

RĪGAS TEHNISKĀ UNIVERSITĀTE

**PROMOCIJAS
DARBS**

2012

RĪGAS TEHNISKĀ UNIVERSITĀTE

Datorzinātnes un informācijas tehnoloģijas fakultāte

Lietišķo datorsistēmu institūts

Ieva Zeltmate

Doktora studiju programmas „Datorsistēmas” doktorante

**INTELEKTUĀLAS SISTĒMAS
IZSTRĀDE SAREŽĢĪTU SISTĒMU
STRUKTŪRMODELĒŠANAI**

Promocijas darbs

Zinātniskais vadītājs

Dr. habil. sc. ing., profesors

J. Grundspenķis

Rīga 2012



Šis darbs izstrādāts ar Eiropas Sociālā fonda atbalstu projektā „Atbalsts RTU doktora studiju īstenošanai”.

ANOTĀCIJA

Attīstoties zinātnei un informācijas komunikāciju tehnoloģijām, apstākļu kopums, kas ietekmē dažādu nozaru izpēti, attīstību un inovāciju, kļūst arvien sarežģītāks. Līdz ar to ir aktuāla sarežģītu sistēmu izpēte un analīze, kas ir būtisks nosacījums, lai šādas sistēmas izprastu, pārvaldītu un pilnveidotu. Tomēr sarežģītu sistēmu specifika un par tām pieejamā informācija ir kritēriji, kas ierobežo atbilstošas modelēšanas pieejas un rīka izvēli. Struktūrmodelēšanas pieeja ļauj īstenot sarežģītu sistēmu struktūras modelēšanu, bet līdz šim tā nebija implementēta piemērotā datorsistēmā.

Promocijas darbs ir veltīts struktūrmodelēšanas pieejas pilnveidošanai un intelektuālas sistēmas sarežģītu sistēmu struktūrmodelēšanai izstrādei un realizācijai. Darbā ir analizētas sarežģītas un intelektuālas sistēmas, freimi, kā arī detalizēti aplūkota struktūrmodelēšanas pieeja, ar mērķi izstrādāt intelektuālas sistēmas arhitektūru. Ir pilnveidota struktūrmodelēšanas pieejas sintakse un semantika un izstrādāta zināšanu izgūšanas un atspoguļošanas shēma freimu kopa. Shēma, implementēta intelektuālā sistēmā I4S, ļauj īstenot zināšanu izgūšanu un atspoguļošanu no eksperta un piemērot to struktūrmodelēšanas mērķiem. Promocijas darba ietvaros izstrādātā sistēma I4S ir praktiski pārbaudīta, veicot sarežģītas tehniskas sistēmas robots AGR8 struktūrmodelēšanu un struktūras analīzi. Par promocijas darba galvenajiem rezultātiem ir ziņots četrās starptautiskās konferencēs un tie ir publicēti piecos LZP atzītos izdevumos.

Darbs sastāv no ievada, 4 nodaļām un secinājumiem. Darba pamattekstā ir 168 lappuses, tajā skaitā 102 attēli un 153 nosaukumu literatūras saraksts un darbam pievienoti 3 pielikumi.

ANNOTATION

The development of information and communication technologies creates increasingly complex set of circumstances that influence research, development and innovation in various industries. Complex system research and analysis have become significant, because it is an essential condition to understand, manage and improve these kinds of systems. However, the nature of complex systems and available information are the criteria, which limit the choice of appropriate modelling approach and tool. Structural modelling approach allows to perform complex system structure modelling, but until now it was not implemented in a suitable computer system.

The thesis is devoted to the improvement of structural modelling approach and development of an intelligent system for complex system structure modelling. Complex and intelligent systems, frames have been analysed and structural modelling approach reviewed to develop architecture of the intelligent system. Structural modelling syntax and semantics have been improved and knowledge acquisition and representation schema frame set developed. The schema have been implemented in the intelligent system I4S, that allows the acquisition and representation of expert knowledge and to apply it for structural modelling purposes. The developed system I4S have been practically verified performing structural modelling and analysis for complex technical system robot AGR8.

The thesis consists of introduction, four chapters and conclusions. The main part of the dissertation contains 168 pages, 102 figures and 153 bibliographical sources, additionally the thesis contains 3 appendices.

SATURA RĀDĪTĀJS

IEVADS	11
1. SAREŽĢĪTAS SISTĒMAS UN STRUKTŪRMODELĒŠANA.....	18
1.1. Problēmsfēras jēdzieni un to interpretācija	18
1.1.1. Sistēma un ar to saistītie jēdzieni	19
1.1.2. Sarežģītība un ar to saistītie jēdzieni	21
1.1.3. Sarežģītas sistēmas un ar tām saistītie jēdzieni	26
1.2. Sarežģītu sistēmu modelēšana.....	30
1.2.1. Sarežģītas sistēmas un modelēšana	31
1.2.2. Sistēmu struktūrmodelēšanas pieeja.....	38
1.3. Intelektuālas sistēmas un to īpašības	44
1.3.1. Intelektuālas sistēmas uzbūve	46
1.3.2. Intelektuālas sistēmas darbības mehānismi	48
1. nodaļas kopsavilkums un secinājumi	50
2. STRUKTŪRMODELĒŠANAS PIEEJAS APRAKSTS.....	52
2.1. Struktūras modeļu sintakse un semantika	53
2.1.1. Morfoloģiskās struktūras modeļa apraksts	53
2.1.2. Funkcionālās struktūras modeļu apraksts	61
2.2. Transformācijas starp struktūras modeļiem	68
2.2.1. Transformācija no morfoloģiskās struktūras modeļa uz funkcionālās struktūras modeli funkciju telpā.....	69
2.2.2. Plūsmu kombināciju varianti.....	72
2.2.3. Transformācija no morfoloģiskās struktūras modeļa uz funkcionālās struktūras modeli uzvedības un parametru telpā.....	78
2. nodaļas kopsavilkums un secinājumi	84
3. FREIMU KOPAS LIETOJUMS STRUKTŪRMODELĒŠANĀ UN SISTĒMAS I4S DARBĪBAS PRINCIPI	86
3.1. Freimu kopas apraksts un lietojums	87
3.1.1. Freima jēdziens un tā struktūra	87
3.1.2. Freimu kopas apraksts	92
3.2. Sistēma I4S un transformācijas no freimu kopas	103
3.2.1. Sistēmas I4S arhitektūra un funkcionalitāte	104
3.2.2. Transformācija no freimu kopas uz struktūras modeļiem	108
3.2.3. Transformācija no freimu kopas uz notikumu kokiem.....	117
3. nodaļas kopsavilkums un secinājumi	119
4. PRAKTISKA PIEMĒRA REALIZĀCIJA SISTĒMĀ I4S.....	121
4.1. Zināšanu iegūšana un atspoguļošana sistēmā I4S	122
4.1.1. Robota AGR8 daļu un to saistību atspoguļojums.....	122
4.1.2. Robota AGR8 uzvedības un parametru atspoguļojums.....	126
4.1.3. Papildus iespējas sistēmā I4S	129
4.2. Robota AGR8 struktūrmodelēšana un analīze ar I4S.....	133
4.2.1. Struktūras modeļu izveide.....	133
4.2.2. Sistēmas struktūras analīze.....	143
4. nodaļas kopsavilkums un secinājumi	152
5. REZULTĀTI UN SECINĀJUMI	153
LITERATŪRAS SARAKSTS	156

PAMATTEKSTA ATTĒLU SARAKSTS

1.1. att. Cynefin ietvarstruktūra [SNO 2010].....	23
1.2. att. Struktūras – uzvedības modelis [APP 2011].....	25
1.3. att. Saistības un struktūras.....	33
1.4. att. Dati, informācija, zināšanas	46
2.1. att. Objekta atspoguļojums.....	54
2.2. att. Izejas transformēšana ieejā	55
2.3. att. Kontaktu atspoguļojums.....	57
2.4. att. Plūsmas atspoguļojums	57
2.5. att. Kontaktu un paralēlu plūsmu atspoguļojums.....	58
2.6. att. Loģiskā operatoru atspoguļojums struktūras modeļos.....	59
2.7. att. MSM atspoguļojums diagrammas veidā.....	60
2.8. att. MSM atspoguļojums grafa veidā	60
2.9. att. Funkcijas atspoguļojums.....	61
2.10. att. Cēloņseku saites atspoguļojums.....	61
2.11. att. FSM FT atspoguļojums.....	62
2.12. att. Uzvedības stāvokļu atspoguļojums.....	63
2.13. att. FSM UT atspoguļojums	64
2.14. att. Parametru kopas un parametru atspoguļojums	65
2.15. att. Defekta atspoguļojums.....	67
2.16. att. FSM PT atspoguļojums.....	67
2.17. att. Virsotņu grafs un loku grafi	68
2.18. att. Transformācijā lietotais MSM piemērs.....	69
2.19. att. FSM FT virsotņu iegūšanas piemērs.....	70
2.20. att. Virsotņu savienošanas gadījumi.....	70
2.21. att. Transformācijas rezultātā iegūtais FSM FT	70
2.22. att. Jaunajā transformācijā lietotais MSM piemērs	71
2.23. att. Jaunās transformācijas rezultātā iegūtais FSM FT	72
2.24. att. MSM un atbilstošais FSM FT (a) gadījums.....	73
2.25. att. MSM un FSM FT (b), (c), (d), (e) gadījumi	74
2.26. att. MSM un atbilstošie FSM FT (f), (g), (h), (i) gadījumi	75

2.27. att. MSM un atbilstošais FSM FT (j) un (k) gadījumi	76
2.28. att. MSM un atbilstošais FSM FT (l) un (m) gadījumi	77
2.29. att. MSM un atbilstošais FSM FT (n) gadījums.....	78
2.30. att. MSM un atbilstošais FSM FT (o) gadījums.....	78
2.31. att. MSM uz FSM UT (uzvedība) virsotnes.....	79
2.32. att. MSM uz FSM UT (uzvedības stāvokļi) virsotnes.....	80
2.33. att. Transformācijas rezultātā iegūtais FSM UT (uzvedība)	81
2.34. att. Transformācijas rezultātā iegūtais FSM UT (uzvedības stāvokļi).....	82
2.35. att. Transformācijas no FSM UT uz FSM PT piemērs	83
2.36. att. Transformācijas piemērs FSM PT	83
3.1. att. Zināšanu atspoguļošanas shēma freims	88
3.2. att. Slotu vērtību saraksts.....	89
3.3. att. Ar norādi saistīti freimi	90
3.4. att. Mantoti un lokāli sloti	91
3.5. att. Freimu kopas	93
3.6. att. Freimu kopa un freimu sistēma.....	94
3.7. att. Saistības freimu sistēmā.....	94
3.8. att. Klases un objektu hierarhijas	95
3.9. att. Freimu hierarhija un klases freims	96
3.10. att. Funkciju hierarhija un kontaktu freims	98
3.11. att. Freimu hierarhija un īpašību freims	100
3.12. att. Alternatīvu freims	100
3.13. att. Procedūru freims	101
3.14. att. Uzvedības freims.....	102
3.15. att. I4S arhitektūra.....	104
3.16. att. Freimu sistēmas hierarhija un klases, kontaktu freimi.....	109
3.17. att. Objektu un saistību saraksti transformācijai uz MSM	110
3.18. att. MSM ar kontaktiem atspoguļojums sistēmā I4S	111
3.19. att. Saistību saraksta transformācijai uz FSM FT piemērs.....	112
3.20. att. FSM FT atspoguļojums I4S	113
3.21. att. Saistību saraksta transformācijai uz FSM UT piemērs	114
3.22. att. FSM UT atspoguļojums I4S	115
3.23. att. Saistību saraksta transformācijai uz FSM PT piemērs.....	117
3.24. att. FSM PT atspoguļojums I4S	117

3.25. att. Loģiskā operatoru atspoguļojums notikumu kokā	118
3.26. att. Notikumu koka atspoguļojums I4S	119
4.1. att. Freima pievienošana un freimu hierarhija sistēmai <i>robots AGR8</i>	123
4.2. att. Ieejas izejas kontaktu attēlošana I4S	124
4.3. att. Ieejas un izejas kontaktu attēlošana I4S	125
4.4. att. Plūsmas, funkcijas un uzvedības stāvokļu identifikatori	125
4.5. att. Plūsmas slotu labošana	126
4.6. att. Uzvedības slotu atspoguļojums	127
4.7. att. Parametru pievienošana I4S	128
4.8. att. Parametru saišu pievienošana	128
4.9. att. Objekta priekšteča labošana	130
4.10. att. Objektu nosaukumu labošana	130
4.11. att. Sistēmas daļu dzēšana	131
4.12. att. Robota AGR8 atspoguļojums freimu hierarhijā	131
4.13. att. Kontaktu freima un funkciju hierarhijas atspoguļojums	132
4.14. att. Īpašību freima un īpašību hierarhijas atspoguļojums	132
4.15. att. MSM (tikai objekti) atspoguļojums	134
4.16. att. MSM (ar kontaktiem) attēlojums	135
4.17. att. MSM (ar uzvedības stāvokļiem) atspoguļojums	136
4.18. att. FSM (funkciju telpā) attēlojums	136
4.19. att. FSM (uzvedības telpā - uzvedība) atspoguļojums	137
4.20. att. FSM (uzvedības telpā – uzvedības stāvokļi) attēlojums	138
4.21. att. FSM (parametru telpā-parametru kopas) atspoguļojums	138
4.22. att. FSM (parametru telpā-parametri) atspoguļojums	139
4.23. att. MSM atspoguļojums pirms izmaiņām	139
4.24. att. MSM atspoguļojums pēc izmaiņām	140
4.25. att. Sistēmas robots AGR8 uzbūve	141
4.26. att. Funkcionālā struktūra sistēmā robots AGR8	141
4.27. att. Blakusvirsotņu matricas atspoguļojums	144
4.28. att. Elementu sadalījuma atspoguļojums	145
4.29. att. Sasniedzamības matricas atspoguļojums	145
4.30. att. Elementu strukturizācija pa līmeņiem	146
4.31. att. Elementu rangi	147
4.32. att. Attālumu matrica	149

4.33. att. Ceļu saraksts starp virsotnēm.....	149
4.34. att. Notikumu koka atspoguļojums	150
4.35. att. Produkciju likumu tabula	151
4.36. att. Sistēmas pārbaudes tabula	151

IEVADS

Sistēmas, ar ko mūsdienās cilvēki sastopas ikdienā, raksturo sarežģīta struktūra un darbības principi. Strādājot ar sistēmām, kā arī pētot un risinot ar tām saistītās situācijas, cilvēkiem pastāvīgi ir nepieciešams izdarīt izvēli un pieņemt lēmumus. Sarežģītība kļūst aktuāla brīdī, kad ir jāizprot sarežģītas sistēmas [HOR 1995, VIC 2002, ASH 2004, JOH 2009], lai veiktu adekvātu lēmumu pieņemšanu. Ar izpratni tiek domāta situācija, kad cilvēkam ir pieejams sistēmas modelis (iespējams domās), kurā ir skaidra sistēmas struktūra un funkcionēšanas mehānismi, kas ļauj loģiski un objektīvi spriest par sistēmu [SIM 1987, ASH 2007, SOK 2010]. Augot sistēmu sarežģītībai, arvien aktuālāka kļūst sistēmu un ar tām saistīto problēmu un situāciju iezīmju izpēte un likumsakarību analīze, kas ir būtiska lēmumu pieņemšanas procesā [CAR 2004]. Analīze tiek veikta, balstoties uz indivīda zināšanām, pieredzi un izpratnes spējām, tomēr cilvēka spējas uztvert un izprast ir ierobežotas, jo vidēji indivīds vienlaicīgi var apskatīt līdz septiņām neatkarīgām informācijas daļām [BAR 1997, YOU 2007]. Ja sistēmas sarežģītība pārsniedz cilvēka prāta uztveres spēju robežu, tad, veicot sistēmas struktūras un likumsakarību izpēti, nav iespējams sasaistīt (organizēt) sistēmas elementus, kā arī nav iespējams sasaistīt cēloņus ar sekām. Līdz ar to cilvēks sistēmu vairs nespēj uztvert kā vienu veselu un izprast [BAR 1997, RIC 2000, ASH 2007]. Modelēšanas teorija ir attīstījusies, ņemot par pamatu jēdzienu „organizētība” [KRI 1986]. Lai analizētu sarežģītas sistēmas, izmanto sistēmu modelēšanu un imitāciju ar datoru, lietojot piemērotus rīkus [JOS 2000, WEA 2004, VAZ 2009], kas veicina izpratni par pētāmo objektu un atbalsta lēmumu pieņemšanu [RIC 2000].

Mūsdienās informācijas un komunikācijas tehnoloģijas (IKT) nodrošina pamata infrastruktūru sociālos un ekonomiskos procesos un ir būtiskas, lai veicinātu inovāciju rūpniecības nozarēs. IKT ir kļuvušas par pamatnosacījumu dažādu problēmsfēru izpētē, attīstībā un inovācijā [NGU 2010, INF 2011]. Pieaugot IKT lietojumam un aktualitātei atbilstoši pieaug arī nepieciešamība pēc esošo tehnisko risinājumu novērtēšanas un jaunu risinājumu izstrādes. Šajos procesos būtiska loma ir sarežģītu tehnisku sistēmu izpētei [INF 2011, PEI 2011]. Prasības pret rīku un pieeju, kas ļauj īstenot sistēmas modelēšanu, dotajā gadījumā nosaka sarežģītu tehnisku sistēmu specifika un par tām pieejamā informācija [GRU 1997a, KHA 2010]. Galvenās sarežģītu tehnisku sistēmu īpašības ir: daudzveidīgas komponentes un mijiedarbības starp tām, grūti analizējama hierarhiska struktūra un sarežģīta uzvedība. Zināšanas par šādām sistēmām bieži vien ir vairākiem ekspertiem nevis tikai vienam [GRU 2002, KHA 2010].

Rīgas Tehniskajā universitātē ir izstrādāta struktūrmodelēšanas pieeja, kas ir piemērota sarežģītu tehnisku sistēmu struktūrmodelēšanai un analīzei normālos funkcionēšanas apstākļos, kā arī kļūdu gadījumā. Pieejai ir dažādas priekšrocības: tā ir piemērota struktūras modelēšanai nepilnīgas informācijas apstākļos, modelēšanā tiek lietota grafiskā notācija un tā ļauj pētīt dažādus sistēmas morfoloģiskos un funkcionālos aspektus. Tomēr pieeja nav implementēta atbilstošā datorsistēmā, kas ļautu veikt automatizētu modeļu izveidi. Lai sarežģītu sistēmu struktūrmodelēšanu varētu veikt ar datoru, pieeja ir jāimplementē intelektuālā sistēmā [GRU 1997a, GRU 2002]. Tas ir būtisks priekšnosacījums, lai iegūtu, atspoguļotu un rezultatīvi apstrādātu zināšanas par sarežģītām sistēmām, kas piemīt dažādiem indivīdiem un dažādos laikos, un īstenotu automatizētu sistēmas struktūras modeļu izveidi. Lai izveidotu atbilstošu intelektuālu sistēmu, ir jānosaka kopsakarības starp sarežģītām sistēmām, to modelēšanas principiem un izvēlētās pieejas aspektiem.

Tēmas aktualitāte

Promocijas darba tēmas aktualitāte ir saistīta ar strauji augošu sistēmu sarežģītību un šādu sistēmu lomu informācijas un komunikācijas tehnoloģiju (IKT) pielietojumā. IKT ir būtiskas, lai uzlabotu produktivitāti un veicinātu attīstību rūpniecības nozarēs un zinātnē, kā arī apmierinātu sabiedrības prasības, attiecībā uz publiskiem pakalpojumiem (piemēram, veselību, izglītību un transportu). Līdz ar to IKT izpēte ir viena no Eiropas savienības septītās ietvarprogrammas (FP7) prioritātēm 2013.gadā [EUR 2012]. Sarežģītu tehnisku sistēmu izmantošana ir izplatīta gan sabiedrībā, gan rūpniecības nozarēs un zinātnē, lietojot dažādas IKT. Lai šādas sistēmas veidotu, pārvaldītu un pilnveidotu, ir jāveic to izpēte, lietojot atbilstošas pieejas un rīkus. Modelis ir galvenais izpētes līdzeklis sarežģītu sistēmu gadījumā [SKY 2006, YOU 2007, SOK 2010]. Sarežģītām sistēmām piemīt specifiskas īpašības, kas ierobežo modelēšanas pieeju izvēli. Attīstot IKT, aug modelēšanas iespējas. Lai gan ir izstrādātas daudzas pieejas un rīki, sarežģītu sistēmu uzbūves un funkcionēšanas, kā arī uzvedības modelēšana nav pilnībā atrisināta, jo parasti katra aspekta modelēšanai lieto savu matemātisko aparātu. Lai šo trūkumu novērstu, pagājušā gadsimta 70. gados tika radīta struktūrmodelēšanas pieeja. Pieejā tika izstrādāti sarežģītu sistēmu uzbūves un funkcionēšanas modeļu pamatprincipi un struktūras analīzes metodes. Struktūrmodelēšanā piedāvātās metodes un algoritmi, ļauj risināt ar sarežģītību saistītās problēmas, attiecībā uz tehniskām sistēmām ar fiziski daudzveidīgiem elementiem [GRU 1997b]. Tomēr pieejā aprakstītās iespējas nebija implementētas rīkā, kā arī nebija izstrādāti algoritmi modeļu automātiskai ģenerēšanai ar datoru (modeļus veidoja manuāli). Papildus no

struktūrmodelēšanas pirmsākumiem ir parādījušies jauni sarežģītu sistēmu aspekti, kas jāmodelē. Minētās problēmas ir risinātas šajā promocijas darbā.

Promocijas darba mērķis

Darba mērķis ir izstrādāt zināšanu izgūšanas un atspoguļošanas shēmu un ieviest to intelektuālā datorsistēmā, kas nodrošina sarežģītu tehnisku sistēmu ar heterogēniem elementiem struktūrmodelēšanu, īstenojot modeļu automātisku ģenerēšanu, kā arī aprobēt intelektuālu sistēmu, kā piemēru lietojot konkrētu sarežģītu sistēmu.

Darba uzdevumi

Promocijas darba mērķa sasniegšanai ir izvirzīti šādi uzdevumi:

- Izpētīt sarežģītu un intelektuālu sistēmu uzbūves un darbības principus un identificēt īpašības, kas ir jāņem vērā, veicot sistēmas modelēšanu;
- Izanalizēt struktūrmodelēšanas pieejas pašreizējās iespējas un noteikt tās trūkumus, kas ir būtiski intelektuālas datorsistēmas izstrādē;
- Pilnveidot struktūras modeļu sintaksi un semantiku, kā arī modeļu transformācijas algoritmus;
- Izstrādāt zināšanu izgūšanas un atspoguļošanas shēmu sarežģītu sistēmu struktūrmodelēšanas nodrošināšanai, ko var implementēt intelektuālā datorsistēmā;
- Izstrādāt transformācijas algoritmus, lai automatizētu struktūras modeļu izveidi;
- Izstrādāt intelektuālas datorsistēmas arhitektūru un veikt tās praktisko realizāciju, iekļaujot tajā struktūras modelēšanas un analīzes metodes;
- Pārbaudīt izstrādātās datorsistēmas iebūvēto funkcionalitāti un tās piemērotību struktūrmodelēšanas mērķu īstenošanai, realizējot tajā reālās pasaules sistēmas struktūras modelēšanu.

Pētījumu objekts

Darba pētījumu objekts ir struktūrmodelēšanas pieeja, kas piemērota sarežģītām tehniskām sistēmām ar heterogēniem elementiem.

Pētījumu priekšmets

Promocijas darba pētījuma priekšmets ir intelektuālas datorsistēmas arhitektūra un tās realizācija, kas ļauj īstenot sarežģītu sistēmu struktūrmodelēšanu un analīzi.

Darba zinātniskais jaunieguvums ir šāds:

- Ir izstrādāta zināšanu izgūšanas un atspoguļošanas shēma – freimu kopa, kas ir implementēta intelektuālā sistēmā un nodrošina ekspertu zināšanu izgūšanu un

atspoguļošanu par sarežģītām sistēmām, kā arī ļauj īstenot iegūto zināšanu saglabāšanu un apstrādi;

- Ir izstrādāti astoņi transformācijas algoritmi no freimu kopas uz morfoloģiskās un funkcionālās struktūras modeļiem, kas ir implementēti intelektuālā sistēmā sarežģītu sistēmu struktūrmodelēšanai (I4S) un nodrošina struktūrmodelēšanā aprakstīto struktūras modeļu automatizētu izveidi un vizualizāciju;
- Ir izstrādāta intelektuālas sistēmas arhitektūra sarežģītu sistēmu struktūrmodelēšanai, kas realizēta I4S un nodrošina izpētes sistēmas struktūras modeļu automatizētu izveidi, kā arī struktūras topoloģisko un kvalitatīvo analīzi.

Teorētiskā vērtība

Darba teorētiskā vērtība ir šāda:

- Struktūras modeļiem ir izstrādāti jauni pamatelementi (loģiskie operatori), izveidots to apraksts un vizualizēts attēlojums, kā arī pārbaudītas loģisko operatoru pielietošanas iespējas;
- Izstrādāts papildus morfoloģiskā struktūras modeļa pamatelementa apzīmējums, kas ļauj noteikt vai atspoguļotais sistēmas objekts ir elements vai komponente;
- Izveidota notācija jaunam funkcionālās struktūras modelim uzvedības telpā;
- Pilnveidota struktūras modeļu sintakse un semantika un transformācijas algoritmi starp struktūras modeļiem;
- Izstrādāta zināšanu izgūšanas un atspoguļošanas shēma un transformācijas algoritmi no freimu kopas uz struktūras modeļiem;
- Izstrādāta intelektuālas sistēmas arhitektūra sarežģītu sistēmu struktūrmodelēšanai.

Praktiskā nozīmība

Pētījumu praktiskā nozīmība ir šāda:

- Zināšanu izgūšanas un atspoguļošanas shēma – freimu kopa ir implementēta intelektuālas sistēmas arhitektūrā, kas ļauj netikai izgūt un atspoguļot eksperta zināšanas, bet arī saglabāt tās tā, lai zināšanas var koplietot, atkārtoti pielietot, kā arī izmantot automatizētai struktūras modeļu konstruēšanai un struktūras analīzei;
- Izveidotā intelektuālas sistēmas I4S arhitektūra ir praktiski īstenota programmatūras veidā, tādējādi pirmo reizi praktiski realizējot intelektuālu sistēmu, kas sevī ietver visus struktūrmodelēšanas pieejas aspektus – gan struktūras modeļu izveidi un attēlošanu, gan struktūras analīzi;

- Pārbaudīta izstrādātās sistēmas I4S darbība un atbilstība struktūrmodelēšanas mērķiem, atspoguļojot tajā zināšanas par sarežģītu tehnisku sistēmu – robots AGR8 un veicot izpētes sistēmas struktūras modeļu ģenerēšanu, kā arī struktūras analīzi.

Darba aprobācija

Par promocijas darba galvenajiem rezultātiem ir nolasīti četri referāti starptautiskās zinātniskās konferencēs Itālijā, Vācijā un Amerikas Savienotajās Valstīs:

- 2011. gada 12.–14. septembris. The 10th International Conference on Modeling and Applied Simulation. Roma, Itālija.
- 2010. gada 6.–9. aprīlis. International multi-conference on complexity, informatics and cybernetics (IMCIC 2010). Orlando, Amerikas Savienotās Valstis.
- 2008. gada 13.–15. oktobris. IADIS International Conference on Cognition and Exploratory Learning in Digital Age (CELDA 2008). Freiburga, Vācija.
- 2008. gada 17.–19. septembris. The 5th International Mediterranean Modelling and Latin American Modeling Multiconference, The international Workshop on Modelling & Applied Simulation. Amantea, Itālija.

Turklāt par atsevišķiem ar promocijas darbu saistītiem rezultātiem ir referēts konferencēs:

- 2005. gada 16.–17. jūnijs. The 6th International Conference on Computer Systems and Technologies (CompSysTech 2005). Varna, Bulgārija.
- 2005. gada 1.–4. jūnijs. The 19th European Conference on Modelling and Simulation (ECMS 2005). Rīga, Latvija

Promocijas darba ietvaros veikto pētījumu rezultāti ir atspoguļoti 5 publikācijās starptautiskos zinātniskos izdevumos:

- Zeltmate I. Logical Operator Usage in Structural Modelling, In: Proceedings of the 10th International Conference on Modeling and Applied Simulation, 2011, Rome, Italy, pp. 338-346.
- Zeltmate I., Grundspenkis J. An extension of frame-based knowledge representation schema, In: Proceedings of International Multi-conference on Complexity, Informatics and Cybernetics (IMCIC 2010), Vol I, 2010, Orlando, USA, pp. 401-406. (indeksēts: IIS, KGCM 2010, Google scholar).
- Zeltmate I., Grundspenkis J., Kirikova M., Prototype for the Knowledge Representation Supporting Inter-institutional Knowledge Flow Analysis, Chapter 6,

Learning and Instruction in the Digital Age, Springer, 2010, pp. 87-99. (indeksēts: SpringerLink, Google scholar).

- Zeltmate I., Grundspenkis J. Formal Method of Functional Model Building Based on Graph Transformations. In: Proceedings of the 5th International Mediterranean and Latin American Modeling Multiconference, The international Workshop on Modelling & Applied Simulation, 2008, Amantea, Italy, pp. 140-147.
- Zeltmate I., Grundspenkis J., Kirikova M., The Challenges in Knowledge Representation for Analysis of Inter - Institutional Knowledge Flows. In: Proceedings of the IADIS International Conference on Cognition and Exploratory Learning in Digital Age (CELDA 2008), 2008, Freiburg, Germany, pp. 145-152. (indeksēts: IADIS, Google scholar).

Papildus promocijas darba galveno rezultātu publicēšanai darba autorei vēl ir šādas ar promocijas darba tematiku saistītas publikācijas:

- Valkovska I., Grundspenkis J. Development of Frame Systems Shell for Learning of Knowledge Representation Issues. In: Proceedings of the 6th International Conference on Computer Systems and Technologies (CompSysTech 2005), pp. IV.11.-1 – IV.11.-6. (indeksēts: ECET).
- Valkovska I., Grundspenkis J. Representation of Complex Agents by Frames for Simulation of Internal Relationships in Structural Modelling. In: Proceedings of the 19th European Conference on Modelling and Simulation (ECMS 2005), 2005, pp. 151-157. (indeksēts: ECMS).
- Graudina V., Grundspenkis J., Valkovska I. Usage of Frame System for Modelling of Intelligent Tutoring System Architecture. In: Annual Proceedings of Vidzeme University College. ICTE in Regional Development. Valmiera, 2005, pp. 105-109 (indeksēts: EBSCO HOST).

Darba autore ir bijusi izpildītāja šādos zinātniskos projektos:

- Valsts pētījumu programmas "Inovātīvu daudzfunkcionālu materiālu, signālapstrādes un informātikas tehnoloģiju izstrāde konkurētspējīgiem zinātņu ietilpīgiem produktiem" 5. projekta "Jaunas informācijas tehnoloģijas balstītas uz ontoloģijām un modeļu transformācijām" 5.2.apakšprojekts (no 2010. gada maija);
- Rīgas Tehniskās universitātes projekts (Eiropas Reģionālās attīstības fonda projekta 2.1.1.1.aktivitātes "Atbalsts zinātnei un pētniecībai" ietvaros) „Daudzaģentu

robotizētas intelektuālas sistēmas tehnoloģijas izstrāde” (no 2010. gada decembra līdz 2011. gada septembrim);

- Latvijas Republikas Izglītības un Zinātnes ministrijas un Rīgas Tehniskās universitātes pētnieciskais projekts, „Starpinstitucionālas zināšanu plūsmas atbalsta sistēmas prototipa izstrāde” (no 2007. gada oktobra līdz 2008. gada septembrim).

Darba struktūra

Promocijas darbs sastāv no ievada, četrām nodaļām, secinājumiem, bibliogrāfiskā saraksta un trīs pielikumiem.

Ievadā ir aprakstīta sarežģītu sistēmu izpētes nozīmība un pamatota veikto pētījumu aktualitāte, formulēts darba mērķis un uzdevumi, aprakstīta pētījumu zinātniskā novitāte, iegūtie teorētiskie rezultāti un to praktiskā nozīmība.

Darba pirmajā nodaļā ir izanalizētas sarežģītas sistēmas un identificētas to galvenās īpašības, kas ir būtiskas šādu sistēmu modelēšanā. Ir aprakstītas struktūrmodelēšanas iespējas un ierobežojumi un konstatēts, ka, lai atbalstītu pieejas mērķu īstenošanu, to ir nepieciešams implementēt intelektuālā sistēmā. Tālāk ir izanalizēta intelektuālu sistēmu uzbūve un tās darbības mehānismi.

Darba otrā nodaļa ir veltīta struktūrmodelēšanas pieejas detalizētam aprakstam. Tajā ir aplūkota struktūras modeļu notācija un aprakstīta esošo un jaunizveidoto pamatelementu sintakse un semantika. Nodaļā ir aprakstītas transformācijas starp struktūras modeļiem, ievērojot loģisko operatoru lietojumu.

Darba trešajā nodaļā ir skaidrots freima jēdziens, dota tā struktūras un pielietojuma analīze, ar mērķi izstrādāt zināšanu izgūšanas un atspoguļošanas shēmu. Balstoties uz veikto analīzi, ir izveidota freimu kopa, kas ir implementēta intelektuālā sistēmā I4S un atbalsta struktūras modeļu automātisku konstruēšanu. Ir aprakstīti izstrādātās sistēmas I4S uzbūves un darbības pamatprincipi, kā arī transformācijas no freimu kopas uz struktūras modeļiem.

Darba ceturtajā nodaļā ir izpētīta sarežģītas tehniskas sistēmas – robota AGR8 uzbūve un darbības principi, un veikta dotās sistēmas struktūras modelēšana, lai pārbaudītu izveidotās sistēma I4S funkcionalitāti. Praktiskā piemēra realizācija apstiprina sarežģītu tehnisku sistēmu struktūrmodelēšanas un analīzes iespējas izstrādātajā sistēmā I4S.

Promocijas darba galvenie rezultāti un secinājumi ir doti darba noslēguma nodaļā.

1. SAREŽĢĪTAS SISTĒMAS UN STRUKTŪRMODELĒŠANA

Meklējot risinājumus dažādām problēmām, strauji attīstās zinātne. Tas rada nepieciešamību pilnveidot esošās, kā arī radīt jaunas, mūsdienu prasībām un sasniegumiem atbilstošas tehnoloģijas. Moderno tehnoloģiju ietekmē palielinās mijiedarbību un savstarpējo saistību skaits starp zinātnes nozarēm, dažādiem darbības veidiem un objektiem. Ar objektu šeit tiek domāts „jebkas, uz ko var būt vērstas domas vai darbības” [OXF 2009]. Tiek uzlabotas un paplašinātas cilvēku savstarpējās komunikācijas iespējas, to veidi un ātrums, kas veicina izzini. Rezultātā pieaug cilvēku informētība, kā arī prasības pret ārējo vidi. Mainās skatījums uz pasaulē eksistējošiem objektiem, likumsakarībām un agrāk pastāvējušiem pieņēmumiem. Citiem vārdiem sakot, apstākļu kopums, kas ietekmē cilvēku dzīvi un darbības, kļūst arvien daudzveidīgāks un plašāks. Cilvēki izstrādā, vada un uztur arvien sarežģītākus artefaktus¹ un sistēmas, tas ir, sastopas ar strauji augošu sarežģītību.

1.1. Problēmsfēras jēdzieni un to interpretācija

Attiecībā uz sarežģītiem objektiem ir jāizprot ne tikai izpētes objekts, bet arī ar to saistītais apstākļu kopums un pastāvošās problēmas. Jēdziens (jeb koncepts) „problēma” ir cēlies no sengrieķu valodas un to skaidro kā „uzdevums, izvirzīts jautājums” [OXF 2009]. No sistēmu domāšanas (angļu val. – system thinking) pieejas viedokļa, problēmas apskata kā sistēmas daļu [CAP 1996]. Turpmāk ar pētāmo objektu vai izpētes objektu tiek domāta sarežģīta situācija, sistēma un/vai problēma.

Lai izvēlētos piemērotu metodi un /vai rīku pētāmā objekta analīzei, ir jāpārzina sfēra un ar to saistītie jēdzieni, un šādā gadījumā ir būtiska viennozīmīga jēdziena izpratne [ASH 1956, VIC 2002, APP 2011]. Tomēr bieži kā sinonīmus lieto dažādus konceptus, lai gan to jēdzieniskās nianse un skaidrojums ir atšķirīgs. Lai viestu izpratni par izmantotiem jēdzieniem, un skaidrotu to nepieciešamību sarežģītu sistēmu izpētē un atbilstoša modelēšanas rīka izvēlē, tālāk ir sniegta promocijas darbā lietoto konceptu interpretācija.

¹ Objekts, ko ir radījis cilvēks [OXF 2009]

1.1.1. Sistēma un ar to saistītie jēdzieni

Par sistēmu uzskata jebkuru objektu (reālu vai abstraktu), kam piemīt noteiktas iezīmes, darbības principi, funkcijas un uzvedība [HAL 1968, ACK 1971, ROS 1979, BEE 1995, YOU 2007]. Iezīmes ir objekta pazīmes vai īpašības, kas ir raksturīgas objektam un ļauj to identificēt [OXF 2009]. No sistēmu teorijas viedokļa objektu aplūko trīs dažādos veidos [ASH 1956, SKY 2006]:

- 1) kā apskatīto (pētījuma vai izpētes) objektu, kas atbilst pasaules realitātei;
- 2) kā apskatīto objektu, ko uztver cilvēks. Uztvere par objektu būs atšķirīga dažādiem cilvēkiem;
- 3) kā uztvertā objekta modeli vai atspoguļojumu.

Jēdzienu „sistēma” attiecina uz pirmo un otro gadījumu, bet attiecībā uz trešo piemēro konceptu „modelis”. Modelis ir vienkāršots izpētes objekta vai tā struktūras apraksts/atspoguļojums, kas parasti ietver sevī tikai būtiskās oriģinālā objekta iezīmes, kas interesē pētnieku dotā uzdevuma ietvaros [KRI 1986, AMA 2004, OXF 2009]. Apskatot no kibernetikas² viedokļa, sistēma ir cilvēka (novērotāja) izveidota. Ja sistēma apraksta, imitē vai prognozē iespējamus risinājumus attiecībā uz kādu daļu no savas vides, tad tā var tikt uzskatīta kā modelis attiecībā uz šo daļu. Šādas sistēmas neeksistē bez novērotāja (turpmāk darbā, saukts arī par ekspertu) [KRI 1986]. Sistēmas atspoguļošana ir process, kura laikā izveido izpētes objekta modeli, kas ataino abstraktu un formālu struktūru vai sistēmas stāvokļu telpu, vienlaicīgi norādot arī saistības starp attēlotiem vienumiem [HEY 1990].

Sistēmu definē kā elementu un attieksmju kopu, kas nosaka sistēmas eksistenci [BER 1969, ACK 1971, ROS 1979, CHU 1979, BEE 1995, AMA 2004, SKY 2006, BOP 2008, SOK 2010]. Aprakstot sistēmas, jēdzienus „daļa”, „komponente” un „elements” bieži vien lieto kā sinonīmus [HAK 2006, BOP 2008]. Tomēr koncepti ir jāizmanto, ņemot vērā atbilstošu skaidrojumu un apskatītās realitātes struktūru. Jēdziens „elements” attiecas uz sistēmas primitīviem (sastāvdaļām) jeb pamatelementiem; monolītām daļām vai daļām, kurām neveic dekompozīciju [YOU 2007, OXF 2009, AST 1996]. Informāciju par sistēmas daļām, to savstarpējo saistību un sistēmas uzbūvi var iegūt dekompozīcijas procesā [BAR 1997, BRO 1998, AMA 2004, SKY 2006, HAK 2006]. Dekompozīcija ir konceptuāla vai fiziska metode,

² Zinātne par komunikācijas un automātiskās kontroles sistēmām, likumiem – dabā un cilvēku radītos mehānismos [Terminu vārdnīca, OXF 2009]

kas ļauj izpētes objektu sadalīt mazākās sastāvdaļās [BRO 1998, HAK 2006] un tādējādi vienkāršot skatījumu uz sistēmu un izprast katru izvēlēto sistēmas līmeni.

