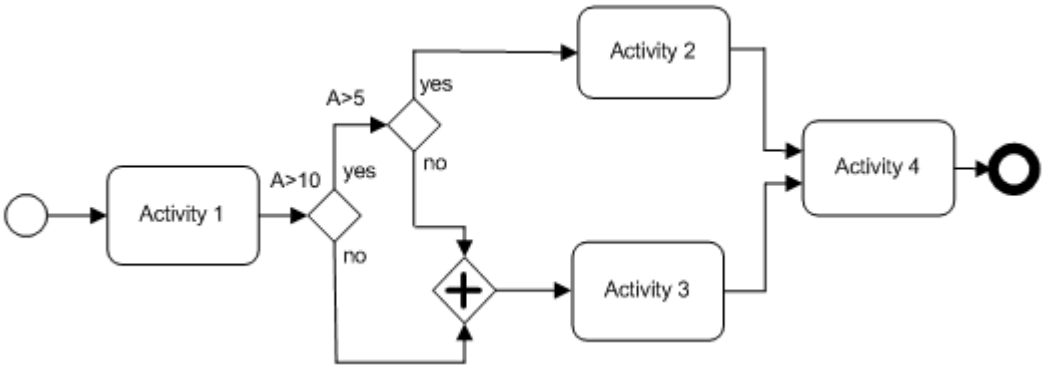
Izolirana pravila
(eng. *Isolated rules*)

Opis – rezultat anomalije ***Izolirana pravila*** nema nikakav efekt na tijek poslovnog procesa. Anomalija ***Izolirana pravila*** razlikuje se od anomalije ***Redundantnih pravila*** koja u nekim slučajevima može biti i korisna [16]. Pravilo je da ***Izolirana pravila*** nastaju ako i samo ako [69]:

- svi prijedlozi unutar premise su kompleksni i ne mogu biti objašnjeni u terminima osnovnih koncepta U.
- zaključak R nije temeljni koncept niti postoji pravilo R za tvrdnju α gdje je $\alpha \in \Psi$.

Primjer –

```
IF A > 10 THEN
   IF A > 5 THEN
      …
   END IF
END IF
```



Slika 70. Primjer anomalije ***Izolirana pravila***

Način prepoznavanja anomalije – navedena anomalija može se otkriti prema sljedećim pravilima:

- u modelu poslovnog procesa postoje minimalno dva elementa odlučivanja,
- u modelu poslovnog procesa drugi element odlučivanja slijedi nakon prvog elementa odlučivanja i
- u modelu poslovnog procesa uvjet jednog elementa odlučivanja mora biti sadržan u uvjetu drugog elementa.

Ako se navedena pravila opišu pseudokodom onda to izgleda ovako:

```
RULE: IsolatedRules:
IF: Count(Gateway)=>2
AND Gateway(i) Precede Gateway(i-1)
ABD (Gateway(i).Condition include Gateway(i-1).Condition)
THEN:
   IsolatedRulesAnomaly()
ENDIF
```

Kôd 93. Način prepoznavanja anomalije ***Izolirana pravila***

Način rješavanja anomalije – u slučaju da se prepozna anomalija ***Izolirana pravila*** u modelu poslovnog procesa potrebno je korisnika obavijestiti da je takva anomalija otkrivena u modelu i da model mora ispraviti.

```
FIX: IsolatedRulesAnomaly()
     strMessage = = "There is anomaly isolated rules in process model"
```

Kôd 94. Način rješavanja anomalije ***Izolirana pravila***

Beskorisni zaključci

(eng. *Useless conclusions*)

Opis - glavni koncept anomalije ***Beskorisni zaključci*** sastoji se od pravila uvjeta koji se može sastojati od brojnih atributa odluka koji proizlaze iz pravila. S obzirom na unos, ako izvedeni zaključak nije glavni koncept ili ako neki ili svi atributi ne mogu biti zaključeni iz temeljnog pravila, set pravila primijenjenih na taj ulaz je nepotpun. Prema tome glavni koncept je ovisan o domeni i zahtjeva eksplicitnu definiciju [69]. Anomalija ***Beskorisni zaključci*** prema predloženoj taksonomiji je ***Strukturna anomalija*** koja spada u grupu ***Anomalija pravila*** i koja spada u podgrupu ***Obuhvat pravila***.

Primjer –

IF (A > 10 AND A>1 AND A>2 AND A>3) THEN

END IF

Način prepoznavanja anomalije – autori Dohring i Heublein u radu *Anomalies in Rule-Adopted Workflows – A Taxonomy and Solutions for vBPMN* [16] objavili su klasu kojom se prepoznaje navedena anomalija u *eCore/oclInEcore* objektima u *jBoss Drools* verifikatoru pravila.

```
class AdaptationRule

operation noImpact ():Boolean [1]
{body:
 self.appliedOnEmptySegment() and self.adaptationApplication.adaptation->exists (
      x|x.oclAsType( AdaptationTemplate ).emptyAdaptation ());
 }

 operation appliedOnEmptySegment() : Boolean [ 1 ]
 {body :
   let en : AdaptationStartEvent = self.segmentEntry .segmentEntryNode in
     en.arcsBetween ( en.CounterpartEnd)->select (x| not ( x.oclAsType(bpmn2 ::
        SequenceFlow).sourceRef=en and x.oclAsType (bpmn2 : :SequenceFlow).
        targetRef=en.CounterpartEnd ) )->size () =0;
 }

class AdaptationTemplate extends bpmn2 : : Process
operation emptyAdaptation ( ) : Boolean [1]
{body :
let fn : Set (bpmn2 : : FlowNode) =
     self.adaptationGraph.flowElements->select
    (x|x. oclIsKind Of ( bpmn2 : : FlowNode ) )->
    collect ( oclAsType (bpmn2 : : FlowNode ) )->asSet() in
    let patternEndEvents : Set (bpmn2 : : EndEvent ) = fn ->
        select (x|x . oclIsKindOf (bpmn2 :: EndEvent ) ) ->
        collect ( oclAsType (bpmn2 : :EndEvent ) )->
        asSet () in
        patternEndEvents .eventDefinitions->size ()=0 and fn->
     select (x|x . oclIsKindOf (bpmn2 :: Activity ) or x. oclIsKindOf (bpmn2:: Event) )->
     size ()<=2;
 }
```

Kôd 95. Način prepoznavanja anomalije ***Beskorisni zaključci***
(Izvor: Dohring, M., Heublein, S.: *Anomalies in Rule-Adopted Workflows – A Taxonomy and solutions for vBPMN*, 6th European Conference on Software Maintenance and Reengineering, pages 117-126. Szeged, Hungary : IEEE.)

Na temelju sljedećih pravila može se prepoznati navedena anomalija u modelu poslovnog procesa:

- u modelu poslovnog procesa postoji barem jedan element odlučivanja i
- u modelu poslovnog procesa u navedenom elementu uvjet odlučivanja mora imati više manjih pravila koji su podskup određenog pravila u uvjetu.

Ako se navedena pravila prikažu u pseudokodu, onda to izgleda ovako:

```
RULE: UselessConclusion:
IF: Count(Gateway)=>1
AND ConditionIsUseless(Gateway.Condition)=True
THEN:
   UselessConclusionAnomaly()
ENDIF
```

Kôd 96. Način prepoznavanja anomalije ***Beskorisni zaključci***

Način rješavanja anomalije – u slučaju nailaska na navedenu anomaliju, potrebno je korisnika obavijestiti da treba korigirati model.

```
FIX: UselessConclusionsAnomaly:
     strMessage =“There is anomaly useless nonclusions in process model”
```

Kôd 97. Način rješavanja anomalije ***Beskorisni zaključci***

Viseći uvjeti

(eng. *Dangling conditions*)

Opis – anomalija ***Viseći uvjeti*** je obrnuti slučaj od anomalije nepotrebnih zaključaka gdje se uvjeti vjerojatno mogu zadovoljiti u temeljnom kontekstu, ali ti zaključci su modelirani na način da nikad ne mogu imati ikakav učinak. Kod anomalija ***Visećih uvjeta*** situacija je obrnuta, zaključci imaju efekt, no uvjeti se nikad ne mogu podudariti [69]. Anomalija ***Viseći uvjeti*** je prema predloženoj taksonomiji ***Strukturna anomalija*** u grupi ***Anomalija pravila*** u podgrupi ***Obuhvat podataka***.

Način prepoznavanja anomalije – temelje se na sljedećim pravilima:

- u modelu poslovnog procesa postoji barem jedan element odlučivanja i
- u modelu poslovnog procesa navedeni element odlučivanja ima barem jedan ulaz koji nije spojen poslovnom stavkom za neki drugi element.

Ako se navedena pravila opišu pseudokodom, onda to izgleda ovako:

```
1  RULE: DanglingConditions:
2  IF: Count(Process.Gateway)>0
3  AND ((Gateway.Input DEFINED)
4  AND (Connection(Gateway.Input))=0))
5  THEN:
6     DanglingConnectionsAnomaly()
7  ENDIF
```

Kôd 98. Način prepoznavanja anomalije ***Viseća pravila***

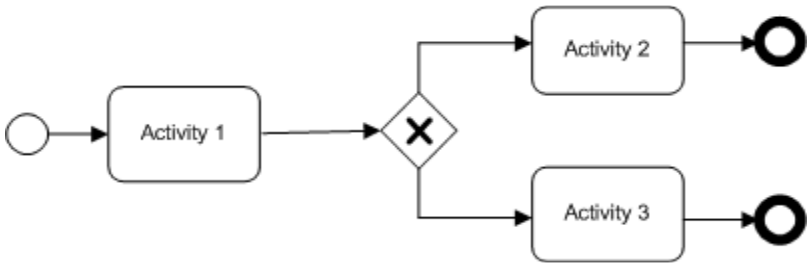
Način rješavanja anomalije – u slučaju da se u modelu procesa naiđe na navedenu anomaliju onda korisnika treba obavijestiti da je navedena anomalija pronađena u modelu procesa kako bi mogao ispraviti model.

```
1  FIX: DanglingConnectionsAnomaly:
2       strMessage =“There is anomaly Dangling Conditions in process model”
```

Kôd 99. Način rješavanja anomalije ***Viseća pravila***

Nepostojeća pravila
(eng. *Missing rules*)

Opis – anomalija ***Nepostojeća pravila*** nastaje ako pravilo nije definirano. Takva anomalija nastaje ako modelar poslovnih procesa definira pravilo koje se izvodi u određenom kontekstu, ali ne definira pravilo koje se primjenjuje ako se kontekst ne primjeni [16]. Drugim riječima radi se o anomaliji koja nastaje ako se očekuje neki određeni rezultat, no ne postoje pravila koja bi producirala očekivani rezultat. Obično se radi u određenim uvjetima koje je potrebno zadovoljiti da bi se došlo do određenog rezultata. U najjednostavnijem slučaju navedenu anomalija može se opisati pomoću običnih vrata odlučivanja, koja nema definirane uvjete, tj. pravila pomoću kojeg se može ostvariti određeni rezultat. Anomalija ***Nepostojeća pravila*** je prema predloženoj taksonomiji ***Strukturna anomalija*** u grupi ***Anomalija pravila*** u podgrupi ***Obuhvat podataka***.



Slika 71. Anomalija ***Nepostojeća pravila***

Način prepoznavanja anomalije – ostvarivo je pomoću sljedećih pravila:

- u modelu poslovnog procesa postoji barem jedan element odlučivanja i
- u modelu poslovnog procesa u navedenom elementu odlučivanja nije definirano pravilo.

Ako se navedena pravila izraze u pseudokodu, onda to izgleda ovako:

```
1  RULE: MissingRules:
2  IF: Count(geteway)>0
3  AND (Gateway.Condition)->IsEmpty
4  THEN: MissingRulesAnomaly()
5  ENDIF
```

Kôd 100. Način prepoznavanja anomalije ***Nepostojeća pravila***

Način rješavanja anomalije – u slučaju da se naiđe na navedenu anomaliju u modelu poslovnog procesa, potrebno je korisnika obavijestiti o postojanju anomalije u modelu kako bi ispravio model.

```
1  FIX: MissingRulesAnomaly:
2       strMessage ="There is anomaly missing rules in process model"
```

Kôd 101. Način rješavanja anomalije ***Nepostojeća pravila***

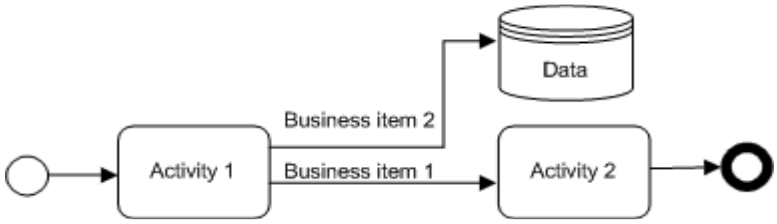
Redundantni podaci

(eng. *Redundant data*)

Opis - anomalija ***Redundantni podaci*** opisuju anomaliju koja nastaje ukoliko se kroz tok sekvence provlače podaci koji nisu potrebni da bi se on izvršio. Podaci koji se kreiraju u procesu, no nikad se ne koriste nakon njihovog nastanka nazivaju se redundantni podaci [16]. Autori Trčka, Sidorova i Aalst i u svojem radu *Data-Flow Anti-patterns: Discovering Data-Flow Errors in Workflows* [62] napominju da postoji dva tipa anomalije ***Redundantni podaci*** i to: ***Jako redundantni podaci*** i ***Slabo redundantni podaci***. Anomalija ***Jako redundantni podaci*** je anomalija koja se pojavljuje kad postoji radni korak koji zapisuje podatke nakon kojeg se u svim mogućim tokovima izvođenja zapisani podatak nikad ne koristi prije nego je uništen ili je nego je poslovni proces završio. Anomalije ***Slabi redundantni podaci*** nastaju kad postoji scenarij u kojem je određeni podatak u poslovnom procesu zapisan no nikad poslije nije upotrijebljen, nego ostaje nepročitan prije nego je uništen ili je poslovni proces završio. Anomalija ***Jaki redundantni podaci*** posljedica je greške i teško da je namjerno tako izmodelirana, dok anomalija ***Slabi redundantni podaci*** mogu biti posljedica odluke da poslovni procesa mora na taj način funkcionirati. U radu *Detecting DataFlow Errors u BPMN 2.0* autori Stackelberg, Putze, Mulle i Bohm anomaliju redundantnih podataka razlikuju

- jaka anomalija redundantnih podataka,
- slaba anomalija redundantnih podataka i
- redundancija opcionalnih podataka.

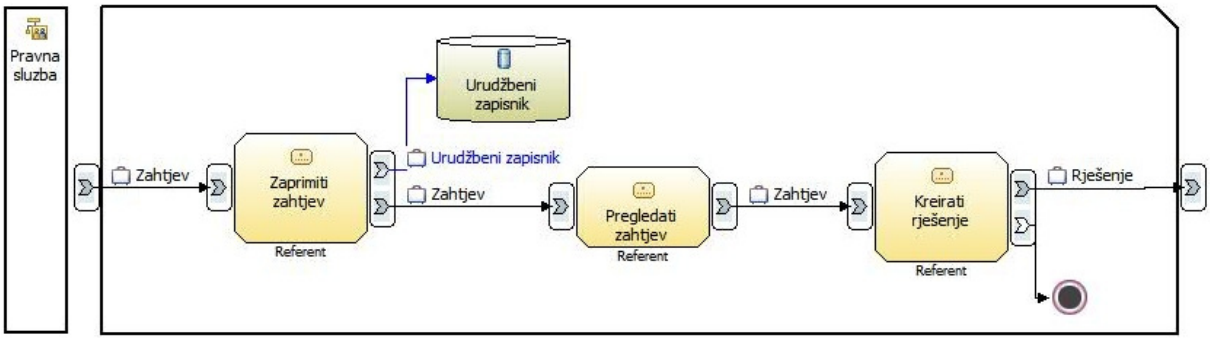
Jaka anomalija redundantnih podataka nastaje kad objekt podataka ima obavezni izlaz, no ni jedan element toka ne koristi podatke iz objekta podataka kao obavezan ili kao opcionalan u svim izvršnim tokovima modela poslovnih procesa. ***Slaba anomalija redundantnih podataka*** nastaje kad postoje barem jedan izvršni tok procesa u kojem je podatak iz objekta podataka obavezan ili opcionalan, a da je obavezan izlaz iz objekta podataka. Redundancija opcionalnih podataka nastaje ako objekta podataka ima opcionalni izlaz koji se ne koristi u daljnjem tijeku procesa, jer objekt podataka nema obavezni izlaz ili opcionalni ulazu pojedinoj izvršnoj putanji.



Slika 72. Anomalija ***Redundantni podaci***

U predloženoj taksonomiji anomalija ***Redundantni podaci*** predstavljaju ***Strukturnu anomaliju***, koja je u grupi ***Anomalija pravila*** i koja je u podgrupi anomalija ***Tok podataka***.

Primjer - pokazuje scenarij poslovnog procesa u kojem prvi radni korak zapisuje podatke o *Urudžbenom zapisniku* u bazu podataka. Podaci o *Urudžbenom zapisniku* se u daljnjem toku u navedenom poslovnom procesu ne koriste. Ako u cijeloj organizaciji ne postoji poslovni proces koji koristi zapisane podatke o *Urudžbenom zapisniku* u tom slučaju takve podatke treba proglasiti nepotrebnima. Nadalje, moguć je slučaj da zapisane podatke koristi neka organizacija izvan organizacije u kojem je poslovni proces, kao što su državna regulatorna tijela, agencije, itd., u tom slučaju takvi podaci nisu redundantni.



Slika 73. Primjer anomalije ***Redundantni podaci***

Način prepoznavanja anomalije – temelji se na sljedećim pravilima:

- u modelu poslovnog procesa postoji aktivnost koja upisuje podatke u bazu podataka, no one se ne koriste u daljem toku poslovnog procesa ni u bilo kojem drugom poslovnom procesu organizacije ili izvan nje i
- u modelu poslovnog procesa postoji objekt baze podataka u koji se upisuje podaci, no nikad se ne čitaju.

Navedena pravila mogu se prikazati pomoću OCL pravila, u tom slučaju to izgleda ovako:

```
1  context DataObject
2  inv DataRedundancy:
3     (self.outputSequenceFlows()-> IsEmpty())
4     implies
5     DataRedundancyAnomaly
```

Kôd 102. Način prepoznavanja anomalije ***Redundantni podaci***

Način rješavanja anomalije - ovakva anomalija može se spriječiti OCL pravilom u kojem se ne dozvoljava da postoji objekt podataka koji nema izlaz.

```
1   context DataObject
2   inv DataRedundancyAnomaly:
3      (self.outputSequenceFlows()-> IsEmpty())
```

Kôd 103. Način rješavanja anomalije ***Redundantni podaci***

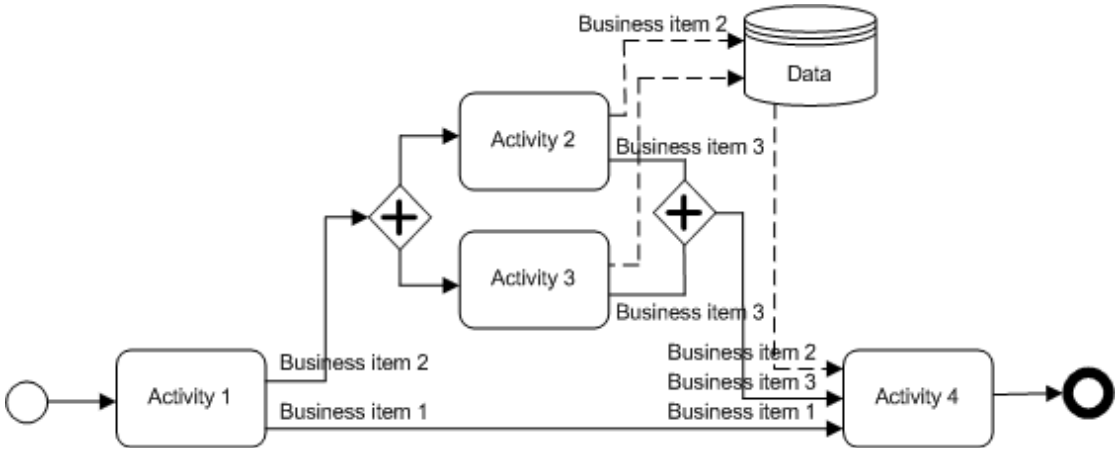
Izgubljeni podaci

(eng. *Lost data*)

Opis - anomalija pod nazivom ***Izgubljeni podaci*** nastaje kad je podatak prepisan, a da prije nije bio pročitan. Dakle, kad jedna aktivnost kreirani podatak promijeni prije nego je bio pročitan od strane druge aktivnosti, time je zapravo njen sadržaj izgubljen [16].

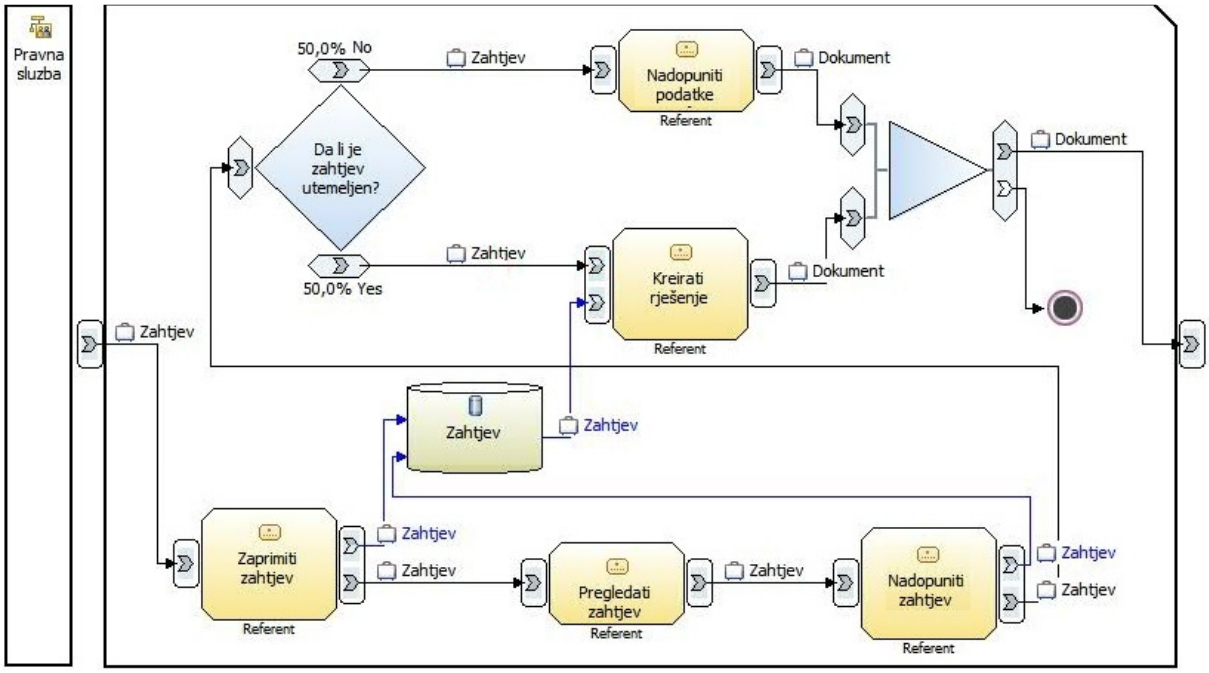
Autori Trčka, der Aalst i Sidorova u svojem radu *Data-Flow Anti-patterns: Discovering Data-FLow Errors in Workflows* [62] napominju da postoji dva tipa anomalije Izgubljeni podaci i to: ***Jako izgubljeni podaci*** i ***Slabo izgubljeni podaci***. ***Jako izgubljeni podaci*** je anomalija koja nastaje ako postoji aktivnost koja zapisuje podatke nakon koje svi mogući tokovi izvršenja mogu prebrisati navedeni podatak bez da je prethodna vrijednost bila pročitana. ***Slabo izgubljeni podaci*** je anomalija koja je nastala ako postoji izvršna sekvenca koja može prebrisati podatak bez da je prethodna vrijednost bila pročitana. ***Anomalija Jako izgubljeni podaci*** implicira anomaliju slabo izgubljenih podataka, dok to obrnuto nije slučaj. Anomalija ***Jako izgubljeni podaci*** obično indicira grašku u modeliranju, dok anomalija ***Slabo izgubljeni podaci*** može biti odluka tijekom modeliranja modela. Dakle anomalija ***Slabo izgubljeni podaci*** može biti normalno ponašanje modela poslovnog procesa.

U radu *Detecting DataFlow Errors u BPMN 2.0* autori Stackelberg, Putze, Mulle i Bohm razlikuju ***Jako izgubljene podatke***, ***Slabo izgubljene podatke*** i ***Opcionalno izgubljene podatke.*** Anomalija ***Jako izgubljeni podaci*** nastaje ako objekt podataka ima nekoliko obaveznih ulaza podataka prije nego se dogodi obavezni izlaz podatka. Ova situacija se događa u svim slučajevima gdje se prethodni podaci izgube. Anomalija ***Slabo izgubljeni podaci*** nastaje ako objekt podataka ima obavezni izlaz i gdje postoji barem jedna aktivnost koji ima obavezan izlaz, a koji nema obavezan ulaz. Anomalija ***Opcionalno izgubljeni podaci*** nastaje ako u poslovnom procesu postoji objekt podataka koji ima obavezni ili opcionalni izlaz, a koji ima i aktivnost koja zapisuje podatak u objekta podataka, a da nije prethodno opcionalno ili obavezno pročitao podatak. U predloženoj taksonomiji anomalija ***Izgubljeni podaci*** predstavljaju ***Strukturnu anomaliju***, koja je u grupi ***Anomalija pravila*** i koja je u podgrupi anomalija ***Tok podataka***.



Slika 74. Anomalija ***Izgubljeni podaci***

Primjer - koji je već prije korišten u ovome radu prikazuje slučaj modela poslovnog procesa u kojem je moguća anomalija ***Izgubljeni podaci***. Dakle podaci koji se zapisuju u bazu podataka mogu u određenom radnom koraku biti izbrisani ili promijenjeni prije nego ih dohvati drug radni korak.



Slika 75. Poslovni proces sa mogućom anomalijom ***Izgubljeni podaci***

Način prepoznavanja anomalije – temelje se na sljedećim pravilima:

- u modelu poslovnog procesa postoji više aktivnosti koji upisuju, ažuriraju i čitaju podatke po istom dokumentu u istome poslovnom procesu,

- u modelu poslovnog procesa postoji aktivnost koja briše određeni podatak i izvodi se prije aktivnosti čitanja podataka i
- u modelu poslovnog procesa ne postoji mehanizam provjere vremena čitanja i zapisivanja podatka tijekom rada poslovnog procesa.

Ako se navedena pravila pretvore u pseudokod, onda to izgleda ovako:

```
RULE: LostData:
IF: (Database.inputSequenceFlows()->notEmpty()
AND Count(Database.inputSequenceFlows())>1
AND Activity.Delete() PRECEDE Activity.Read()
THEN: LostDataAnomaly()
ENDIF
```

Kôd 104. Način prepoznavanja anomalije ***Izgubljeni podaci***

Način rješavanja anomalije - zbog složenosti anomalije teško je provesti automatiziranje rješavanja, stoga je korisnika potrebno obavijestiti da je anomalija prepoznata u modelu i da ga je potrebno korigirati. Za rješavanje potrebno u ugraditi mehanizam provjere da li se određeni podatak briše prije nego je se podataka pročita u daljnjim aktivnostima.

```
FIX: LostDataAnomaly:
     strMessage ="There is anomaly lost data in process model"
```

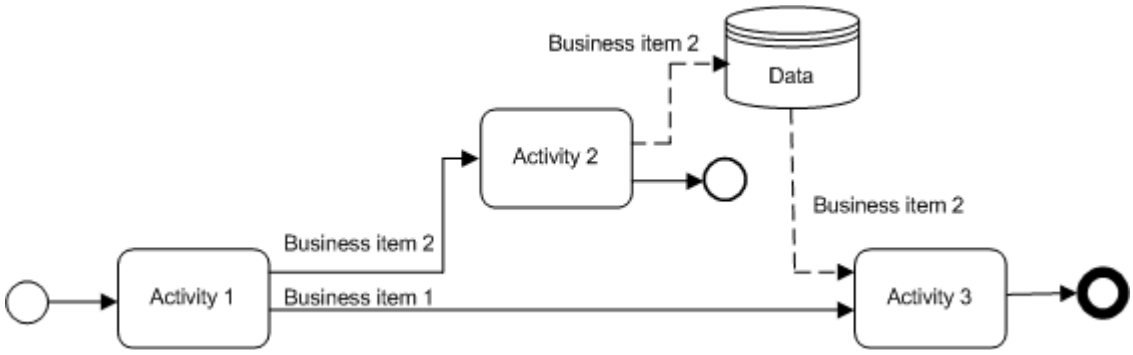
Kôd 105. Način rješavanja anomalije ***Izgubljeni podaci***

Nepostojeći podaci

(eng. *Missing data*)

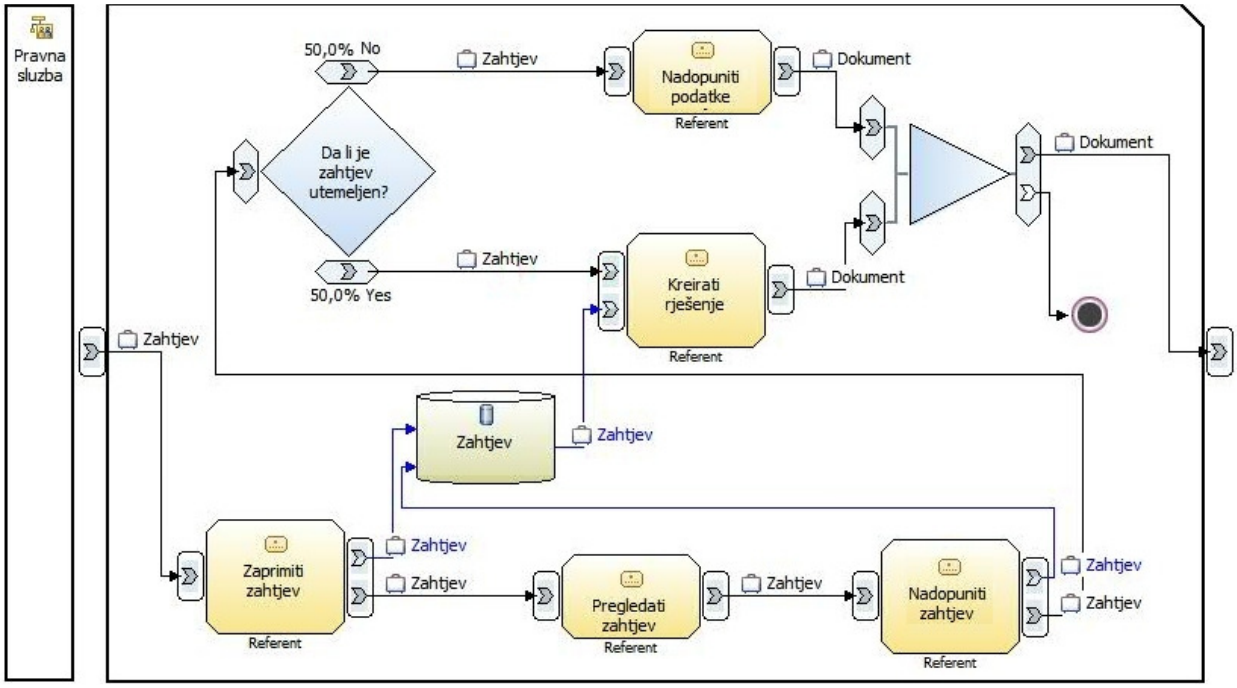
Opis - anomalija ***Nepostojeći podaci*** nastaje kada određena aktivnost ne može pročitati podatak koji mu je potreban za njegovo izvršenje. Takva anomalija može nastati kad podatak nije osiguran od strane jedne aktivnosti za drugu [16]. U tom slučaju podatak nije nastao ili je izbrisan prije nego je tok došao do aktivnosti koja zahtjeva podatak.

U radu *Detecting DataFlow Errors u BPMN 2.0* autori Stackelberg, Putze, Mulle i Bohm razlikuju anomalije ***Nepostojeći podaci*** i ***Opcionalno nepostojeći podaci***. Anomalija ***Nepostojeći podaci*** nastaje ako u modelu procesa postoji obavezno zapisivanje podataka, no ne postoji obavezno čitanje podataka. Anomalija ***Opcionalno nepostojeći podaci*** nastaje ako u poslovnom procesu postoji aktivnost za čitanje podataka inicirana prije nego aktivnosti za pisanje podataka. U predloženoj taksonomiji anomalija ***Nepostojeći podaci*** predstavljaju ***Strukturnu anomaliju***, koja je u grupi ***Anomalija pravila*** i koja je u podgrupi anomalija ***Tok podataka***.



Slika 76. Anomalija ***Nepostojeći podaci***

Primjer – ako se uzme isti primjer koji je opisan kod anomalije ***Nekonzistentnost promjene iz prošlosti.*** U tome primjeru vidljivo je da se u bazu podataka ***Zahtjev*** upisuju podaci nakon radnog koraka ***Zaprimiti zahtjev.*** Nakon toga nakon radnog koraka ***Nadopuniti zahtjev*** ti podaci se ažuriraju i ponovo se ti podaci koriste prije radnog koraka ***Kreirati rješenje***. Problem nastaje ako se podaci i toj poslovnoj stavci izbrišu prije nego se kreira rješenje u radnom koraku ***Kreirati rješenje***. U tom slučaju radni korak ***Kreirati rješenje*** se ne može aktivirati, a samim time cijeli poslovni proces se ne može završiti.



Slika 77. Primjer modela poslovnog procesa sa mogućom anomalijom ***Nepostojeći podaci***

Način prepoznavanja anomalije – temelji se na sljedećim pravilima:

- u modelu poslovnog procesa postoji barem jedan element baze podataka,

- u modelu poslovnog procesa postoje aktivnosti koje upisuju, ažuriraju i čitaju isti podatak u istoj bazi podataka i
- u modelu poslovnog procesa ne postoji mehanizam provjere postojanja određenog podatka prije njegovog korištenja u poslovnom procesu.

Navedena pravila opisana u pseudokodu bi izgledala ovako:

```
1  RULE: MissingData:
2  IF Count(Database)>0
3  AND Database.inputSequenceFlows()->notEmpty()
4  AND Database.outputSequenceFlows())->notEmpty
5  AND Exist(self.BusinessItem(ID))= False)
6  THEN: MissingDataAnomaly
7  ENDIF
```

Kôd 106. Način prepoznavanja anomalije ***Nepostojeći podaci***

Način rješavanja anomalije - zbog složenosti anomalije teško je provesti automatiziranje rješavanja, stoga je korisnika potrebno obavijestiti da je anomalija prepoznata u modelu i da ga je potrebno korigirati. Za rješavanje ove anomalije potrebno je ispitati da li aktivnost čitanja određenog podatka slijedi nakon aktivnosti upisivanja određenog podatka.

```
1  FIX: MissingDataAnomaly:
2     strMessage = ”There is anomaly missing data in process model”
```

Kôd 107. Način rješavanja anomalije ***Nepostojeći podaci***

Nekonzistentni podaci

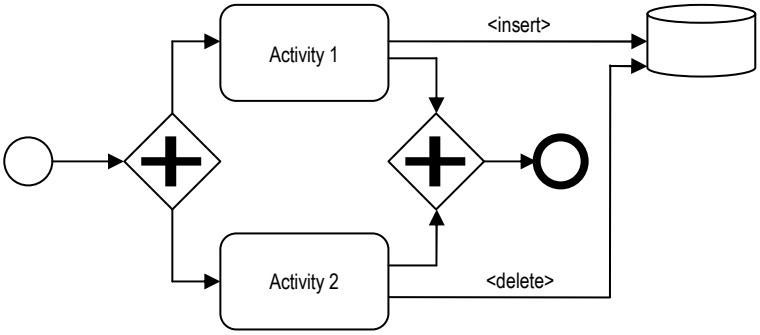
(eng. *Inconsistent data*)

Opis – anomalija ***Nekonzistentni podaci*** ako u model poslovnog procesa određeni radni korak koristi određene podatke (npr. određeni dokument), dok istovremeno drugi radni korak ili druga instanca iste aktivnosti uređuje ili briše taj podatak.

Autori Trčka, der Aalst i Sidorova u svojem radu *Data-Flow Anti-patterns: Discovering Data-FLow Errors in Workflows* [62] spominju anomaliju ***nekonzistentni podaci*** i smatraju je pravim propustom tijekom rada u poslovnom procesu.

U radu *Detecting DataFlow Errors u BPMN 2.0* autori Stackelberg, Putze, Mulle i Bohm opisuju anomaliju ***Nekonzistentni podaci*** ako postoje dva elementa koji su u konfliktu u slučaju kad jedan element želi pročitati određeni podatak, dok druga aktivnost želi zapisati taj

podatak u isti element podataka. U predloženoj taksonomiji anomalija ***Nekonzistentni podaci*** predstavljaju ***Strukturnu anomaliju***, koja je u grupi ***Anomalija pravila*** i koja je u podgrupi anomalija ***Tok podataka***.



Slika 78. Anomalija ***Nekonzistentni podaci***

Pravila prepoznavanja – prema kojoj se može prepoznati navedena anomalija su sljedeća:

- u modelu poslovnog procesa postoji barem jedan element baze podataka i
- u modelu poslovnog procesa postoji više aktivnosti koji upisuju isti podatak u objekt podataka.

Ako se navedena pravila prikažu u pseudokodu, onda to izgleda ovako:

```
1  RULE: InconsistentData:
2  IF: Count(Database.inputSequenceFlows)>1
3  AND Count(Activity.Insert)>0
4  AND Count(Activity.Delete)>0
5  AND Locked(Database.Data(ID))= True
6  THEN: InconsistentDataAnomaly(Database.Data(ID))
7  ENDIF
```

Kôd 108. Način prepoznavanja anomalije ***Nekonzistentni podaci***

Način rješavanja anomalije – za rješavanje ove anomalije trebalo bi uvesti mehanizam provjere da li je određeni slog zaključan ili nije. Također, preporuča se obavijestiti korisnika da u anomaliji postoji anomalija nepostojeći podaci kako bi mogao ispraviti model.

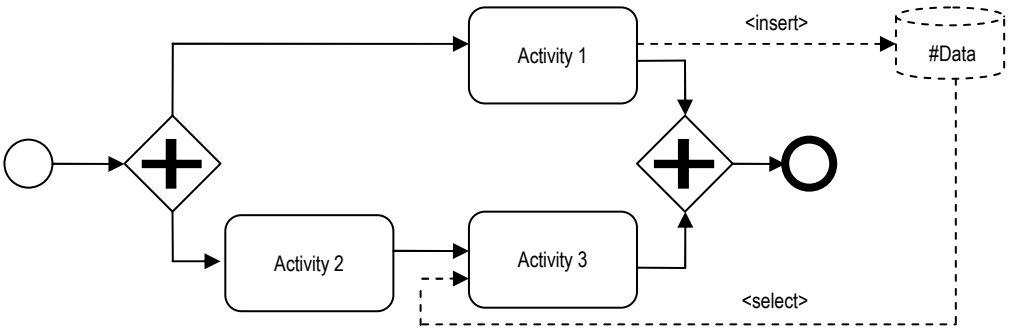
```
1  FIX: InconsistentDataAnomaly(Database.Data(ID)as DataRow):
2  IF Locked(Database.Data(ID))=True
3  THEN  strMessage = ”There is anomaly missing data in process model”
```

Kôd 109. Način rješavanja anomalije ***Nekonzistentni podaci***

Nikad izbrisani podaci

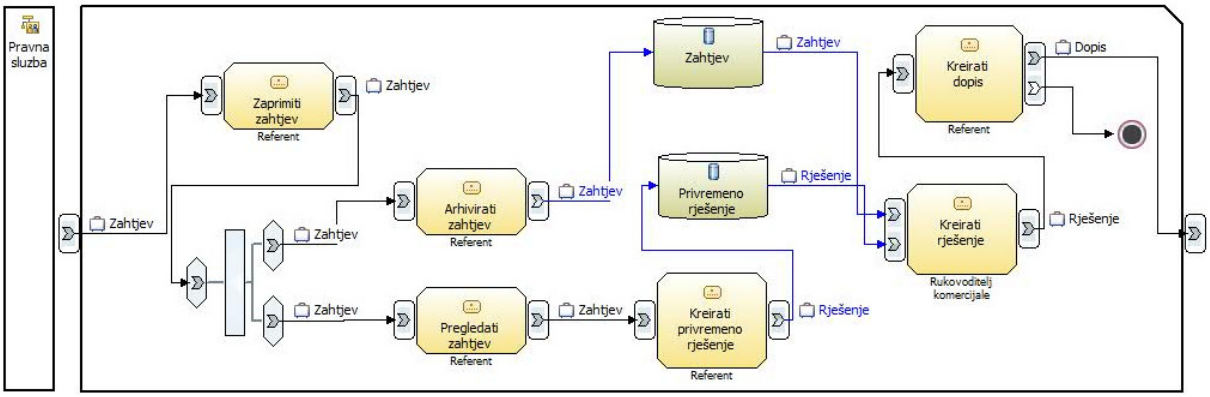
(eng. *Never destroyed data*)

Opis - anomalija ***Nikad izbrisani podaci*** pojavljuje se u scenariju u kojem se podaci kreiraju, no nikad se ne brišu nakon toga. Ovakve anomalije mogu prouzročiti nastajanje bespotrebnih podataka u poslovnom procesu. Ukoliko poslovni proces u pojedinoj aktivnosti kreira privremene podatke koji služe za podršku u pojedinim drugim aktivnostima, u tom slučaju nakon što posluže svojoj svrsi takvi privremeni podaci se moraju izbrisati ili pretvoriti u regularne podatke. Ova anomalija spominje se u više radova, dakle i kod autora Trčka, der Aalst i Sidorova u radu *Data-Flow Anti-patterns: Discovering Data-FLow Errors in Workflows* [62] i u radu *Detecting DataFlow Errors u BPMN 2.0* autora Stackelberg, Putze, Mulle i Bohm-a. U ovome slučaju privremenu bazu podataka i privremeni podatak označen je sa prefixom #. U predloženoj taksonomiji anomalija ***Nikad izbrisani podaci*** predstavljaju ***Strukturnu anomaliju***, koja je u grupi ***Anomalija pravila*** i koja je u podgrupi anomalija ***Tok podataka***.



Slika 79. Anomalija ***Nikad izbrisani podaci***

Primjer – na sljedećem primjeru može se vidjeti anomalija ***Nikad izbrisani podaci*** u modelu poslovnog procesa. Iz procesa se vidi aktivnost **Kreirati privremeno rješenje** koja sprema poslovnu stavku ***Rješenje*** u privremenu tablicu ***Privremeno rješenje***. Iz daljnjeg toka poslovnog procesa vidi se da aktivnosti **Kreirati rješenje** koriste podatke iz baze ***Privremeno rješenje***, no u tom poslovnom procesu nema aktivnosti brisanja tog privremenog podatka iz baze podataka.



Slika 80. Primjer anomalije ***Nikad izbrisani podaci***

Način prepoznavanja anomalije – navedena anomalija može se prepoznati na temelju sljedećih pravila:

- u modelu poslovnog procesa postoji barem jedan privremeni element baze podataka,
- u modelu poslovnog procesa postoji aktivnost koja privremeno sprema podatke u bazu podataka, a ne postoji aktivnost brisanja podataka iz privremene baze i
- u modelu poslovnog procesa postoji aktivnost koja privremeno sprema podatke u bazu podataka, a ne postoji aktivnost čitanja tih podataka onda se radi o anomaliji redundantnih podataka.

Kad se navedena pravila prikažu pseudokodom, onda to izgleda ovako:

```
1   RULE: NeverDestroyedData:
2   IF: Database.Type.Temp = True
3   AND Database.inputSequenceFlows())-> Not Empty
4   AND Database.outputSequenceFlows())-> Not Empty
5   AND Activity.Delete() NOT PRECEDE Activity.Read()
6   AND (Activity(ID).Read(DataObject.BusinesItem(ID))=
7        Activity(ID).Delete(DataObject.BusinessItem(ID))
8   THEN
9       NeverDestroyedDataAnomaly(Database)
10  ENDIF
```

Kôd 110. Način prepoznavanja anomalije ***Nikad izbrisani podaci***

Način rješavanja anomalije – navedena anomalija zahtjeva promjenu strukture modela, pa je potrebno korisnika obavijestiti da je anomalija pronađena u modelu kako bi mogao ispraviti model.

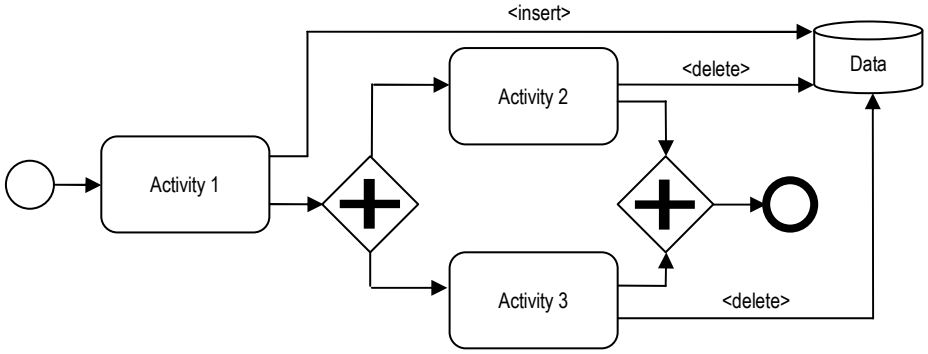
```
1   FIX: NeverDestroyedDataAnomaly(Database):
2   IF IFExist(Database)=True
3   THEN  strMessage = ”There is anomaly never destroyed data in process model”
4   ENDIF
```

Kôd 111. Način rješavanja anomalije ***Nikad izbrisani podaci***

Dva puta izbrisani podaci

(eng. *Twice destroyed data*)

Opis - ova anomalija nastaje ako je pojedini podatak dva puta izbrisan. Obično se to događa u poslovnim procesima u kojima postoje minimalno dvije aktivnosti koje brišu određeni podatak, a između ne postoji radni korak koji ga kreira. Ova anomalija slična je anomaliji ***Jako izgubljeni podaci*** i može se smatrati posebnim slučajem anomalije ***Jako izgubljeni podaci***. Kod ove anomalije treba definirati mehanizam provjere postojanja određenog podatka. U predloženoj taksonomiji anomalija ***Dva puta izbrisani podaci*** predstavljaju ***Strukturnu anomaliju***, koja je u grupi ***Anomalija pravila*** i koja je u podgrupi anomalija ***Tok podataka***.



Slika 81. Anomalija ***Dva puta izbrisani podaci***

Način prepoznavanja anomalije – navedena anomalija može se prepoznati uz sljedeća pravila:

- u modelu poslovnog procesa postoji element baze podataka i
- u modelu poslovnog procesa postoje barem dvije aktivnosti koje brišu određene slogove iz baze podataka.

Ako se navedena pravila prikažu u pseudokodu, onda to izgleda ovako:

```
RULE: LostData
IF Count(Database)>0.inputSequenceFlows()->notEmpty()
AND Count(Activity.Delete)=>2
AND Activity.insert() PRECEDE Activity.Delete()
AND Activity.Delete() PRECEDE Activity.Delete()
THEN LostDataAnomaly()
ENDIF
```

Kôd 112. Način prepoznavanja anomalije ***Dva puta izbrisani podaci***

Način rješavanja anomalije – kad se navedena anomalija prepozna u modelu poslovnog procesa, korisnika je potrebno obavijestiti da postoji u modelu i da je potrebno ispraviti model.

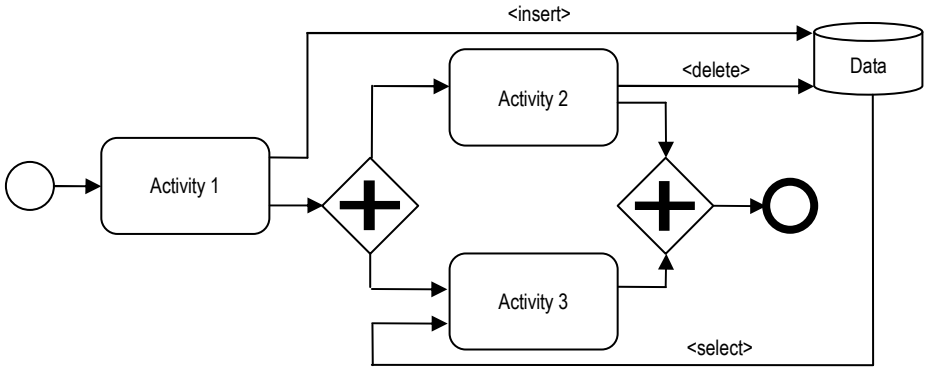
```
FIX: LostDataAnomaly:
     strMessage ="There is anomaly lost data in process model"
```

Kôd 113. Način rješavanja anomalije ***Dva puta izbrisani podaci***

Podaci neizbrisani na vrijeme

(eng. *Data not deleted on time*)

Opis - anomalija ***Podaci neizbrisani na vrijeme*** događa se kad određeni podatak nije obrisan na vrijeme, a druga aktivnost ga čita bez da je njegova vrijednost bila prethodno izbrisana. Kod ove anomalije je bitno da se nakon te aktivnosti podatak ponovo ne čita bez obzira kojim tokom poslovni proces završio svoje izvođenje. Sljedeći crtež pokazuje anomaliju ***Podaci neizbrisani na vrijeme***. Ako postoje dvije aktivnosti od kojih jedna aktivnost briše podatak, a druga ga čita, treba paziti da aktivnost koja čita podatak dohvati ispravnu vrijednost podatka. U predloženoj taksonomiji anomalija ***Podaci neizbrisani na vrijeme*** predstavljaju ***Strukturnu anomaliju***, koja je u grupi ***Anomalija pravila*** i koja je u podgrupi anomalija ***Tok podataka***.



Slika 82. Anomalija ***Podaci neizbrisani na vrijeme***

Primjer – scenarij gdje bi se navedena anomalija mogla ispoljiti je slučaj kad se kod neke duže obrade podataka u bazu podataka insertira slog s identifikatorom određene aktivnosti. Nakon što aktivnost završi s radom, slog se briše iz baze podataka. Druga aktivnost može započeti s izvođenjem tek kad je slog druge aktivnosti izbrisan. Navedeni mehanizam vrlo je čest kod provjere rada određene aktivnosti, problem nastupa kad se određeni podatak ne izbriše na vrijeme pa aktivnost koja ovisi o radu aktivnosti koja briše slog mora čekati da se određeni podatak izbriše iz baze podataka.