Jēdzienu „komponente” attiecina uz saliktu sastāvdaļu (apakšsistēmu), kurai var veikt dekompozīciju. Komponente var sastāvēt no citām komponentēm un/vai elementiem [OXF 2009]. Daļa ir elements vai komponente, kas ir būtiska apskatītajam objektam [OXF 2009]. Sistēma sastāv no daļām un to var hierarhiski sadalīt komponentēs un elementos. Sistēmas daļām var būt atšķirīgi izmēri un, tās var būt gan viendabīgas (homogēnas) – tādas, kurām nav atšķirīgu iezīmju, gan neviendabīgas (heterogēnas) – ar atšķirīgām elementu un/vai struktūras iezīmēm. Daļas spēj funkcionēt bez pastāvīgas uzraudzības un realizēt izvirzītos mērķus ar vadību un/ vai kontroli, bet bez tiešas iejaukšanās darbību izpildē [KRI 1986, BAR 1997, AST 1996]. Neskatoties uz to, kādā abstrakcijas līmenī sistēmu pēta, to var aplūkot kā sastāvošu no objektiem [HAL 1968, AMA 2004, WEI 2009]. Līdz ar to, runājot par sistēmas daļām, var lietot konceptus „daļa” un „objekts”.

Sistēmai un tās daļām ir noteikta struktūra [YOU 2007], kas raksturo uzbūvi [BEE 1995, GRU 1999]. Struktūra ir saistība starp daļām, kas līdz ar daļu identitāti, veido vienotu veselu, ņemot vērā to, ka starp daļām pastāv noteikta kārtība [MAT 1974, OXF 2009]. Tas ir, sistēmas daļas ir sakārtotas pēc noteiktiem principiem cita attiecībā pret citu. Divām sistēmām ir vienāda struktūra, ja tām ir ekvivalentas saites starp ekvivalentām sistēmas daļām [MAT 1974]. Mijiedarbības un saites starp sistēmas daļām ir tikpat būtiskas kā pašas daļas [ROS 1979]. Sistēma ir kopa, kas sastāv no objektiem, kurus vieno regulāras saites un mijiedarbības, veidojot organizētu vienumu [HAL 1968, ROE 1969, BER 1969]. Caur mijiedarbībām sistēma, kā arī tās daļas, realizē noteiktu funkcionalitāti, kas ļauj sasniegt konkrētus mērķus [ROS 1979, BAR 1997, SKY 2006]. Mijiedarbību definē kā attiecības starp daļām, ar nosacījumu, ka izmaiņa vienā daļā rada izmaiņas citās daļās un visā sistēmā kopumā [GRU 1972, KUH 1974, CAP 1996]. Mijiedarbības veido noteiktu organizētību sistēmā. Jēdziens „organizētība” ir cēlies no grieķu vārda „organon”, un tas nozīmē – rīks kāda uzdevuma veikšanai. Šis koncepts ir saistīts ar sistēmas daļas funkciju vai lomu, ko īsteno sistēmā, veidojot vienotu veselu [MAT 1974]. Organizētība sistēmā pastāv, ja attiecības starp diviem sistēmas elementiem A un B kļūst par nosacījumu, kas ietekmē elementa C īpašības, iezīmes, vērtības vai stāvokli [ASH 2004]. Ņemot vērā to, ka sistēma ir viens vesels, kurā nepastāv izolēti elementi [HAL 1968, ACK 1971, BEE 1995, CAP 1996], tad, ja ietekmē vienu elementu, tad tiek ietekmēti citi elementi un visa sistēma kopumā. Organizētību definē kā sistēmas iezīmi, kuru raksturo struktūra, kas ir mērķtiecīgi izveidota, lai realizētu noteiktu funkcionalitāti [MAT 1974, HEY 2001, YOU 2007]. Tiek minēts, ka sistēmas struktūra ir

relatīvi nemainīga laikā [GRU 1997a, GRU 1997b, MOU 2009], ar to ir domāta struktūra, kas atbilst sistēmas organizēšanai [MAT 1974, GRU 1997a, BAR 1997, SKY 2006, APP 2011].

Jēdziens „organizēšana” atšķirībā no koncepta „struktūra”, attiecas uz saitēm, kas veido vienotu sistēmu, un līdz ar to nosaka arī sistēmas iezīmes [MAT 1974, ASH 2004]. Turklāt koncepts neattiecas uz sistēmas komponentu iezīmēm, kas var būt jebkādas, kamēr vien tās nodrošina noteikto saišu eksistenci. Līdz ar to, veicot sistēmu analīzi, ir būtiski izprast atšķirību starp šiem abiem jēdzieniem. Divas sistēmas ir organizētas vienādi, ja saites, kas veido tās kā vienotu veselu ir vienādas, neskatoties uz to, kā šīs saites ir iegūtas. Tādējādi divas sistēmas var būt vienādi organizētas pat tad, ja tām ir atšķirīgas struktūras. Divas sistēmas ir ekvivalentas tikai tad, ja tās ir organizētas vienādi. Ja veids, kā sistēma ir organizēta, mainās, tad sistēmas identitāte mainās un tā kļūst par citu vienumu ar atšķirīgām iezīmēm. Ja veids, kā sistēma ir organizēta, paliek nemainīgs, kad sistēmas struktūra mainās (piemēram, sistēmai attīstoties un mācoties), tad sistēma paliek tā pati, nezaudējot savu identitāti [MAT 1974, GRU 1997a, BAR 1997, SKY 2006]. Veids, kā sistēma ir organizēta, atspoguļo to kā vienotu veselu jebkurā telpā, bet struktūra veido konkrētu vienumu sistēmas komponentu telpā [MAT 1974]. Atbilstoši, veicot sistēmu analīzi, parasti pēta iezīmes, kas ir saistītas ar sistēmas būtību un veidu, kā tā ir organizēta, un tās ir neatkarīgas no: problēmsfēras, kurā sistēma tiek pētīta; sistēmas tipa; sistēmas izmēra; kā arī matērijas, no kuras sistēma sastāv. Sistēmām izšķir tādas iezīmes kā [ASH 1956, SKY 2006]: struktūru (komponentes un to saistība); mijiedarbības un saistības (funkcionālās, strukturālās); funkcijas (mērķi); uzvedību; likumus. Funkcijas raksturo mērķu sasniegšanai nepieciešamās darbības, bet uzvedība – funkciju realizācijas veidus [BEE 1995, GRU 1999]. Likumi nosaka un regulē funkciju pareizu izpildi, kā arī struktūru un organizēšanu sistēmā [ASH 1956, SKY 2006].

1.1.2. Sarežģītība un ar to saistītie jēdzieni

Sarežģītību apskata kā apstākļu un iezīmju kopumu, kas, pieņemot realitātes³ dabu, ļauj attīstīties, pašorganizēties, ir nelineārs, jutīgs pret sākuma nosacījumiem, un to ietekmē daudzas likumsakarības un negaidīti radušās īpašības [WOO 2004]. Pašorganizēšanās ir sarežģītām sistēmām raksturīga uzvedība, sistēmas iekšējās dinamikas rezultāts, kā rezultātā mainās sistēmas struktūra, lai nodrošinātu eksistenci [MCC 2000, ASH 2004, YOU 2007,

³ Objekts, kas reāli pastāv, eksistē [OXF 2009]

HEY 2008]. Pašorganizēšanās bieži rodas sistēmas mācīšanās, attīstīšanās un adaptēšanās procesos, un tā izpaužas kā spontāna jaunu īpašību, sistēmas struktūru un uzvedības formu veidošana. Piemēram, vairāki sistēmas elementi atbild par noteiktas funkcionalitātes īstenošanu, un pašorganizēšanās procesā tos apvieno vienā komponentē, kurai var būt īpašības un uzvedība, kas nav atkarīga no pakļauto elementu iezīmēm. Pašorganizēšanās rezultātā palielinās organizētība sistēmā [MCC 2000, BOP 2008].

Mācīšanās ir zināšanu vai uzvedības maiņa, ņemot vērā iepriekšējo pieredzi, kā rezultātā sistēma ir spējīga uzlabot savu funkcionalitāti [SKY 2006, BOP 2008]. Adaptēšanās attiecas uz izmaiņu procesu, ko īsteno, lai sasniegtu noteiktu mērķi vai lai pielāgotos iekšējām vai ārējās vides izmaiņām, vai lai mainītu ārējo vidi [ACK 1971, BAR 1997, MCC 2000, BOP 2008].

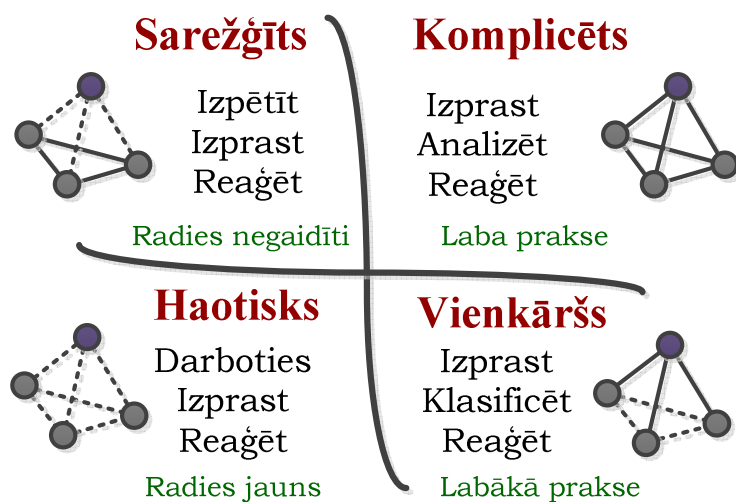
Attīstīšanās ir spēja pielāgoties izmaiņām – tā ir reakcija uz mainītām prasībām, ierobežojumiem attiecībā uz sistēmas funkcijām, un sistēmai, elementiem un/vai saitēm raksturīgām īpašībām [YOU 2007, HEY 2008]. Attīstību īsteno, ņemot vērā atgriezenisko saiti, un attīstīšanās rezultātā iegūst jaunas iezīmes un struktūras, kuras ir iespējams koplietot [BAR 1997, YOU 2007].

Jēdzienu „nelineārs” skaidro kā linearitātes trūkumu jeb gadījumu, kad nepastāv tieši proporcionālas izmaiņas starp diviem saistītiem vienumiem (piemēram, ieeja un izeja) [OXF 2009]. Tiek pieņemts, ka sarežģītas sistēmas ir sistēmas, kas ir izveidojušās no vienkāršām funkcionējošām sistēmām, pašorganizējoties un attīstoties laika gaitā [SIM 1962, YOU 2007, APP 2011].

Nav viena vienīga veida kā definēt un izmērīt sarežģītību. Jēdziens „sarežģīts” ir multidimensionāls un multidisciplinārs [SIM 1962, HOR 1995, MCC 2000, RIC 2001, WOO 2004, HEY 2008, JOH 2009], un to skaidro, izmantojot tādas iezīmes kā: „sastāvošs no savstarpēji saistītām vai savītām daļām” un „grūti izprotams vai analizējams” [BAR 1997, MAG 2004, HEY 2008, OXF 2009]. Doto konceptu var attiecināt uz jebkuru kopumu, izpētes objektu, kuram piemīt minētās raksturiezīmes. Tomēr visbiežāk sarežģītību skaidro ar iezīmēm (piemēram, īpašībām un uzvedību), kas raksturo sarežģītas sistēmas un nosaka katras sistēmas unikalitāti [GLO 2002, SKY 2006, HEY 2008]. Jēdzienu „sarežģīts” attiecina uz sistēmas stāvokli, kad daļas ir savstarpēji integrētas, veidojot vienu veselu, bet daļu daudzums un/vai daudzveidība ir pārāk liela, lai šādu sistēmu varētu skaidrot un izprast vienkāršos, vispārpieņemtos veidos [MCC 2000].

Pētot sarežģītību, bieži vien apskata jēdzienu „vienkāršs”, „komplicēts” un „sarežģīts” salīdzinājumu [MCC 2000, HEY 2008, JOH 2009, SNO 2010, APP 2011]. Lai skaidrotu

sarežģītu sistēmu attīstību un minētos konceptus, Snowden (Snowden) izveidoja ietvarstruktūru (no angļu val. framework) Cynefin [SNO 2005, SNO 2010]. Ietvarstruktūrā ir četras dimensijas (skatīt 1.1. att.), kas ļauj klasificēt un noteikt kopsakarības starp pasaulē eksistējošām problēmām, situācijām un sistēmām, kā arī atbalsta lēmumu pieņemšanu. Uzsverot konceptu kopīgās un atšķirīgās nianšes, tiek skaidrotas iezīmes, kas ir jāņem vērā, lai saprastu katru no jēdzieniem [MCC 2000, GLO 2002, HEY 2008, JOH 2009]. Nereti jēdzienus „komplicēts” un „sarežģīts” lieto kā sinonīmus, kaut gan konceptu skaidrojumam un pielietojumam ir būtiskas atšķirības. Veicot pētāmā objekta analīzi, ir svarīgi noteikt vai objekts ir komplicēts vai sarežģīts. Pieņemot, ka iepriekš minētie jēdzieni ir identiski, pētāmam objektam var tikt piemērota neatbilstoša izpētes metode un/vai rīks.



1.1. att. Cynefin ietvarstruktūra [SNO 2010]

Sarežģītas sistēmas sastāv no daļām, kas var būt gan vienkāršas, gan komplicētas, gan sarežģītas. Atbilstoši, tālāk ir sniegts jēdzienu „vienkāršs”, „komplicēts” un „sarežģīts” skaidrojums:

- koncepts „vienkāršs” attiecas uz pētāmiem objektiem, kas ir zināmi [SNO 2010, APP 2011]. Vienkāršu objektu gadījumā saistības starp cēloņiem un sekām ir izprotamas, tās var atveidot un par tām var spriest. Vienkāršām sistēmām ir raksturīgas likumsakarības, kas laikā būtiski nemainās [SNO 2010] – struktūra, organizētība un uzvedība ir noteikta un stabila [SNO 2010, APP 2011]. Vienkāršas problēmas var atrisināt, sekojot priekšrakstiem, kas iepriekš ir izpētīti un cilvēkiem labi zināmi, standartizēti [GLO 2002]. Tas, kas vienam indivīdam, šķiet vienkāršs, citam var šķist komplicēts, un to nosaka zināšanu, prasmju un iespēju esamība. Tomēr, sastopoties ar vienkāršu objektu, cilvēkam nav nepieciešams daudz laika,

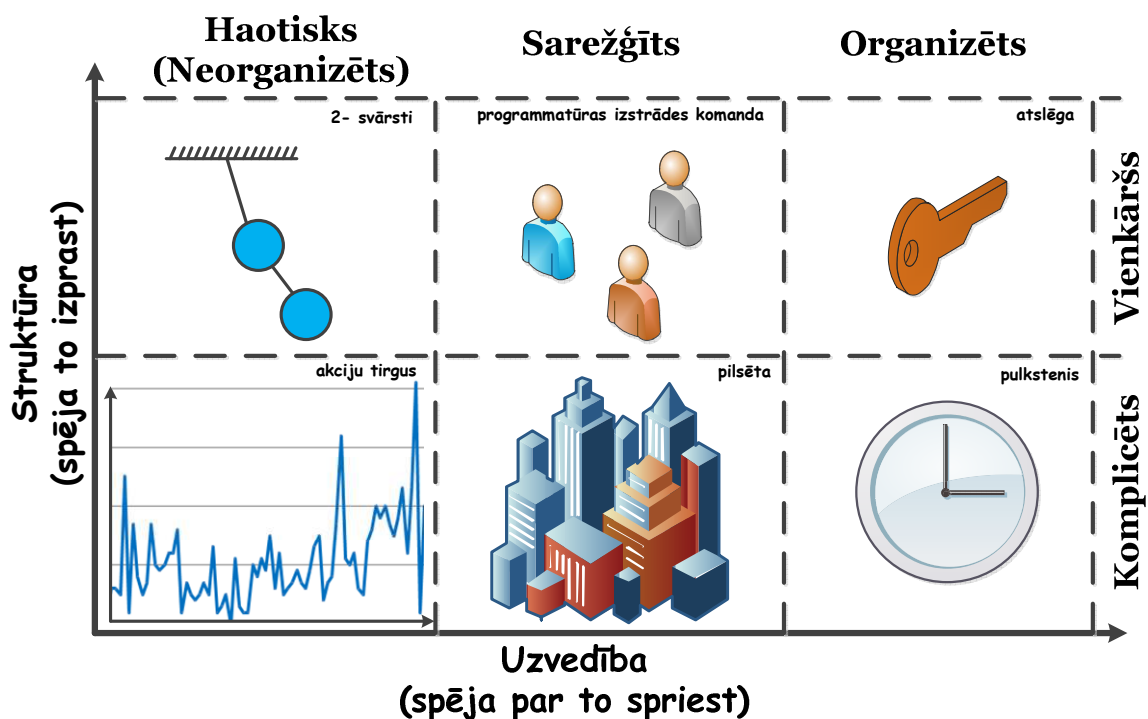
lai to izprastu un par to paskaidrotu citiem. Piemēram, automašīnas atslēga ir vienkāršs objekts [APP 2011].

- jēdziens „komplicēts” attiecas uz izpētes objektiem, ko ir iespējams pārzināt (no angļu val. knowable). Komplicētu objektu gadījumā, lai izprastu saites starp cēloņiem un sekām, ir nepieciešams veikt analīzi vai izmantot citu izpētes formu un/vai piemērot eksperta zināšanas [SNO 2010]. Komplicētību raksturo kā: „sastāvošs no daudz, savstarpēji saistītām daļām” [OXF 2009]. Daudzu komplicētu pasaules fenomenu pamatā ir vienkārši likumi [HOR 1995]. Komplicētas problēmas ir iespējams vispārināt, un tās satur vienkāršu problēmu apakškopas, bet tās nevar reducēt tikai uz vienkāršām problēmām. Komplicētība bieži vien ir saistīta ar apjomu (elementu, saišu skaitu), kopsakarību meklēšanu un specifiskām zināšanām [GLO 2002]. Komplicētas problēmas, līdzīgi kā vienkāršas problēmas, var atrisināt, sekojot instrukcijām. Piemēram, aizpildīt deklarāciju ir komplicēti, bet ne sarežģīti [JOH 2009].
- koncepts „sarežģīts” attiecas uz objektiem, kurus nevar pilnībā pārzināt un pārvaldīt, bet var ietekmēt. Sarežģītu objektu gadījumā saites starp cēloņiem un sekām var izprast, izpētot zināmās objekta iezīmes un uzvedību [SNO 2010]. Sarežģītas problēmas var iekļaut vienlaikus gan vienkāršas, gan komplicētas problēmas, bet tās nevar būt reducētas uz nevienu no tām. Ir jāņem vērā, ka, ja pētāmais objekts ir sarežģīts, tas nenozīmē, ka tas noteikti ir komplicēts. Sarežģītām problēmām ir raksturīgas specifiskas prasības, nelinearitāte, spēja pielāgoties, ja mainās nosacījumi [GLO 2002]. Piemēram, sarežģīta situācija rodas, ja deklarāciju cenšas aizpildīt trīs savstarpēji saistīti cilvēki vienā un tajā pašā laikā. Papildus, aizpildot deklarāciju, katrs ņem vērā savas zināšanas, seko savām stratēģijām un vienlaicīgi reaģē arī uz citu indivīdu darbībām. Ir jāņem vērā, ka elementi, kas mijiedarbojas, var būt vienkārši, bet mijiedarbību rezultātā var rasties sarežģītas īpašības un uzvedība [HOR 1995, GLO 2002, JOH 2009]. Sarežģītām sistēmām ir raksturīga augsta pašorganizēšanās [SNO 2010]. Sarežģītība var izpausties arī tad, ja ir tikai divas komponentes, kas mijiedarbojas un ir saistītas tādā veidā, ka tās ir grūti atdalīt. Sarežģītība pieaug, palielinoties īpatnību (atšķirīgas komponentes, stāvokļi, aspekti), kā arī saistību un/ vai mijiedarbību skaitam [HEY 2008].

Vienkāršus un komplicētus izpētes objektus var izprast, veicot to dekompozīciju, izpētot un analizējot to iezīmes. Lai izprastu sarežģītus objektus ir nepieciešams tos izpētīt, analizēt,

kā arī novērot to darbību un uzvedību [APP 2011]. Sarežģītas sistēmas var būt arī tādas sistēmas, kas sastāv tikai no dažām daļām, bet tai pašā laikā īsteno sarežģītu uzvedību [ASH 2004, HAK 2006, APP 2011]. Tiek uzskatīts, ka sistēmas kopējā uzvedība veidojas no mijiedarbībām starp sistēmas daļām [MCC 2000, APP 2011], un bieži sarežģītas sistēmas pēta tieši saistībā ar to uzvedību – pašorganizēšanos, attīstību, mācīšanos, adaptēšanos. Atbilstoši struktūras un uzvedības saistībai ar sarežģītību, Appelo ir izveidojis skaidrojošu struktūras–uzvedības modeli (skatīt 1.2. att.). Struktūras – uzvedības modelī izmanto divas dimensijas:

- *sistēmas struktūra* un spēja to izprast. Struktūra var būt:
 - **vienkārša** - vienkārši izprotama;
 - **komplicēta** – ļoti grūti izprotama.
- *sistēmas uzvedība* un spēja to iepriekš noteikt jeb prognozēt (no angļu val. predict). Uzvedība attiecībā uz spēju par to spriest var būt:
 - **organizēta** – uzvedība ir pilnībā iepriekš nosakāma;
 - **sarežģīta** – dažas lietas ir iespējams iepriekš noteikt;
 - **haotiska** – praktiski neprognozējama.



1.2. att. Struktūras – uzvedības modelis [APP 2011]

Atbilstoši izveidotajam modelim, atslēga un pulkstenis ir organizētas sistēmas, un to uzvedība ir iepriekš nosakāma. Nav nepieciešams ilgs laiks, lai izprastu to konstrukciju un komponentes. Trīs personu programmatūras izstrādes komandas un pilsētas uzvedība ir

iepriekš prognozējama, bet ne pilnībā – „nekad nevar būt pārlicināts, par to kas notiks rīt”. Savukārt divi saistīti svārsti un akciju tirgus ir sistēmas, kuras ir ļoti grūti izprast, tās ir jutīgas attiecībā uz dažādiem faktoriem un to uzvedību ir ļoti grūti prognozēt [APP 2011].

1.1.3. Sarežģītas sistēmas un ar tām saistītie jēdzieni

Cilvēki saskaras ar sarežģītām sistēmām dažādos veidos: tās projektējot, izstrādājot, analizējot, pilnveidojot, kā arī ekspluatējot un pārvaldot. Jo sarežģītāka ir sistēma, jo sarežģītāki ir tās uzbūves un darbības principi, un jo grūtāk to izprast. Pieaugot sistēmu sarežģītības pakāpei, pieaug nepieciešamo zināšanu daudzums sistēmu pārvaldītājam un lietotājam. Ja sistēmu procesi ir automatizēti, cilvēkiem, kuri lieto dažādas sistēmas, nav nepieciešams iedziļināties sistēmas darbības un uzbūves pamatīpatnībās⁴. Bet, lai sistēmas radītu, pārvaldītu un pilnveidotu, tajās ir jāorientējas. Līdz ar to ir nepieciešamas atbilstošas, visaptverošas un specifiskas zināšanas, analītiskas spējas, kvalifikācija un tehnoloģiskais atbalsts. Sarežģītu sistēmu izpēti un ar to saistīto pieeju attīstības mērķis ir izprast un atspoguļot esošās sistēmas tā, lai tās varētu mainīt, tādējādi radot jaunas sistēmas, kurām ir pielietojums dažādās dzīves sfērās [BAR 1997, HEY 2008]. Lai spētu orientēties sarežģītās sistēmās un lai risinātu ar tām saistītos jautājumus, piemēram, sistēmas procesu automatizāciju, ir jāpārzina sistēmu struktūra un veids kā tā organizēta. Par sarežģītām sistēmām bieži vien nav pieejama pilnvērtīga informācija [GRU 1972, SKY 2006] un cilvēki spriež par iespējamām situācijām un pieņem lēmumus, balstoties uz paraugiem (no angļu val. patterns) [SNO 2005]. Sarežģītas sistēmas var izprast, izpētot to darbības principus un tām raksturīgos paraugus [BAR 1997, MCC 2000, SKY 2006, HEY 2008]. Paraugi var būt: a) attieksmju kopa, kas atbilst aplūkotai sistēmai vai sistēmu kolekcijai; b) negaidīti radusies iezīme, kas raksturīga visai sistēmai kopumā c) sistēmas īpašība, kas ļauj sistēmas aprakstu saīsināt, ja salīdzina ar sistēmas daļu aprakstu (piemēram, kāda sistēmas īpašība, kuru manto visas sistēmas daļas) [BAR 1997]. Paraugi ir struktūra, kas var atkārtoties laikā un telpā [HOR 1995, BAR 1997, HEY 2008].

⁴ Uzskatāms piemērs ir automašīnas pretaizdzīšanas sistēma, kuras aktivizēšanai vai deaktivizēšanai ir jānospiež tikai atbilstošs slēdzis.

Attiecībā uz sarežģītību un sistēmām pēta fenomenu⁵, kas rodas sistēmas daļām savstarpēji mijiedarbojoties un var izpausties kā sistēmas līmenī negaidīti radusies raksturīga iezīme: īpašība, stāvoklis, attieksme, uzvedība, utt. Ar konstatējumu „negaidīti radies” šeit vienlaicīgi tiek apzināti divi aspekti: a) parādība var rasties pēkšņi b) parādība, kas izveidojas, attiecas uz visu sistēmu. Fenomens piemīt sistēmai kopumā, bet nepiemīt atsevišķām sistēmas daļām [BAR 1997, EDM 1999, MCC 2000, SKY 2006, JOH 2009, SNO 2010, APP 2011]. Pie noteiktiem kritērijiem parādība var kļūt uzskatāma, bet par to nevar spriest un pieņemt lēmumus, ņemot vērā tikai zināšanas par atsevišķām sistēmas komponentēm [OXF 2009, JOH 2009]. Piemēram, var zināt automašīnu iezīmes, bet tas neizskaidros to, kāpēc ir radies ceļu satiksmes sastrēgums noteiktā vietā un laikā. Ievērojot to, ka fenomens ir raksturīgs visai sistēmai kopumā, turpmāk darbā, lai apzīmētu attiecīgo parādību, lieto salikteni, kas sastāv no vārda „sistēma” un vārda, kas raksturo konkrētu sistēmas iezīmi (turpmāk sistēmiezīmes), piemēram, sistēmīpašība, sistēmuzvedība.

Sarežģītības teorija darbojas starp diviem ekstrēmiem: pilnīgu (jeb absolūtu) organizētību un pilnīgu neorganizētību jeb haosu [HOR 1995, EDM 1999, WEA 2004, SKY 2006, HEY 2008, JOH 2009, JOH 2009, SNO 2010]. Nekas jauns nevar rasties sistēmā, kurā pastāv pilnīga organizētība un stabilitāte (kā piemēru var apskatīt kristālu). Toties pilnīgi haotiskā sistēmā nevar izšķirt noteiktus paraugus un saistības starp daļām (šeit kā piemēru var minēt uzsildītu gāzi) [HOR 1995]. Šādā gadījumā nav pieejami fakti, kurus var droši izmantot spriešanai. Izdala divas sistēmu sarežģītības formas – neorganizēta un organizēta sarežģītība [BAR 1997, WEA 2004, SKY 2006, VAZ 2009]. Sarežģīta, neorganizēta sistēma sastāv no ļoti daudz daļām un saitēm, un to daudzums rada grūtības izprast un identificēt atsevišķas sistēmas daļas un saites. Līdz ar to šādu sistēmu izpētē plaši pielieto varbūtību teorijas un statistiskas metodes [WEA 2004]. Organizētas sarežģītības gadījumā, daļu skaits ir relatīvi liels, tomēr samērojams, salīdzinot ar neorganizētas sarežģītības gadījumu. Sistēmā ir organizēta sarežģītība, ja sistēmai piemīt īpašības un uzvedība, kas nav raksturīga atsevišķām sistēmas daļām, tas ir, sistēmai piemīt sistēmīpašības un sistēmuzvedība. Veicot sarežģītas organizētas sistēmas izpēti, būtiskāk ir apskatīt un izprast sistēmas īpašības un uzvedību, nevis sistēmas daļu skaitu [WEA 2004, VAZ 2009].

Jēdzienam „sarežģīta sistēma” nav vienotas definīcijas, jo izpratne par esošo terminoloģiju, kā arī tās interpretācija un pielietojums praksē ir atšķirīgs [SIM 1962, HOR

⁵ Neparasta, ārkārtēja parādība [OXF 2009]

1995, BAR 1997, VIC 2002, HAK 2006, SKY 2006]. Tomēr jēdzienam „sarežģīta sistēma” pastāv dažādi skaidrojumi, kas sniedz priekšstatu par sarežģītas sistēmas iezīmēm un par jēdzienu „sarežģīts” un „sistēma” kopsakarībām:

- Sarežģīta sistēma sastāv no daudzām un dažādām savstarpēji saistītām daļām, kas dinamiski mijiedarbojas dažādos veidos [SIM 1962, MAT 1974, GRU 1972, ROS 1979, ASH 1981, BAR 1997, WHI 1999, EDM 1999, RIC 2000, GLO 2002, HAK 2006, SKY 2006, JOH 2009]. Katrs elements tieši vai netieši ir saistīts ar citiem sistēmas elementiem [ACK 1971, GLO 2002].
- Sarežģītu sistēmu raksturo grūti analizējama struktūra, saites un daudzpusīgi funkcionalitātes kritēriji. Funkcionalitātes kritēriji ir nosacījumi, kas ļauj realizēt sistēmas mērķus [ROS 1979, BOU 2004].
- Sarežģīta sistēma ir telpiski un daudzveidīgi strukturēta sistēma [GRU 1972, BAR 1997, GOL 1999, HEY 2008], kuras struktūrai piemīt variācijas [GOL 1999]. Sarežģītai sistēmai ir daudzveidīga (tīklveida un hierarhiska) [HOR 1995, BAR 1997] un sadalāma (var veikt dekompozīciju) struktūra [HOR 1995]. Ar hierarhiju šeit ir domāts gadījums, kad starp elementiem vienā hierarhijas līmenī nav saišu (piemēru skatīt 1.2.1 nodaļā). Tīklveida struktūra atspoguļo funkcionālās saites starp sistēmas daļām noteiktā hierarhijas līmenī. Tīklveida un hierarhiska struktūra kopā veido sistēmas arhitektūru [YOU 2007]. Sistēmas iekšējās struktūras un saites mainās un mijiedarbojas [GLO 2002].
- Sarežģītas sistēmas daļas ir organizētas dažādos hierarhijas līmeņos, un pastāv daudzveidīgas saites gan starp atsevišķiem elementiem, gan starp dažādiem hierarhijas līmeņiem [SIM 1962, ROS 1979, BAR 1997, VIC 2002, SKY 2006, YOU 2007, HEY 2008]. Hierarhija ir savstarpēja pakārtotība (subordinācija) izpētes objektā, kas ir izveidota saskaņā ar kādu noteiktu aspektu vai paraugu [SIM 1962, BAR 1997, SKY 2006, HEY 2008, OXF 2009]. Katrā hierarhijas līmenī var saskatīt specifiska veida organizētību un/vai struktūru [HAK 2006, SKY 2006]. Sarežģītas sistēmas daļas un saites starp tām ir pakļautas variācijām un atlasei [HEY 2008].
- Sarežģītai sistēmai piemīt viena vai vairākas sistēmīpašības, kā arī šāda sistēma realizē noteiktas funkcijas un, neskatoties uz daļu daudzveidību, demonstrē vienotu uzvedību – sistēmuzvedību, kas kvalitatīvi atšķiras no atsevišķu daļu funkcijām un uzvedības [SIM 1962, HAL 1968, ACK 1971, GRU 1972, ROS 1979, BAR 1997,

JOS 2000, RIC 2000, MCC 2000, GLO 2002, VIC 2002, SKY 2006, YOU 2007, HEY 2008]. Katras daļas uzvedība ir relatīvi atkarīga no citu ar to saistīto daļu uzvedības. Pastāv savstarpēja saistība starp cēloņiem un sekām. Sistēmas uzvedība iespaido tās uzvedību nākotnē [SIM 1962, HAL 1968, GRU 1972, MCC 2000, GLO 2002, SKY 2006, SON 2008].

- Sarežģītā sistēmā mijiedarbības starp komponentēm ir nelineāras un uzvedība ir stohastiska [ROS 1979, RIC 2001, GLO 2002, VIC 2002, WOO 2004, SKY 2006]. Sarežģīta sistēma var izrādīt gan organizētu, gan neorganizētu uzvedību [JOH 2009]. Sistēmas uzvedību ietekmē tas, vai šādā sistēmā [HAK 2006, YOU 2007]:
 - *eksistē atgriezeniskās saites* [SIM 1962, HAL 1968, GRU 1972, SKY 2006, HEY 2008, JOH 2009]. Atgriezeniskā saite sistēmā ir stratēģija, kas ļauj sistēmai kompensēt negaidītus traucējumus [SKY 2006] un īstenot pašorganizēšanos, kā arī izvairīties no atgriešanās pie pastāvējušas organizētības formas, kas neapmierina funkcionalitātes kritērijus [MCC 2000];
 - *eksistē atmiņa* [JOH 2009];
 - *sistēma ir atvērta* [HOR 1995, JOH 2009]; Atvērta sistēma sadarbojas ar ārējo vidi [HAL 1968, ASH 1981, BAR 1997, MCC 2000, GLO 2002, SKY 2006, YOU 2007, HEY 2008] un to var ietekmēt ārējā vide [JOH 2009].
- Sarežģīta sistēma tiek raksturota kā spējīga pašorganizēties, adaptēties, attīstīties un mācīties [HAL 1968, BAR 1997, MCC 2000, RIC 2000, GLO 2002, WOO 2004, SKY 2006, HEY 2008]. Attīstība un pašorganizēšanās spēja, kā arī iepriekš iegūto zināšanu uzkrāšana, nodrošina iespēju sistēmai pielāgoties izmaiņām un mācīties [GRU 1972, BAR 1997, RIC 2001, GLO 2002, SKY 2006].
- Sarežģīta sistēma nepārtraukti attīstās un mainās laikā [VIC 2002, JOH 2009, APP 2011]. Sarežģītai sistēmai eksistē daudz dažādu stāvokļu, kuros tā spēj realizēt funkcionalitāti [MCC 2000, HEY 2008]. Sarežģīta sistēma var attīstīties dažādos veidos (angļu val. pathway) [WHI 1999, HEY 2008].

Sarežģītās sistēmās iezīmes ir savstarpēji saistītas. Piemēram, pašorganizēšanās procesā rodas jaunas iezīmes un pats process iekļauj atgriezeniskās saites, kas var būt par iemeslu sistēmas nelinearitātei [HEY 2008]. Rezumējot iepriekš dotos jēdziena „sarežģīta sistēma” skaidrojumus, promocijas darbā ir piemērots šāds jēdziena formulējums:

„Sarežģīta sistēma ir atvērta **sistēma**, kas ir **organizēta** noteiktā veidā, kurai ir **struktūra**, un kas sastāv vismaz no divām daļām, starp kurām eksistē daudzveidīgas saites,

turklāt daļas **savstarpēji mijiedarbojas**, kā rezultātā sistēmai piemīt **sistēmiezīmes** (tādas kā **sistēmīpašības** un **sistēmuzvedība**).”

1.2. Sarežģītu sistēmu modelēšana

Ja sistēma ir sarežģīta, tad galvenais izpētes līdzeklis ir sistēmas modelis [GRU 1972, BAR 1997, SKY 2006, IOP 2007, YOU 2007, SOK 2010]. Modelēšana ir process, kurā konstruē izpētes objekta atspoguļojumu jeb modeli. Veidojot modeli, izmanto abstrakciju, kurā ir vispārinātas, idealizētas reālās sistēmas būtiskās iezīmes, ignorējot nebūtiskās [AMA 2004, SKY 2006, YOU 2007]. Abstrakcija ir tehnika, ko lieto, lai aprakstītu vai lai iegūtu priekšstatu par izpētes objekta īpašībām, bez to konkrētas realizācijas [YOU 2007, BOP 2008]. Modelis ir nepieciešams [HOR 1995, BAR 1997, GAR 2001, SKY 2006, SOK 2010, MEN 2010]:

- 1) lai iegūtu un atspoguļotu zināšanas par izpētes objektu (nevis lai uzkrātu datus), kas ļautu sistēmu uztvert (kopumā), izprast, izpētīt – lai izprastu struktūru, organizētības principus un uzvedības paraugus;
- 2) lai varētu kontrolēti eksperimentēt ar sistēmu un tās elementiem (piemēram, mainot īpašības, iezīmes, parametrus), neriskējot ar reālo sistēmu. Tas ir īpaši svarīgi, ja reālā sistēma nav pieejama vai var tikt apdraudēta tās darbība, ja tā ir pakļauta izpētei. Eksperimentēšana ar sarežģītām sistēmām reālā pasaulē prasa daudz resursu (cilvēku, laika, materiālu) un ir pakārtota dažādiem riskiem;
- 3) lai pieņemtu lēmumus attiecībā uz sarežģītām sistēmām, kas prasa izpēti un analīzi;
- 4) lai veiktu sistēmas diagnostiku. Diagnostikas mērķis ir noskaidrot, kā sistēma funkcionē, un izdarīt spriedumus par defektiem un to izraisītajām sekām, kad sistēma darbojas kļūdaini [GRU 1993, GAR 2001];
- 5) lai iegūtu jaunas zināšanas par sistēmu un koplietotu esošās, kā arī projektētu jaunas sistēmas.

Lai izveidotu modeli, ir nepieciešamas zināšanas par izpētes objektu [GRU 1999, GAR 2001, CAR 2004, BOP 2008, SOK 2010]. Modeļi ir balstīti uz novērojumiem, kā arī uz pieejamās informācijas izvērtēšanu un spriedumiem [HOR 1995, SOK 2010]. Modeļi ir abstrakti [HEY 1990, STA 2006], un to izveidē piemēro dekompozīcijas, abstrakcijas un hierarhijas principus [YOU 2007, WEI 2009]. Sistēmas modelis apraksta un atspoguļo sistēmu no dažādiem skatu punktiem, ļauj izprast un analizēt tās uzbūvi, funkcionēšanu un

uzvedību, kā arī izvērtēt un pieņemt piemērotus risinājumus attiecībā uz reālās pasaules sistēmu un tās darbību [MIN 1975a, BAR 1997, GRU 1997b, RIC 2000, VIC 2002, SKY 2006, YOU 2007, SOK 2010].

Neskatoties uz to, ka jebkura sarežģīta sistēma ir cieši saistīta ar noteiktu problēmsfēru, sistēmu pētījumos lieto informācijas tehnoloģiju risinājumus [HOR 1995], veic datorizētu sistēmu analīzi, projektēšanu un modelēšanu [VIC 2002]. Lai izpētītu un izprastu sarežģītas sistēmas, ir jāveido modeļi pēc tādiem pašiem sarežģītības principiem, kādi piemīt pētītam objektam [MCC 2000, GAR 2001, HEY 2008]. Tomēr rezultātā iegūtajiem modeļiem ir jābūt izprotamiem, kompaktiem, adekvātiem, caurskatāmiem un atkārtoti izmantojamiem [LIN 1994, CAR 2004, STA 2006]. Ja sākotnēji ir izveidots reālai sistēmai neatbilstošs apraksts un/vai modelis, tas var novest pie kļūdām tālākā sistēmas izpētē un pie nepieciešamības veikt korekcijas, kas prasa papildus resursus [HOP 2007]. Neeksistē tāds modelis, kas pilnīgi un viennozīmīgi attēlotu reālās pasaules sistēmu [DUR 1994]. Jo modelis ir specializētāks, jo grūtāk to ir pielāgot iespējamām izmaiņām. Jo tas ir universālāks, jo mazāk piemērots noteiktu uzdevumu veikšanai [HEY 1990, STO 2001, YOU 2007, BOP 2008]. Lai izveidotu lietderīgu un atbilstošu sistēmas modeli, ir nepieciešami rīki, kuros ir implementētas modelēšanas metodes un tehnikas, kas ļauj tikt galā ar sarežģītības radītajām problēmām un ierobežojumiem un izgūt un sistematizēt pieejamās zināšanas [ROS 1979, GRU 1999, RIC 2000, MCC 2000, AMA 2004, SOK 2010]. Lietderīgs modelis ir tāds, kas ļauj īstenot izvirzītos mērķus [EDM 1999, SKY 2006, STA 2006].

1.2.1. Sarežģītas sistēmas un modelēšana

Ņemot vērā jēdziena „modelis” interpretācijas [KRI 1986, SIM 1987, EDM 1999, SKY 2006, STA 2006, ASH 2007, BOP 2008, OXF 2009, SOK 2010] turpmāk promocijas darbā autore izmanto šādu formulējumu:

„**Sistēmas modelis** ir izpētes objekta **apraksts** un/vai **atspoguļojums** no specifiskas **perspektīvas**, kas attēlo sistēmas morfoloģiju, funkcionalitāti, uzvedību vai citus modelēšanas mērķim būtiskus aspektus.”