Način prepoznavanja anomalije – temelji se na sljedećim pravilima:

- u modelu poslovnog procesa mora postojati barem jedan element baze podataka,
- u modelu poslovnog procesa mora postojati barem jedan element aktivnosti koji briše podatak u bazi podataka,
- u modelu poslovnog procesa mora postojati barem jedna aktivnost koja čita podatak i
- u modelu poslovnog procesa postoji aktivnost čitanja koja mora biti poslije nakon aktivnosti brisanja.

Ako se navedena pravila prikažu u pseudokodu onda to izgleda ovako:

```
1  RULE: DataNotDeletedOnTime
2  IF Count(Database)>0
3  AND Count(Activity.Delete) => 2
4  AND Activity.Select PRECEDE Activity.Delete()
5  AND Activity.Delete() PRECEDE Activity.Delete()
6  THEN DataNotDeletedOnTimeAnomaly()
7  ENDIF
```

Kôd 114. Način prepoznavanja anomalije ***Podaci neizbrisani na vrijeme***

Način rješavanja anomalije – ako model poslovnog procesa sadrži navedenu anomaliju korisnika je potrebno obavijestiti o njenom postojanju da bi mogao ispraviti navedenu anomaliju.

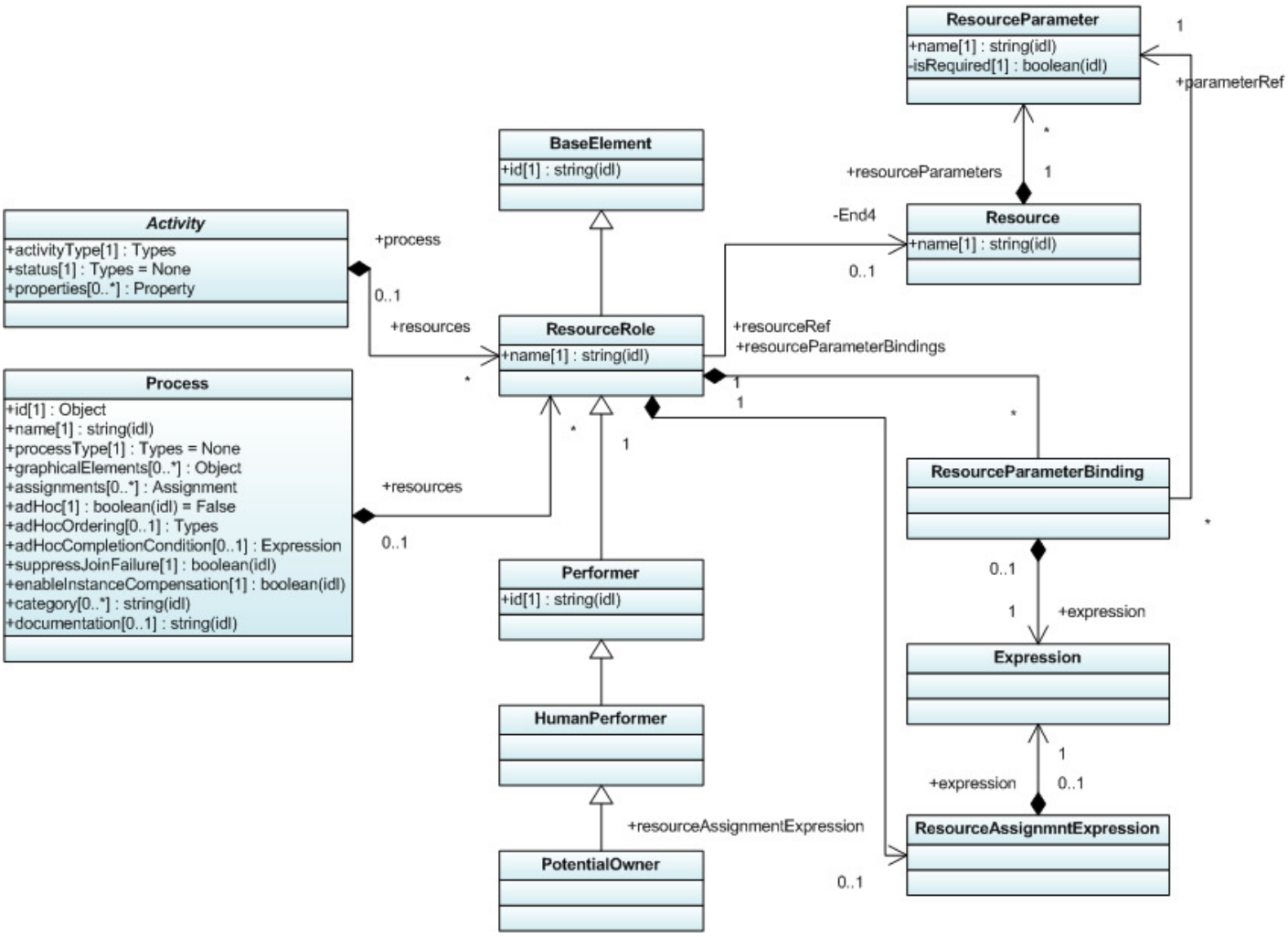
```
1  FIX: DataNotDeletedOnTimeAnomaly:
2       strMessage =“There is anomaly data not deleted on time in process model”
```

Kôd 115. Način rješavanja anomalije ***Podaci neizbrisani na vrijeme***

3.3. Anomalije resursa

BPMN 2.0 ima relativno slabu podršku za resurse. Na 52. stranici formalne specifikacije BPMN 2.0 notacije u točci 7.1. piše da je definiranje organizacijskog modela i resursa izvan područja BPMN 2.0 notacije. Samim time taj aspekt modeliranja poslovnih procesa je vrlo slabo zastupljen u BPMN 2.0 notaciji. Anomalije resursa predstavljaju jedan od originalnih doprinosa ovoga rada, jer uglavnom nisu obrađene u literaturi.

No, prisutnost grafičkih elemenata staza i bazena za prikaz uloga može davati kontradiktoran dojam. Staze i role nisu dovoljan mehanizam za pokrivanje svih potrebnih definicija resursa za njihovo kvalitetno upravljanje u poslovnim procesima pojedine organizacije. Na sljedećoj slici prikazan je BPMN 2.0 metamodel resursa, koji će se koristiti prilikom prepoznavanja i rješavanja anomalija resursa.



Slika 83. BPMN 2.0 metamodel resursa

Klasifikacija resursa

Na početku ovog poglavlja objašnjen je pogled na resurse i njihovu klasifikaciju koji se je koristio u okviru ove disertacije. Navedeni pogled na resurse nije univerzalni i detaljno razrađen, no dovoljan za potrebe ovoga rada.

Resursi se dijele na materijalne i ljudske. ***Ljudski resurs*** je dio organizacije. Organizacija predstavlja formalnu grupu resursa koja može poduzeti predmete rada s ciljem postizanja organizacijskih ciljeva. Ljudski resursi obično imaju određenu poziciju unutar organizacije. Obično ta određena pozicija nosi sa sobom određene karakteristike koje su više u relaciji s pozicijom nego sa samim karakteristikama resursa. Dakle, upravo ta pozicija nosi resursu određene privilegije. Određeni ljudski resurs može biti član jedne ili više organizacijskih jedinica koje su obično stalne i permanentne i koje poduzimaju određene oblike rada s ciljem postizanja zajedničkih ciljeva organizacije. Također, ljudski resursi mogu biti članovi jednog ili više organizacijskih timova. Organizacijski timovi su slični organizacijskim jedinicama, no po prirodi svojeg nastanka nisu stalnog karaktera već se formiraju i rasformiraju po potrebi. Ljudski resursi također, mogu biti pripadnici određenih cjelina kao što su divizije, sektori, odjeli, razine, itd. Svaka od navedenih klasifikacija određuju poziciju svakog pojedinog resursa u organizaciji. Iz tog proizlazi da resurs ima brojne veze i odnose s drugim resursima unutar organizacije. Osim toga svaki resurs izravno odgovara drugom resursu za svoj rad, taj drugi resurs je odgovoran za rad podređenog resursa. Obično se radi o starijem resursu na višoj organizacijskoj razini. Stariji resursi obično imaju nekoliko podređenih za koje su odgovorni.

Svaki resurs ima i tzv. zamjenu ili delegata koji je alternativa resursu u slučaju da resurs ne može obaviti prethodno dodijeljen posao. Preusmjeravanje posla može biti privremeno ili trajno. Resurs može imati jednu ili više pridruženih rola. Role služe kao još jedan mehanizam grupiranja.

Jedno od obilježja resursa je i kvalifikacija za obavljanje određenog posla. To može biti određeno znanje ili vještine kao i drugi osobni atributi pojedinačnog resursa. Navedeni atributi su vrlo važni kod alokacije predmeta rada.

Materijalni resursi mogu biti trajni ili potrošni. Trajni materijalni resursi su oni čiji kapacitet ne opada radom, dok se potrošni resursi troše tokom rada i to u cijelosti ili djelomično. Potrošni resursi imaju stupanj potrošnje ili kapacitet koji određuje koliko se rada s navedenim resursom može obaviti prije nego je potrebna zamjena.

Svaki resurs može imati raspored ili povijest tj. „log“ u kojima se vode prethodni radovi koje je resurs obavio. Raspored je lista radova za koje se je resurs obavezao da će napraviti u budućnosti. Log sadrži bilješke o radovima koje je resurs uspješno ili neuspješno obavio u prošlosti.

Ako se neki resurs dodijeli nekoj aktivnosti, onda je izvršenje te aktivnosti usko vezana uz raspoloživost dodijeljenih resursa. Kalendar resursa omogućava finu raspodjelu i dodjeljivanje resursa pojedinim resursima po različitim poslovnim procesima i po više instanci. Informacije spremljene unutar kalendara resursa bitne su za učinkovito izvršenje aktivnosti, a isto su vrlo korisne i u svrhu rasporeda. Raspoloživost resursa određeno je njegovim statusom. Za potrebe klasifikacije status resursa se označava sa: raspoloživ, neraspoloživ, zauzet, rezerviran, blokiran, itd. Status resursa može se definirati kao stanje raspoloživosti resursa. Resurs može biti u određenom stanju iz nekog razloga. Kod raspoloživosti resursa u obzir se mora uzeti i lokacija resursa, naime neki resurs se ne može upotrijebiti u drugoj aktivnosti samo, jer je završio s prethodnom aktivnošću, nego treba vidjeti da li je taj resurs i na ispravnoj lokaciji. Resurs mora biti dostupan i spreman za izvršenje aktivnosti. Resursi mogu biti predmetom interesa poslovnih pravila i to treba uzeti u obzir tokom planiranja poslovnih procesa. To se može odrediti pravilima promjena stanja. Pravila promjena stanja određuju ograničenja kako se resurs mijenja kad prelazi iz jednog stanja u drugi.

Kada se govori o anomalijama resursa u modelima poslovnih procesa, tada se mora posebno usmjeriti pažnju prema alokaciji resursa.

Alokacija resursa se koristi za dodjelu raspoloživih sredstava na isplativ način. Alokacija resursa sastavni je dio upravljanja resursima i predstavlja raspoređivanje materijala, ljudstva i ostalih sredstava potrebnih za izvršenje pojedine aktivnosti. Alokacijom resursa pojedinoj aktivnosti na raspolaganje se stavljaju resursi s ciljem izvršenja aktivnosti. Ciljevi alokacije resursa su: efikasnost, pravičnost i distribucija odgovornosti. Daljnji ciljevi alokacije: financijski plan, kontrola troškova, planiranje kadrova, odgovornost za poslovne rezultate.

Alokacijska efikasnost je situacija kad ne postoji nikakav način da se poslovni proces preoblikuje na način da poveća iskoristivost jednog resursa nauštrb drugog. Drugim riječima poslovni proces je u savršenoj ravnoteži što se tiče iskoristivosti resursa.

Tehnička efikasnost je odnos rezultata (output-a) i ulaganja (input).

Horizontalna pravičnost je ideja da resursi jednakih sposobnosti i znanja budu jednako opterećeni i da prema tome budu jednako i plaćeni.

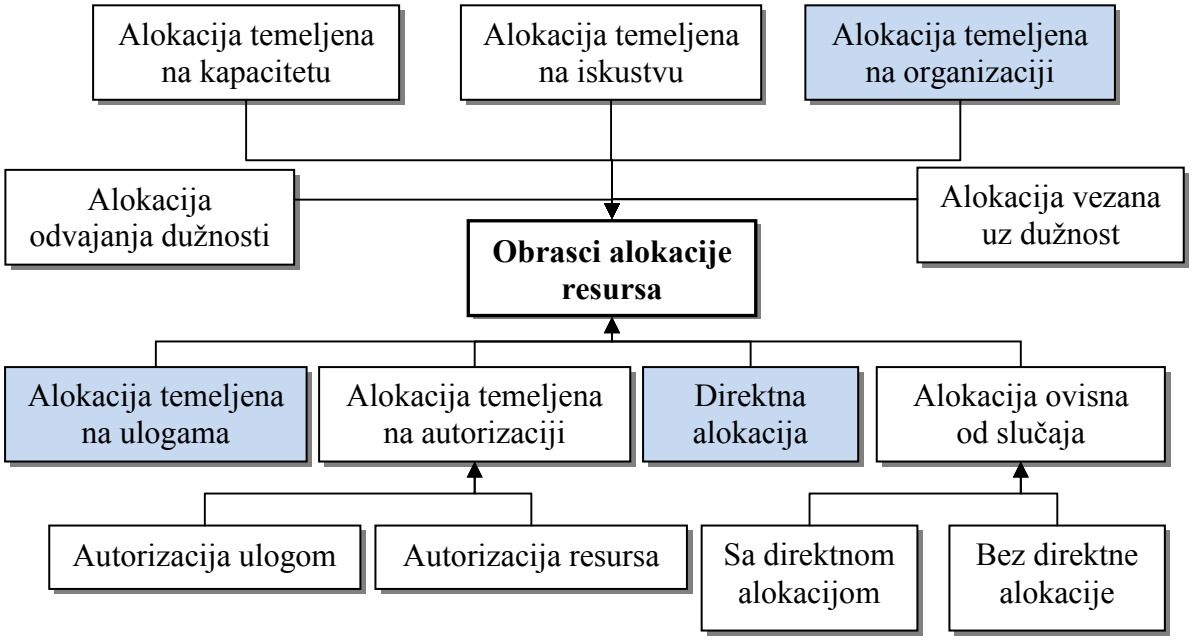
Vertikalna pravičnost odnosi se na ideju da bolji resursi s većim opterećenjem i većim sposobnostima moraju biti i bolje plaćeni, ova ideja funkcionira i obrnuto, da resursi sa manjim opterećenjem i slabijim sposobnostima moraju biti i manje plaćeni.

Distribucija odgovornosti omogućava raspoređivanje odgovornosti na sve sudionike ovisno o njihovom angažmanu.

Vrste alokacija prema obrascima resursa autora Russell, Hofstede, Edmond i Aalst su [51]:

- **Direktna alokacija** – je sposobnost određivanja identiteta resursa koji će izvršiti određeni zadatak tokom oblikovanja rada.
- **Alokacija temeljena na ulogama/rolama** – je sposobnost određivanja tijekom oblikovanja rada da će određenu aktivnost biti izvedeno od nekog resursa koji ima obilježja određene uloge/role.
- **Odgođena alokacija** - je vrsta alokacije u kojoj se posao dodjeljuje naknadno, dakle ne u vrijeme projektiranja poslovnog procesa, već kasnije kod pokretanja poslovnog procesa ili tijekom izvođenja.
- **Alokacija temeljena na kapacitetu** – alokacija se izvodi na temelju određenih sposobnosti resursa. Podrška ovakvom tipu alokacije zahtjeva postojanje informacija o pojedinim resursima.
- **Alokacija temeljena na iskustvu** – resurs se alocira na temelju njihovog prethodnog iskustva. Podrška ovakvom alociranju zahtjeva postojanje povijesti prethodnih zadataka pojedinih resursa.
- **Alokacija temeljena na organizaciji** – kod ovog tipa alokacije resurs se alocira na osnovu pozicije resursa unutar organizacije. Ovakva alokacija zahtjeva postojanje organizacijskog modela sa pozicijama i odnosima. Alokacija može biti određena na temelju apsolutne i relativne pozicije resursa u organizacijskom modelu.
- **Slučajna alokacija** – je alokacija na temelju slučajnog odabira.
- **Alokacija inicirana od strane resursa** – događa se u okolinama u kojima postoji lista raspoloživih poslova i kod kojih resurs sam može inicirati izvođenje određenog zadatka. Ovakve vrste alokacija uobičajene su kod resursa koji su plaćeni ovisno o obavljenom poslu.
- **Alokacija kružnim dodjeljivanjem** – koristi se kad se opterećenje rada distribuira po resursima tako da su ravnomjerno opterećeni. U tom slučaju treba voditi zabilješke o tome tko je zadnji puta neki posao obavljao te tko sljedeći treba taj posao obaviti.

U radu *Enabling Resource Assignment Constraints in BPMN* [6] autori: Awad, Grosskopf, Meyer i Weske postavljaju odnose između pojedinih tipova alokacija.



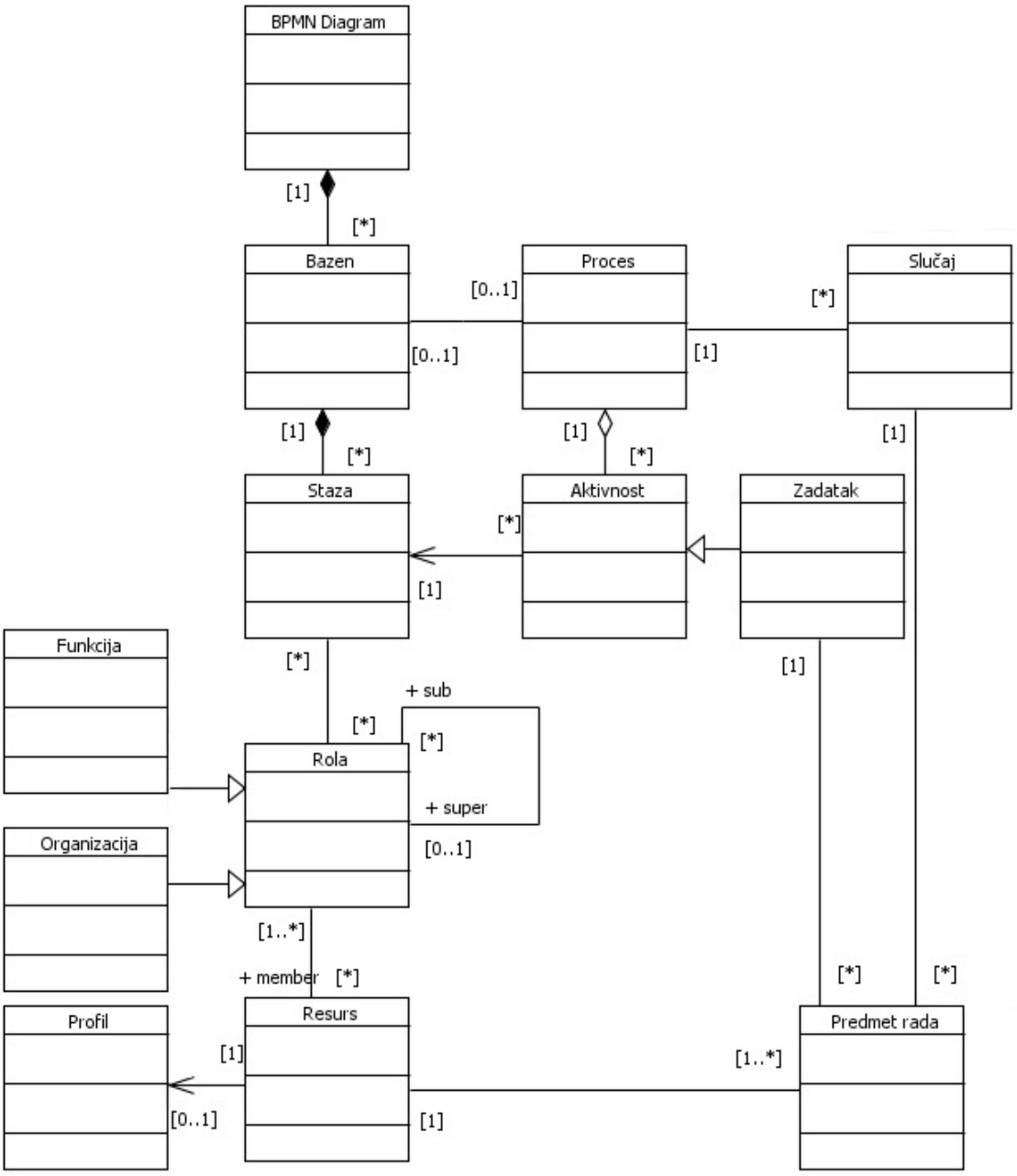
Slika 84. Odnos između obrazaca alokacija resursa
(Izvor: Awad, A; Grosskopf, A, Meyer, A, Weske, M: *Enabling Resource Assigment Contraints in BPMN*, Hasso Plattner Institute, Potsdam, Germany)

Od navedenih alokacija BPMN 2.0 podržava:

- direktnu alokaciju,
- alokaciju temeljenu na ulogama/rolama i
- alokaciju temeljenu na organizaciji.

Direktna alokacija je podržana direktno imenovanjem određenog resursa kod definicije pojedinog radnog koraka. **Organizacijska alokacija** je podržana preko atributa resursa aktivnosti koji određuje resurs koji može realizirati određeni zadatak. Resursi koji su uneseni na taj način mogu između ostalog imati neku organizacijsku ulogu. S obzirom da resurs u tom slučaju predstavlja tekstualni podatak bez jedinstvenog identifikatora problem nastaje kad se takav resurs koristi u kompleksnim okolinama. Iz navedenog može se zaključiti da postojeći metamodel BPMN 2.0 notacije u potpunosti podržava samo direktnu alokaciju, dok su druga dva oblika izvedena iz postojećeg mehanizma direktne alokacije.

Alokacija temeljena na ulogama/rolama je pokrivena uporabom entiteta role, jer postoji razlika između organizacijske i funkcionalne role.



Slika 85. Proširen BPMN metamodel sa podrškom za alokaciju resursa
(Izvor: Awad, A; Grosskopf, A, Meyer, A, Weske, M: *Enabling Resource Assigment Contraints in BPMN*, Hasso Plattner Institute, Potsdam, Germany)

Anomalije resursa obično su vezane uz alokaciju rada resursa. Vrlo ih je teško prikazati grafički, pa se obično prikazuju opisno. Anomalije resursa mogu se podijeliti na:

- anomalije obrazaca kreiranja,
- anomalije obrazaca guranja,
- anomalije obrazaca povlačenja,
- anomalije obrasca zaobilaženja,

- anomalije samopokretajućih obrazaca,
- anomalije obrasca vidljivosti i
- anomalije višestrukih obrazaca resursa.

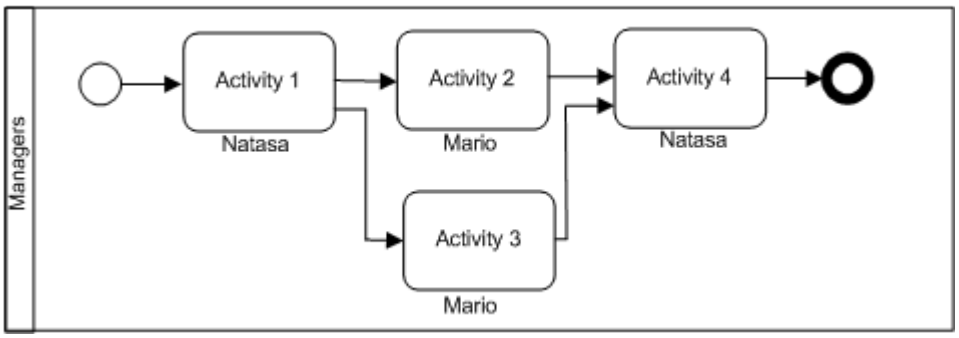
3.3.1. Anomalije obrazaca kreiranja

Anomalije obrazaca kreiranja predstavljaju zajedničku grupu ***anomalija resursa*** u koju spadaju anomalije: ***Direktna alokacija u slučaju da više resursa ima isti naziv***; ***Direktna alokacija resursa bez mehanizma eskalacije***; ***Alokacija temeljena na ulogama kad odabrani resurs nije pripadnik određene uloge/role***; ***Alokacija temeljena na ulogama kad resurs pripada u više od jedne grupe***; ***Alokacija temeljena na ulogama kod grupe sa jednim resursom*** i ***Automatsko izvođenje bez mehanizma provjere izvođenja***.

Direktna alokacija u slučaju da više resursa ima isti naziv

(eng. *Direct allocation where resource have non-unique name*)

Opis - BPMN 2.0 notacija podržava direktnu alokaciju na način, da se za svaku pojedinu aktivnost odabire resurs poimence. No, problem nastaje ako više resursa ima isti naziv, u tom slučaju može se dogoditi da se odabere krivi resurs. Anomalija ***Direktna alokacija u slučaju da više resursa ima isti naziv*** prema predloženoj taksonomiji spada u ***Anomalije resursa*** u grupu anomalija ***Anomalije obrazaca kreiranja***.



Slika 86. Primjer anomalije ***Direktna alokacija u slučaju da više resursa ima isti naziv***

Način prepoznavanja anomalije – ova vrsta anomalija može nastati samo ako postoje različiti resursi istog nazivlja.

Prema navedenom, temeljno pravilo po kojem se može prepoznat navedeno pravilo je:

- u modelu procesa postoji više različitih resursa s istim nazivom.

Ako se navedena pravila prikažu kao OCL pravilo, onda to izgleda ovako:

```
1  context Resource
2  inv ResourceNonUniqueName:
3     (IsUnique(Resource.Name)=False)
4         implies
5         ResourceNonUniqueNameAnomaly()
```

Kôd 116. Način prepoznavanja anomalije ***Direktna alokacija u slučaju da više resursa ima isti naziv***

Način rješavanja anomalije - ovu anomaliju najbolje je izbjeći ugradnjom ograničenja u metamodel BPMN 2.0 notacije. OCL pravilom treba definirati pravilo koje će osigurava da će određeni radni korak odraditi resurs s jedinstvenim identifikatorom.

```
1  context Resource
2  inv ResourceNonUniqueNameAnomaly:
3         IsUnique(Resource.Name)=True
```

Kôd 117. Način rješavanja anomalije ***Direktna alokacija u slučaju da više resursa ima isti naziv***

Direktna alokacija resursa bez mehanizma eskalacije

(eng. *Direct resource allocation without escalation*)

Opis - obrazac direktne alokacije pojedinog resursa funkcionira dobro kad resurs neometano izvršava svoje zadatke. No, ako iz nekog razloga resurs ne može obaviti zadane zadatke, direktna alokacija nema mehanizam preusmjeravanja zadataka drugim resursima. Time u poslovnom procesu nastaje usko grlo, što utječe na performanse poslovnog procesa. Kao primjer, može se uzeti Slika 87. na kojoj se vidi, da su aktivnosti dodijeljene pojedinim resursima. Ukoliko jedan od resursa ne može izvršiti svoj posao u poslovnom procesu nastaje usko grlo i poslovni proces se ne može izvršiti do kraja. Upravo zbog toga kod direktne alokacije resursa potrebno je postaviti mehanizam eskalacije zadataka i preusmjeravanja zadataka drugim slobodnim resursima. Anomalija ***Direktna alokacija resursa bez mehanizma eskalacije*** prema predloženoj taksonomiji spada u ***Anomalije resursa*** u grupu anomalija ***Anomalije obrazaca kreiranja***.

Primjer – prikazuje obrazac ***Direktne alokacije resursa bez mehanizma eskalacije.***

Slika 87. Anomalija ***Direktna alokacija resursa bez mehanizma eskalacije***

Način prepoznavanja anomalije - mehanizam eskalacije trebao bi u slučaju prepoznavanja zastoja realocirati resurse za određenu aktivnost. Mehanizam eskalacije bi trebao biti definiran u posebnoj proceduri, koja bi se izvodila u slučaju prepoznatog zastoja. Zastoj se najčešće može prepoznati po tome da se pojedina aktivnost nikad ne realizira, odnosno završi. Tako da se može izvesti sljedeće pravilo, model procesa ima anomaliju ***Direktne alokacije resursa bez mehanizma eskalacije*** ako:

- u modelu poslovnog procesa postoji aktivnost koje nema određeno maksimalno vrijeme trajanja i
- u modelu poslovnog procesa ne postoji aktivnost koja se aktivira ako istekne maksimalno vrijeme trajanja neke aktivnosti.

Ako se navedeno pravilo prikaže kao OCL pravilo, onda to izgleda ovako:

```
1  context Activity
2  inv DirectAlocationWithoutEscalation:
3         (Activity.MaxDuration is null) and (Activity.WhatElse is null)
4         implies
5         DirectAlocationWithoutEscalationAnomaly()
```

Kôd 118. Način prepoznavanja anomalije ***Direktna alokacija resursa bez mehanizma eskalacije***

Način rješavanja anomalije – ovu anomaliju najbolje je riješiti dodavanjem maksimalnog vremena čekanja da se pojedina aktivnost završi. Ako se ne izvrši potrebno je odrediti alternativnu sekvencu koja predstavlja mehanizam eskalacije.

```
1  context Activity
2  inv DirectAlocationWithoutEscalationAnomaly:
3         Activity.MaxDuration is not null
```

Kôd 119. Način rješavanja anomalije ***Direktna alokacija resursa bez mehanizma eskalacije***

Alokacija temeljena na ulogama kad odabrani resurs nije pripadnik određene uloge/role

(eng. *Role-based allocation when resource is not in role*)

Opis – ovakva vrsta anomalije može se dogoditi ukoliko se odabere resurs koji nema dodijeljenu predviđenu ulogu za taj radni korak. Anomalija ***Alokacija temeljena na ulogama kad odabrani resurs nije pripadnik određene uloge/role*** prema predloženoj taksonomiji spada u ***Anomalije resursa*** u grupu anomalija ***Anomalije obrazaca kreiranja***.

Način prepoznavanja anomalije – ova anomalija se prepoznaje ako je sljedeće pravilo zadovoljeno:

- u modelu poslovnog procesa postoji aktivnost koji ima pripadajući resurs čija rola nije sukladna zahtjevima određene aktivnosti.

Ako se navedeno pravilo prikaže pomoću OCL jezika, onda to izgleda ovako:

```
1  context Activity
2   inv roleBasedAllocationResourceNotInRole:
3    self.Resource->forAll(x | x.role->select(self.ResourceRole.Requirement)->size()=0)
4     implies
5      roleBasedAllocationResourceNotInRoleAnomaly()
```

Kôd 120. Način prepoznavanja anomalije ***Alokacija temeljene na ulogama kad odabrani resurs nije pripadnik određene uloge/role***

Način rješavanja anomalije - oslanja se na mehanizam OCL ograničenja koji dozvoljava samo određenim resursima da budu vidljivi tijekom odabira potrebnog resursa u određenom koraku.

```
1  context Activity
2  inv roleBasedAllocationResourceNotInRoleAnomaly:
3     self.Resource ->
4     forAll(x | x.role->select(self.ResourceRole.Requirement)->size()>0)
```

Kôd 121. Način rješavanja anomalije ***Alokacija temeljene na ulogama kad odabrani resurs nije pripadnik određene uloge/role***

Alokacija temeljena na ulogama kad resurs pripada u više od jedne grupe

(eng. *Role-based allocation when resource belongs to many groups*)

Opis - grupe predstavljaju apstraktni mehanizam kategorizacije resursa, iz čega proizlazi da pojedini resurs može biti svrstan u više različitih grupa. Ako resurs pripada u više različitih grupa, u tom slučaju je moguće da kod alokacije temeljene na ulogama pojedini resurs može dobiti zadatak od strane više različitih grupa u koje je uključen. Time se resurs opterećuje i performanse poslovnog

procesa se mogu pogoršati. Anomalija ***Alokacija temeljena na ulogama kad resurs pripada u više od jedne grupe*** prema predloženoj taksonomiji spada u ***Anomalije resursa*** u grupu anomalija ***Anomalije obrazaca kreiranja***.

Primjer – prikazuje anomaliju ***Alokacija temeljena na ulogama kad resurs pripada u više od jedne grupe.***

Slika 88. Anomalija ***Alokacija temeljena na ulogama kad resurs pripada u više od jedne grupe***

Način prepoznavanja anomalije – navedena anomalija prepoznaje se po sljedećem pravilu:

- u modelu poslovnog procesa pojedinim aktivnostima zadani su resursi koji se nalaze u više od jedne role.

Ako se navedeno pravilo opiše OCL jezikom, onda to izgleda ovako:

```
1  context Activity
2   inv roleBasedAllocationResourceInManyGroups:
3      self.Resource->
4      forAll(x.role->select(name = self.Resource.Name)->size()>1)
5       implies
6       roleBasedAllocationResourceInManyGroupsAnomaly()
```

Kôd 122. Način prepoznavanja anomalije ***Alokacija temeljena na ulogama kad resurs pripada u više od jedne grupe***

Način rješavanja anomalije - kod odabira resursa treba izlistati samo one resurse koji imaju jednu rolu.

```
1  context Activity
2  inv roleBasedAllocationResourceNotInGroupsAnomaly:
3    self.Resource -> forAll(x | x.role->select(self.ResourceRole.Requirement)->size()=1)
```

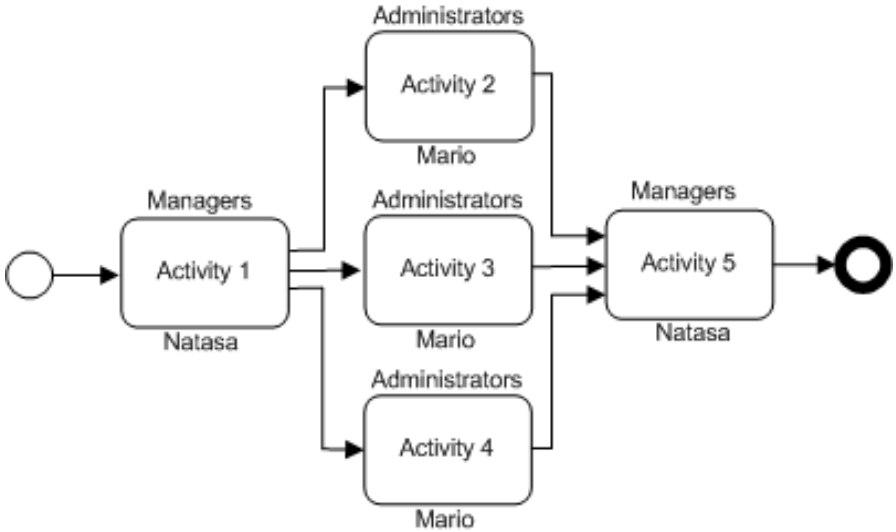
Kôd 123. Način rješavanja anomalije***Aalokacije temeljene na ulogama kad odabrani resurs nije pripadnik određene uloge/role***

Alokacija temeljena na ulogama kod grupe sa jednim resursom

(eng. *Role-based allocation when role have single resource*)

Opis – kod anomalije ***Alokacije temeljene na ulogama kod grupe s jednim resursom*** alokacija temeljena na grupama predstavlja direktnu alokaciju, pa ukoliko se ne postavi mehanizam preusmjeravanja zadataka pojavljuje se slučaj ***Anomalije direktne alokacije resursa bez mehanizma eskalacije***. Anomalija ***Alokacije temeljene na ulogama kod grupe s jednim resursom*** prema predloženoj taksonomiji spada u ***Anomalije resursa*** u grupu anomalija ***Anomalije obrazaca kreiranja***.

Primjer – prikazuje anomaliju ***Alokacije temeljene na ulogama kod grupe s jednim resursom.***



Slika 89. Primjer ***Alokacije temeljene na ulogama kada pojedina grupa ima samo jedan resurs***

Način prepoznavanja anomalije – prepoznavanje navedene anomalije može se izvesti pomoću sljedećeg pravila:

- u modelu poslovnog procesa postoje aktivnosti koje se izvode paralelno, a koje imaju dodijeljene role sa samo jednim resursom.

Ako se navedeno pravilo prikaže OCL pravilom onda to izgleda ovako:

```
context Process
 inv roleBasedAllocationRoleWithSingleResource:
  Count(Fork)>0 and
  Activity.Resource ->
  forAll(x | x.role -> select(self.ResourceRole.Requirement)->size()=1)
    implies
    roleBasedAllocationRoleWithSingleResourceAnomaly()
```

Kôd 124. Način prepoznavanja anomalije ***Alokacija temeljene na ulogama kada pojedina grupa ima samo jedan resurs***

Način rješavanja anomalije – navedena anomalija rješava se na način da se ne smije dozvoliti odabir role s jednim resursom kod više paralelnih aktivnosti.

```
1 context Process
2  inv roleBasedAllocationRoleWithSingleResourceAnomaly:
3   Count(Fork)>0 and
4   Activity.Resource ->
5   forAll(x | x.role -> select(self.ResourceRole.Requirement)->size()>1)
```

Kôd 125. Način rješavanja anomalije ***Alokacije temeljene na ulogama kada pojedina grupa ima samo jedan resurs***

Automatsko izvođenje bez mehanizma provjere izvođenja

(eng. *Automatic execution without status check*)

Opis - u slučaju da je u poslovnom procesu definirano automatsko izvođenje pojedinog zadatka, potrebno je ugraditi mehanizam provjere da li se je zadatak izveo ili ne, te u slučaju da se zadatak nije izveo da se automatski izvođenje pokrene. Taj mehanizam također, mora imati provjeru da li je izvođenje zadatka u tijeku te po potrebi provjeriti u kojoj aktivnosti izvođenje traje duže od očekivanog, te po potrebi uključiti alternativni tok izvođenja. Navedena anomalija nije statička, već ulazi u područje dinamičke analize modela poslovnog procesa, jer mehanizam provjere izvođenja uzima u obzir stanje modela tijekom izvođenja. Anomalija ***Automatsko izvođenje bez mehanizma povjere izvođenja*** prema predloženoj taksonomiji spada u ***Anomalije resursa*** u grupu anomalija ***Anomalije obrazaca kreiranja***.

Način prepoznavanja anomalije – s obzirom da je ova anomalija dinamičke prirode, prepoznati se može samo simulacijom modela poslovnog procesa. Posljedice nepostojanja mehanizma provjere izvođenja ogleda se u nekoliko mogućih stanja aktivnosti. Aktivnost može biti u stanjima:

- **ActivityFinished** – rad aktivnosti je završio,
- **ActivityStuck**, - aktivnost je zapela,
- **ActivityRuning** – aktivnost se izvršava i
- **ActivityPending** – aktivnost čeka na izvršenje.

Način rješavanja anomalije – za rješavanje ove anomalije moraju postojati funkcije provjere stanja pojedine aktivnosti. Obično se tu radi o funkciji npr. **`GetStatus(Activity)`** koja vraća trenutno stanje aktivnosti. Procedura izvođenja pojedinih aktivnosti mora znati koja

aktivnost se mora u kojem trenutku pokrenuti, te u kojem je stanju pojedina aktivnost. Algoritam za prepoznavanje navedene anomalije izgledao bi ovako:

Funkcija **GetStatus(Activity)** morala bi izgledati ovako:

- **Da li se je aktivnost već izvršila?**
  - **Da** - ActivityFinished
  - **Ne**
    - **Da li se aktivnost izvršava?**
      - **Da** - provjeriti da li se izvršava predugo?
        - **Da** - ActivityStuck
        - **Ne** - ActivityRuning
      - **Ne** – provjeriti da se je aktivnost već izvršila?
        - **Da** - ActivityFinished
        - **Ne** - ActivityPending

Na temelju rezultata funkcije GetStatus(Activity) poduzimaju se određeni koraci, i to;

- ako je **GetStatus(CurrentActivity) = ActivityFinished** onda idi na sljedeću aktivnost,
- ako je **GetStatus(CurrentActivity) = ActivityStuck** onda javi grešku,
- ako je **GetStatus(CurrentActivity) = ActivityRuning** onda čekaj da se aktivnost izvrši i
- ako je **GetStatus(CurrentActivity) = ActivityPending** onda idi na sljedeću aktivnost.

3.3.2. Anomalije samopokretajućih obrazaca

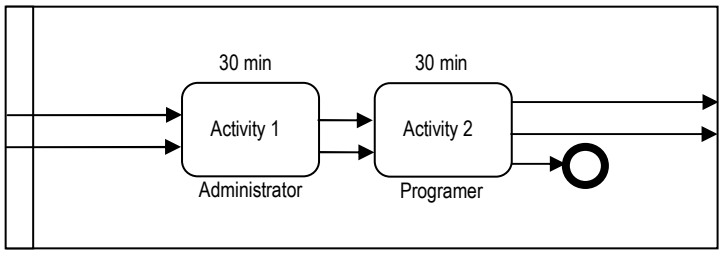
Anomalije samopokretajućih obrazaca prema predloženoj taksonomiji spadaju u ***Anomalije resursa*** i kao kategorija sadrže anomalije: ***Početak kreiranja sa alokacijom resursa*** i ***Djelomično lančano izvršenje u poslovnom procesu.***

Početak kreiranja sa alokacijom resursa

(eng. *Creation with resource allocation*)

Opis - ako resurs može početi raditi na predmetu rada odmah nakon što predmet rada postane dostupan, u tom slučaju alokacija resursa nije potrebna, već resurs odmah može početi s radom. Alokacija resursa u tom slučaju predstavlja gubitak vremena s obzirom da resurs ima sve potrebno za početak aktivnosti. Dakle problem je da anomalija ***Početak kreiranja sa alokacijom resursa*** ima alokaciju resursa iako resurs može početi s radom odmah nakon što predmet rada postane dostupan. Prema predloženoj taksonomiji anomalija ***Početak kreiranja sa alokacijom resursa*** predstavlja ***Anomaliju resursa*** iz grupe anomalija ***Anomalije samopokretajućih obrazaca***.

Primjer – prikazuje anomaliju ***Početak kreiranja sa alokacijom resursa.***



Slika 90. Anomalija ***Početak kreiranja sa alokacijom resursa***

Način prepoznavanja anomalije – navedena anomalija prepoznaje se po sljedećim pravilima:

- u modelu poslovnog procesa ne postoji start event, nego se proces okida izvana,
- u modelu poslovnog procesa postoji aktivnost ima vrijednost čekanja resursa (eng. *ResourceWaitTime*) veći o nule i
- u modelu poslovnog procesa navedena aktivnost odmah dobiva sve poslovne stavke potrebne za rad.

Ako se navedeno pravilo prikaže OCL jezikom, onda to izgleda ovako:

```
1 context Process
2  inv CreationWithResourceAllocation:
3   Count(StartEvent)=0 and
4   Acivity.ResourceWaitTIme>0 and
5   Activity.Resource ->
6   forAll(x | x.role -> select(self.ResourceRole.Requirement)->size()>0)
7    implies
8    CreationWithResourceAllocationAnomaly()
```

Kôd 126. Način prepoznavanja anomalije ***Početak kreiranja sa alokacijom resursa***

Način rješavanja anomalije – u slučaju da je prepoznata anomalija ***Početak kreiranja sa alokacijom resursa***, treba vrijeme čekanja resursa postaviti na nulu.

```
1 context Process
2  inv CreationWithResourceAllocationAnomaly:
3   Acivity.ResourceWaitTIme = 0
```

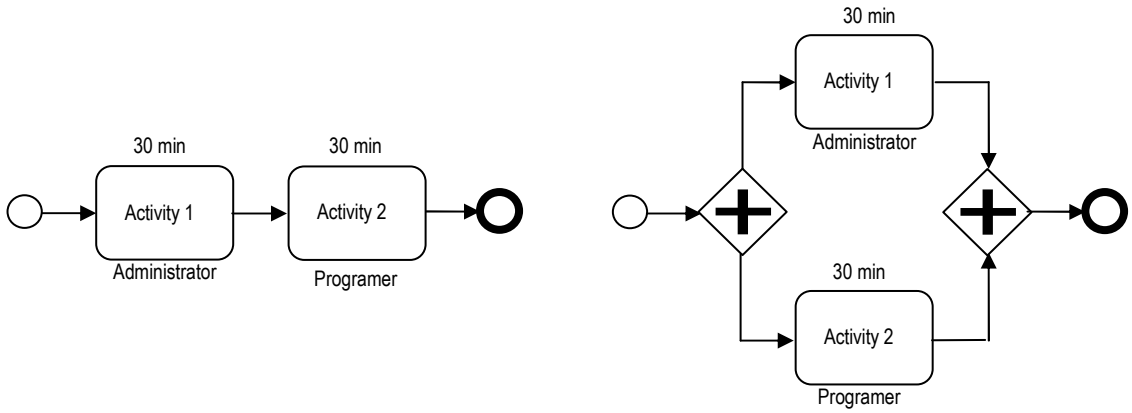
Kôd 127. Način prepoznavanja anomalije ***Početak kreiranja sa alokacijom resursa***

Djelomično lančano izvršenje u poslovnom procesu

(eng. *Partly chained execution*)

Opis - lančano izvršenje ima smisla ako je cijeli poslovni proces organiziran na taj način, ako samo dio poslovnog procesa ne podržava ovaj način rada onda lančano izvršenje u tom poslovnom procesu nema smisla. Nastaju uska grla koja utječu na performanse poslovnog procesa. U slučaju da organizacija ima dovoljni kapacitet, nema razloga zašto se pojedini dijelovi ne bi odvijali paralelno, čime bi se smanjilo vrijeme izvođenja cijelog procesa. Prema predloženoj taksonomiji anomalija ***Djelomično lančano izvođenje u poslovnom procesu*** predstavlja ***Anomaliju resursa*** iz grupe anomalija ***Anomalije samopokretajućih obrazaca***.

Primjer – na primjeru se s lijeve strane vidi poslovni proces u kojem se vidi sekvencionalno izvođenje aktivnosti koja međusobno ne ovise jedna o drugoj i koja se izvode od strane različitih resursa.



Slika 91. Anomalija ***Djelomično lančano izvršenje u poslovnom procesu i rješenje***

Način prepoznavanja anomalije – temelji se na sljedećim pravilima:

- u modelu poslovnog procesa postoje više aktivnosti koje se odvijaju sekvencijalno,
- u modelu poslovnog procesa ne postoji element razdvajanja i stapanja sekvenci,
- u modelu poslovnog procesa svaku od navedenih aktivnosti obavlja različit resurs,
- u modelu poslovnog procesa navedene aktivnosti ne ovise slijedno jedna od drugoj i
- u modelu poslovnog procesa postoje dovoljni kapacitet resursa da se aktivnosti odvijaju paralelno.

Ako se navedena pravila prikažu pomoću OCL jezika, onda to izgleda ovako:

```
1  context Process
2   inv PartlyChainedExecution:
3    Count(self.Activities)>1 and
4    Count(self.Forks)=0 and
5    Activity PRECEDE Activity
6    Activity.Resource ->
7    forAll(x | x.role -> select(self.ResourceRole.Requirement)->size()>1)
8    implies
9       PartlyChainedExecutionAnomaly()
```

Kôd 128. Način prepoznavanja anomalije ***Djelomično lančano izvršenje u poslovnom procesu***

Način rješavanja anomalije – ovakav tip anomalije zahtjeva intervenciju korisnika u samu strukturu modela, naime neke od aktivnosti koje sad slijede jedna iza druge, potrebno je izmodelirati paralelno. Automatizacija takvih postupaka je vrlo složena, pa se u ovome tekstu neće detaljnije razrađivati.

```
1  Sub PartlyChainedExecutionAnomaly()
2    strMessage = "There are partly chained execution in business process anomaly in model"
3  End Sub
```

Kôd 129. Način rješavanja anomalije ***Djelomično lančano izvršenje u poslovnom procesu***

3.3.3. Anomalije višestrukih obrazaca resursa

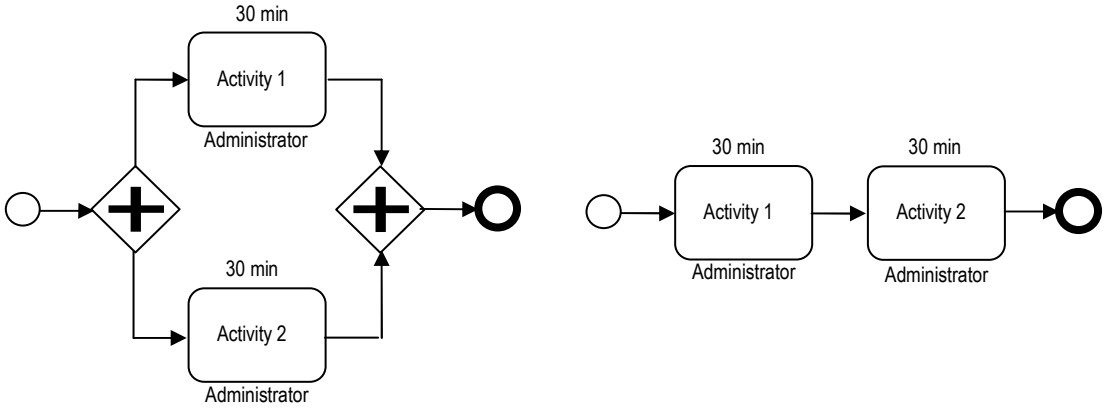
Prema predloženoj taksonomiji anomalija u modelima poslovnih procesa ***Anomalije višestrukih obrazaca resursa*** je grupa anomalija koja spada u ***Anomalije resursa***, u navedenu grupu resursa spada anomalija ***Istovremeno izvođenje kod ograničenih kapaciteta resursa***.

Istovremeno izvođenje kod ograničenih kapaciteta resursa

(eng. *Simultaneous execution with limited capacity*)

Opis - istovremeno izvođenje više instanci resursa ima smisla ako poslovni proces ima dovoljne kapacitete resursa. Ako ti kapaciteti ne postoje performanse instanci će biti dodatno usporene. Navedena anomalija je vrlo slična anomaliji ***Alokacija temeljena na ulogama kod grupe sa jednim resursom***, naime i u ovom slučaju resursi nemaju dovoljni kapacitet. Također je totalno obrnuta od situacije koja postoji u anomaliji ***Djelomično lančano izvršenje u poslovnom procesu*** u kojoj se, jer to kapacitet resursa, dopušta da se neki dijelovi poslovnog procesa mogu izvesti paralelno. Ovdje se neki dijelovi moraju ipak izvesti slijedno, jer resursi nemaju dovoljan kapacitet. No, u ovome slučaju radi se o dinamičkoj anomaliji

koju je teško prepoznati bez simulacije. Prema predloženoj taksonomiji anomalija ***Istovremeno izvođenje kod ograničenih kapaciteta resursa*** predstavlja ***Anomaliju resursa*** i koja se nalazi u grupi ***Anomalije višestrukih obrazaca resursa.***



Slika 92. Anomalija ***Istovremeno izvođenje kod ograničenih kapaciteta resursa***

Način prepoznavanja anomalije – temelji se na sljedećim pravilima:

- u modelu poslovnog procesa postoji više od jedne aktivnosti,
- u modelu poslovnog procesa dva ili više aktivnosti odvijaju se paralelno od strane istih resursa i
- u modelu poslovnog procesa resursi nemaju dovoljno kapaciteta za paralelni rad.