Attiecībā uz modeļa izveidi eksistē: a) notācija (sintakse, valoda, atspoguļojums, likumi, ierobežojumi), kas ļauj izveidot modeli, aprakstot un formalizējot pieejamās zināšanas; b) process, kas ietver aktivitātes un metodes, kuras ļauj sistemātiski izveidot modeli; c) rīks (-i), tehnikas, kas atbalsta modeļu izveidi un ietver likumus, iezīmes, ierobežojumus attiecībā uz

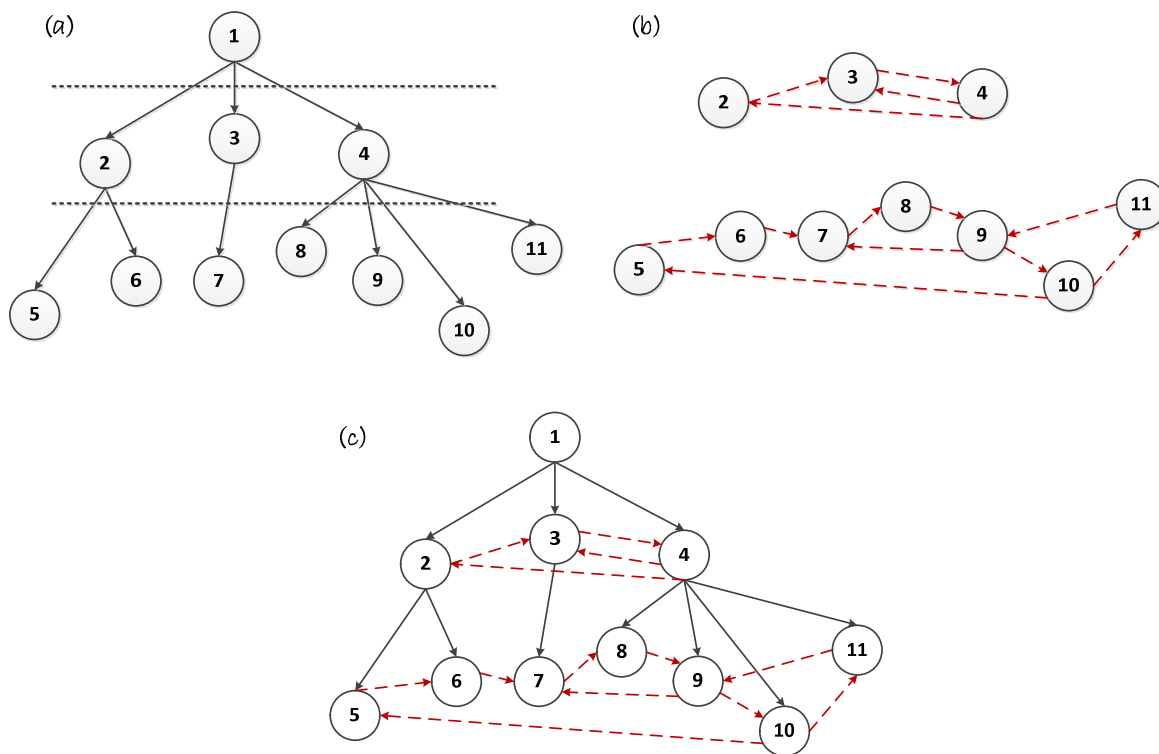
modeļiem [YOU 2007, SOK 2010]. To, kas ir jāmodelē un līdz ar to arī prasības pret nepieciešamo modeļa notāciju, nosaka izvēlētais modelēšanas mērķis [LIN 1994, BAR 1997].

Sarežģītās sistēmās ir būtiski tādi aspekti: kā daļu un saišu un raksturīgo īpašību skaits, to daudzveidība, sistēmas struktūra, organizētība, uzvedība [BAR 1997, SKY 2006, SNO 2010] un sistēmiezīmes [JOH 2009]. Jo ir vairāk elementu, jo pastāv lielāka varbūtība, ka, aprakstot elementus, netiek definētas visas tiem raksturīgās īpašības. Kā jau iepriekš minēts (skatīt 1.1. nodaļu), modelēšanu izmanto, lai tiktu galā ar sarežģītību, kas rodas pieaugot elementu, saišu, īpašību vai uzvedības paraugu un izmaiņu skaitam [EDM 1999, SKY 2006, YOU 2007]. Liels skaits elementu un/vai saišu norāda uz to, ka, veicot sistēmas izpēti, var piemērot dažādus kvantitatīvus mērus, lai noteiktu kopsakarības un paraugus sistēmā [BAR 1997, MCC 2000, YOU 2007]. Tomēr daļu no sarežģītu sistēmu īpašībām var kvantitatīvi mērīt (piemēram, daļu un saišu skaits), bet citas ir grūti vai pat neiespējami kvantitatīvi izmērīt (piemēram, mērķa sasniegšanas pakāpi) [BAR 1997, GRU 1997a]. Veidojot sarežģītas sistēmas modeļi, ir būtiski izvēlēties pieeju, kas ļauj atspoguļot lielu skaitu sistēmas objektu, kā arī izveidot modeļi nepilnīgas informācijas apstākļos.

Sarežģītu sistēmu daļas var būt vienkāršas un saliktas, kā arī gan sistēma un tās daļas var būt homogēnas un heterogēnas, [HAK 2006] un tām eksistē morfoloģija, kas iekļauj formu un struktūru [YOU 2007]. Līdz ar to ir nepieciešama pieeja, kas ļauj attēlot gan sistēmas elementus, gan komponentes, gan visu sistēmu kopumā, norādot to īpašības. Katra sistēmas daļa ir saistīta ar citām sistēmas daļām, veidojot vienotu veselu [HAL 1968, AMA 2004, SOK 2010]. Dažas sistēmas īpašības ir iespējams noteikt tikai, izpētot saistību starp sistēmas daļām [ACK 1971, HOR 1995, SKY 2006, JOH 2009, MOU 2009], līdz ar to iespējamās saistības un to īpašības ir jāspēj attēlot sistēmas modelī. Starp daļām var izdalīt divu veidu saites, un abos gadījumos, veido struktūru [SIM 1962, ROS 1979, UEN 1991, BAR 1997, GRU 1997a, SKY 2006, JAR 2006, YOU 2007, BOP 2008, WEI 2009]:

- Funkcionālās saites ir saites, caur kurām daļas mijiedarbojas un realizē noteiktu funkcionalitāti. Mijiedarbība atspoguļo saistību starp vienas daļas izeju un citas (vai tās pašas) daļas ieeju. Funkcionālās saites lieto, lai norādītu uz procesiem, darbībām, funkcijām, uzvedību, kas norisinās sistēmā, un ko bieži attēlo kā plūsmas. Sistēmas objektu un mijiedarbību atspoguļojums noteiktā detalizācijas līmenī attēlo organizētību sistēmā – atspoguļo tīklveida struktūru (skatīt 1.3. att. b gadījumu);
- Hierarhiskas saites ir saites starp sistēmas daļām, kas atspoguļo saistības starp hierarhijas līmeņiem, jeb to kā viena sistēmas daļa iekļauj un pakārto citas mazākas

daļas. Veidojot sistēmu modeļus, izmanto tādas hierarhiskās saites, kā: „ir” saite (angļu val. IS-A); ”piederības” saite (angļu val. HAS-A); „tipa” saite (angļu val. A-KIND-OF); „daļa no” saite (angļu val. PART-OF zināma arī kā „vesels/daļa” saite). Atspoguļo hierarhisku struktūru (skatīt 1.3. att. a gadījumu).



1.3. att. Saistības un struktūras

Neatkarīgi no sistēmas satura, darbības sfēras un izvēlētā skatījuma katrai sistēmai ir noteikta struktūra, kurai piemīt organizētība, paraugi un mērķi [AMA 2004, SKY 2006]. Veicot struktūras izpēti, tiek noskaidrota daļu loma sistēmā. Sistēmas daļas veido hierarhijas, kuras līmeņi ir savstarpēji saistīti un katru no tiem var apskatīt kā apakšsistēmu [SIM 1962, HAL 1968, UEN 1991, AMA 2004, HAK 2006, IOP 2007, SON 2008, MEN 2010]. Katrā sistēmas dekompozīcijas līmenī apskata noteiktas sistēmas komponentes un/vai elementus, nevis visas sistēmas daļas. Katram hierarhijas līmenim piemīt individuāla sarežģītība un tiek piemērota atšķirīga vadība un kontrole [BAR 1997, YOU 2007]. Izvēloties vienu hierarhijas līmeni, ir iespējams aplūkot sistēmas daļu kolekciju, kas sadarbojas, lai realizētu pašu daļu eksistencei un augstākiem līmeņiem nepieciešamo funkcionalitāti [YOU 2007]. Sarežģītās sistēmās komponentes mijiedarbojas arī starp dažādiem hierarhijas līmeņiem [ROS 1979]. Veicot dekompozīciju, ir jāņem vērā, ka, atspoguļojot tikai atsevišķas sistēmas daļas, var pazaudēt sistēmiezīmes, jo sarežģītās sistēmās daļas nav pilnīgi neatkarīgas. Izvēloties pieeju,

kas ļauj veikt modelēšanu, ir būtiski attēlot dekompozīcijas principu un sistēmas struktūru, un citas sistēmas iezīmes gan izvēlētajā hierarhijas līmenī, gan visai sistēmai kopumā.

Sarežģītās sistēmās hierarhija ir viens no galvenajiem sistēmas uzbūves pamatprincipiem [SIM 1962, BAR 1997, YOU 2007]. Hierarhiju var izveidot, izvēloties kādu aspektu no sistēmas, un tām nav noteikti jābūt sistēmas daļām – tās var būt arī, piemēram, funkcijas, ko īsteno sistēma. Ņemot vērā saišu veidu, kas izmantots hierarhijas izveidē, iegūst klašu („ir” saite, „tipa” saite) vai objektu hierarhiju / struktūru („daļa no” saite). Klašu hierarhiju lieto, ja nepieciešams apskatīt sistēmas daļu alternatīvas, atspoguļojot noteiktas klases ar līdzīgām īpašībām. Attēlojot sarežģītas sistēmas uzbūvi, parasti veido objektu jeb struktūras hierarhiju [BAR 1997, YOU 2007]. Sarežģītās sistēmās tipiska ir īpašību, funkciju, kā arī uzvedības pārņemšana no augstāka uz zemāku hierarhijas līmeni – hierarhijā augstāk stāvošs objekts var nodot iezīmes tam pakļautajiem objektiem [UEN 1991, BAR 1997, YOU 2007]. Līdz ar to, izvēloties modelēšanas rīku, ir nepieciešams, lai tajā implementētā pieeja atbalstītu mantošanas iespējas.

Lai izprastu sarežģītas sistēmas, veido struktūras jeb kauzālus modeļus [GRU 1997b, GRU 1999, MOU 2009]. Struktūras modelis atspoguļo vai nu sistēmas daļas, vai funkcijas, vai uzvedību un mijiedarbības starp atspoguļotiem vienumiem [GAR 2001, MOU 2009, MEN 2010]. Modelī, akcentējot morfoloģiskās un/vai funkcionālās iezīmes, ir iespējams attēlot [ROS 1979, GRU 1997a, EDM 1999, SKY 2006, YOU 2007]:

- gan sistēmas hierarhisko struktūru;
- gan sistēmas struktūru izvēlētajā detalizācijas līmenī, kas parasti atbilst vienam hierarhijas līmenim;
- gan hierarhijā atspoguļot sistēmas sadalījumu sīkākās daļās apvienojumā ar skatījumu uz daļām un to savstarpējām mijiedarbībām [SIM 1962].

Atspoguļojumu, kurā sistēma vienlaicīgi attēlota gan no morfoloģiskā (strukturālā), gan funkcionālā skatu punkta (skatīt 1.3. att. (c) gadījumu), apskata kā atsevišķu sistēmas modeļa veidu un sauc par sistēmas arhitektūru [YOU 2007, MEN 2010]. Morfoloģiskie un funkcionālie aspekti sistēmā ir cieši saistīti [GRU 1997a, APP 2011]. Sistēmas morfoloģiskās iezīmes ir saistītas ar sistēmu, tās daļām un saitēm, to daudzveidību un formu un struktūru telpiskā izpratnē. Funkcionālās iezīmes attiecas uz mijiedarbībām starp elementiem, kas ļauj apmainīties ar enerģiju, matēriju un informāciju, un ir saistītas ar sistēmas funkcijām un uzvedību, sistēmas struktūru laikā. Tās ir izmaiņas, plūsmas, attīstība un izaugsme [ROS 1979, GRU 1997a, YOU 2007]. Reakcijas un darbības ir saistītas ar uzvedību [ACK 1971]. Uzvedība ir veids kā tiek īstenotas funkcijas un apstrādāta enerģija, matērija un informācija

[GRU 1997a, MCC 2000, SKY 2006]. Sarežģītā sistēmā, īstenotās uzvedības ietekmē, var mainīties elementi un saites [MCC 2000, YOU 2007, APP 2011].

Lai sarežģītās sistēmās varētu izprast iespējamās funkcionalitātes izmaiņas un veikt diagnostiku, ir būtiski izpētīt uzvedību un ar to saistītās īpašības [GRU 1997b, GRU 1999, GAR 2001]. Arī gadījumos, kad nav iespējams veikt nepieciešamo analīzi, izmantojot tikai atspoguļotās daļas un likumsakarības starp tām, apraksta sistēmas uzvedību un citas sistēmas īpašības dažādos hierarhijas līmeņos un veic to modelēšanu [FER 2001]. Izvēloties piemērotu modelēšanas rīku, ir jāņem vērā sarežģītu sistēmu uzvedības nianšes – ir jāspēj izveidot jaunas un mainīt (labot, pārvietot, dzēst) atspoguļotās daļas, īpašības, struktūras, funkcijas, uzvedības stāvokļus, parametrus un citas iezīmes.

Modeļi var būt fiziski un simboliski. Fiziskie modeļi ir reāli un izskatās kā pētāmais objekts. Simboliskie modeļi ir abstrakti un lieto simbolus, lai atspoguļotu zināšanas par pētāmo objektu. Simbolus izmanto, lai atspoguļotu dažāda veida zināšanas, piemēram, faktus, konceptus un likumus. Simbolisko modeļu piemēri ir [HEY 1990, ACK 1971, STO 2001, NEG 2004, SKY 2006, BOP 2008, SOK 2010]:

- matemātiskie modeļi. Lai aprakstītu un izskaidrotu izpētes sistēmu, lieto matemātiskos simbolus;
- lingvistiskie modeļi jeb aprakstošie modeļi, kas attēlo sistēmu, lietojot dabīgo valodu (piemēram, loģiski – lingvistiskais modelis ir freimu tīkls [BOP 2008]);
- loģiskie jeb lēmumu pieņemšanas modeļi, kas attēlo sistēmas elementus, un starp tiem esošās cēloņseku saites.

Loģiskais modelis atspoguļo pētāmā objekta struktūru [YOU 2007]. Loģiskos modeļus var aprakstīt dabīgā valodā un atspoguļot grafiski [JOR 1999, MCC 1997]. Grafiskā veidā modelis ir vieglāk uztverams nekā tad, ja tas ir neformāli aprakstīts. Grafiskos modeļus attēlo orientētu vai neorientētu grafu veidā [JOR 1999, BOP 2008], jo grafu teorija ir vispārpieņemta struktūru apraksta valoda [GRU 1993, GRU 1997a] tādēļ, ka grafi ļauj atspoguļot un analizēt struktūru, funkcijas, uzvedību un cēloņseku saites. Grafus var izmantot gan hierarhisku, gan tīklveida struktūru atspoguļošanai [KRI 1986, GRU 1997a, SKY 2006, MOU 2009, SOK 2010]. Lai aprakstītu sarežģītas sistēmas ir nepieciešami dažādu tipu modeļi, bet tiem ir jābūt savstarpēji saistītiem [GRU 1997a, SOK 2010, MEN 2010].

Modeļos sakņotās sistēmas, salīdzinot ar likumos sakņotām sistēmām, ļauj labāk atspoguļot, izskaidrot un izprast sarežģītu sistēmu struktūru, funkcijas un uzvedību [GRU 1997b, GRU 1999]. Tomēr, lai izprastu sarežģītas sistēmas, nepietiek ar modeļu izveidi un attēlošanu, bet ir nepieciešama arī izgūto zināšanu apstrāde, kas atbalsta lēmumu pieņemšanu.

Līdz ar to izstrādā datorsistēmas, kas iekļauj dažādus atspoguļojuma veidus, un palīdz izprast, analizēt, projektēt un pieņemt lēmumus attiecībā uz sarežģītām sistēmām. Konstruējot izpētes objekta modeli datorsistēmā, izmanto zināšanas par pētāmo objektu un simbolus, kas apraksta zināšanu vienumus [HEY 1990]. Veidojot modeļus datorsistēmās, lieto aprakstus (formālus, neformālus), matemātiskos (matricas, kas ir struktūras atspoguļošanas veids, kurā ar skaitļu palīdzību attēlo struktūras elementu saistību), grafiskos kā arī dažādus hibrīdus modeļus [GRU 1997a, HOR 1995, MCC 2000, VIC 2002, IOP 2007].

Modelēšana datorsistēmā iekļauj vairākus saistītus etapus [GAR 2001, STO 2001, SKY 2006, YOU 2007, SOK 2010]:

- 1) Sistēmas izvēle, prasību jeb modelēšanas mērķa noskaidrošana – novērtē, kādam mērķim modelis paredzēts. Iezīmes, kas atbilst sarežģītām sistēmām, ir aprakstītas iepriekš. Tās ir būtiskas modeļa izstrādē un zināšanu izgūšanas un atspoguļošanas shēmas izvēlē.
- 2) Pieejas, metožu un rīku izvēle:
 - a. kas atbalsta modelēšanas mērķi;
 - b. kas atbalsta modelēšanas etapus.
- 3) Modeļa izveide – zināšanu formalizēšana:
 - a. vienkāršošana/ idealizēšana / abstrakcija – modelis satur galvenās sistēmas iezīmes. Detalizācijas līmeņa izvēle, kas atbilst sistēmas izpētes un modelēšanas prasībām;
 - b. zināšanu par sistēmu iegūšana/ aprakstīšana. Modeļa izveidi veic eksperts, kas izprot sistēmas objektus, to saistību un funkcionēšanas principus. Veidojot modeli, no eksperta ir jāizgūst zināšanas par sistēmu pēc noteiktas shēmas, un jāpastāv iespējai atspoguļoto atjaunot un to lietot atkārtoti. Modeļa izstrādē var izdalīt vairākus posmus [HEY 1990, SON 2008, SNO 2010]:
 - i. Izpētes objektam veic dekompozīciju, kuras laikā īsteno atsevišķu vienumu izdalīšanu, kuriem ir noteikta jēga un konteksts;
 - ii. Nosaka struktūru vai paraugu (ne tikai atsevišķiem vienumiem ir raksturīga noteikta jēga), ko veido iepriekš izdalītie vienumi un kas atbilst izpētes objektam;
 - iii. Atspoguļo zināšanas par atsevišķiem vienumiem un izpētes objektu kopumā;

- iv. Organizē izgūtās zināšanas, norādot katra vienuma nozīmi sistēmā un/vai loģiskās mijiedarbības secības, kas veido jēgpilnu aprakstu un ļauj spriest par izpētes objektu.
 - c. Veic manipulācijas/ transformācijas ar modeli; zināšanu saglabāšanu; jaunu zināšanu izgūšanu; saistīto modeļu automatizētu izveidi.
- 4) Modeļu pārbaude:
- a. attiecībā uz atbilstību izpētes objektam;
 - b. attiecībā uz spriešanas iespējām un iegūto zināšanu atbilstību izpētes objektam;
 - c. attiecībā uz zināšanu atkārtotas izmantošanas iespējām;
- 5) Izgūto zināšanu uzturēšana.

Atbilstoši iepriekš minētajam, lai izprastu sarežģītas sistēmas, ir jāveic to izpēte un modelēšana, ņemot vērā gan atsevišķu daļu uzbūvi, savstarpējās mijiedarbības un to skaitu, gan īstenoto uzvedību, gan arī kopējo sistēmas struktūru, darbības principus, likumsakarības un sistēmiezīmes. Ņemot vērā to, ka sarežģītu sistēmu izpēte ir multidisciplināra sfēra un katru no sarežģītām sistēmām apskata noteiktās sfēras kontekstā, piemērot vienādas modelēšanas metodes un pieejas visām sarežģītām sistēmām nav iespējams atšķirīgo iezīmju dēļ [HOR 1995, BAR 1997, RIC 2000, VIC 2002, SKY 2006]. Eksistējošie rīki un pieejas parasti ir izveidotas noteiktai problēmsfērai [GAR 2001], līdz ar to ir jāsašaurina izpētes sfēra. Turpmāk promocijas darbā kā pētījuma objektu apskata tikai tehniskas sarežģītas sistēmas, kas ir būtiskas informācijas un komunikācijas tehnoloģiju izpētē [MAG 2004, PEI 2011]. Sarežģītām tehniskām sistēmām bieži ir daudzveidīgas programmatūras un tehniskās komponentes, kas iekļauj elektroniskos, mehāniskos, hidrauliskos, pneimatiskos un/vai ķīmiskos elementus [KUR 1995, GRU 2002, MAG 2004, KHA 2010]. Ir izveidots daudz dažādu pieeju sarežģītu tehnisku sistēmu modelēšanai un diagnostikai. Tajā pašā laikā aprakstītās metodes un modeļi nav piemēroti tādu sarežģītu tehnisku sistēmu modelēšanai, kurās ir heterogēnas komponentes, dažāda veida (enerģijas, matērijas, informācijas) mijiedarbības starp komponentēm, daudzveidīgas struktūras un sarežģīta uzvedība. Šādām sistēmām nav iespējams izveidot vienotu matemātisko (analītisko) modeli [GRU 2002, ZEL 2008b]. Šo problēmu dažādās pieejās cenšas atrisināt, izveidojot vispārējus kvalitatīvus modeļus [FAN 1994, GRU 1997a, HAR 1990], vai vienas pieejas ietvaros, integrējot dažādu modeļu veidus [ABU 1994, GRU 2001]. Nav piemērota rīka, kas vienlaicīgi atbalstītu dziļu cēloņseku zināšanu izgūšanu un spriešanu par šādām sarežģītām sistēmām, vienotu zināšanu atspoguļošanu no morfoloģiskiem un funkcionāliem aspektiem, zināšanu bāzes uzturēšanu un

zināšanu koplietošanu starp vairākiem lietotājiem [UEN 1991, GRU 1997b, GRU 2002, ZEL 2008a, ZEL 2010a, ZEL 2010b]. Lai realizētu sarežģītu tehnisku sistēmu modelēšanu datorsistēmā, ir jāizmanto pieeja, kas ļauj īstenot iepriekš minētās prasības attiecībā uz sarežģītām sistēmām un to modelēšanu. Sarežģītu sistēmu specifika un pieejamais informācijas apjoms apgrūtina kopēja matemātiska apraksta izveidi un ir noteicošie faktori pieejas izvēlē. Tāpēc ir izvēlēta struktūrmodelēšanas pieeja, kas atbalsta sarežģītu tehnisku sistēmu modeļu izveidi, kvalitatīvu un kvantitatīvu sistēmu analīzi, un kuru var lietot arī diagnostikai [GRU 1997a, GRU 1997b, GRU 2002].

1.2.2. Sistēmu struktūrmodelēšanas pieeja

Struktūrmodelēšana (SM) ir sistemātiska, daļēji formāla pieeja, kas sākotnēji struktūras modeļos un freimos, un ir izveidota ar mērķi izgūt, atspoguļot un apstrādāt zināšanas par sarežģītām tehniskām sistēmām ar daudzveidīgiem elementiem un saitēm, nepilnīgas informācijas apstākļos, kā arī automatizēt zināšanu bāzes izstrādi [GRU 1993, GRU 1997b, GRU 1999, GRU 2002, ZEL 2010a]. SM ir izstrādāta Rīgas Tehniskajā universitātē pagājušā gadsimta 70. gadu sākumā, lietojot topoloģiskā modeļa [OSI 1969] koncepciju un pieejas autors ir J.Grundspenķis [GRU 1972, GRU 1993, GRU 1997a, GRU 1999].

Pieeja ir piemērota sarežģītu tehnisku sistēmu struktūras analīzē (kad sistēma jau eksistē), kā arī modelēšanas un projektēšanas (kad sistēmu izstrādā) etapos, lai spriestu par sistēmu un lai struktūrmodelēšanu pielietotu tehniskā diagnostikā [GRU 1993, GRU 1997a, GRU 1997b, GRU 1999, GRU 2002]. Pieeja pieder pie „ātrajām” sistēmu pētīšanas metodēm, kurā akcentē kvalitatīvu sistēmu analīzi [GRU 1993, GRU 1997a, GRU 1997b]. Izmantojot SM, pēta sistēmas topoloģiskās un funkcionālās īpašības, attiecībā uz apskatīto sistēmu, kā piemēram [GRU 1993, GRU 1997a, GRU 1997b]:

- dažādu saišu eksistence starp sistēmas daļām;
- elementu un saišu relatīvā nozīmība sistēmas struktūrā un funkcionēšanas procesā;
- seku novērtēšana gadījumos, ja elementi pārstāj darboties;
- uzvedības un parametru izmaiņu iemesli un izmaiņu radītās sekas.

Struktūrmodelēšanā ņem vērā četrus dažādus aspektus: struktūru, funkcijas, uzvedību un dziļas cēloņseku zināšanas [GRU 1999, GRU 2002], un pieejā ir integrētas divas dažādas

paradigmas⁶: morfoloģiskā un funkcionālā [GRU 1997b, ZEL 2011]. Tas ir izdarīts ar mērķi pārvarēt mākslīgi radīto atšķirību starp objektu un funkciju, lai būtu iespējams ar vienu matemātisko aparātu, no vienota skatupunkta veidot sistēmas morfoloģisko un funkcionālo atspoguļojumu [GRU 1993, GRU 1997a, GRU 1997b]. Skatoties no sistēmas uzbūves viedokļa, attēlo sistēmas daļas un saistības starp tām, turklāt, var apskatīt gan hierarhisku, gan tīklveida struktūru. Skatoties no funkcionālā viedokļa, specificē sistēmā iekļauto komponentu mērķi un reakcijas uz stimuliem, kurus komponentes tiecas realizēt pēc savas būtības. Uzvedība attiecas uz komponentu darbību izpausmes veidiem, ko specificē ar dažāda veida plūsmām (materiālu, enerģijas vai informācijas). Dziļu cēloņseku zināšanas nosaka uzvedību pie noteiktām strukturālām izmaiņām un atbalsta spriešanu par sistēmas darbību [GRU 1993, GRU 1999, ZEL 2010a]. Izmantojot struktūrmodelēšanu, ir iespējams aptvert [GRU 1997b, GRU 2002]:

- gan „seklās” zināšanas, kas atspoguļo ekspertu pieredzi. Tās ir subjektīvas zināšanas par izpētes objektu [UEN 1991, GRU 1997b], ko bieži attēlo produkciju likumu formā (IF (ja) nosacījums, THEN (tad) secinājums) [GRU 1999].
- gan „dziļās” jeb cēloņseku (kauzālās) zināšanas. Tās ir zināšanas par saitēm starp daļu ieejām un izejām, starp daļu funkcijām un to raksturojošiem parametriem, cēloņseku saistībām starp komponentēm, kā arī par uzvedības izmaiņām, ko rada sistēmas struktūras izmaiņas (ārējās vides ietekmē, iekšējo izmaiņu dēļ, pašorganizēšanās rezultātā) [GRU 1997b]. Tās ir objektīvas zināšanas par struktūru, uzvedību, principiem attiecībā uz izpētes objektu [UEN 1991].

SM zināšanu izgūšanai un atspoguļošanai ir izveidots ietvars jeb shēma (angļu val. framework), kas sastāv no 4 dažādiem, savstarpēji saistītiem struktūras modeļiem un freimu kopas [GRU 1993, GRU 1997a, GRU 1997b, GRU 1999, ZEL 2007]. SM lietotās zināšanu aprakstīšanas iespējas ļauj iekapsulēt eksperta zināšanas mazās, neatkarīgās, apvienojamās un sadalāmās zināšanu vienībās. Lai atspoguļotu zināšanas par sistēmas uzbūvi, veido morfoloģiskās struktūras modeļi (MSM), bet, lai aprakstītu sistēmas funkcionālās īpašības, konstruē funkcionālās struktūras modeļus (FSM). FSM tiek veidots funkciju, uzvedības un parametru telpās [GRU 1997a, GRU 1999, ZEL 2011]. Piemērojot SM aprakstītās formālās metodes, viena veida struktūras modeļi var transformēt cita veida struktūras modeļi (skatīt

⁶ Paradigma ir kopums, kas atspoguļo pēc noteiktiem principiem, iezīmēm, kritērijiem grupētus vienumus, kas ir savstarpēji saistīti [OXF 2009]

3.nodaļu). Izmantojot FSM parametru telpā, var veidot notikumu kokus. Notikumu koks attēlo notikumu, kurā ir redzama cēloņu un seku saistība, kas ļauj spriest par sistēmu, un to pielietot sistēmas diagnostikā [GRU 1997a, GRU 1997b]. Freimu kopa ir zināšanu atspoguļošanas shēma, kas ļauj izgūt eksperta zināšanas par sistēmu un veidot atsevišķu sistēmas daļu, kā arī visas sistēmas attēlojumu, aprakstam lietojot dabīgo valodu. Freimu kopu lieto arī zināšanu saglabāšanai zināšanu bāzē [VAL 2005a, ZEL 2007].

Struktūras modeļi atspoguļo deklaratīvas zināšanas par sistēmas uzbūvi un funkcionalitāti normālos funkcionēšanas apstākļos un kļūdu gadījumā [GRU 1999, GRU 2002]. Freimu kopa apraksta gan deklaratīvas, gan arī procedurālas zināšanas, ko var lietot sistēmas izmaiņu gadījumā [GRU 1999, ZEL 2007]. Deklaratīvās zināšanas ir saistītas ar faktiem un notikumiem par izpētes objektu. Tās ļauj izprast, kas tiek apskatīts. Procedurālās zināšanas ir saistītas ar darbībām, to demonstrēšanu, un ļauj izprast, kā kaut ko darīt. Procedurālās zināšanas apraksta, norādot saistības starp objektu ieejām un izejām, un attēlojot funkciju izpildi, atbilstošo uzvedību un tās izmaiņas [FRI 2011]. Aprakstot procedurālās zināšanas, izmanto likumus [FRI 2011].

Zināšanu izgūšanu un atspoguļošanu veic, realizējot noteiktu soļu secību, kas ir piedāvāti struktūrmodelēšanas metodoloģijā [GRU 1999, GRU 2002, ZEL 2007]. Zināšanu izgūšanas process ir zināšanu un pieredzes pārnese, parasti lietojot informācijas tehnoloģiju rīkus. Zināšanu izgūšanu realizē ar mērķi atrisināt problēmas, kuras ir saistītas ar izpētes objektu, un procesa rezultātā izveido aprakstu, kas ir cilvēkam un/vai datoram izprotamā formā [GRU 1999]. Struktūrmodelēšanā pirmais solis ir problēmsfēras apzināšana un zināšanu avotu (eksperts, grāmatas, rokasgrāmatas) identifikācija. Tradicionālā atspoguļošanas gadījumā zināšanu avots ir eksperts. Cilvēku zināšanas ir sarežģītas un slikti formulētas, nav pietiekami dokumentētas un aprakstītas/atspoguļotas [GRU 1999]. Struktūrmodelēšanā piedāvā zināšanas izgūt no eksperta, izmantojot zināšanu izgūšanas sistēmu [GRU 1993, GRU 1997b, GRU 1999].

Zināšanu izgūšana parasti ir vājākais posms, kad veido zināšanās sakņotu sistēmas atspoguļojumu. Tas ir uzskatāmi redzams, ja sarežģītas sistēmas aprakstu veido vairāki eksperti. Problēmsfēras ekspertu ieguldītais darbs ir laikietilpīgs un iegūto rezultātu interpretācija bieži vien nav viennozīmīga, kā arī ir grūti sasaistīt un saskaņot dažādos izveidotos modeļus [GRU 1999, NEG 2004, ZEL 2008b]. Iegūtās zināšanas saglabā zināšanu bāzē (ZB). Struktūrmodelēšanā ZB izstrāde balstās uz sistemātisku zināšanu iegūšanu un attēlošanu, kuras laikā veido/konstruē modeļus. Sistemātisks atspoguļojums ir sistēmas

elementu organizēšana un attēlošana modelī, pakāpeniski veidojot noteiktu struktūru [GRU 1997b].

Zināšanu bāze ir sadalīta divās daļās: topoloģiskā un dziļu zināšanu likumu (jeb cēloņseku) bāzē [GRU 1997a]. Topoloģiskā ZB satur zināšanas par morfoloģiskām un funkcionālām struktūrām. Dziļu cēloņseku ZB sastāv no kauzāliem jeb cēloņseku likumiem, kas atspoguļo kauzālās saites starp cēloņiem (kļūdām) un sekām (piemēram, izmaiņām parametros). Kauzālie likumi nodrošina izpratni par sistēmas darbībām un ir būtiski diagnostikā, kas ir viens no SM mērķiem. Kauzālos likumus var transformēt produkciju likumos [GRU 1997b, GRU 1999]. Implementējot SM datorsistēmā, šajā promocijas darbā tika mainīts zināšanu izgūšanas mehānisms, kā arī zināšanu bāze. Tas tika īstenots, lai datorā varētu automatizēti un vizuāli atspoguļot sistēmas struktūras modeļus, morfoloģisko un funkcionālo hierarhiju. Izmantojot freimu kopu un freimu sistēmu koncepciju, zināšanu bāze netiek sadalīta un visas zināšanas par atsevišķiem objektiem un to saitēm, kā arī sistēmu kopumā saglabā vienotā zināšanu bāzē [ZEL 2007].

Izgūtās zināšanas izmanto, lai automātiski izveidotu sistēmas struktūras modeļus, notikumu kokus un matricas [ZEL 2010a]. Struktūras modeļus lieto, lai atbalstītu secīgu analīzi, projektēšanu, spriešanu un lēmumu pieņemšanu attiecībā uz pētīto sistēmu, lai izgūtu jaunas zināšanas par to, kā arī diagnostikas problēmu risināšanā [GRU 1997a, GRU 1999, GRU 2002, GRU 2004, ZEL 2008b, ZEL 2011]. Tā kā atspoguļošanai izmanto grafus, veicot sistēmas izpēti, pamatojoties uz grafu teorijas metodēm, var piemērot topoloģisko, kvalitatīvo un kvantitatīvo sistēmu struktūras analīzi [GRU 1999]. Struktūras modeļi atbalsta loģisko spriešanu attiecībā uz struktūru, uzvedību un kauzālām saitēm. Loģiskā spriešanā, izmantojot nosacījumus, secinājumus un likumus, iegūst spriedumus par sistēmu, ko lieto, lai atbalstītu lēmumu pieņemšanu [GRU 1997a, GRU 2002]. Tomēr iepriekš struktūras modeļos netika attēloti loģiskie operatori (skatīt 2. un 3. nodaļu), kas loģiskā spriešanā ir būtiski. Lai iegūtu patiesus spriedumus un jaunas, nepretrunīgas zināšanas, ir jāizmanto patiesi, reālai situācijai atbilstoši fakti [HAR 1990, VAL 2005b, ZEL 2008b, FRI 2011]. Lai uzlabotu struktūras modeļu lasāmību un nodrošinātu viennozīmīgu izpratni par atspoguļoto sistēmas uzbūvi un funkcionalitāti, promocijas darbā modeļi ir papildināti ar loģiskiem operatoriem [ZEL 2008b, ZEL 2011]. Modeļi ļauj spriest par [GRU 1999, GRU 2002]:

- sistēmas struktūru un funkcionalitāti (strukturāla spriešana), ņemot vērā sistēmas daļas un saistības starp tām. Struktūras modeļi ļauj veikt ceļu un ciklu analīzi sistēmā, jo tie attēlo cēloņseku saites starp tajā esošām daļām [GAR 2001, MOU 2009];

- uzvedību (uzvedības spriešana), ņemot vērā plūsmu eksistenci starp sistēmas daļām;
- cēloņiem un sekām (kauzāla spriešana), ņemot vērā sistēmas daļu uzvedību un ieejas un izejas parametrus. Kauzālai spriešanai var būt divi veidi: diagnosticēšana (no sekām uz cēloņiem) un prognozēšana (no cēloņiem uz sekām) [GRU 1999].

Struktūrmodelēšanā aprakstītie spriešanas mehānismi nodrošina jaunu zināšanu izgūšanu par sistēmu normālos funkcionēšanas apstākļos (saistības starp komponentēm, to loma un nozīme sistēmas struktūrā) un kļūdu gadījumos (uzvedības izmaiņu iemesli un sekas, ko radījušas kļūdas) [GRU 1999].

Struktūrmodelēšana ir piemērota sarežģītu tehnisku sistēmu izpētei, bet, lai varētu atspoguļotās zināšanas rezultātīvi apstrādāt un analizēt (vienkopus par daudz un dažādiem elementiem, saitēm un īpašībām) un automātiski iegūt dažādus aprēķinus, spriedumus, kas atbilst aprakstītai sistēmai, pieeja ir jārealizē datorsistēmā. Attiecībā uz SM, ir izveidota automatizēta struktūru modelēšanas sistēma ASMOS [GRU 1993, GRU 1997a, GRU 1997b, GRU 1999], kas faktiski ir morāli un tehniski novecojusi, jo tai nav iespējams daudzlietotāju režīms, kā arī nav iespējams attēlot gan freimu kopu, gan modeļus. SM līdz šim nebija implementēta rīkā, kas atbalsta pieejas piedāvātās iespējas izmantot datorā, tai skaitā zināšanu izgūšanu, atspoguļošanu un apstrādi. Nebija iespējama automatizēta struktūras modeļu izstrāde, kā arī bija citi trūkumi:

- nebija iespējams attēlot sistēmas daļas, norādot vai tā ir komponente, vai elements;
- struktūras modeļos netika apskatīti loģiskie operatori, kā arī nebija izstrādātas atbilstošas modeļu transformācijas metodes, kurās lieto loģiskos operatorus;
- nebija iespējams datorizēti, lietojot vienu zināšanu bāzi, vairākiem ekspertiem veidot sistēmas aprakstu un analīzi;
- nebija iespējams atkārtoti izmantot izgūtās zināšanas citu modeļu izveidē.

Minētos trūkumus novērsa, darba ietvaros izveidojot papildus iespējas atspoguļojumā, pārveidojot pieejas piedāvātās iespējas datoram piemērotā formā un implementējot pieeju rīkā (skatīt 2. un 3. nodaļu). Struktūrmodelēšanas pieejai ir vairākas priekšrocības:

- pieeja ir piemērota sarežģītu tehnisku sistēmu struktūras modelēšanai nepilnīgas informācijas apstākļos, kad nav iespējams izveidot matemātisko modeli visai sistēmai. Nepilnīgas informācijas apstākļi ir situācija, kad sākotnējās informācijas apjoms par sistēmu nav pietiekams, lai izmantotu tradicionālās matemātiskās

shēmas, tomēr matemātisko aprakstu var izveidot atsevišķiem sistēmas elementiem [GRU 1993, GRU 1997a, GRU 1997b];

- pieeju var lietot intelektuālu diagnostikas sistēmu izveidē [GRU 1999, GRU 2002];
- pieejā izmantotais ietvars ļauj izpētīt dažādus sistēmas aspektus (morfoloģiskos, funkcionālos, uzvedības, parametrus, īpašības) no vienota skatījuma, un tas ļauj sistēmu labāk izprast. Tiek lietoti dažādi savstarpēji saistīti atspoguļošanas veidi (skatīt 2., 3. un 4. nodaļu): freimu kopa, dažāda veida struktūras modeļi, matricas, kas ļauj aprakstīt [GRU 1993, GRU 1997a, GRU 1997b, ZEL 2007, ZEL 2008b, ZEL 2010a, ZEL 2011]:
 - sistēmas hierarhisko un tīklveida struktūru un organizētību;
 - objektu un klašu hierarhiju;
 - fiziski daudzveidīgus elementus un komponentes, dažādas saites starp tiem un dažādas daļu un saišu īpašības;
 - loģiskās saistības starp plūsmām, izmantojot loģiskos operatorus;
 - sistēmas daļu uzvedību un ar to saistītos parametrus;
 - atsevišķas sistēmas komponentes un to īpašības, kā visu sistēmu kopumā un sistēmiezīmes;
- freimu kopā lieto dabisko valodu, kas ir saistīta ar automātiski piešķirtiem apzīmējumiem [ZEL 2007, ZEL 2010a];
- modelēšanā izmanto grafisku notāciju, struktūras modeļi ir formāli un vizuāli viegli uztverami [GRU 1993, GRU 1997a, GRU 1997b];
- struktūras modeļi satur visus datus, kas ir nepieciešami topoloģisko īpašību analīzei un spriešanai par struktūru [GRU 1993, GRU 1997b];
- pieeja ļauj atspoguļot, saglabāt, izgūt jaunas zināšanas, kā arī lietot zināšanas modelēšanas, analīzes, projektēšanas un diagnostikas mērķiem [GRU 1993, GRU 1997a, GRU 1997b, GRU 2002];
- pastāv metodes struktūras modeļu izveidei un transformācijām starp modeļiem [GRU 1999, GRU 2002], kā arī metodes, piemēram cēloņseku likumu izgūšanai no ZB [GRU 1997b], sistēmu struktūru analīzei [GRU 1997a, GRU 1997b], likumu izgūšanai par plūsmu un elementu saistībām, izmantojot loģiskos operatorus [ZEL 2008b, ZEL 2011];
- lietojot pieeju, var izveidot alternatīvus modeļus un veikt to novērtēšanu [GRU 1997b].