Ako se navedena pravila prikažu pomoću OCL jezika, onda to izgleda ovako:

```
1  context Process
2   inv SimultaneousExecutionLimitedCapacity:
3    Count(Fork)>0 and
4    Activity.Resource ->
5    forAll(x | x.role -> select(self.ResourceRole.Requirement)->size()=1)
6     implies
7     SimultaneousExecutionLimitedCapacityAnomaly()
```

Kôd 130. Način prepoznavanja anomalije ***Istovremeno izvođenje kod ograničenih kapaciteta resursa***

Način rješavanja anomalije – temelji se na izbjegavanju odabira istog resursa na više paralelnih aktivnosti.

```
1  context Process
2   inv SimultaneousExecutionLimitedCapacityAnomaly:
3    Count(Fork)>0 and
4    Activity.Resource ->
5    forAll(x | x.role -> select(self.ResourceRole.Requirement)->size()>1)
```

Kôd 131. Način prepoznavanja anomalije ***Istovremeno izvođenje kod ograničenih kapaciteta resursa***

3.4. Semantičke anomalije

Semantičke anomalije vezane su uz nazivlje elemenata modela. Mogu se podijeliti na tri temeljna tipa:

- nekonzistentnost nazivlja,
    - naziv aktivnosti je imenica,
    - vrata odlučivanja je imenovana kao aktivnost,
    - i/ili u nazivu aktivnosti.
- nepravilno nazivlje elemenata i
- nepravilne vrijednosti atributa.

3.4.1. Nepravilno nazivlje elemenata

Nepravilno nazivlje ili neimenovanje elemenata dovodi do semantičkih anomalija. Najčešći uzroci nepravilnog nazivlja su nekonzistentnost nazivlja elemenata, nejedinstvenost nazivlja elemenata i uporaba nedozvoljenih znakova u nazivlju.

U ovu skupinu anomalija spadaju problemi s sinonimima i homonimima.

Sinonimi su leksemi koji pripadaju istoj vrsti riječi, imaju različite izraze, a sadržaj im se potpuno ili djelomično podudara.

Primjer: sustav – sistem, *računalo - kompjuter*.
(eng. Honest - Honorable, Fair, Sincere, Trustworthy.
Intelligent - Smart, Bright, Brilliant, Sharp)

Homonimi su leksemi jednakog izraza, a različitog značaja.

Primjer: luk (povrće) – luk (oružje).
(eng. *Right (You were right*) - *Right (Make a right turn at the light)*.

I sinonimi i leksemi predstavljaju problem u modelima poslovnih procesa, jer se na modelima obično radi ručno, što često dovodi do pogreške, pogotovo ako je model složeniji pa se smanjuje njegova preglednost.

Prepoznavanje sinonima i homonima uključuje konzultiranje tablica koje sadrže liste homonima i sinonima.

Sinonimi se pojavljuju u slučaju kad različiti nazivi objekata u različitim aplikacijama predstavljaju isti objekt. Postoji nekoliko temeljnih pristupa kod rješavanja sinonima kao što su:

- rječnici sinonima koji uspoređuju nazive objekata,
- pristup temeljen na kriterijima usporedbe karakteristika objekata i
- usporedba vrijednosti objekata.

Način rješavanja anomalije - pristup rješavanja sinonima koji uključuje uporabu rječnika sinonima temelje se na usporedbi naziva objekata s rječnicima sinonima. Ukoliko se naiđe na objekte koji imaju nazive koje povezuje rječnik sinonima u tom slučaju se nazivi tih objekata zamjenjuju s jedinstvenim nazivom.

Problemi koji se mogu pojaviti uporabom ove metode su:

- neažuriranost rječnika sinonima novim izrazima,
- ograničenost baze sinonima zbog višejezičnosti i
- različita terminologija nazivlja.

Pristup temeljen na kriterijima usporedbe karakteristika objekata temelji se u ovome slučaju na usporedbi atributa pojedinog objekta. Autori Larson, Navathe i Elmasri u svojem radu *A theory of attribute equivalence in database with application to shema integration* [31], postavili su temelje pristupa usporedbe atributa u bazama podataka. Taj mehanizam u funkcionalnim modelima može se primijeniti kod sadržaja razmjene (eng. *Business items*). Za njih vrijede ista pravila kao i sa tablicama baza podataka.

Karakteristike atributa pomoću kojih se uspoređuju atributi sadržaja razmjene su:

- jedinstvenost (eng. *Uniqueness*),
- kardinalnost (eng. *Cardinality*),
- domena (eng. *Domain*),
- statička semantička ograničenja koja osiguravaju integritet (eng. *Static Semantic Integrity Constraints*),
- dinamička semantička ograničenja koja osiguravaju integritet (eng. *Dynamic Semantic Integrity Constraints*),
- sigurnosna ograničenja (eng. *Security Constraints*),
- dopuštene operacije (eng. *Allowable Operations*) i
- mjerilo (eng. *Scale*).

Kada se govori o usporedbi pojedinih aktivnosti, tada se u obzir moraju uzeti atributi pojedinih aktivnosti. Karakteristike atributa koji se uspoređuju kod sadržaja razmjene nisu primjenjive na atribute aktivnosti, naime atributi aktivnosti su uvijek isti. Atributi aktivnosti su:

- naziv,
- troškovi obrade (eng. *Processing cost*),
- troškovi pokretanja (eng. *Start up cost*),
- prihodi (eng. *Revenue*),
- trajanje (eng. *Duration*),
- ulazi (eng. *Inputs*),
    - poslovne stavke (eng. *Business items*),
- izlazi (eng. Outputs),
    - Poslovne stavke (eng. *Business items*),
- resursi i
- organizacijska jedinica.

Ako usporedimo dvije aktivnosti različitog naziva, koje imaju slične atribute vrlo je vjerojatno da su one iste.

Tabela 15. Primjer usporedbe atributa objekta

Aktivnost Atributi	**Otpremiti dopis**	**Poslati dokument**
Troškovi obrade (eng. *Processing cost*)	5 kn	5 kn
Troškovi pokretanja (eng. *Startup cost*)	5 kn	5 kn
Prihodi (eng. *Revenue*)	0	0
Trajanje (eng. *Duration*)	10’	10’
Ulazi (eng. *Inputs*)	Dopis	Dokument
Izlazi (eng. Outputs)	Potvrda	Potvrda
Resursi	Dostavljač	Dostavljač
Organizacijska jedinica	Opći poslovi	Opći poslovi

Jakubowski u svojem radu *Discovering synonyms with use of frequent itemsets and association rules* [27] sinonime definira kao različite riječi ili fraze sa sličnim ili istim značenjem i koje mogu biti zamjenjive. Sinonim može biti: imenica, glagol, prilog, pridjev pod uvjetom da su članovi istog para. Medin, Goldsone i Genrer [35] razlikuju dvije vrste sličnosti; atributivnu i relacijsku. Atributivna sličnost je odnos između atributa neke riječi, a relacijska sličnost bavi se pitanjem koji je odnos između relacija neke riječi. Kad neke dvije riječi imaju visoki stupanj atributivne sličnosti, te ako imaju jednak odnos prema drugim riječima, onda ih se može nazvati ***sinonimima***. Kad dvije riječi imaju visoki stupanj relacijske sličnosti, onda ih se može nazvati ***analognima***.

Razvijeno je poprilični broj algoritama koji se bave atributivnom sličnošću. Međutim algoritmi koji se bave relacijskom sličnošću između dviju riječi su daleko rjeđi.

Homonimi i sinonimi su obično sistematizirani i kategorizirani u rječnicima i tezaurusima. ***Rječnik*** je kolekcija odnosno zbirka riječi sa svojim opisima, definicijama i značenjima. Dakle za svaku riječ rječnik sadrži njezino značenje, oblike, kontekste uporabe i tako dalje. Rječnici daju detalje značenja, definiciju, uporabu, porijeklo riječi.

Tezaurus predstavlja riječi i njihove familije i njihove sinonime, bez objašnjenja ili načina uporabe. Oni ne sadrže sve riječi nekog rječnika, nego daju nekoliko alternativa za svaku pojedinu riječ. Tezaurusi mogu biti poredani abecedno ili konceptualno.

Kad se sinonimi i homonimi otkrivaju u nekom tekstu algoritmi otkivanja drugačiji su od otkrivanja tih pojava u nekom modelu. Kad je tekst u pitanju obično se tekst prvo mora preoblikovati na način da je primjena tezaurusa moguća. Dakle, tekst treba pretvoriti u niz riječi koji su međusobno odvojeni u rečenice, obično se pretpostavlja da se isti sinonimi ne nalaze u istoj rečenici. U modelima nazivi elemenata i nazivi atributa su već odvojeni. Provjera je bitna ako sinonimi i homonimi postoje na razini nazivlja pojedinih tipova elemenata i na razini atributa pojedinog elementa. Svaku od pojedinih razina treba promatrati odvojeno.

Nekonzistentnost nazivlja elemenata

(eng. *Inconsistency in element names*)

Opis – anomalija ***Nekonzistentnost nazivlja*** ogleda se u obliku na koji se elementi imenuju. Pa tako naziv aktivnosti temeljen na imenici pokazuje da se radi o događaju, objektu podataka ili proces dakle suprotno od aktivnosti. Vrata odlučivanja imenovana kao aktivnost pokazuje da vrata predstavljaju aktivnost. Riječi i/ili u nazivu aktivnosti pokazuju da se radi o višestrukim aktivnostima objedinjenima u jednoj. Dugačak naziv aktivnosti pokazuju da su naglašeni podaci aktivnosti, a ne cilj što se s određenom aktivnosti želi postići. Prema predloženoj taksonomiji anomalija u modelima poslovnih procesa anomalija ***Nekonzistentnost nazivlja elemenata*** spadaju u ***Semantičke anomalije*** u grupu anomalija ***Nepravilno nazivlje elemenata***.

Način prepoznavanja anomalije – temelji se na sljedećim pravilima:

- u modelu poslovnog procesa naziv procesa i potprocesa nisu u nominativu,
- u modelu poslovnog procesa postoje aktivnosti kojima naziv ima više riječi i više glagola,
- u modelu poslovnog procesa postoje aktivnosti kojima naziv nema glagol i
- u modelu poslovnog procesa postoji baza podataka kojoj naziv nije imenica.

Ako se navedena pravila prikažu pseudokodom, onda to izgleda ovako:

```
 1  RULE: InconsistencyInElementNames
 2   VerbCount int = 0;
 3   NounCount int = 0;
 4   FOR EACH GraphicalObject IN GraphicalObjects
 5    SELECT CASE GraphicalObject.Category
 6     CASE Process, SubProcess
 7         IF IsNominative(Process)=FALSE OR IsNominative(SubProcess)=FALSE THEN
 8                ProcessSubProcessNominativeAnomaly()
 9         ENDIF
10     CASE Acivity
11         IF Activity.Name.Words.Count >=1 THEN
12            FOR EACH Word IN Activity.Name.Words
13                IF  IsVerb(Word) = TRUE
14                       VerbCount  = VerbCount+1
15                ENDIF
16            NEXT
17            SELECT CASE VerbCount
18              CASE 0
19                NoVerbsInActivityAnomaly()
20             CASE  > 1
21                TooManyVerbsActivityAnomaly()
22            END SELECT
23             VerbCount = 0
24             IF IsNominative(Activity.Name)=FALSE THEN
25                NotInNominativeActivityAnomaly()
26             ENDIF
27           ELSE
28             NameIsEmptyActivityAnomaly()
29           ENDIF
30      CASE Database
31         IF Database.Name.Words.Count >=1 THEN
32            FOR EACH Word IN Database.Name.Words
33                IF  IsNoun(Word) = TRUE
34                       NounCount  = NounCount  +1
35                ENDIF
36            NEXT
37            SELECT CASE NounCount
38              CASE 0
39                NoNounsInDatabaseAnomaly
40             CASE  > 1
41                TooManyNounsInDatabaseAnomaly
42            END SELECT
43             NounCount = 0
44         ELSE
45             NameIsEmptyDatabaseAnomaly()
46         ENDIF
47     CASE Gateway
48         IF IsQuestion(Gateway.Condition)=FALSE THEN
49                GatewayConditionAnomaly()
50         ENDIF
51    END SELECT
52   NEXT
```

Kôd 132. Prepoznavanje anomalije ***Nekonzistentnost nazivlja elemenata***

Način rješavanja anomalije - kod navedene anomalije vrlo teško je automatizirati ispravku nazivlja, navedena pravila u biti služe više kao preporuka, koju bi valjalo poštovati kako bi dobili smislen i ispravan model poslovnog procesa.

```
FIX: ProcessSubProcessNominativeAnomaly()
    strMessage = "Process or SubProcess name is not in nominative"

FIX: NoVerbsInActivityAnomaly()
    strMessage = "Activity name have no verbs"

FIX: TooManyVerbsActivityAnomaly()
    strMessage = "Activity name have more than one verb"

FIX: NotInNominativeActivityAnomaly()
    strMessage = "Anomaly name is not in nominative"

FIX: NameIsEmptyActivityAnomaly()
    strMessage = "Anomaly name is empty"

FIX: AnomalyNameIsEmptyDatabase ()
    strMessage = "Database name is empty"

FIX: NoNounsInDatabaseAnomaly()
    strMessage = "Database name have no nouns"

FIX: TooManyNounsInDatabaseAnomaly()
    strMessage = "Database name have more than one nouns"

FIX:AnomalyGatewayCondition()
    strMessage = "Gateway condition is not question"
```

Kôd 133. Procedura rješavanja anomalije ***Nekonzistentnost nazivlja elemenata***

Sinonimi u nazivlju elemenata modela

(eng. *Synonyms in element names*)

Opis - u modelima poslovnih procesa sinonimi se pojavljuju u slučaju kad postoje različiti nazivi polja u istim kategorijama objekata. Prema predloženoj taksonomiji anomalija u modelima poslovnih procesa anomalija ***Sinonimi u nazivlju elemenata modela*** spadaju u ***Semantičke anomalije*** u grupu anomalija ***Nepravilno nazivlje elemenata***.

Način prepoznavanja anomalije - u daljnjem tekstu opisati će se najjednostavniji algoritam koji će omogućiti pretraživanje sinonima u modelima poslovnih procesa. Navedeni algoritam je samo skica na koji način prepoznati sinonime. Sinonimi moraju biti spremljeni na jednom mjestu, u ovome slučaju tablici **tblSynonymes.**

Tabela 16. Izgled tablice sa sinonimima

tblSynonymes		
Naziv polje	Tip podataka i dužina	Opis
strSynonymes	String(255)	Sustav;Sistem
strDescription	String(255)	Skup elemenata koji tvore određenu strukturu

Temeljni koraci najjednostavnijeg algoritma za prepoznavanje sinonima bio bi:

- za svaki element u pojedinoj kategoriji treba pronaći sinonime,
- svaku listu tih sinonima treba usporediti sa svim nazivima svih elemenata iz pojedine kategorije i
- korisniku treba dati obavijest o parovima mogućih sinonima koji se pojavljuju u modelu.

U najjednostavnijem obliku algoritam za prepoznavanje sinonima pretvoren u pseudokod izgledao bi ovako:

```
1   RULE: FindSynonymeInModel()
2    strCurrentElementName string =“”;
3    arrCurrentSynonymesList Array = Nothing;
4    intGO int =0;
5    intGO_S int =0M
6    FOR EACH GraphicalObject.Category in GraphicalObject.Categories
7     FOR EACH GraphicalObject IN GraphicalObjects
8      strCurrentElementName = GraphicalObject.Name
9      arrCurrentSynonymesList = GetSynonymes(strCurrentElementName)
10      FOR i=0 TO arrCurrentSynonymesList.Lenght-1
11       FOR EACH GraphicalObject IN GraphicalObjects
12        IF arrCurrentSynonymes(i).ToString = GraphicalObject(intGo_S).ToString THEN
13         IF GraphicalObject(intGo_S).Name <> GraphicalObject(intGo).Name THEN
14          SynonymeAnomaly(GraphicalObject(intGo_S).Name + “-“ + GraphicalObject(intGo).Name)
15         ENDIF
16        ENDIF
17        intGO_S = intGO_S+1
18       NEXT
19      NEXT
20     NEXT
21     intGO = intGO+1
22    NEXT
```

Kôd 134. Algoritam prepoznavanja anomalije ***Sinonimi u nazivlju elemenata modela***

Funkcija ***GetSynonymes*** sastoji se od dijelova koji dohvaćaju za svaki naziv pripadajuće sinonime i u ovome slučaju smještaju sinonime u listu što omogućava lakše pretraživanje i uspoređivanje sinonima i naziva objekata. U tabeli **tblSynonymes** je polje u kojoj je lista sinonima i koje se zove *strSynonymes*, vrijednosti upisane u to polje su odvojene znakom **;** (točka-zarez). Npr. `Sustav;Sistem`**,** onda bi pseudokod te funkcija izgledao ovako:

```
1   Function GetSynonymes(ElementName as String) as Array
2
3   Dim strSynonymesTemp as String =“”
4   SELECT
5   strSynonymesTemp=strSynonymes
6   FROM tblSynonymes
7   WHERE strSynonyme like “%ElementName%“
8
9   GetSynonymes = SPLIT(strSynonymesTemp,“;“)
10
11  End Function
```

Kôd 135. Funkcija dohvaćanja sinonima ***GetSynonymes***

Način rješavanja anomalije - kod nailaska na ***sinonime u nazivlju elemenata modela***, potrebno je korisnika obavijestiti da u modelu postoje elementi sa sinonimima. Kod procedure koja obavještava korisnika o sinonimima potrebno je osigurati ulazni parametar sa parovima sinonima.

```
1  FIX: SynonymeAnomaly(strSynonymePair as string)
2    strMessage = "Synonymes in model in following elements:" + strSynonymePair
```

Kôd 136. Način rješavanja anomalije ***Sinonimi u nazivlju elemenata modela***

Homonimi u nazivlju elemenata modela

(eng. *Homonyms in element names*)

Opis - isto kao i kod algoritma za prepoznavanje i rješavanje sinonima, problem homonima je djelomično sličan. Metoda rješavanja podrazumjeva da mora postojati tablica u kojoj će se držati svi homonimi i njihova značenja. Za razliku od sinonima gdje se korisniku može sugerirati da u modelu postoje dva modela iste kategorije koja su sinonimi, sa homonimima je to malo drugačije.

Način prepoznavanja anomalije – kod algoritma za prepoznavanje homonima nije dovoljno samo da unutar modela postoje istoimeni objekti istih kategorija, jer u većini slučajeva tu bi se radilo o istome objektu unutar modela. Dakle, kod prepoznavanja homonima potrebno je postaviti dodatne mehanizme provjere da li se radi o istome objektu ili različitim objektima. Ovaj problem djelomično je pokriven rješavanjem anomalije ***Nejedinstvenosti nazivlja elemenata***, koji se rješava OCL pravilom:

```
1  context Process
2  inv NonUniqueName:
3     (IsUnique(self.GraphicalObjects.Name)=False)
4     implies
5     NonUniqueNameAnomaly()
```

Kôd 137. Način prepoznavanja anomalije ***Nejedinstvenosti nazivlja elemenata***

Navedeno pravilo pri nailasku na više objekata istog naziva, odmah to smatra anomalijom ***Nejedinstvenosti nazivlja elemenata***, no može uključiti i dodatna provjera ostalih atributa kako bi bili sigurni da se radi o različitim objektima.

Algoritam bi imao sljedeće korake:

- unutar svake kategorije pronaći više elemenata istog imena,
- ako takvi postoje onda provjeriti da li su im nazivi u tablici homonima i
- ako takvi postoje onda provjeri da li ima iste ulaze i izlaze.

Ako se navedena pravila pretvore u pseudokod, onda to izgleda ovako:

```
1   RULE: FindHomonymeInModel()
2
3    strCurrentElementName string =“”;
4    intCurrentInputCount as integer;
5    intCurrentOutputCount as integer;
6    strHomonymeObjectName as string;
7    intHomonymeInputCount as integer;
8    intHomonymeOutputCount as integer;
9
10   FOR EACH GraphicalObject.Category in GraphicalObject.Categories
11    FOR EACH GraphicalObject IN GraphicalObjects
12     IF GraphivalObject.IsGlobal = False THEN
13      strCurrentObjectName = GraphicalObject.Name
14      intCurrentInputCount = GraphicalObject.Inputs.Count
15      intCurrentOutputCount = GraphicalObject.Outpsts.Count
16       IF Count(strCurrentObjectName)>1 AND GetHomonymes(strCurrentObjectName) = TRUE THEN
17        strHomonymeObjectName = GraphicalObject.Name
18        intHomonymeInputCount = GraphicalObject.Inputs.Count
19        intHomonymeOutputCount = GraphicalObject.Outputs.Count
20       ENDIF
21       IF strCurrentObjectName = strHomonymeObjectName THEN
22        IF intCurrentInputCount = intHomonymeInputCount AND
23        intCurrentOutputCount = intHomonymeOutputCount AND
24        THEN
25          DuplicateObjectInModelAnomaly()
26        ELSE
27          HomonymesInModelAnomaly()
28        ENDIF
29       ELSE
30           DuplicateObjectInModelAnomaly()
31       ENDIF
32      ENDIF
33     NEXT
34    strHomonymeObjectName = “”
35    intHomonymeInputCount = 0
36    intHomonymeOutputCount = 0
37   NEXT
```

Kôd 138. Algoritam prepoznavanja anomalije ***Homonimi u nazivlju elemenata modela***

U proceduri se koristi funkcija ***GetHomonymes*** koja po ulaznom parametru naziva elementa pretražuje tablicu ***tblHomonymes*** i traži da li postoji broj homonima veći od 1 za traženi naziv elementa. Ako homonim postoji funkcija vraća `TRUE`, ako ne postoji funkcija vraća `FALSE`. Struktura tablice ***tblHomonymes*** izgleda ovako:

Tabela 17. Tablica homonima ***tblHomonymes***

tblHomonymes		
Naziv polje	Tip podataka i dužina	Opis
strHomonymes	String(255)	Luk
strDescription	String(255)	Oružje

Navedena funkcija opisana u pseudokodu izgledala bi ovako:

```
Function GetHomonymes(ElementName as String) As Boolean

 Dim intHomonymesTemp as Integer = 0

 SELECT
 intHomonymesTemp = count(strHomonymes)
 FROM tblHomonymes
 WHERE strHomonymes like '%ElementName%'

 IF intHomonymesTemp>0 THEN
  GetHomonymes=TRUE
 ELSE
  GetHomonymes=FALSE
 ENDIF
End Function
```

Kôd 139. Funkcija dohvaćanja sinonima ***GetHomonymes***

Način rješavanja anomalije - procedura u slučaju da sve uvjete ima zadovoljene, dakle pronalazak elementa koji ima naziv koji se više puta pojavljuje u modelu, koji imaju naziv koji se nalazi u tablici homonima i koji imaju iste ulaze i izlaze, okida proceduru **DuplicateObjectInModelAnomaly()**, koja vraća korisniku obavijest o postojanju takve anomalije. Dakle u ovoj proceduri za pronalaženje homonima od atributa uspoređuje se samo broj ulaza i izlaza što je dovoljno za prikaz ove skice algoritma, no za primjenu u izgradnji alata za modeliranje trebalo bi uvrstiti i usporedbu drugih atributa poput resursa, vremena trajanja, cijene koštanja, itd.

```
FIX: DuplicateObjectInModelAnomaly()
  strMessage = "There are two identical objects in model"
```

Kôd 140. Način rješavanja u slučaju duplog objekta u modelu

U slučaju da procedura prepozna slučaj da postoji više elemenata koji nemaju isti broj ulaza i izlaza i koji nisu globalni, onda se pokreće procedura **HomonymesInModelAnomaly().** Koja obavještava korisnika da postoje homonimi u modelu poslovnih procesa.

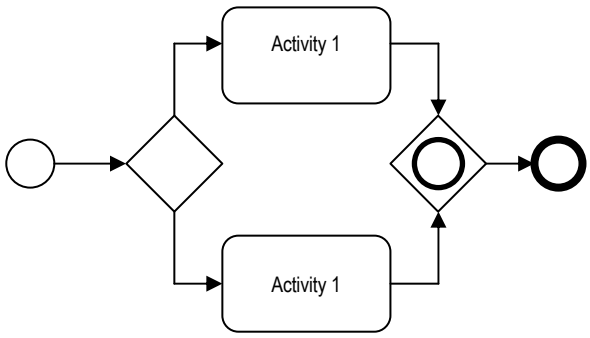
```
FIX: HomonymesInModelAnomaly()
 strMessage = "There are objects with homonymus names in model"
```

Kôd 141. Način rješavanja anomalije ***Homonimi u nazivlju elemenata modela***

Nejedinstvenost nazivlja elemenata

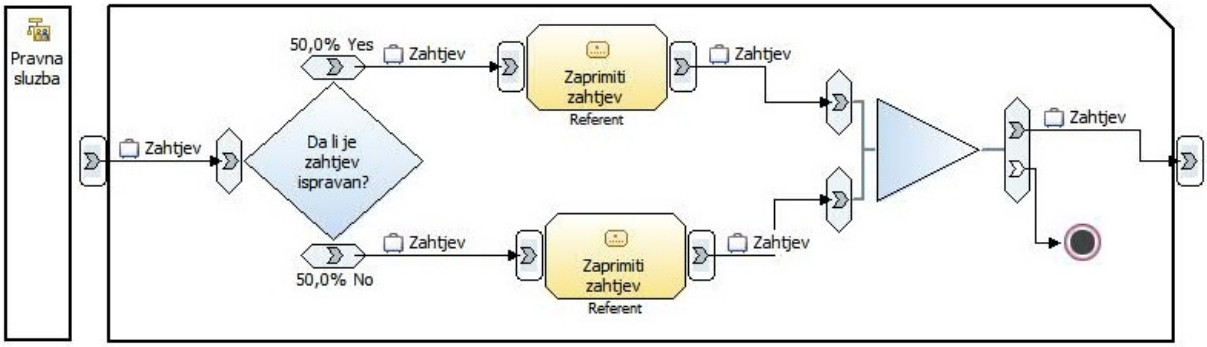
(eng. *Non-unique element names*)

Opis – ova vrsta anomalije nastaje ako više elemenata u modelu poslovnih procesa imaju isti naziv. U tom slučaju lako može doći do zabune koja aktivnost je aktivna i koja je sljedeća aktivnost i koji resurs treba biti uključen u realizaciju pojedine aktivnosti. Prema predloženoj taksonomiji anomalija modela poslovnih procesa anomalija ***Nejedinstvenost nazivlja elemenata*** dio ***Semantičkih anomalija*** u grupi anomalija ***Nepravilno nazivlje elemenata***.



Slika 93. Primjer semantičke anomalije ***Nejedinstvenost nazivlja elemenata***

Primjer – na sljedećem primjeru vidi se kako bi izgledao model poslovnog procesa, ako bi u modelu postojale dvije aktivnosti istog nazivlja.



Slika 94. Semantička anomalija ***Nejedinstvenost nazivlja aktivnosti***

Način prepoznavanja anomalije – pravilo pomoću kojeg bi mogli pronaći ovu anomaliju je:

- u modelu poslovnog procesa postoji grafički element kojem naziv nije jedinstven.

Ako se navedeno pravilo prikaže pomoću OCL jezika, to izgleda ovako:

```
1 context Process
2 inv NonUniqueName:
3    (IsUnique(self.GraphicalObjects.Name)=False)
4    Implies
5    NonUniqueNameAnomaly()
```

Kôd 142. Način prepoznavanja anomalije ***Nejedinstvenost nazivlja elemenata***

Način rješavanja anomalije – rješenje za ovu vrstu anomalije trebalo bi ugraditi u metamodel BPMN 2.0 notacije kao OCL ograničenje u kojem treba definirati da unutar poslovnog procesa naziv grafičkih elemenata mora biti jedinstven.

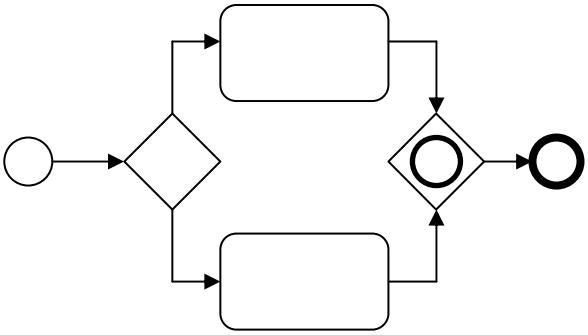
```
1  context Process
2  inv NonUniqueNameAnomaly:
3         IsUnique(self.GraphicalObjects.Name)=True
```

Kôd 143. Način rješavanja anomalije ***Nejedinstvenost nazivlja elemenata***

Neimenovanje elemenata

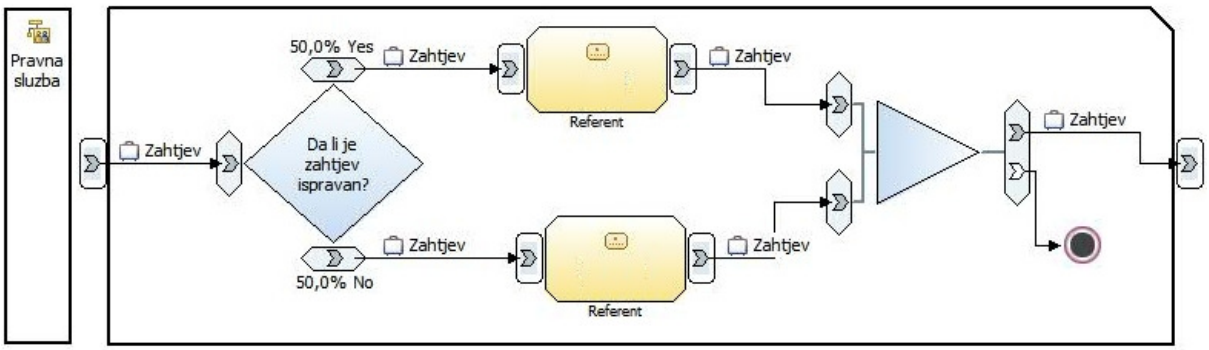
(eng. *Unnamed element names*)

Opis - Neimenovanje pojedinih objekata također su uzrok pojave semantičkih anomalija. Prema predloženoj taksonomiji anomalija modela poslovnih procesa anomalija ***Neimenovanje elemenata*** dio ***Semantičkih anomalija*** u grupi anomalija ***Nepravilno nazivlje elemenata***.



Slika 95. Primjer semantičke anomalije ***Neimenovanje elemenata***

Primjer - sljedeći primjer prikazuje model poslovnog procesa u kojem postoji anomalija ***Neimenovanje elemenata*** modela.



Slika 96. Primjer semantičke anomalije ***Neimenovanje elemenata***

Način prepoznavanja anomalije – pravilo prepoznavanja može se opisati na sljedeći način:

- u modelu poslovnog procesa naziv nekog od grafičkih elemenata u modelu procesa prazan.

Ako se navedeno pravilo prikaže pomoću OCL jezika, onda to izgleda ovako:

```
1  context Process
2  inv UnNamedElements:
3     (self.GraphicalObjects.Name -> IsEmpty())
4     implies
5     UnNamedElementsAnomaly()
```

Kôd 144. Način prepoznavanja anomalije ***Neimenovanje elementa modela***

Način rješavanja anomalije – ovu anomaliju treba rješavati ugrađivanjem OCL pravila u metamodel BPMN 2.0 notacije.

```
1  context Process
2  inv UnNamedElementsAnomaly:
3     (self.GraphicalObject.Name -> IsNotEmpty())
```

Kôd 145. Način rješavanja anomalije ***Neimenovanje elementa modela***

3.4.2. Nepravilne vrijednosti atributa

Nepravilne vrijednosti atributa predstavljaju grupu anomalija koje spadaju u ***Semantičke anomalije***. U grupu ***Nepravilne vrijednosti atributa*** spadaju anomalije kao što su: ***Ne postoji vrijednost atributa***, ***Nejedinstvenost vrijednosti atributa***, ***Neispravan tip podataka vrijednosti atributa*** i ***Nedozvoljeni znakovi u vrijednosti atributa***.

Ne postoji vrijednost atributa

(eng. *Empty attribute value*)

Opis – kod ove anomalije bitna je provjera da li atribut ima ikakvu vrijednost. Primjerice da li je **nazivlje** elementa dužine nula ili veće od maksimalne vrijednosti. U slučaju da nazivlje elementa nije u skladu s provjerom, obično se korisnik upozori da postoji problem s nazivom elemenata ili da se naziv tada promjeni u neku unaprijed određenu vrijednost. Ova anomalija na nadskup anomalije ***Neimenovanje elemenata***, jer i naziv elementa spada u vrijednosti

atributa. Prema predloženoj taksonomiji anomalija ***Ne postoji vrijednost atributa*** dio je grupe anomalija ***Nepravilne vrijednosti atributa*** koje spadaju u ***Semantičke anomalije.***

Način prepoznavanja anomalije - provjera atributa elementa poslovnog procesa može se prikazati sljedećim pravilom:

- u modelu poslovnog procesa atributi elementa nemaju dodijeljenu nikakvu vrijednost.

Ako se navedeno pravilo prikaže OCL jezikom, onda to izgleda ovako:

```
1  context Process
2  inv EmptyAttributeValue:
3     (self.GraphicalObjects.Attribute.Value -> IsEmpty())
4     implies
5     EmptyAttributeValueAnomaly()
```

Kôd 146. Način prepoznavanja anomalije ***Ne postoji vrijednost atributa***

Način rješavanja anomalije - procedura obavještava korisnika da u modelu postoji element bez vrijednosti atributa.

```
1  FIX: EmptyAttributeValueAnomaly()
2     strMessage = 'There is element without name in model'
```

Kôd 147. Način prepoznavanja anomalije ***Ne postoji vrijednost atributa***

Nejedinstvenost vrijednosti atributa

(eng. *Non-unique attribute value*)

Opis – kod ove anomalije provjerava se da li element ima istu vrijednost kao i neku drugi istovrsni element iako mora imati jedinstvenu vrijednost. Primjerice ukoliko je naziv elementa već u uporabi kod istovrsnog elementa u tom slučaju se nazivu pridodaje brojčana instanca objekta. Prema predloženoj taksonomiji anomalija ***Nejedinstvenost vrijednosti atributa*** dio je grupe anomalija ***Nepravilne vrijednosti atributa*** koje spadaju u ***Semantičke anomalije.***

Način prepoznavanja anomalije – temelji se na sljedećem pravilu:

- u modelu poslovnog procesa postoje elementi gdje vrijednost atributa nije jedinstvena, a trebala bi biti.

Ako se navedeno pravilo prikaže pomoću OCL-a, to izgleda ovako:

```
1  context Process
2  inv NonUniqueAttributeValue:
3     (IsUnique(self.GraphicalObjects.Attribute(i).Value)=False)
4     implies
5     NonUniqueAttributeValueAnomaly()
```

Kôd 148. Način prepoznavanja anomalije ***Nejedinstvenost vrijednosti atributa***

Način rješavanja anomalije - procedura obavještava korisnika da u modelu postoji element čija vrijednost atributa nije jedinstvena.

```
1  Sub NonUniqueAttributeValueAnomaly()
2     strMessage = "There is element with NonUnique names in model"
```

Kôd 149. Način rješavanja anomalije ***Nejedinstvenost vrijednosti atributa***

Neispravan tip podataka vrijednosti atributa

(eng. *Wrong attribute data type value*)

Opis - kod ove provjere provjerava se da li je podatak numerički ili alfanumerički ili nekog drugog tipa, te da li je vrijednost atributa u skladu s unaprijed definiranom vrijednošću. Prema predloženoj taksonomiji anomalija ***Neispravan tip podataka vrijednosti atributa*** dio je grupe anomalija ***Nepravilne vrijednosti atributa*** koje spadaju u ***Semantičke anomalije.***

Način prepoznavanja anomalije - ako se navedeno prikaže pomoću pravila, onda to izgleda ovako:

- u modelu poslovnog procesa postoje elementi gdje tip podataka atributa nije ispravan i
- u modelu poslovnog procesa postoje elementi gdje vrijednost atributa nije unutar zadanih granica.

Ako se navedena pravila prikažu pomoću OCL jezika, onda to izgleda ovako:

```
1  context Process
2  inv WrongAttributeDataTypeValue:
3            (IsNumeric(self.GraphicalObjects.Attibute(i).Value)=False)
4         OR (self.GraphicalObjects.Attribute(i).Value)< intMinValue)
5         OR (self.GraphicalObjects.Attribute(i).Value)> intMaxValue)
6         implies
7         WrongAttributeDataTypeValueAnomaly()
```

Kôd 150. Prepoznavanje anomalije ***Neispravan tip podataka vrijednosti atributa***

Način rješavanja anomalije - procedura obavještava korisnika da u modelu postoji anomalija gdje tip podataka vrijednosti atributa nije ispravan.

```
1  Sub WrongAttributeDataTypeValueAnomaly ()
2          strMessage = "There is element data type anomaly"
3  End Sub
```

Kôd 151. Način rješavanja anomalije ***Neispravan tip podataka vrijednosti atributa***

Nedozvoljeni znakovi u vrijednosti atributa

(eng. *Special characters in attribute value*)

Opis – kod navedene anomalije provjerava se da li u nazivlju ili vrijednosti atributa postoje neki nedozvoljeni znakovi koji mogu utjecati ispravnost uporabe modela. Obično se tu radi o znakovima koji se u nekim sintaksama mogu protumačiti kao operatori (=, +, -, <, >, @,…itd.). Prema predloženoj taksonomiji anomalija ***Nedzvoljeni znakovi u vrijednosti atributa*** dio je grupe anomalija ***Nepravilne vrijednosti atributa*** koje spadaju u ***Semantičke anomalije.***

Način prepoznavanja anomalije – temelju se na sljedećem pravilu:

- u modelu poslovnog procesa postoje elementi gdje u vrijednostima atributa postoje nedozvoljeni znakovi.

OCL pravilom pomoću regularnih izraza, navedena anomalija može se prepoznati na sljedeći način:

```
1  context Process
2  inv SpecialCharactersInAttribute:
3          Regex.IsMatch(GraphicalObjects.Attibute.Value, "^[A-Za-z0-9]+$") = False
4         implies
5         AttributeSpecialCharactersAnomaly()
```

Kôd 152. Prepoznavanje anomalije ***Nedozvoljeni znakovi u vrijednosti atributa***

Način rješavanja anomalije - procedura vraća poruku korisniku da u modelu postoji elementa sa nedozvoljenim znakovima u vrijednostima atributa.

```
1  Sub AttributeSpecialCharactersAnomaly ()
2          strMessage = "There is element with special characters in name"
3  End Sub
```

Kôd 153. Način rješavanja anomalije ***Nedozvoljeni znakovi u vrijednosti atributa***

Slaba provedba regulacijskih standarda koji se odnose na nazive podataka, formata, vrijednosti ograničenja dovodi do čestih problema zbog naziva. Vrlo je teško provesti standardizaciju kroz sve aplikacije u kojima se nalaze duplicirani podaci koji su raspoređeni kroz cijelu organizaciju. Primjerice, ako standard organizacije nalaže da kupac ima polje od 35 bytes.

3.5. Anomalije prikaza modela

Uloga modela poslovnih procesa je prikazati poslovni proces unutar organizacije na način da je jednostavan i pregledan za promatranje i izučavanje. Ukoliko model nije kreiran na način da je to moguće, onda model gubi svoj temeljni smisao. Obično anomalije prikaza modela ne utječu na ponašanje modela, no utječu na njegov izgled i prezentaciju korisniku.

Anomalije prikaza modela mogu se smatrati estetskim anomalijama. Mogu se podijeliti prema zakonitostima estetike koji se krše. Pa tako postoje: ***Anomalije organizacije percepcije modela*** i ***Anomalije percepcije razlikovanja***.

3.5.1. Anomalije organizacije percepcije modela

Percepcija je proces opažanja, kojim mozak organizira podatke dospjele iz raznih osjetila i koje zatim interpretira tvoreći smislenu cjelinu. Organizacija percepcije modela je u biti organizacija podataka koje iz modela dobiva mozak promatrača. U predloženoj taksonomiji anomalija modela poslovnih procesa ***Anomalije organizacije percepcije*** modela sastoji se od sljedećih grupa anomalija: ***Anomalije jednostavnosti***, ***Anomalije sličnosti***, ***Anomalije prisnosti***, ***Anomalije grupiranja***, ***Anomalije povezanosti*** i ***Anomalije prisnosti.***

3.5.1.1. Anomalije jednostavnosti

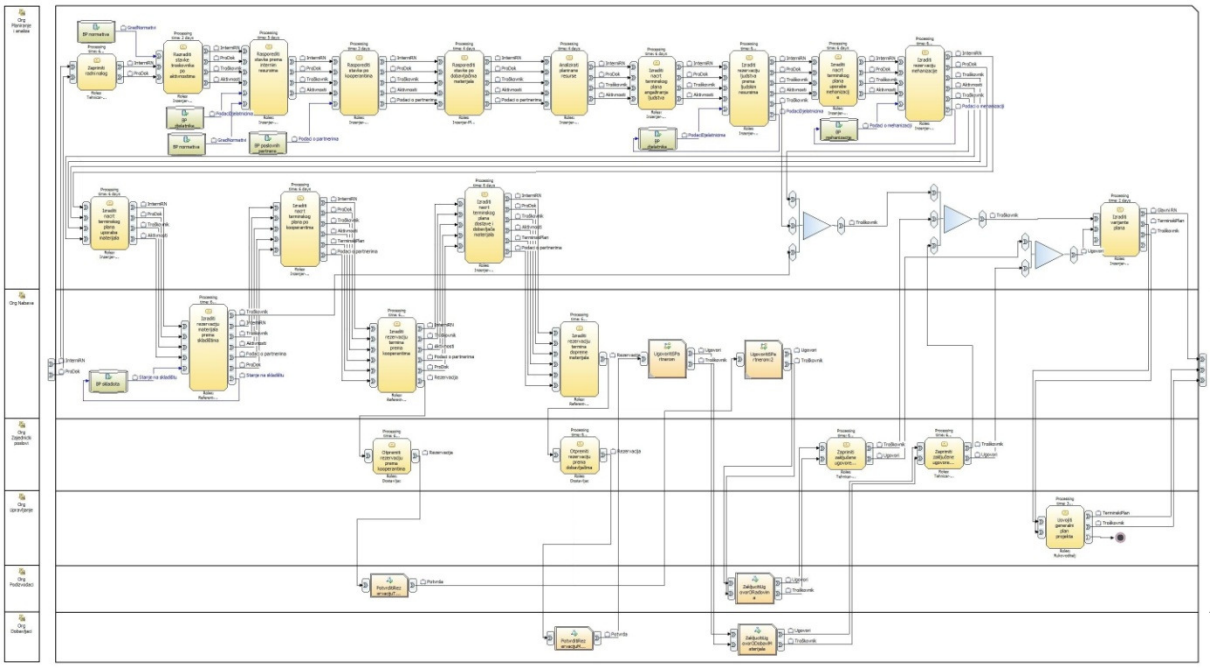
Anomalije jednostavnosti nastaju kršenjem estetskog zakona jednostavnosti. Taj zakon govori o tome da treba težiti pojednostavljenju strukture modela koliko je to moguće. Prikazivanje svih detalja pojedinog poslovnog procesa komplicira njegovo promatranje u smislu potrebnom za njegovo potpuno razumijevanje. Veliki i gusti modeli su teški za čitanje, pa je potrebno selektirati informacije koje će se prikazati u modelu. Percepcija je proces kojim mozak organizira podatke dospjele iz raznih osjetila i interpretira ih tvoreći smislenu cjelinu. Kognitivnost označava spoznaju, to jest mentalne procese za koje se pretpostavlja da ističu ponašanje. Ovaj pojam pokriva široki raspon područja istraživanja, istražujući pitanja o djelovanju pamćenja, pozornosti, percepcije, predstavljanju znanja, mišljenja, kreativnosti i rješavanju problema. Granice percepcije i kognitivne granice glavni su ograničavajući faktori veličine modela. Općenito, može se reći da se složenost čitanja modela povećava kvadratom veličine dijagrama.

Temeljne anomalije koje spadaju u kategoriju ***Anomalija jednostavnosti*** su: ***Previše elemenata u modelu*** i ***Neselektivni prikaz elemenata u modelu***.

Previše elemenata u modelu

(eng. *Too many elements in model*)

Opis - kognitivni limit broja elemenata u dijagramu određen je limitom kapaciteta memorije. Obično se uzima brojka od sedam plus minus dva [28]. Neki autori govore o brojci 5 +/-2 [55]. Kad se prijeđe ova brojka nastaje stanje kognitivnog preopterećenja i razumijevanje modela opada. Najučinkovitiji način smanjenja složenosti modela je njegova podjela na manje dijelove, te modularizacija pojedinih dijelova dijagrama. Sljedeća slika (Slika 97.) prikazuje primjer modela poslovnog procesa sa previše elemenata u modelu, što dovodi do nepreglednosti modela čime se gubi temelj same svrhe modela. U predloženoj taksonomiji anomalija modela poslovnih procesa anomalija ***Previše elemenata u modelu*** dio je grupe anomalija ***Anomalije organizacije percepcije*** i podgrupe ***Anomalije jednostavnosti***.



Slika 97. Primjer anomalije ***Previše elemenata u modelu***

Način prepoznavanja anomalije – ako se prihvati pravilo da bi optimalni broj elemenata u modelu trebao biti 7 +/- 2 elementa, tj. da se kao anomalija uzmima svaki model koji ima više od 9 elemenata u modelu, onda pravilo za pronalaženje te anomalije izgleda ovako:

- Model poslovnog procesa sadrži ukupni broj elemenata veći od 9.

```
context Process
inv TooManyElements:
        (count(self.GraphicalObjects())> 9)
        implies
        TooManyElementsAnomaly()
```

Kôd 154. Način prepoznavanja anomalije ***Previše elemenata u modelu***

Način rješavanja anomalije - pravilo od 9 elemenata ne treba slijepo slijediti, naime vrlo često poslovni proces može imati više elemenata, a da ipak model procesa izgleda pregledno i jasno. Iz tog razloga preporuka je da se kod nailaska na modele s većim brojem elemenata daje obavijest korisniku da model može imati previše elemenata što utječe na njegovu preglednost.

```
FIX: TooManyElementsAnomaly()
   strMessage = "There is anomaly too many elements in model"
```

Kôd 155. Način rješavanja anomalije ***Previše elemenata u modelu***

Neselektivni prikaz elemenata u modelu

(eng. *Unselective elements in model*)

Opis - previše informacija na modelu može dovesti do njegovog otežanog čitanja. U tom slučaju velika količina informacija djeluje kao ograničavajući faktor. Stoga je potrebno dobro procijeniti koje informacije treba, a koje ne treba prikazati na modelu s ciljem njegovog pojednostavljenja. S obzirom da je vrlo teško procijeniti što je važno prikazati na modelu, a što nije, ta odluka ostaje na projektantu modela da ovisno o cilju modela neke elemente naglasi, a druge da ignorira. Prethodna slika (Slika 97.) može isto poslužiti i kao primjer neselektivnog prikaza podataka u modelu, neselektivni prikaz može se odnositi na sve elemente u modelu, dakle aktivnosti, sadržaje razmjene, elementi odluke, itd. Potrebno je dobro odmjeriti i promisliti koji elementi modela daju modelu potrebnu vrijednost, a koji elementi unose nepotrebnu konfuziju u model. U predloženoj taksonomiji anomalija modela poslovnih procesa anomalija ***Neselektivni prikaz elemenata u modelu*** dio je grupe anomalija ***Anomalije organizacije percepcije*** i podgrupe ***Anomalije jednostavnosti***.

Način prepoznavanja anomalije – temelji se na sljedećim pravilima:

- model poslovnog procesa sadrži aktivnosti koje ne spadaju u isti kontekst procesa,
- model poslovnog procesa prikazuje sadržaje razmjene koji nisu bitni za prikaz modela poslovnog procesa i
- model poslovnog procesa koristi resurse koji nisu bitni za prikaz modela poslovnog procesa.

Ako se navedena pravila opišu pseudokodom, onda to izgleda ovako:

```
1  RULE: UnselectiveElementsInModel
2  IF: Count(Process.GraphicElements.Context)>1
3  AND Count(Process.BusinessItems.Context)>1
4  AND Count(Process.Resources.Context)>1
5  THEN:
6  UnselectiverElementsInModelAnomaly
7  ENDIF
```

Kôd 156. Način prepoznavanja anomalije ***Neselektivni prikaz elemenata u modelu***

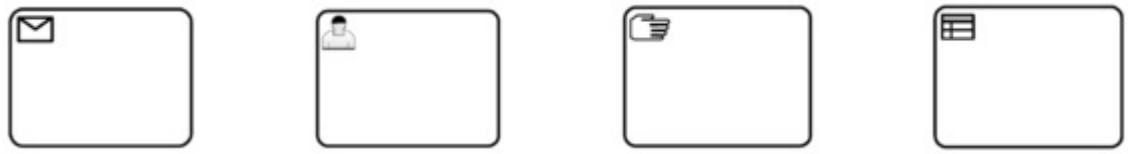
Način rješavanja anomalije – u slučaju nailaska na navedenu anomaliju korisnika je potrebno obavijestiti da navedena anomalija postoji u modelu i da treba korigirati model.

```
1  FIX: UnselectiverElementsInModelAnomaly
2     strMessage = "There is anomaly unselective elements in model"
```

Kôd 157. Način rješavanja anomalije ***Neselektivni prikaz elemenata u modelu***

3.5.1.2. Anomalije sličnosti

Zakon sličnosti govori da slične elemente modela treba grupirati zajedno. Ukoliko u modelu postoji više elemenata sličnog tipa, poželjno je takve elemente grupirati zajedno, na taj način moguće je povećati njegovu čitljivost.



Aktivnost zaprimanja | Aktivnost korisnika | Manualna aktivnost | Aktivnost poslovnih pravila

Slika 98. Elementi sa potencijalnom ***Anomalijom sličnosti***

U predloženoj taksonomiji anomalija modela poslovnih procesa grupa anomalija pod nazivom ***Anomalije sličnosti*** modela dio su ***Anomalija organizacije percepcije*** i sadrže sljedeće anomalije: ***Slični elementi nisu grupirani zajedno***, ***Slični elementi nemaju iste boje***, ***Slični elementi nemaju istu veličinu***, ***Slični elementi nemaju isti oblik.***

Slični elementi nisu grupirani zajedno

(eng. *Similar elements are not grouped together*)

Opis - anomalije sličnosti nastaju zbog kršenja zakona sličnosti. Pod anomalije sličnosti mogu se uvrstiti situacije u kojima ***Slični elementi nisu grupirani zajedno***, zatim kad ***Slični elementi nemaju iste boje***, kad ***Slični elementi nemaju istu veličinu*** i kad ***Slični objekti nemaju isti oblik***. Alati obično poštuju pravila zakona sličnosti, pa tako redovno isti elementi imaju iste boje. Problem se javlja u alatima koji svoje elemente ne boje, nego elementi ostaju crno-bijeli, u tom slučaju jedina razlika predstavlja mala oznaka koja ih međusobno razlikuje. Pa se tako vrlo lako mogu zamijeniti elementi aktivnosti zaprimanja, aktivnosti korisnika, manualna aktivnost ili aktivnost poslovnih pravila. Navedeni elementi imaju isti obris, dok se međusobno razlikuju po oznakama unutar elementa, u slučaju smanjivanja veličine objekta, lako takvi elementi mogu postati nečitljivi i kao takvi se mogu krivo interpretirati od strane korisnika. Prema predloženoj taksonomiji anomalija, anomalija ***Slični elementi nisu grupirani zajedno*** nalazi se u podgrupi ***Anomalije sličnosti*** modela koja je dio su ***Anomalija organizacije percepcije***.

Način prepoznavanja anomalije - za prepoznavanje ove vrste anomalija potrebno je kategorizirati grafičke elemente na način kojim se mogu slični objekt grupirati pod zajedničku kategoriju.

Pravila prepoznavanja navedene anomalije su sljedeća:

- model poslovnog procesa ima više od jednog elementa,
- u modelu poslovnog procesa postoje elementi čije kategorija objekta ima obilježje sličnog objekta i
- u modelu poslovnog procesa broj sličnih objekata u modelu procesa je veći od 1.

Ako se navedena pravila prikažu OCL pravilima, onda to izgleda ovako:

```
1  context Process
2  inv ElementSimilarity:
3     (count(self.GraphicalObjects())> 1) and
4     (count(select(self.GraphicalObject.Category -> SimilarObject()))>1))
5     implies
6     ElementSimilarityAnomaly()
```

Kôd 158. Način prepoznavanja anomalije ***Slični elementi nisu grupirani zajedno***

Način rješavanja anomalije - za rješavanje ove vrste anomalija, treba dobro postaviti grafičke objekte modela, na način da oblikom ne podsjećaju jedan na drugi. Ako je već neizbježno elemente oblikovati na taj način, onda bi trebalo takve elemente razlikovati temeljnom bojom za svaku kategoriju grafičkog elementa zasebno.

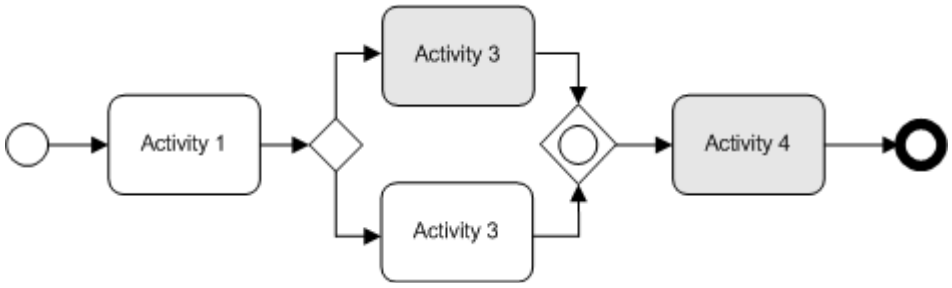
```
FIX: ElementSimilarityAnomaly
IF Process.graphicalElements.Count > 1
THEN
FOR EACH Process.graphicalElement in Process.graphicalElements
  FOR EACH Process.graphicalElement.Category in
     Process.graphicalElement.Categorys
     Process.graphicalElement.Color = Process.graphicalElement.DefaultColor
  NEXT
NEXT
ENDIF
```

Kôd 159. Način rješavanja anomalije ***Slični elementi nisu grupirani zajedno***

Slični elementi nemaju iste boje

(eng. *Similar elements have diferent colours*)

Opis - elementi istih karakteristika moraju imati iste boje. Moderni alati za modeliranje već unaprijed imaju definirane boje određenih tipova elemenata čime povećavaju čitljivost modela. No, treba biti pažljiv u slučajevima kad alat za modeliranje dozvoljava promjenu boje elemenata, jer bi se onda lako mogao zamijeniti s nekim drugim elementom. Prema predloženoj taksonomiji anomalija modela poslovnih procesa, anomalija ***Slični elementi nemaju iste boje*** nalazi se u grupi ***Anomalija organizacije percepcije*** u podgrupi ***Anomalije sličnosti***.



Slika 99. Anomalija ***Slični elementi nemaju iste boje***

Način prepoznavanja anomalije – za prepoznavanje ove anomalije primjenjuje se sljedeća pravila:

- model poslovnog procesa ima više od jednog elementa i
- u modelu poslovnog procesa postoje grafički elementi iste kategorije koji imaju različite boje.

Ako se navedena pravila opišu pomoću pseudokoda, onda to izgledala ovako:

```
RULE: SimilarElementsDiferentColor
IF: Process.graphicalElements.Count > 1
THEN:
FOR EACH Process.graphicalElement in Process.graphicalElements
   FOR EACH Process.graphicalElement.Category in
             Process.graphicalElement.Categorys
        IF (Process.graphicalElement.Color <>
          Process.graphicalElement.Category.Color)
        THEN SimilarElementsDiferentColorAnomaly
        ENDIF
    NEXT
NEXT
ENDIF
```

Kôd 160. Način prepoznavanja anomalije ***Slični elementi nemaju iste boje***

Način rješavanja anomalije - s obzirom da proizvođači alata za modeliranje iz raznih razloga mogu u alate ugraditi mogućnost da elementi iste kategorije imaju različite boje, takva anomalija ne utječe na rezultate performansi modela ili na preciznost samog modela. Iz navedenih razloga korisnika samo treba obavijestiti da u modelu postoje elementi iste kategorije koji imaju različite boje.

Ako bi išli na automatiziranje rješavanja ove anomalije, onda je najbolje sve grafičke objekte unutar modela procesa postaviti na osnovnu boju pojedine kategorije objekata.

```
FIX: SimilarElementsDiferentColorAnomaly
IF: Process.graphicalElements.Count > 1
THEN:
  FOR EACH Process.graphicalElement in Process.graphicalElements
    FOR EACH Process.graphicalElement.Category in Process.graphicalElement.Categorys
       Process.graphicalElement.Color = Process.graphicalElement.DefaultColor
    NEXT
  NEXT
ENDIF
```

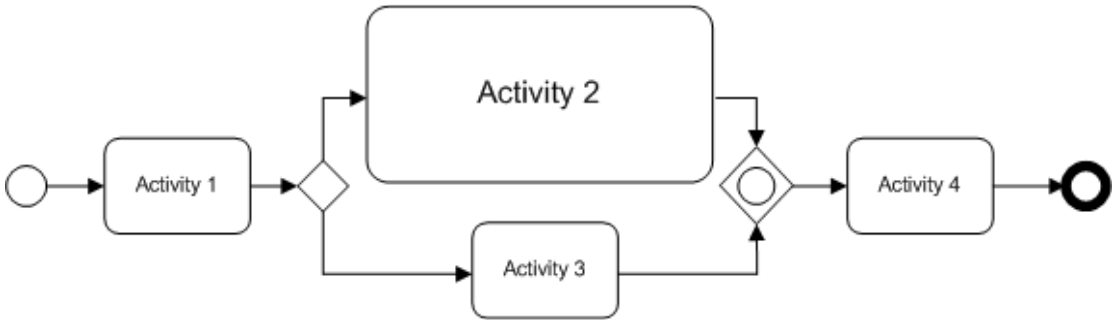
Kôd 161. Način rješavanja anomalije ***Slični elementi nemaju iste boje***

Slični elementi nemaju istu veličinu

(eng. *Similar elements have diferent size*)

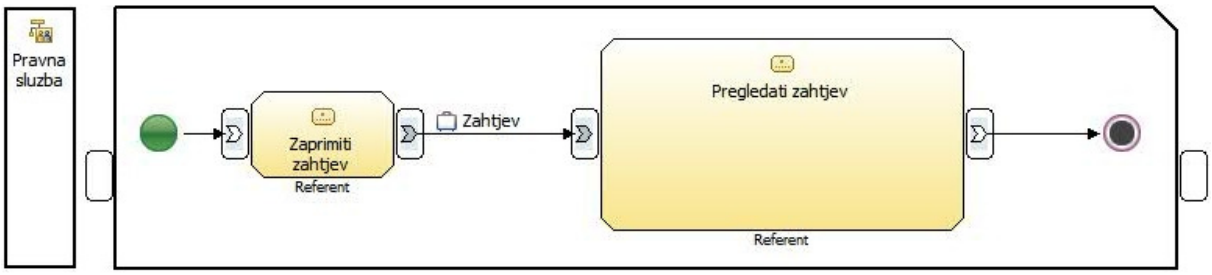
Opis - objekti koji imaju različitu (veću) veličinu od susjednog elemenata mogu privući pažnju [22]. S druge strane, objekti iste veličine mogu izgledati kao grupa iako to nisu [2]. Potrebno je obratiti pažnju na takve detalje i kontrolirati veličinu pojedinih elemenata da bi se dobila potrebna percepcija modela. Prema predloženoj taksonomiji anomalija modela

poslovnih procesa, anomalija ***Slični elementi nemaju istu veličinu*** nalazi se u grupi ***Anomalija organizacije percepcije*** u podgrupi ***Anomalije sličnosti***.



Slika 100. Anomalija ***Slični elementi nemaju istu veličinu***

Primjer – slijedeća slika (Slika 101.) pokazuje slučaj anomalije s elementima različitih veličina. Već na prvi pogled vidi se da je aktivnost **Pregledati zahtjev** naglašen, čime promatrač može dobiti dojam da je aktivnost **Pregledati zahtjev** važniji od aktivnosti **Zaprimiti zahtjev.**



Slika 101. Primjer anomalije ***Slični elementi nemaju istu veličinu***

Način prepoznavanja anomalije – ako je grafički element iste kategorije dva puta veći po širini i visini od ostalih elemenata, tada se može reći da u modelu postoji anomalija s elementima različitih veličina.

Pravila prepoznavanja navedene anomalije su sljedeća:

- model poslovnog procesa ima više od jednog elementa i
- u modelu poslovnog postoji element iste kategorije, a da li mu je širina i visina dvostruko veća od prethodnog elementa.

Ako se navedena pravila opišu u pseudokodu, onda to izgleda ovako:

```
RULE: SimilarElementsDiferentSize
IF Process.graphicalElements.Count > 1
  FOR EACH Process.graphicalElement in Process.graphicalElements
     FOR EACH Process.graphicalElement.Category in Process.graphicalElements.Categorys
     intCount = +intCount
      IF (intCount = 1)
      THEN
               PreGraphicalObjectCategory = Process.graphicalElement.Category
               PreGraphicalObjectWidth = Process.graphicalElement.Width
               PreGraphicalObjectHeight = Process.graphicalElement.Height
       ELSE
               CuGraphicalObjectCategory = Process.graphicalElement.Category
               CuGraphicalObjectWidth = Process.graphicalElement.Width
               CuGraphicalObjectHeight = Process.graphicalElement.Height

              IF CuGraphicalObjectCategory = PreGraphicalObjectCategory
                  THEN
                      IF CuGraphicalObjectWidth > PreGraphicalObjectWidth
                      AND CuGraphicalObjectHeight > PreGraphicalObjectHeight

                      THEN
                             IF (PreGraphicalObjectWidth * 2) <
                                (CuGraphicalObjectWidth)
                             AND (PreGraphicalObjectHeight * 2) <
                                 (CuGraphicalObjectHeight)
                             THEN SimilarElementsDiferentSizeAnomaly
                             ENDIF
                      ENDIF
                      PreGraphicalObjectWidth = Process.graphicalElement.Width
                      PreGraphicalObjectHeight = Process.graphicalElement.Height
                   ENDIF
               ELSE
                  PreGraphicalObjectCategory = Process.graphicalElement.Category
                  PreGraphicalObjectWidth = Process.graphicalElement.Width
                  PreGraphicalObjectHeight = Process.graphicalElement.Height
               ENDIF
        ENDIF
   NEXT
  NEXT
ENDIF
```

Kôd 162. Način prepoznavanja anomalije ***Slični elementi nemaju istu veličinu***

Način rješavanja anomalije - za automatiziranje rješavanja ove anomalije, najbolje je sve grafičke objekte unutar modela procesa postaviti na osnovnu dimenziju pojedine kategorije objekata.

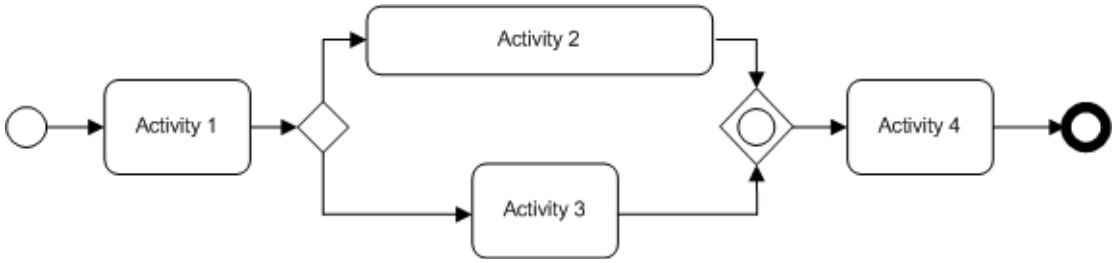
```
FIX: SimilarElementsDiferentSizeAnomaly
IF: Process.graphicalElements.Count > 1
THEN:
   FOR EACH Process.graphicalElement in Process.graphicalElements
     FOR EACH Process.graphicalElement.Category in
       Process.graphicalElement.Categorys
       Process.graphicalElement.Width = Process.graphicalElement.DefaultWidth
       Process.graphicalElement.Height = Process.graphicalElement.DefaultHeight
     NEXT
   NEXT
ENDIF
```

Kôd 163. Način rješavanja anomalije ***Slični elementi nemaju istu veličinu***

Slični elementi nemaju isti oblik

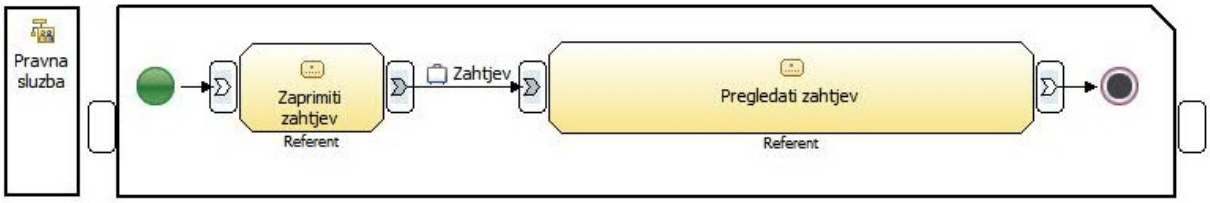
(eng. *Similar elements have diferent shape*)

Opis - kao što je već spomenuto, današnji alati za modeliranje omogućavaju da pojedini tipovi elemenata kao što su aktivnosti ili grane budu različitih boja, čime se slične vrste elemenata brzo međusobno mogu razlikovati. Modeli poslovnih procesa pojedine elemente razlikuju i svojim oblikom, što je isto posljedica zakona sličnosti. Dakle, sve aktivnosti iste vrste moraju imati isti oblik, svi događaji unutar modela moraju imati isti oblik, itd. Na taj način izbjegavaju se krive interpretacije modela. Prema predloženoj taksonomiji, anomalija modela poslovnih procesa, anomalija ***Slični elementi nemaju isti oblik*** nalazi se u grupi ***Anomalija organizacije percepcije*** u podgrupi ***Anomalije sličnosti***.



Slika 102. Anomalija ***Slični objekti nemaju isti oblik***

Primjer - pokazuje anomaliju ***Slični objekti nemaju isti oblik***. Promatrač ovoga modela može steći dojam da je aktivnost **Pregledati zahtjev** važnija od aktivnosti **Zaprimiti zahtjev**.



Slika 103. Primjer anomalije ***Slični objekti nemaju isti oblik***

Način prepoznavanja anomalije – ako je grafički element iste kategorije dva puta veći po širini ili visini od ostalih elemenata, tada se može reći da u modelu postoji anomalija ***Slični objekti nemaju isti oblik***. Navedena anomalija prepoznaje se pomoću sljedećih pravila:

- u modelu poslovnog procesa ima više od jednog elementa i
- u modelu poslovnog procesa postoji element iste kategorije, a da li mu je širina ili visina dvostruko veća od prethodnog elementa.

Ako se navedena pravila opišu pseudokodom, onda to izgleda ovako:

```
 1  RULE: SimilarElementsDiferentShape
 2  IF: Process.graphicalElements.Count > 0
 3  THEN:
 4     FOR EACH Process.graphicalElement in Process.graphicalElements
 5       FOR EACH Process.graphicalElement.Category in Process.graphicalElements.Categorys
 6       intCount = +intCount
 7
 8        IF (intCount = 1)
 9        THEN
10                PreGraphicalObjectCategory = Process.graphicalElement.Category
11                PreGraphicalObjectWidth = Process.graphicalElement.Width
12                PreGraphicalObjectHeight = Process.graphicalElement.Height
13         ELSE
14                CuGraphicalObjectCategory = Process.graphicalElement.Category
15                CuGraphicalObjectWidth = Process.graphicalElement.Width
16                CuGraphicalObjectHeight = Process.graphicalElement.Height
17
18               IF CuGraphicalObjectCategory = PreGraphicalObjectCategory
19                 THEN
20                        IF CuGraphicalObjectWidth > PreGraphicalObjectWidth
21                        AND CuGraphicalObjectHeight > PreGraphicalObjectHeight
22                        THEN
23                                IF ((PreGraphicalObjectWidth * 2) <
24                                   (CuGraphicalObjectWidth))
25                                OR ((PreGraphicalObjectHeight * 2) <
26                                    (CuGraphicalObjectHeight))
27                                THEN SimilarElementsDiferentShapeAnomaly
28                        ENDIF
29
30                        PreGraphicalObjectWidth = Process.graphicalElement.Width
31                        PreGraphicalObjectHeight = Process.graphicalElement.Height
32                    ENDIF
33                ELSE
34                   PreGraphicalObjectCategory = GraphicalObject.Category
35                   PreGraphicalObjectWidth = GraphicalObject.Width
36                   PreGraphicalObjectHeight = GraphicalObject.Height
37                ENDIF
38       END IF
39      NEXT
40    NEXT
41  ENDIF
```

Kôd 164. Način prepoznavanja anomalije ***Slični objekti nemaju isti oblik***

Način rješavanja anomalije - nakon što algoritam za prepoznavanje otkrije ovaj tip anomalije, najbolje je obavijestiti korisnika da postoji navedena anomalija, te da je moguće pokrenuti proceduru za ispravljanje takve anomalije u modelu.

Ako model poslovnog procesa ima više od jednog elementa, tada za svaki grafički element u skupu grafičkih elemenata poslovnog procesa i za svaku kategoriju unutar svih kategorija grafičkih elemenata u modelu poslovnih procesa procedura postavlja predefinirane osnovne vrijednosti visine i širine grafičkih elemenata. Navedene vrijednosti određene su konstantama `Process.graphicalElement.DefaultWidth` i `Process.graphicalElement.DefaultHeight` u minimalnoj specifikaciji koja slijedi. Navedene konstante određene su za svaku kategoriju grafičkog elementa zasebno. Vrijednost konstanti ovisi o obliku grafičkog elementa, njegovoj svrsi, informaciji moraju prikazivati, itd.

```
FIX: SimilarElementsDiferentShapeAnomaly
IF: Process.graphicalElements.Count > 1
THEN:
  FOR EACH Process.graphicalElement in Process.graphicalElements
    FOR EACH Process.graphicalElement.Category in
     Process.graphicalElement.Categorys
     Process.graphicalElement.Width = Process.graphicalElement.DefaultWidth
     Process.graphicalElement.Height = Process.graphicalElement.DefaultHeight
    NEXT
  NEXT
ENDIF
```

Kôd 165. Način rješavanja anomalije ***Slični objekti nemaju isti oblik***

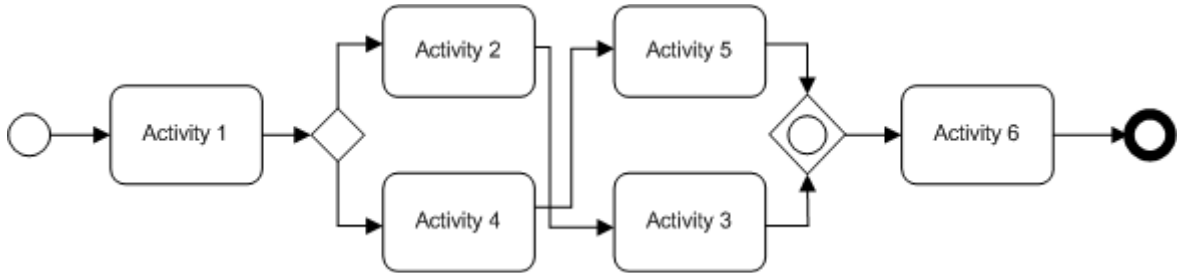
3.5.1.3. Anomalija kontinuiranosti

Anomalije kontinuiranosti predstavljaju podgrupu ***Anomalija organizacije percepcije*** koja pak spada u ***Anomalije prikaza modela***. Grupa ***Anomalije kontinuiranosti*** sadrži samo jednu anomaliju pod nazivom ***Križanje i savijanje veza između elemenata***.

Križanje i savijanje veza između elemenata

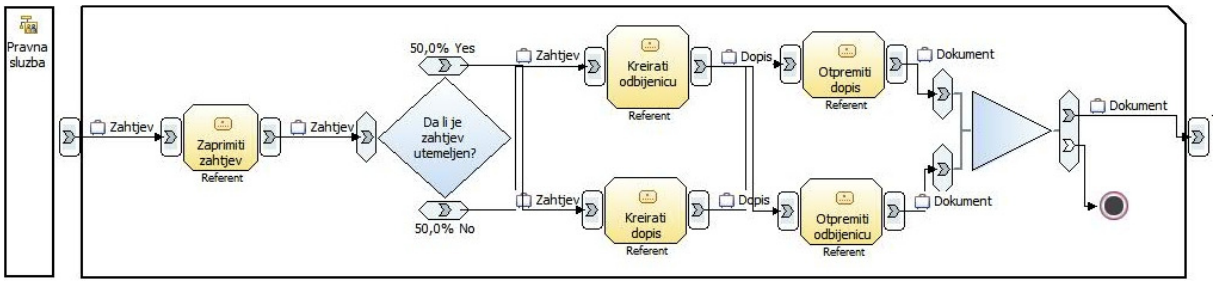
(eng. *Crossing and bending connenction between elements*)

Opis – navedena anomalija naziva se još i ***Anomalija kontinuiranosti***, ona nastaje kod kršenja zakona kontinuiranosti. Zakon kontinuiranosti govori da elementi trebaju biti poredani na način da je vidljiva sljednost. Primjenom ovog zakona pokušavaju se izbjeći križanja i lomljenja veza.



Slika 104. Anomalija ***Križanje i savijanje veza između elemenata***

Primjer – prikazuje model poslovnog procesa sa anomalijom ***Križanja i savijanja veza između elemenata***.



Slika 105. Primjer anomalije ***Križanje i savijanje veza između elemenata***

Način prepoznavanja anomalije – odvija se pomoću sljedećih pravila:

- model poslovnog procesa ima više od jednog elementa,
- u modelu poslovnog procesa povezani elementi iste kategorije nisu na istoj visini i
- element kontrole toka siječe se sa drugim elementom kontrole toka.

Ako se navedena pravila opišu pseudokodom, onda to izgleda ovako:

```
1  RULE: Continuity
2  IF: ((Control-Flow EXIST BETWEEN Process.graphicalElements) and
3      Process.graphicalElements(n).top <> Process.graphicalElements(n+1).top
4  THEN: ContinuityAnomaly
5  ENDIF
```

Kôd 166. Način prepoznavanja anomalije ***Križanje i savijanje veza između elemenata***

Način rješavanja anomalije - u principu, treba izbjegavati sjecanje kontrolnih tokova, jer to može zbuniti korisnike modela. Navedena anomalija ne utječe na funkcionalnost modela, pa je dovoljno korisnika porukom upozoriti da u modelu postoji anomalija ***Kontinuiranosti*** odnosno anomalija ***Križanja i savijanja veze između elemenata***.

Za automatizaciju ovog rješavanja može se uvesti pravilo da ukoliko se prepozna anomalija ***Križanja i savijanja veza između elemenata*** da grafičke elemente iste kategorije treba poredati na istoj visini i tako da se međusobno ne preklapaju. Prema prvom elementu ostali grafički elementi se ravnaju s obzirom na visinu. S obzirom na poziciju prema lijevom obrubu, početni element je udaljen od lijevog ruba za svoju širinu, dok je sljedeći element udaljen od ruba ovisno o lijevoj poziciji prethodnog elementa, njegovoj širini i po vrijednosti zadane osnovne konstante `Process.graphicalElements.DefaultHorizontalGap`. Navedena vrijednost može varirati ovisno o kategoriji grafičkog elementa.

```
FIX: ContinuityAnomaly
intCount=0
DO UNTIL Process.graphicalElements.Count -1
    intCount =+ intCount
    IF intCount = 1
    THEN
       Process.graphicalElements.DefaultTop = Process.graphiclElement(intCount).Top
       Process.graphicalElement(intCount).Left =
       Process.graphicalElement(intCount).Width
       Process.graphicalElement(intCount).Top =
       Process.graphicalElement(intCount).Top
    ELSE
       Process.graphicalElement(intCount).Left =
       Process.graphicalElement(intCount-1).Width +
       Process.graphicalElement(intCount-1).Left +
       Process.graphicalElements.DefaultHorizontalGap
       Process.graphicalElement(intCount).Top = Process.graphicalElements.DefaultTop
    END IF
LOOP
```

Kôd 167. Način rješavanja anomalije ***Križanje i savijanje veza između elemenata***

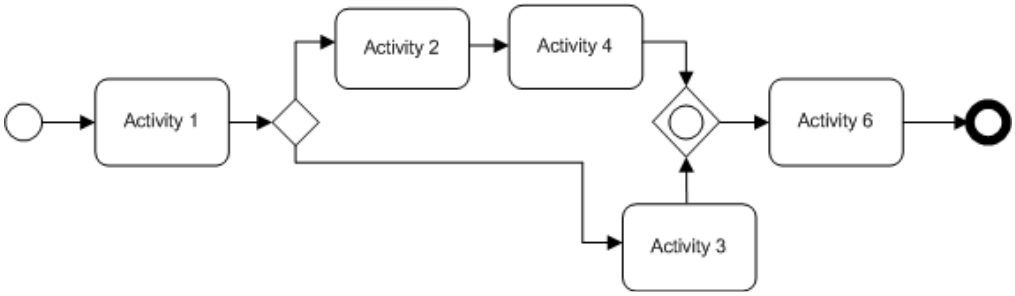
3.5.1.4. Anomalije grupiranja

Anomalije grupiranja bave se problemom preglednosti modela s obzirom na kontekst pojedinih elemenata modela. Prema predloženoj taksonomiji ***Anomalije grupiranja*** spadaju u ***Anomalije organizacije percepcije*** koja pak spada u ***Anomalije prikaza modela***.

Nekompaktni i razvučeni model

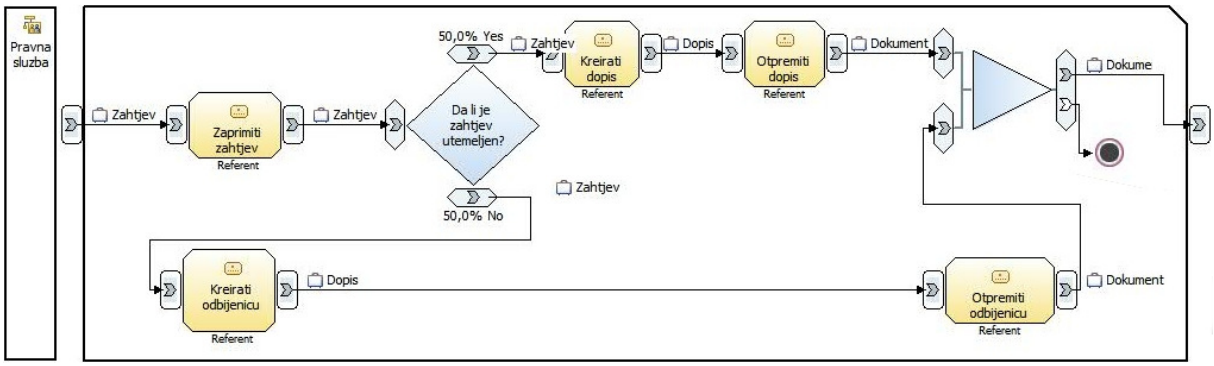
(eng. *Uncompact and stretched model*)

Opis - ovaj tip anomalije posljedica je nepridržavanja zakona grupiranja (aproksimiteta). Zakon grupiranja govori da elementi koji su blizu jedan uz drugi da su zajedno grupirani. Dijagrami bi trebali biti kompaktni s ciljem lakšeg čitanja. Grane koje treba promatrati zajedno trebaju biti smješteni zajedno. Treba izbjegavati jako duge sekvence, jer se teško čitaju. Prema predloženoj taksonomiji anomalija ***Nekompaktni i razvučeni model*** dio je grupe ***Anomalije grupiranja*** koja pak spada u ***Anomalije organizacije percepcije***, a koja je dio ***Anomalija prikaza modela***.



Slika 106. Primjer anomalije ***Nekompaktni i razvučeni model***

Primjer – prikazuje anomaliju ***Nekompaktni i razvučeni model***.



Slika 107. Primjer anomalije ***Nekompaktni i razvučeni model***

Način prepoznavanja anomalije - da bi se mogla prepoznati i riješiti ova anomalija, moraju postojati kategorizirani objekti svrsani po određenim kriterijima. Kriteriji mogu biti organizacijske jedinice, resursi koji se koriste tijekom izvođenja procesa ili neki drugi proizvoljni kriterij kategoriziranja.

Navedena anomalija može se prepoznati pomoću sljedećih pravila:

- model poslovnog procesa ima više od jednog elementa,
- model sadrži grafički element kategorije koji se horizontalno ili vertikalno nalazi između dva spojena elementa iste kategorije i
- mora postojati i uvjet koji govori o tome da se neki element nalazi previše udaljen od drugog elementa.

Ako se navedena pravila prikažu pomoću pseudokoda:

```
RULE: UncompactModel
intCount = 0
FOR EACH Process.graphicalElement in Process.graphicalElements
 IF (Process.graphicalElement(n+1).left - Process.graphicalElement(n).left) >
      (Process.graphicalElement(n).width * 2)
  THEN
      UncompactModelAnomaly
  ENDIF
NEXT
```

Kôd 168. Način prepoznavanja anomalije ***Nekompaktni i razvučeni model***

Način rješavanja anomalije – temelji se na reorganizaciji pozicije elemenata. Prijedlog rješenja svaki element pozicionira na jednake udaljenosti jedan od drugog i to obično u vrijednosti vlastite širine i visine.

```
FIX: UncompactModelAnomaly
DO UNTIL Process.graphicalElements.Count -1
        IF: ((Process.graphicalElements(n).left +
            Process.graphicalElements(n).width) -
            (Process.graphicalElements(n+1).left +
             Process.graphicalElements(n+1).width)) <
            (Process.graphicalElements(n).width))
        THEN
            Process.graphicalElement(n).left = Process.graphicalElement(n+1).left
        -   (Process.graphicalElement(n).width * 2)
        ENDIF
        IF  ((Process.graphicalElements(n).top +
            Process.graphicalElement(n).height) -
            (Process.graphicalElements(n+1).top +
             Process.graphicalElements(n+1).height)) <
            (Process.graphicalElements(n).height)
        THEN
            Process.graphicalElement(n).top = Process.graphicalElement(n+1).top -
            (Process.graphicalElement(n).height * 2)
        ENDIF
LOOP
```

Kôd 169. Način rješavanja anomalije ***Nekompaktni i razvučeni model***

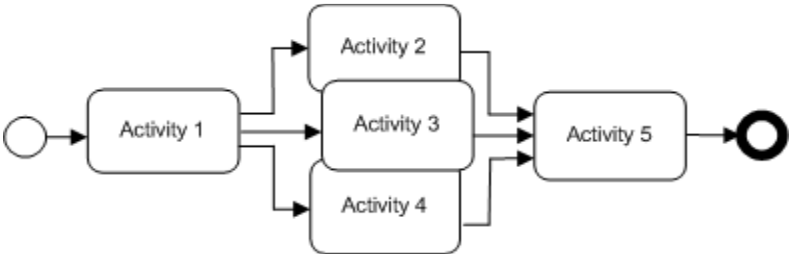
3.5.1.5. Anomalije povezanosti

Anomalije povezanosti bave se problemom preklapanja elemenata u modelu što dovodi do nepreglednosti te mogućoj krivoj interpretaciji modela. Prema predloženoj taksonomiji anomalija modela poslovnih procesa ***Anomalije povezanosti*** spadaju u ***Anomalije organizacije percepcije*** koja pak spada u ***Anomalije prikaza modela.***

Preklapanje elemenata modela

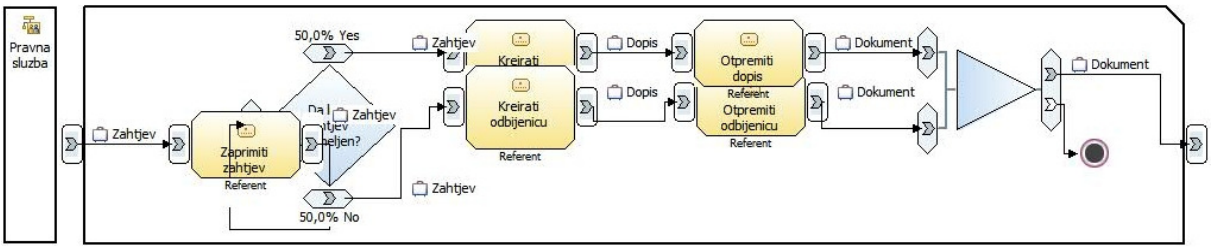
(eng. *Overlap elements*)

Opis - ***anomalija povezanosti*** odnosno anomalija ***Preklapanje elemenata modela*** nastaje kad se model ne pridržava zakona povezanosti. To je zakon koji govori da se elementi koji su zajedno povezani ili se preklapaju, mogu smatrati cjelinom. U modelima valja izbjegavati međusobno preklapanje i dodirivanje elemenata.



Slika 108. Anomalija ***Preklapanje elemenata modela***

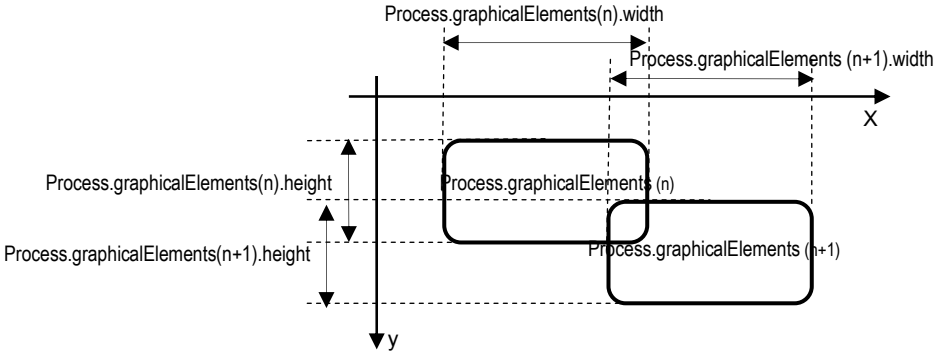
Primjer – na primjeru se vidi model poslovnog procesa u kojem se pojedini elementi procesa preklapaju. Time model gubi na preglednosti, te su moguće krive interpretacije modela.



Slika 109. Primjer ***Anomalije preklapanja elemenata***

Način prepoznavanja anomalije – temelje se na sljedećim pravilima:

- model poslovnog procesa ima više od jednog elementa,
- u modelu poslovnog procesa postoji element čija je lijeva pozicije i širina manja od lijeve pozicije i širine sljedećeg elementa i
- u modelu poslovnog procesa postoji element čija je gornja pozicija i visina manja od gornje pozicije i visine sljedećeg elementa.



Slika 110. Shematski prikaz anomalije ***Preklapanja elemenata modela***

Ako se navedena pravila prikažu pomoću pseudokoda, onda to izgleda ovako:

```
RULE: Overlap
Process.graphicalElement objElementBefore = Nothing
int objElementCount = 0
FOR EACH Process.graphicalElement in Process.graphicalElements
    IF objElementCount > 0 then
        IF: ((objElementBefore.left + objElementBefore.width) -
            (Process.graphicalElement.left + Process.graphicalElement.width)) <
            (objElementBefore.width))
        OR  ((objElementBefore.top + objElementBefore.height) -
            (Process.graphicalElement.top + Process.graphicalElement.height)) <
            (objElementBefore.height)
        THEN
            OverlapAnomaly
        ENDIF
    END IF
    Set objElementBefore as Process.graphicalElement
    objElementCount = +objElementCount
NEXT
```

Kôd 170. Način prepoznavanja ***Anomalije preklapanje elemenata modela***

Način rješavanja anomalije - u slučaju da je u modelu prepoznata ***Anomalija preklapanja elemenata modela***, najbolje je presložiti elemente modela, na način da svaki element bude od drugog udaljen za vrijednost svoje širine, odnosno visine. Prije promjene pozicije elemenata unutar modela poslovnog procesa svakako treba o tome obavijestiti korisnika.

```
1   FIX: OverlapAnomaly
2   Process.graphicalElement objElementBefore = Nothing
3   int objElementCount = 0
4   FOR EACH Process.graphicalElement in Process.graphicalElements
5       IF objElementCount > 0 then
6           IF: ((objElementBefore.left + objElementBefore.width) -
7               (Process.graphicalElement.left + Process.graphicalElement.width)) <
8               (objElementBefore.width))
9           THEN
10              Process.graphicalElement.left = objElementBefore.left +
11              (objElementBefore.width + (Process.graphicalElement.width * 2))
12          ENDIF
13          IF  ((objElementBefore.top + objElementBefore.height) -
14              (Process.graphicalElement.top + Process.graphicalElement.height)) <
15              (objElementBefore.height)
16          THEN
17              Process.graphicalElement.top = objElementBefore.top -
18              (Process.graphicalElement(n).height * 2)
19          ENDIF
20
21      END IF
22      Set objElementBefore as Process.graphicalElement
23      objElementCount = +objElementCount
24  NEXT
```

Kôd 171. Način rješavanja anomalije ***Preklapanje elemenata modela***

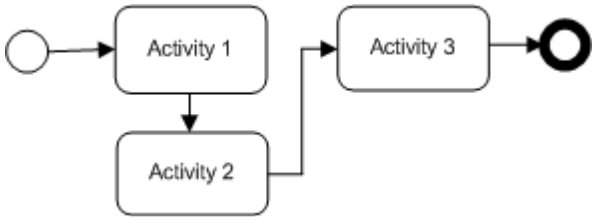
3.5.1.6. Anomalije prisnosti

Navedene anomalije bave se problemom vizualne interpretacije hijerarhije elementa unutar modela. Prema predloženoj taksonomiji ***Anomalije prisnosti*** spadaju u ***Anomalije organizacije percepcije***, s druge strane ***Anomalije prisnosti*** sadrže anomaliju ***Nepoštivanje pozicije hijerarhije elemenata*** modela.

Nepoštovanje pozicije hijerarhije elemenata

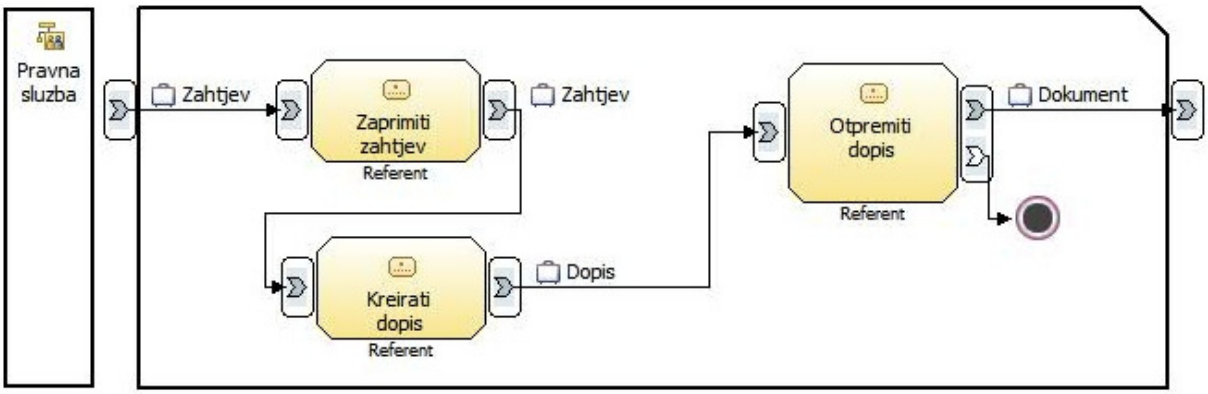
(eng. *Familiarity anomaliy*)

Opis - ***anomalija prisnosti*** odnosno anomalija ***Nepoštovanje pozicije hijerarhije elemenata modela*** nastaje kršenjem zakona prisnosti. Taj zakon govori da će elementi biti grupirani ukoliko su prisni ili familijarni. Pozicijom pojedinih elemenata unutar modela može se postići efekt hijerarhije, čime se model može krivo interpretirati.



Slika 111. Anomalija ***Nepoštovanja pozicije hijerarhije elemenata***

Primjer – prikazuje anomaliju ***Nepoštovanje pozicije hijerarhije elemenata***, u ovome slučaju svi elementi su iste hijerarhije, pa njihovo pozicioniranje može dati dojam da postoji hijerarhija u modelu procesa.



Slika 112. Primjer anomalije ***Nepoštovanje pozicije hijerarhije elemenata***

Način prepoznavanja anomalije - ključno za prepoznavanje ove anomalije je definiranje mreže koja će omogućiti ravnomjeran smještaj elemenata unutar modela poslovnog procesa.

Na taj način mogu se otkriti, da li postoje elementi koje se nižu jedan za drugim, a da im je pozicija jedan iznad drugog.

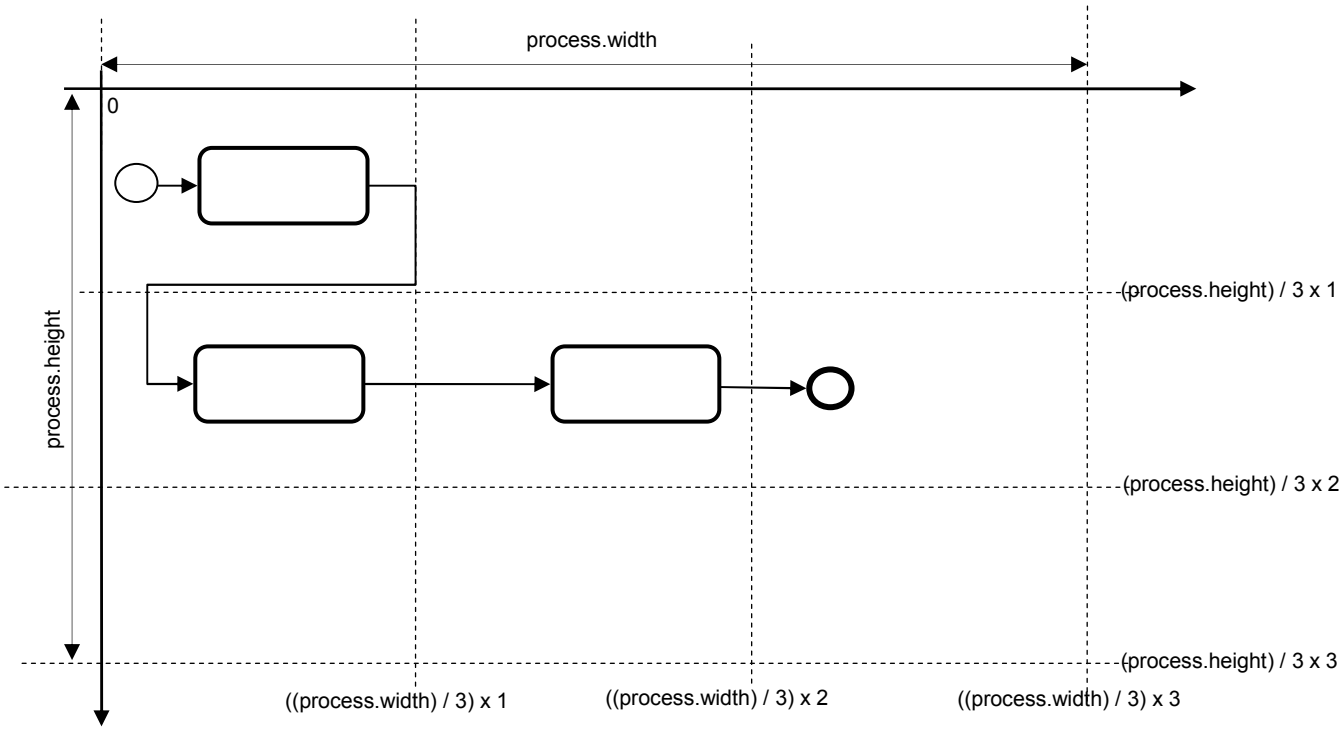
Navedena anomalija može se prepoznati pomoću sljedećih pravila:

- model poslovnog procesa ima više od jednog elemenata iste kategorije i
- u modelu poslovnog procesa postoji element koji je spojen s drugim elementom iste kategorije, a jedan je iznad drugoga i u istoj mjeri udaljen od ruba modela.

Ako se navedena pravila opišu pomoću pseudokoda onda to izgleda ovako:

```
1   RULE: Familiarity
2   Process.graphicalElement objElementBefore = Nothing
3   int objElementCount = 0
4   FOR EACH Process.graphicalElement in Process.graphicalElements
5       IF objElementCount > 0 then
6         IF ((Control-Flow EXIST BETWEEN Process.graphicalElements)
7           And Process.graphicalElement.left = Process objElementBefore.left
8           And Process.graphicalElement.top > objElementBefore.left
9           And Process.graphicalElement.Category = objElementBefore.Category
10        THEN FamiliarityAnomaly
11        ENDIF
12      ENDIF
13      Set objElementBefore as Process.graphicalElement
14      objElementCount = +objElementCount
15  NEXT
```

Kôd 172. Način prepoznavanja anomalije ***Nepoštovanje pozicije hijerarhije elemenata***



Slika 113. Shematski prikaz prepoznavanja anomalije ***Nepoštovanje pozicije hijerarhije elemenata***

Način rješavanja anomalije - postoji nekoliko načina na koji se može riješiti:

- svaki elementi iste kategorije koji su međusobno spojeni moraju biti na istoj visini i
- prijelaz u novi red morao bi biti samo ako je element spojen s drugom kategorijom elementa, na taj način elementi iste kategorije ne djeluju hijerarhijski jedan iznad drugog.

```
1   FIX: FamiliarityAnomaly
2   Process.graphicalElement objElementBefore = Nothing
3   int objElementCount = 0
4     FOR EACH Process.graphicalElement in Process.graphicalElements
5       IF objElementCount > 0 THEN
6        IF ((Control-Flow EXIST BETWEEN Process.graphicalElements)
7           And Process.graphicalElement.left = Process objElementBefore.left
8           And Process.graphicalElement.top > objElementBefore.left
9           And Process.graphicalElement.Category = objElementBefore.Category
10       THEN:
11          Process.graphicalElement.top = objElementBefore.top
12          Process.graphicalElement.left = objElementBefore.left +
13                                          (Process.graphicalElement.Width * 2)
14       ENDIF
15      ELSE
16        Process.graphicalElement.top = (Process.height -
17                                          Process.graphicalElement.height)/2
18     ENDIF
19      Set objElementBefore as Process.graphicalElement
20      objElementCount = +objElementCount
21  NEXT
```

Kôd 173. Način rješavanja anomalije ***Nepoštivanje pozicije hijerarhije elemenata***

3.5.2. Anomalije percepcije razlikovanja (izdvajanja)

Anomalije percepcije razlikovanja je skupina uvjetnih anomalija koje utječu na odstupanje od pravila koja olakšavaju percepciju modela. Prema predloženoj taksonomiji ***Anomalije percepcije razlikovanja*** sadrže tri grupe anomalija i to: ***Anomalije simetrije***, ***Anomalije orijentacije*** i ***Anomalije obrisa.***

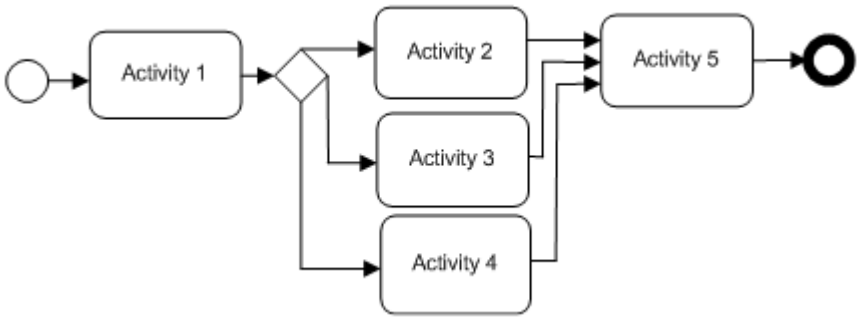
3.5.2.1. Anomalija simetrije

Anomalije simetrije kao zajedničku karakteristiku imaju vizualni debalans i neravnotežu. Modeli koji imaju ***Anomalije simetrije*** su u neravnoteži, prostor za prikazivanje elemenata ne iskorištavaju jednakomjerno, već se primjećuje vizualni debalans pojedinih dijelova i pozicija modela. Prema predloženoj taksonomiji anomalija modela poslovnog procesa ***Anomalije simetrije*** spadaju pod ***Anomalije percepcije razlikovanja***, a sadrži anomaliju ***Necentrirani i nesimetrični model***.