Ņemot vērā SM piedāvātās zināšanu izgūšanas, atspoguļošanas un modelēšanas iespējas, zināšanu bāzes un spriešanas mehānismu nepieciešamību, sarežģītu sistēmu īpašības un modelēšanas prasības, ir vēlams izmantot dažādas mākslīgā intelekta metodes, jo tās atbalsta zināšanu apstrādi un lēmumu pieņemšanu. Lai tās implementētu, ir jāveido intelektuāla datorsistēma, kas ietver ekspertu sistēmu un zināšanās sakņotu sistēmu iezīmes un ļautu iegūt jaunas zināšanas par sistēmu no eksperta atspoguļotajām zināšanām. Datorsistēmas lietošana paātrina uzdevumu izpildes laiku un ļauj automatizēt dažādas aktivitātes, vieglāk nekā nedigitalizēta risinājuma gadījumā, veikt izmaiņas sistēmas aprakstā – pārnest, koplietot, atkārtoti izmantot, integrēt, apvienot, atjaunināt izgūtās zināšanas no vairākiem avotiem un dažādos laika momentos, kā arī saglabāt izmaiņu vēsturi [GRU 1999, YOU 2007, ZEL 2010b, RUS 2010].

Intelektuāla sistēma, kurā implementēta struktūrmodelēšanas pieeja, salīdzinājumā ar diagnostikas ekspertu sistēmām, ļauj izgūt no eksperta „dziļas” zināšanas par sistēmu. Izgūtās zināšanas, ļauj saskatīt likumsakarības attiecībā uz sistēmu no dažādiem skatupunktiem un dažādos laika momentos, un tās var pielietot, lai spriestu par sistēmas uzbūvi un skaidrotu tās darbības un uzvedību. Intelektuālā sistēmā zināšanu bāzei ir jābūt organizētai tā, lai tā atbalstītu nepieciešamo spriešanu par sistēmu. Struktūrmodelēšanā izmanto freimus, kas nodrošina sistēmas struktūras modelēšanas mērķim atbilstošu zināšanu izgūšanu, atspoguļošanu un saglabāšanu zināšanu bāzē, kā arī kopā ar struktūras modeļiem nodrošina spriešanas mehānismu īstenošanu [GRU 1999].

1.3. Intelektuālas sistēmas un to īpašības

Jēdzienu „intelektuāla sistēma” plaši lieto mākslīgā intelekta (MI) jomā attiecībā uz sistēmām, kurās lieto MI tehnikas un metodes. Pateicoties intelektuālu sistēmu pielietojumam, tas ir kļuvis nozīmīgs arī dažādās cilvēku darbības sfērās, kurās ir nepieciešams atbalsts zināšanu apstrādei, apmācībai, problēmu risināšanai un lēmumu pieņemšanai, izmantojot zināšanas par izpētes objektu [NEG 2004, YOU 2007, RUS 2010]. Sistēma var būt intelektuāla tikai saistībā ar tās mērķi [ASH 1981, POL 2002]. Intelektuālas sistēmas ir paredzētas, lai atvieglotu lēmumu pieņemšanu, bet gala slēdzieni parasti pieņem sistēmas lietotājs. Intelektuālas sistēmas parasti lieto dažādu uzdevumu veikšanai, kas atbalsta lietotāja lēmumu pieņemšanu [DUR 1994, BRO 1998, NEG 2004, BOP 2008], piemēram:

- diagnostikai – analizējot situāciju un objekta uzvedību, nosaka diagnozi, ja objekts funkcionē nepareizi (kļūdu gadījumos), un rekomendē risinājumus;
- izvēles izdarīšanai/selekcijai – rekomendē labāko risinājumu no iespējamo alternatīvu saraksta;
- prognozēšanai – prognozē objekta iespējamo uzvedību nākotnē, ņemot vērā tā uzvedību pagātnē. Izdara secinājumus par esošo situāciju;
- klasificēšanai – objektu iedalīšana klasēs pēc izvēlētiem kritērijiem;
- apkopošanai (angļu val. clustering) – heterogēnas objektu grupas sadalīšana homogēnās apakšgrupās;
- optimizācijai – risinājumu kvalitātes uzlabošana, līdz atrod optimālu risinājumu;
- kontrolei/ pārraudzībai – objektu uzvedības noteikšana un pārvaldība, lai izpildītu prasības attiecībā uz reālo sistēmu noteiktā laika momentā. Salīdzina pieejamo informāciju par sistēmas uzvedību ar to uzvedību, kas ir norādīta kā nepieciešama, lai sistēma spētu darboties;
- projektēšanai – sasaistīt un izvietot objektus, ņemot vērā ierobežojumus un likumus;
- apmācībai, instruēšanai – nodrošina apmācību un kalpo kā metodiskais līdzeklis par noteiktu tematiku;
- interpretācijai – veido izpratni par noteiktu situāciju, izmantojot pieejamos resursus;
- plānošanai – organizē sistēmas darbības, lai sasniegtu noteiktus mērķus, ņemot vērā uzdotos ierobežojumus;
- modelēšanai/ imitēšanai – veido izpētes objekta modeli, ko var lietot, lai pētītu sistēmas darbību, pie dažādiem nosacījumiem.

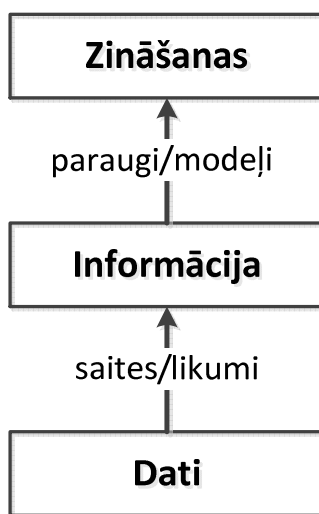
Ņemot vērā plašo pielietojuma sfēru, jēdzienu „intelektuāla sistēma” skaidro dažādi, bet tam nav vienotas definīcijas [RUD 2008]. Sistēmu uzskata par intelektuālu, ja tā spēj sasniegt cilvēkam līdzīgu veikspēju kādā izziņas uzdevumā vai spēj atveidot vienu vai vairākus cilvēku uzvedības aspektus: apstrādāt zināšanas, mācīties, adaptēties, spriest, pieņemt lēmumus, sadarboties [MCC 2000, NEG 2004, YOU 2007, RUD 2008]. Intelektuāla sistēma ļauj izgūt zināšanas no eksperta par izpētes objektu, tās organizēt, izmantot un uzturēt (atjaunināt atbilstoši reālajai situācijai) [NEG 2004, YOU 2007, BOP 2008]. Lai izveidotu intelektuālu sistēmu, ir jānosaka tās īpašības, kas ir saistītas ar intelektuālas sistēmas uzbūvi un darbības mehānismiem, un ir būtiskas šādas sistēmas projektēšanā un izstrādē.

1.3.1. Intelektuālas sistēmas uzbūve

Intelektuālas sistēmas uzbūvi veido un tās funkcionalitāti nodrošina četras komponentes: zināšanu bāze, izvedumu mašīna, datu bāze un lietojums [BIE 1991, DUR 1994, LIE 1997, PEA 2002, NEG 2004, BOP 2008]. Intelektuālās sistēmās starp zināšanu un datu bāzi pastāv cieša saistība [UEN 1991].

Bieži vien literatūrā saistībā ar sistēmām, kas izmanto zināšanu un datu bāzes, lieto tādus jēdzienus kā dati, informācija un zināšanas. Minētie jēdzieni ir savstarpēji saistīti, tomēr to nozīme ir atšķirīga un starp tiem nevar likt vienādības zīmi (skatīt 1.4. att.).

Dati ir fakti, bez konteksta, kas neinformē un līdz ar to tā nav informācija. Dati ir neapstrādāts materiāls, ko var izmantot, lai veidotu izpratni par kaut ko. `19102006` ir dati, kuriem var būt dažādas nozīmes, piemēram, tas var būt datums vai identifikatora numurs. Informācija ir dati, kuriem ir piešķirts konteksts - tie ir pārveidoti dati, kuriem ir nozīme. Piemēram, ja tabulā ir kolona ar nosaukumu „identifikators”, tad tai pašā kolonā esoša simbolu virkne `19102006` iegūst noteiktu nozīmi. Zināšanas no informācijas atšķir sarežģītība, kas izpaužas, kad tās ir jāizgūst. Ja informāciju var iegūt, piemēram, tikai sasaistot divus jēdzienus, tad zināšanas jau ir informācija, kas ir organizēta noteiktā struktūrā [SHE 2000, CAR 2008]. Zināšanas ir mērķtiecīga informācijas un datu kombinācija, kurai ir noteikts konteksts. Zināšanas atspoguļo izpratni par izpētes objektu [NEG 2004]. Intelektuāla sistēma izmanto gan datus un informāciju no datu bāzes, gan zināšanas no zināšanu bāzes, lai nodrošinātu atbilstošu funkcionalitāti.



1.4. att. Dati, informācija, zināšanas

Zināšanu bāze (ZB) satur zināšanas, ko iegūst no eksperta par pasauli un/vai noteiktu izpētes objektu un saglabā kā faktu, likumu, konceptu un starp tiem esošo saistību kopumu/kolekciju, kas ir lietderīga noteiktu problēmu risināšanai [UEN 1991, DUR 1994, LIE 1997, PEA 2002, NEG 2004, BOP 2008]. Zināšanu bāzē zināšanas ir skaidri izteiktas ar formālām definīcijām, aksiomām un ierobežojumiem [PEA 2002]. Attiecībā uz zināšanu bāzi lieto divu veidu atmiņu [LIE 1997, POL 2004]:

- īstermiņa atmiņa, kas ir kombinēta ar uztveri un satur datus;
- ilgtermiņa atmiņa, kas satur zināšanas (datus un informāciju organizētā veidā). Doto atmiņu sadala pielietojuma ZB (angļu val. Application Knowledge base) un spriešanas ZB (angļu val. Reasoning Knowledge base). Pielietojuma ZB satur deklaratīvas zināšanas un zināšanas par saistībām.

Zināšanu bāzi apstrādā ar izvedumu mašīnas palīdzību, kas ļauj secināt, izgūt zināšanas par izpētes objektu un tā īpašībām, kaut gan tiešā veidā šādas zināšanas ZB nepastāv [PEA 2002, BOP 2008].

Izvedumu mašīna darbojas ar pieejamo informāciju, kas glabājas datu bāzē un zināšanām, kas glabājas ZB, brīdī, kad sistēmas lietotājs veic darbības ar sistēmu. Mēģinot atbildēt uz lietotāja veiktajām darbībām, sistēma dinamiski kombinē faktus un likumus, lai izdarītu secinājumus, iegūtu risinājumus un sniegtu ieteikumus, kas atbalsta lietotāja lēmumu pieņemšanu [DUR 1994, LIE 1997, PEA 2002, NEG 2004]. Izvedumu mašīna sasaista arī zināšanu bāzē esošos likumus ar datiem un informāciju, kas glabājas datu bāzē [LIE 1997, NEG 2004].

Datu bāzē glabājas strukturētu un indeksētu datu kolekcijas – faktu kopas, pierādījumi, dokumenti, hipotēzes, mērķi [UEN 1991, NEG 2004]. Zināšanas no datu bāzes var iegūt netieši, veicot procedūras un izmantojot dažādas tehnikas [PEA 2002, BOP 2008]. Zināšanas izgūst no datu bāzēm, lietojot arī datizraces (angļu val. data mining) metodes [FAY 1996, POD 2012]: klasifikāciju (angļu val. classification), regresiju (angļu val. regression), grupēšanu klasteros (angļu val. clustering), apkopošanu (angļu val. summarization), saistību modelēšanu (angļu val. dependency modelling) un izmaiņu un neprecizitātes/kļūdu noteikšanu (angļu val. change and deviation detection). Datizraces algoritmos lieto attēlojumu modeli [FAY 1996].

Lai nodrošinātu mijiedarbības starp lietotāju un/vai ekspertu un intelektuālu sistēmu, lai izgūtu, atspoguļotu, saglabātu un apstrādātu zināšanas lietotājam uzskatāmā un saprotamā veidā, izmanto lietojumu jeb saskarni [DUR 1994, LIE 1997, NEG 2004, BOP 2008]. Lietojums ir tehnisko un programmatūras līdzekļu kopums, kas nodrošina komunikāciju ar

lietotāju dabīgā valodā. [DUR 1994, BOP 2008]. Lietojums intelektuālu sistēmu sasaista arī ar citām ārējām sistēmām, datubāzēm, algoritmiem un programmatūras līdzekļiem, kas atbalsta tās darbību [LIE 1997]. Intelektuālā sistēmā var izdalīt arī skaidrojošu komponenti, kas sniedz skaidrojumus par sistēmas veiktajiem spriedumiem [DUR 1994, LIE 1997], tomēr parasti dotā komponente ir intelektuālas sistēmas lietojuma sastāvdaļa [BOP 2008].

1.3.2. Intelektuālas sistēmas darbības mehānismi

Lai nodrošinātu intelektuālas sistēmas darbību, izmanto dažādas mākslīgā intelekta tehnikas un metodes [DUR 1994, LIE 1997, BRO 1998, NEG 2004, BOP 2008, RUS 2010, HOP 2012]. Tehnikas ir tie līdzekļi (resursi, rīki un paņēmieni), ko lieto sistēmā nepieciešamās funkcionalitātes un noteiktu mērķu īstenošanai, kā arī, ar kuriem sistēma sadarbojas ar savu vidi [BOP 2008]. Intelektuālas sistēmas var īstenot dabīgās valodas apstrādi, zināšanu atspoguļošanu un apstrādi, kā arī mašīnāpmācību, un var iekļaut neironu tīklus un citas tehnikas [NEG 2004, RUS 2010]. Ja intelektuālā sistēmā ir kombinētas dažādas tehnikas un metodes, tad to sauc par hibrīdu intelektuālu sistēmu [LIE 1997, NEG 2004, RUD 2008]. Intelektuālās sistēmās var izmantot [RUD 2008, HOP 2012]:

- skaitļošanas jeb aparatūras tehnikas, kas ir saistītas ar MI tehnikām. Skaitļošana ir orientēta uz analīzi un sistēmas uzbūves un procesu projektēšanu. Tās galvenās iezīmes ir precizitāte, formalitāte un kategorizēšana. Skaitļošana ir balstīta uz bināro loģiku, skaitlisku/ kvantitatīvu analīzi, varbūtību teoriju, diferenciālvienādojumiem, funkciju analīzi un matemātisko programmēšanu.
- programmatūras tehnikas ir sakņotas datorizētā/programmatūras intelektā (angļu val. Computational intelligence). Tās ir saistītas ar MI metodēm, un ir orientētas tieši uz intelektuālu sistēmu analīzi un izstrādi. Tās ir balstītas uz nestriktu loģiku, mākslīgiem neironu tīkliem, varbūtīgu spriešanu, iekļaujot ģenētiskos algoritmus, mašīnāpmācību. To galvenās iezīmes ir saistītas ar aproksimāciju, darbībām un uzvedību.

Ņemot vērā pielietotās tehnikas, var izdalīt zināšanās sakņotas (piemēram, likumos sakņotas, freimos sakņotas sistēmas, intelektuāliem aģentiem), programmatūras intelektā sakņotas (izmanto neironu tīklus, ģenētiskos algoritmus) un hibrīdas sistēmas (lieto nestriktu loģiku), un tās visas var attiecināt uz intelektuālām sistēmām [HOP 2012]. Kā būtiskākās no intelektuālo sistēmu tehnikām tiek minētas [DUR 1994, BRO 1998, NEG 2004, RUS 2010]:

- dabīgās un mākslīgās valodas apstrāde, kas ļauj sistēmai komunicēt ar tās lietotāju un attēlot zināšanas gan cilvēkam, gan datoram saprotamā veidā. Mākslīgā valoda ir zīmju sistēma, kas ļauj precīzi un ātri apstrādāt datus;
- zināšanu izgūšana un atspoguļošana. Atspoguļošana ir kaut kāda veida organizācija, kas kā viens vesels vai kā daļa, atbilst, ir saistīts ar, aizstāj, simbolizē, vai īpašā veidā sadarbojas, vai citādāk attēlo izpētes objektu [PAL 1978]. Lai zināšanas varētu ne tikai aprakstīt, bet arī izmantot atkārtoti un ilglaicīgi, tās ir jāizgūst no zināšanu avota, jāorganizē un jā saglabā saskaņā ar noteiktu zināšanu atspoguļošanas shēmu. Zināšanas var aprakstīt, lietojot [LUG 2005]: a) loģiskās zināšanu atspoguļošanas shēmas (piemēram, izteikumu rēķini, predikātu rēķini); b) tīklveida zināšanu atspoguļošanas shēmas (piemēram, semantiskie tīkli, konceptuālie grafi); c) strukturizētas zināšanu atspoguļošanas shēmas (piemēram, freimi, scenāriji); d) procedurālas zināšanu atspoguļošanas shēmas (piemēram, produkciju sistēmas);
- mašīnāpmācība ir mehānisms, kas ļauj datorizētai sistēmai mācīties no pieredzes, paraugiem, piemēriem un analogijām. Mācīšanās spējas uzlabo intelektuālas sistēmas darbību, tā ļauj adaptēties jaunu zināšanu, izmaiņu un kļūdu gadījumā. Mašīnāpmācība ir adaptīvu sistēmu pamatā. Populārākās mašīnāpmācībā izmantotās tehnikas ir lēmumu koki, neironu tīkli un ģenētiskie algoritmi.
- automatizēta spriešana un secināšana. Kvalitatīva spriešana, ļauj spriest par plūsmām, darbībām un uzvedību, ņemot vērā struktūru, mijiedarbības starp objektiem un plūsmu parametrus. Spriešanu veic, lietojot zināmu likumsakarību kopu (paraugus, modeļus, likumus, ierobežojumus). Uz gadījumiem vai paraugiem balstītā spriešanā izmanto zināšanas par iespējamo struktūru, darbībām pagātnē vai arī par plānotām darbībām, lai iegūtu esošās situācijas novērtējumu. Lieto likumus, pārmeklēšanu, lai skaidrotu secinājumu iegūšanas ceļus. Var īstenot divas spriešanas stratēģijas:
 - no mērķa virzīta, ja ir zināmas sekas – noteikts gadījums ar specifisku īpašību kopu, kuram var būt daudz iespējamo cēloņu;
 - no datiem virzīta, ja ir zināmi sākuma nosacījumi un iespējamie rezultāti, un apskata risinājumu ceļus.

- meklēšana/ pārmeklēšana, ko veic, lai paplašinātu esošās zināšanas. Meklēšanu īsteno, nezinot kāds precīzi būs iegūtais rezultāts, jo ir zināmi tikai sākuma kritēriji.

Meklēšanu realizē:

- meklējot pēc noteikta parauga, struktūras, lai pārbaudītu vai pastāv paraugam līdzīgi izpētes objekti;
- meklējot pēc konteksta, lai izprastu jēdzienu, izpētes objektu saistību un būtību;
- meklējot līdzības un atšķirības objektos (salīdzinot), lai izvēlētos labāko.

Pārmeklēšana ir metode, ko lieto, lai pārmeklētu stāvokļu telpu un iegūtu/ atrastu mērķa stāvokli, atrodot atrisinājuma ceļu no sākuma uz mērķa stāvokli. Izdala informētu un neinformētu pārmeklēšanu.

Ņemot vērā intelektuālas sistēmas īpašības un tās mērķi, jēdzienu „intelektuāla sistēma” promocijas darbā definē šādi:

„**Intelektuāla sistēma** ir zināšanās sakņota datorizēta sistēma, (1) kas operē ar organizētām zināšanām, (2) izmanto vienu vai vairākas mākslīgā intelekta tehnikas un metodes, un (3) to var lietot sarežģītu sistēmu struktūrmodelēšanas mērķiem.”

1. nodaļas kopsavilkums un secinājumi

Pirmās nodaļas mērķis ir noteikt sarežģītu sistēmu īpašības, kas ir būtiskas to izpētes un modelēšanas procesā, kā arī atbilstošas datorizētas sistēmas izstrādē, kas atbalsta sarežģītu tehnisku sistēmu struktūru modelēšanu. Galvenie rezultāti ir:

- skaidroti jēdzieni “sistēma”, “struktūra”, “organizētība”, “modelis” un “sarežģītība” un dota sarežģītas sistēmas definīcija;
- aprakstītas sarežģītu sistēmu galvenās īpašības (daudz un dažādas daļas un mijiedarbības veidi; daudzveidīga un sadalāma struktūra; gan sistēmai, gan atsevišķām tās daļām piemīt īpašības un uzvedība);
- norādītas prasības, kuras ir jāņem vērā, izvēloties sarežģītu sistēmu modelēšanai piemērotu pieeju un/vai rīku.
- raksturota struktūrmodelēšanas pieeja, tās iespējas un trūkumi un pieejas piemērotība izvēlētajam mērķim – sarežģītu sistēmu struktūrmodelēšanai;
- izpētītas intelektuālu sistēmu pielietojuma iespējas, galvenās iezīmes un darbības mehānismi, kā arī sniegta intelektuālas sistēmas definīcija.

Galvenie pirmās nodaļas secinājumi ir šādi:

- Sarežģītām sistēmām ir daudz savstarpēji saistītu iezīmju un, lai realizētu prasības, ko izvirza sarežģītu sistēmu struktūru modelēšana, ir jāizvēlas piemērota pieeja, kas ļauj izveidot lietderīgus modeļus;
- Struktūrmodelēšana ir piemērota sarežģītu tehnisku sistēmu struktūru modelēšanai, tomēr, lai īstenotu pieejas iespējas un automatizētu zināšanu izgūšanu un apstrādi, ir nepieciešama intelektuālas sistēmas izstrāde;
- Sistēmai, kurā ir implementēta struktūrmodelēšanas pieeja, ir jābūt hibrīdai intelektuālai sistēmai, kurā ir nepieciešams izmantot dažādas tehnikas, jo tas sistēmai sniedz priekšrocības gan zināšanu atspoguļošanā, gan to apstrādē.

2. STRUKTŪRMODELĒŠANAS PIEEJAS APRAKSTS

Struktūrmodelēšana (SM) atbalsta problēmsfērā sakņota, daļēji formāla sistēmas atspoguļojuma izveidi. Sistēmas struktūra, funkcijas un uzvedība ir savstarpēji saistīti vienumi un ietekmē viens otru. Tāpēc ir nepieciešama dažādu modeļu izmantošana, kas veido vienotu skatījumu uz sistēmu no dažādiem skatupunktiem. SM ļauj izveidot vizuālus, viegli izprotamus struktūras modeļus, kas apraksta gan izpētes sistēmas morfoloģiskos, gan funkcionālos aspektus. SM ir ieviesti: morfoloģiskās struktūras modelis (MSM) un trīs funkcionālās struktūras modeļi (FSM). FSM var izveidot funkciju (FSM FT), uzvedības (FSM UT) un parametru (FSM PT) telpās [GRU 1997a, ZEL 2008b, ZEL 2010a, ZEL 2011].

SM modeļu izstrādes procesa atbalsts ir formālās transformācijas un detalizēti aprakstīti spriešanas mehānismi. Piemērojot SM formālās metodes, morfoloģiskās struktūras modeli transformē funkcionālās struktūras modelī izvēlētajā telpā, un to veic daļēji automatizēti. Transformācijas ļauj izveidot atbilstīgus (nepretrunīgus) modeļus un nodrošina nepārtrauktu sistēmas atspoguļojumu [GRU 1972, GRU 1993, GRU 1997a, GRU 1997b, GRU 2002, ZEL 2008b, ZEL 2011]. Modeļu transformācijas ir labi zināma tehnika, ko piemēro, lai modificētu un izvērstu modeļus, saglabājot noteiktas iezīmes un atspoguļojot sistēmu no dažādiem skatupunktiem: morfoloģiskā, funkcionālā, uzvedības un citiem aspektiem. Attiecībā uz sarežģītām sistēmām plaši lieto grafu transformācijas [MEN 2010].

Lai SM varētu pilnvērtīgāk izmantot sarežģītu tehnisku sistēmu modelēšanā, promocijas darbā ir pilnveidoti pieejā lietotie pamatelementi, kā arī ir pievienoti papildus elementi. Pamatelementu un modeļu vizualizācija ir veidota, ņemot vērā struktūrmodelēšanas aspektus (struktūru, funkcijas, uzvedību un dziļas cēloņseku zināšanas) un modelēšanas mērķi. SM izmanto dažādus atspoguļošanas veidus, līdz ar to arī pamatelementi ir daudzveidīgi. Tomēr to pieraksta veids un nelielais skaits ļauj izveidot sistēmas atspoguļojumu, ko tālāk lieto sistēmas izpētē, analizē un projektēšanā. Lai sniegtu izpratni par to, kādas zināšanas var atspoguļot struktūras modeļos, kādi ir SM pieejas ierobežojumi un iespējas, tālāk ir dots modeļu, to pamatelementu, kā arī transformāciju detalizēts apraksts un vizualizācija.

2.1. Struktūras modeļu sintakse un semantika

Struktūras modeļi pamatojas uz topoloģiskās telpas jēdzienu $T(X,Q)$, kur X ir elementu kopa, bet Q ir topoloģija, ko veido loku kopa [OSI 1969, GRU 1993, GRU 1997a, GRU 1999]. Katram sistēmas dekompozīcijas līmenim var izveidot struktūras modeļus. Struktūras modeļus iesaka konstruēt sistemātiski, lai nodrošinātu topoloģisko telpu nepārtrauktu atspoguļojumu. Modeļu skaitu un tipus nosaka sistēmas pētījuma mērķis un detalizācijas līmenis [GRU 1972, GRU 1993].

Struktūras modeļus vizualizē kā orientētus grafus, kur virsotnes atspoguļo objektus, funkcijas, uzvedības stāvokļus vai parametrus, bet saites starp virsotnēm attēlo plūsmas vai cēloņseku saistības. Katram no modeļiem ir definēta noteikta sintakse un semantika [GRU 1997a, ZEL 2011]. Tā kā struktūras modeļus atspoguļo grafa veidā, tad tiem var izveidot atbilstošas matricas [SIM 1962, GRU 1993] (piemēram, blakusvirsotņu). Tālāk ir detalizēti apskatīti SM izmantotie struktūras modeļi.

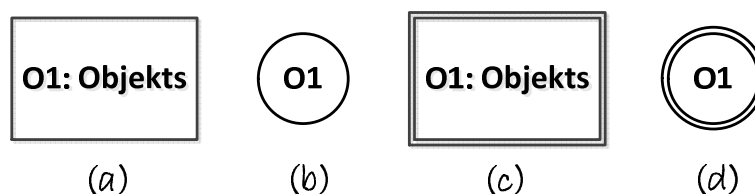
2.1.1. Morfoloģiskās struktūras modeļa apraksts

Struktūras modeļu izveide sākas ar morfoloģiskās struktūras modeļa (**MSM**) konstruēšanu, atspoguļojot iegūtās zināšanas par pētāmās sistēmas uzbūvi. MSM attēlo sistēmas iekšējo struktūru izvēlētajā dekompozīcijas līmenī, sistēmas daļas un saites starp tām, un strukturālās jeb cēloņseku attiecības. MSM atbalsta tā saucamo strukturālo spriešanu, kas balstās uz sistēmas daļu tiešo un netiešo saistību izpēti [GRU 1993, GRU 1997a, GRU 1999]. MSM ir definēts kā orientēts grafs $G(X,Q)$, kur katra virsotne $x \in X$ atbilst sistēmas daļai, reālam fiziskam elementam, ko apraksta, izmantojot objektus. Saistības starp objektiem atbilst plūsmām, ko modelī attēlo ar orientēta grafa lokiem Q . Struktūrmodelēšanā objektus apskata no statiskā viedokļa, jo sistēmas atspoguļojumu veido attiecībā uz noteiktu laika momentu [GRU 1993, GRU 1997a, GRU 1997b, GRU 1999].

Jēdziens *objekts* attiecas uz individuālu, identificējamu lietu, vienību vai realitāti, kas ir reāla vai abstrakta, ar labi definētu lomu/mērķi problēmsfērā. Objekts ir jebkas, kam ir stingri definēta robeža: materiāla un/vai redzama lieta; kaut kas, ko aptver intelektuāli; kaut kas, uz ko ir vērsta doma vai darbība; kaut kas, kas eksistē laikā un telpā [AMA 2004, YOU 2007, BOP 2008]. Objektam ir stāvoklis, tam piemīt labi definēta uzvedība un identitāte, un tas īsteno funkcijas [AMA 2004, YOU 2007, WEI 2009]. Objekta stāvoklis ietver visas objekta

īpašības (parasti statiskas) un šo īpašību vērtības (parasti dinamiskas), kas ir patiesas attiecībā uz objektu noteiktā laikā. Īpašība ir raksturīga vai atšķirīga pazīme, iezīme, kvalitāte vai īpatnība, kas raksturo objektu [YOU 2007, BOP 2008, OXF 2009]. Īpašības vērtība ir aprakstošs mainīgais, kas var norādīt uz citu objektu. Objekta identitāte ir īpašības, kas atšķir objektu no visiem citiem objektiem. Tās ir definējošās īpašības, kas nosaka objekta būtību. Pārējās īpašības, kurām nav svarīga loma objekta mērķu realizēšanā, sauc par pavadošām. Objekta uzvedība norāda kā objekts darbojas un reaģē objekta stāvokļa izmaiņu gadījumā un/vai ārējās vides ietekmē. Objekta uzvedība ir saskatāma darbībās, ko īsteno objekts attiecībā pret citu objektu [GRU 1997a, GRU 1997b, YOU 2007].

Struktūrmodelēšanā objektu apskata kā pamatvienību, ko pieņemts attēlot divos dažādos veidos (skatīt 2.1. att. (a) un (b) gadījumi) [GRU 1997a, ZEL 2008b]. MSM vizizplatītākais objekta atspoguļojums atbilst 2.1. att. (a) attēlā apskatītajam gadījumam [GRU 1997a, GRU 1997b], kurā norāda objekta nosaukumu un papildus iezīmi (identifikatoru), kas ļauj identificēt objektu sistēmā. Simbols *O* apzīmē objektu, skaitlis norāda tā kārtas numuru. Identifikators (piemēram, O1), ļauj attēlot modelī arī tādus sistēmas elementus, kuru nosaukumi ir vienādi. Modelī ir iespējams atspoguļot pēc struktūras un nosaukuma vienādus objektus, kas veic gan vienādas, gan arī dažādas darbības un īsteno atšķirīgas funkcijas. Katrs elements, kas ir būtisks sistēmas funkcionalitātes īstenošanā, sistēmas atspoguļojumā ir jāizdala atsevišķi. Piemēram, sistēmā var būt divi objekti ar nosaukumu *ventilators*, un šādā gadījumā tie tiks attēloti kā *O1: Ventilators* un *O2: Ventilators*. Otrs veids kā SM attēlo objektu ir grafa virsotne, kurā norāda tikai objekta identifikatoru (skatīt 2.1. att. (b) gadījumu).



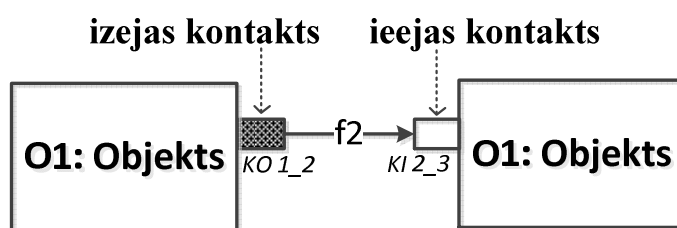
2.1. att. Objekta atspoguļojums

Lai SM varētu atspoguļot gan elementus, gan komponentes (definīcijas skatīt 1.1.1. nodaļā), darba autore SM ir ieviesusi papildus objekta apzīmējumus ar dubultlīniju (skatīt 2.1. att. (c) un (d) gadījumus). Līdz ar to ir mainīta objekta notācija modelī. Ja objektu attēlo ar vienu līniju (2.1. att. (a) un (b)), tas nozīmē, ka ir apskatīts sistēmas elements, bet ja ar dubultlīniju (2.1. att. (c) un (d)), tad sistēmas komponente.

Papildus struktūrmodelēšanā izdala divu veidu objektus: aktīvos un pasīvos. Par aktīviem objektiem uzskata sistēmas daļas, piemēram, tādas, kas veic aktīvas darbības, kustības, īsteno procesus vai reakcijas, kuras ietekmē citus objektus (piemēram, sūknis). Pasīvs objekts ir sistēmas daļa, kas īsteno noteiktu funkcionalitāti, bet tā veiktās darbības ir inertas, un tas aktīvi nepiedalās sistēmas funkcionalitātes realizēšanā (piemēram, pulksteņa korpuss, skrūve mehānismā) [OXF 2009]. Ir gadījumi, kad sistēma var funkcionēt un īstenot savus mērķus arī tad, ja pasīvos objektus neiekļauj sistēmā. Pasīvie objekti sistēmā ir saistošie elementi, kas ļauj izveidot komponentes. Atšķirības starp objektiem ir saskatāmas, veicot to funkciju, kā arī uzvedības un parametru izpēti. Sistēmā un tās atspoguļojumā visi objekti ir iekļauti mērķtiecīgi. Struktūras modelī attēlo gan aktīvos, gan pasīvos objektus, un tādējādi šajā darbā, ir mainīta MSM koncepcija, jo sākotnējā SM pieejā attēloja tikai aktīvos objektus.

Sistēmu apskata kā objektu un saišu kopu, un katrs objekts ir viens elements/komponente, kas nepieciešams, lai sistēma funkcionētu un spētu sasniegt savus mērķus. Ņemot vērā to, ka visas komponentes ir savstarpēji saistītas, un sistēmā nav izolētu daļu, veicot modeļa izveidi, nosaka attiecības starp sistēmas daļām. Lai norādītu kā objekti ir savstarpēji saistīti, MSM izmanto elementus: *kontakts* un *plūsma* [GRU 1997a, GRU 1997b].

Kontakts ir saskares punkts, kurā definē saistību nosacījumus (īpašības un to vērtības, kas nosaka plūsmas eksistenci) un saiti, kas pastāv starp apskatīto objektu un citu objektu sistēmā (vai to pašu objektu atgriezeniskās saites gadījumā). Kontakti atbilst objekta ieejām un izejām jeb ievadei un izvadei. Caur viena objekta izeju un cita objekta (vai tā paša objekta) ieeju pastāv saistība, kas ļauj realizēt kādu objekta darbību (aktivitāti, procesu). Procesa laikā izeju transformē ieejā, nodrošinot objekta funkcijas izpildi (skatīt 2.2. att.). Kontaktā definē, kā tiks īstenota konkrēta objekta funkcija vai novērtēts iegūtais rezultāts. Kontaktā kontrolē arī plūsmas stiprumu [ROS 1979, KUH 1974, GRU 1997b].



2.2. att. Izejas transformēšana ieejā

Ieejas kontaktā uztver informācijas/enerģijas/materiāla plūsmu (kustība), kas nāk no cita vai apskatītā objekta. Ieejā objekts iegūst resursus, kas nepieciešami tā darbību īstenošanai. Struktūrmodelēšanā izdala divu veidu ieejas: virknes ieejas, kas ir saistītas ar citu objektu

izejām; un atgriezeniskās saites ieejas. Atgriezeniskā saite kalpo kā kontroles mehānisms, ko sistēmā īsteno divos veidos:

- daļu no objekta izejām (informācija, materiāls vai enerģija) atgriež objekta ieejā, lai regulētu objekta izeju nākamajam darbības ciklam;
- objekts, lai varētu pats sevi vadīt, saņem informāciju no savas izejas, kas nodrošina atgriezeniskās saites procesa īstenošanu.

Izejas kontaktā plūsmu nodod no viena objekta uz citu vai uz to pašu objektu sistēmā.

Izdala trīs veidu izejas [GRU 1997a, GRU 1997b]:

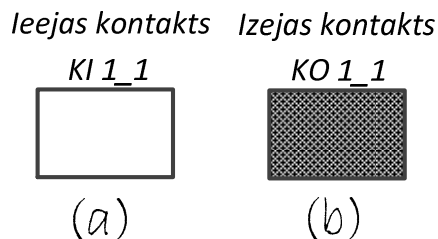
- izejas caur kurām objekts iedarbojas uz citiem objektiem jeb uz savu ārējo vidi;
- izejas, kuras ieiet tā paša objekta ieejā un kuras objekts patērē nākamā darbības ciklā (atgriezeniskās saites gadījums);
- izejas, kuras nepatērē apskatītais objekts un citi objekti sistēmā.

Kontakts struktūras modelī atspoguļo tad, kad objektu attēlo, izmantojot atspoguļojumu, kas atbilst 2.1. att. (a) un (c) apskatītajiem variantiem. Katram objektam struktūrmodelēšanā apraksta divu veidu kontaktus (skatīt 2.2. att.) [GRU 1997a, GRU 1997b, ZEL 2007]. Iepriekš SM kontakta identifikatoru attēloja norādot burtu K un kontakta kārtas numuru, ar kādu tas ir pievienots sistēmas atspoguļojumā. Darba autore ir mainījusi kontakta identifikatoru. Kontakta identifikatorā norādot informāciju par to, vai apskatītais kontakts ir ieejas vai izejas, ir iespējams kontaktus automātiski grupēt, pēc to tipa (ieejas, izejas). Grupēšanu lieto sistēmas analīzē. Savukārt norādot objekta kārtas numuru identifikatorā, ir iespējams veikt saistīto kontaktu analīzi, neizmantojot papildus objektu sarakstu. Kontaktu analīze ir nepieciešama, lai automatizētu sistēmas modeļu atspoguļošanu.

Kontaktus atspoguļo, lietojot jaunievistos identifikatorus:

- 1) Ieejas kontaktu (skatīt 2.3. att. (a)) attēlo kā neaizpildītu taisnstūri un apzīmē ar identifikatoru (piemēram, KI 1_1), kas sastāv no:
 - simbola „**K**”;
 - simbola „**I**”, kas norāda, ka apskatītais ir ieejas kontakts;
 - kārtas skaitļa, kas norāda objekta numuru, kuram pievienots kontakts;
 - simbola „_”, kas atdala objekta numuru un kontakta numuru;
 - skaitļa, kas norāda kontakta kārtas numuru, ar kādu tas ir pievienots attiecīgā objekta atspoguļojumam. Kontakta kārtas numuru pievieno automātiski, ņemot vērā objekta kontaktu skaitu un pievienošanas secību.

- 2) Izejas kontakts (skatīt 2.3. att. (b)). Kontakta atspoguļo kā aizpildītu taisnstūri. Izejas kontaktu arī apzīmē ar identifikatoru (piemēram, KO 1_1), ko veido tieši tāpat kā ieejas kontaktu. Vienīgā atšķirība ir apzīmējumā – aiz simbola K izejas kontakta gadījumā ir simbols O.



2.3. att. Kontakta atspoguļojums

Objektus, funkcijas un uzvedību, ko realizē sistēmā, saista attiecsmes. MSM saistību, ko atspoguļo starp kontaktiem, sauc par plūsmu. Plūsma ir funkcionēšanas pazīme. Objekts rada plūsmu un nodrošina tās īstenošanu. Plūsmas ir vienīgais saistību veids, ar kuru objekti var mijiedarboties. Plūsma ir jebkura darbība (aktivitāte, process), kas pārnes enerģiju, informāciju vai materiālu. Atspoguļojot objektus un starp tiem esošās plūsmas, attēlo sistēmas organizāciju izvēlētajā detalizācijas līmenī. Plūsmu eksistencei starp objektiem ir pastāvīgs raksturs, un tās nodrošina sistēmas funkciju izpildi. Kamēr plūsmas ir organizētas noteiktā struktūrā un šī struktūra nemainās, sistēma funkcionē normāli un nezaudē savu identitāti [KUH 1974, ROS 1979, OXF 2009, GRU 1997a, GRU 1997b].

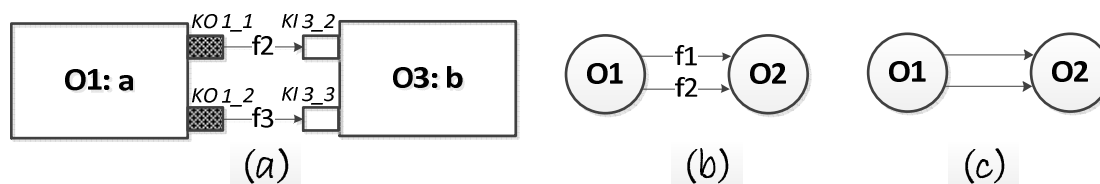
MSM plūsmu attēlo kā līniju ar bultu jeb orientētu loku (skatīt 2.4. att.) uz tā norādot plūsmas identifikatoru. Plūsmas identifikators sastāv no mazā burta f un skaitļa, ko piešķir secīgi, ņemot vērā to, kura plūsma pēc kārtas ir pievienota sistēmā.



2.4. att. Plūsmas atspoguļojums

Ņemot vērā to, ko plūsma pārnes, struktūrmodelēšanā izdala trīs dažādas plūsmas: enerģijas, materiālas un informācijas. Plūsmas veidu norāda, lietojot krāsas: enerģijai – sarkanu; materiālai – zaļu; informācijai – zilu [GRU 1997b].

Kontaktu un plūsmu identifikatoru norādīšana ļauj atspoguļot vairākas funkcijas, ko īsteno paralēli (skatīt 2.5. att. (a)). Ja morfoloģisko modeli attēlo kā grafu, neizmantojot kontaktus un plūsmu identifikatorus, tad atspoguļojums nav tik uzskatāms un izprotams (2.5. att. (c)), līdz ar to ir nepieciešams attēlot plūsmu identifikatorus (skatīt 2.5. att. (b)).



2.5. att. Kontakta un paralēlu plūsmu atspoguļojums

Ja objektā plūsma (-as) ieiet, bet no tā plūsmas neiziet, un dotais objekts ir iekļauts sistēmā, tad iespējams, ka par to nav pieejama visa nepieciešamā informācija vai arī tas ir gala produkts (piemēram, baterija). Gadījumā, kad no objekta plūsma (-as) iziet, bet tajā neienāk, normālas funkcionēšanas gadījumā objektu var uzskatīt par īslaicīgu enerģijas (informācijas/matērijas) avotu.

Veicot sistēmas struktūras analīzi, problēmsfēras eksperts pārzina sistēmas elementus, saistības starp tiem, plūsmu savietojamības iespējas un likumus, kas jāievēro, lai nodrošinātu sistēmai paredzētās funkcionalitātes izpildi. Plūsmu secība sistēmā atspoguļo objektu loģiskās mijiedarbību secības. Lai varētu attēlot organizētību un likumsakarības sistēmā, veidojot sistēmas atspoguļojumu, starp objektiem norāda plūsmas, bet starp plūsmām norāda loģiskos operatorus, kas ļauj attēlot plūsmu kombinācijas (ir vairākas ieejošās un/vai vairākas izejošās plūsmas) un spriest par sistēmas daļu saistībām un plūsmu eksistences nosacījumiem [GRU 1997b, GRU 1999, ZEL 2011].

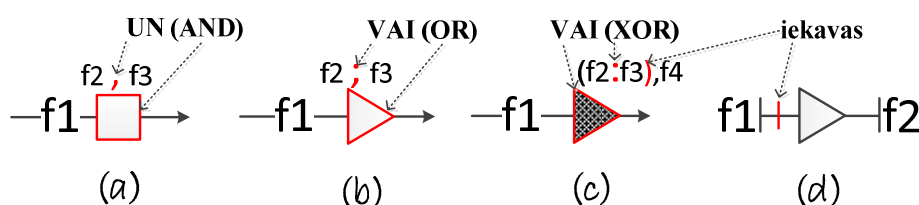
Struktūrmodelēšanas pieejā loģiskos operatorus izmantoja arī iepriekš FSM FT un notikumu kokos [GRU 1999]. Tomēr nebija izveidoti loģisko operatoru vizualizēti apzīmējumi un detalizēti skaidrojumi. Tā kā funkcionālās struktūras modeli iegūst no MSM, lai pievienotu loģiskos operatorus FSM, pēc MSM transformācijas FSM modelī, ekspertam ir jāiegulda papildus darbs. Struktūrmodelēšanas pieeju darba autore papildināja ar vizualizētiem loģiskiem operatoriem, pievienojot tos jau MSM notācijā [ZEL 2008b, ZEL 2011]:

- lai iegūtu zināšanas par objektu mijiedarbībām, plūsmu eksistenci un savietojamību, likumsakarībām un ierobežojumiem, kas pastāv sistēmā un nepieļautu kļūdas jau sākotnējā zināšanu apstrādē un sistēmas struktūras modeļa izveidē;
- lai izdalītu ceļus un ciklus sistēmas struktūrā. Operatoru pievienošana rada ierobežojumus attiecībā uz plūsmām un sistēmas daļām. Piemēram, ja objektam ir 2 ieejošās un 2 izejošās plūsmas, tad, pievienojot loģisko operatoru VAI un plūsmu

kombināciju, parāda, ka: (a) plūsmas neizpilda vienlaicīgi, (b) viena ieejas plūsma ietekmē vienu izejas plūsmu, un otra ieejas plūsma ietekmē otru izejas plūsmu;

- lai samazinātu eksperta darbu pie modeļu izveides. Loģiskos operatorus starp struktūras modeļiem pārnes, veicot transformācijas, līdz ar to ekspertam nav atkārtoti jādomā par loģisko operatoru parādīšanu, veidojot funkcionālās struktūras modeļus.

Loģiku struktūras modeļos attēlo, izmantojot loģiskos operatorus UN, VAI un izslēdzošo VAI (angļu val. – AND, OR, XOR), kā arī iekavas. Lai atspoguļotu loģisko operatoru UN, lieto kvadrātu un/vai komatu (skatīt 2.6. att. (a)). Operators UN starp plūsmām nozīmē to, ka visas ar šo loģisko operatoru saistītās plūsmas ir nepieciešamas objekta funkcionalitātes īstenošanai. Lai atspoguļotu VAI, lieto trīsstūri un/vai semikolu (skatīt 2.6. att. (b)). Izmantojot operatoru VAI norāda, ka ir nepieciešama kāda no saistītām plūsmām, bet ne obligāti visas. Saistītās plūsmas ir plūsmas, kuras vieno noteiktas funkcionalitātes nodrošināšana sistēmā. Tomēr (b) gadījumā apskatītais loģiskais operators VAI nav izslēdzošais VAI. Tas nozīmē, ka plūsmas var īstenot arī vienlaicīgi. Lai atspoguļotu izslēdzošo VAI, lieto kolu un/vai pildītu trīsstūri (skatīt 2.6. att. (c)). Izslēdzošā loģiskā operatora VAI gadījumā īsteno tikai vienu no saistītajām plūsmām. Iekavas izmanto, lai veidotu sarežģītākas izteiksmes, vai lai norādītu loģisko operatoru secību (skatīt 2.6. att. (d)). Ja loģiskos operatorus nenorāda (papildus loģisko operatoru lietošanas skaidrojumus skatīt pielikumā 2.1.), tad automātiski pieņem, ka starp objektā ieejošām kā arī izejošām plūsmām pastāv loģiskais operators UN [GRU 1999, ZEL 2008b].

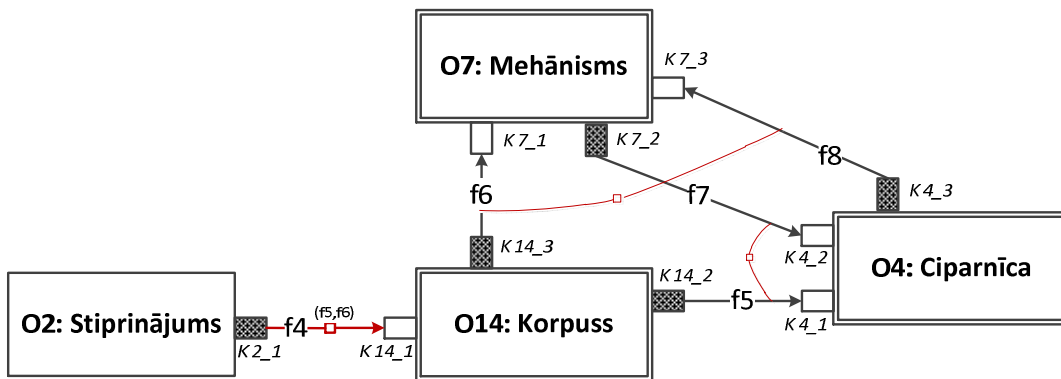


2.6. att. Loģiskā operatoru atspoguļojums struktūras modeļos

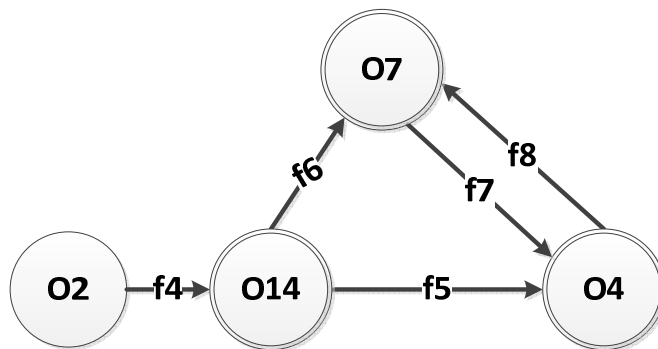
Morfoloģiskās struktūras modeļi loģiskos operatorus attēlo tikai uz ieejas plūsmām vai uz saitēm starp ieejas plūsmām. Ņemot vērā plūsmu kombinācijas, pastāv dažādi gadījumi plūsmu attēlošanai (skatīt 2.2.2. nodaļu).

Lai sniegtu priekšstatu par to kā MSM attēlo, izmantojot dažādas iespējamās pamatelementu notācijas, ir atspoguļota tehniska sistēma – mehāniskais pulkstenis (papildus apraksti un attēli ir pievienoti 2.2. pielikumā). Visbiežāk, veicot modeļu manuālu izveidi, MSM atspoguļo kā diagrammu, kas līdzīga blokshēmai. Šādā gadījumā pie objekta pievieno

ieejas un izejas kontaktus (skatīt 2.7. att.) vai uzvedības stāvokļus (detalizētāk aprakstīti 2.1.2. nodaļā un MSM piemērs sniegts 2.2. pielikumā). Vai arī MSM attēlo vienkāršoti, kā orientētu grafu (orgrafu) – norādot tikai objektus un plūsmas starp tiem (skatīt 2.8. att.) [GRU 1997a, GRU 1999]. To, kā tiks atspoguļots modelis, nosaka eksperts, ņemot vērā modelēšanas mērķim nepieciešamo informāciju [GRU 1997a, GRU 1997b, GRU 1999].



2.7. att. MSM atspoguļojums diagrammas veidā



2.8. att. MSM atspoguļojums grafa veidā

Kad MSM ir izveidots, to var detalizēt, komponentes sadalot elementos, transformēt citā modelī un izmantot spriešanai par sistēmu [GRU 1997a]. MSM nesatur informāciju par funkcionēšanas īpatnībām, līdz ar to modelim trūkst īpašības, kas ir nepieciešamas spriešanai par sistēmas funkcionalitāti un diagnostikai, kā arī tas ir nederīgs objektu darbības kļūdu analīzei [GRU 1972, GRU 1993, GRU 1997a, ZEL 2011]. Lai atbalstītu spriešanu par darbībām, kas norisinās sistēmā, pieejā ir ieviests funkcionālā modeļa jēdziens un izveidoti algoritmi, kas transformāciju rezultātā no MSM ļauj iegūt funkcionālās struktūras modeļus [GRU 1997a, GRU 1999]. Struktūras modeļu transformāciju apraksts, algoritmi un gadījumi ir doti 2.2.nodaļā.

2.1.2. Funkcionālās struktūras modeļu apraksts

Funkcionālās struktūras modelis apraksta sistēmas funkcionālās īpašības un to saistības organizētā veidā. Funkcionālā struktūra attiecas uz sistēmas daļas funkcijām (mērķiem), darbībām un uzvedību, tāpēc izšķir dažādus funkcionālās struktūras modeļus. Funkcionālās struktūras modeli funkciju telpā (FSM FT), līdzīgi kā MSM, attēlo kā orientētu grafu [GRU 1997a]. FSM FT virsotnes ir funkcijas un loki attēlo cēloņu–seku (jeb cēloņseku) saites (binārās attiecības) starp funkcijām [GRU 1997b, GRU 1999, ZEL 2008b].

Funkciju definē kā objekta eksistences mērķi, uzdevumu vai lomu, kas var norādīt uz saistību ar lietām un atkarību no lietu eksistences, vērtības vai jēgas, un ko ietekmē dažādi parametri [KRI 1986, OXF 2009]. Struktūrmodelēšanā funkcija atspoguļo mērķi jeb nolūku, kāpēc objekts ir ievietots sistēmā. Funkciju īsteno objekts, veicot darbību vai aktivitāti, kuru ierobežo noteikta uzvedība. Funkcijas realizācijas iespējas nosaka objekta stāvoklis (raksturo īpašības un uzvedība) noteiktā laika momentā. Funkcija īsteno laikā un tā ir viena no plūsmas īpašībām [GRU 1997a, GRU 1997b, GRU 1999]. Funkcionālās struktūras modelī funkciju telpā funkciju atspoguļo kā virsotni, lietojot apli un norādot tajā identifikatoru (skatīt 2.9. att.).



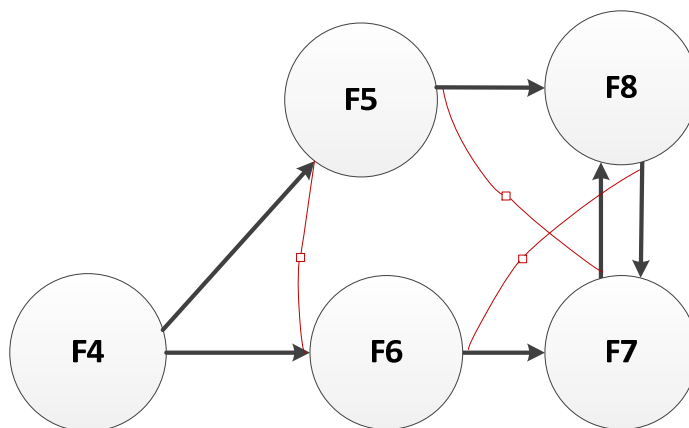
2.9. att. Funkcijas atspoguļojums

Funkcijas identifikators sastāv no lielā burtā F un skaitļa, kas norāda, kura pēc kārtas šī funkcija ir pievienota atspoguļojumam. Iepriekš struktūrmodelēšanā netika skaidri norādīts plūsmas un funkcijas apzīmējums un bieži vien abiem pamatelementiem identifikatorā izmantoja lielo burtu F [GRU 1999]. Funkcijas norāda visiem objektiem, kas iekļauti sistēmas struktūras atspoguļojumā. FSM starp funkcijām norāda cēloņu–seku saites, ko attēlo ar bultiņu (skatīt 2.10. att.). Starp cēloņseku saitēm norāda loģiskās saistības, lietojot vizualizētus loģisko operatoru apzīmējumus (skatīt 2.11. att.).



2.10. att. Cēloņseku saites atspoguļojums

FSM FT piemērs (2.11. att.) atbilst iepriekš apskatītajam morfoloģiskās struktūras modelim (skatīt 2.7. att.).



2.11. att. FSM FT atspoguļojums

FSM FT ļauj kvalitatīvi analizēt sistēmas funkcionēšanu un skaidrot, kā sistēma strādā normālos apstākļos, kad elementi pilda tiem paredzētās funkcijas un strādā bez kļūdām [GRU 1993, GRU 1997a, GRU 1997b, ZEL 2008b].

Funkcionālās struktūras modelis uzvedības telpā (FSM UT) jeb uzvedības stāvokļu modelis atspoguļo sistēmas daļu uzvedības stāvokļus un to saistības, kas norāda, kādā veidā sistēmai ir jādarbjas, lai funkcionētu atbilstoši tās mērķiem [GAR 2001, GRU 1997a]. Lai gan uzvedības stāvokļi tika apskatīti arī iepriekš struktūrmodelēšanā, atsevišķs uzvedības modelis netika izdalīts. Tā kā funkcionālās struktūras modelis uzvedības telpā (skatīt 2.13. att. (b) gadījumu) ir izomorfs FSM FT (skatīt 2.11. att.), tad uzvedības modelis tika apskatīts kā FSM FT atšķirīga interpretācija [GRU 1997a, GRU 1997b]. Pētījuma rezultātā tika konstatēts, ka abi modeļi atspoguļo funkcionālās iezīmes, tomēr modeļiem ir atšķirīga virsotņu semantika, un tāpēc tika izveidots FSM UT. Uzvedības modelis atspoguļo nevis mērķus, kas jāsasniedz, bet kādā veidā ir jāīsteno darbības, lai mērķus sasniegtu.

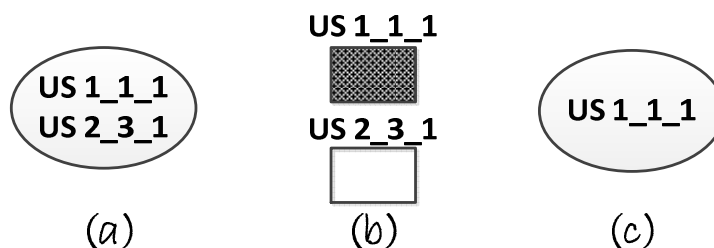
Uzvedības modeļa virsotnēm atbilst uzvedības stāvokļi. Līdzīgi kā FSM FT, starp virsotnēm attēlo cēloņseku saites, starp kurām norāda loģiskos operatorus. Uzvedību definē kā nosacījumus, kas raksturo objekta/sistēmas funkcionēšanu, un tā norāda veidu kā objekts darbojas. Uzvedība izpaužas objekta darbībās [OXF 2009]. Struktūrmodelēšanā uzvedība ir objekta ieejas un izejas iezīme, kas izskaidro, kādā veidā objekts nodrošina funkcijas izpildi un kā veic atbilstošo darbību. Tāpat kā funkciju, arī uzvedību atspoguļo saistībā ar noteiktu plūsmu. Katrai plūsmai norāda uzvedību, kas iekļauj vismaz divus uzvedības stāvokļus (ieejas, izejas), tomēr var eksistēt vairāk nekā divi uzvedības stāvokļi, kas izpaužas dažādos laika momentos. Tādēļ uzvedības atspoguļošanai ir vajadzīga uzvedības stāvokļu kopa. Tomēr noteiktā laika momentā var īstenot tikai vienu uzvedības stāvokli objekta izejā un vienu uzvedības stāvokli ieejā. Uzvedības stāvoklis ir plūsmas kvalitatīva iezīme, kas raksturo, kā

plūsma darbojas, kamēr uztur nepārtrauktu mijiedarbību no viena objekta uz citu. No struktūru modelēšanas viedokļa uzvedības stāvokļi norāda uz ierobežojumiem vai robežām, kuru ietvaros īsteno objektu funkcijas [GRU 1997a, GRU 1999, ZEL 2011].

Uzvedību atspoguļo izmantojot ovālu, kurā norāda divus uzvedības stāvokļus. Pirmais uzvedības stāvoklis atbilst plūsmas izejai, bet otrais – plūsmas ieejai (skatīt 2.12. att. (a) gadījumu). Uzvedībai nav identifikatora, un to apraksta, lietojot uzvedības stāvokļu apzīmējumus. Katrs uzvedības stāvoklis attiecas uz noteiktu ieeju/izeju un specificē to, kā plūsmu īsteno objekta ieejā vai izejā.

Uzvedības stāvokļus atspoguļo FSM UT, kā arī MSM, lai izprastu kāda uzvedība ir īstenota, objektam realizējot noteiktu plūsmu [GRU 1997a, GRU 1997b, GRU 1999]. MSM uzvedības stāvokļus attēlo līdzīgi kā kontaktus, norādot identifikatoru (skatīt 2.12. att. (b) gadījumu). FSM uzvedības stāvokli attēlo izmantojot ovālu, kurā norādīts identifikators (skatīt 2.12. att. (c) gadījumu). Struktūras modeļos lieto vienotu identifikatoru, kas sastāv no:

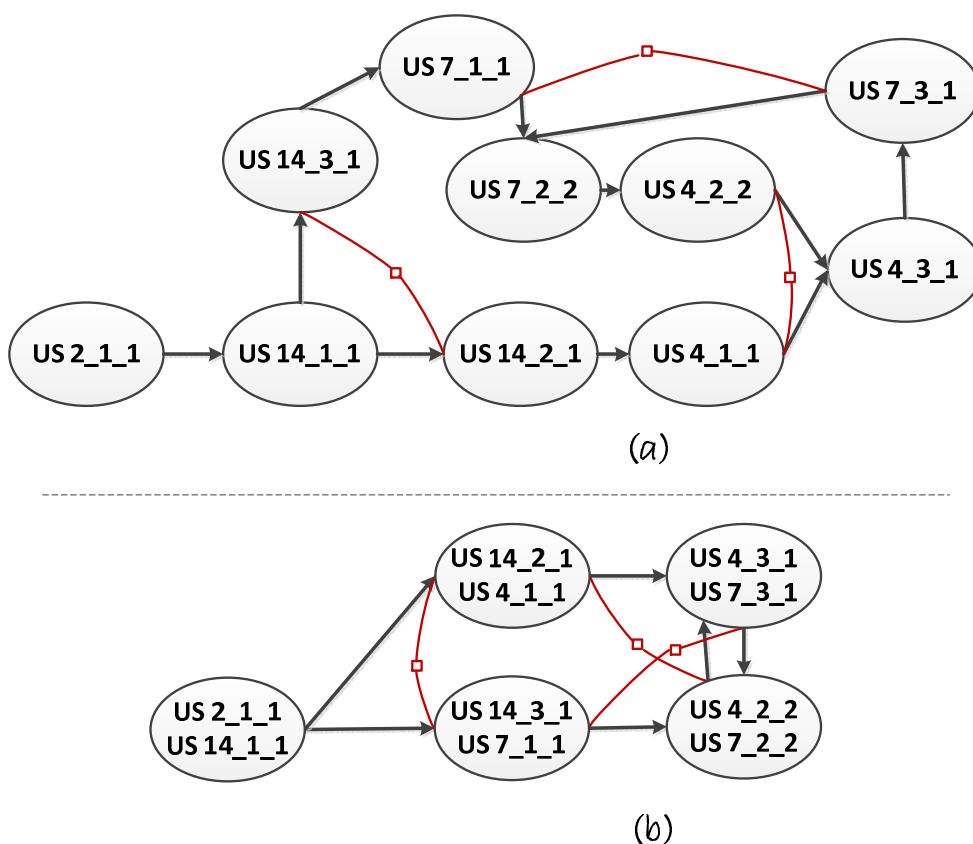
- lieliem burtiem „US”. Uzvedības stāvokļus MSM vizuāli atspoguļo līdzīgi kā kontaktus, un simbolus US norāda tāpēc, lai atšķirtu, vai attēlo uzvedības stāvokli vai kontaktu;
- kārtas skaitļa, kas norāda objekta numuru, kuram pievienots uzvedības stāvoklis;
- simbola „_”, kas atdala objekta numuru un kontakta numuru. Kontakta numuru uzvedības stāvoklim norāda, lai varētu apskatīt vairākus alternatīvus uzvedības stāvokļus;
- skaitļa, kas norāda, kurš kontakts pēc kārtas ir pievienots attiecīgā objekta atspoguļojumam.
- simbola „_”, kas atdala kontakta numuru un uzvedības stāvokļa numuru;
- skaitļa, kas norāda, kurš uzvedības stāvoklis pēc kārtas ir pievienots atspoguļojumam.



2.12. att. Uzvedības stāvokļu atspoguļojums

Cēlonseku saites starp virsotnēm attēlo tieši tāpat kā FSM FT (skatīt 2.10. att.), bet ņemot vērā virsotņu semantiku FSM UT var attēlot divējādi [GRU 1993, GRU 1997a, GRU 1997b]:

- kā secīgu uzvedības stāvokļu virkni, kas iegūta no MSM, kurā apskatīti uzvedības stāvokļi (skatīt 2.2. pielikumu), ieejas un izejas kontaktu vietā (skatīt 2.13. att. (a) gadījumu). Katrs uzvedības stāvoklis atbilst vienam kontaktam, starp kuriem pastāv cēlonseku saistības;
- kā divu secīgu uzvedības stāvokļu virkni (jeb uzvedību) vienā virsotnē (skatīt 2.13. att. (b) gadījumu).



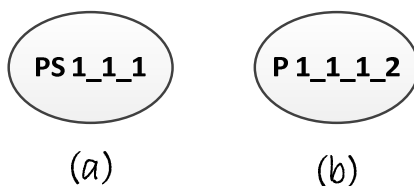
2.13. att. FSM UT atspoguļojums

Sistēmas daļām uzvedība mainās laikā, noteiktās robežās. Tas nozīmē, ka sistēmai ir vairāki stāvokļi, kas atbilst normālai funkcionēšanai. Uzvedības stāvokļi tāpat kā funkcijas ir kvalitatīvas iezīmes, kas ļauj spriest par sistēmas darbību, bet nav lietderīgas diagnostikai un sistēmas darbības un uzvedības prognozēšanai [GRU 1993, GRU 1997a, GRU 1997b]. Lai aprakstītu noteiktu uzvedību, uzvedības stāvokļos ir jālieto parametri vai mainīgie, kas raksturo objekta funkciju realizācijas efektivitāti. Tāpēc katru uzvedības stāvokli apraksta ar vienu parametru kopu.

Funkcionālās struktūras modelis parametru telpā (FSM PT) jeb parametru modelis ļauj atspoguļot sistēmas dinamiku, un īstenot diagnostisko spriešanu. FSM PT konstruē aizstājot uzvedības stāvokļus ar parametru kopām vai parametriem un izmantojot eksperta zināšanas par parametru savstarpējām saitēm parametru kopā [GRU 1993, GRU 1997a, ZEL 2011]. Parametru modelis atspoguļo to, kā tiek pārvaldīti sistēmas parametri [GAR 2001].

Parametrs ir kvantitatīvs rādītājs, novērtējams faktors, kuram ir piesaistīta noteikta vērtība vai vērtību apgabals konkrētā laikā periodā. Parametrā atspoguļo faktiskās jeb izmērāmās vērtības, kas atbilst reālai sistēmai pie noteiktiem funkcionēšanas apstākļiem. Parametri ir savstarpēji saistīti parametru kopās, un tie definē sistēmu vai sistēmas darbības un uzvedības nosacījumus. Ja objektam mainās mērķi, resursi, struktūra, tad var mainīties tā uzvedība un atbilstošie parametri. Sistēmas normālas funkcionēšanas gadījumā parametri, plūsmas sākumā un beigās, nemainās un nepārsniedz definētās robežas. Tas ir laikā, kad līdz ar plūsmu realizē noteiktu funkciju, nav pievienoti jauni vai mainīti esošie parametri. Plūsmu pārnesošā objekta (piemēram, asinsvads) ietekmē, ja tāds ir, var mainīties parametru vērtības, tomēr tās mainās noteiktajās robežās. Līdz ar parametru un/vai to vērtību izmaiņām mainās plūsmas un īstenotās funkcijas kvalitāte. Struktūrmodelēšanā izšķir divus sistēmas funkcionēšanas gadījumus: 1) sistēma funkcionē normāli; 2) sistēmā ir defekti. Ja parametra vērtība ir ārpus noteiktām robežām, tad, iespējams, ir radies defekts, kas izraisījis kļūdu sistēmā.

Parametrs struktūrmodelēšanā apraksta un ierobežo uzvedības stāvokli. Parametri ietekmē objekta darbības funkcijas īstenošanas laikā un raksturo sistēmas dinamiku. Struktūrmodelēšanā attiecībā uz parametru ir lietots arī jēdziens „mainīgais”, ko definē kā rādītāju, kas bieži mainās. Tomēr šis koncepts neatbilst struktūrmodelēšanas mērķim, jo atspoguļojot būtiski ir norādīt vērtības, kas raksturo sistēmas darbību un organizētību ilgtermiņā un ļauj spriest par sistēmu [GRU 1997a, OXF 2009, ZEL 2011]. FSM parametru kopu (skatīt 2.14. att. (a) gadījumu) un parametru attēlo izmantojot ovālu un identifikatoru (skatīt 2.14. att. (b)).



2.14. att. Parametru kopas un parametru atspoguļojums

Parametra un parametru kopas identifikatoru veido šādi:

- lielā burta „P”, bet parametru kopai ir divi burti „PS”;
- kārtas skaitļa, kas norāda objekta numuru, uz kuru attiecas apskatītais parametrs (parametru kopa);
- simbola „_”, kas atdala objekta numuru un kontakta numuru. Kontakta numuru uzvedības stāvoklim norāda, lai varētu atspoguļot vairākus alternatīvus uzvedības stāvokļus;
- skaitļa, kas norāda, kurš kontakts pēc kārtas ir pievienots attiecīgā objekta atspoguļojumam.
- simbola „_”, kas atdala kontakta numuru un uzvedības stāvokļa numuru, kas ir arī parametru kopas numurs, jo vienam uzvedības stāvoklim atbilst viena parametru kopa;
- simbola „_”, kas atdala parametru kopas numuru un parametra numuru. Parametru kopai šāda simbola, kā arī tālāk apskatītā skaitļa nav;
- skaitļa, kas norāda, kurš parametrs pēc kārtas ir pievienots atspoguļojumam.

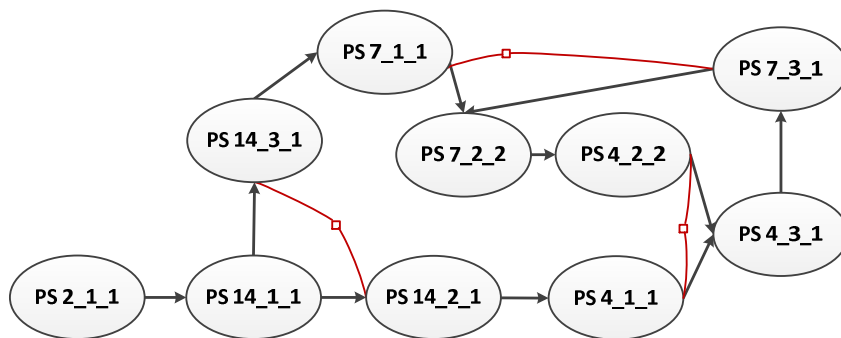
Ja kopā ir vairāki parametri, tad starp tiem nosaka saites un loģiskos operatorus. Daļu no loģiskiem operatoriem iegūst automātiski, veicot transformāciju no MSM uz FST PT, bet citus iegūst, veicot parametru kopas detalizāciju un izmantojot eksperta zināšanas par parametru saistībām (skatīt 2.16. att. (b)). Saites starp dažādu uzvedību stāvokļu parametriem sauc par ārējām, bet saites starp parametriem viena uzvedības stāvokļa robežās sauc par iekšējām saitēm. Saites starp virsotnēm attēlo tāpat kā FSM UT. Ja sistēmas funkcionēšanā ir kļūdas, tad veido FSM PT paplašināto modeli [GRU 1997a], kurā attēlo iespējamus defektus.

Defekta izpausmi var novērot parametru vērtībās, kā arī sistēmas uzvedībā, komponentēs un saistībās starp tām. Defekti rodas objektu neatbilstošas uzvedības, citu objektu vai spēku ilgstošas ietekmes rezultātā, vai funkcionalitātes nodrošināšanai nepiemērota objekta izvēles rezultātā. Defekts var būt cēlonis, kas rada struktūras un objekta uzvedības izmaiņas, un līdz ar to sistēmas funkcionēšanas kļūdas. Defektu var radīt gan sistēmas iekšējie, gan ārējās vides objekti, gan objektu nesavietojamība, gan jebkuras neparedzētas izmaiņas sistēmā. Plūsma pārnes defekta radīto ietekmi [GRU 1997a, OXF 2009, ZEL 2011]. Neatkarīgi no sistēmas dekompozīcijas līmeņa izvēles un atspoguļojumam izvēlētā struktūras modeļa, defektus attēlo, lietojot taisnstūri (skatīt 2.15. att.). Defektu apzīmē izmantojot identifikatoru, kas sastāv no lielā burta D, skaitļa, kas norāda objekta numuru, uz kuru attiecas apskatītais defekts un skaitļa, kas norāda, kurš defekts pēc kārtas ir pievienots atspoguļojumam.

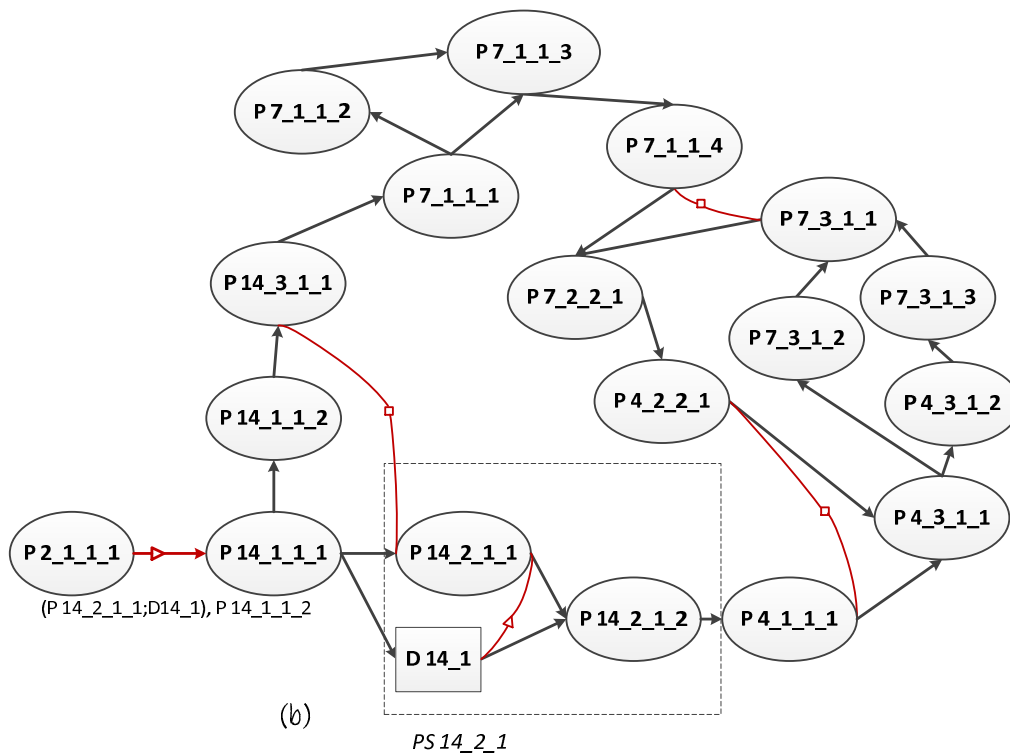
D 20_1

2.15. att. Defekta atspoguļojums

Modeļi, kurā atspoguļoti parametri, iegūst detalizējot parametru kopas (piemēram, skatīt 2.16. att. (b) gadījumu parametru kopu *PS 14_2_1* un tai atbilstošos parametrus). Parametru modelis (2.13. att. (a)) atbilst iepriekš apskatītās sistēmas funkcionālās struktūras modelim uzvedības telpā.



(a)



(b)

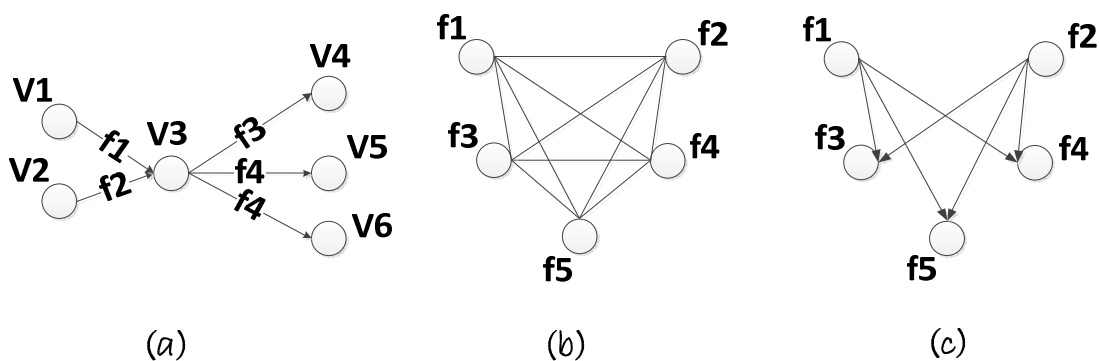
PS 14_2_1

2.16. att. FSM PT atspoguļojums

Atspoguļotās zināšanas par iespējamiem defektiem sistēmā, var lietot, lai diagnosticētu kļūdu cēloņus, ja sistēmas nespēj realizēt funkcijas un nedarbojas pareizi. Kļūdu ietekmes novērtējums balstās uz notikumu koku izveidi [GRU 1993, GRU 1997a, GRU 1997b].

2.2. Transformācijas starp struktūras modeļiem

Struktūrmodelēšanā iekļautās formālās metodes un algoritmi veido modeļu topoloģiju, nepārtrauktu un vienotu skatījumu uz sistēmu. Topoloģija nodrošina atbilstību starp dažādiem modeļiem un MSM granularitāti, kad turpina modeļu izveidi citos sistēmas detalizācijas līmeņos. Tā kā starp struktūras modeļiem pastāv līdzības, ir iespējams veikt modeļu transformācijas [GRU 1997a]. SM transformācija ir pāreja jeb zināšanu pārnese no vienas topoloģiskās telpas uz citu, vai no viena atspoguļošanas veida uz citu, ko realizē, izmantojot algoritmus. Modeļu transformācijas ir izveidotas, ņemot vērā formālu metodi no grafu teorijas [TUT 2001, MEN 2010], kas paredzēta neorientētiem grafiem. Sākotnējā grafa, ko sauc arī par virsotņu grafu (skatīt 2.17. att. (a) gadījumu), lokus pārveido par virsotnēm. Jauniegūtās virsotnes savstarpēji savieno, sekojot likumsakarībām, kas attēlotas virsotņu grafā, un izveido loku grafu (skatīt 2.17. att. (b) gadījumu).



2.17. att. Virsotņu grafs un loku grafi

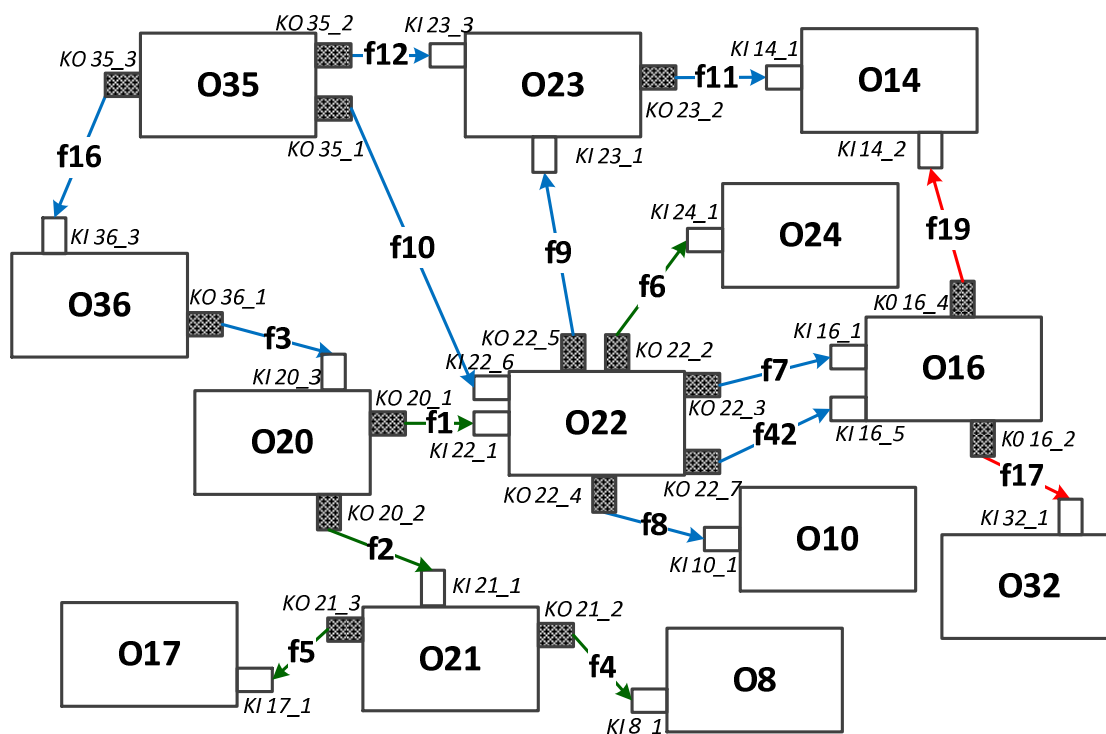
Struktūrmodelēšanā šī metode ir pielāgota orientētiem grafiem un transformācija atšķiras no tās, kas apskatīta grafu teorijā. SM, transformējot grafus, ņem vērā ierobežojumus: loku grafā starp virsotnēm pastāv saistība tikai tad, ja atbilstošiem lokiem virsotņu grafā ir viens virziens. Rezultātā atšķirības starp loku grafiem (skatīt 2.17. att. (b) un (c) gadījumu), kas iegūti, lietojot grafu teorijas un struktūrmodelēšanā aprakstītās metodes, ir ievērojamas. Lai gan virsotņu skaits un to identifikatori ir vienādi, ir atšķirīgs saišu skaits starp virsotnēm, kā arī SM gadījumā, iegūtie loki ir orientēti. Tādējādi, izmantojot SM aprakstītās

transformācijas metodes, izveidoto morfoloģiskās struktūras modeli, kas atbilst virsotņu grafam, var transformēt funkcionālās struktūras modeļos, kas atbilst orientētiem loku grafiem [GRU 2002, ZEL 2008b, ZEL 2011].