Necentrirani i nesimetrični model

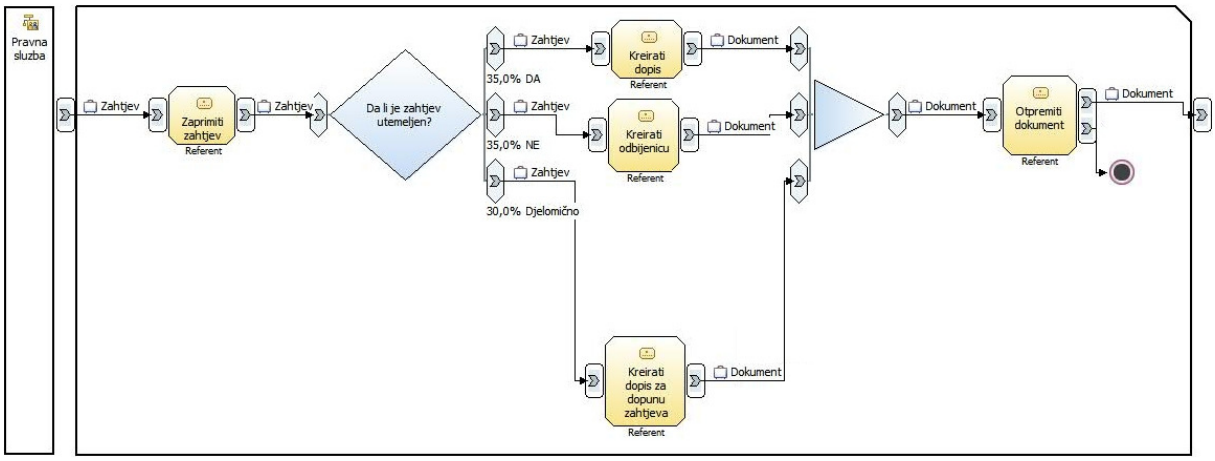
(eng. *Off-center and asymmetric model*)

Opis - anomalija simetrije odnosno anomalija ***Necentrirani i nesimetrični model*** nastaje kad model ne zadovoljava načelo simetrije. Načelo simetrije u modelima poslovnih procesa ogleda se u tome da je model vizualno u balansu bez obzira na kut promatranja. S obzirom na navedeno simetrija je poželjna u modelima poslovnih procesa. Uporaba simetrije u modelu treba biti primijenjena na način da se temeljni elementi procesa prikažu u centru dijagrama, time cijeli model dobiva na «stabilnosti» prikaza. Simetrija može biti horizontalna i vertikalna.



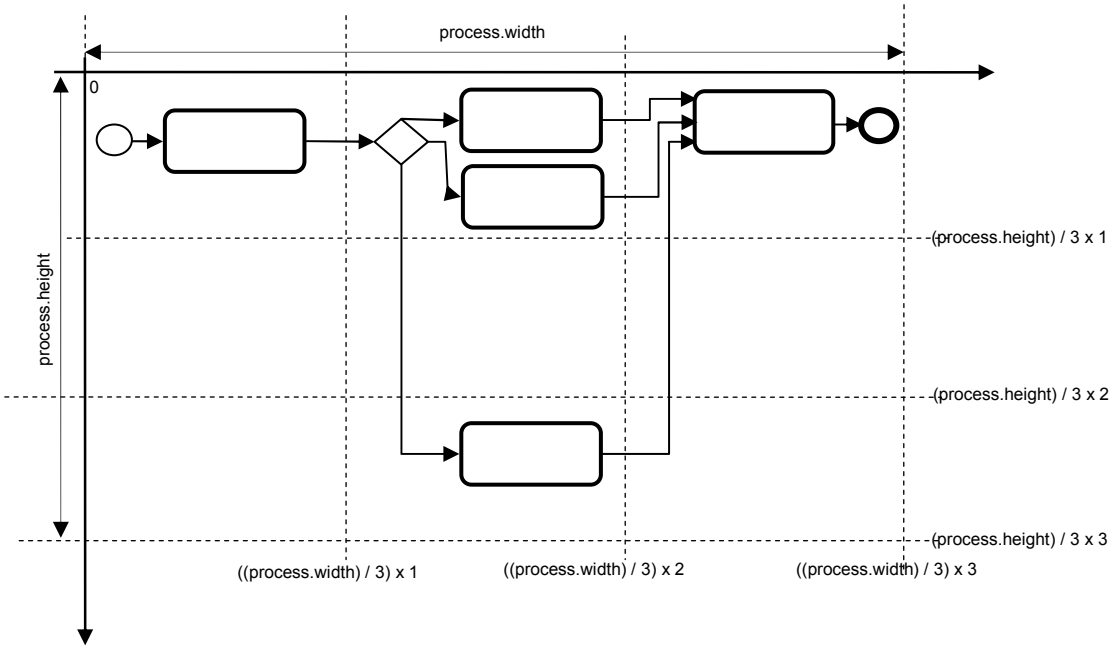
Slika 114. Anomalija ***Necentrirani i nesimetrični model***

Primjer – na sljedećem primjeru prikazan je model sa vertikalnom anomalijom simetrije.



Slika 115. Primjer ***Necentrirani i nesimetrični model***

Način prepoznavanja anomalije - s obzirom da se razlikuju vertikalna i horizontalna anomalija, provjeru asimetričnosti potrebno je izvesti po obje kategorije.



Slika 116. Shematski prikaz pozicije elemenata kod razrade pravila prepoznavanja anomalije ***Necentrirani i nesimetrični model***

```
RULE: HorizontalAsymmetry
Int HorizontalLeft=0
Int HorizontalRight=0
FOR EACH Process.graphicalElement in Process.graphicalElements
       SELECT CASE Process.graphicalElement.left
              CASE 0 to ((process.width / 3) * 1)
                     HorizontalLeft =+ HorizontalLeft
              CASE ((process.width / 3) * 1)  to ((process.width / 3) * 2)
                     HorizontalCenter =+ HorizontalCenter
              CASE ((process.width / 3) * 2)  to ((process.width / 3) * 3)
                     HorizontalRight =+ HorizontalRight
       END SELECT
       IF: (HorizontalLeft <> HorizontalRight)
       THEN: AsymmetryAnomaly  /*HORIZONTAL_ASYMMETRY_ANOMALY*/
       ENDIF
NEXT
```

Kôd 174. Način prepoznavanja anomalije ***Horizontalne asimetrije***

Pravila prepoznavanja ***Horizontalne asimetrije***:

- u modelu poslovnog procesa broj elementa koji se nalaze u lijevom horizontalnom kvadrantu različit je od broja elemenata koji se nalaze u desnom horizontalnom kvadrantu.

```
RULE: VerticalAsymmetry
Int VerticalTop=0
Int VerticalBottom=0
FOR EACH Process.graphicalElement in Process.graphicalElements
        SELECT CASE Process.graphicalElement.Top
                CASE 0 to ((process.height / 3) * 1)
                        VerticalTop =+ VerticalTop
                CASE ((process.height / 3) * 1)  to ((process.height / 3) * 2)
                        VerticalCenter =+ VerticalCenter
                CASE ((process.height / 3) * 2)  to ((process.height / 3) * 3)
                        VerticalBottpm =+ VerticalBottom
        END SELECT
        IF: (VerticalTop <> VerticalBottom)
        THEN: AsymmetryAnomaly  /*VERTICAL_ASYMMETRY_ANOMALY*/
        ENDIF
NEXT
```

Kôd 175. Način prepoznavanja anomalije ***Vertikalne asimetrije***

Pravila prepoznavanja ***Vertikalne asimetrije***:

- u modelu poslovnog procesa broj elementa koji se nalaze u gornjem vertikalnom kvadrantu različit je od broja elemenata koji se nalaze u donjem vertikalnom kvadrantu

Na shematskom prikazu (Slika 118.) vidi se model koji je horizontalno simetričan, dakle lijeva i desna strana su u ravnoteži, dok ne postoji vertikalna simetrija. Ako se model podijeli na tri polja horizontalno i tri polja vertikalno, tada se dobije devet kvadrata. Ako se za svaki element modela procesa odredi u kojem kvadratu se nalazi, tada se može usporediti lijeva i desna strana, ako je broj elemenata s lijeve i desne strane isti, tada je model u simetriji, ako je broj elemenata različit, tada je model u debalansu odnosno asimetriji.

Način rješavanja anomalije - s obzirom da navedena anomalija ne utječe na funkcionalnost modela, korisnika valja porukom obavijestiti da je model u asimetriji.

No, ako se i u ovome slučaju primijeni pravilo da svaki element može biti udaljen od drugog elementa za svoju širinu, onda se i ova anomalija može riješiti na taj način.

```
FIX: AsymmetryAnomaly
Int HorizontalLeft=0
Int HorizontalRight=0
Int VerticalTop=0
Int VerticalBottom=0

FOR EACH Process.graphicalElement in Process.graphicalElements
       SELECT CASE Process.graphicalElement.left
              CASE 0 to ((process.width / 3) * 1)
                     HorizontalLeft =+ HorizontalLeft
              CASE ((process.width / 3) * 1)  to ((process.width / 3) * 2)
                     HorizontalCenter =+ HorizontalCenter
              CASE ((process.width / 3) * 2)  to ((process.width / 3) * 3)
                     HorizontalRight =+ HorizontalRight
       END SELECT
       IF: (HorizontalLeft <> HorizontalRight) /* HORIZONTAL_ASYMMETRY_ANOMALY */
       THEN:  Process.graphicalElement(n).left = Process.graphicalElement(n+1).left -
              (Process.graphicalElement(n).width * 2)
       ENDIF

       SELECT CASE Process.graphicalElement.Top
              CASE 0 to ((process.height / 3) * 1)
                     VerticalTop =+ VerticaTop
              CASE ((process.height / 3) * 1)  to ((process.height / 3) * 2)
                     VerticalCenter =+ VerticalCenter
              CASE ((process.height / 3) * 2)  to ((process.height / 3) * 3)
                     VerticalBottom =+ VerticalBottom
       End SELECT
       IF: (VerticalTop <> VerticalBottom) /* VERTICAL_ASYMMETRY_ANOMALY */
       THEN:    Process.graphicalElement(n).top = Process.graphicalElement(n+1).top -
           (Process.graphicalElement(n).height * 2)
       ENDIF
LOOP
```

Kôd 176. Način rješavanja ***Necentrirani i nesimetrični model***

3.5.2.2. Anomalija orijentacije

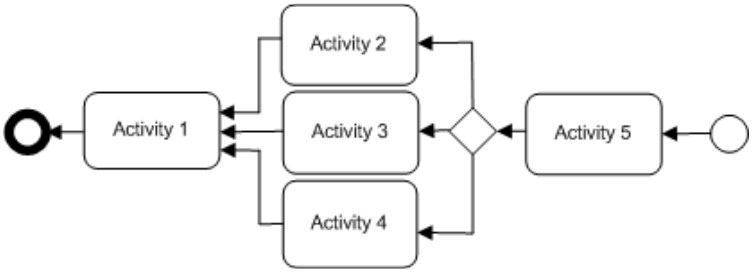
Anomalija orijentacije nastaje kad model ne zadovoljava načelo orijentacije. ***Vertikalne ili horizontalne orijentacije*** imaju veću vjerojatnost da budu uočene nego druge orijentacije. S obzirom da ljudi na zapadu prirodno čitaju s lijeva na desno i od vrha prema dolje i modele treba oblikovati da se tako čitaju. Nazivi elemenata trebaju biti horizontalni, jer se tako lakše čitaju. S obzirom na predloženu taksonomiju ***Anomalija orijentacije*** spada pod grupu ***Anomalija percepcije razlikovanja***, koja pad spada pod ***Anomalije prikaza modela***.

Modeli orijentirani s desna na lijevo

(eng. *Wrong horizontal orientation*)

Opis - zapadnjački način čitanja je s lijeva na desno, tako da izrada modela s desna na lijevo predstavlja prilično veliki propust koji utječe na razumijevanje modela. U zemljama u kojima se čita s desna na lijevo modeli bi morali biti obrnuto postavljeni, no u tom slučaju i nazivi elemenata moraju poštovati isto pravilo čitanja s desna na lijevo. S obzirom na predloženu taksonomiju anomalija u modelima poslovnih procesa anomalija ***Modeli orijentirani s desna***

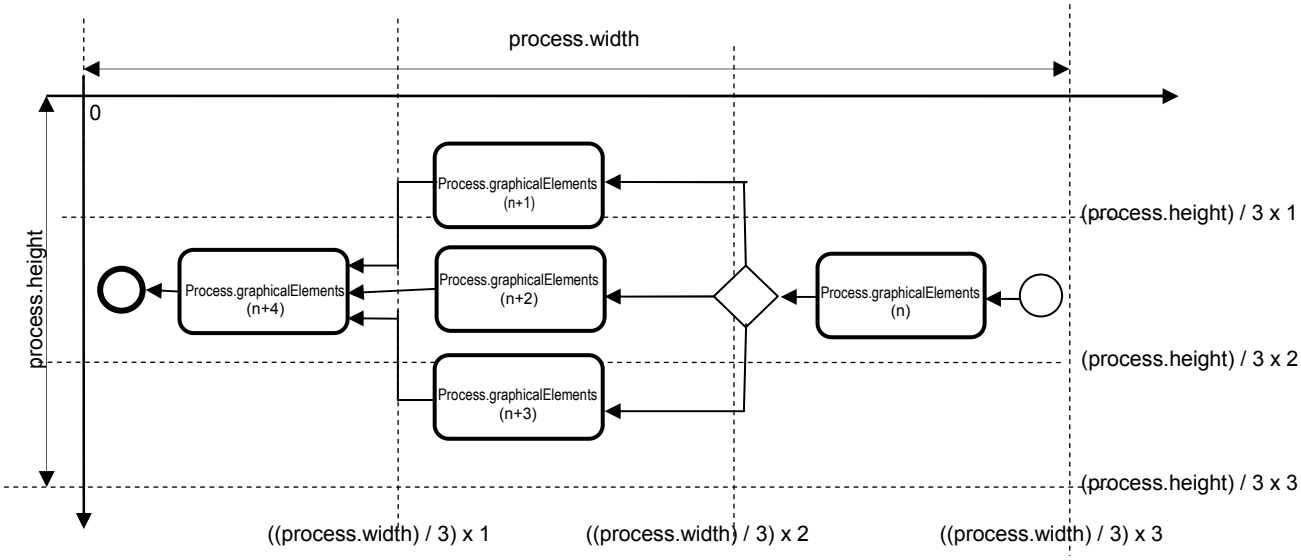
na lijevo nalaze se u grupi anomalija ***Anomalije orijentacije***, a koje su dio ***Anomalija prikaza modela***.



Slika 117. Anomalija ***Modeli orijentirani od desna na lijevo***

Način prepoznavanja anomalije – može se svesti na sljedeća pravila:

- model poslovnog procesa ima više od jednog elementa i
- u modelu poslovnog procesa postoji tok između dva grafička elementa gdje je slijedni grafički element lijevo od promatranog.



Slika 118. Shematski prikaz prepoznavanja anomalije ***Modeli orijentirani od desna na lijevo***

```
RULE: WrongHorizontalOrientation
IF: ((Control-Flow EXIST BETWEEN Process.graphicalElements) and
   Process.graphicalElement(n).left > Process.graphicalElement(n+1).left
THEN: WrongHorizontalOrientationAnomaly
ENDIF
```

Kôd 177. Način prepoznavanja anomalije ***Modeli orijentirani od desna na lijevo***

Način rješavanja anomalije - kad model sadrži anomaliju ***Model orijentiran od desna na lijevo***, tada je poželjno promijeniti pozicije elemenata na način da je pozicija pojedinog elementa udaljena za svoju dužinu od drugog elementa u smjeru ovisno o načinu čitanja. Prije

promjene pozicije elemenata treba korisnika obavijestiti da će se pozicije svih elemenata promijeniti u skladu s pravilima BPMN modeliranja.

```
FIX: WrongHorizontalOrientationAnomaly
IF: ((Control-Flow EXIST BETWEEN Process.graphicalElements) and
        Process.graphicalElement(n).left > Process.graphicalElement(n+1).left
THEN:
  FOR EACH Process.graphicalElement IN Process.graphicalElements
       IF: intCount = 1
       THEN:
       ELSE:
          Process.graphicalElement(n).Left = Process.graphicalElement(n-1).left +
          Process.graphicalElement(n-1).width
       ENDIF
  LOOP
ENDIF
```

Kôd 178. Način rješavanja anomalije ***Modeli orijentirani od desna na lijevo***

Modeli orijentirani od dolje prema gore

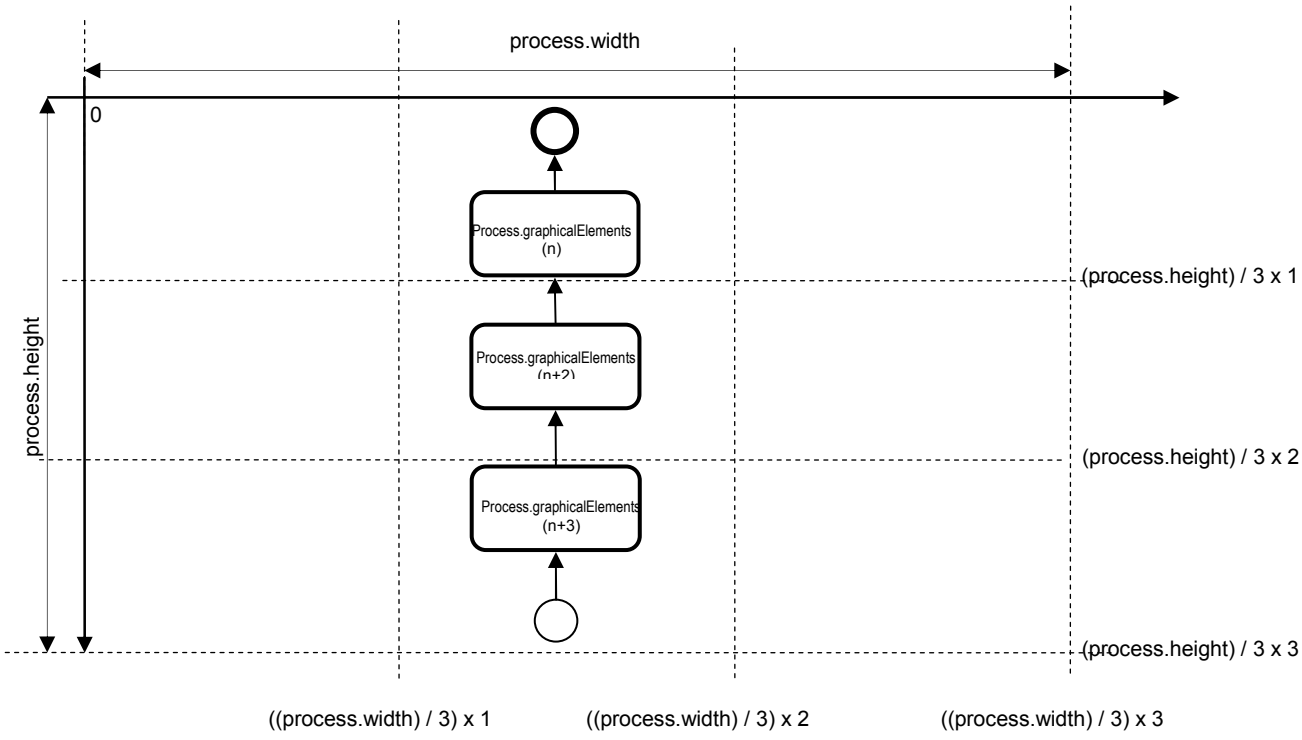
(eng. *Wrong vertical orientation*)

Opis - sve kulture čitaju od gora prema dolje, pa kreiranje modela čiji je tok od dna prema gore veliki propust za razumijevanje modela. Anomalija ***Modeli orijentirani od dolje prema gore*** spada u ***Anomalije orijentacije***, koje pak spadaju u ***Anomalije percepcije razlikovanja*** koje su dio ***Anomalija prikaza modela***.

Slika 119. Anomalija ***Modeli orijentirani od dolje prema gore***

Način prepoznavanja anomalije– temelji se na sljedećim pravilima:

- model poslovnog procesa sadrži više od jednog elementa i
- u modelu poslovnog procesa postoji element kod kojih postoji kontrola toka, a kod kojih je prethodni element niže od slijednog elementa.



Slika 120. Shematski prikaz anomalije ***Modeli orijentirani od dolje prema gore***

```
RULE: WrongVerticalOrientation
IF: ((Control-Flow EXIST BETWEEN Process.graphicalElement) and
    Process.graphicalElement(n).top > Process.graphicalElement(n+1).top
THEN: WrongVerticalOrientationAnomaly
ENDIF:
```

Kôd 179. Način prepoznavanja anomalije ***Modeli orijentirani od dolje prema gore***

Način rješavanja anomalije - u slučaju da model procesa sadrži anomaliju ***Modeli orijentirani od dolje prema gore***, onda elemente treba postaviti na način da je trenutni objekt viši od sljedećeg objekta barem za vrijednost svoje visine.

Prije promjene pozicije elemenata unutar modela procesa potrebno je obavijestiti korisnika da će se pozicije elemenata promijeniti u skladu s pravilima BPMN notacije.

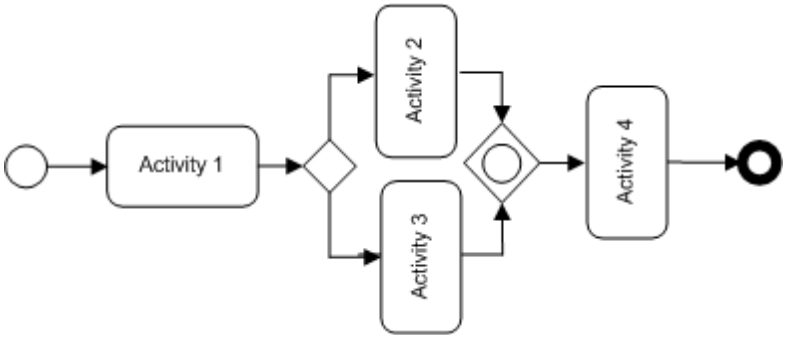
```
FIX: WrongVerticalOrientationAnomaly
IF: ((Control-Flow EXIST BETWEEN Process.graphicalElement) and
    Process.graphicalElement(n).top > Process.graphicalElement(n+1).top
THEN:
    Process.graphicalElement(n).top = Process.graphicalElement(n+1).top -
    (Process.graphicalElement(n).height * 2)
ENDIF:
```

Kôd 180. Način rješavanja anomalije ***Modeli orijentirani od dolje prema gore***

Vertikalni nazivi elemenata

(eng. *Vertical element titles*)

Opis – anomalija ***Vertikalni nazivi elemenata*** spada u **Anomalije orijentacije**, zatim i u kategoriju ***Anomalije percepcije razlikovanja*** koja spada pod ***Anomalije prikaza modela***. Treba uvijek težiti tome da nazivi aktivnosti i ostalih elemenata budu horizontalni.



Slika 121. Anomalija ***Vertikalni nazivi elemenata***

Način prepoznavanja anomalije – temelji se na sljedećem pravilu:

- u modelu poslovnog procesa postoji grafička kontrola iste kategorije koja ima vertikalno obilježje orijentacije naslova.

Ako se navedena pravila opišu pomoću pseudokoda, onda to izgleda ovako:

```
RULE: VerticalTitle
FOR EACH Process.graphicalElement in Process.graphicalElements
   FOR EACH Process.graphicalElement.Category in
            Process.graphicalElement.Categorys
      IF Process.graphicalElement.TitleOrientation <>
         CONSTANT.TITLE_HORIZONTAL
      THEN VerticalTitleAnomaly
      ENDIF
   NEXT
NEXT
```

Kôd 181. Način prepoznavanja anomalije ***Vertikalni nazivi elemenata***

Način rješavanja anomalije – ako postoji element iz iste kategorije iz skupa grafičkih elemenata koji ima svojstvo da mu je naziv elementa vertikalan, tada takav element treba postaviti na horizontalnu vrijednost.

Prije promjene orijentacije nazivlja pojedinih elemenata potrebno je korisnika obavijestiti da je u modelu pronađena navedena anomalija i da će ne orijentacija nazivlja promijeniti u skladu s pravilima BPMN notacije.

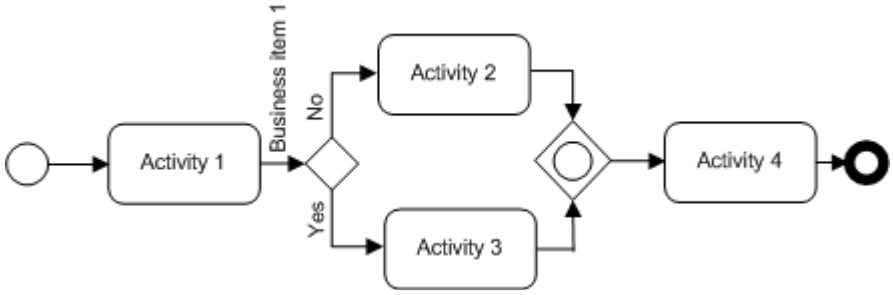
```
FIX: VerticalTitleAnomaly
FOR EACH Process.graphicalElement in Process.graphicalElements
   FOR EACH Process.graphicalElement.Category in
            Process.graphicalElement.Categorys
      IF: Process.graphicalElement.TitleOrientation <>
         CONSTANT.TITLE_HORIZONTAL
      THEN: Process.graphicalElement.TitleOrientation  =
         CONSTANT.TITLE_HORIZONTAL
      ENDIF
    NEXT
NEXT
```

Kôd 182. Način rješavanja anomalije ***Vertikalni nazivi elemenata***

Vertikalni opisi unutar modela

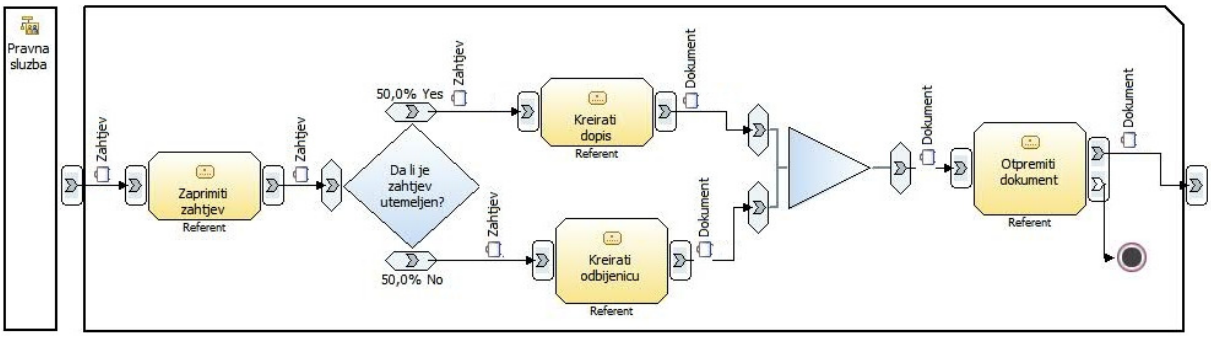
(eng. *Vertical elements description*)

Opis – anomalija ***Vertikalni opisi unutar modela*** pojavljuje se ako bilo koji grafički element ima opis vertikalno, umjesto horizontalno. To obično nakon dužeg rada dovodi do umora tijekom rada na modelima.



Slika 122. Anomalija ***Vertikalni opisi unutar modela***

Primjer – sljedeća slika prikazuje anomaliju ***Vertikalni opisi unutar modela*** u kojem poslovne stavke imaju vertikalne natpise. Rad na takvim modelima zahteva više vremena, jer umara, a i povećava mogućnost pogreške, zbog nepreglednosti natpisa.



Slika 123. Primjer anomalije ***Vertikalni opisi unutar modela***

Način prepoznavanja anomalije – sastoji se od sljedećeg pravila:

- u modelu poslovnog procesa postoji element koji ima vertikalni opis, tada takav model procesa ima anomaliju ***Vertikalni opis unutar modela***.

Navedeno pravilo opisano u pseudokodu, izgleda ovako:

```
1  RULE: VerticalDescription
2  FOR EACH Process.graphicalElement in Process.graphicalElements
3     IF: Process.graphicalElement.TitleOrientation <>
4        CONSTANT.TITLE_HORIZONTAL
5     THEN: VerticalDescriptionAnomaly
6     ENDIF
7  NEXT
```

Kôd 183. Način prepoznavanja anomalije ***Vertikalni opisi unutar modela***

Način rješavanja anomalije – ako model procesa ima anomaliju ***Vertikalni opis unutar modela***, onda se to rješava na način da se tom elementu natpis stavi na horizontalnu vrijednost.

Navedeno pravilo opisano u pseudokodu, izgleda ovako:

```
1  FIX: VerticalDescriptionAnomaly
2  FOR EACH Process.graphicalElement in Process.graphicalElements
3     IF: Process.graphicalElement.TitleOrientation <>
4         CONSTANT.TITLE_HORIZONTAL
5     THEN: Process.graphicalElement.TitleOrientation =
6         CONSTANT.TITLE_HORIZONTAL
7     ENDIF
8  NEXT
```

Kôd 184. Način rješavanja anomalije ***Vertikalni opisi unutar modela***

Prije promjene orijentacije natpisa u elementima modela poslovnog procesa, potrebno je korisnika obavijestiti o postojanju anomalije i o promjeni orijentacije nazivlja pojedinih elemenata.

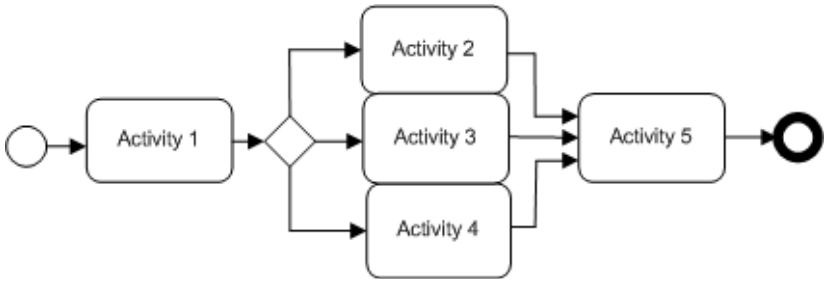
3.5.2.3. Anomalija obrisa (konture)

Modeli koji imaju ***Anomaliju obrisa*** vrlo često su nečitljivi, elementi takvih modela se međusobno dodiruju ili preklapaju što dovodi do otežanog čitanja modela. Prema predloženoj taksonomiji grupa ***Anomalije obrisa*** nalazi se u grupi ***Anomalije percepcije razlikovanja*** koje su naravno dio ***Anomalija prikaza modela.***

Elementi modela se međusobno dodiruju

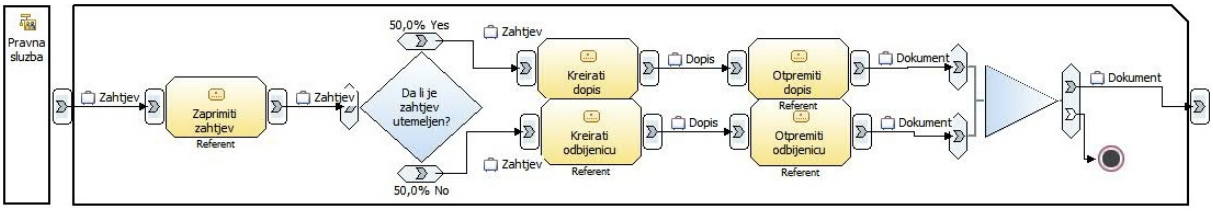
(eng. *Touching elements*)

Opis - anomalija ***Elementi modela se međusobno dodiruju*** nastaje kad je prekršen zakon obrisa (konture). Valja izbjegavati dodirivanje elemenata modela, jer se time gube obrisi elemenata čime se otežava čitanje modela. Sukladno tome i natpisi naziva pojedinih elemenata se ne bi smjeli dodirivati ni preklapati.



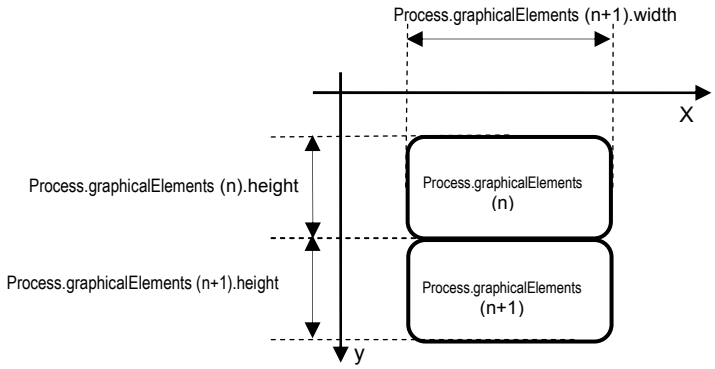
Slika 124. Anomalija obrisa (konture) – ***Elementi modela se međusobno dodiruju***

Primjer – na sljedećoj slici prikazan je primjer modela u kojem se pojedini elementi modela međusobno dodiruju. Takvi modeli mogu biti problematični za proučavanje, jer smanjuju preglednost elemenata.



Slika 125. Primjer anomalije ***Elementi modela se međusobno dodiruju***

Način prepoznavanja anomalije – sljedeća slika prikazuje situaciju u kojoj se elementi modela međusobno dodiruju.



Slika 126. Shematski prikaz anomalije ***Elementi modela se međusobno dodiruju***

Temeljno pravilo po kojem se mogu prepoznati da li je navedena anomalija u modelu je:

- model poslovnog procesa sadrži više od jednog elementa,
- u modelu poslovnog procesa postoji grafički element u skupu objekata grafičkih elemenata, koji bez obzira na kategoriju ima poziciju koja zbrojena sa dimenzijom kontrole i oduzeta od pozicije susjednog elementa zbrojene s dimenzijom kontrole daje iznos manju od širine grafičke kontrole i
- u modelu poslovnog procesa elementi se međusobno dodiruju i vertikalno i horizontalno.

Navedena pravila pretvorena u pseudokod bi izgledala ovako:

```
RULE: Touching
FOR EACH Process.graphicalElement in Process.graphicalElements
    intCount = +intCount
    IF intCount<>1
    THEN
       IF ((Process.graphicalElement(intCount-1).left +
           Process.graphicalElement(intCount-1).width) -
          (Process.graphicalElement(intCount).left +
           Process.graphicalElement(intCount).width)) <
           Process.graphicalElement(intCount).width)
       OR
          ((Process.graphicalElement(intCount-1).top +
            Process.graphicalElement(intCount-1).height) -
           (Process.graphicalElement(intCount).top +
            Process.graphicalElement(intCount).height)) <
            Process.graphicalElement(intCount).height)
       THEN TouchingAnomaly
       ENDIF
    ENDIF
NEXT
```

Kôd 185. Način prepoznavanja anomalije ***Elementi modela se međusobno dodiruju***

Način rješavanja anomalije - ako se prepozna postojanje anomalije obrisa tada je poželjno pozicionirati elemente na način da su međusobno udaljeni za jednu širinu, odnosno visinu elementa. Navedeno rješenje pretvoreno u pseudokod izgleda ovako:

```
FIX: TouchingAnomaly
FOR EACH Process.graphicalElement in Process.graphicalElements
    intCount = +intCount
    IF intCount<>1
    THEN
       IF ((Process.graphicalElement(intCount-1).Left +
           Process.graphicalElement(intCount-1).Width) -
          (Process.graphicalElement(intCount).Left +
           Process.graphicalElement(intCount).Width)) <
           Process.graphicalElement(intCount).Width)
        THEN
           Process.graphicalElement(intCount).Left =
           Process.graphicalElement(intCount-1).Left +
           (Process.graphicalElement(intCount-1).Width * 2)
        ENDIF
       IF ((Process.graphicalElement(intCount-1).Top +
            Process.graphicalElement(intCount-1).Height) -
           (Process.graphicalElement(intCount).Top +
            Process.graphicalElement(intCount).Height)) <
            Process.graphicalElement(intCount).Height)
       THEN
           Process.graphicalElement(intCount).Top =
           Process.graphicalElement(intCount-1).Top +
           (Process.graphicalElement(intCount-1).Height * 2)
       ENDIF
    ENDIF
NEXT
```

Kôd 186. Način rješavanja anomalije gdje se ***Elementi modela se međusobno dodiruju***

Ako se spoje skripte za prepoznavanje i rješavanje navedene anomalije onda to izgleda ovako:

```
FIX: TouchingAnomaly
FOR EACH Process.graphicalElement in Process.graphicalElements
    intCount = +intCount
    IF intCount<>1
       IF OverlapAnomaly = TRUE
       THEN
         Process.graphicalElement(intCount).Top =
         Process.graphicalElement(intCount-1).Top +
         (Process.graphicalElement(intCount-1).Height * 2)
       ENDIF
    ENDIF
NEXT
```

Kôd 187. Prepoznavanje i rješavanje anomalije ***Elementi modela se međusobno dodiruju***

Svakako prije promjene pozicije pojedinih elemenata u modelu poslovnog procesa potrebno je korisnika obavijestiti da će elementi u modelu procesa promijeniti svoje pozicije u skladu s pravilima BPMN notacije.

4. SMJERNICE AUTOMATIZACIJE PREPOZNAVANJA I RJEŠAVANJA ANOMALIJA U MODELIMA POSLOVNIH PROCESA

U prethodnom poglavlju izložena je taksonomija anomalija u modelu poslovnih procesa, osim toga svaka anomalija opisana je primjerom, načinom prepoznavanja i načinom rješavanja. Istraživanje u okviru ove disertacije sadržavalo je i razradu smjernica automatizacije i rješavanja anomalija u modelima poslovnih procesa. U ovome poglavlju opisati će se poredak prepoznavanja i rješavanja anomalija. Krajnji rezultat ovog poglavlja mora biti tablica pristupa rješavanja anomalija pomoću koje će se postaviti koncept na temelju kojih treba rješavati anomalije u modelima poslovnih procesa. Opisani postupak nije jedini, ali je provjeren i predlaže se zbog svoje jednostavnosti. S obzirom da je postavljen prijedlog taksonomije anomalija modela poslovnih procesa u kojoj postoji kategorizacija anomalija prema sličnim karakteristikama i obilježjima, logično je da se nameće pitanje da li nam navedeno grupiranje sličnih anomalija pomaže kod automatizacije prepoznavanja i rješavanja anomalija. Taksonomija anomalija modela poslovnih procesa podijeljena je na pet temeljnih kategorija i to: ***Sintaktičke anomalije***, ***Strukturne anomalije***, ***Anomalije resursa***, ***Semantičke anomalije*** i ***Anomalije prikaza modela***. Složenost pojedinih kategorija anomalija varira, stoga je logično da slijed njihovog prepoznavanja i rješavanja bude određen njihovim međusobnim vezama i karakteristikama. Poredak njihovog prepoznavanja i rješavanja mora slijediti njihove karakteristike i teći od jednostavnih prema složenijim. Prema navedenome poredak rješavanja trebao bi biti u sljedećem poretku:

1. anomalije prikaza modela,
2. sintaktičke anomalije,
3. semantičke anomalije,
4. strukturne anomalije i
5. anomalije resursa.

Anomalije prikaza modela su stavljene na prvo mjesto, jer se njihove karakteristike oslanjaju na vrijednosti visine, širine elemenata, te horizontalne i vertikalne pozicije, a te atribute svi grafički elementi i modeli imaju same po sebi, pa nije potrebna dodatna provjera postojanja tih vrijednosti. Otkrivene i riješene ***Anomalije prikaza modela*** ne utječu na daljnje rješavanje, jer su potpuno nezavisne. No, uspješno prepoznate i riješene ***Anomalije prikaza modela*** mogu bitno poboljšati preglednost modela, čime će se olakšati pronalaženje i rješavanje daljnjih anomalija u modelima poslovnih procesa.

Logika određivanja poretka rješavanja ***Anomalija prikaza modela*** je poredak čijim se izvođenjem najmanje narušava prethodni izgled modela s obzirom na prethodnu anomaliju. Dakle, cilj je da određena procedura rješavanja ne naruši ispravke prethodnih procedura za rješavanje anomalija. Sukladno tome prvo se provjeravaju ***Anomalije organizacije percepcije*** i zatim ***Anomalije percepcije razlikovanja***. Pronalaženje anomalija u modelu poslovnih procesa kreće od ***Anomalija strukturnih prikaza***, prvo se promatraju ***Anomalije organizacije percepcije*** i grupu anomalija ***Anomalije jednostavnosti***. Dakle, prvo se provjerava postojanje anomalije P***reviše elemenata u modelu***. Model s previše elemenata je nepregledan i teško se održava. Zatim se provjerava da li u modelu poslovnog procesa postoji anomalija ***Neselektivni prikaza elemenata u modelu***. U oba slučaja predloženi algoritam ne reducira broj elemenata nego se obavještava korisnika da model ima previše elemenata, te se predlaže podjela modela na više manjih modela. Nakon provjere postojanja ***Anomalija jednostavnosti*** slijedi provjera da li u modelu poslovnog procesa postoje anomalije iz grupe ***Anomalija grupiranja.*** Dakle provjerava se da li model ima anomaliju ***Nekompaktni i razvučen model***. Sljedeći korak je prepoznavanje i rješavanje ***Anomalija kontinuiranosti***, dakle rješavaju se grane koje se međusobno križaju i grane koje se savijaju. Nakon procedure za rješavanje anomalije ***Križanje i savijanja veza između elemenata modela***, potrebno je vidjeti da li procedura rješavanja te anomalije možda poremetili vizualnu hijerarhiju modela. Pa slijedi ispitivanje modela, da li u njemu postoji ***Anomalija nepoštivanja pozicije hijerarhije*** modela koja je sastavni dio grupe ***Anomalija prisnosti***. Slijedi rješavanje ***Anomalija sličnosti***, dakle provjerava se postojanje anomalija ***Slični elementi nisu grupirani zajedno***, zatim ***Slični elementi nemaju iste boje***, pa ***Slični elementi nemaju istu veličinu*** i na kraju ***Slični elementi nemaju isti oblik***. Nakon ***Anomalija sličnosti*** provjerava se postojanje ***Anomalije povezanosti***. Provjerava se da li u modelu poslovnog procesa postoji anomalija ***Preklapanje elementa modela***. Navedene anomalije bile su sastavni dio grupe ***Anomalija organizacije percepcije***. Slijedi provjera anomalija iz grupe ***Anomalije percepcije razlikovanja***. Pa grupe anomalija iz ***Anomalija obrisa*** i anomalije ***Elementi modela se međusobno dodiruju***. Nakon toga provjeravaju se ***Anomalije orijentacije*** u koje spadaju anomalije ***Modeli orijentirani s desna na lijevo***, ***Modeli orijentirani od dolje prema gore***, ***Vertikalni nazivi elemenata*** i ***Vertikalni opisi unutar modela***. Poslije ovih provjera provjerava se da li je model ima ***Anomaliju simetrije***, dakle provjerava se vertikalna i horizontalna simetrija tj da li u modelu postoji ***Necentrirani i nesimetrični model***. Kod ovog tipa anomalije provjerava se posebno horizontalna i posebno vertikalna anomalija simetrije.

Tabela 18. Redoslijed prepoznavanja i rješavanja anomalija i pripadajućih procedura ***Anomalija prikaza modela***

Red.br.	Naziv anomalije	Prepoznavanje anomalije	Rješavanje anomalije
	Anomalije organizacije percepcije		
	Anomalije jednostavnosti		
1.	Previše elemenata u modelu	`TooManyElements`	`TooManyElementsAnomaly()`
2.	Neselektivni prikaz elemenata u modelu	`UnselectiveElementsInModel`	`UnselectiverElementsInModelAnomaly`
	Anomalija grupiranja		
3.	Nekompaktni i razvučen model	`UncompactModel`	`UncompactModelAnomaly`
	Anomalije kontinuiranosti		
4.	Križanje i savijanje veza između elemenata modela	`Continuity`	`ContinuityAnomaly`
	Anomalije pristnosti		
5.	Nepoštivanje pozicije hijerarhije modela	`Familiarity`	`FamiliarityAnomaly`
	Anomalije sličnosti		
6.	Slični elementi nisu grupirani zajedno	`ElementSimilarity`	`ElementSimilarityAnomaly`
7.	Slični elementi nemaju iste boje	`SimilarElementsDiferentColor`	`SimilarElementsDiferentColorAnomaly`
8.	Slični elementi nemaju istu veličinu	`SimilarElementsDiferentSize`	`SimilarElementsDiferentSizeAnomaly`
9.	Slični elementi nemaju isti oblik	`SimilarElementsDiferentShape`	`SimilarElementsDiferentShapeAnomaly`
	Anomalije povezanosti		
10.	Preklapanje elemenata modela	`Overlap`	`OverlapAnomaly`
	Anomalije percepcije razlikovanja		
	Anomalije obrisa		
11.	Elementi modela se međusobno dodiruju	`Touching`	`TouchingAnomaly`
	Anomalije orijentacije		
12.	Modeli orijentirani od desna na lijevo	`WrongHorizontalOrientation`	`WrongHorizontalOrientationAnomaly`
13.	Modeli orijentirani od dolje prema gore	`WrongVerticalOrientation`	`WrongVerticalOrientationAnomaly`
14.	Vertikalni nazivi elemenata	`VerticalTitle`	`VerticalTitleAnomaly`
15.	Vertikalni opisi unutar modela	`VerticalDescription`	`VerticalDescriptionAnomaly`
	Anomalije simetrije		
16.	Necentrirani i nesimetrični model	`HorizontalAsymmetry` `VerticalAsymmetry`	`AsymmetryAnomaly`

Nakon ***Anomalija prikaza modela*** slijedi prepoznavanje i rješavanje ***Sintaktičkih anomalija***. Kod rješavanja ***Sintaktičnih anomalija*** logika rješavanja ide od anomalija ***Nepravilna uporaba spremnika***, preko anomalija ***Nepravilna uporabe objekata toka***, pa anomalija ***Nepravilna uporabe veza*** i završava anomalijama ***Nepravilne uporabe toka poruka***. Kao što je već rečeno prepoznavanje i rješavanje započinje anomalijama ***Nepravilne uporabe spremnika***, dakle prvo se provjerava anomalija ***Bazen bez elemenata*****,** zatim anomalija ***Nepravilna uporaba bazena.*** Takav poredak je logičan iz razloga što su bazeni elementi najniže razine, te manje ovise o drugim elementima. Slijede anomalije vezane uz staze unutar bazena. Pa slijedi prepoznavanje i rješavanje anomalija ***Staza bez elemenata*** i ***Nepravilna uporaba staza***. Nakon procedure prepoznavanja i rješavanja anomalija ***Nepravilne uporabe spremnika***, slijedi druga veća cjelina, a to su anomalije vezane uz ***Nepravilnu uporaba objekata toka.*** U prvome redu provjerava se da li postoji ***Više početnih događaja modelu.*** Nakon toga povjerava se anomalija ***Nepovezanost početnog događaja s ostalim elementima modela***, zatim ***Nepostojanje završnog događaja***, pa ***Nepovezanost završnog događaja s ostalim elementima modela***. Nakon anomalija ***Nepravilne uporabe objekata toka***, slijedi rješavanje anomalija ***Nepravilna uporabe veza***. Kod prepoznavanja ove grupe anomalija prvo se provjeravaju anomalije vezane uz ulazne/izlazne sekvence i događaje. Dakle, prvo se provjerava da li postoje anomalije ***Ulazne sekvence sa početnim događajem***, zatim se provjerava da li postoje anomalije ***Izlazna sekvenca sa završnim događajem***. Zatim se provjerava da li postoje anomalije ***Nepravilna uporabe toka poruka***. To su anomalije koje nastaju kad su elementi povezani s tokom poruke na način koji nije dozvoljen sintaksom BPMN 2.0 notacije. Kod te kategorije anomalija u prvom redu provjerava se da li je ***Početni događaj poruke vezan tokom poruke s elementom početnog događaja***, slijedi anomalija ***Početni događaj poruke vezan je tokom poruke s elementom bazena,*** pa anomalija ***Početni događaj poruke vezan je tokom poruke s elementom potprocesa.*** Zatim slijede ostale anomalije u navedenoj kategoriji anomalija ako što su: ***Početni događaj poruke vezan je tokom poruke s elementom aktivnosti***, ***Početni događaj vezan je tokom poruke s elementom stop događaja poruke***, ***Početni događaj vezan je tokom poruke s elementom završnog događaja poruke, Početni događaj poruke vezan je tokom poruke s završnim događajem***, ***Bazen je vezan tokom poruke s završnim događajem***, ***Potproces je vezan tokom poruke s završnim događajem***, ***Aktivnost je vezana tokom poruke s završnim događajem***, ***Stop događaj poruke je vezan tokom poruke s završnim događajem*** i na kraju anomalija ***Element završnog događaja poruke vezan je tokom poruke s elementom završnog događaja***.