2.2.1. Transformācija no morfoloģiskās struktūras modeļa uz funkcionālās struktūras modeli funkciju telpā

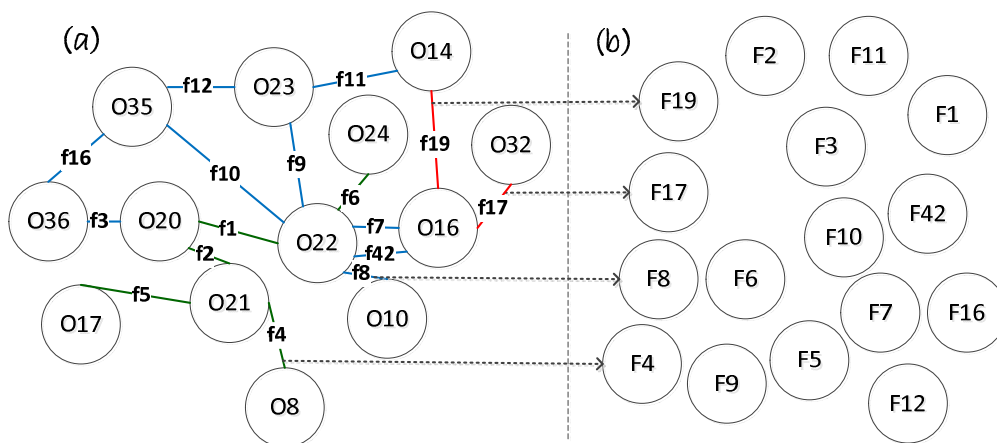
SM pieejā, lai iegūtu FSM topoloģiju funkciju telpā (FSM FT) no MSM, iepriekš lietoja transformācijas algoritmu, kas sastāv no 3 soļiem [GRU 1997a]:

- 1) MSM (skatīt 2.18. att.) apskata kā neorientētu grafu (skatīt 2.19. att. (a));



2.18. att. Transformācijā lietotais MSM piemērs

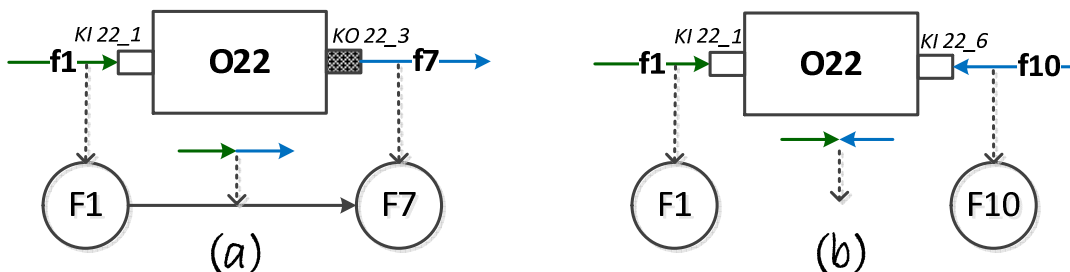
- 2) Iegūst loku grafa virsotnes (skatīt 2.19. att. (b)). Katrai plūsmai atbilst noteikta funkcija. Funkciju nosaukumiem, ko atspoguļo loku grafā, un plūsmu nosaukumiem MSM, ir atšķirīgi apzīmējumi (skatīt 2.1.2. nodaļu).



2.19. att. FSM FT virsotņu iegūšanas piemērs

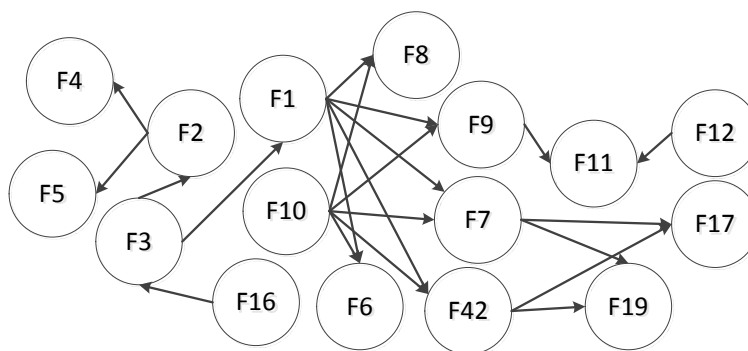
3) Savieno loku grafa virsotnes, ņemot vērā 2 likumus:

- a. ja vienas virsotnes diviem gadījuma (nejauši izvēlētiem) lokiem MSM orientētajā grafā ir viens virziens, tad loku grafā izveido orientētu loku no pirmās virsotnes uz otro (skatīt 2.20. att. (a));
- b. ja vienas virsotnes diviem gadījuma lokiem ir pretēji virzieni, tad atbilstošās virsotnes loku grafā nesavieno (skatīt 2.20. att. (b)).



2.20. att. Virsotņu savienošanas gadījumi

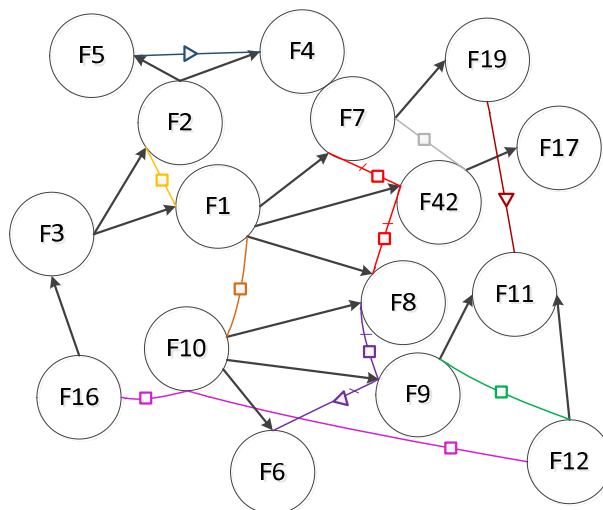
Transformācijas rezultātā iegūst FSM FT (skatīt 2.21. att.). Struktūras modeļu vizualizācijas piemēri dotās nodaļas ietvaros ir izveidoti programmas Microsoft Visio vidē.



2.21. att. Transformācijas rezultātā iegūtais FSM FT

- ii. ja uz ieejošā loka ir norādīts kādu izejošo loku tas ietekmē, tad loku grafā izveido orientētu loku no pirmās virsotnes uz otro tikai tad, ja uz apskatītā ieejošā loka ir norādīts apskatītais izejošais loks;
- iii. kad ir apskatīti vienas virsotnes visi ieejošie un izejošie loki, pārnes loģiskos operatorus, ņemot vērā iespējamās plūsmu kombināciju variantus (skatīt 2.2.2 nodaļu). Ja ir vairākas ieejas un izejas un nav norādīti loģiskie operatori, tad pieņem, ka starp ieejas plūsmām, kā arī starp izejas plūsmām pastāv loģiskie operatori UN.

Veicot soļus, kas aprakstīti jaunajā transformācijas algoritmā, iegūst FSM (skatīt 2.23. att.), kas atšķiras no struktūras modeļa, ko ieguva, veicot transformācijas algoritmu, kas SM pastāvēja iepriekš (skatīt 2.21. att.). Pirmkārt atšķiras saišu skaits, otrkārt 2.23. att. attēlā redzamajā modelī ir norādīti loģiskie operatori.



2.23. att. Jaunās transformācijas rezultātā iegūtais FSM FT

Loģisko operatoru izmantošana sākotnējā virsotņu grafā, veicot transformāciju, pareizas loku kopas izveidošanu FSM, nepieļaujot nesavienojamu saistību parādīšanos modelī. Ņemot vērā dažādos loģiskos operatorus un plūsmu kombināciju variantus, pastāv dažādi funkciju saistību varianti FSM, ko iegūst transformācijas rezultātā.

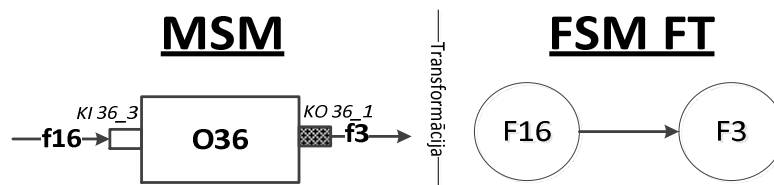
2.2.2. Plūsmu kombināciju varianti

Darba ietvaros ir aprakstīti 5 dažādi plūsmu kombināciju varianti MSM un struktūras modeļu transformācijas gadījumi, lietojot loģiskos operatorus [ZEL 2008b, ZEL 2011]:

- 1) viena ieejas plūsma un viena izejas plūsma;

- 2) viena ieejas plūsma un vairākas izejas plūsmas;
- 3) vairākas ieejas plūsmas un viena izejas plūsma;
- 4) vairākas ieejas plūsmas un vairākas izejas plūsmas;
- 5) viena vai vairākas ieejas plūsmas un neviena izejas plūsma vai neviena ieejas plūsma un viena vai vairākas izejas plūsmas.

Pirmais plūsmu kombināciju variants ir vienkāršs (skatīt 2.24. att.). Kaut gan uz ieejošās plūsmas var norādīt izejošās plūsmas nosaukumu, to parasti nedara, jo lieki apzīmējumi padara modeli grūti uztveramu.



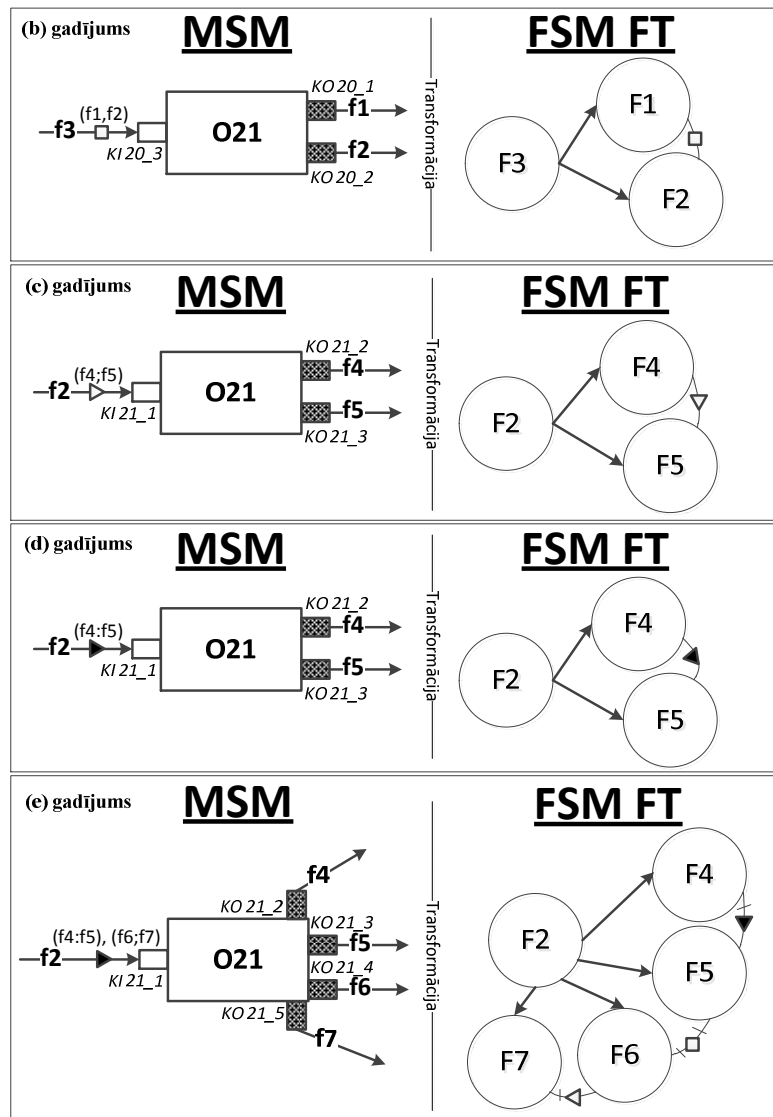
2.24. att. MSM un atbilstošais FSM FT (a) gadījums

Transformācijas rezultātā izveidotajam FSM visas starp virsotnēm esošās saistības apraksta, arī lietojot produkciju likumus. Veicot 2.21. att. atspoguļotā MSM fragmenta transformāciju, iegūst divas funkcijas starp kurām pastāv cēloņu–seku saistība, kurai atbilst likums: **IF F16 THEN F3**. Ja realizē funkciju F16, tad objekts O36 īsteno arī funkciju F3.

Otrajam plūsmu kombināciju variantam izdala 4 dažādus gadījumus:

1. Pirmajā gadījumā pastāv viena ieejas plūsma un vairākas izejas plūsmas, kas ir saistītas ar loģisko operatoru UN (skatīt 2.25. att. (b) gadījumu). Ja realizē funkciju F3, tad objekts vienlaicīgi īsteno funkciju F1 un F2. Atbilstošais likums: **IF F3 THEN F1 AND F2**.
2. Otrais gadījums no pirmā atšķiras ar lietoto loģisko operatoru. Viena ieejas plūsma ir saistīta ar divām izejas plūsmām, starp kurām ir loģiskais operators VAI (skatīt 2.25. att. (c) gadījumu). Ja realizē funkciju F2, tad objekts īsteno funkciju F4 vai F5 vai arī abas funkcijas vienlaicīgi, un tam atbilst likums: **IF F2 THEN F4 OR F5**.
3. Trešā gadījumā ir viena ieejas plūsma un vairākas izejas plūsmas, starp kurām izmanto loģisko operatoru, kuram atbilst izslēdzošais VAI (skatīt 2.25. att. (d) gadījumu) Apskatītam gadījumam atbilst likums **IF F2 THEN F4 XOR F5**. Ja realizē funkciju F2, tad objekts īsteno vai nu tikai funkciju F4 vai arī funkciju F5, bet nekādā gadījumā abas funkcijas vienlaicīgi.
4. Ceturtajā gadījumā ir viena ieejas plūsma un vairākas izejas plūsmas, kas saistītas ar dažādiem loģiskiem operatoriem (skatīt 2.25. att. (e) gadījumu). Ja realizē

funkciju F2, tad objekts īsteno vai nu funkciju F4 vai tikai funkciju F5, kā arī vienlaicīgi īsteno funkciju F6 vai/un F7. Atbilstošais likums ir šāds: **IF F2 THEN (F4 XOR F5) AND (F6 OR F7)**.

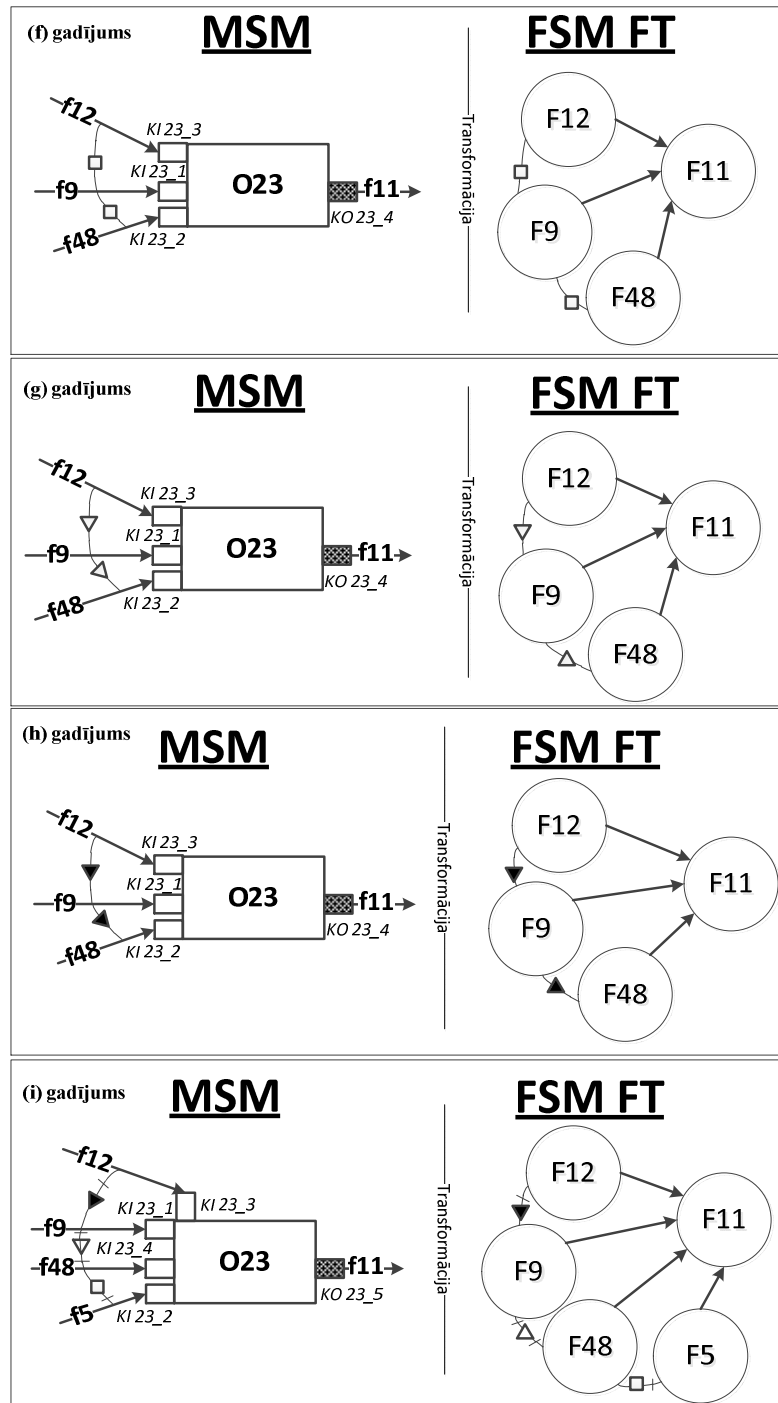


2.25. att. MSM un FSM FT (b), (c), (d), (e) gadījumi

Trešais plūsmu kombināciju variants ir sadalīts četros gadījumos:

- 1) vairākas ieejas plūsmas saistītas ar loģisko operatoru UN un viena izejas plūsma (skatīt 2.26. att. (f)). Gadījumam (f) atbilst likums: **IF F12 AND F9 AND F48 THEN F11**. Ja vienlaicīgi īsteno plūsmas F12, F9 un F48, tad objekts izpilda funkciju F11;
- 2) vairākas ieejas plūsmas saistītas ar loģisko operatoru OR un viena izejas plūsma (skatīt 2.26. att. (g)). Likums **IF F12 OR F9 OR F48 THEN F11** atbilst

gadījumam (g). Ja ir realizēta kāda vai vairākas no funkcijām objekta ieejā, F12, F9 vai F48, tad objekts īsteno funkciju F11;



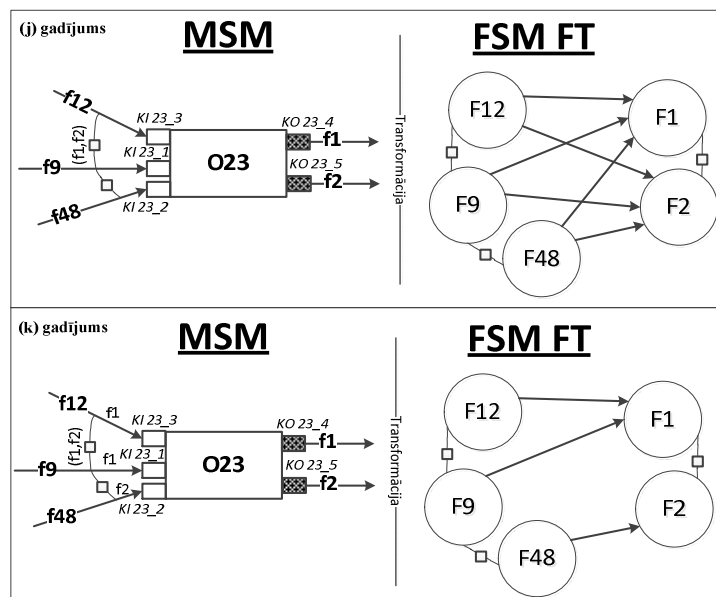
2.26. att. MSM un atbilstošie FSM FT (f), (g), (h), (i) gadījumi

- 3) vairākas ieejas plūsmas saistītas ar loģisko operatoru XOR un viena izejas plūsma (skatīt 2.26. att. (h)). Ja funkcijas F12, F9 vai F48 neīsteno vienlaicīgi un realizē tikai viena no minētajām funkcijām, tad objekts izpilda funkciju F11. Gadījumam (h) atbilstošais likums ir šāds: **IF F12 XOR F9 XOR F48 THEN F11;**

- 4) vairākas ieejas plūsmas saistītas ar dažādām loģisko operatoru kombinācijām un viena izejas plūsma (skatīt 2.26. att. (i)). Gadījumā (i) lieto dažādus loģiskos operatorus un gadījumam atbilst likums: **IF (F12 XOR F9) OR (F48 AND F5) THEN F11**. Ja realizē tikai vienu no funkcijām F12 vai F9 vai/un vienlaicīgi īsteno funkcijas F48 un F5, tad objekts izpilda funkciju F11.

Ceturtais plūsmu kombināciju variants ir vissarežģītākais no visiem variantiem, jo objektam ir vairākas ieejas un izejas plūsmas un tāpēc pastāv specifisks nosacījums: ja visas ieejas plūsmas vienlaicīgi neietekmē visas izejas plūsmas, tad papildus loģiskiem operatoriem uz ieejas plūsmām norāda arī konkrētas izejas plūsmas. Dotajam plūsmu kombināciju variantam var būt daudzi gadījumi.

Pirmajā gadījumā izmanto tikai loģisko operatoru UN starp ieejas un izejas plūsmām, bez papildus nosacījumiem, jo visas ieejas plūsmas vienlaicīgi ietekmē visas izejas plūsmas (skatīt 2.27. att. (j)). Gadījumam (j) atbilst likums: **IF F12 AND F9 AND F48 THEN F1 AND F2**. Ja ir vienlaicīgi realizētas funkcijas F12, F9 un F48, tad objekts vienlaicīgi īsteno funkcijas F1 un F2.

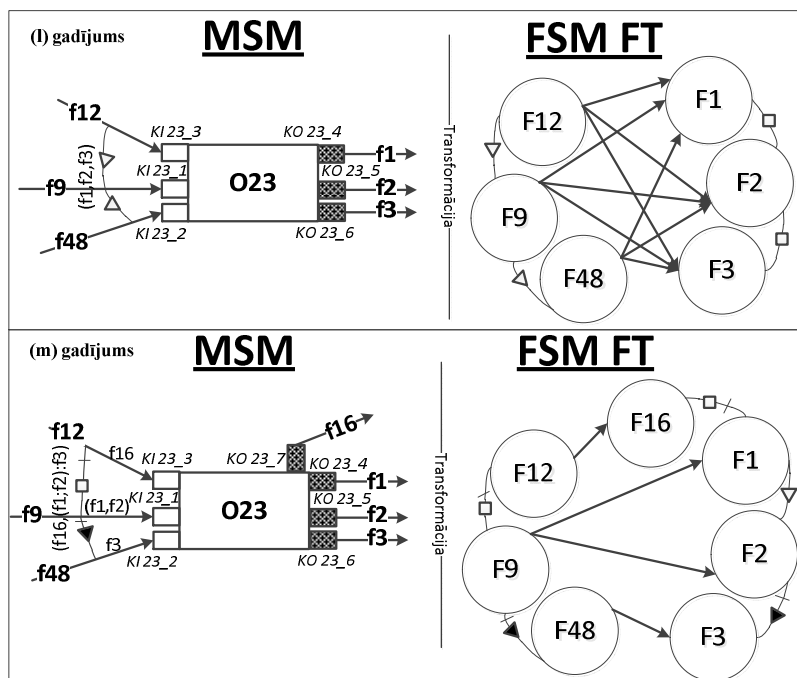


2.27. att. MSM un atbilstošais FSM FT (j) un (k) gadījumi

Gadījumā (k) izmanto tikai loģisko operatoru UN starp ieejas un izejas plūsmām, bet uz ieejas plūsmām norāda izejas plūsmas. Gadījumos (j) un (k) ir apskatīta līdzīga ieejas un izejas plūsmu kombinācija un lietoti vienādi loģiskie operatori. Tomēr transformācijas rezultātā katrā no minētajiem gadījumiem iegūst atšķirīgus FSM FT (skatīt 2.27. att.), ar atšķirīgu saišu skaitu starp virsotnēm. Gadījumam (k) atbilst likums: **IF F12 AND F9 THEN**

F1 AND IF F48 THEN F2. Ja ir īstenotas funkcijas F12 un F9, tad objekts realizē funkciju F1 un vienlaicīgi, ja ir īstenota funkcija F48, objekts realizē funkciju F2. Pēc līdzīgiem principiem, kā aprakstīts (j) un (k) gadījumos, veic arī transformācijas, kad loģiskā operatora UN vietā, starp visām ieejošām un izejošām plūsmām izmanto viena veida loģisko operatoru VAI un izslēdzošo VAI.

Gadījumā (l) ir apskatīts MSM fragments, kurā lieto viena veida loģisko operatoru ieejas plūsmām un cita veida loģisko operatoru attiecībā uz izejas plūsmām (skatīt 2.28. att.(j)). Apskatītam gadījumam atbilst likums: **IF F12 OR F9 OR F48 THEN F1 AND F2.** Ja ir īstenota kāda no funkcijām F12, F9 vai F48, vai tās vienlaicīgi, tad objekts vienlaicīgi īsteno funkcijas F1, F2 un F3.

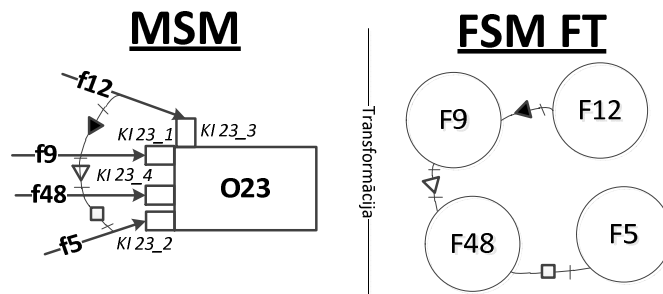


2.28. att. MSM un atbilstošais FSM FT (l) un (m) gadījumi

Pēdējā apskatītā gadījumā (m) ir izmantoti dažādi loģiskie operatori uz ieejas un izejas plūsmām, kā arī papildus nosacījumi (skatīt 2.28. att.(m)). Ja vienlaicīgi ir realizētas funkcijas F12 un F9, tad objekts vienlaicīgi īsteno funkcijas F16 un kādu no funkcijām F1 vai F2, vai abas kopā, vai arī, ja ir realizēta tikai funkcija F48, tad objekts īsteno funkciju F3. Gadījumam atbilst šāds likums: **IF F12 AND F9 THEN F16 AND (F1 OR F2) XOR IF F48 THEN F3**

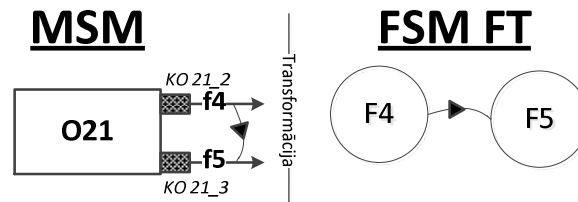
Piektajam plūsmu kombināciju variantam arī izšķir vairākus gadījumus, ievērojot loģisko operatoru lietojumu. Ja ir tikai viena ieejas vai tikai viena izejas plūsma, tad loģiskos operatorus nelieto. Toties, ja objektam ir vairākas ieejas vai izejas plūsmas, tad starp tām lieto vienādus vai dažādus loģiskos operatorus. Gadījumā (n) ir apskatīts piemērs, kad izmanto

dažādus loģiskos operatorus (skatīt 2.29. att.). Tā kā plūsmas neseko viena otrai, tad FSM FT, atbilstoši struktūras modeļu transformācijas likumiem, apskatītās funkcijas nesavieno ar cēloņu–seku saitēm. Iegūst likuma daļu, bet ne likumu: **(F12 XOR F9) OR (F48 AND F5)**. Tomēr tas norāda, ka sistēmā ir nepieciešams realizēt funkciju F12 vai F9 vai/un vienlaicīgi īstenojot funkcijas F48 un F5.



2.29. att. MSM un atbilstošais FSM FT (n) gadījums

Ja objektam ir tikai izejas plūsmas, tad MSM arī norāda loģiskos operatorus starp izejas plūsmām (skatīt 2.30. att.). Objekts īsteno funkciju F4 vai F5, bet ne abas vienlaicīgi. Gadījumam (o) atbilst likuma daļa: **F4 XOR F5**.



2.30. att. MSM un atbilstošais FSM FT (o) gadījums

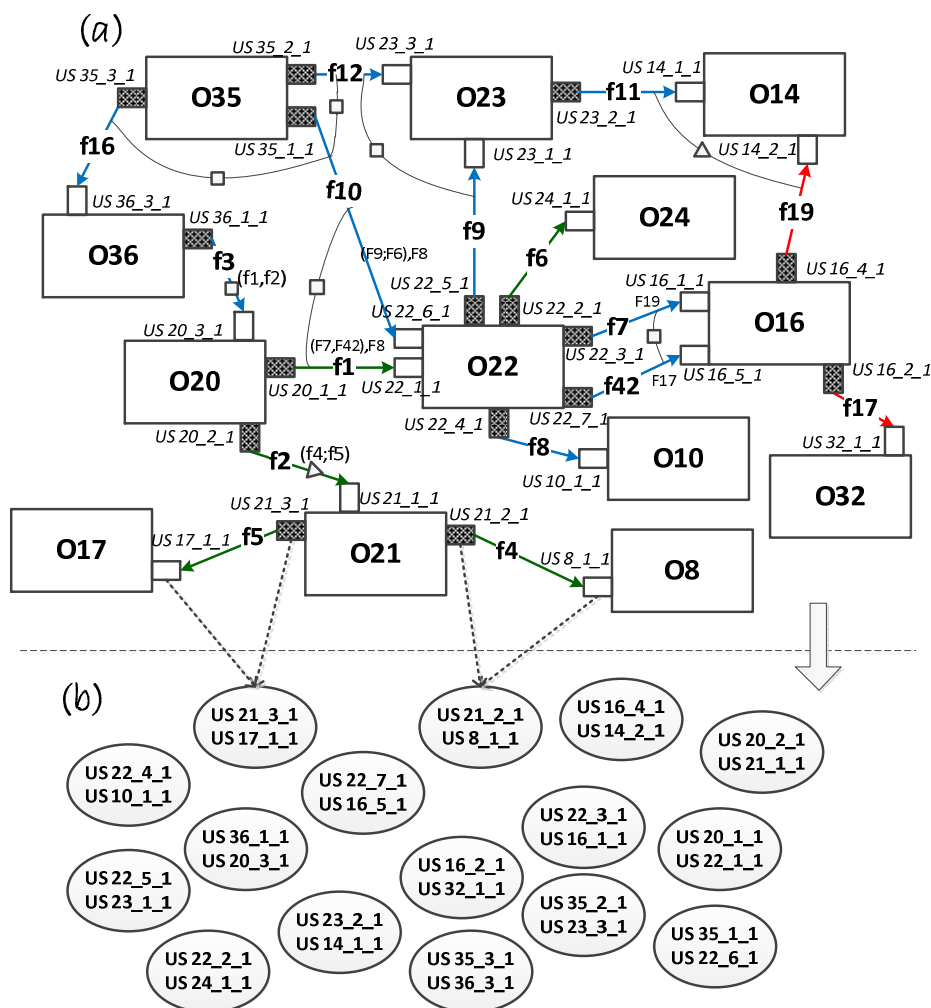
Loģiskos operatorus no MSM uz FSM UT un FSM PT pārnes līdzīgi kā norādīts apskatītajos gadījumos, bet pastāv atšķirības struktūras modeļu transformācijās, kas aprakstītas 2.2.3. nodaļā.

2.2.3. Transformācija no morfoloģiskās struktūras modeļa uz funkcionālās struktūras modeli uzvedības un parametru telpā

FSM UT topoloģiju, tāpat kā FSM FT, iegūst no morfoloģiskās struktūras modeļa, veicot transformāciju [GRU 1993, GRU 1997a]. Tomēr iepriekš SM nebija izveidots formāls algoritms transformācijai no MSM uz FSM UT, jo nebija atsevišķi definēts un aprakstīts

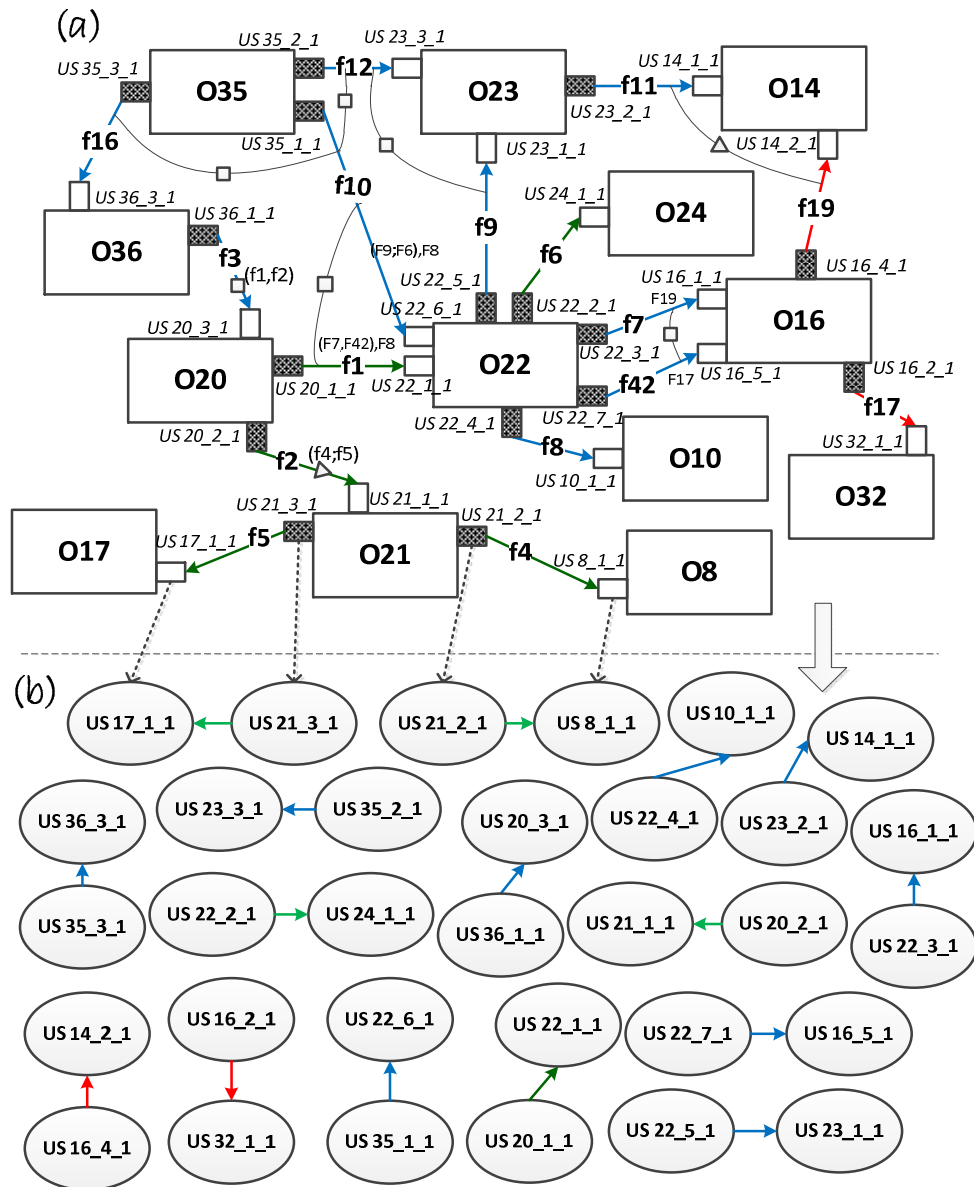
uzvedības modelis. Darba ietvaros ir izveidots transformācijas algoritms, kas sastāv no trīs soļiem:

- 1) Izvēlas FSM UT veidu: attēlot uzvedību vai uzvedības stāvokļus;
- 2) No MSM (skatīt 2.31. att. (a)) iegūst uzvedības stāvokļu pārus, kurus saista noteiktas plūsmas, un attēlo virsotnēs:
 - a. Ja izvēlēts attēlot FSM UT, norādot uzvedību, kas atbilst diviem uzvedības stāvokļiem, tad virsotnēs norāda divus atbilstošos uzvedības stāvokļus (skatīt 2.31. att. (b));



2.31. att. MSM uz FSM UT (uzvedība) virsotnes

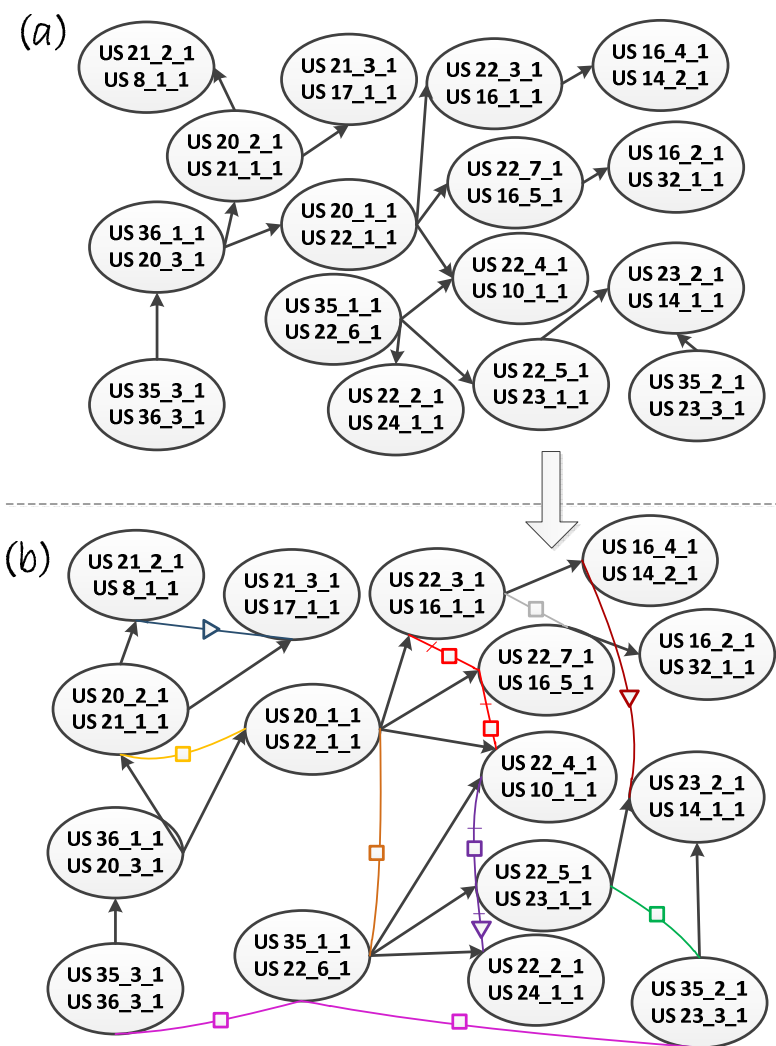
- b. Ja izvēlēts attēlot FSM UT, norādot uzvedības stāvokļus, tad virsotnēs norāda uzvedības stāvokļus, saglabājot saites starp uzvedības stāvokļiem, kas pastāv MSM (skatīt 2.32. att.(b));



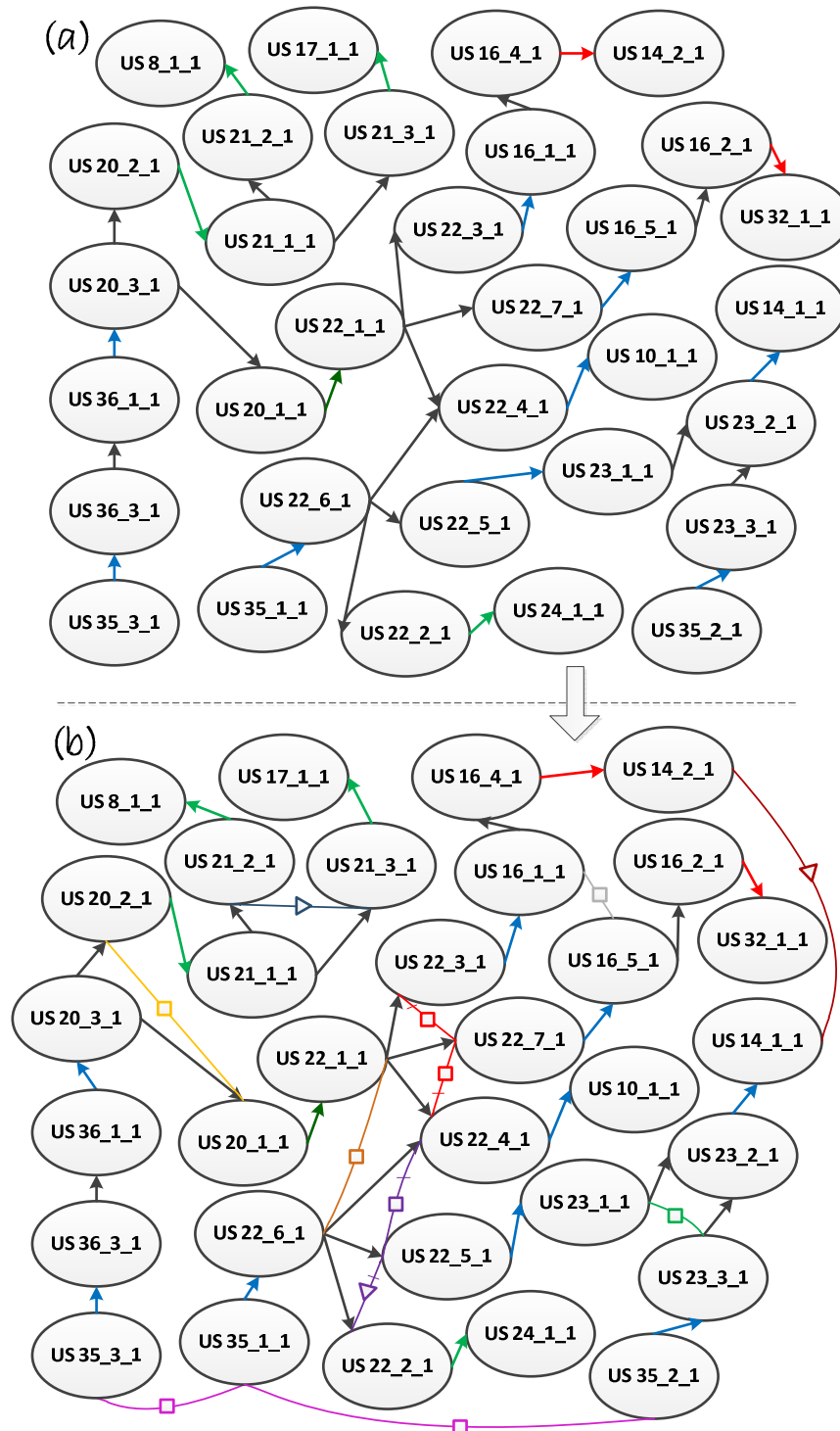
2.32. att. MSM uz FSM UT (uzvedības stāvokļi) virsotnes

- 3) Savieno grafa virsotnes (skatīt 2.33. att. un 2.34. att. (a) gadījumus), ņemot vērā 3 likumus (līdzīgs transformācijas algoritma no MSM uz FSM FT trešajam solim):
 - a. secīgi pārbauda visas virsotnes MSM, sākot no izvēlētās,
 - b. ja vienas virsotnes diviem gadījuma lokiem ir pretēji virzieni, tad atbilstošās virsotnes izveidotajā grafā nesavieno;
 - c. ja vienas virsotnes diviem gadījuma lokiem MSM orientētajā grafā ir viens virziens, tad pārbauda loģiskos operatorus un nosacījumus, kas atspoguļoti uz lokiem, kas ieiet apskatītajā virsotnē:

- i. ja loģiskie operatori nav norādīti un uz ieejošā loka nav norādīts, kādu izejošo loku tas ietekmē, izveidotajā grafā attēlo orientētu loku no pirmās virsotnes uz otro;
- ii. ja uz ieejošā loka ir norādīts, kādu izejošo loku tas ietekmē, tad izveidotajā grafā attēlo orientētu loku no pirmās virsotnes uz otro tikai tad, ja uz apskatītā ieejošā loka ir norādīts apskatītais izejošais loks;
- iii. kad ir apskatīti vienas virsotnes visi ieejošie un izejošie loki, pārnes loģiskos operatorus, ievērojot iespējamus plūsmu kombināciju variantus (2.33. att. un 2.34. att. (b) gadījumus)

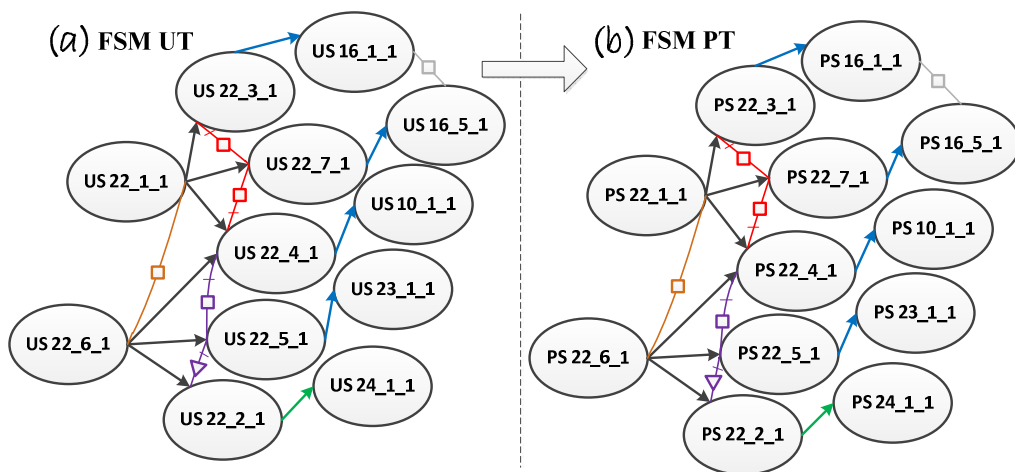


2.33. att. Transformācijas rezultātā iegūtais FSM UT (uzvedība)



2.34. att. Transformācijas rezultātā iegūtais FSM UT (uzvedības stāvokļi)

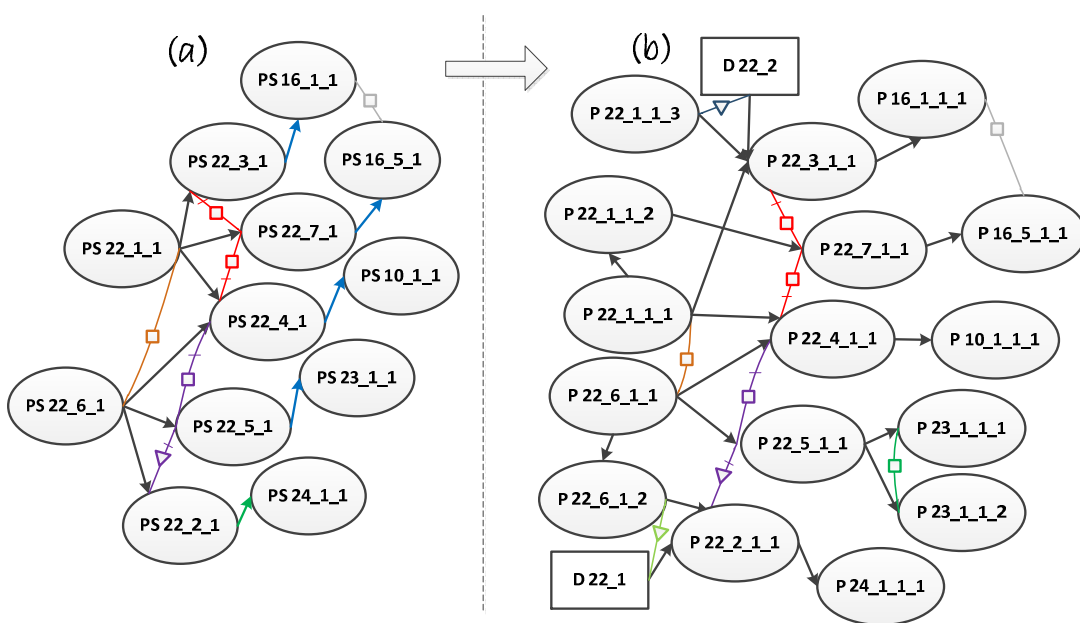
FSM PT iegūst no uzvedības modeļa, veicot transformāciju [ZEL 2011], un virsotnēs var būt attēlotas parametru kopas vai parametri. Katram uzvedības stāvoklim atbilst parametru kopa, un FSM PT iegūst, **aizstājot uzvedības stāvokļus ar parametru kopām**, līdz ar to modelis parametru telpā ir izomorfs FSM UT (skatīt 2.35. att.).



2.35. att. Transformācijas no FSM UT uz FSM PT piemērs

Ja katrai parametru kopai ir tikai viens parametrs un nav norādīti defekti, tad FSM PT, attēlojot parametrus, ir izomorfs FSM PT, kurā attēlo parametru kopas. Ja kādā parametru kopā ir vairāk nekā viens parametrs, tad ir nepieciešams iegūt zināšanas no eksperta par parametriem, to savstarpējām saistībām un iespējamajiem defektiem, ja tādi pastāv. Lai iegūtu FSM PT veic šādus soļus:

- 1) Apskata FSM PT, kurā norādītas parametru kopas (skatīt 2.36. att.);
- 2) Izmantojot eksperta zināšanas, detalizē katru virsotni un iegūst parametrus, kā arī defektus, kas ietilpst virsotnē attēlotajā parametru kopā.
- 3) Lietojot eksperta zināšanas pievieno loģiskos operatorus, kas iepriekš nav norādīti, bet ir būtiski sistēmas izpētē.



2.36. att. Transformācijas piemērs FSM PT

Transformācijas rezultātā iegūto funkcionālās struktūras modeli parametru telpā lieto, lai novērtētu elementu atteikumu un defektu izraisītās sekas un konstruētu notikumu koku, kas paplašina FSM PT izmantošanas iespējas [GRU 1993].

2. nodaļas kopsavilkums un secinājumi

Otrās nodaļā ir apskatīti struktūrmodelēšanas pieejā izmantotie struktūras modeļi, to pamatelementu sintakse un semantika, kā arī transformācijas starp modeļiem. Tas ir būtiski datorizētas sistēmas izstrādē, kurā ir implementēta struktūrmodelēšanas pieeja, jo nepieciešams izprast, kā attēlot no eksperta iegūtās zināšanas par izpētes sistēmu. Otrās nodaļas galvenie rezultāti ir šādi:

- papildināti esošie un izveidoti jauni struktūras modeļu pamatelementi (loģiskie operatori) un skaidrots to lietojums struktūrmodelēšanā;
- aprakstītas struktūras modeļu un to pamatelementu (objekta, kontakta, plūsmas, loģiskā operatora, funkcijas, uzvedības, uzvedības stāvokļa, parametru kopas, parametra un defekta) vizualizācijas iespējas un dots to skaidrojums;
- skaidrots jēdziens „loģiskais operators”, kas darba ietvaros ir ieviests struktūrmodelēšanā, jau sākot ar pirmo struktūras modeli MSM, lai nodrošinātu reālai sistēmai atbilstošu modeļu izveidi;
- skaidrots SM pieejā iepriekš lietotais struktūras modeļu transformācijas algoritms no morfoloģiskās struktūras modeļa uz funkcionālās struktūras modeli funkciju telpā, kā arī aprakstīti jaunizveidotie struktūras modeļu transformācijām atbilstošie algoritmi un doti transformāciju īstenošanas piemēri;
- aplūkoti plūsmu kombināciju varianti un loģisko operatoru izmantošanas iespējas struktūras modeļos.

Galvenie otrās nodaļas secinājumi ir šādi:

- loģiskos operatorus ir nepieciešams atspoguļot jau pirmajā sistēmas struktūras modelī, lai radītu izpratni par sistēmas uzbūvi, funkcionēšanu, kā arī samazinātu darbu, ko ekspertam jāiegulda, veidojot funkcionālās struktūras modeļus;
- sistēmā esošo ieejas un izejas *plūsmu kombināciju* un plūsmu loģisko secību (konkrēti kādas ieejas plūsmas ietekmē kādas izejas plūsmas) analīze ļāva iegūt iespējamās plūsmu saistību gadījumus;

- SM iepriekš aprakstītā struktūras modeļu transformācija nodrošina nepārtrauktu sistēmas atspoguļojumu, bet, lai iegūtu apskatītajai sistēmai atbilstošus FSM (nepieļaujot nesavienojamu saišu parādīšanās modeli), ir jāievēro papildus nosacījumi attiecībā uz loģisko operatoru lietojumu MSM (atbilstošs loģisko operatoru pieraksta veids un uz ieejas plūsmām norādītās izejas plūsmas);
- darba ietvaros izveidotais transformācijas algoritms no MSM uz FSM FT atšķiras no SM pieejā iepriekš izmantotā algoritma. Struktūras modeļiem, ko iegūst algoritmu īstenošanas rezultātā, ir atšķirīga sintakse, kā arī saišu skaits starp virsotnēm;
- struktūras modeļus veido manuāli, zīmējot tos vizuālās apstrādes rīkos, tomēr modeļu manuāla izveide ir darbietilpīgs process, tāpēc to nepieciešams automatizēt.

3. FREIMU KOPAS LIETOJUMS STRUKTŪRMODELĒŠANĀ UN SISTĒMAS I4S DARBĪBAS PRINCIPI

Iepriekš SM pieejā dziļas zināšanas par sistēmas morfoloģiju, funkcionalitāti un uzvedību normālas funkcionēšanas apstākļos, kā arī gadījumos, kad sistēma funkcionē kļūdaini, ieguva no problēmsfēras eksperta, izmantojot modeļus un struktūrmodelēšanas pieejas metodes [GRU 2002]. Pēc tam iegūtās zināšanas saglabāja freimu hierarhijā [GRU 1997a, GRU 1997b]. Tomēr struktūras modeļu manuāla izveide ir darbietilpīgs process [GRU 1993], kā arī, lai izveidotu intelektuālas sistēmas komponentes – zināšanu bāzi un izvedumu mašīnu, kas atbalsta sarežģītu sistēmu struktūrmodelēšanu, ir nepieciešama mērķim atbilstoša zināšanu atspoguļošanas shēma. Freimu hierarhija bija pārāk vienkārša (nebija izdalīti dažādi slotu veidi un ieviesti īpašību un alternatīvu freimi) un nebija piemērota, lai to lietotu intelektuālā sistēmā [ZEL 2007, ZEL 2010a]. Tāpēc promocijas darba ietvaros, implementējot SM pieeju intelektuālā sistēmā (turpmāk I4S), lai veiktu sarežģītu sistēmu struktūrmodelēšanu, tika mainīta SM koncepcija, zināšanu izgūšanas un atspoguļošanas pamatprincipi.

Darba ietvaros, lai noteiktu būtiskākās freimu un freimu sistēmu īpašības un to pielietojumu zināšanu atspoguļošanā, ir veikta 35 literatūras avotu analīze. Pamatojoties uz SM pieejā iepriekš integrēto freimu hierarhiju un citās pieejās lietoto freimos sakņotu zināšanu atspoguļošanas shēmu analīzi (skatīt 1. pielikumu), darba autore ir izstrādājusi freimu kopu [VAL 2005a, ZEL 2007, ZEL 2010a]. Jaunizveidoto zināšanu atspoguļošanas shēmu lieto intelektuālas sistēmas zināšanu bāzē, kā arī lietojumā, lai no eksperta iegūtu un vienuviet atspoguļotu zināšanas par izpētes objektu. Implementējot SM pieeju sistēmā I4S, darba autore ir izstrādājusi transformācijas algoritmus no freimu kopas uz struktūras modeļiem, lai īstenotu automatizētu modeļu izveidi, izvēlētajam izpētes sistēmas detalizācijas līmenim. Struktūras modeļu automatizēta izveide ļauj ietaupīt resursus, kas nepieciešami sistēmas atspoguļošanas laikā, kā arī nodrošina plūsmu un loģisko operatoru atbilstību dažādos modeļos [ZEL 2008b, ZEL 2011]. Šīs nodaļas mērķis ir apskatīt freima un freimu kopas lietojumu SM pieejā, skaidrot sistēmas I4S darbības principus un jaunizveidotās transformācijas no freimu kopas uz struktūras modeļiem.

3.1. Freimu kopas apraksts un lietojums

Freimu kopa ir izveidota struktūrmodeļēšanas mērķiem, lai izgūtu, atspoguļotu un saglabātu zināšanas par pētāmo sistēmu kā arī atsevišķiem objektiem [ZEL 2007, ZEL 2010a]. Freimu kopa ir specifiska zināšanu atspoguļošanas shēma, kura sastāv no saistītiem dažādas nozīmes freimiem, kur katram no tiem ir noteikts pielietojums [VAL 2005a, VAL 2005b, ZEL 2007, ZEL 2010a].

3.1.1. Freima jēdziens un tā struktūra

Freimus izmanto, lai atspoguļotu nelielas zināšanu vienības (piemēram, par realitāti, funkciju, uzvedību) vienuviet un pēc vienotas shēmas [MIN 1975a, KAR 1993, WHE 1993, BRO 1999, JAR 2006, BOP 2008]. Freims ir datu struktūra (skatīt 3.1. att.), kas paredzēta zināšanu atspoguļošanai par stereotipisku situāciju [MIN 1975a, MIN 1975b, CZO 1991, MAR 2006]. Freimi ir zināšanu atspoguļošanas formalisms, kas ļauj izgūt zināšanas strukturizētā veidā un organizēt tās hierarhijās [NEG 2004, MAR 2006]. Zināšanu atspoguļošanas veids freimos ir līdzīgs atspoguļošanai trīsvietīgā kortežā objekts–atribūts (īpašība)–vērtība, tomēr freims ir piemērots daudz sarežģītākai zināšanu atspoguļošanai [FAR 1997, LAS 1990, CZO 1991, BRO 1999, BOP 2008]. Freimā var attēlot dažāda veida informāciju: gan par freimā attēloto realitāti, gan ar to saistītos faktus, analogijas, darbības, mērķus, kā arī cēloņu–seku saites, realitātes izmaiņas laikā, un informāciju par to, kas var notikt pie aprakstītās realitātes izmaiņām. Freims nodrošina atsevišķu objektu struktūras, kā arī vairāku savstarpēji saistītu objektu atspoguļojumu [MIN 1975b, BRA 1984, FIK 1985, PAR 1989, CZO 1991, KAR 1993, KUI 1994, KUS 1997, GRU 1997a, SHA 2003, NEG 2004, BOP 2008, ZEL 2010a]. Freims, kā specifiska datu struktūra, ir izmantots dažādos pētījumos, bet katrā no tiem freima interpretācija un skaidrojums ir nedaudz atšķirīgs (skatīt 1. pielikumu). Freimi ir lietoti daudzu intelektuālu sistēmu izveidē [GRU 1999, YOU 2007]. Struktūrmodeļēšanā freimu izmanto kā zināšanu atspoguļošanas shēmu, kas ļauj organizēt kā arī atspoguļot un saglabāt zināšanu bāzē sistēmas struktūras modelēšanai nepieciešamās zināšanas [GRU 1997a, VAL 2005a, ZEL 2008a]. Tomēr SM freimi nav detalizēti aprakstīti un izskaidroti, tāpēc šajā apakšnodaļā ir skaidrota freimu struktūra, kā arī zināšanu atspoguļošanas iespējas freimos.

Aplūkojot freimu, parasti ņem vērā tā galvenās iezīmes. Tomēr izpratne, par to kā freimi izskatās vizuāli, ir dažāda. Freimus bieži vien attēlo kā tabulu vai ontoloģijām⁷ līdzīgu vizualizāciju. Tālāk apskatīto freima pieraksta veidu (skatīt 3.1. att.) izmanto struktūrmodelēšanā, un tas atbilst veidam, kas nepieciešams, lai īstenotu atsevišķu objektu un visas sistēmas atspoguļošanu, kā arī automatizētu sistēmu struktūras modeļu izveidi. Freims sastāv no nosaukuma, kas ir unikāls freimu sistēmā, un termināļiem jeb slotiem. Freimā var pastāvēt nosacījums, ka katrs freima slots un/vai vērtība ir unikāla aplūkotā freima ietvaros [CZO 1991]. Freima nosaukums apraksta objektu, ko tas atspoguļo. Savukārt freima slotus lieto, lai aprakstītu objekta īpašības [MIN 1975a, CZO 1991, KAR 1993].

Freimu sistēma sastāv no radniecīgu freimu kolekcijām/kopām, kas ir savstarpēji saistītas ar saitēm, tādējādi veidojot tīklu, kas vienkāršā un strukturālā veidā atspoguļo izvēlēto pētījuma objektu no dažādām perspektīvām [MIN 1975b]. Šādu zināšanu atspoguļošanas veidu lieto, lai konstruētu sistēmas modeļi vai vairākus savstarpēji saistītus sistēmu modeļus [MIN 1975b, KAR 1993], kas ir būtiski struktūrmodelēšanas pieejā.

Freims

Nosaukums:	<i>O1: Mehānisms</i>
Priekštecis	<i>Nezināms</i>
Sloti	
Nosaukums	Vērtība
<i>Krāsa</i>	<i>Nezināma</i>
<i>Svars</i>	<i>25 grami</i>
<i>Materiāls</i>	<i>Metāls</i>

3.1. att. Zināšanu atspoguļošanas shēma freims

Ņemot vērā slotu izmantošanas mērķus, dažādos avotos tos sauc par īpašībām (no angļu val. – properties, attributes) un/vai iezīmēm (angļu val. – characteristics) [MIN 1975a, MIN 1975b, KAR 1993]. Bieži slotus pārdēvē par īpašībām, ja freimos apskata mantošanu. Pastāv arī gadījumi, kad slotus lieto, lai attēlotu strukturālu informāciju, piemēram objektu daļas [KAR 1993, SHA 2003, FIK 1985]. Struktūrmodelēšanā slotus sauc arī par īpašībām [GRU 1997a, ZEL 2010a]. Katrs slots sastāv no slotā nosaukuma un vērtības [MIN 1975b] vai

⁷ Ontoloģija ir skaidri definēta, koplietojama konceptualizācijas specifika. Ontoloģija definē jēdzienus noteiktā izpētes jomā, saites starp tiem un likumus, kas ļauj kombinēt jēdzienus [LIE 1997, MAR 2006, BOP 2008].

vērtību saraksta [ROB 1977, GRE 1980, ROB 1994, GRE 1996, BRA 2000, SHA 2003], vai fasetes. Fasete vai fasešu saraksts principā aizvieto slotu vērtību – tās sauc arī par slotu īpašībām [FIK 1985, CZO 1991, KUS 1997, COR 2003, NEG 2004, MAR 2006]. Parasti viena no vērtībām sarakstā ir noklusētā un pārējās ir alternatīvas vērtības vai arī vērtības, kurām ir mazāka nozīme attiecībā uz atspoguļojuma mērķi (skatīt 3.2. att.). Slotu vērtību sarakstu attēlo gan kā vērtību virkni (skatīt 3.2. att. (a) gadījumu), gan kā izvēlni (skatīt 3.2. att. (b) gadījumu), un pirmā vērtība sarakstā ir noklusētā. Slotu vērtības virknē var būt saistītas ar loģiskiem operatoriem [CHA 1978, STE 1978, FIK 1985, CZO 1991, COR 2003, MER 2003, SPE 2004]. Fasetēs bieži vien ir atspoguļota procedurāla informācija – demoni (skaidrojums tālāk 89.lpp.), likumi, nosacījumi, metodes un ierobežojumi [ROB 1977, FIK 1985, CZO 1991, GRE 1996, KUS 1997, COR 2003, NEG 2004], un tādēļ fasetes reizēm kļūdaini uzskata par procedūrām.

Sloti	
Nosaukums	Vērtība
<i>Krāsa</i>	zaļa; sarkana; balta

(a)

Sloti	
Nosaukums	Vērtība
<i>Krāsa</i>	zaļa
	sarkana
	balta

(b)

3.2. att. Slotu vērtību saraksts

Freima un slotu nosaukumā un/vai vērtībā var būt arī vairāk kā viens vārds, kas apraksta atspoguļoto realitāti. Slotu vērtība var būt: cits freims, aprakstošs mainīgais vai procedūra. Veicot zināšanu atspoguļošanu esošās, uz freimiem balstītās pieejās izplatītākais risinājums ir izmantot citu freimu kā slotu vērtību [GRU 1997a, FIK 1985, KAR 1993, MIN 1975a, ROB 1977, STE 1978, VAL 2005c, WHE 1993, GRE 1980, MAR 2006, ADE 1988, SPE 2004, KUS 1997, STE 1979]. Šādā gadījumā, piemēram, ja slotu nosaukums būs *auglis*, tad slotu vērtība būs *citrons*, pie nosacījuma, ka apskatītā freima nosaukums ir *citronkoks*.

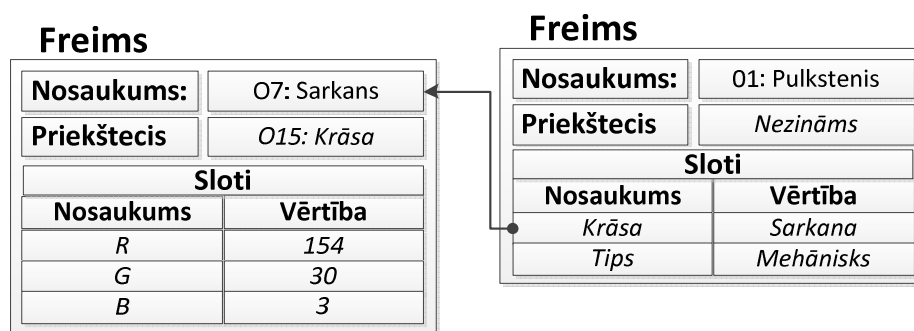
Arī aprakstošos mainīgos lieto kā slotu vērtības [GRU 1997a, FIK 1985, KAR 1993, MIN 1975a, ROB 1977, VAL 2005c, WHE 1993, SHA 2003]. Piemēram, slotu vērtība var būt *sarkans*, ja slotu nosaukums ir *krāsa*; skaitlis 4, ja slotu nosaukums ir *riteņu skaits*. Vienas un tās pašas vērtības var būt dažādiem slotiem pie dažādiem slotu nosaukumiem.

Trešais gadījums ir procedūru vai likumu (Ja ...Tad formātā) izmantošana slotos [FIK 1985, ROB 1977, KAR 1993, KUI 1994, BRA 2000, MER 2003, COO 2001, LEE 1999, NEG 2004]. Dažkārt procedūras, ko attēlo slotos vai pievieno slotiem, sauc par demoniem vai „kalpotājiem” (no angļu val. – demon vai daemon, servant) vai nosacījumiem (angļu val.

conditions). Démoni, saukti arī par „ja pievienots” procedūrām (angļu val. if added), ir procedūras, ko automātiski īsteno, kad mainās slotu vērtība. „Kalpotāji” jeb „ja nepieciešams” (angļu val. if needed) procedūras īsteno, kad ir nepieciešama vai ir pieprasīta slotu vērtība [ROB 1994, MAR 2006]. Var pievienot arī likumu daļas, ko apskata kā vairākus atsevišķus slotus vai pat atsevišķus freimus [MAR 2006, ADE 1988, LEE 1999]. Gadījumā, kad priekšnosacījums jeb likuma sākuma daļa (Ja...) ir slotu nosaukums, tad darbība jeb likuma beigu daļa (Tad...) ir slotu vērtība. Piemēram, slotu nosaukums – [Ja enerģijas avots = nav], bet slotu vērtība – [Tad pulksteņa stāvoklis = nefunkcionē]. Papildus trīs minētajiem gadījumiem, slotu vērtība var būt arī *nezināma* [MIN 1975a, SPE 2004], un šāds risinājums ļauj atspoguļot zināšanas nepilnīgas informācijas apstākļos [GRU 1997a].

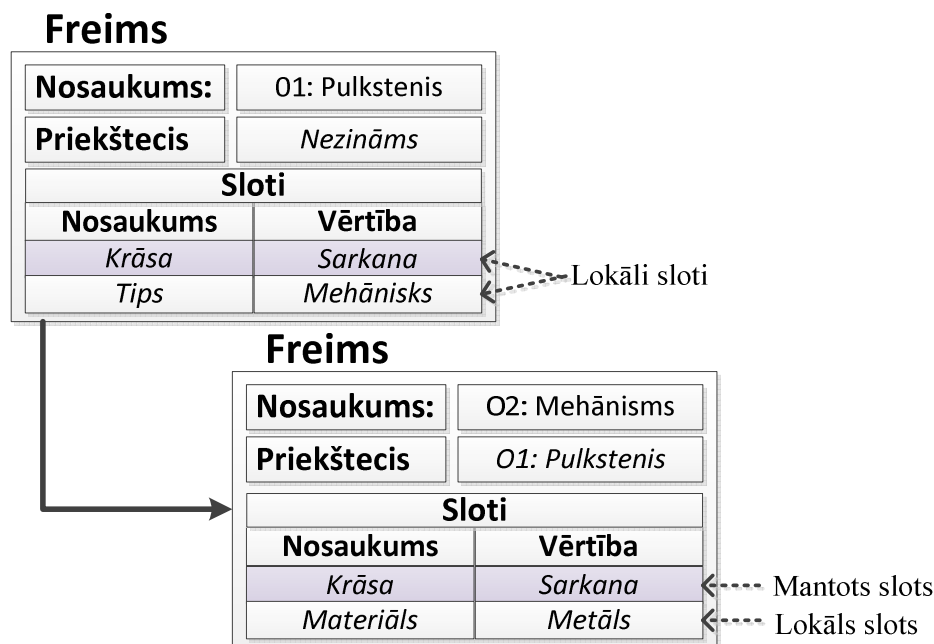
Atbilstoši slotu mērķim vai lomai freimā un tajā atspoguļotajai informācijai, slotus var sadalīt dažādos tipos [FIK 1985, MIN 1975a, GRU 1997a, KAR 1993, WHE 1993, COO 2001, MAR 2006, LEE 1999, MOR 2009], ko, savukārt, var apvienot kopās. Freimos sākotnās atspoguļošanas pieejās norāda uz slotu iespējamām atšķirībām, dažādām lomām, ierobežojumiem, bet uzskatāmi neizdala slotu tipus [CZO 1991, GRE 1980, KUS 1997, OKA 2007]. Struktūrmodelēšanā slotus pēc to lomas freimā sadala dažādos tipos, piemēram, īpašības, kontakti, utt. [ZEL 2007, ZEL 2010a].

Slotus var būt norādes (skatīt 3.3. att.), kas sasaista tajos attēloto informāciju ar citiem freimiem [MIN 1975b, CHA 1978, BRA 1984]. Norāde var būt piesaistīta gan slotu nosaukumam, gan slotu vērtībai, gan slotam kopumā, un tā atspoguļo saiti uz citu freimu, kura nosaukums ir vienāds ar slotu nosaukumu (vai vērtību), tādējādi piesaistot informāciju, kas ļauj skaidrot slotu vai tā vērtību. Norāde var būt saistīta ne tikai ar kontekstuālu informāciju, bet arī ar ierobežojumiem un likumiem attiecībā uz slotu vai apskatīto freimu. Šādas norādes kopā ar procedūrām un likumiem ļauj nodrošināt dinamisku freimu sistēmā. Norādes tāpat kā citu ar freimu saistīto informāciju atspoguļo apskatītā freima slotus vai pat atsevišķos freimos [MIN 1975a, KAR 1993, KAR 1995, GRU 1999, MAR 2006, ZEL 2010a].



3.3. att. Ar norādi saistīti freimi

Dažkārt freimos iekļauj netikai freima nosaukumu, bet arī virsklases (priekštecis) vai apakšklases (pēctecis) freima nosaukumu (skatīt 3.3. att. iezīmi „priekštecis”). Dažādi freimi sistēmā var koplietot vienus un tos pašus slotus. Iezīmes iekļaušana un koplietošanas iespējas ļauj organizēt freimus taksonomijās jeb mantošanas hierarhijās [ROB 1977, BRA 1984, WHE 1993, GRE 1980, SPE 2004, GAN 1993, KAR 1993, GRU 1997a], kurās katrs freims ir saistīts ar vienu (dažās sistēmās ar vairāk kā vienu) priekšteci.



3.4. att. Mantoti un lokāli sloti

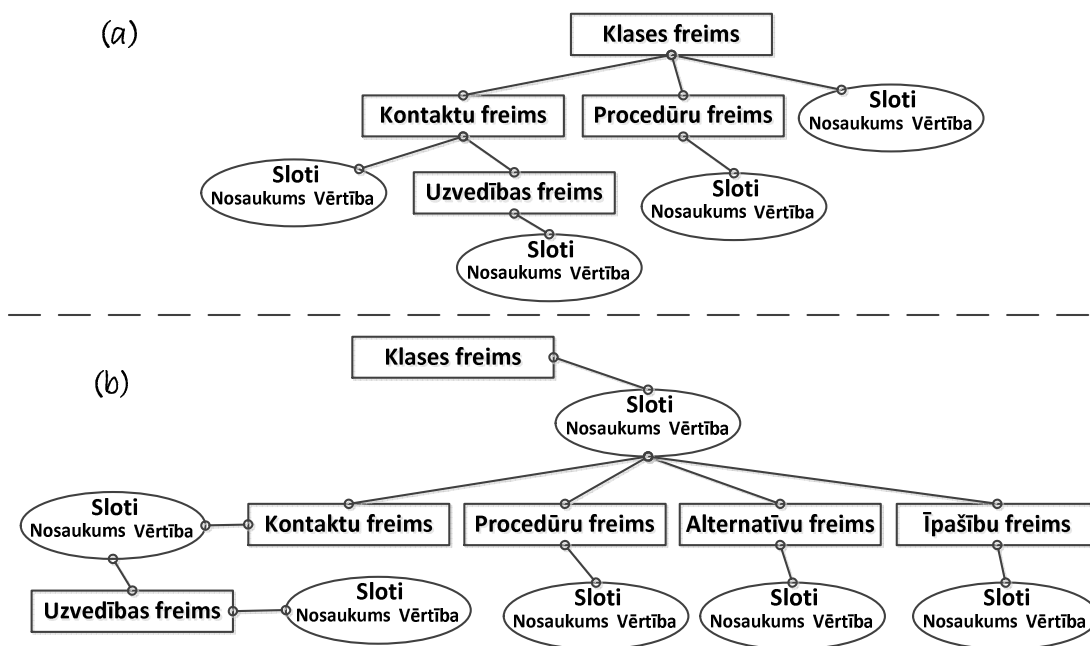
Mantošana freimu sistēmā ir būtisks mehānisms, kas ļauj koplietot atspoguļotās zināšanas starp vairākiem freimiem. Rakstos, kuros skatīta freimos saknota zināšanu atspoguļošana, ir minēts, ka slotus (jeb īpašības) manto no priekšteča uz pēcteci [EVE 1991, KAR 1993, ROB 1977, LAS 1990, EVE 1991, GAN 1993]. Mantošana pēctecim ļauj iegūt dažus vai visus datu elementus no eksistējoša viena vai vairākiem priekštečiem. Tomēr iegūtās iezīmes pēctecim pieder netieši un, ja kaut ko maina priekštecī, attiecībā uz mantoto, tad arī tiek ietekmēti visi pēcteci. Vienkārša mantošana ir tad, ja pēctecis manto no viena priekšteča. Daudzkārša mantošana ir tad, ja pēctecis manto no vairākiem priekštečiem. Daļēja mantošana ir tad, ja manto tikai dažas iezīmes vai kādu daļu no iezīmes, pārējās ignorējot [NIE 1989, YOU 2007, WEI 2009]. Struktūrmodelēšanā, izdala mantotos un lokālos slotus (skatīt 3.4. att.). Ja sloti ir iegūti no cita freima, koplietoti, un netieši pieder apskatītajam freimam, tad tos sauc par mantotiem slotiem. Ja apskatītam freimam slotus definē pirmo reizi, tad tos sauc par lokāliem.

Freimi atbalsta hierarhisku atspoguļojumu un zināšanu organizēšanu. Izmantojot hierarhijas principu, izvēlēto sistēmu var atspoguļot dažādos abstrakcijas līmeņos no ļoti vispārēja līdz detalizētam. Augstākā līmeņa freimi hierarhijā atspoguļo lietas/objektus/realitātes, kas bieži nemainās, ir abstraktas un vienmēr patiesas attiecībā uz atspoguļoto situāciju/sistēmu. Zemākā līmeņa freimi ir skaidri definēti un atspoguļo specifisku situāciju gadījumus, eksemplārus vai sistēmas elementus. Detalizētāk hierarhijas, ko ir iespējams atspoguļot freimos, ir apskatītas pie freimu kopas apraksta. Svarīgākās freimu īpašības ir specifiska atspoguļošanas forma, mantošana un klases–apakšklases hierarhija [GRU 1997a, ZEL 2010a].

3.1.2. Freimu kopas apraksts

Dažādos literatūras avotos minētās, uz freimiem balstītās zināšanu atspoguļošanas shēmās (skatīt 1.pielikumu) definē tikai viena veida freimu, turpretim citās izšķir divus vai vairākus freimu tipus (piemēram, klases freims un eksemplāra freims) [CHA 1978, RAM 1997, KAR 1993]. Atbilstoši SM pieejas aspektiem freimu kopā izdala vienu klases freimu un vienu vai vairākus kontaktu, procedūru (likumu), īpašību (kontekstuālie), alternatīvu un uzvedības freimus. Darba ietvaros izveidota jauna freimu kopa (skatīt 3.5. att. (b)), jo iepriekš izmantotai (3.5. att. (a)) freimu kopai [ZEL 2007] bija dažādi trūkumi:

- netika iekļauti īpašības freimi - līdz ar to nebija iespējams pievienot norādes freima īpašībām un to vērtībām. Tas ir būtiski, kad nepieciešams attēlot papildus zināšanas un kontekstu kādai īpašībai;
- nebija iekļauti alternatīvu freimi – līdz ar to attēlotam objektam nebija iespējams norādīt alternatīvas. Tas ir būtiski, ja nepieciešams attēlot objekta alternatīvas, veicot sistēmas projektēšanu, vai alternatīvas, kuras var lietot sistēmas modelī apskatītā objekta vietā pie mainītiem funkcionalitātes nosacījumiem;
- netika detalizēti skaidrota sasaiste starp freimiem, kā arī freimi kopā tika savstarpēji saistīti, neizmantojot slotus.

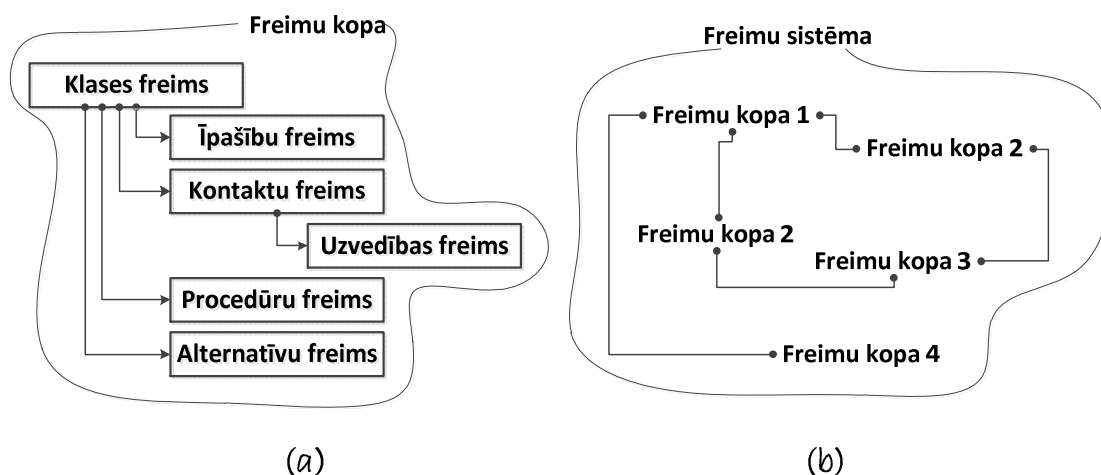


3.5. att. Freimu kopas

Darba ietvaros izstrādātajā jaunajā freimu kopā pielietojumu un specifiskās iezīmes katram freimam nosaka slots, kuram tas ir piesaistīts. Freimu kopā, freimi ir ne tikai saistīti ar norādēm caur slotiem, bet arī organizēti hierarhijā, kas ļauj īstenot mantošanu freimu kopas ietvaros [GRU 1997a, VAL 2005b, ZEL 2010a]. Klases freimam pēcteci ir īpašību, kontaktu, procedūru un alternatīvu freimi (skatīt 3.6. att. (a) gadījumu). Savukārt kontaktu freimam pēctecis ir uzvedības freims. Ja eksperts nolemj veikt izmaiņas, tad freimu kopas klases freimu var aizstāt ar alternatīvu freimu. Savukārt īpašības freims var būt klases freims citā freimu sistēmā.