Tabela 19. Redoslijed prepoznavanja i rješavanja anomalija i pripadajućih procedura ***Sintaktičkih anomalija***

Red.br.	Naziv anomalije	Prepoznavanje anomalije	Rješavanje anomalije
	Nepravilna uporaba spremnika		
1.	Bazen bez elemenata	PoolWithoutElements	PoolWithoutElementsAnomaly
2.	Nepravilna uporaba bazena	InvalidUseOfPool	InvalidUseOfPoolAnomaly
3.	Staza bez elemenata	SwimlineWithoutElements	SwimlineWithoutElementsAnomaly
4.	Nepravilna uporaba staza (eng: swimline)	InvalidUseOfSwimline	InvalidUseOfSwimlineAnomaly
	Nepravilna uporaba objekata toka		
5.	Više početnih događaja u modelu	MultipleStartEvents	MultipleStartEventsAnomaly
6.	Nepovezanost početnog događaja s ostalim elementima modela	StartEventHaveNotOutputSequence	StartEventHaveNotOutputSequenceAnomaly
7.	Početni događaj ima više od jedne izlazne sekvence	StartEventHaveMoreThanOneOutputSequence	StartEventHaveMoreThanOneOutputSequenceAnomaly
8.	Nepostojanje završnog događaja	ProcessDontHaveEndEvent	ProcessDontHaveEndEventAnomaly
9.	Nepovezanost završnog događaja s ostalim elementima modela	EndEventHaveNotInputSequence	EndEventHaveNotInputSequenceAnomaly
10.	Završni događaj ima više od jedne ulazne sekvence	EndEventHaveMoreThanOneInputSequence	EndEventHaveMoreThanOneInputSequenceAnomaly
11.	Nepostojanje početnog događaja	ProcessDontHaveStartEvent	ProcessDontHaveStartEventAnomaly
	Nepravilna uporaba veza		
12.	Ulazna sekvenca sa početnim događajem	StartEventHasInputSequence	StartEventHasInputSequenceAnomaly
13.	Izlazna sekvenca s završnim događajem	EvenEndtHasOutputSequence	EventEndHasOutputSequenceAnomaly
	Nepravilna uporaba toka poruka		
14.	Početni događaj poruke vezan je tokom poruke s elementom početnog događaja	EventStartConnectedToEventStart	EventStartConnectedToEventStartAnomaly
15.	Početni događaj poruke vezan je tokom poruke s elementom bazena	EventStartConnectedToPool	EventStartConnectedToPoolAnomaly
16.	Početni događaj poruke vezan je tokom poruke s elementom potprocesa	EventStartConnectedToSubProcess	EventStartConnectedToSubProcessAnomaly
17.	Početni događaj poruke vezan je tokom poruke s elementom aktivnosti	EventStartConnectedToActivity	EventStartConnectedToActivityAnomaly
18.	Početni događaj vezan je tokom poruke s elementom stop događaja poruke	EventStartConnectedToEventStop	EventStartConnectedToEventStopAnomaly
19.	Početni događaj vezan je tokom poruke s elementom završnog događaja poruke	EventStartConnectedToEventEnd	EventStartConnectedToEventEndAnomaly
20.	Početni događaj poruke vezan je tokom poruke s završnim događajem	EventMsgStartConnectedToEventEnd	EventMsgStartConnectedToEventEndAnomaly
21.	Bazen je vezan tokom poruke s završnim događajem	PoolConnectedToEventEnd	PoolConnectedToEventEndAnomaly
22.	Potproces je vezan tokom poruke s završnim događajem	SubProcessConnectedToEventEnd	SubProcessConnectedToEventEndAnomaly
23.	Aktivnost je vezana tokom poruke s završnim događajem	ActivityConnectedToEventEnd	ActivityConnectedToEventEndAnomaly
24.	Stop događaj poruke je vezan tokom poruke s završnim događajem	EventStopConnectedToEventEnd	EventStopConnectedToEventEndAnomaly
25.	Element završnog događaja poruke vezan je tokom poruke s elementom završnog događaja	EventSEndConnectedToEventEnd	EventSEndConnectedToEventEndAnomaly

Iz navedenog je vidljivo da prepoznavanje i rješavanje ***Sintaktičkih anomalija*** kreće od anomalija koje su manje zahtjevne prema anomalijama veza i toka koje su zahtjevnije za prepoznavanje i rješavanje, s obzirom da njihovo prepoznavanje i rješavanje uključuje više različitih tipova elemenata modela i njihove međusobne veze. Nakon ***Anomalija prikaza modela*** i ***Sintaktičkih anomalija*** slijedi prepoznavanje i rješavanje ***Semantičkih anomalija***. Postupci prepoznavanja i rješavanja ***Semantičkih anomalija*** su složeniji i zahtjevniji od prethodne dvije kategorije anomalija. U ovome tekstu dati će se samo okvir njihovog rješavanja koji se može proizvoljno detaljizirati i precizirati ugradnjom neovisnih algoritama za prepoznavanje pojedinih ***Semantičkih anomalija***. Kod procesa prepoznavanja i rješavanja ***Semantičkih anomalija***, proces počinje razrješavanjem kategorije anomalija pod nazivom ***Nepravilno nazivlje elemenata*** i zatim kategorijom ***Nepravilne vrijednosti atributa***. U kategoriji ***Nepravilno nazivlje elemenata*** prvo se provjerava anomalija ***Neimenovanje elemenata modela***, zatim ***Nejedinstvenost nazivlja elemenata modela***, pa ***Nekonzistentnost nazivlja elemenata modela***. Navedena provjera ispituje, da li su nazivi u obliku kojem moraju biti sukladno najboljoj praksi, da li nazivi baze podataka imaju imenice, da li nazivi aktivnosti imaju glagole, da li su nazivi u infinitivu, itd. Nakon ove provjere slijedi provjera da li u nazivima elemenata postoje sinonimi i homonimi. Dakle, provjera da li posoje anomalije ***Sinonimi u nazivlju elemenata modela*** i ***Homonimi u nazivlju elemenata modela***. Odnosno, da li među nazivima aktivnosti i drugih objekata postoje prikrivene dvosmislenosti i nejasnoće prouzrokovane nazivima elemenata. Nakon kategorije anomalija pod nazivom ***Nepravilno nazivlje elemenata***, proširuje se skup provjera i na drugu kategoriju anomalija pod nazivom ***Nepravilne vrijednosti atributa***. Prva anomalija koja se prepoznaje i rješava iz navedene kategorije je anomalija ***Ne postoji vrijednost atributa***, slijedi anomalija ***Nejedinstvenost vrijednosti atributa***. Nakon toga slijedi prepoznavanje i rješavanje anomalija ***Neispravan tip podataka vrijednosti atributa*** i ***Nedozvoljeni znakovi u vrijednosti atributa***. Iz navedenog, vidljivo je da je i u ovome slučaju redoslijed prepoznavanja i rješavanja anomalija definiran od lakših postupaka prema težim i kompliciranijim. Druga kategorija provjere ***Semantičkih anomalija*** je zapravo proširenje provjere s naziva elemenata na druge atribute elemenata. Na taj način obuhvaćeni su svi elementi u modelu. U znanstvenoj literaturi postoji cijeli niz metoda, metodika i algoritama za prepoznavanje i rješavanje takvih pojava u tekstovima, no s ciljem pojednostavljenja, ovaj rad u biti opisuje samo najjednostavnije mehanizme za rješavanje tih anomalija koje se lako mogu proširiti daleko preciznijim, a time i daleko složenijim algoritmima.

Tabela 20. Redoslijed prepoznavanja i rješavanja ***Semantičkih anomalija***

Red.br.	Naziv anomalije	Prepoznavanje anomalije	Rješavanje anomalije
	Nepravilno nazivlje elemenata		
1.	Neimenovanje elemenata modela	`UnNamedElements`	`UnNamedElementsAnomaly`
2.	Nejedinstvenost nazivlja elemenata modela	`NonUniqueName`	`NonUniqueNameAnomaly`

3.	Nekonzistentnost nazivlja elemenata modela	`InconsistencyInElementNames`	`ProcessSubProcessNominativeAnomaly`
			`NoVerbsInActivityAnomaly`
			`TooManyVerbsActivityAnomaly`
			`NotInNominativeActivityAnomaly`
			`NameIsEmptyActivityAnomaly`
			`AnomalyNameIsEmptyDatabase`
			`NoNounsInDatabaseAnomaly`
			`TooManyNounsInDatabaseAnomaly`
			`AnomalyGatewayCondition`

4.	Sinonimi u nazivlju elemenata modela	`FindSynonymeInModel`	`SynonymeAnomaly`
5.	Homonimi u nazivlju elemenata modela	`FindHomonymeInModel`	`HomonymesInModelAnomaly`
	Nepravilne vrijednosti atributa		
6.	Ne postoji vrijednost atributa	`EmptyAttributeValue`	`EmptyAttributeValueAnomaly`
7.	Nejedinstvenost vrijednosti atributa	`NonUniqueAttributeValue`	`NonUniqueAttributeValueAnomaly`
8.	Neispravan tip podataka vrijednosti atributa	`WrongAttributeDataTypeValue`	`WrongAttributeDataTypeValueAnomaly`
9.	Nedozvoljeni znakovi u vrijednosti atributa	`SpecialCharactersInAttribute`	`AttributeSpecialCharactersAnomaly`

Nakon postupka prepoznavanja i rješavanja ***Semantičkih anomalija*** slijede procedure prepoznavanja i rješavanja još složenijih tipova anomalija, radi se o ***Strukturnim anomalijama***.

Strukturne anomalije su kompleksnije za prepoznavanje i rješavanje od ***Anomalija prikaza modela***, ***Sintaktičkih*** i ***Semantičkih anomalija***. ***Strukturne anomalije*** zahtijevaju poznavanje hijerarhije, odnosa i međuzavisnosti među elementima. Sintaksa i semantika modela treba biti provjerena prije provjere ***Strukturnih anomalija***, dakle elementi moraju biti povezani u skladu sa sintaksom modela, nazivlje i atributi pojedinih elemenata moraju biti postavljeni u skladu s semantičkim pravilima modela, itd. Strukturne anomalije sastoje se o dvije temeljne kategorije anomalija i to: ***Anomalije kontrola toka*** i ***Anomalije pravila***. Kategorija ***Anomalije kontrola*** toka sastoji se od nekoliko potkategorija kao što su: ***Redundancija kontrole toka*** u koje spadaju anomalije: ***Nepotrebna kontrola toka*** i ***Redundancija modela*** i one se prve provjeravaju i rješavaju**.** Nakon navedene kategorije provjerava se kategorija anomalija ***Nedefinirani uvjeti vrata*** u koje spadaju anomalije: ***Nedefinirani uvjeti vrata***, ***Kontradiktorni uvjeti vrata***, ***Deterministički nedostatak sinkronizacije*** i ***Nedeterministički nedostatak sinkronizacije***. Slijedi prepoznavanje i rješavanje kategorije anomalija ***Specijalni slučajevi toka*** u koje spadaju anomalije: ***Viseća stanja***, ***Netolerancija na promjene poretka***, ***Nekonzistentnost kod inserta paralelnih grana***, ***Nekonzistentnost promjene iz prošlosti*** i ***Nedostatak tolerancije u petlji***. Na kraju kategorije ***Kontrola toka*** slijedi kategorija anomalija ***Narušavanje ispravnosti*** u koje spadaju anomalije: ***Zapinjanje toka*** i ***Zastoj toka***. Nakon provjere kategorije anomalija ***Kontrola toka*** slijedi kategorija ***Anomalije pravila*** koja se sastoji od sljedećih kategorija anomalija: ***Anomalije konzistentnosti***, ***Obuhvat pravila*** i ***Tok podataka***. Tim slijedom izvode se i radnje prepoznavanja i rješavanja anomalija. Dakle, prvo se izvodi provjera za anomaliju ***Redundantna pravila***, a zatim i za anomaliju ***Generalna pravila***. Obje anomalije dio su kategorije anomalija ***Redundantna pravila*** koja je podgrupa od kategorije ***Anomalija konzistentnosti***. Unutar kategorije ***Anomalija konzistentnosti*** slijedi kategorija ***Konfliktna pravila*** koja se pak provjerava slijedom: ***Izravna kontradikcija***, ***Kontradikcija u ulazu*** i ***Kontradikcija u zaključku.*** Nakon kategorije ***Konfliktna pravila*** slijedi provjera anomalija ***Petlji*** u koje spada: ***Zapinjanje temeljeno na pravilima*** i ***Zastoj temeljen na pravilima.*** Sljedeća velika kategorija anomalija u ***Anomalijama pravila*** je kategorija ***Obuhvat pravila*** u koje spadaju anomalije koje se izvode sljedećim redoslijedom: ***Izolirana pravila***, ***Beskorisni zaključci (Beskorisna pravila)***, ***Viseći uvjeti*** i ***Nepostojeća pravila***. Nakon kategorije

Obuhvat podataka u kategoriji ***Anomalije pravila*** slijedi još jedna, zadnja kategorija anomalija i to ***Anomalije toka podataka***. Navedena kategorija anomalija bavi se problemima podataka u modelu poslovnih procesa. Redoslijed prepoznavanja i rješavanja ***Anomalija toka podataka*** je: ***Redundantni podaci***, ***Izgubljeni podaci***, ***Nepostojeći podaci***, ***Nekonzistentni podaci***, ***Nikad izbrisani podaci***, ***Dva puta izbrisani podaci*** i ***Podaci neizbrisani na vrijeme.*** Iz navedenog je vidljivo da je i u ovome slučaju redoslijed prepoznavanja i rješavanja anomalija u smjeru od jednostavnijih anomalija prema složenijim. Dakle od anomalija ***Redundancija modela*** preko ***Nedefiniranih uvjeta vrata***, pa ***Specijalnih slučajeva toka***, do ***Anomalija pravila*** gdje je sve završilo anomalijama ***Toka podataka***. Navedene anomalije su anomalije više razine i zahtjevnije za prepoznavanje. Do sad promatrane anomalije su statičkog karaktera, drugim riječima takve anomalije moguće je uočiti tijekom modeliranja i promatranja modela, dakle model ne mora proći kroz testove simulacije da bi se takve anomalije uočile. No, sljedeći tip anomalija, a to su ***Anomalije resursa***, predstavljaju većinom dinamičke anomalije, no u ovome radu obratiti će se pažnja samo na statički set anomalija resursa. Prepoznavanje i rješavanje ***Anomalija resursa*** ne ovise o rezultatima faze simulacije i testiranja već se mogu uočiti tijekom modeliranja tj. kod dodjele resursa pojedinoj aktivnosti ili tijekom statičke analize modela poslovnog procesa. ***Anomalije resursa*** su najviši oblik anomalija koje će se u ovome radu opisati, a da bi se započelo njihovo prepoznavanje i rješavanje moraju biti prepoznate i provjerene ***Anomalije prikaza modela***, ***Sintaktičke anomalije***, ***Semantičke anomalije*** i ***Strukturne anomalije***. ***Anomalije resursa*** sastoje se od tri kategorije anomalija i to ***Anomalije obrazaca kreiranja***, ***Anomalije samopokretajućih obrazaca*** i ***Anomalije višestrukih obrazaca resursa***. Prepoznavanje i rješavanje ***Anomalija resursa*** počinje kategorijom ***Anomalija obrazaca kreiranja*** u kojoj se provjeravaju anomalije: ***Direktna alokacija u slučaju da više resursa ima isti naziv***, ***Direktna alokacija resursa bez mehanizma eskalacije***, ***Alokacija temeljena na ulogama kad odabrani resurs nije pripadnik određene uloge/role***, ***Alokacija temeljena na ulogama kad resurs pripada u više od jedne grupe***, ***Alokacija temeljena na ulogama kod grupe sa jednim resursom*** i ***Automatsko izvođenje bez mehanizma provjere izvođenja***. Nakon kategorije ***Anomalija obrazaca kreiranja*** slijedi kategorija ***Anomalije samopokretajućih obrazaca***. U tu kategoriju spadaju anomalije: ***Početak kreiranja sa alokacijom resursa*** i ***Djelomično lančano izvršenje u poslovnom procesu***. A nakon kategorije ***Anomalije samopokretajućih obrazaca*** slijedi kategorija ***Anomalije višestrukih obrazaca resursa*** u koju spada anomalija: ***Istovremeno izvođenje kod ograničenih kapaciteta resursa***.

Tabela 21. Redoslijed prepoznavanja i rješavanja ***Strukturnih anomalija***

Poredak	Naziv anomalije	Prepoznavanje anomalije	Rješavanje anomalije
	Kontrola toka		
1.	Nepotrebna kontrola toka	`UnnecessaryControlFlow`	`UnnecessaryControlFlowAnomaly`
2.	Redundancija modela	`ModelingRedundancy`	`ModelingRedundancyAnomaly`
3.	Nedefinirani uvjeti vrata	`UndefinedGatewayConditions`	`UndefinedGatewayConditionsAnomaly`
4.	Kontradiktorni uvjeti vrata	`ContradictiveGetewayConditions`	`ContradictiveGatewayConditionsAnomaly`
5.	Nedostatak sinkronizacije (eng. Lack of synchronization)	`LackOfSynchronization`	`LackOfSynchronizationAnomaly`
6.	Deterministički nedostatak sinkronizacije	`DeterministicLackOfSynchronization`	`DeterministicLackOfSynchronizationAnomaly`
7.	Nedeterministički nedostatak sinkronizacije	`UnDeterministicLackOfSynchronization`	`UnDeterministicLackOfSynchronizationAnomaly`
8.	Viseća stanja	`DanglingInputAndOutput`	`DanglingInputAndOutputAnomaly`
9.	Netolerancija na promjene poretka	`NoToleranceOnChangeOfOrder`	`NoToleranceOnChangeOfOrderAnomaly`
10.	Nekonzistentnost kod inserta paralelnih grana	`InconsistenclesWhenInsertingParallelBraches`	`InconsistenclesWhenInsertingParallelBrachesAnomaly`
11.	Nekonzistentnost promjene iz prošlosti	`InconsistentChangeOfPast`	`InconsistentChangeOfPastAnomaly`
12.	Nedostatak tolerancije u petlji	`MissingLoopTolerance`	`MissingLoopToleranceAnomaly`
13.	Zapinjanje toka	`LivelockFlow`	`LivelockFlowAnomaly`
14.	Zastoj toka	`DeadlockFlow`	`DeadlockFlowAnomaly`
	Anomalije pravila		
15.	Redundantna pravila	`RedundantRules`	`RedundantRulesAnomaly`
16.	Generalna pravila	`SubSubsumedRules`	`SubSubsumedRulesAnomaly`
17.	Izravna kontradikcija (eng. Direct contradiction)	`DirectContradiction`	`DirectContradictionAnomaly`
18.	Kontradikcija u ulazu	`ContradictionInInput`	`ContradictionInInputAnomaly`
19.	Kontradikcija u zaključku	`ContradictionInConclusion`	`ContradictionInConclusionAnomaly`
20.	Zapinjanje temeljeno na pravilima	`LivelockByRules`	`LivelockByRulesAnomaly`
21.	Zastoj temeljen na pravilima	`RuleBasedDeadlock`	`RuleBasedDeadlockAnomaly`
22.	Izolirana pravila (eng. Isolated rules)	`IsolatedRules`	`IsolatedRulesAnomaly`
23.	Beskorisni zaključci (eng. Useless conclusions)	`UselessConclusion`	`UselessConclusionsAnomaly`
24.	Viseći uvjeti (eng. Dangling Conditions)	`DanglingConditions`	`DanglingConnectionsAnomaly`
25.	Nepostojeća pravila	`MissingRules`	`MissingRulesAnomaly`
26.	Redundantni podaci	`DataRedundancy`	`DataRedundancyAnomaly`
27.	Izgubljeni podaci	`LostData`	`LostDataAnomaly`
28.	Nepostojeći podaci	`MissingData`	`MissingDataAnomaly`
29.	Nekonzistentni podaci	`InconsistentData`	`InconsistentDataAnomaly`
30.	Nikad izbrisani podaci	`NeverDestroyedData`	`NeverDestroyedDataAnomaly`
31.	Dva puta izbrisani podaci	`LostData`	`LostDataAnomaly`
32.	Podaci neizbrisani na vrijeme	`DataNotDeletedOnTime`	`DataNotDeletedOnTimeAnomaly`

Tabela 22. Redoslijed prepoznavanja i rješavanja anomalija i pripadajućih procedura ***Anomalija resursa***

Red.br.	Naziv anomalije	Prepoznavanje anomalije	Rješavanje anomalije
	Anomalije obrazaca kreiranja		
1.	Direktna alokacija u slučaju da više resursa ima isti naziv	`ResourceNonUniqueName`	`ResourceNonUniqueNameAnomaly`
2.	Direktna alokacija resursa bez mehanizma eskalacije	`DirectAlocationWithoutEscalation`	`DirectAlocationWithoutEscalationAnomaly`
3.	Alokacija temeljena na ulogama kad odabrani resurs nije pripadnik određene uloge/role	`roleBasedAllocationResourceNotInRole`	`roleBasedAllocationResourceNotInRoleAnomaly`
4.	Alokacija temeljena na ulogama kad resurs pripada u više od jedne grupe	`roleBasedAllocationResourceInManyGroups`	`roleBasedAllocationResourceNotInGroupsAnomaly`
5.	Alokacija temeljena na ulogama kod grupe sa jednim resursom	`roleBasedAllocationRoleWithSingleResource`	`roleBasedAllocationRoleWithSingleResourceAnomaly`
6.	Automatsko izvođenje bez mehanizma provjere izvođenja	**`GetStatus(Activity)`**	**`GetStatus(Activity)`**
	Anomalije samopokretajućih obrazaca		
7.	Početak kreiranja sa alokacijom resursa	`CreationWithResourceAllocation`	`CreationWithResourceAllocationAnomaly`
8.	Djelomično lančano izvršenje u poslovnom procesu	`PartlyChainedExecution`	`PartlyChainedExecutionAnomaly`
	3.3.3. Anomalije višestrukih obrazaca resursa		
9.	Istovremeno izvođenje kod ograničenih kapaciteta resursa	`SimultaneousExecutionLimitedCapacity`	`SimultaneousExecutionLimitedCapacityAnomaly`

5. PRIMJERI PREPOZNAVANJA I RJEŠAVANJA ANOMALIJA NA REFERENTNOM SKUPU MODELA POSLOVNIH PROCESA

U prethodnim poglavljima predstavljeni su obrasci anomalija kategorizirani po zajedničkim svojstvima i karakteristikama, nakon toga prikazane su smjernice automatizacije i rješavanja anomalija u modelima poslovnih procesa. U ovome poglavlju će se prethodno opisani detalji primijeniti na konkretnim primjerima. Cilj ovog poglavlja je provjeriti opisani postupak prepoznavanja i rješavanja anomalija na konkretnim primjerima. Time će se obuhvatiti svi elementi prepoznavanja i rješavanja anomalija, od definiranih obrazaca sa pravilima prepoznavanja i rješavanja, do redosljeda prepoznavanja i rješavanja pojedinih anomalija. Ovo poglavlje doprinosi razradi postupka prepoznavanja i rješavanja anomalija u prvom redu kao ogledni primjer uporabe postavljenih koncepata na modelima poslovnih procesa uzetih iz postojećih realnih okolina. Namjena obrađenih obrazaca je njihova uporaba u razvoju alata za modeliranje BPMN 2.0. modela poslovnih procesa. No, s obzirom da je ovaj rad ograničen na razvoj obrazaca anomalija i s druge strane određivanje ispravnog redoslijeda prepoznavanja i rješavanja anomalija, za provjeru ispravnosti procedure koristit će se upitnik s pravilima prepoznavanja, da li određena anomalija postoji ili ne. Za svaku anomaliju izvedena su pravila prepoznavanja anomalija. Ta pravila prepoznavanja oblikovana su na način, da ako se na sva pravila po pojedinoj anomaliji odgovori potvrdno, onda model ima navedenu anomaliju. Na primjer, za anomaliju ***Nepostojanje početnog događaja*** ako u modelu ne postoji ulazna sekvenca i ako je broj početnih događaja jednak nuli onda se sa sigurnošću može reći da u modelu postoji anomalija ***Nepostojanje početnog događaja***.

Tabela 23. Primjer obrasca anomalije ***Nepostojanje početnog događaja*** s pravilima prepoznavanja.

Naziv anomalije	Pravila prepoznavanja	Da	Ne
Nepostojanje početnog događaja	U modelu poslovnog procesa ne postoji ulazna sekvenca.		
	U modelu poslovnog procesa broj početnih događaja jednak je nuli.		

Kad se u tablicu stave sve grupe anomalija, anomalije i pravila njihovog prepoznavanja, dobije se dosta velika tabela, pa je nužno tu tabelu podijeliti na više manjih radi lakše obrade i manipulacije. U ovome slučaju svaka kategorija anomalija ima svoju tabelu, dakle tabelu ***Anomalije prikaza modela***, tabelu ***Sintaktičke anomalije***, tabelu ***Semantičkih anomalija***, tabelu ***Strukturnih anomalija*** i tabelu ***Anomalija resursa***. Tabela sadrži dva stupca **Da** i **Ne** u koje se stavlja oznaka, ovisno o tome da li je pravilo istinito ili neistinito za promatrani model.

Tabela 24. Tabela sa redoslijedom i pravilima prepoznavanja ***Anomalija prikaza modela***

Red.br.	Naziv anomalije	Pravila prepoznavanja	Da	Ne
	Anomalije organizacije percepcije			
	Anomalije jednostavnosti			
1.	Previše elemenata u modelu	Model poslovnog procesa sadrži ukupni broj elemenata veći od 9.		
2.	Neselektivni prikaz elemenata u modelu	Model poslovnog procesa sadrži aktivnosti koje ne spadaju u isti kontekst procesa.		
		Model poslovnog procesa prikazuje sadržaje razmjene koji nisu bitni za prikaz modela poslovnog procesa.		
		Model poslovnog procesa koristi resurse koji nisu bitni za prikaz modela poslovnog procesa.		
	Anomalije grupiranja			
3.	Nekompaktni i razvučen model	Model poslovnog procesa ima više od jednog elementa.		
		Model sadrži grafički element kategorije koji se horizontalno ili vertikalno nalazi između dva spojena elementa iste kategorije.		
		Model sadrži uvjet koji govori o tome da se neki element nalazi previše udaljen od drugog elementa.		
	Anomalije kontinuiranosti			
4.	Križanje i savijanje veza između elemenata modela	Model poslovnog procesa ima više od jednog elementa.		
		U modelu poslovnog procesa povezani elementi iste kategorije nisu na istoj visini.		
		Element kontrole toka siječe se sa drugim elementom kontrole toka.		
	Anomalije prisnosti			
5.	Nepoštivanje pozicije hijerarhije modela	Model poslovnog procesa ima više od jednog elemenata iste kategorije.		
		U modelu poslovnog procesa postoji element koji je spojen s drugim elementom iste kategorije, a jedan je iznad drugoga i u istoj mjeri udaljen od ruba modela.		
	Anomalije sličnosti			
6.	Slični elementi nisu grupirani zajedno	Model procesa ima više od jednog elementa.		
		U modelu poslovnog procesa postoje elementi čije kategorija objekta ima obilježje sličnog objekta.		
		U modelu poslovnog procesa broj sličnih objekata u modelu procesa je veći od 1.		
7.	Slični elementi nemaju iste boje	Model poslovnog procesa ima više od jednog elementa.		
		U modelu poslovnog procesa postoje grafički elementi iste kategorije koji imaju različite boje.		
8.	Slični elementi nemaju istu veličinu	Model poslovnog procesa ima više od jednog elementa.		
		U modelu poslovnog postoji element iste kategorije, a da li mu je širina i visina dvostruko veća od prethodnog elementa.		
9.	Slični elementi nemaju isti oblik	U modelu poslovnog procesa ima više od jednog elementa.		
		U modelu poslovnog postoji element iste kategorije, a da li mu je širina ili visina dvostruko veća od prethodnog elementa.		
	Anomalije povezanosti			
10.	Preklapanje elemenata modela	Model poslovnog procesa ima više od jednog elementa.		
		U modelu poslovnog procesa postoji element čija je lijeva pozicije i širina manja od lijeve pozicije i širine sljedećeg elementa.		
		U modelu poslovnog procesa postoji element čija je gornja pozicija i visina manja od gornje pozicije i visine sljedećeg elementa.		
	Anomalije percepcije razlikovanja			
	Anomalije obrisa			
11.	Elementi modela se međusobno dodiruju	Model poslovnog procesa sadrži više od jednog elementa.		
		U modelu poslovnog procesa postoji grafički element u skupu objekata grafičkih elemenata, koji bez obzira na kategoriju ima poziciju koja zbrojena sa dimenzijom kontrole i oduzeta od pozicije susjednog elementa zbrojene s dimenzijom kontrole daje iznos manju od širine grafičke kontrole.		
		U modelu poslovnog procesa elementi se međusobno dodiruju i vertikalno i horizontalno.		
	Anomalije orijentacije			
12.	Modeli orijentirani s desna na lijevo	Model poslovnog procesa ima više od jednog elementa		
		U modelu poslovnog procesa postoji tok između dva grafička elementa gdje je slijedni grafički element lijevo od promatranog		
13.	Modeli orijentirani od dolje prema gore	Model poslovnog procesa sadrži više od jednog elementa,		
		U modelu poslovnog procesa postoji element kod kojih postoji kontrola toka, a kod kojih je prethodni element niže od slijednog elementa.		
14.	Vertikalni nazivi elemenata	U modelu poslovnog procesa postoji grafička kontrola iste kategorije koja ima obilježje orijentacija naslova a da je vertikalna.		
15.	Vertikalni opisi unutar modela	U modelu poslovnog procesa postoji element koji ima vertikalni opis, tada takav model procesa ima anomaliju vertikalni opis unutar modela.		
	Anomalije simetrije			
16.	Necentrirani i nesimetrični model	U modelu poslovnog procesa broj elementa koji se nalaze u lijevom horizontalnom kvadrantu različit je od broj elemenata koji se nalaze u desnom horizontalnom kvadrantu.		
		U modelu poslovnog procesa broj elementa koji se nalaze u gornjem vertikalnom kvadrantu različit je od broj elemenata koji se nalaze u donjem vertikalnom ...		

Tabela 25. Tabela sa redoslijedom i pravilima prepoznavanja ***Sintaktičkih anomalija***

Red.br.	Naziv anomalije	Pravila prepoznavanja	Da	Ne
	Nepravilna uporaba spremnika			
1.	Bazen bez elemenata	U modelu poslovnog procesa postoji bazen u kojem ne postoji ni jedan grafički element.		
2.	Nepravilna uporaba bazena	U modelu poslovnog procesa postoji bazen ima tok sekvence koji prelazi granice spremnika.		
3.	Staza bez elemenata	U modelu poslovnog procesa postoji staza bez elemenata.		
4.	Nepravilna uporaba staza	U modelu poslovnog procesa postoji staza no ne postoji i bazen koji sadrži navedenu stazu.		
	Nepravilna uporaba objekata toka			
5.	Više početnih događaja u modelu	U modelu poslovnog procesa postoji više od jednog početnog događaja.		
6.	Nepovezanost početnog događaja s ostalim elementima modela	U modelu poslovnog procesa početni događaj nema izlaznu sekvencu.		
7.	Početni događaj ima više od jedne izlazne sekvence	U modelu poslovnog procesa mora postojati barem jedan početni događaj.		
		Početni događaj ima više od jedne izlazne sekvence.		
8.	Nepostojanje završnog događaja	U modelu poslovnog procesa ne postoji završni događaj.		
9.	Nepovezanost završnog događaja s ostalim elementima modela	U modelu poslovnog procesa ulazna sekvenca završnog događaja je prazan skup.		
10.	Završni događaj ima više od jedne ulazne sekvence	U modelu poslovnog procesa mora biti barem jedan završni događaj.		
		Završni događaj ne smije imati više od jedne ulazne sekvence.		
11.	Nepostojanje početnog događaja	U modelu poslovnog procesa ne postoji ulazna sekvenca.		
		U modelu poslovnog procesa broj početnih događaja jednak je nuli.		
	Nepravilna uporaba veza			
12.	Ulazna sekvenca sa početnim događajem	U modelu poslovnog procesa početni događaj ima ulaznu sekvencu.		
13.	Izlazna sekvenca s završnim događajem	U modelu poslovnog procesa završni događaj ima izlaznu sekvencu.		
	Nepravilna uporaba toka poruka			
14.	Početni događaj poruke vezan je tokom poruke s elementom početnog događaja	U modelu poslovnog procesa početni događaj poruke vezan je tokom poruke sa elementom početnog događaja.		
15.	Početni događaj poruke vezan je tokom poruke s elementom bazena	U modelu poslovnog procesa početni događaj poruke vezan je tokom poruke sa elementom bazena.		
16.	Početni događaj poruke vezan je tokom poruke s elementom potprocesa	U modelu poslovnog procesa početni događaj poruke vezan je tokom poruke sa elementom potprocesa.		
17.	Početni događaj poruke vezan je tokom poruke s elementom aktivnosti	U modelu poslovnog procesa početni događaj poruke vezan je tokom poruke sa elementom aktivnosti.		
18.	Početni događaj vezan je tokom poruke s elementom stop događaja poruke	U modelu poslovnog procesa početni događaj poruke vezan je tokom poruke sa elementom aktivnosti.		
19.	Početni događaj vezan je tokom poruke s elementom završnog događaja poruke	U modelu poslovnog procesa početni događaj poruke vezan je tokom poruke sa elementom završnog događaja poruke.		
20.	Početni događaj poruke vezan je tokom poruke s završnim događajem	U modelu poslovnog procesa početni događaj poruke vezan je tokom poruke sa elementom završnog događaja.		
21.	Bazen je vezan tokom poruke s završnim događajem	U modelu poslovnog procesa bazen je vezan tokom poruke sa elementom završnog događaja.		
22.	Potproces je vezan tokom poruke s završnim događajem	U modelu poslovnog procesa potproces vezan je tokom poruke sa elementom završnog događaja.		
23.	Aktivnost je vezana tokom poruke s završnim događajem	U modelu poslovnog procesa aktivnost je vezana tokom poruke sa elementom završnog događaja.		
24.	Stop događaj poruke je vezan tokom poruke s završnim događajem	U modelu poslovnog procesa stop događaj poruke vezan je tokom poruke s elementom završnog događaja.		
25.	Element završnog događaja poruke vezan je tokom poruke s elementom završnog događaja	U modelu poslovnog procesa element završnog događanja poruke vezan je tokom poruke sa elementom završnog događaja.		

Tabela 26. Tabela sa redoslijedom i pravilima prepoznavanja ***Semantičkih anomalija***

Red.br.	Naziv anomalije	Pravila prepoznavanja	Da	Ne
	Nepravilno nazivlje elemenata			
1.	Neimenovanje elemenata modela	U modelu poslovnog procesa postoji grafički element kojem je naziv prazan		
2.	Nejedinstvenost nazivlja elemenata modela	U modelu poslovnog procesa postoji grafički element kojem naziv nije jedinstven.		
3.	Nekonzistentnost nazivlja elemenata modela	U modelu poslovnog procesa naziv procesa i potprocesa nisu u nominativu.		
		U modelu poslovnog procesa postoje aktivnosti kojima naziv ima više riječi i više glagola.		
		U modelu poslovnog procesa postoje aktivnosti kojima naziv nema glagol.		
		U modelu poslovnog procesa postoji baza podataka kojoj naziv nije imenica.		
4.	Sinonimi u nazivlju elemenata modela	Za svaki element u pojedinoj kategoriji treba za svaki naziv pronaći sinonime.		
		Svaku listu tih sinonima treba usporediti sa svim nazivima svih elemenata iz pojedine kategorije.		
		Korisniku dati obavijest o parovima mogućih sinonima koji se pojavljuju u modelu.		
5.	Homonimi u nazivlju elemenata modela	Unutar svake kategorije pronaći više elemenata istog imena		
		Ako takvi postoje onda provjeriti da li su im nazivi u tablici homonima,		
		Ako takvi postoje onda provjeri da li ima iste ulaze i izlaze.		
	Nepravilne vrijednosti atributa			
6.	Ne postoji vrijednost atributa	U modelu poslovnog procesa postoje elementi koji imaju atributa koji nemaju nikakvu vrijednost.		
7.	Nejedinstvenost vrijednosti atributa	U modelu poslovnog procesa postoje elementi gdje vrijednost atributa nije jedinstvena unutar modela poslovnog procesa.		
8.	Neispravan tip podataka vrijednosti atributa	U modelu poslovnog procesa postoje elementi gdje tip podataka atributa nije ispravan.		
		U modelu poslovnog procesa postoje elementi gdje vrijednost atributa nije unutar zadanih granica.		
9.	Nedozvoljeni znakovi u vrijednosti atributa	U modelu poslovnog procesa postoje elementi gdje u vrijednostima atributa postoje nedozvoljeni znakovi.		

Tabela 27. Tabela sa redoslijedom i pravilima prepoznavanja ***Strukturnih anomalija***

Red.br.	Naziv anomalije	Pravila prepoznavanja	Da	Ne
	Kontrola toka			
1.	Nepotrebna kontrola toka	U modelu poslovnog procesa postoji aktivnost koja ima više od jednog izlaza s istom poslovnom stavkom.		
		U modelu poslovnog procesa postoji aktivnost koja ima više od jednog ulaza s istom poslovnom stavkom.		
		Navedene poslovne stavke su po atributima potpuno jednake.		
		Navedene poslovne stavke imaju istu jedinstvenu oznaku (ID).		
		Navedene poslovne stavke su nastale u istoj aktivnosti.		
		Navedene poslovne stavke su nastale u isto vrijeme.		
		Navedene poslovne stavke su nastale od strane istog resursa.		
2.	Redundancija modela	U modelu poslovnog procesa u jednoj stazi postoji više istih aktivnosti koji obavljaju isti posao.		
		U modelu poslovnog procesa u jednoj stazi postoji niz aktivnosti kojeg obavljaju isti resursi više puta.		
3.	Nedefinirani uvjeti vrata	U modelu poslovnog procesa postoji barem jedan element odlučivanja.		
		U modelu poslovnog procesa postoji element odlučivanja koji nema definiran uvjet.		
		U modelu poslovnog procesa ne postoji početna vrijednost uvjeta.		
4.	Kontradiktorni uvjeti vrata	U modelu poslovnog procesa postoji barem jedna vrata.		
		U modelu poslovnog procesa svi uvjeti navedenih vrata odvojeni su veznikom AND uvijek su istiniti ili neistiniti.		
5.	Deterministički nedostatak sinkronizacije	U modelu poslovnog procesa postoje elementi razdvajanja i XOR element spajanja		
		U modelu poslovnog procesa postoji element razdvajanja prethodi XOR element spajanja.		
6.	Nedeterministički nedostatak sinkronizacije	U modelu poslovnog procesa ima barem jedan element OR razdvajanja		
		U modelu poslovnog procesa ima barem jedan element XOR spajanja		
		U modelu poslovnog procesa OR element razdvajanja prethodi XOR elementu spajanja		
7.	Viseća stanja	U modelu poslovnog procesa postoji element koji ima izlaz iz elementa koji nije spojen na drugi element niti ne izlazi iz modela, niti ne završava završnim elementom.		
		U modelu poslovnog procesa postoji element koji ima ulaz u element koji nije spojen na drugi element niti ne predstavlja ulaz u model.		

8.	Netolerancija na promjene poretka	U modelu poslovnog procesa postoje minimalno dvije aktivnosti koje slijede jedna drugu.		
		Navedene aktivnosti imaju određeno vrijeme trajanja.		
		Navedene aktivnosti obavljaju različiti međusobno nezavisni resursi.		
		U modelu poslovnog procesa poslovne stavke navedenih aktivnosti su jedinstvene.		
9.	Nekonzistentnost kod inserta paralelnih grana	U modelu poslovnog procesa postoje minimalno dvije aktivnosti koje se odvijaju paralelno.		
		U modelu poslovnog procesa navedene aktivnosti imaju određeno vrijeme trajanja.		
		U modelu poslovnog procesa navedene aktivnosti obavljaju isti resursi.		
10.	Nekonzistentnost promjene iz prošlosti	U modelu poslovnog procesa postoji ima barem jedan element baze podataka.		
		U modelu poslovnog procesa u navedenu bazu podataka u jednom koraku se piše, a u drugom se čita isti slog.		
		U modelu poslovnog procesa navedena aktivnost koja čita slijedi aktivnost koja zapisuje slog.		
11.	Nedostatak tolerancije u petlji	U modelu poslovnog procesa poslije elementa odluke postoji povratna veza.		
		U modelu poslovnog procesa postoji uvjet u elementu odluke koji ima vrlo malu toleranciju zadovoljenja.		
12.	Zapinjanje toka	U modelu poslovnog procesa postoji minimalno dva elementa OR spajanja.		
		U modelu poslovnog procesa jedan element OR spajanja slijedi drugi element OR spajanja.		
		U modelu poslovnog procesa drugi element OR spajanja ima povratnu vezu s prvim elementom spajanja		
		U modelu poslovnog procesa postoji minimalno dva elementa AND spajanja.		
		U modelu poslovnog procesa element OR spajanja slijedi element AND spajanja.		
		U modelu poslovnog procesa jedan element AND spajanja slijedi drugi element AND spajanja.		
13.	Zastoj toka	U modelu poslovnog procesa mora postojati jedan element razdvajanja i jedan element AND spajanja.		
		U modelu poslovnog procesa element AND spajanja mora slijediti nakon elementa razdvajanja.		
	Anomalije pravila			
14.	Redundantna pravila	U modelu poslovnog procesa dva ili više elemenata odluke unutar istog procesa imaju isti uvjet.		
		U modelu poslovnog procesa dva ili više elemenata odluke istog procesa imaju isti ulaz.		
		U modelu poslovnog procesa dva ili više elemenata odluke istog procesa imaju iste izlaze.		
15.	Generalna pravila	U modelu poslovnog procesa dva ili više elemenata odluke unutar procesa i potprocesa imaju uvjete s istim ili detaljnijim zahtjevima.		
		U modelu poslovnog procesa dva ili više elemenata odluke istog procesa i potprocesa imaju isti ulaz.		
		U modelu poslovnog procesa dva ili više elemenata odluke istog procesa i potprocesa imaju iste izlaze.		
16.	Izravna kontradikcija (eng. Direct contradiction)	U modelu poslovnog procesa postoji barem jedan element vrata		
		U modelu poslovnog procesa navedena vrata imaju uvjet koje je kontradiktorno samo po sebi, dakle bez obzira na ulazne parametre izlaz može biti različit.		
17.	Kontradikcija u ulazu	U modelu poslovnog procesa postoji jedan element odluke.		
		U modelu poslovnog procesa navedeni uvjet odluke u svim slučajevima ne zadovoljava		
18.	Kontradikcija u zaključku	U modelu poslovnog procesa postoje elementi razdvajanja i element stapanja.		
		U modelu poslovnog procesa nakon tih elemenata slijedi element odluke.		
		U modelu poslovnog procesa poslovna stavka je ista kroz sve navedene elemente.		
19.	Zapinjanje temeljeno na pravilima	U modelu poslovnog procesa postoje element OR spajanja.		
		U modelu poslovnog procesa postoje element AND razdvajanja.		
		U modelu poslovnog procesa AND razdvajanje slijedi nakon OR spajanja.		
		U modelu poslovnog procesa AND element razdvajanja ima povratnu vezu na element OR spajanja.		
20.	Zastoj temeljen na pravilima	U modelu poslovnog procesa postoji element OR razdvajanja.		
		U modelu poslovnog procesa postoji element OR spajanja.		
		U modeli poslovnog procesa element OR razdvajanja prati element OR spajanja.		
		U modelu poslovnog procesa element OR spajanja povratno je spojen s elementom OR razdvajanja.		
21.	Izolirana pravila (eng. Isolated rules)	U modelu poslovnog procesa postoje minimalno dva elementa odlučivanja.		
		U modelu poslovnog procesa drugi element odlučivanja slijedi nakon prvog elementa odlučivanja.		
		U modelu poslovnog procesa uvjet jednog elementa odlučivanja mora biti sadržan u uvjetu drugog elementa.		
22.	Beskorisni zaključci (eng. Useless conclusions)	U modelu poslovnog procesa postoji barem jedan element odlučivanja.		
		U modelu poslovnog procesa u navedenom elementu uvjet odlučivanja mora imati više manjih pravila koji su podskup određenog pravila u uvjetu.		
23.	Viseći uvjeti (eng. Dangling Conditions)	U modelu poslovnog procesa postoji barem jedan element odlučivanja.		
		U modelu poslovnog procesa navedeni element odlučivanja ima barem jedan ulaz koji nije spojen poslovnom stavkom za neki drugi element.		
24.	Nepostojeća pravila	U modelu poslovnog procesa postoji barem jedan element odlučivanja.		

		U modelu poslovnog procesa u navedenom elementu odlučivanja nije definirano pravilo.		
25.	Redundantni podaci	U modelu poslovnog procesa postoji aktivnost koja upisuje podatke u bazu podataka, no one se ne koriste u daljem toku poslovnog procesa ni u bilo kojem drugom poslovnom procesu organizacije ili izvan nje.		
		U modelu poslovnog procesa postoji objekt baze podataka u koji se upisuje podaci, no nikad se ne čitaju.		
26.	Izgubljeni podaci	U modelu poslovnog procesa postoji više aktivnosti koji upisuju, ažuriraju i čitaju podatke po istom dokumentu u istome poslovnom procesu.		
		U modelu poslovnog procesa postoji aktivnost koja briše određeni podatak i izvodi se prije aktivnosti čitanja podataka.		
		U modelu poslovnog procesa ne postoji mehanizam provjere vremena čitanja i zapisivanja podatka tijekom rada poslovnog procesa		
27.	Nepostojeći podaci	U modelu poslovnog procesa postoji barem jedan element baze podataka.		
		U modelu poslovnog procesa postoje aktivnosti koje upisuju, ažuriraju i čitaju isti podatak u istoj bazi podataka.		
		U modelu poslovnog procesa ne postoji mehanizam provjere postojanja određenog podatka prije njegovog korištenja u poslovnom procesu.		
28.	Nekonzistentni podaci	U modelu poslovnog procesa postoji barem jedan element baze podataka.		
		U modelu poslovnog procesa postoji više aktivnosti koji upisuju isti podatak u objekt podataka.		
29	Nikad izbrisani podaci	U modelu poslovnog procesa postoji barem jedan privremeni element baze podataka.		
		U modelu poslovnog procesa postoji aktivnost koja privremeno sprema podatke u bazu podataka, a ne postoji aktivnost brisanja podataka iz privremene baze.		
		U modelu poslovnog procesa postoji aktivnost koja privremeno sprema podatke u bazu podataka, a ne postoji aktivnost čitanja tih podataka onda se radi o anomaliji redundantnih podataka.		
30.	Dva puta izbrisani podaci	U modelu poslovnog procesa postoji element baze podataka		
		U modelu poslovnog procesa postoje barem dvije aktivnosti koje brišu određene slogove iz baze podataka.		
31.	Podaci neizbrisani na vrijeme	U modelu poslovnog procesa mora postojati barem jedan element baze podataka.		
		U modelu poslovnog procesa mora postojati barem jedan element aktivnosti koji briše podatak u bazi podataka.		
		U modelu poslovnog procesa mora postojati barem jedna aktivnost koja čita podatak.		
		U modelu poslovnog procesa postoji aktivnost čitanja koja mora biti poslije nakon aktivnosti brisanja.		

Tabela 28. Tabela sa redoslijedom i pravilima prepoznavanja ***Anomalija resursa***

Red.br.	Naziv anomalije	Prepoznavanje anomalije	Da	Ne
	Anomalije obrazaca kreiranja			
1.	Direktna alokacija u slučaju da više resursa ima isti naziv	U modelu procesa postoji više različitih resursa s istim nazivom.		
2.	Direktna alokacija resursa bez mehanizma eskalacije	U modelu poslovnog procesa postoji aktivnost koje nema određeno maksimalno vrijeme trajanja.		
		U modelu poslovnog procesa ne postoji aktivnost koja se aktivira ako istekne maksimalno vrijeme trajanja.		
3.	Alokacija temeljena na ulogama kad odabrani resurs nije pripadnik određene uloge/role	U modelu poslovnog procesa postoji aktivnost koji ima pripadajući resurs čija rola nije sukladna zahtjevima određene aktivnosti.		
4.	Alokacija temeljena na ulogama kad resurs pripada u više od jedne grupe	U modelu poslovnog procesa pojedinim aktivnostima zadani su resursi koji se nalaze u više od jedne role.		
5.	Alokacija temeljena na ulogama kod grupe sa jednim resursom	U modelu poslovnog procesa postoje aktivnosti koje se izvode paralelno, a koje imaju dodijeljene role sa samo jednim resursom.		
6.	Automatsko izvođenje bez mehanizma provjere izvođenja	U modelu poslovnog procesa provjerava se da li se je aktivnost već izvršila?		
	Anomalije samopokretajućih obrazaca			
	Početak kreiranja sa alokacijom resursa	U modelu poslovnog procesa ne postoji start event, nego se proces okida iz vana.		
		U modelu poslovnog procesa postoji aktivnost ima vrijednost čekanja resursa (eng. *ResourceWaitTime*) veći o nule.		
		U modelu poslovnog procesa navedena aktivnost odmah dobiva sve poslovne stavke potrebne za rad.		
8.	Djelomično lančano izvršenje u poslovnom procesu	U modelu poslovnog procesa postoje više aktivnosti koje se odvijaju sekvencijalno.		
		U modelu poslovnog procesa ne postoji element razdvajanja i stapanja sekvenci.		
		U modelu poslovnog procesa svaku od navedenih aktivnosti obavlja različit resurs		
		U modelu poslovnog procesa navedene aktivnosti ne ovise slijedno jedna od drugoj.		
		U modelu poslovnog procesa postoje dovoljni kapacitet resursa da se aktivnosti odvijaju paralelno.		
	Anomalije višestrukih obrazaca resursa			
9.	Istovremeno izvođenje kod ograničenih kapaciteta resursa	U modelu poslovnog procesa postoji više od jedne aktivnosti.		
		U modelu poslovnog procesa dva ili više aktivnosti odvijaju se paralelno od strane istih resursa.		
		U modelu poslovnog procesa resursi nemaju dovoljno kapaciteta za paralelni rad.		

S obzirom da su obrasci opisani tekstualno, primjerima, pravilima prepoznavanja i načinom rješavanja, za demonstraciju rada navedenih mjernica poslužiti će modeli iz realnog života koji već imaju svoju uporabnu vrijednosti. Referentni skup modela koji će se koristiti u ovome radu dolaze iz magistarskog rada Vidačić, T.: *Referentni model procesa visokogradnje uz intenzivnu ICT potporu* [64]. Kao primjere uzeti su modeli pod nazivom ***Pratiti realizaciju po fazama*** i ***Nadzirati gradilište***.

Primjer modela poslovnog procesa *Pratiti realizaciju po fazama*

Poslovni proces ***Pratiti realizaciju po fazama*** kao što je vidljivo na Slika 129. modela poslovnog procesa nalazi se u obliku slobodne forme, što znači da neke anomalije vezane uz objekte staza u bazenu u ovome slučaju nije moguće provjeriti. Sam poslovni proces opisan je slikom u BPMN notaciji i tabelarnim opisom u kojem se nalaze nazivi elemenata i njihova uloga u modelu poslovnog procesa. Kao što je vidljivo iz tabele objekata i iz slike modela poslovnog procesa navedeni proces ***Pratiti realizaciju po fazama*** sastoji se o cijelog niza objekata, kao što su: *Početni događa*j, *Aktivnosti*, *Potprocesi*, *Petlje*, *Servisi*, *Baze podataka*, *Element dijeljenja* i *Završni element*.

Sa prepoznavanjem anomalija počinje se tabelom ***Anomalija prikaza modela***. Dakle odgovara se na tvrdnju, a ako tvrdnja stoji, označi se prvi stupac sa **Da**, a ako tvrdnja ne stoji, označi se stupac **Ne**.

Iz tabele sa redoslijedom i pravilima prepoznavanja ***Anomalija prikaza modela*** za proces *Pratiti realizaciju po fazama*, vodljivo je da je prepoznato sedam anomalija.

Prepoznata je anomalija ***Previše elemenata u modelu***, jer model sadrži ukupni broj elemenata veći od 9, jer je u promatranom modelu prikazano 20 elemenata. Prema predloženoj taksonomiji navedena anomalija je iz kategorije ***Anomalije organizacije percepcije*** iz potkategorije ***Anomalije jednostavnosti***.

Nadalje, prepoznata je anomalija ***Nekompaktni i razvučen model***, jer model zadovoljava pravila prepoznavanja da model ima više od jednog element, model sadrži grafičke elemente jedne grane koji se nalaze između dva grafička elemenata druge grane. U ovome slučaju kao vrijednost udaljenosti jednog elementa od drugog upotrijebljen je uvjet od prosječne širine objekta, no s obzirom naočitost ove anomalije moguće je uzeti i četverostruki iznos širine objekta i anomalija bi i dalje bila prepoznata. Prema predloženoj taksonomiji navedena anomalija je iz kategorije ***Anomalije organizacije percepcije*** iz potkategorije ***Anomalije grupiranja***.

Dalje, u modelu je prepoznata anomalija ***Križanje i savijanje veza između elemenata*** modela, jer anomalija zadovoljava sljedeća pravila prepoznavanja navedene anomalije. Dakle, model ima više od jednog elementa, elementi iste kategorije nisu na istoj visini i elementi toka sječu se sa drugim elementima kontrole toka. Prema predloženoj taksonomiji navedena anomalija je iz kategorije ***Anomalije organizacije percepcije*** iz potkategorije ***Anomalije kontinuiranosti***.

Tabela 29. Tabela objekata modela poslovnog procesa ***Pratiti realizaciju po fazama***

Pratiti realizaciju po fazama			**Trajanje [JMj]**
Pratiti realizaciju po fazama je proces koji se ponavlja tako dugo dok se svi radni koraci iz razrađenog ugovornog *Troškovnika* ne obave. Proces ***Pratiti realizaciju po fazama*** započinje nakon procesa ***Otvoriti gradilište***, dok ga slijedi proces ***Primopredati objekt***. Realizacija radova se izvodi prema fazama *Terminskog plana*. Ulaz u proces su; *Glavni radni nalog*, *Troškovnik*, *Terminski plan*. Izlaz iz ovog procesa je *Projektna dokumentacija*.			
Radno mjesto /Org. Jed.	**Naziv radnih koraka**	**Opis radnih koraka**	
Voditelj projekta	*Pregledati terminski plan*	Pregledava se *Terminski plan*.	20 '
Voditelj projekta	*Pregledava se terminski plan.*	Pregledava se *Terminski plan*.	2 h
◄	**Pratiti resurse**	Ulaz u proces ***Pratiti resurse*** su *Glavni radni nalog*, *Troškovnik* i *Terminski plan*. U početku procesa provjerava se da li su svi resursi raspoloživi za rad. Ukoliko jesu tada se evidentira potrošnja materijala, evidentira se rad mehanizacije, evidentira se rad podizvođača, evidentira se rad ljudstva. Ukoliko svi resursi nisu raspoloživi pribavljaju se potrebni resursi toliko dugo dok svi resursi nisu na raspolaganju izvođačima radova. Utvrđuje se ukupna dosadašnja realizacija, ažurira se *Građevinski dnevnik* te se zatim reprogramira realizacijski plan. Izlaz iz procesa ***Pratiti Resurse*** su *Glavni radni nalog*, *Troškovnik* i *Terminski plan*.	►
Voditelj projekta	*Kreirati zahtjev za atestima podizvođača*	Kreiraju se *Zahtjevi* za atestima podizvođača ako nisu priloženi ugovoru.	
Dostavljač	*Otpremiti zahtjev*	Otprema se *Zahtjev*.	6 '
Podizvođač	**Izdati atest radova**	Servis koji opisuje kako **Podizvođači** izdaju *Ateste* da su radovi izvedeni u skladu s zahtjevima projekta.	
Tehničar-Administrator	*Zaprimiti atest*	*Atesti* se zaprimaju.	6 '
Tehničar-Administrator	*Ažurirati BP atesta*	Ažurira se baza podataka atesta	5 '
Voditelj projekta	*Ažurirati građevinsku knjigu*	Ažurira se *Građevinska knjiga*.	1 h
◄	**Nadzirati gradilište**	Proces ***Nadzor gradilišta*** je poslovni proces koji se ponavlja tako dugo dok se ne isprave manjkavosti na gradilištu. Ovaj proces se izvodi u poslovnom procesu ***Realizacija po fazama***, a slijede ga servisi u kojem se ovjerava *Građevinska knjiga* i *Građevinski dnevnik*. Ulaz u ovaj proces su *Građevinska knjiga*, *Projektna dokumentacija* i *Zahtjev*, Dok su izlazi iz ovog procesa *Građevinska knjiga* i *Građevinski dnevnik*.	►
Nadzor	**Ovjeriti građevinsku knjigu**	Ovjerava se *Građevinska knjiga* s strane **Nadzora**.	
Nadzor	**Ovjeriti građevinski dnevnik**	Ovjerava se *Građevinski dnevnik* s strane **Nadzora**.	
◄	**Situirati**	Ulaz u ***Situiranje*** je *Građevinska knjiga* i *Građevinski dnevnik*, ta dva dokumenta su temelj za kreiranje *Situacije*. Oba dokumenta moraju biti ovjerena s strane **Nadzora**. U početku procesa se prikupljaju potrebni podaci za kreiranje *Situacije* i zatim se situacija kreira Izlaz je *Situacija*.	►
Nadzor	**Ovjeriti Situaciju**	**Neovisni nadzor** mora ovjeriti *Situaciju*.	
◄	**Izraditi fakturu situacije**	Na temelju *Situacije* koju mora potpisati **Izvođač**, **Neovisni nadzor** i **Investitor** izrađuje se *Račun* koji **Investitor** mora platiti. Prvo se zaprima *Situacija*, pa se provjerava da nema nekih manjkova i onda se izrađuje *Račun*. Ulaz u proces je *Situacija*, dok je izlaz *Račun*.	►
Investitor	**Platiti Fakturu**	Na temelju situacije plaća se *Račun*.	
◄	**Zaprimiti potvrdu plaćanja**	U ovome procesu prvo se zaprima *Obavijest o plaćanju* s strane vanjskih organizacija, ta *Obavijest* može biti putem Interneta, primanjem kopije virmana faxom ili fizički. *Obavijest o plaćanju* se uspoređuje s bankovnim izvodima, te se zatim taj *Ulazni račun* knjiži. *Ulazni račun* se tereti na *Glavni radni nalog* s ciljem jednostavnijeg praćenja troškova pojedinog projekta.	►

Zatim, prepoznata je anomalija ***Nepoštivanje pozicije hijerarhije modela***, jer u modelu postoje više od jednog elementa iste kategorije (aktivnosti) i vidljivo je da se jedna aktivnost veže s drugom aktivnošću na način da su jedan iznad drugoga i u istoj mjeri udaljeni od ruba, čime se dobiva dojam da je jedan dio grane hijerarhijski važniji od druge grane. Prema predloženoj taksonomiji navedena anomalija je iz kategorije ***Anomalije organizacije percepcije*** iz potkategorije ***Anomalije prisnosti***.

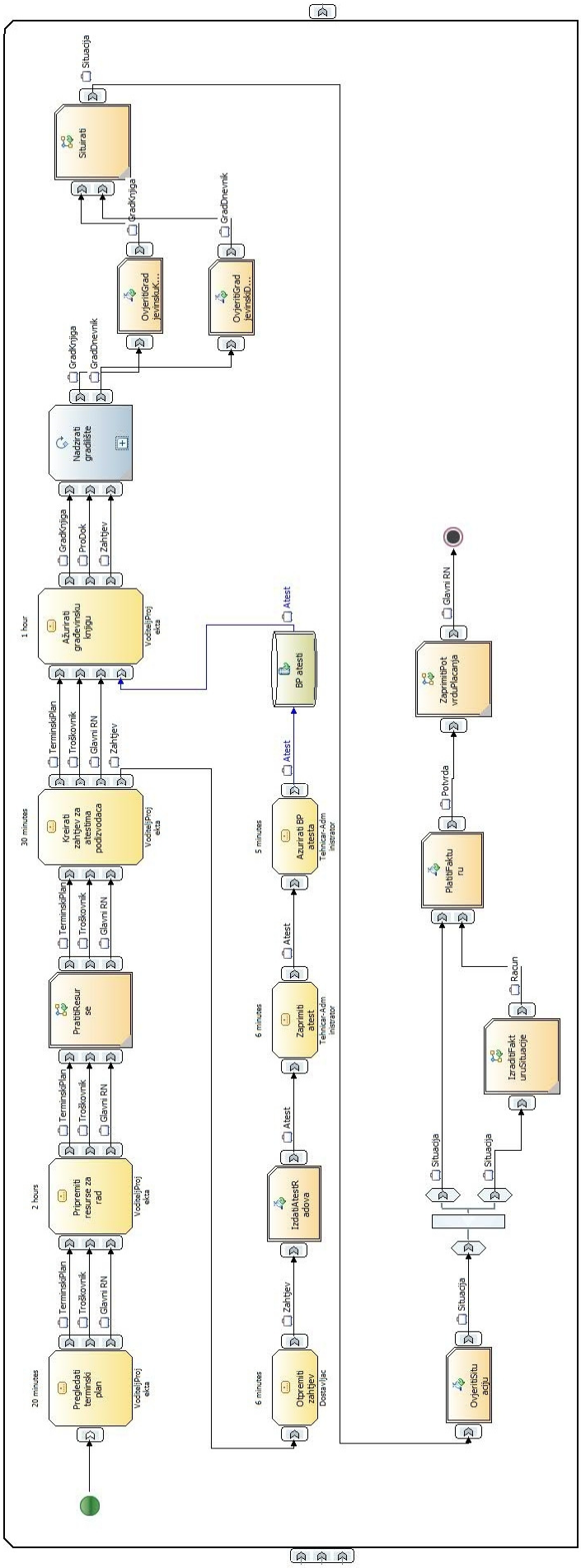
Nadalje, prepoznata je anomalija ***Slični elementi nemaju istu veličinu***. Model ima više od jednog elementa i elementi iste kategorije (aktivnosti) imaju u pojedinim elementima dvostruku veličinu i širinu od druge aktivnosti. Prema predloženoj taksonomiji navedena anomalija je iz kategorije ***Anomalije organizacije percepcije*** iz potkategorije ***Anomalije sličnosti***.

Nadalje, prepoznata je i anomalija ***Slični elementi nemaju isti oblik***, navedena anomalija vrlo je bliska s prethodnom anomalijom, no u ovome slučaju može se reći da su pojedine aktivnosti dvostruke visine od drugih aktivnosti. Prema predloženoj taksonomiji navedena anomalija je iz kategorije ***Anomalije organizacije percepcije*** iz potkategorije ***Anomalije sličnosti.***

Ako se model podijeli na četiri kvadranta po širini i po dužini može se prepoznati anomalija ***Necentrirani i nesimetrični model***. Ako se model podijeli horizontalno vidi se da je u gornjem kvadrantu više elemenata nego u donjem, a ako se model podijeli vertikalno, vidi se da lijeva strana modela ima više elemenata od desne strane. Prema predloženoj taksonomiji navedena anomalija je iz kategorije ***Anomalije organizacije percepcije*** iz potkategorije ***Anomalije simetrije.***

Prema već unaprijed definiranom redoslijedu provjere anomalija slijedi provjera sljedeće velike kategorije anomalija i to ***Sintaktičkih anomalija.*** Iz tabele sa redoslijedom i pravilima prepoznavanja ***Sintaktičkih anomalija*** za model poslovnog procesa ***Pratiti realizaciju po fazama*** vidi se da nije prepoznata ni jedna anomalija. To ne treba čuditi, jer je model izveden u IBM WebSphere alatu za modeliranje poslovnih procesa u kojem je ugrađena sintaktička provjera modela.

Redoslijed provjere zatim traži provjeru anomalija za ***Semantičke anomalija***. S obzirom da je kod izgradnje rada Vidačić, T.: *Referentni model procesa visokogradnje uz intenzivnu ICT potporu* [64] posebna pozornost bila usmjerena na imenovanje objekata aktivnosti, baza i potprocesa, u ovome modelu nije pronađena ni jedna ***Semantička anomalija***. Osim toga navedeni alat IBM WebSphere ima ugrađenu provjeru jedinstvenosti nazivlja elemenata u modelu.



Slika 127. ***Pratiti realizaciju po fazama*** - po poslovnim funkcijama

Slijedi provjera ***Strukturnih anomalija***. Kao što je vidljivo iz tabele sa redoslijedom i pravilima prepoznavanja strukturnih anomalija pronađene su dvije potencijalne strukturne anomalije. I to, anomalija ***Nekonzistentnost promjene iz prošlosti*** koja ima tri pravila prepoznavanja i to: u modelu poslovnog procesa postoji ima barem jedan element baze podataka (BP atesti), u modelu poslovnog procesa u navedenu bazu podataka u jednom koraku se piše (Aktivnost: **Ažurirati BP atesta**), a u drugom se čita isti slog (Aktivnost: **Ažurirati građevinsku knjigu**). Osim toga u modelu poslovnog procesa navedena aktivnost koja čita slijedi aktivnost koja zapisuje slog. Osim navedene anomalije koja spada u kategoriju ***Anomalija kontrola toka*** i to u potkategoriju ***Specijalni slučajevi toka*** prepoznata je i potencijalna anomalija ***Zastoj toka***. Ova tvrdnja proizlazi iz toga što u modelu postoji jedan element razdvajanja i jedno AND spajanje koje se nalazi na ulazu u servis **PlatiFakturu**. Servis **PlatiFakturu** čeka tako dugo dok ne dobije obje ulazne stavke razmjene i to *Situaciju* i *Račun*. Osim toga potvrđeno je i drugo pravilo, a to je da AND spajanje slijedi nakon elementa razdvajanja. Prema predloženoj taksonomiji navedena anomalija ***Zastoj toka*** je iz kategorije ***Kontrola toka*** u potkategoriji ***Narušavanje ispravnosti.***

Slijedi provjera ***Anomalija resursa***. Kao što je već prije napomenuto, BPMN notacija ima ograničen naglasak na rad s resursima i to zapravo ne predstavlja ključno područje uporabe BPMN notacije. No, prema tabeli sa redoslijedom i pravilima prepoznavanja ***Anomalija resursa*** za poslovni proces ***Pratiti realizaciju po fazama*****,** prepoznata je anomalija ***Direktna alokacija resursa bez mehanizma eskalacije***. Pravila prepoznavanja navedene anomalije su se pokazala potvrđenima i to, da u modelu poslovnog procesa postoji aktivnost koje nema određeno maksimalno vrijeme trajanja, a zatim da u modelu poslovnog procesa ne postoji aktivnost koja se aktivira ako istekne maksimalno vrijeme trajanja.

Prema svemu navedenom u ovom modelu prepoznato je u ***Anomalijama prikaza modela*** sedam anomalija. U ***Anomalijama strukturnih anomalija*** su prepoznate dvije anomalije, u ***Anomalijama resursa*** je prepoznata jedna anomalija dok u ***Sintaktičkim*** i ***Semantičkim*** anomalijama nije prepoznata ni jedna anomalija u modelu poslovnog procesa ***Pratiti Realizaciju po fazama***.

Tabela 30. Tabela sa redoslijedom i pravilima prepoznavanja ***Anomalija prikaza modela*** za proces **Pratiti realizaciju po fazama**

Red.br.	Naziv anomalije	Pravila prepoznavanja	Da	Ne
	Anomalije organizacije percepcije			
	Anomalije jednostavnosti			
1.	Previše elemenata u modelu	Model poslovnog procesa sadrži ukupni broj elemenata veći od 9.	●	
2.	Neselektivni prikaz elemenata u modelu	Model poslovnog procesa sadrži aktivnosti koje ne spadaju u isti kontekst procesa.		●
		Model poslovnog procesa prikazuje sadržaje razmjene koji nisu bitni za prikaz modela poslovnog procesa.		●
		Model poslovnog procesa koristi resurse koji nisu bitni za prikaz modela poslovnog procesa.		●
	Anomalije grupiranja			
3.	Nekompaktni i razvučen model	Model poslovnog procesa ima više od jednog elementa.	●	
		Model sadrži grafički element kategorije koji se horizontalno ili vertikalno nalazi između dva spojena elementa iste kategorije.	●	
		Model sadrži uvjet koji govori o tome da se neki element nalazi previše udaljen od drugog elementa.	●	
	Anomalije kontinuiranosti			
4.	Križanje i savijanje veza između elemenata modela	Model poslovnog procesa ima više od jednog elementa.	●	
		U modelu poslovnog procesa povezani elementi iste kategorije nisu na istoj visini.	●	
		Element kontrole toka siječe se sa drugim elementom kontrole toka.	●	
	Anomalije prisnosti			
5.	Nepoštivanje pozicije hijerarhije modela	Model poslovnog procesa ima više od jednog elemenata iste kategorije.	●	
		U modelu poslovnog procesa postoji element koji je spojen s drugim elementom iste kategorije, a jedan je iznad drugoga i u istoj mjeri udaljen od ruba modela.	●	
	Anomalije sličnosti			
6.	Slični elementi nisu grupirani zajedno	Model procesa ima više od jednog elementa.	●	
		U modelu poslovnog procesa postoje elementi čije kategorija objekta ima obilježje sličnog objekta.		●
		U modelu poslovnog procesa broj sličnih objekata u modelu procesa je veći od 1.		●
7.	Slični elementi nemaju iste boje	Model poslovnog procesa ima više od jednog elementa.	●	
		U modelu poslovnog procesa postoje grafički elementi iste kategorije koji imaju različite boje.		●
8.	Slični elementi nemaju istu veličinu	Model poslovnog procesa ima više od jednog elementa.	●	
		U modelu poslovnog postoji element iste kategorije, a da li mu je širina i visina dvostruko veća od prethodnog elementa.	●	
9.	Slični elementi nemaju isti oblik	U modelu poslovnog procesa ima više od jednog elementa.	●	
		U modelu poslovnog postoji element iste kategorije, a da li mu je širina ili visina dvostruko veća od prethodnog elementa.	●	
	Anomalije povezanosti			
10.	Preklapanje elemenata modela	Model poslovnog procesa ima više od jednog elementa.	●	
		U modelu poslovnog procesa postoji element čija je lijeva pozicije i širina manja od lijeve pozicije i širine sljedećeg elementa.		●
		U modelu poslovnog procesa postoji element čija je gornja pozicija i visina manja od gornje pozicije i visine sljedećeg elementa.		●
	Anomalije percepcije razlikovanja			
	Anomalije obrisa			
11.	Elementi modela se međusobno dodiruju	Model poslovnog procesa sadrži više od jednog elementa.	●	
		U modelu poslovnog procesa postoji grafički element u skupu objekata grafičkih elemenata, koji bez obzira na kategoriju ima poziciju koja zbrojena sa dimenzijom kontrole i oduzeta od pozicije susjednog elementa zbrojene s dimenzijom kontrole daje iznos manju od širine grafičke kontrole.		●
		U modelu poslovnog procesa elementi se međusobno dodiruju i vertikalno i horizontalno.		●
	Anomalije orijentacije			
12.	Modeli orijentirani s desna na lijevo	Model poslovnog procesa ima više od jednog elementa	●	
		U modelu poslovnog procesa postoji tok između dva grafička elementa gdje je slijedni grafički element lijevo od promatranog		●
13.	Modeli orijentirani od dolje prema gore	Model poslovnog procesa sadrži više od jednog elementa,	●	
		U modelu poslovnog procesa postoji element kod kojih postoji kontrola toka, a kod kojih je prethodni element niže od slijednog elementa.		●
14.	Vertikalni nazivi elemenata	U modelu poslovnog procesa postoji grafička kontrola iste kategorije koja ima obilježje orijentacija naslova a da je vertikalna.		●
15.	Vertikalni opisi unutar modela	U modelu poslovnog procesa postoji element koji ima vertikalni opis, tada takav model procesa ima anomaliju vertikalni opis unutar modela.		●
	Anomalije simetrije			
16.	Necentrirani i nesimetrični model	U modelu poslovnog procesa broj elementa koji se nalaze u lijevom horizontalnom kvadrantu različit je od broj elemenata koji se nalaze u desnom horizontalnom kvadrantu.	●	
		U modelu poslovnog procesa broj elementa koji se nalaze u gornjem vertikalnom kvadrantu različit je od broj elemenata koji se nalaze u donjem vertikalnom ..	●	

Tabela 31. Tabela sa redoslijedom i pravilima prepoznavanja ***Sintaktičkih anomalija*** za model poslovnog procesa ***Pratiti realizaciju po fazama***

Red.br.	Naziv anomalije	Pravila prepoznavanja	Da	Ne
	Nepravilna uporaba spremnika			
1.	Bazen bez elemenata	U modelu poslovnog procesa postoji bazen u kojem ne postoji ni jedan grafički element.		●
2.	Nepravilna uporaba bazena	U modelu poslovnog procesa postoji bazen ima tok sekvence koji prelazi granice spremnika.		●
3.	Staza bez elemenata	U modelu poslovnog procesa postoji staza bez elemenata.		●
4.	Nepravilna uporaba staza	U modelu poslovnog procesa postoji staza no ne postoji i bazen koji sadrži navedenu stazu.		●
	Nepravilna uporaba objekata toka			
5.	Više početnih događaja u modelu	U modelu poslovnog procesa postoji više od jednog početnog događaja.		●
6.	Nepovezanost početnog događaja s ostalim elementima modela	U modelu poslovnog procesa početni događaj nema izlaznu sekvencu.		●
7.	Početni događaj ima više od jedne izlazne sekvence	U modelu poslovnog procesa mora postojati barem jedan početni događaj.	●	
		Početni događaj ima više od jedne izlazne sekvence.		●
8.	Nepostojanje završnog događaja	U modelu poslovnog procesa ne postoji završni događaj.		●
9.	Nepovezanost završnog događaja s ostalim elementima modela	U modelu poslovnog procesa ulazna sekvenca završnog događaja je prazan skup.		●
10.	Završni događaj ima više od jedne ulazne sekvence	U modelu poslovnog procesa mora biti barem jedan završni događaj.	●	
		Završni događaj ne smije imati više od jedne ulazne sekvence.		●
11.	Nepostojanje početnog događaja	U modelu poslovnog procesa ne postoji ulazna sekvenca.	●	
		U modelu poslovnog procesa broj početnih događaja jednak je nuli.		●
	Nepravilna uporaba veza			
12.	Ulazna sekvenca sa početnim događajem	U modelu poslovnog procesa početni događaj ima ulaznu sekvencu.		●
13.	Izlazna sekvenca s završnim događajem	U modelu poslovnog procesa završni događaj ima izlaznu sekvencu.		●
	Nepravilna uporaba toka poruka			
14.	Početni događaj poruke vezan je tokom poruke s elementom početnog događaja	U modelu poslovnog procesa početni događaj poruke vezan je tokom poruke sa elementom početnog događaja.		●
15.	Početni događaj poruke vezan je tokom poruke s elementom bazena	U modelu poslovnog procesa početni događaj poruke vezan je tokom poruke sa elementom bazena.		●
16.	Početni događaj poruke vezan je tokom poruke s elementom potprocesa	U modelu poslovnog procesa početni događaj poruke vezan je tokom poruke sa elementom potprocesa.		●
17.	Početni događaj poruke vezan je tokom poruke s elementom aktivnosti	U modelu poslovnog procesa početni događaj poruke vezan je tokom poruke sa elementom aktivnosti.		●
18.	Početni događaj vezan je tokom poruke s elementom stop događaja poruke	U modelu poslovnog procesa početni događaj poruke vezan je tokom poruke sa elementom aktivnosti.		●
19.	Početni događaj vezan je tokom poruke s elementom završnog događaja poruke	U modelu poslovnog procesa početni događaj poruke vezan je tokom poruke sa elementom završnog događaja poruke.		●
20.	Početni događaj poruke vezan je tokom poruke s završnim događajem	U modelu poslovnog procesa početni događaj poruke vezan je tokom poruke sa elementom završnog događaja.		●
21.	Bazen je vezan tokom poruke s završnim događajem	U modelu poslovnog procesa bazen je vezan tokom poruke sa elementom završnog događaja.		●
22.	Potproces je vezan tokom poruke s završnim događajem	U modelu poslovnog procesa potproces vezan je tokom poruke sa elementom završnog događaja.		●
23.	Aktivnost je vezana tokom poruke s završnim događajem	U modelu poslovnog procesa aktivnost je vezana tokom poruke sa elementom završnog događaja.		●
24.	Stop događaj poruke je vezan tokom poruke s završnim događajem	U modelu poslovnog procesa stop događaj poruke vezan je tokom poruke s elementom završnog događaja.		●
25.	Element završnog događaja poruke vezan je tokom poruke s elementom završnog događaja	U modelu poslovnog procesa element završnog događanja poruke vezan je tokom poruke sa elementom završnog događaja.		●

Tabela 32. Tabela sa redoslijedom i pravilima prepoznavanja ***Semantičkih anomalija*** za model poslovnog procesa ***Pratiti realizaciju po fazama***

Red.br.	Naziv anomalije	Pravila prepoznavanja	Da	Ne
	Nepravilno nazivlje elemenata			
1.	Neimenovanje elemenata modela	U modelu poslovnog procesa postoji grafički element kojem je naziv prazan		●
2.	Nejedinstvenost nazivlja elemenata modela	U modelu poslovnog procesa postoji grafički element kojem naziv nije jedinstven.		●
3.	Nekonzistentnost nazivlja elemenata modela	U modelu poslovnog procesa naziv procesa i potprocesa nisu u nominativu.		●
		U modelu poslovnog procesa postoje aktivnosti kojima naziv ima više riječi i više glagola.		●
		U modelu poslovnog procesa postoje aktivnosti kojima naziv nema glagol.		●
		U modelu poslovnog procesa postoji baza podataka kojoj naziv nije imenica.		●
4.	Sinonimi u nazivlju elemenata modela	Za svaki element u pojedinoj kategoriji treba za svaki naziv pronaći sinonime.		●
		Svaku listu tih sinonima treba usporediti sa svim nazivima svih elemenata iz pojedine kategorije.		●
		Korisniku dati obavijest o parovima mogućih sinonima koji se pojavljuju u modelu.		●
5.	Homonimi u nazivlju elemenata modela	Unutar svake kategorije pronaći više elemenata istog imena		●
		Ako takvi postoje onda provjeriti da li su im nazivi u tablici homonima.		●
		Ako takvi postoje onda provjeri da li ima iste ulaze i izlaze.		●
	Nepravilne vrijednosti atributa			
6.	Ne postoji vrijednost atributa	U modelu poslovnog procesa postoje elementi koji imaju atributa koji nemaju nikakvu vrijednost.		●
7.	Nejedinstvenost vrijednosti atributa	U modelu poslovnog procesa postoje elementi gdje vrijednost atributa nije jedinstvena unutar modela poslovnog procesa.		●
8.	Neispravan tip podataka vrijednosti atributa	U modelu poslovnog procesa postoje elementi gdje tip podataka atributa nije ispravan.		●
		U modelu poslovnog procesa postoje elementi gdje vrijednost atributa nije unutar zadanih granica.		●
9.	Nedozvoljeni znakovi u vrijednosti atributa	U modelu poslovnog procesa postoje elementi gdje u vrijednostima atributa postoje nedozvoljeni znakovi.		●

Tabela 33. Tabela sa redoslijedom i pravilima prepoznavanja ***Strukturnih anomalija*** za poslovni proces ***Pratiti realizaciju po fazama***

Red.br.	Naziv anomalije	Pravila prepoznavanja	Da	Ne
	Kontrola toka			
1.	Nepotrebna kontrola toka	U modelu poslovnog procesa postoji aktivnost koja ima više od jednog izlaza s istom poslovnom stavkom.		●
		U modelu poslovnog procesa postoji aktivnost koja ima više od jednog ulaza s istom poslovnom stavkom.		●
		Navedene poslovne stavke su po atributima potpuno jednake.		●
		Navedene poslovne stavke imaju istu jedinstvenu oznaku (ID).		●
		Navedene poslovne stavke su nastale u istoj aktivnosti.		●
		Navedene poslovne stavke su nastale u isto vrijeme.		●
		Navedene poslovne stavke su nastale od strane istog resursa.		●
2.	Redundancija modela	U modelu poslovnog procesa u jednoj stazi postoji više istih aktivnosti koji obavljaju isti posao.		●
		U modelu poslovnog procesa u jednoj stazi postoji niz aktivnosti kojeg obavljaju isti resursi više puta.		●
3.	Nedefinirani uvjeti vrata	U modelu poslovnog procesa postoji barem jedan element odlučivanja.		●
		U modelu poslovnog procesa postoji element odlučivanja koji nema definiran uvjet.		●
		U modelu poslovnog procesa ne postoji početna vrijednost uvjeta.		●
4.	Kontradiktorni uvjeti vrata	U modelu poslovnog procesa postoji barem jedna vrata.		●
		U modelu poslovnog procesa svi uvjeti navedenih vrata odvojeni su veznikom AND uvijek su istiniti ili neistiniti.		●
5.	Deterministički nedostatak sinkronizacije	U modelu poslovnog procesa postoje elementi razdvajanja i XOR element spajanja		●
		U modelu poslovnog procesa postoji element razdvajanja prethodi XOR element spajanja.		●
6.	Nedeterministički nedostatak sinkronizacije	U modelu poslovnog procesa ima barem jedan element OR razdvajanja		●
		U modelu poslovnog procesa ima barem jedan element XOR spajanja		●
		U modelu poslovnog procesa OR element razdvajanja prethodi XOR elementu spajanja		●
7.	Viseća stanja	U modelu poslovnog procesa postoji element koji ima izlaz iz elementa koji nije spojen na drugi element niti ne izlazi iz modela, niti ne završava završnim elementom.		●
		U modelu poslovnog procesa postoji element koji ima ulaz u element koji nije spojen na drugi element niti ne predstavlja ulaz u model.		●

8.	Netolerancija na promjene poretka	U modelu poslovnog procesa postoje minimalno dvije aktivnosti koje slijede jedna drugu.	●	
		Navedene aktivnosti imaju određeno vrijeme trajanja.	●	
		Navedene aktivnosti obavljaju različiti međusobno nezavisni resursi.	●	
		U modelu poslovnog procesa poslovne stavke navedenih aktivnosti su jedinstvene.		●
9.	Nekonzistentnost kod inserta paralelnih grana	U modelu poslovnog procesa postoje minimalno dvije aktivnosti koje se odvijaju paralelno.		●
		U modelu poslovnog procesa navedene aktivnosti imaju određeno vrijeme trajanja.	●	
		U modelu poslovnog procesa navedene aktivnosti obavljaju isti resursi.	●	
10.	Nekonzistentnost promjene iz prošlosti	U modelu poslovnog procesa postoji ima barem jedan element baze podataka.	●	
		U modelu poslovnog procesa u navedenu bazu podataka u jednom koraku se piše, a u drugom se čita isti slog.	●	
		U modelu poslovnog procesa navedena aktivnost koja čita slijedi aktivnost koja zapisuje slog.	●	
11.	Nedostatak tolerancije u petlji	U modelu poslovnog procesa poslije elementa odluke postoji povratna veza.		●
		U modelu poslovnog procesa postoji uvjet u elementu odluke koji ima vrlo malu toleranciju zadovoljenja.		●
12.	Zapinjanje toka	U modelu poslovnog procesa postoji minimalno dva elementa OR spajanja.		●
		U modelu poslovnog procesa jedan element OR spajanja slijedi drugi element OR spajanja.		●
		U modelu poslovnog procesa drugi element OR spajanja ima povratnu vezu s prvim elementom spajanja		●
		U modelu poslovnog procesa postoji minimalno dva elementa AND spajanja.		●
		U modelu poslovnog procesa element OR spajanja slijedi element AND spajanja.		●
		U modelu poslovnog procesa jedan element AND spajanja slijedi drugi element AND spajanja.		●
13.	Zastoj toka	U modelu poslovnog procesa mora postojati jedan element razdvajanja i jedan element AND spajanja.	●	
		U modelu poslovnog procesa element AND spajanja mora slijediti nakon elementa razdvajanja.	●	
	Anomalije pravila			
14.	Redundantna pravila	U modelu poslovnog procesa dva ili više elemenata odluke unutar istog procesa imaju isti uvjet.		●
		U modelu poslovnog procesa dva ili više elemenata odluke istog procesa imaju isti ulaz.		●
		U modelu poslovnog procesa dva ili više elemenata odluke istog procesa imaju iste izlaze.		●
15.	Generalna pravila	U modelu poslovnog procesa dva ili više elemenata odluke unutar procesa i potprocesa imaju uvjete s istim ili detaljnijim zahtjevima.		●
		U modelu poslovnog procesa dva ili više elemenata odluke istog procesa i potprocesa imaju isti ulaz.		●
		U modelu poslovnog procesa dva ili više elemenata odluke istog procesa i potprocesa imaju iste izlaze.		●
16.	Izravna kontradikcija (eng. Direct contradiction)	U modelu poslovnog procesa postoji barem jedan element vrata		●
		U modelu poslovnog procesa navedena vrata imaju uvjet koje je kontradiktorno samo po sebi, dakle bez obzira na ulazne parametre izlaz može biti različit.		●
17.	Kontradikcija u ulazu	U modelu poslovnog procesa postoji jedan element odluke.		●
		U modelu poslovnog procesa navedeni uvjet odluke u svim slučajevima ne zadovoljava		●
18.	Kontradikcija u zaključku	U modelu poslovnog procesa postoje elementi razdvajanja i element stapanja.		●
		U modelu poslovnog procesa nakon tih elemenata slijedi element odluke.		●
		U modelu poslovnog procesa poslovna stavka je ista kroz sve navedene elemente.		●
19.	Zapinjanje temeljeno na pravilima	U modelu poslovnog procesa postoje element OR spajanja.		●
		U modelu poslovnog procesa postoje element AND razdvajanja.		●
		U modelu poslovnog procesa AND razdvajanje slijedi nakon OR spajanja.		●
		U modelu poslovnog procesa AND element razdvajanja ima povratnu vezu na element OR spajanja.		●
20.	Zastoj temeljen na pravilima	U modelu poslovnog procesa postoji element OR razdvajanja.	●	
		U modelu poslovnog procesa postoji element OR spajanja.		●
		U modeli poslovnog procesa element OR razdvajanja prati element OR spajanja.		●
		U modelu poslovnog procesa element OR spajanja povratno je spojen s elementom OR razdvajanja.		●
21.	Izolirana pravila (eng. Isolated rules)	U modelu poslovnog procesa postoje minimalno dva elementa odlučivanja.		●
		U modelu poslovnog procesa drugi element odlučivanja slijedi nakon prvog elementa odlučivanja.		●
		U modelu poslovnog procesa uvjet jednog elementa odlučivanja mora biti sadržan u uvjetu drugog elementa.		●
22.	Beskorisni zaključci (eng. Useless conclusions)	U modelu poslovnog procesa postoji barem jedan element odlučivanja.		●
		U modelu poslovnog procesa u navedenom elementu uvjet odlučivanja mora imati više manjih pravila koji su podskup određenog pravila u uvjetu.		●
23.	Viseći uvjeti (eng. Dangling Conditions)	U modelu poslovnog procesa postoji barem jedan element odlučivanja.		●
		U modelu poslovnog procesa navedeni element odlučivanja ima barem jedan ulaz koji nije spojen poslovnom stavkom za neki drugi element.		●
24.	Nepostojeća pravila	U modelu poslovnog procesa postoji barem jedan element odlučivanja.		●

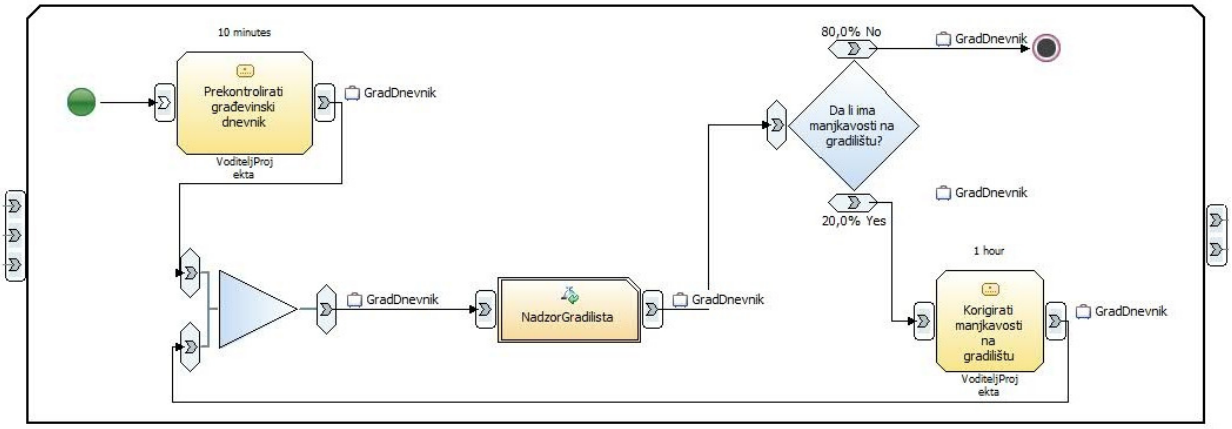
		U modelu poslovnog procesa u navedenom elementu odlučivanja nije definirano pravilo.		●
25.	Redundantni podaci	U modelu poslovnog procesa postoji aktivnost koja upisuje podatke u bazu podataka, no one se ne koriste u daljem toku poslovnog procesa ni u bilo kojem drugom poslovnom procesu organizacije ili izvan nje.		●
		U modelu poslovnog procesa postoji objekt baze podataka u koji se upisuje podaci, no nikad se ne čitaju.		●
26.	Izgubljeni podaci	U modelu poslovnog procesa postoji više aktivnosti koji upisuju, ažuriraju i čitaju podatke po istom dokumentu u istome poslovnom procesu.		●
		U modelu poslovnog procesa postoji aktivnost koja briše određeni podatak i izvodi se prije aktivnosti čitanja podataka.		●
		U modelu poslovnog procesa ne postoji mehanizam provjere vremena čitanja i zapisivanja podatka tijekom rada poslovnog procesa		●
27.	Nepostojeći podaci	U modelu poslovnog procesa postoji barem jedan element baze podataka.	●	
		U modelu poslovnog procesa postoje aktivnosti koje upisuju, ažuriraju i čitaju isti podatak u istoj bazi podataka.	●	
		U modelu poslovnog procesa ne postoji mehanizam provjere postojanja određenog podatka prije njegovog korištenja u poslovnom procesu.	●	
28.	Nekonzistentni podaci	U modelu poslovnog procesa postoji barem jedan element baze podataka.	●	
		U modelu poslovnog procesa postoji više aktivnosti koji upisuju isti podatak u objekt podataka.		●
29	Nikad izbrisani podaci	U modelu poslovnog procesa postoji barem jedan privremeni element baze podataka.	●	
		U modelu poslovnog procesa postoji aktivnost koja privremeno sprema podatke u bazu podataka, a ne postoji aktivnost brisanja podataka iz privremene baze.		●
		U modelu poslovnog procesa postoji aktivnost koja privremeno sprema podatke u bazu podataka, a ne postoji aktivnost čitanja tih podataka onda se radi o anomaliji redundantnih podataka.		●
30.	Dva puta izbrisani podaci	U modelu poslovnog procesa postoji element baze podataka	●	
		U modelu poslovnog procesa postoje barem dvije aktivnosti koje brišu određene slogove iz baze podataka.		●
31.	Podaci neizbrisani na vrijeme	U modelu poslovnog procesa mora postojati barem jedan element baze podataka.	●	
		U modelu poslovnog procesa mora postojati barem jedan element aktivnosti koji briše podatak u bazi podataka.		●
		U modelu poslovnog procesa mora postojati barem jedna aktivnost koja čita podatak.	●	
		U modelu poslovnog procesa postoji aktivnost čitanja koja mora biti poslije nakon aktivnosti brisanja.		●

Tabela 34. Tabela sa redoslijedom i pravilima prepoznavanja ***Anomalija resursa*** za poslovni proces ***Pratiti realizaciju po fazama***

Red.br.	Naziv anomalije	Prepoznavanje anomalije	Da	Ne
	Anomalije obrazaca kreiranja			
1.	Direktna alokacija u slučaju da više resursa ima isti naziv	U modelu procesa postoji više različitih resursa s istim nazivom.		●
2.	Direktna alokacija resursa bez mehanizma eskalacije	U modelu poslovnog procesa postoji aktivnost koje nema određeno maksimalno vrijeme trajanja.	●	
		U modelu poslovnog procesa ne postoji aktivnost koja se aktivira ako istekne maksimalno vrijeme trajanja.	●	
3.	Alokacija temeljena na ulogama kad odabrani resurs nije pripadnik određene uloge/role	U modelu poslovnog procesa postoji aktivnost koji ima pripadajući resurs čija rola nije sukladna zahtjevima određene aktivnosti.		●
4.	Alokacija temeljena na ulogama kad resurs pripada u više od jedne grupe	U modelu poslovnog procesa pojedinim aktivnostima zadani su resursi koji se nalaze u više od jedne role.		●
5.	Alokacija temeljena na ulogama kod grupe sa jednim resursom	U modelu poslovnog procesa postoje aktivnosti koje se izvode paralelno, a koje imaju dodijeljene role sa samo jednim resursom.		●
6.	Automatsko izvođenje bez mehanizma provjere izvođenja	U modelu poslovnog procesa provjerava se da li se je aktivnost već izvršila?		
	Anomalije samopokretajućih obrazaca			
7.	Početak kreiranja sa alokacijom resursa	U modelu poslovnog procesa ne postoji start event, nego se proces okida iz vana.		●
		U modelu poslovnog procesa postoji aktivnost ima vrijednost čekanja resursa (eng. *ResourceWaitTime*) veći o nule.		●
		U modelu poslovnog procesa navedena aktivnost odmah dobiva sve poslovne stavke potrebne za rad.		●
8.	Djelomično lančano izvršenje u poslovnom procesu	U modelu poslovnog procesa postoje više aktivnosti koje se odvijaju sekvencijalno.		●
		U modelu poslovnog procesa ne postoji element razdvajanja i stapanja sekvenci.		●
		U modelu poslovnog procesa svaku od navedenih aktivnosti obavlja različit resurs		●
		U modelu poslovnog procesa navedene aktivnosti ne ovise slijedno jedna od drugoj.		●
		U modelu poslovnog procesa postoje dovoljni kapacitet resursa da se aktivnosti odvijaju paralelno.		●
	Anomalije višestrukih obrazaca resursa			
9.	Istovremeno izvođenje kod ograničenih kapaciteta resursa	U modelu poslovnog procesa postoji više od jedne aktivnosti.		●
		U modelu poslovnog procesa dva ili više aktivnosti odvijaju se paralelno od strane istih resursa.		●
		U modelu poslovnog procesa resursi nemaju dovoljno kapaciteta za paralelni rad.		●

Primjer modela poslovnog procesa *Nadzirati gradilište*

Model poslovnog procesa ***Nadzirati gradilište*** je potproces koji se poziva iz procesa ***Pratiti realizaciju u fazama***. Kao što se može vidjeti na slici modela ovaj model poslovnog procesa je u slobodnoj formi, dakle nema staza.



Slika 128. ***Nadzirati gradilište*** u slobodnoj formi

Tabela 35. Detalji poslovnog procesa ***Nadzirati gradilište***

Nadzirati gradilište			**Trajanje [JMj]**
Proces ***Nadzirati gradilište*** je poslovni proces koji se ponavlja tako dugo dok se ne isprave manjkavosti na gradilištu. Ovaj proces se izvodi u poslovnom procesu ***Pratiti realizaciju po fazama***, a slijede ga servisi u kojem se ovjerava *Građevinska knjiga* i *Građevinski dnevnik*. Ulaz u ovaj proces su *Građevinska knjiga*, *Projektna dokumentacija* i *Zahtjev*, Dok su izlazi iz ovog procesa *Građevinska knjiga* i *Građevinski dnevnik*.			3 h 50 '
Radno mjesto /Org. Jed.	**Naziv radnih koraka**	**Opis radnih koraka**	
Voditelj projekta	*Prekontrolirati građevinski dnevnik*	**Nadzor** kontrolira *Građevinsku knjigu*.	10 '
Nadzor	**Nadzor gradilišta**	**Nadzor** kontrolira *Građevinsku knjigu*.	
	Da li ima manjkavosti na gradilištu?	Provjerava se da li postoje manjkavosti na gradilištu.	
Voditelj projekta	*Korigirati manjkavosti na gradilištu*	Korigiraju se manjkavosti na gradilištu.	1 h

Analizom navedenom modela poslovnog procesa prepoznata je jedna ***Strukturna anomalija*** i to anomalija ***Zapinjanje temeljeno na pravilima*** iz razloga što u modelu poslovnog procesa postoje element OR spajanja, zatim u modelu poslovnog procesa postoje element AND razdvajanja, osim toga u modelu poslovnog procesa AND razdvajanje slijedi nakon OR spajanja i postoji iz AND elementa razdvajanja povratna veza na OR element spajanja. Osim toga prepoznata je i jedna ***Anomalija resursa*** i to anomalija ***Direktna alokacija resursa bez mehanizma eskalacije***, jer u modelu poslovnog procesa postoji aktivnost koje nema određeno maksimalno vrijeme trajanja i u modelu poslovnog procesa ne postoji aktivnost koja se aktivira ako istekne maksimalno vrijeme trajanja.

Tabela 36. Tabela sa redoslijedom i pravilima prepoznavanja ***Anomalija prikaza modela*** za proces ***Pratiti realizaciju po fazama***

Red.br.	Naziv anomalije	Pravila prepoznavanja	Da	Ne
	Anomalije organizacije percepcije			
	Anomalije jednostavnosti			
1.	Previše elemenata u modelu	Model poslovnog procesa sadrži ukupni broj elemenata veći od 9.		●
2.	Neselektivni prikaz elemenata u modelu	Model poslovnog procesa sadrži aktivnosti koje ne spadaju u isti kontekst procesa.		●
		Model poslovnog procesa prikazuje sadržaje razmjene koji nisu bitni za prikaz modela poslovnog procesa.		●
		Model poslovnog procesa koristi resurse koji nisu bitni za prikaz modela poslovnog procesa.		●
	Anomalije grupiranja			
3.	Nekompaktni i razvučen model	Model poslovnog procesa ima više od jednog elementa.	●	
		Model sadrži grafički element kategorije koji se horizontalno ili vertikalno nalazi između dva spojena elementa iste kategorije.		●
		Model sadrži uvjet koji govori o tome da se neki element nalazi previše udaljen od drugog elementa.		●
	Anomalije kontinuiranosti			
4.	Križanje i savijanje veza između elemenata modela	Model poslovnog procesa ima više od jednog elementa.		●
		U modelu poslovnog procesa povezani elementi iste kategorije nisu na istoj visini.		●
		Element kontrole toka siječe se sa drugim elementom kontrole toka.		●
	Anomalije prisnosti			
5.	Nepoštivanje pozicije hijerarhije modela	Model poslovnog procesa ima više od jednog elemenata iste kategorije.		●
		U modelu poslovnog procesa postoji element koji je spojen s drugim elementom iste kategorije, a jedan je iznad drugoga i u istoj mjeri udaljen od ruba modela.		●
	Anomalije sličnosti			
6.	Slični elementi nisu grupirani zajedno	Model procesa ima više od jednog elementa.	●	
		U modelu poslovnog procesa postoje elementi čije kategorija objekta ima obilježje sličnog objekta.		●
		U modelu poslovnog procesa broj sličnih objekata u modelu procesa je veći od 1.		●
7.	Slični elementi nemaju iste boje	Model poslovnog procesa ima više od jednog elementa.	●	
		U modelu poslovnog procesa postoje grafički elementi iste kategorije koji imaju različite boje.		●
8.	Slični elementi nemaju istu veličinu	Model poslovnog procesa ima više od jednog elementa.	●	
		U modelu poslovnog postoji element iste kategorije, a da li mu je širina i visina dvostruko veća od prethodnog elementa.		●
9.	Slični elementi nemaju isti oblik	U modelu poslovnog procesa ima više od jednog elementa.	●	
		U modelu poslovnog postoji element iste kategorije, a da li mu je širina ili visina dvostruko veća od prethodnog elementa.		●
	Anomalije povezanosti			
10.	Preklapanje elemenata modela	Model poslovnog procesa ima više od jednog elementa.	●	
		U modelu poslovnog procesa postoji element čija je lijeva pozicije i širina manja od lijeve pozicije i širine sljedećeg elementa.		●
		U modelu poslovnog procesa postoji element čija je gornja pozicija i visina manja od gornje pozicije i visine sljedećeg elementa.		●
	Anomalije percepcije razlikovanja			
	Anomalije obrisa			
11.	Elementi modela se međusobno dodiruju	Model poslovnog procesa sadrži više od jednog elementa.	●	
		U modelu poslovnog procesa postoji grafički element u skupu objekata grafičkih elemenata, koji bez obzira na kategoriju ima poziciju koja zbrojena sa dimenzijom kontrole i oduzeta od pozicije susjednog elementa zbrojene s dimenzijom kontrole daje iznos manju od širine grafičke kontrole.		●
		U modelu poslovnog procesa elementi se međusobno dodiruju i vertikalno i horizontalno.		●
	Anomalije orijentacije			
12.	Modeli orijentirani s desna na lijevo	Model poslovnog procesa ima više od jednog elementa	●	
		U modelu poslovnog procesa postoji tok između dva grafička elementa gdje je slijedni grafički element lijevo od promatranog		●
13.	Modeli orijentirani od dolje prema gore	Model poslovnog procesa sadrži više od jednog elementa,	●	
		U modelu poslovnog procesa postoji element kod kojih postoji kontrola toka, a kod kojih je prethodni element niže od slijednog elementa.		●
14.	Vertikalni nazivi elemenata	U modelu poslovnog procesa postoji grafička kontrola iste kategorije koja ima obilježje orijentacija naslova a da je vertikalna.		●
15.	Vertikalni opisi unutar modela	U modelu poslovnog procesa postoji element koji ima vertikalni opis, tada takav model procesa ima anomaliju vertikalni opis unutar modela.		●
	Anomalije simetrije			
16.	Necentrirani i nesimetrični model	U modelu poslovnog procesa broj elementa koji se nalaze u lijevom horizontalnom kvadrantu različit je od broj elemenata koji se nalaze u desnom horizontalnom kvadrantu.		●
		U modelu poslovnog procesa broj elementa koji se nalaze u gornjem vertikalnom kvadrantu različit je od broj elemenata koji se nalaze u donjem vertikalnom ..	●	

Tabela 37. Tabela sa redoslijedom i pravilima prepoznavanja ***Sintaktičkih anomalija*** za model poslovnog procesa ***Pratiti realizaciju po fazama***

Red.br.	Naziv anomalije	Pravila prepoznavanja	Da	Ne
	Nepravilna uporaba spremnika			
1.	Bazen bez elemenata	U modelu poslovnog procesa postoji bazen u kojem ne postoji ni jedan grafički element.		●
2.	Nepravilna uporaba bazena	U modelu poslovnog procesa postoji bazen ima tok sekvence koji prelazi granice spremnika.		●
3.	Staza bez elemenata	U modelu poslovnog procesa postoji staza bez elemenata.		●
4.	Nepravilna uporaba staza	U modelu poslovnog procesa postoji staza no ne postoji i bazen koji sadrži navedenu stazu.		●
	Nepravilna uporaba objekata toka			
5.	Više početnih događaja u modelu	U modelu poslovnog procesa postoji više od jednog početnog događaja.		●
6.	Nepovezanost početnog događaja s ostalim elementima modela	U modelu poslovnog procesa početni događaj nema izlaznu sekvencu.		●
7.	Početni događaj ima više od jedne izlazne sekvence	U modelu poslovnog procesa mora postojati barem jedan početni događaj.	●	
		Početni događaj ima više od jedne izlazne sekvence.		●
8.	Nepostojanje završnog događaja	U modelu poslovnog procesa ne postoji završni događaj.		●
9.	Nepovezanost završnog događaja s ostalim elementima modela	U modelu poslovnog procesa ulazna sekvenca završnog događaja je prazan skup.		●
10.	Završni događaj ima više od jedne ulazne sekvence	U modelu poslovnog procesa mora biti barem jedan završni događaj.	●	
		Završni događaj ne smije imati više od jedne ulazne sekvence.		●
11.	Nepostojanje početnog događaja	U modelu poslovnog procesa ne postoji ulazna sekvenca.	●	
		U modelu poslovnog procesa broj početnih događaja jednak je nuli.		●
	Nepravilna uporaba veza			
12.	Ulazna sekvenca sa početnim događajem	U modelu poslovnog procesa početni događaj ima ulaznu sekvencu.		●
13.	Izlazna sekvenca s završnim događajem	U modelu poslovnog procesa završni događaj ima izlaznu sekvencu.		●
	Nepravilna uporaba toka poruka			
14.	Početni događaj poruke vezan je tokom poruke s elementom početnog događaja	U modelu poslovnog procesa početni događaj poruke vezan je tokom poruke sa elementom početnog događaja.		●
15.	Početni događaj poruke vezan je tokom poruke s elementom bazena	U modelu poslovnog procesa početni događaj poruke vezan je tokom poruke sa elementom bazena.		●
16.	Početni događaj poruke vezan je tokom poruke s elementom potprocesa	U modelu poslovnog procesa početni događaj poruke vezan je tokom poruke sa elementom potprocesa.		●
17.	Početni događaj poruke vezan je tokom poruke s elementom aktivnosti	U modelu poslovnog procesa početni događaj poruke vezan je tokom poruke sa elementom aktivnosti.		●
18.	Početni događaj vezan je tokom poruke s elementom stop događaja poruke	U modelu poslovnog procesa početni događaj poruke vezan je tokom poruke sa elementom aktivnosti.		●
19.	Početni događaj vezan je tokom poruke s elementom završnog događaja poruke	U modelu poslovnog procesa početni događaj poruke vezan je tokom poruke sa elementom završnog događaja poruke.		●
20.	Početni događaj poruke vezan je tokom poruke s završnim događajem	U modelu poslovnog procesa početni događaj poruke vezan je tokom poruke sa elementom završnog događaja.		●
21.	Bazen je vezan tokom poruke s završnim događajem	U modelu poslovnog procesa bazen je vezan tokom poruke sa elementom završnog događaja.		●
22.	Potproces je vezan tokom poruke s završnim događajem	U modelu poslovnog procesa potproces vezan je tokom poruke sa elementom završnog događaja.		●
23.	Aktivnost je vezana tokom poruke s završnim događajem	U modelu poslovnog procesa aktivnost je vezana tokom poruke sa elementom završnog događaja.		●
24.	Stop događaj poruke je vezan tokom poruke s završnim događajem	U modelu poslovnog procesa stop događaj poruke vezan je tokom poruke s elementom završnog događaja.		●
25.	Element završnog događaja poruke vezan je tokom poruke s elementom završnog događaja	U modelu poslovnog procesa element završnog događanja poruke vezan je tokom poruke sa elementom završnog događaja.		●

Tabela 38. Tabela sa redoslijedom i pravilima prepoznavanja ***Semantičkih anomalija*** za model poslovnog procesa ***Pratiti realizaciju po fazama***

Red.br.	Naziv anomalije	Pravila prepoznavanja	Da	Ne
	Nepravilno nazivlje elemenata			
1.	Neimenovanje elemenata modela	U modelu poslovnog procesa postoji grafički element kojem je naziv prazan		●
2.	Nejedinstvenost nazivlja elemenata modela	U modelu poslovnog procesa postoji grafički element kojem naziv nije jedinstven.		●
3.	Nekonzistentnost nazivlja elemenata modela	U modelu poslovnog procesa naziv procesa i potprocesa nisu u nominativu.		●
		U modelu poslovnog procesa postoje aktivnosti kojima naziv ima više riječi i više glagola.		●
		U modelu poslovnog procesa postoje aktivnosti kojima naziv nema glagol.		●
		U modelu poslovnog procesa postoji baza podataka kojoj naziv nije imenica.		●
4.	Sinonimi u nazivlju elemenata modela	Za svaki element u pojedinoj kategoriji treba za svaki naziv pronaći sinonime.		●
		Svaku listu tih sinonima treba usporediti sa svim nazivima svih elemenata iz pojedine kategorije.		●
		Korisniku dati obavijest o parovima mogućih sinonima koji se pojavljuju u modelu.		●
5.	Homonimi u nazivlju elemenata modela	Unutar svake kategorije pronaći više elemenata istog imena		●
		Ako takvi postoje onda provjeriti da li su im nazivi u tablici homonima.		●
		Ako takvi postoje onda provjeri da li ima iste ulaze i izlaze.		●
	Nepravilne vrijednosti atributa			
6.	Ne postoji vrijednost atributa	U modelu poslovnog procesa postoje elementi koji imaju atributa koji nemaju nikakvu vrijednost.		●
7.	Nejedinstvenost vrijednosti atributa	U modelu poslovnog procesa postoje elementi gdje vrijednost atributa nije jedinstvena unutar modela poslovnog procesa.		●
8.	Neispravan tip podataka vrijednosti atributa	U modelu poslovnog procesa postoje elementi gdje tip podataka atributa nije ispravan.		●
		U modelu poslovnog procesa postoje elementi gdje vrijednost atributa nije unutar zadanih granica.		●
9.	Nedozvoljeni znakovi u vrijednosti atributa	U modelu poslovnog procesa postoje elementi gdje u vrijednostima atributa postoje nedozvoljeni znakovi.		●

Tabela 39. Tabela sa redoslijedom i pravilima prepoznavanja ***Strukturnih anomalija*** za poslovni proces ***Pratiti realizaciju po fazama***

Red.br.	Naziv anomalije	Pravila prepoznavanja	Da	Ne
	Kontrola toka			
1.	Nepotrebna kontrola toka	U modelu poslovnog procesa postoji aktivnost koja ima više od jednog izlaza s istom poslovnom stavkom.		●
		U modelu poslovnog procesa postoji aktivnost koja ima više od jednog ulaza s istom poslovnom stavkom.		●
		Navedene poslovne stavke su po atributima potpuno jednake.		●
		Navedene poslovne stavke imaju istu jedinstvenu oznaku (ID).		●
		Navedene poslovne stavke su nastale u istoj aktivnosti.		●
		Navedene poslovne stavke su nastale u isto vrijeme.		●
		Navedene poslovne stavke su nastale od strane istog resursa.		●
2.	Redundancija modela	U modelu poslovnog procesa u jednoj stazi postoji više istih aktivnosti koji obavljaju isti posao.		●
		U modelu poslovnog procesa u jednoj stazi postoji niz aktivnosti kojeg obavljaju isti resursi više puta.		●
3.	Nedefinirani uvjeti vrata	U modelu poslovnog procesa postoji barem jedan element odlučivanja.	●	
		U modelu poslovnog procesa postoji element odlučivanja koji nema definiran uvjet.		●
		U modelu poslovnog procesa ne postoji početna vrijednost uvjeta.		●
4.	Kontradiktorni uvjeti vrata	U modelu poslovnog procesa postoji barem jedna vrata.	●	
		U modelu poslovnog procesa svi uvjeti navedenih vrata odvojeni su veznikom AND uvijek su istiniti ili neistiniti.		●
5.	Deterministički nedostatak sinkronizacije	U modelu poslovnog procesa postoje elementi razdvajanja i XOR element spajanja	●	
		U modelu poslovnog procesa postoji element razdvajanja prethodi XOR element spajanja.		●
6.	Nedeterministički nedostatak sinkronizacije	U modelu poslovnog procesa ima barem jedan element OR razdvajanja		●
		U modelu poslovnog procesa ima barem jedan element XOR spajanja	●	
		U modelu poslovnog procesa OR element razdvajanja prethodi XOR elementu spajanja		●
7.	Viseća stanja	U modelu poslovnog procesa postoji element koji ima izlaz iz elementa koji nije spojen na drugi element niti ne izlazi iz modela, niti ne završava završnim elementom.		●
		U modelu poslovnog procesa postoji element koji ima ulaz u element koji nije spojen na drugi element niti ne predstavlja ulaz u model.		●
8.	Netolerancija na promjene poretka	U modelu poslovnog procesa postoje minimalno dvije aktivnosti koje slijede jedna drugu.		●
		Navedene aktivnosti imaju određeno vrijeme trajanja.		●

		Navedene aktivnosti obavljaju različiti međusobno nezavisni resursi.		●
		U modelu poslovnog procesa poslovne stavke navedenih aktivnosti su jedinstvene.		●
9.	Nekonzistentnost kod inserta paralelnih grana	U modelu poslovnog procesa postoje minimalno dvije aktivnosti koje se odvijaju paralelno.		●
		U modelu poslovnog procesa navedene aktivnosti imaju određeno vrijeme trajanja.		●
		U modelu poslovnog procesa navedene aktivnosti obavljaju isti resursi.		●
10.	Nekonzistentnost promjene iz prošlosti	U modelu poslovnog procesa postoji ima barem jedan element baze podataka.		●
		U modelu poslovnog procesa u navedenu bazu podataka u jednom koraku se piše, a u drugom se čita isti slog.		●
		U modelu poslovnog procesa navedena aktivnost koja čita slijedi aktivnost koja zapisuje slog.		●
11.	Nedostatak tolerancije u petlji	U modelu poslovnog procesa poslije elementa odluke postoji povratna veza.	●	
		U modelu poslovnog procesa postoji uvjet u elementu odluke koji ima vrlo malu toleranciju zadovoljenja.		●
12.	Zapinjanje toka	U modelu poslovnog procesa postoji minimalno dva elementa OR spajanja.		●
		U modelu poslovnog procesa jedan element OR spajanja slijedi drugi element OR spajanja.		●
		U modelu poslovnog procesa drugi element OR spajanja ima povratnu vezu s prvim elementom spajanja		●
		U modelu poslovnog procesa postoji minimalno dva elementa AND spajanja.		●
		U modelu poslovnog procesa element OR spajanja slijedi element AND spajanja.		●
		U modelu poslovnog procesa jedan element AND spajanja slijedi drugi element AND spajanja.		●
13.	Zastoj toka	U modelu poslovnog procesa mora postojati jedan element razdvajanja i jedan element AND spajanja.		●
		U modelu poslovnog procesa element AND spajanja mora slijediti nakon elementa razdvajanja.		●
	Anomalije pravila			
14.	Redundantna pravila	U modelu poslovnog procesa dva ili više elemenata odluke unutar istog procesa imaju isti uvjet.		●
		U modelu poslovnog procesa dva ili više elemenata odluke istog procesa imaju isti ulaz.		●
		U modelu poslovnog procesa dva ili više elemenata odluke istog procesa imaju iste izlaze.		●
15.	Generalna pravila	U modelu poslovnog procesa dva ili više elemenata odluke unutar procesa i potprocesa imaju uvjete s istim ili detaljnijim zahtjevima.		●
		U modelu poslovnog procesa dva ili više elemenata odluke istog procesa i potprocesa imaju isti ulaz.		●
		U modelu poslovnog procesa dva ili više elemenata odluke istog procesa i potprocesa imaju iste izlaze.		●
16.	Izravna kontradikcija (eng. Direct contradiction)	U modelu poslovnog procesa postoji barem jedan element vrata	●	
		U modelu poslovnog procesa navedena vrata imaju uvjet koje je kontradiktorno samo po sebi, dakle bez obzira na ulazne parametre izlaz može biti različit.		●
17.	Kontradikcija u ulazu	U modelu poslovnog procesa postoji jedan element odluke.	●	
		U modelu poslovnog procesa navedeni uvjet odluke u svim slučajevima ne zadovoljava		●
18.	Kontradikcija u zaključku	U modelu poslovnog procesa postoje elementi razdvajanja i element stapanja.	●	
		U modelu poslovnog procesa nakon tih elemenata slijedi element odluke.		●
		U modelu poslovnog procesa poslovna stavka je ista kroz sve navedene elemente.		●
19.	Zapinjanje temeljeno na pravilima	U modelu poslovnog procesa postoje element OR spajanja.	●	
		U modelu poslovnog procesa postoje element AND razdvajanja.	●	
		U modelu poslovnog procesa AND razdvajanje slijedi nakon OR spajanja.	●	
		U modelu poslovnog procesa AND element razdvajanja ima povratnu vezu na element OR spajanja.	●	
20.	Zastoj temeljen na pravilima	U modelu poslovnog procesa postoji element OR razdvajanja.		●
		U modelu poslovnog procesa postoji element OR spajanja.	●	
		U modeli poslovnog procesa element OR razdvajanja prati element OR spajanja.		●
		U modelu poslovnog procesa element OR spajanja povratno je spojen s elementom OR razdvajanja.		●
21.	Izolirana pravila (eng. Isolated rules)	U modelu poslovnog procesa postoje minimalno dva elementa odlučivanja.		●
		U modelu poslovnog procesa drugi element odlučivanja slijedi nakon prvog elementa odlučivanja.		●
		U modelu poslovnog procesa uvjet jednog elementa odlučivanja mora biti sadržan u uvjetu drugog elementa.		●
22.	Beskorisni zaključci (eng. Useless conclusions)	U modelu poslovnog procesa postoji barem jedan element odlučivanja.	●	
		U modelu poslovnog procesa u navedenom elementu uvjet odlučivanja mora imati više manjih pravila koji su podskup određenog pravila u uvjetu.		●
23.	Viseći uvjeti (eng. Dangling Conditions)	U modelu poslovnog procesa postoji barem jedan element odlučivanja.	●	
		U modelu poslovnog procesa navedeni element odlučivanja ima barem jedan ulaz koji nije spojen poslovnom stavkom za neki drugi element.		●
24.	Nepostojeća pravila	U modelu poslovnog procesa postoji barem jedan element odlučivanja.	●	
		U modelu poslovnog procesa u navedenom elementu odlučivanja nije definirano pravilo.		●
25.	Redundantni podaci	U modelu poslovnog procesa postoji aktivnost koja upisuje podatke u bazu podataka, no one se ne koriste u daljem toku poslovnog procesa ni u bilo kojem drugom		●

		poslovnom procesu organizacije ili izvan nje.		
		U modelu poslovnog procesa postoji objekt baze podataka u koji se upisuje podaci, no nikad se ne čitaju.		●
26.	Izgubljeni podaci	U modelu poslovnog procesa postoji više aktivnosti koji upisuju, ažuriraju i čitaju podatke po istom dokumentu u istome poslovnom procesu.		●
		U modelu poslovnog procesa postoji aktivnost koja briše određeni podatak i izvodi se prije aktivnosti čitanja podataka.		●
		U modelu poslovnog procesa ne postoji mehanizam provjere vremena čitanja i zapisivanja podatka tijekom rada poslovnog procesa.		●
27.	Nepostojeći podaci	U modelu poslovnog procesa postoji barem jedan element baze podataka.		●
		U modelu poslovnog procesa postoje aktivnosti koje upisuju, ažuriraju i čitaju isti podatak u istoj bazi podataka.		●
		U modelu poslovnog procesa ne postoji mehanizam provjere postojanja određenog podatka prije njegovog korištenja u poslovnom procesu.		●
28.	Nekonzistentni podaci	U modelu poslovnog procesa postoji barem jedan element baze podataka.		●
		U modelu poslovnog procesa postoji više aktivnosti koji upisuju isti podatak u objekt podataka.		●
29	Nikad izbrisani podaci	U modelu poslovnog procesa postoji barem jedan privremeni element baze podataka.		●
		U modelu poslovnog procesa postoji aktivnost koja privremeno sprema podatke u bazu podataka, a ne postoji aktivnost brisanja podataka iz privremene baze.		●
		U modelu poslovnog procesa postoji aktivnost koja privremeno sprema podatke u bazu podataka, a ne postoji aktivnost čitanja tih podataka onda se radi o anomaliji redundantnih podataka.		●
30.	Dva puta izbrisani podaci	U modelu poslovnog procesa postoji element baze podataka		●
		U modelu poslovnog procesa postoje barem dvije aktivnosti koje brišu određene slogove iz baze podataka.		●
31.	Podaci neizbrisani na vrijeme	U modelu poslovnog procesa mora postojati barem jedan element baze podataka.		●
		U modelu poslovnog procesa mora postojati barem jedan element aktivnosti koji briše podatak u bazi podataka.		●
		U modelu poslovnog procesa mora postojati barem jedna aktivnost koja čita podatak.		●
		U modelu poslovnog procesa postoji aktivnost čitanja koja mora biti poslije nakon aktivnosti brisanja.		●

Tabela 40. Tabela sa redoslijedom i pravilima prepoznavanja ***Anomalija resursa*** za poslovni proces ***Pratiti realizaciju po fazama***

Red.br.	Naziv anomalije	Prepoznavanje anomalije	Da	Ne
	Anomalije obrazaca kreiranja			
1.	Direktna alokacija u slučaju da više resursa ima isti naziv	U modelu procesa postoji više različitih resursa s istim nazivom.		●
2.	Direktna alokacija resursa bez mehanizma eskalacije	U modelu poslovnog procesa postoji aktivnost koje nema određeno maksimalno vrijeme trajanja.	●	
		U modelu poslovnog procesa ne postoji aktivnost koja se aktivira ako istekne maksimalno vrijeme trajanja.	●	
3.	Alokacija temeljena na ulogama kad odabrani resurs nije pripadnik određene uloge/role	U modelu poslovnog procesa postoji aktivnost koji ima pripadajući resurs čija rola nije sukladna zahtjevima određene aktivnosti.		●
4.	Alokacija temeljena na ulogama kad resurs pripada u više od jedne grupe	U modelu poslovnog procesa pojedinim aktivnostima zadani su resursi koji se nalaze u više od jedne role.		●
5.	Alokacija temeljena na ulogama kod grupe sa jednim resursom	U modelu poslovnog procesa postoje aktivnosti koje se izvode paralelno, a koje imaju dodijeljene role sa samo jednim resursom.		●
6.	Automatsko izvođenje bez mehanizma provjere izvođenja	U modelu poslovnog procesa provjerava se da li se je aktivnost već izvršila?		●
	Anomalije samopokretajućih obrazaca			
7.	Početak kreiranja sa alokacijom resursa	U modelu poslovnog procesa ne postoji start event, nego se proces okida iz vana.		●
		U modelu poslovnog procesa postoji aktivnost ima vrijednost čekanja resursa (eng. *ResourceWaitTime*) veći o nule.		●
		U modelu poslovnog procesa navedena aktivnost odmah dobiva sve poslovne stavke potrebne za rad.		●
8.	Djelomično lančano izvršenje u poslovnom procesu	U modelu poslovnog procesa postoje više aktivnosti koje se odvijaju sekvencijalno.		●
		U modelu poslovnog procesa ne postoji element razdvajanja i stapanja sekvenci.		●
		U modelu poslovnog procesa svaku od navedenih aktivnosti obavlja različit resurs		●
		U modelu poslovnog procesa navedene aktivnosti ne ovise slijedno jedna od drugoj.		●
		U modelu poslovnog procesa postoje dovoljni kapacitet resursa da se aktivnosti odvijaju paralelno.		●
	Anomalije višestrukih obrazaca resursa			
9.	Istovremeno izvođenje kod ograničenih kapaciteta resursa	U modelu poslovnog procesa postoji više od jedne aktivnosti.		●
		U modelu poslovnog procesa dva ili više aktivnosti odvijaju se paralelno od strane istih resursa.		●
		U modelu poslovnog procesa resursi nemaju dovoljno kapaciteta za paralelni rad.		●

6. ZAKLJUČAK

U prethodnim poglavljima razrađena je taksonomija anomalija po određenim zajedničkim karakteristikama, razrađeni su pojedini uzorci anomalija pomoću kojih je moguće prepoznati i razvrstati pojedinu anomaliju čime se mogu poduzeti određeni koraci njihovog rješavanja u modelima poslovnih procesa. Za svaku od obrađenih anomalija razrađen je pripadajući uzorak sa pravilima prepoznavanja, primjerom i koracima rješavanja.

U poglavlju gdje su prikazani primjeri prepoznavanja i rješavanja anomalija u referentnom izravno je potvrđeno da je predložena taksonomija koja se temelji na uzorcima dizajna dovoljno praktična da omogući prepoznavanje i razvrstavanje svih anomalija u referentnom skupu modela poslovnih procesa. Osim toga modeli korišteni u primjerima su modeli provjereni u renomiranim alatima za modeliranje poslovnih procesa koji su se već po kvaliteti i upotrebljivosti potvrdili u praksi (IBM Websphere), predložen pristup se je i u tako provjerenim modelima pokazao dovoljno dobrim da pronađe potencijalne slabe točke pojedinih modela. Samim time se je predložen pristup prepoznavanja i rješavanja anomalija pokazao potencijalno vrlo zanimljivim za daljnju razradu i uporabu u alatima za modeliranje poslovnih procesa. Ovime je potvrđena hipoteza H1 koja glasi:

H1: Nova taksonomija anomalija, koja se temelji na uzorcima dizajna, omogućava prepoznavanje i razvrstavanje svih anomalija na referentnom skupu modela poslovnih procesa.

U poglavlju pod nazivom *Smjernice automatizacije prepoznavanja i rješavanja anomalija u modelima poslovnih procesa* razrađen je mehanizam prepoznavanja i rješavanja anomalija prema redoslijedu da jedan uzorak prepoznavanja i rješavanja minimalno utječe na prethodni obrazac anomalije. U navedenom mehanizmu vrlo bitnu ulogu igra redoslijed rješavanja pojedinih anomalija. Nepravilni redoslijed može prouzročiti pojavu anomalija koje su prethodno riješene u prethodnim koracima, te se pravilnim redoslijedom rješavanja izbjegavaju takve pojave. Redoslijed koji se je pokazao dobrim u prvom koraku povjerava ***Anomalije prikaza modela***, navedena kategorija anomalije i njezini ispravci ne utječu na strukturu, odnose i na kraju krajeva svojstva pojedinih elemenata, već se koncentrira na njihove pozicije u modelu i prema drugim elementima modela. Nakon ***Anomalija prikaza modela*** slijedi provjera ***Sintaktičkih anomalija***. Navedene anomalije su obično najbolje

razrađene u suvremenim alatima za modeliranje poslovnih procesa. U odabranim primjerima ni jedan model nije imao ***Sintaktičku anomaliju***. Osim toga ti alati imaju i drugačiji pristup rješavaju navedenih anomalija, a koje se obično temelje na Petrijevim mrežama i sličnim formalnijim metodama. Nakon ***Sintaktičkih anomalija***, predloženi poredak rješavanja anomalija nastavlja se sa ***Semantičkim anomalijama***. U ovoj kategoriji anomalija postupci prepoznavanja i rješavanja anomalija bave se pretežno nazivljima i vrijednostima elemenata i njihovih atributa. Sve navedene kategorije anomalija moraju biti prepoznate i provjerene prije nego se krene u prepoznavanje i rješavanje ***Strukturnih anomalija*** koje su više kategorije, jer u obzir uzimaju međusobne odnose među elementima i njihove uloge u samome modelu. Navedene anomalije su prije svega statičkog karaktera. Što znači da se mogu otkriti i riješiti prije postupka simulacije. No, postoje i dinamičke anomalije koje ulaze u sferu ***Anomalija resursa*** i ***Vremenskih anomalija***. U ovome radu obrađeno je nekoliko ***Anomalija resursa*** koje imaju statički karakter, no postoji cijeli niz anomalija koje su dinamičke prirode i koje je nemoguće otkriti bez simulacija. Također, u ovome radu nisu obrađene ni ***Vremenske anomalije*** koje su izrazito dinamičke prirode. Navedene anomalije mogle bi se analizirati tek nakon što su prepoznate i riješene prethodno opisane anomalije niže kategorije. Iz svega navedenog, a i na temelju prethodnog poglavlja u kojemu je primjerom prikazan način prepoznavanja i rješavanja anomalija može se smatrati da je potvrđena H2 hipoteza, koja glasi:

H2: Redoslijed provjere uzoraka anomalija u modelima poslovnih procesa utječe na uspješnost i učinkovitost postupaka otklanjanja anomalija.

Osim navedenih hipoteza u ovome radu posebna je pozornost usmjerena na postizanje već unaprijed definiranih ciljeva istraživanja. Ciljevi istraživanja su:

- istraživanje i sistematizacija literature o anomalijama u procesnim modelima,
- definiranje osnovnih pojmova vezanih uz anomalije procesnih modela,
- razrada taksonomije anomalija na temelju metode koja se temelji na teoriji uzoraka dizajna,
- određivanje pravila prepoznavanja i otklanjanja anomalija i
- prijedlog koncepta automatizacije prepoznavanja i otklanjanja anomalija.

Prva točka ciljeva istraživanja bavi se literaturom, svaki od u radu prikazanih obrazaca anomalija obrađen je na način da je navedena i literatura u kojima se anomalija izvorno

pojavljuje ili u kojoj su ponuđeni koraci prepoznavanja i rješavanja anomalije. Razrada anomalija u detalje nalazi se u poglavlju *Taksonomija anomalija u modelima poslovnih procesa.* Literatura je posebno obrađena u poglavlju pod nazivom *Pregled dosadašnjih istraživanja i istraživačkih problema* u kojima je dat pregled literature i autora koji su se bavili istraživanjem navedene problematike.

Drugi cilj istraživanja bavi se definiranjem osnovnih pojmova vezanih uz anomalije modela procesa. Poglavlje koje se posebno bavi temeljnim definicijama osnovnih pojmova nosi naziv *Područje, predmeti i objekti istraživanja*. U navedenom poglavlju odgovara se na pitanja, kao što su: Koje je područje koje se istraživanje bavi? Što je razvoj temeljen na modelima? Što je modeliranje? Što je model? Što su to funkcionalni modeli? Što je modeliranje poslovnih procesa? Što predstavljaju anomalije u modelima poslovnih procesa? Za svako od navedenih pitanja dat je odgovor uz referentnu literaturu i autore čije su definicije spomenute u radu.

Treći cilj istraživanja odnosi se na razradu taksonomije anomalija. Prvo je u poglavlju pod nazivom *Metodologija oblikovanja taksonomije* opisana metodologija razrade taksonomije, a zatim je u poglavlju *Taksonomija anomalija u modelima poslovnih procesa* u iznesen prijedlog taksonomije anomalija modela poslovnih procesa koji je podijeljen na ***Anomalije prikaza modela***, ***Sintaktičke anomalije***, ***Semantičke anomalije***, ***Strukturne anomalije*** i ***Anomalije resursa***.

Četvrti cilj istraživanja u ovome radu je određivanje pravila prepoznavanja i otklanjanje anomalija, razrađen je kroz cijeli niz poglavlja. U prvome redu opisane su metode opisa prepoznavanja i rješavanja anomalija u poglavlju *Metode prepoznavanja i rješavanja anomalija u modelu poslovnih procesa*, gdje je ukratko opisan metamodel BPMN 2.0 notacije, MOF 2.0 standard, OCL i Pseudokod. Na temelju navedenih metoda u poglavlju *Taksonomija anomalija u modelima poslovnih procesa* svaka pojedinačna anomalija je opisana na koji način se prepoznaje i kako je riješiti.

U poglavlju pod nazivom *Smjernice automatizacije prepoznavanja i rješavanja anomalija u modelima poslovnih procesa* iznesen je prijedlog rješavanja anomalija koji se temelji na uzorcima anomalija. U poglavlju je razrađen redoslijed prepoznavanja i rješavanja anomalija. Dok je u poglavlju *Primjer prepoznavanja i rješavanja anomalija u referentnom skupu modela poslovnih procesa* dokazano da bi navedena metodologija mogla dati prilično dobre rezultate.

Kao što je već opisano u prethodnim poglavljima, ovdje prikazani način prepoznavanja i rješavanja anomalija u modelima poslovnih procesa moguće je unaprijediti

daljnjom razgradnjom taksonomije anomalija, zatim unaprijediti pravila i procedure prepoznavanja i rješavanja anomalija. S obzirom na testne rezultate navedenog pristupa u rješavanju anomalija ovaj pristup bi potencijalno mogao dati vrlo dobre rezultate.

Mjesta za proširenje ima i u horizontalnim i u vertikalnom smjeru.

Pod horizontalnim proširenjem smatra se proširivanje skupa anomalija koje se mogu prepoznati i riješiti. Kao što je prethodno navedeno, u ovome radu prikazana su isključivo statičke anomalije koje je moguće prepoznati prije faze simuliranja modela. No, ***Anomalije resursa*** i ***Anomalije vremena*** ulaze u područja koje je teško otkriti prije provedbi simulacija. Pa bi svakako skup ovako razrađenih anomalija trebalo proširiti za te dvije kategorije. Dakle, razrada anomalija viših razina složenosti. ***Anomalije vremena*** su složene anomalije koje nastaju na višoj hijerarhiji složenosti, gdje dolazi do preklapanja vremena zauzeća pojedinih aktivnosti i resursa. Osim ***Anomalija vremena*** trebalo bi detaljnije razraditi i ***Anomalije resursa***. To su anomalije vezane uz uska i široka grla modela, zatim zagušenja i čekanja pojedinih aktivnosti unutar modela, i tako dalje. Kod toga posebno treba obratiti pažnju kod anomalija koje nastaju kod višestrukih instanci pojedinih procesa. U tom slučaju moguće je dodatno opterećenje pojedinih aktivnosti, a samim time i resursa što može dovesti do smanjivanja performansi procesa. Osim toga proučavajući literaturu potrebnu za ovaj rad, naiđeno je na cijeli niz radova koje se bave proširenjem BPMN notacije. Ovisno o svojim poljima interesa pojedini autori nadodali su određene elemente koje bi olakšale modeliranje određenih scenarija koje dolaze iz realnog života.

Osim navedenog horizontalnog proširenja, daljnja istraživanja morala bi biti usmjerena i u vertikalni smjer. Dakle produbiti preciznost pojedinih prepoznavanja anomalija. Nadodati pravila prepoznavanja kako bi razradili daleko preciznije postupke detekcije anomalija. Nadalje, osim taksonomije, svaki od pronađenih obrazaca anomalija može biti dodatno detaljnije opisan s načinima rješavanja na nekoliko različitih pristupa. Kod anomalija kod kojih nisu ponuđena rješenja, već će korisniku samo daje obavijest o prepoznatoj anomaliji, može se korisniku ponuditi rješavanje po unaprijed definiranim koracima u kojima bi korisnik u par koraka interakcije ispravio anomaliju po principima najbolje prakse. Takvi pristupi ispravljanju grešaka bili bi dobrodošli u slučajevima anomalija više razine, gdje rješenje zahtjeva intenzivniju korekciju modela. Posebno bi trebalo istražiti prigodne algoritme za prepoznavanje sinonima i homonima u nazivima elemenata modela, zatim pojačati OCL pravila ograničavanja u dostupnim metamodelima i poboljšati pravila pisana u pseudokodu. Vertikalno proširenje ima na kraju samo jedan cilj, a to je dovesti ovdje predstavljanu metodologiju na razinu pogodnu na ugradnju u alate za modeliranje poslovnih

procesa. To bi dovelo do daleko boljih alata, koji bi već tijekom modeliranja, dakle prije faze simulacija mogli korisnika obavijestiti o slabim točkama unutar modela.

Dosadašnji pristupi rješavanja anomalija koji su opisani u dostupnoj literaturi većinom se je temeljio na uporabi formalnih metoda poput Petrijevih mreža ili proširenih Petrijevih mreža. No, navedeni pristup ograničio se je na prepoznavanje određenih ***Sintaktičkih anomalija*** i nekih ***Strukturnih anomalija***. U ovome radu odabran je drugačiji pristup, za kojeg se je pretpostavilo da je bolji.

Istražene su anomalije u modelima poslovnih procesa te njihove karakteristike i posebnosti. Za svaku anomaliju formiran je obrazac koji sadrži opis, primjer, način rješavanja i način prepoznavanja anomalije. Na temelju tih podataka uočene su određene sličnosti između pojedinih anomalija, navedene sličnosti i obilježja omogčile su njihovo grupiranje i kategoriziranje. Na temelju navedenih karakteristika razrađena je taksonomija kategorija anomalija. Klasifikacijom anomalija u funkcionalnim modelima dobivena je mogućnost grupiranja mehanizama njihovog prepoznavanja i rješavanja.

Navedenim pristupom razrađene su pet temeljnih kategorija anomalija i to; ***Anomalije prikaza modela***, ***Sintaktičke anomalije***, ***Semantičke anomalije***, ***Strukturne anomalije*** i ***Anomalije resursa***. Ukupni broj prepoznatih anomalija u ovome radu doveden je do brojke od 90 anomalija smještene u navedenih pet kategorija. Ovdje valja napomenuti da ovo samo vrh ledene sante, jer su u obzir uzete samo statičke anomalije, dok postoji cijeli niz anomalija dinamičke prirode, kao što su ***Anomalije vremena*** i dobar dio ***Anomalija resursa***.

Osim toga uočena je potreba za definiranjem redoslijeda rješavanja anomalija, što dovodi do razrađivanja smjernica automatizacije prepoznavanja i rješavanja anomalija u modelima poslovnih procesa. Na kraju naveden pristup ispitan je na nekoliko primjera referentnih modela. Opisani pristup pokazao se je potencijalno jako dobrim, jer je u već postojećim modelima koje su postojeći alati smatrali ispravnima, pronašao određene anomalije koje su samo potvrdile slabe točke pojedinih modela. Naravno, za ozbiljniji pristup navedeni skup anomalija i navedene metode njihovog otkrivanja i rješavanja treba dodatno proširiti i razraditi. Krajnji cilj je ugradnja opisanih koncepata u alate za modeliranje poslovnih procesa.

Tomislav Vidačić

LITERATURA

[1] Aalst, W.M.P.: *Verification of Workflow Nets*, Proceedings of the 18th International Conference on Application and Theory of Petri Nets, p.407-426, June 23-27, 1997.

[2] Ambler, T.:Marketing and the Bottom Line: Marketing Metrics to Pump Up Cash Flow., Prentice Hall, New Jersey, 2003.

[3] An, Y., Hu, X. , Song I.Y.: *Round-Trip Engineering for Maintaining Conceptual-Relational Mappings*, Proceedings 20th International Conference, CAiSE 2008 Montpellier, France, June 16-20, 2008. Advanced Information Systems Engineering, pp 296-311. Springer Berlin Heidelberg

[4] *ArgoUML*, version 0.34, 2011.,<argouml-users.net/ubdex.php>, (10.5.2014.)

[5] Awad, A., Decker, G., Lohmann, N.: *Diagnosing and Repairing Data Anomalies in Process Models*, Business Process Management Workshops, BPM 2009 International Workshops Ulm, Germany, September 2009, Springer-Verlag Berlin Heidelberg 2010. str 5-16.

[6] Awad, A., Grosskopf , A., Meyer, A., Weske, M.: *Enabling Resource Assignment Constraints in BPMN*, Hasso Plattner Institute, Potsdam, Germany, 2008.

[7] Bailey, K. D.: *A Three-Level Measurement Model. Quality and Quantity,* Kluwer Academic Publishers vol. 18,University of California, Los Angeles, CA, U.S.A, pp. 225-245, 1984.

[8] *Business Process Model and Notation (BPMN) 2.0*, Object Management Group, Tech. Rep. January, 2011.

[9] Bitsaki, M., Danylevych, O., van den Heuvel, W. J., Koutras, G. D., Leymann, F., Mancioppi, M., Nikolaou, C. N., Papazoglou, M. P.: *Model Transformations to Leverage Service Networks*; ICSOC workshop proceedings, WESOA 2008., Springer, März 2009.

[10] BPMN Specifications, 10.10.2008., <http://www.omg.org/technology/documents/spec_summary.htm>, (10.10.2008.)

[11] Breton, E., Bézivin, J.: *An Overview of Industrial Process Meta-Models*, 13th international conference software and system engineering and their applications, ICSSEA 2000-14, Paris, France, 2000.

[12] Celms, E., Kalnins, A., Lace, L.: *Diagram definition facilities based on metamodel mappings,* Invited talk at the Third OOPSLA Workshop on Domain-Specific Modeling, October 2003.

[13] Chandola, V., Banerjee, A., Kumar, V.: *Anomaly Detection : A Survey*, ACM Computing Surveys, Vol. 41(3), Article 15, July 2009.

[14] Diskin, Z., Dingeu, J.: *A metamodel independent framework for model transformation: Towards generic model management patterns in reverse engineering,* Proc. 3rd Int.Workshop on Metamodels, Schemas, Grammas and Ontologies for reverse engineering, ATEM-2006.

[15] Dijkman, R. M., Dumas, M., Ouyang, C.: *Formal semantics and analysis of BPMN process models*, Technical Report Preprint 7115, Queensland University of Technology, 2007. <https://eprints.qut.edu.au/archive/00007115>

[16] Dohring, M., Heublein, S.: *Anomalies in Rule-Adopted Workflows – A Taxonomy and solutuions for vBPMN*, 6th European Conference on Software Maintenance and Reengineering, pages 117-126. Szeged, Hungary : IEEE.

[17] Dohring, M., Zimmermann, B.: *vBPMN: Event-Aware Workflow Variants by Weaving BPMN2 and Business Rules*, EMM-SAD. London: Springer, 2011, pp. 332–341.

[18] Dumas, M.: *Case study: BPMN to BPEL model transformation.* 2009., <http://is.ieis.tue.nl/staff/pvgorp/events/grabats2009/cases/grabats2009synthesis.pdf> (2010-07-11)

[19] *Eclipse Model Development Tools* (MDT), 2014., <http://www.eclipse.org/modeling/mdt/?project=ocl>, (1.5.2014.)

[20] Eessaar, E.: *Using metamodeling in order to evaluate data models*, Proceeding, AIKED'07 Proceedings of the 6th Conference on 6th WSEAS Int. Conf. on Artificial Intelligence, Knowledge Engineering and Data Bases - Volume 6, Pages 181-186, 2007.

[21] Emerson, M. J., Sztipanovits, J.: *Implementing a MOF-Based Metamodeling Environment Using Graph Transformations*, 4TH OOPSLA WORKSHOP ON DOMAIN-SPECIFIC MODELING, Pages: 83-92, 2004.

[22] Goldstein, B.: *Sensation and Perception - With CD,* 8th edition, Wadsworth, Inc., Belmont, CA, 2009.

[23] Hausmann,, J.H., Kent, S.: *Metamodeling Relations – Relating metamodels. submitted to UML*, 2003., June 2003

[24] Hammoudi, S., de Souza, J., Lopes G., Bontempo, A.: *Metamodel Matching: Experiments and Comparison*, Software Engineering Advances, International Conference, IEEE, Tahiti, 2006.

[25] Hettel, T., *Model Round-Trip Engineering,* PhD Thesis, Discipline of Computer Science, Faculty of Science and Technology, Queensland University od Technology, 2010., <http://eprints.qut.edu.au/32082/1/Thomas_Hettel_Thesis.pdf>

[26] Hidaka, S., *On the traceability in a graph roundtrip transformation system GRoundTram,* National Institute od Informatics, Workshop on Bidirectional Transformation, 2010. < http://www.biglab.org/4th-Btrans/slides/SoichiroHidaka.pdf >

[27] Jakubowski, A.: *Discovering synonyms with use of frequent itemsets and association rules*, Warsaw University Of Technology, <http://home.elka.pw.edu.pl/~ajakubo2/adam_jakubowski/AdamJakubowskiArtykul.pdf>, (1.5.2014.)

[28] Johansson, L. O., Wärja, M., Carlsson, S.: *An evaluation of business process model techniques, using Moody's quality criterion for a good diagram*, CEUR Workshop Proceedings, < http://ceur-ws.org/Vol-963/paper5.pdf >, (10.9.2014.)

[29] Kim, G.W., Lee, J.H., Soon, J.H.: *Classification and analyses of business process anomalies*, Communication Software and Networks, 2009. ICCSN '09. International Conference on Date of Conference: 27-28 Feb. 2009, p. 433 – 437.

[30] Kretschmer, P.: *JWT metamodel compared to BPMN metamodel*, <http://wiki.eclipse.org/JWT_Metamodel>, (09/2014)

[31] Larson, A. J., Navathe, B. S., Elmasri, R.: *A theory of attribute equivalence in database with application to shema integration*, Transaction on Software Engineering, 15(4), str 449-463, April 1989.

[32] Laue, R. , Awad, A.: *Visualization of Business Process Modeling Anti Patterns*, ECEASST, vol. 25, 2010.

[33] Liu, R., Kumar, A.: *An analysis and taxonomy of unstructured workfows*, BPM'05. Springer, 2005, pp. 268–28.

[34] Lohmann, N., Verbeek, E., Dijkman, R.: *Petri Net Transformations for Business Processes, A Survey;*- Transactions on Petri Nets and Other Models of Concurrency II, Pages 46 - 63, Springer-Verlag Berlin, Heidelberg 2009.

[35] Medin, D.L., Goldstone,R.L., Gentner, D: *Similarity involving attributes and relations*: Judgments of similarity and difference are not inverses, Psychological Science, 1(1):64–69., 1990.

[36] Miller, J., Mukerji, J.: *MDA Guide Version 1.0.1.*, Object Management Group, 2003.

[37] *MOF, Version 2.0*, Release Date: January 2006, (1.9.2014.), <http://www.omg.org/spec/MOF/2.0/>, (1.9.2014)

[38] Morrison, J.H.: *How to create effective taxonomy*, 18.8.2004, <http://www.zdnet.com/how-to-create-effective-taxonomy-2039190441/>

[39] Mork, P., Bernstein, P.A., Melnik, S.: *Translating Schemas and Data between Metamodels*, Microsoft, 2007.

[40] Morville, P., Rosenfeld, L.: *Information Architecture for the World Wide Web: Designing Large-Scale Web Sites*, 3nd edition, O'Reilly Media, 2006.

[41] Nickerson, R., Varshney, U., Isaac, H.: *Taxomomy development in information systems: developing a taxonomy of mobile applications*, ECIS 2009: 1138-1149.

[42] Nguyen, T. A., Perkins, W. A., Laffey, T. J., Pecora, D.: *Knowledge verification base*, AI Magazine, vol. 8, no. 2, pp. 69–75, 1987.

[43] Object Management Group (OMG), *Business Process Modeling Notation Specification*, 01.12.2007., <http:www.bpmi.org> , (01.12.2007.)

[44] *Object Constraint Language*, Version 2.3.1, Release Date: January 2012., (1.9.2014.), <http://www.omg.org/spec/OCL/2.3.1/>, (1.9.2014.)

[45] *Object Constraint Language Environment*, version 2.0, (2005), <lci.cs.ubbcluj.ro/ocle>, (20.5.2014.)

[46] Olkhovich, L.: *Semi-Automatic Business Process Performance Optimization Based On Redundant Control Flow Detection*, AICT-ICIW'06, pp. 146–146, 2006.

[47] Onoda, S., Ikkai, Y., Kobayashi, T. , Komoda, N.: *Definition of deadlock patterns for business processes workflow models*, HICSS'99. IEEE Computer Society, 1999, p. 11.

[48] Overbeek, J.,F.: *Meta Object Facility (MOF) investigation of the state of the art*, Software Engineering, Electrical Engineering, Mathematics and Computer Science, University of Twente, 2006.

[49] Preece, A. D., Shinghal, R.: *Foundation and application of knowledge base verification*, International Journal of Intelligent Systems, vol. 9, no. 8, pp. 683–701, 1994.

[50] Rinderle, S., Reichert, M., Dadam, P. : *Correctness Criteria for Dynamic Changes in Workflow Systems: A Survey*, DKE, vol. 50, no. 1, pp. 9–34, 2004.

[51] Russell, N., Hofstede, A.H.M., Edmond, D., Aalst, W.M.P. van der: *Workflow Resource Patterns*, <http://www.workflowpatterns.com/patterns/resource/>, WP 127, Eindhoven University of Technology, Eindhoven, 2004. (1.5.2014)

[52] Sadiq, W., Orlowska, M. E.: *Applying graph reduction techniques for identifying structural conflicts in process models*, CAiSE'99. London and UK: Springer, 1999.

[53] Sadiq, S., Orlowska, M., Sadiq, W., Foulger, C.: *Data flow and validation in workflow modelling*, ADC'04, vol. 27. Dunedin, New Zealand: ACS Inc., 2004, pp. 207–214. Seidewitz, E., What Models Mean, IEEE Computer Society, 2003.

[54] Seidewitz, E.: *What Models Mean*, IEEE Software, vol.20, no. 5, pp. 26-32, September/October 2003

[55] Silingas, D., Mileviciene, E.: *Business Analyst Articles: Business Analysis & Systems Analysis*, (1.10.2013.) <http://www.modernanalyst.com/Resources/Articles/tabid/115/articleType/ArticleView/articleId/2438/Efficient-BPMN-from-AntiPatterns-to-Best-Practices.asp>

[56] Solti-Rogge, A., Kunze, M., Awad A., Weske M.: *Bussiness process configuration wizard and consistency checker for BPMN 2.0.* 12th International Conference, BPMDS 2011 and 16 th International Conference, EMMSAD 2011 herld at CAiSE 2011, London, UK, June 2011, Proceedings, Springer-Verlag, Berlin Heidelberg 2011. str. 231-245.

[57] Sun, S.X, Zhao, J.L., Nunamaker, J.F i Sheng, O.R.L.: *Formulating the data-flow perpective for business process management*, Info.Sys. Research 17(4),374-291 (2006)

[58] Suwa, M., Scott, A. C., Sortliffe, E. H. Sh.: *An approach to verifying completeness and consistency in a rule-based expert system*, Stanford University, Tech. Rep., 1982.

[59] Šešić, B.: O*snovi metodologije društvenih nauka*, naučna knjiga, Beograd, 1982.

[60] Teng, H., Chen, K., Lu, S.: *Adaptive real-time anomaly detection using inductively generated sequential patterns*, Proceedings of IEEE Computer Society Symposium on Re-search in Security and Privacy. IEEE Computer Society Press, 1990. str. 278-284.

[61] *The Rules of BPMN*, 8.4.2013. <http://brsilver.com/the-rules-of-bpmn/> , (8.4.2013.)

[62] Trcka, N., Sidorova, N., van der Aalst, W. M. P.: *Data-Flow Anti-Patterns: Discovering Dataflow Errors in Workflows*, CAiSE'09. Amsterdam: Springer, 2009, pp. 425-439.

[63] *Unified Modeling Language*, Version 2.0, 2011., <http://www.omg.org/spec/UML/>, (1.5.2014.)

[64] Vidačić, T.: *Referentni model procesa visokogradnje uz intenzivnu ICT potporu*, FOI magistarski rad, 2009.

[65] Vidačić, T.: Strahonja, V.: *Taxonomy of anomalies in business process models*, 22nd International Conference on Information Systems Development (ISD2013). Seville, 2013. (predavanje,međunarodna recenzija,objavljeni rad,znanstveni).

[66] Vitolins, V., Kalnins, A.: *Semantics of UML 2.0 Activity Diagram for Business Modeling by Means of Virtual Machine*, Proceeding EDOC '05 Proceedings of the Ninth IEEE International EDOC Enterprise Computing Conference Pages 181-194, IEEE Computer Society Washington, DC, USA 2005.

[67] Wang, C.H., Wang, F.,J.: *Detecting artifact anomalies in business process specifications with a formal model*, Journal of Systems and Software archive, Volume 82 Issue 10, October, 2009, Pages 1600-1619

[68] Zaharov N.N.: *Tehničko normiranje procesa rada*, Panorama Zagreb, 1964., str 12.

[69] Zaidi, A. K., Levis, A. H.: *Validation & verification of decision making rules*, Automatica, vol. 33, no. 2, pp. 155–169, 1997.

[70] Zelenika, R.: *Metodologija i tehnologija izrade znanstvenog i stručnog djela*, Ekonomski fakultet u Rijeci, Rijeka, 2000.

[71] Zhang, D., Nguyen, D.: *Prepare: a tool for knowledge base verification*, TKDE, vol. 6, no. 6, pp. 983–989, 1994.

Životopis

Autor je rođen 29.12.1973. u Varaždinu. Osnovnu školu završio je u Varaždinu. 1988. godine upisuje Srednjoškolski centar za obrazovanje metalskih i elektrotehničkih kadrova Varaždin - smjer elektronika automatika. Nakon srednje škole, 1992. godine upisuje Fakultet Organizacije i Informatike u Varaždinu, gdje 1998. godine i diplomira. Na istom fakultetu magistrirao je 2009. godine na temi *Referentni model procesa visokogradnje uz intenzivnu ICT potporu*, pod mentorstvom prof.dr.sc. Josipa Brumeca. 2009. godine upisao je doktorski studij na Fakultetu organizacije i informatike, Varaždin gdje pod mentorstvom prof.dr.sc. Vjerana Strahonje piše disertaciju pod naslovom *Prepoznavanje i rješavanje anomalija u modelima procesa uporabom uzoraka*. Tijekom svoje karijere radio je na nizu informatičkih projekata iz područja kemijske industrije, logistike, građevine, medicine, bankarstva i javne uprave. Radi u Hrvatskoj poljoprivrednoj agenciji u Odjelu za informacijsko komunikacijske tehnologije. Oženjen je i živi u Varaždinu.

Važniji radovi:

1. Vidačić T., Strahonja V.; Taxonomy of anomalies in business process models, 22nd International Conference on Information Systems Development, (ISD2013), Seville, Spain, Springer, Berlin - Heidelberg, September, 2013.
2. Gerić, S.; Vidačić, T., XML Digital Signature and its Role in Information System Security // 35th International Convention MIPRO - proceedings. 2012. 1844-1849
3. Danić, D., Vidačić, T., Mijoč, I.; *Referentni model procesa kartičnog poslovanja u računovodstveno-informacijskom okruženju*, Ekonomska misao i praksa, Sveučilište u Dubrovniku, No.2., 12.2013.
4. Vidačić, T.; Gerić, S., Sigurnost u servisno orijentiranoj arhitekturi, CASE 23., Rijeka, 2011.
5. Vidačić, T.; Gerić, S., Servisne mješavine - što su one zapravo?, CASE 22., Rijeka, 2010.
6. Vidačić, T.; Keretić, M.; Dizajn korisničkog sučelja - put do zadovoljnog korisnika, CASE 17, Rijeka, 2005.