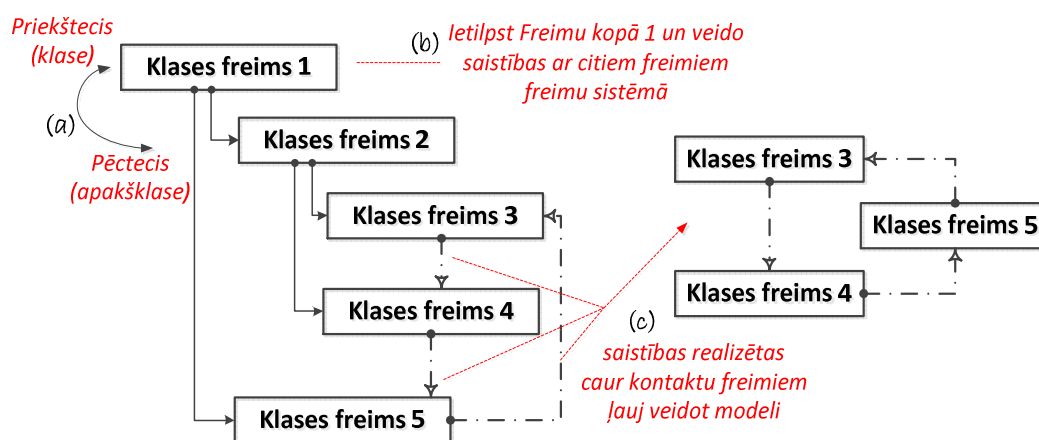
Katra freimu kopa ir savstarpēji saistīta ar citām freimu kopām un veido:

- freimu sistēmu (skatīt 3.6. att. (b) gadījumu), ar norādēm, kas piesaistītas slotiem un freimiem (skatīt 3.7. att. (a), (b) un (c) gadījumu);
- freimu hierarhiju, ar klases freimiem un to iezīmi „priekštecis” (skatīt 3.7. att. (a) gadījumu);
- modeli, kas attēlo noteiktu apskatītās sistēmas detalizācijas līmeni – ar kontaktu freimiem freimu kopā (skatīt 3.7. att. (c) gadījumu).



3.6. att. Freimu kopa un freimu sistēma

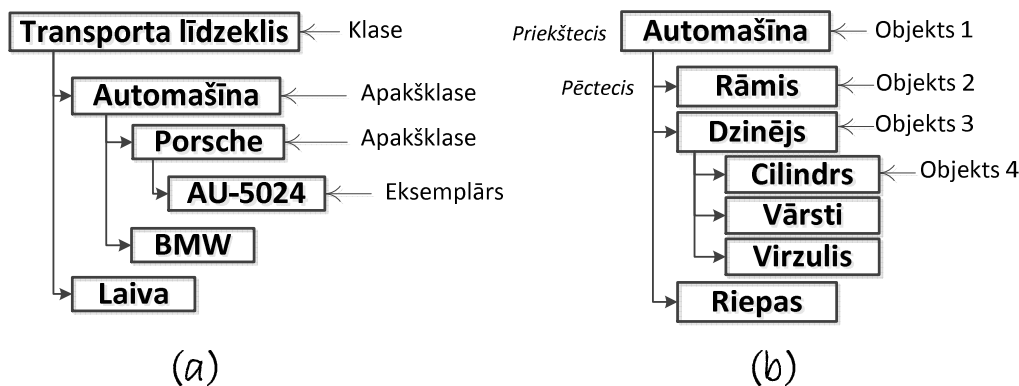
Klases freimi un atbilstošā freimu kopa, kurā tie ir iekļauti, ir sakārtoti taksonomiskā hierarhijā, un katrs klases freims ir saistīts ar vienu priekšteča freimu (skatīt 3.7. att.). Sarežģītu sistēmu gadījumā sistēmas noteiktu detalizācijas līmeni var veidot ne tikai viens, bet dažādi freimu hierarhijas līmeņi. Tāpēc, veidojot struktūras modeļus, būtiskas ir tieši saistības, ko īsteno ar kontaktu freimiem.



3.7. att. Saistības freimu sistēmā

Freimu sistēmā var atspoguļot objektu hierarhiju un klašu hierarhiju [MIN 1975a, KAR 1993, GRU 1999] (skatīt 3.8. att.). Gan klašu, gan objektu hierarhijas var būt savstarpēji saistītas (ar norādēm) un aprakstītas vienas freimu sistēmas ietvaros. Klases vai tipa hierarhijas, zināmas arī kā mantošanas hierarhijas, veido, izmantojot vispārināšanas – specializēšanas saites *ir* (no angļ.val IS-A) vai *ir tips* (angļu val. A-KIND-OF), un objekti šādās hierarhijās ir sistemātiski sadalīti klasēs, ievērojot to kopīgās, līdzīgās pazīmes [KAR 1993, SHA 2003, GAN 1993, YOU 2007]. Objektu hierarhijas veido, lietojot saiti *ir daļa* (no angļu val. A-PART-OF) [KAR 1993, KAK 2003]. Objektu hierarhijā objekti ir sakārtoti pēc

noteiktiem sistēmas uzbūves principiem un atspoguļo sistēmas morfoloģiju, vadības iespējas un organizētību [KAR 1993, ZEL 2010a]. Ja apskata tehnisku sistēmu, tad lieto objektu hierarhiju (skatīt 3.8. att. (b) gadījumu), savukārt, ja izvēlas attēlot objektu klasifikāciju, tad izmanto klašu hierarhiju (skatīt 3.8. att. (a) gadījumu).



3.8. att. Klases un objektu hierarhijas

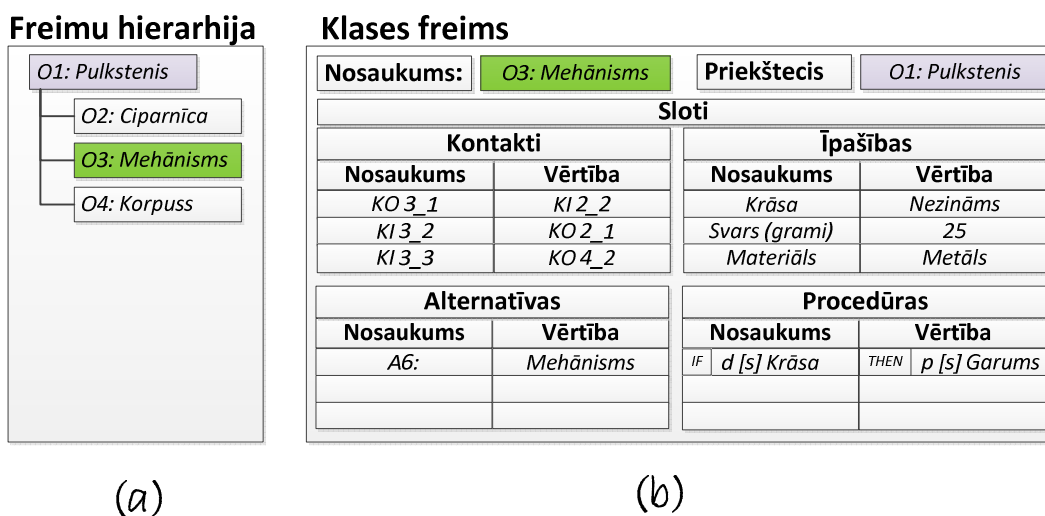
Tālāk ir detalizēti apskatīti freimu kopā iekļautie freimi, to īpašības un zināšanu atspoguļošanas iespējas, kas ir būtiskas attiecībā uz SM un sarežģītām sistēmām. Lai atspoguļotu atsevišķu objektu, intelektuālā sistēmā lieto freimu kopu. Vairākas saistītas freimu kopas veido freimu sistēmu un ļauj aprakstīt izpētes sistēmu. Freimu kopā norāda objekta nosaukumu, objekta īpašības (to nosaukumus un vērtības), funkcijas, uzvedību, likumus, kontaktus un saites, kas atspoguļo objekta saistību ar citiem objektiem, un citas objektam būtiskās īpašības [GRU 1997a, ZEL 2008b, ZEL 2008b]. Darba autore katram freimam, kas ir iekļauts freimu kopā ir izveidojusi vizualizētu attēlojumu, kas ir implementēts I4S lietojumā.

Klases freims ir pirmais freims kopā, ar kuru uzsāk secīgu izpētes objekta atspoguļošanu. Ja izveidotais klases freims ir pirmais pievienotais, tad tas atbilst sistēmai, ko pēta. Ja nav pirmais, tad to atspoguļo freimu hierarhijā, zem iepriekš iezīmēta freima (skatīt 3.9. att. (a) gadījumu), kas kļūst tam par priekšteci (piemēram, *O1: pulkstenis* kļūst par priekšteci *O3: Mehānisms*).

Klases freims sistēmā I4S, kā parādīts 3.9. att. (b) gadījumā, sastāv no:

- 1) **freima nosaukuma**, kurā atspoguļo objekta identifikatoru (*O3:*) un nosaukumu (*Mehānisms*). Objekta nosaukums ir vārds vai abreviatūra, kas ir atpazīstams kā apzīmējums izpētes objektam un kalpo par identitātes pierādījumu. Freima nosaukums var sastāvēt no vairākiem vārdiem un būt līdz 75 simboliem garš. I4S

freima identifikatoru piešķir automātiski un ekspertam ir jānorāda tikai objekta nosaukums;



3.9. att. Freimu hierarhija un klases freims

- 2) **priekšteča nosaukuma**, kurā atspoguļo priekšteča identifikatoru (O1:) un nosaukumu (Pulkstenis). Priekšteča nosaukums ļauj noteikt apskatītā freima atrašanās vietu freimu hierarhijā. Sistēma I4S priekšteča nosaukumu klases freimā piešķir un attēlo automātiski, ievērojot iepriekš iezīmēto freimu. Ja freims ir pirmais pēc kārtas freimu hierarhijā, tad freimu sistēmā tam nav priekšteču, un priekštecim piešķir vērtību „NAV”;
- 3) dažādas nozīmes **slotiem**, kas, atbilstoši freimu pieraksta veidam, sastāv no nosaukuma un vērtības vai vērtību saraksta. Klases freimā izšķir četrus dažādus slotu tipus:
 - a. **kontaktu** slotus, kas atspoguļo apskatītā objekta mijiedarbības ar citiem objektiem sistēmā. Katrā kontaktu slotā attēlo vienu ieejas un izejas kontaktu pāri. Kontaktu slota nosaukumā norāda kontaktu, kas pieder apskatītam objektam. Slota vērtība atbilst kontaktam, kas pieder citam objektam sistēmā. Ja slota nosaukumā ir apskatīts ieejas kontakts, tad slota vērtībai I4S automātiski piešķir izejas kontaktu un otrādi. Piemēram, slots {KO 3_1: KI 2_2}. Katrs kontaktu slots sistēmā I4S ir saistīts ar vienu kontaktu freimu;
 - b. **īpašību** slotus, kas atspoguļo izpētes objektam raksturīgās iezīmes. Tās var būt kvalitatīvas vai kvantitatīvas. Piemēram, slots {krāsa: sarkans}, kur *krāsa* ir slota nosaukums un *sarkans* ir slota vērtība; vai slots {garums (cm):

4}, kur slota nosaukums *garums (cm)* un slota vērtība ir 4. Katrs īpašību slots I4S ir saistīts ar vienu vai vairākiem īpašību freimiem;

- c. **alternatīvu** slotus, kas atspoguļo alternatīvus objektus, ko, veidojot modeļus, iespējams izmantot apskatītā objekta vietā, un kas nodrošina atbilstošu (līdzvērtīgu) funkcionēšanu sistēmā. Pie slota nosaukuma norāda identifikatoru (piemēram, *A6:*), bet pie slota vērtības attēlo objekta nosaukumu (piemēram, *Mehānisms*). Un, lai gan dotais objekta nosaukums ir vienāds ar apskatītā objekta nosaukumu, tomēr identifikatori ir atšķirīgi;
- d. **procedūru** slotus, kas ļauj atspoguļot kritērijus un instrukcijas attiecībā uz apskatīto objektu produkciju likumu formā. Likumi ļauj īstenot darbības vai soļu secību, kas dinamiski rada izmaiņas atspoguļojumā. Slota nosaukumam automātiski piešķirt atslēgas vārdu IF, bet slota vērtībai atslēgvārdu THEN. Izstrādātajā sistēmā I4S pievienojot slotu, tā nosaukumā papildus ir jāapraksta kritēriji jeb darbības, kuru rezultātā atspoguļojumā ir nepieciešams veikt izmaiņas. Savukārt, pie vērtības ir jānorāda darbības, kuras I4S veic automatizēti, ja izpildās definētie kritēriji. Gan slota nosaukumā, gan vērtībā izmanto šādus apzīmējumus:
 - i. **p** – ja apskatītajam freimam pievieno pēcteci (attēlo kā **p** [fp]) vai slotu (**p** [s]), vai citu freimu tai pašā detalizācijas līmenī (**p** [f]);
 - ii. **m** – ja apskatītajam freimam maina freima nosaukumu (apraksta kā **m** [f]), slotu (**m** [s]), slota nosaukumu (**m** [sn]) un slota vērtību vai vērtību sarakstu (**m** [sv]);
 - iii. **d** – ja apskatītajam freimam dzēš pēcteci (atspoguļots kā **d** [fp]), slotu (**d** [s]), un slota vērtību vai vērtību sarakstu (**d** [sv]), vai dzēš apskatīto freimu (**d** [f])

Aprakstot kritērijus, lieto simbolus, ar šādu nozīmi: 1) **f** – freims; 2) **fp** – freima pēctecis; 3) **s** – slots; 4) **sn** – slota nosaukums; 5) **sv** – slota vērtība. Papildus apzīmējumiem norāda arī noteiktus freimu un slotu nosaukumus.

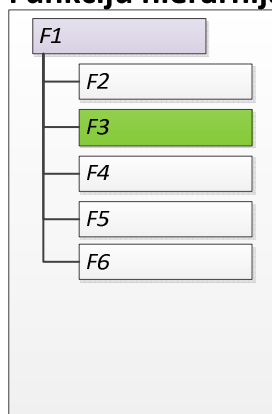
Struktūrmodelēšanā attiecībā uz sistēmas daļām apraksta dažāda veida attieksmes: saites, kas attēlo hierarhiju (objektu, klašu un funkciju) un saites starp sistēmas daļām (objektu mijiedarbības). Objektu un klašu hierarhijas veido, izmantojot freimu hierarhiju un klases freimus, bet funkciju hierarhiju un saites starp daļām veido, lietojot kontaktu freimu. Kontaktu freims ir būtisks struktūrmodelēšanā, jo tajā norāda visas objekta īpašības, kas ir

saistītas ar noteiktas plūsmas īstenošanu: plūsmas veidu, funkciju, atbilstošās īpašības, uzvedību, parametrus.

Kontaktu freimā ir realizēta ideja par funkcionālo īpašību detalizēšanu – funkciju apskata kā elementu, ko var pakārtot citām funkcijām funkciju hierarhijā (skatīt 3.10. att. (a)). Kontakta freimu vizuāli attēlo līdzīgi kā klases freimu (3.10. att. (b)), bet to apraksta, izmantojot:

- 1) **freima nosaukumu**, kurā atspoguļo funkcijas, ko īsteno apskatītās plūsmas ietvaros, identifikatoru ($F3$). Funkcijas nosaukumu I4S piešķir automātiski, atbilstoši funkciju skaitam sistēmā.
- 2) **priekšteča nosaukumu**, kurā atspoguļo priekšteča identifikatoru ($F1$). Ja tā ir pirmā funkcija sistēmā, tad pie priekšteča vērtības atspoguļo „Nav”. Priekšteča nosaukums ir jānorāda, izvēloties starp funkcijām, kas atbilst apskatītā objekta priekštecim;

Funkciju hierarhija



(a)

Kontaktu freims

Nosaukums: F3		Priekštecis: F1	
Sloti			
Plūsma		Īpašības	
Nosaukums	Vērtība	Nosaukums	Vērtība
$f3$	<i>enerģija</i>	<i>Impulss (sec)</i>	<i>1</i>
$F3$	<i>laika atspoguļošana</i>		
<i>ieejas</i>	$(f2,f9)$		
<i>izejas</i>	$(f3,f4)$		
Uzvedība		Procedūras	
Nosaukums	Vērtība	Nosaukums	Vērtība
<i>US 3_1_1</i>	<i>US 2_2_1</i>	<i>IF m [ff] F1</i>	<i>THEN d [s] Impulss</i>
<i>US 3_1_2</i>	<i>US 2_2_2</i>		

(b)

3.10. att. Funkciju hierarhija un kontaktu freims

- 3) dažādas nozīmes **slotus**:

- a. **plūsmas** sloti atspoguļo funkcionālās īpašības plūsmā. Ievadot kontaktu freimu, ir noteikti jānorāda: (a) plūsmas identifikators ($f3$) un nosaukums (piemēram, *enerģija*), (b) funkcijas identifikators ($F3$) un nosaukums (piemēram, *laika atspoguļošana*). Abus minētos slotu nosaukumus I4S izveido automātiski, un ekspertam ir jānorāda slotu vērtības. Plūsmas tipu (ieejas vai izejas) pie plūsmas slotiem nenorāda, jo to attēlo pie kontakta slotiem klases freimā. Plūsma vienlaicīgi var pārnest vairākas vielas, kurām kopumā ir dots viens nosaukums. Piemēram, asinis sastāv no

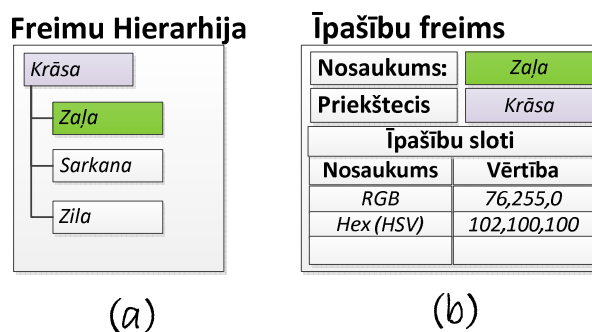
asinsķermenīšiem un asins plazmas. Līdz ar to, veicot izpētes sistēmas detalizāciju, plūsmu un tās īstenoto funkciju detalizē. Plūsmas slotos norāda arī loģiskos operatorus, kas saista apskatīto plūsmu ar citām plūsmām, kuras eksistē izpētes objektam. Piemēram (skatīt 3.10. att. (b)), plūsma f3 vai plūsma f4 (f3;f4) iziet no objekta O3, tad, ja objekta ieejā ir plūsmas f2 un f9 (f2,f9);

- b. **īpašību** sloti, kas, līdzīgi kā klases freimā, atspoguļo izpētes objekta raksturīgās iezīmes. Šajā gadījumā izpētes objekts ir plūsma. Piemēram, slots {impulss (sec): 1} (skatīt 3.10. att. (b)). Ja plūsma ir matērija, tad tai ir iespējams norādīt krāsu, konsistenci, utt.;
- c. **uzvedības** sloti atspoguļo apskatītā objekta uzvedības stāvokļus. Viens uzvedības stāvoklis atbilst uzvedībai, ko īsteno līdz ar apskatīto plūsmu. Var norādīt noklusēto un alternatīvus uzvedības stāvokļus. Noklusētais uzvedības stāvoklis ir pirmais slotu sarakstā. Uzvedības slotā nosaukumā norāda uzvedības stāvokli, kas pieder apskatītajam objektam noteiktā kontaktā. Slotā vērtība atbilst uzvedības stāvoklim, kas, līdz ar plūsmu, ir saistīts ar slotā nosaukumā norādīto uzvedības stāvokli. Piemēram, slots {US 3_1_1: US 2_2_1}, kur slotā nosaukums *US 3_1_1*, atbilst trešā objekta, pirmā kontakta pirmajam (aktuālajam) uzvedības stāvoklim, bet slotā vērtība *US 2_2_1* atbilst otrā objekta, otrā kontakta pirmajam uzvedības stāvoklim. Alternatīvu uzvedības stāvokli var izmantot, ja eksperts veic izmaiņas atspoguļojumā, vai ja I4S izmaiņas veic automātiski, sekojot eksperta definētām procedūrām procedūru slotos. Struktūras modeļos atspoguļo tikai noklusēto uzvedības stāvokli. Katrs uzvedības slots I4S ir saistīts ar vienu uzvedības freimu.
- d. **procedūru** sloti, kontaktu freimā, tāpat kā klases freimā, ir apskatīti, izmantojot tos pašus apzīmējumus. Tikai šajā gadījumā, ja aplūko freima labošanu vai pievienošanu (m [f]), tad tas attiecas nevis uz pašu freimu vai freima nosaukumu, bet priekšteča nosaukumu.

Objektiem un plūsmām var būt dažādas īpašības. Līdz ar to klases freimā ir norādīti īpašību sloti, kas ir saistīti ar īpašību freimiem. Īpašību slotam un freimam ir informatīva nozīme, un tie veido izpratni par izpētes objektu, un nosaka kontekstu, kādā tas ir skatīts. Īpašību freimā veido freimu hierarhiju (skatīt 3.11. att. (a)), kas atspoguļo saistītus jēdzienus.

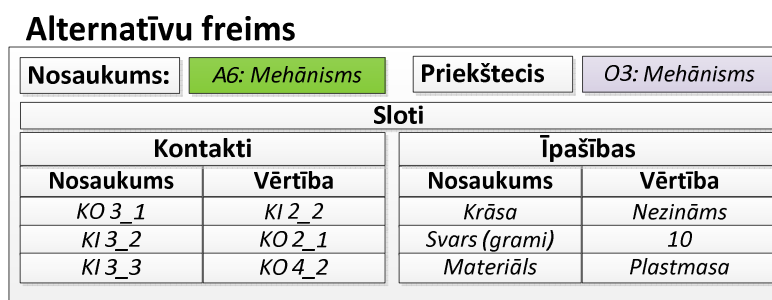
Īpašību slotā apskatīto slota nosaukumu attēlo kā pirmo freimu hierarhijā, savukārt vērtības kā tam pakārtotus freimus. Īpašību freims, kā tas ir attēlots 3.11. att. (b) attēlā, sastāv no:

- 1) **freima nosaukuma**, kurā apraksta iezīmi, kas kopā ar citām saistītām iezīmēm, (attēlotas freimu hierarhijā), raksturo izpētes objektu (piemēram, *Zaļa* un *Krāsa*).
- 2) **priekšteča nosaukuma**, kurā attēlo priekšteča nosaukumu (piemēram, *Krāsa*);
- 3) **īpašību slotiem**, kuros atspoguļo detalizētu informāciju par apskatīto iezīmi. Piemēram, krāsu *zaļa*, var raksturot ar krāsu modeli RGB (no angļu val. Red, Green, Blue), kur trīs pamatkrāsas, savienotas kopā noteiktā intensitātē, veido apskatīto krāsu specifiskā tonī. Īpašību freimā apskatītos īpašību slotus nedetalizē, jo freimu kopā tiem nav paredzēts pievienot norādes uz citiem freimiem.



3.11. att. Freimu hierarhija un īpašību freims

Alternatīvu freimā atspoguļo apskatītā izpētes objekta alternatīvu (skatīt 3.12. att.). Alternatīvu freims var mantot klases freimā atspoguļotos kontaktu un īpašību slotus, ko tālāk var mainīt, atbilstoši alternatīvai raksturīgajām iezīmēm. To, vai alternatīvu freims manto kaut ko no klases freima, nosaka eksperts.



3.12. att. Alternatīvu freims

Alternatīvu freims sastāv no:

- 1) **freima nosaukuma**, kurā attēlo objekta alternatīvas identifikatoru un nosaukumu (*A6: Mehānisms*).

- 2) **priekšteča nosaukuma**, kurā norāda priekšteča identifikatoru, kas atbilst klases freimā attēlotājam izpētes objektam (*O3: Mehānisms*);
- 3) divu veidu **slotiem**, kuru nozīme ir identiska klases freimā aprakstītajiem slotiem:
 - a. **kontakta**;
 - b. **īpašību**.

Dinamisku skatījumu uz sistēmu var iegūt, ja zināšanas par sistēmu atspoguļo izmaiņu gadījumā. Tas nozīmē, ka izpētes sistēmu novēro, analizē un struktūras modelī atspoguļo sistēmā notikušās izmaiņas. Lai attēlotu dinamiku, I4S lieto procedūru slotus, kuriem ir piesaistīti procedūru freimi. Procedūru freimā apskata likumus, kas ietekmē apskatīto objektu, plūsmu vai uzvedību, un kas ļauj veikt izmaiņas sistēmas atspoguļojumā. Procedūru freimā norāda (skatīt 3.13. att.):

- 1) **freima nosaukumu**, kurā attēlo procedūras identifikatoru, kas sastāv no:
 - a. simboliem „Pr”;
 - b. kārtas skaitļa, kas norāda objekta numuru, kuram procedūra ir pievienota;
 - simbola „_”, kas atdala objekta numuru un procedūras numuru;
 - skaitļa, kas norāda, procedūras kārtas numuru, ar kādu tā pievienota attiecīgā objekta atspoguļojumam.
- 2) **procedūru slotus**, kuros attēlo vienu procedūru. Slota nosaukumos lieto rezervētos vārdus **IF** (nosacījumam), **AND** (loģiskais operators UN), **OR** (loģiskais operators VAI), **THEN** (secinājumam/ darbībai) un pie klases freima minētos apzīmējumus (piemēram, *m [f]*). Pie slotu vērtībām norāda, ko darīt izmaiņu gadījumā. Ja ir pievienoti sloti, kuru nosaukumi sākas ar rezervēto vārdu *IF* vai *THEN*, tad pēc tiem vairs nevar pievienot citus slotus ar tādiem pašiem rezervētiem vārdiem.

Procedūru freims

Nosaukums:	Pr 3_1
Procedūru sloti	
Nosaukums	Vērtība
<i>IF m [f]</i>	<i>F1</i>
<i>OR m [s]</i>	<i>f1</i>
<i>THEN m [s]</i>	<i>F1</i>
<i>AND m [s]</i>	<i>f1</i>

3.13. att. Procedūru freims

Pēdējais freims freimu kopā ir uzvedības freims (skatīt 3.14. att.). Uzvedības freimā apraksta parametru kopu, kas atbilst uzvedības slotā norādītajam uzvedības stāvoklim. Uzvedības freims sastāv no:

- 1) **freima nosaukuma**, kurā norāda parametru kopas identifikatoru, kas ir saistīts ar uzvedības stāvokļa identifikatoru. Piemēram, ja apskatīts uzvedības stāvoklis *US 3_1_1*, tad parametru kopas identifikators ir *PS 3_1_1*;

Uzvedības freims

Nosaukums: <i>PS 3_1_1</i>	
Sloti	
Parametri	
Nosaukums	Vērtība
<i>P 3_1_1_1</i>	<i>200</i>
<i>P 3_1_1_2</i>	<i>1-10</i>
<i>P 3_1_1_3</i>	<i>15, 25, 30,100</i>
<i>D1</i>	<i>Nezināms</i>
Saites	
Nosaukums	Vērtība
<i>P 3_1_1_1</i>	<i>P 3_1_1_2</i>
<i>P 3_1_1_2</i>	<i>P 3_1_1_3</i>
<i>P 3_1_1_3</i>	<i>P 2_2_1_2</i>
<i>D1</i>	<i>P 3_1_1_2</i>
<i>AND</i>	<i>P 3_1_1_1, D1</i>
Procedūras	
Nosaukums	Vērtība
<i>IF m [s] P 3_1_1_1</i>	<i>THEN m [s] P 3_1_1_2</i>

3.14. att. Uzvedības freims

- 2) dažādiem **slotiem**:

- a. **parametru** slotiem, kuros apraksta parametrus. Slota nosaukumā norāda parametra identifikatoru, bet pie slota vērtības parametra vērtību vai vērtību sarakstu. Parametru slotos var aprakstīt arī defektus (norādot identifikatoru un vērtību);
- b. **saites** slotiem, kuros attēlo parametru iekšējās un ārējās saites. Ārējā saite ir saite starp slota nosaukumā definēto parametru, kas attiecas uz apskatīto parametru kopu, un parametru, kurš atrodas citā saistītā objektā (norādīts klases freima kontaktu slotā). Iekšējās saites norāda saistības starp parametriem apskatītajā parametru kopā. Pie saitēm norāda arī loģiskos operatorus, atspoguļojot saistību starp parametriem un arī defektiem, ja tādi pastāv. Ja loģiskie operatori nav norādīti, tad starp parametru saistībām eksistē loģiskais operators UN;
- c. **procedūru** sloti (skatīt klases freimu).

Freimu kopas izmantošanai sistēmā I4S ir vairākas priekšrocības:

- 1) zināšanu bāzē un lietotājam redzamajā saskarnē freimu kopu lieto kā zināšanu atspoguļošanas shēmu [GRU 1997a, VAL 2005b]. Vienas un tās pašas datu struktūras lietošana nodrošina vienotu un viennozīmīgu atspoguļojumu. Freimu kopas ļauj sistematizēt izgūtās zināšanas;

- 2) Freimu kopā ir iespējams vienlaikus atspoguļot: a) sistēmas kopējo struktūru un sistēmas komponentu izkārtojumu dažādos hierarhijas līmeņos; b) organizētību un tai atbilstošo sistēmas struktūru atsevišķā hierarhijas līmenī;
- 3) Freimu kopā apraksta jebkuru sistēmas daļu: gan sistēmas komponentes, gan elementus, gan arī elementu un komponentu alternatīvas;
- 4) Freimu kopā norāda mijiedarbības starp sistēmas daļām, kā arī likumus, kurus īsteno dažādu izmaiņu gadījumos attiecībā uz apskatīto sistēmas daļu un tās iezīmēm;
- 5) Freimu kopā var atspoguļot gan objektu, gan klašu hierarhiju – tātad aprakstīt gan sistēmas struktūru, gan ontoloģijas [GRU 1999, VAL 2005c, ZEL 2010a, ZEL 2010b].

3.2. Sistēma I4S un transformācijas no freimu kopas

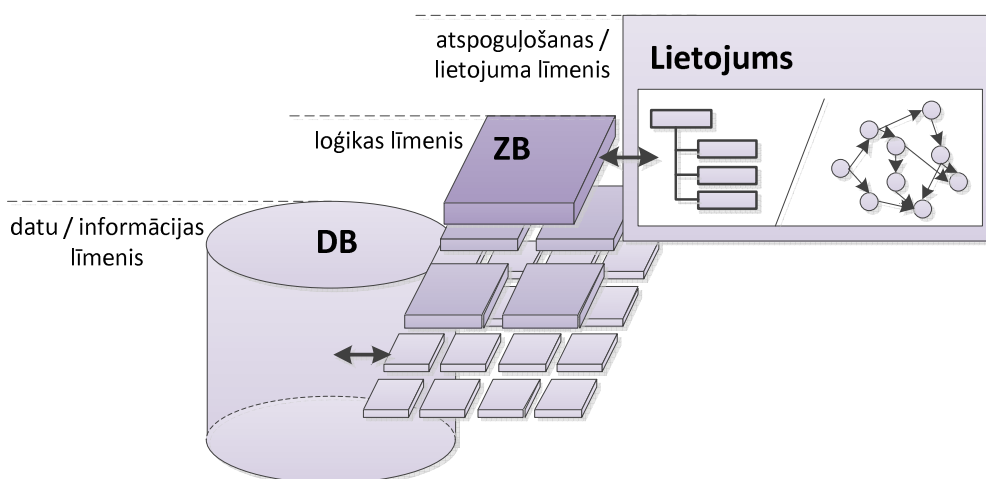
Intelektuāla sistēma sarežģītu sistēmu struktūrmodelēšanai (I4S) ir datorsistēma, kurā ir implementēta SM pieeja, lai atbalstītu sarežģītu tehnisku sistēmu struktūrmodelēšanu datorā. Sistēma I4S sastāv no 3 pamatkomponentēm: lietojuma, datu bāzes (DB) un zināšanu bāzes (ZB). Lietojums ir izveidots vizualizētā Rad Studio XE2 C++Builder integrētā izstrādes vidē, kas paredzēta lietojumprogrammu izstrādei, un kurā izmanto C++ programmēšanas valodu [HIL 2009]. Sistēmas I4S datu bāze ir izveidota atvērtā koda objektu – relāciju datu bāzes pārvaldības sistēmā PostgreSQL (versija 9.1.) [POS 2012]. Zināšanu bāze ir izveidota, lietojot C++Builder un PostgreSQL programmatūru iespējas. Sistēmas I4S pamatkomponentes ir savstarpēji saistītas un iekļauj sešus moduļus, kas ļauj īstenot sistēmas I4S funkcionalitāti.

Ja lieto automatizētas transformācijas, tad zināšanas par izpētes sistēmu var atspoguļot vienreiz un nepieciešamības gadījumā tās atkārtoti izmantot [STA 2006]. Lai nodrošinātu automatizētu struktūras modeļu konstruēšanu, darba ietvaros izstrādātas transformācijas no freimu kopas uz struktūras modeļiem. Katrai transformācijai ir izveidots atbilstošs algoritms, kas, implementēts sistēmā I4S, no noteiktiem freimu kopas freimiem un slotiem secīgi iegūst nepieciešamās zināšanas modeļu pamatelementu un izvēlēta struktūras modeļa izveidei. Transformācijās no freimu kopas uz struktūras modeļiem ņem vērā arī loģiskos operatorus. Transformācijas sastāv no 4 pamatdaļām: sistēmas dekompozīcijas līmeņa izvēle, modeļa tipa izvēle, zināšanu izgūšana no freimu kopas, atbilstošo pamatelementu atspoguļošana un sasaistīšana. Nodaļas ietvaros ir apskatīts abstraktas sistēmas piemērs, kas ir lietots, lai

uzskatāmi demonstrētu transformāciju algoritmu soļus. Šāds piemērs ir nepieciešams, lai atspoguļotu visus transformācijās iespējamus gadījumus.

3.2.1. Sistēmas I4S arhitektūra un funkcionalitāte

Sistēmas I4S arhitektūru veido 3 līmeņi (skatīt 3.15. att.): (1) datu un informācijas līmenis, kas atbilst pamatkomponentei *datu bāze*, (2) loģikas līmenis, kas atbilst *zināšanu bāzei* un (3) lietojuma līmenis, kas atbilst *lietojumam*. Katrā no šiem līmeņiem ir implementēta pielāgota zināšanu atspoguļošanas shēma – darba ietvaros izveidotā freimu kopa, kas nodrošina līmeņu integritāti.



3.15. att. I4S arhitektūra

Datu / informācijas līmenis sistēmā I4S nodrošina pārējās sistēmas komponentes ar nepieciešamajiem datiem un informāciju. Lai gan datu un informācijas līmenim ar citiem sistēmas līmeņiem pastāv saistība, tas ir ārpus lietojuma līmeņa kontroles, izņemot manipulācijas ar datiem (saglabāšana, dzēšana, labošana) un pieslēgšanos vadības sistēmai (sistēmu I4S var uzsākt lietot tikai tad, ja lietotājs ievada lietotājvārdu un paroli, kas atbilst datubāzē saglabātajiem). Tas nozīmē, ka nepieciešamības gadījumā ir iespējams nodrošināt izstrādātās sistēmas pārvietojamību no esošās datu bāzes uz citu. Datu bāzē izmantotā zināšanu atspoguļošanas shēma tiek īstenota kā tabulu kopums, kas ir savstarpēji saistītas un ļauj saglabāt datus un informāciju formātā, kas atbilst freimu kopas idejai, un nodrošināt izvedumu mašīnas funkcijas (brīdī, kad dati un informācija tiek izgūti). Lietojot datizraces metodes, zināšanu bāzē ievieto zināšanas par izpētes sistēmu, ko tālāk atspoguļo I4S lietojumā – vizualizētos freimos. Datubāzē kopumā ir izveidotas 26 tabulas. Katram freimam

no freimu kopas atbilst noteiktas tabulas datu bāzē (skatīt pielikumu 3.1.– datu bāzes shēmu). Papildus datu bāzē ir tabulas, kas attiecas uz sistēmas darbības nodrošināšanu.

Loģikas līmenis sistēmas I4S gadījumā atbilst zināšanu bāzei (ZB), izveduma mehānismam un sistēmas funkcionalitātes nodrošināšanai paredzētajai loģikai. Dotais līmenis kalpo kā starpposms starp datu bāzi un lietojumu un nodrošina piekļuves ierobežojumu ievērošanu atbilstoši lietotājam noteiktajām tiesībām. Līmenī tiek kontrolētas lietotāja sesijas (pieslēgšanās laiku), datu transakcijas (operācijas ar datiem), kā arī ierobežojumi un nosacījumi, kas attiecas uz zināšanu iegūšanu, atspoguļošanu, saglabāšanu un apstrādi. Zināšanu bāze ļauj izgūt, apkopot zināšanas, lielākoties hierarhiski organizētas. Freimos sakņotās zināšanu atspoguļošanas sistēmās ZB parasti ir freimu un ar tiem saistīto slotu un vērtību kolekcija [KAR 1995]. Sistēmas I4S zināšanu bāzes pamatā ir zināšanu atspoguļošanas shēma un mehānismi, kas nodrošina shēmas piemērošanu sistēmas mērķu un funkcionalitātes realizēšanai: (a) zināšanu izgūšanas mehānismi no eksperta un no DB, (b) zināšanu atspoguļošanas mehānismi, kas ļauj īstenot zināšanu atspoguļošanu freimos, (c) mehānismi, kas nodrošina transformācijas sistēmā – freimos aprakstītās zināšanas transformē un attēlo citā atspoguļošanas formā.

Lietojuma līmenis atbilst lietojuma saskarnei, kā arī moduļiem, kas iekļauj procedūras un likumus, kas nodrošina lietotājam iespēju strādāt ar sistēmu I4S. Sistēmas I4S lietojums sastāv no vairākām formām. Forma ir uz datora ekrāna attēlots sistēmas objekts, kas paredzēts datu/informācijas/zināšanu attēlošanai, labošanai un saglabāšanai. Lietojuma līmenis reaģē uz lietotāja veiktajām darbībām un saistībā ar loģikas līmeni realizē atbildes reakciju uz tām, saskaņā ar sistēmā implementēto funkcionalitāti (skatīt pielikumu 3.2.). Lietojuma līmenis visvairāk ir pakļauts izmaiņām, salīdzinot ar pārējiem diviem sistēmas līmeņiem. Formas elementu vizuālais atspoguļojums parasti attiecas uz sistēmas nefunkcionālām prasībām, tomēr sistēmas I4S gadījumā dotā prasība ir nozīmīga sistēmas darbības pilnvērtīgai nodrošināšanai. Tas ir tāpēc, ka zināšanu iegūšanas un atspoguļošanas mehānismu realizē (datu un informācijas sasaistīšana ar likumiem, veidojot vienotu veselu), izmantojot lietojumā implementēto freimu kopas vizualizētu atspoguļojumu. Eksperts strādā ar sistēmas I4S lietojumu, veicot izpēti sistēmas aprakstu (praktiskais piemērs ir apskatīts 4. nodaļā).

Sistēmas I4S pamatkomponentes iekļauj sešus moduļus:

1. Datu bāzes vadības modulis, kas nodrošina pieslēgšanos datu bāzei un ar DB saistīto bieži lietoto funkciju īstenošanu:

- a. lietotāja pieslēgšanos DB, izveidojot lietotāja sesiju, kas ļauj strādāt ar sistēmu I4S. Lietotājs ir eksperts, kas strādā ar sistēmu, vai administrators, kas veic sistēmas I4S uzturēšanu;
 - b. sistēmas I4S pieslēgšanos DB, lai sameklētu un nolasītu vai ierakstītu datus;
 - c. datu pārbaudes (piemēram, pārbauda vai izvēlētā vērtība jau neatrodas DB), iegūšanas (piemēram, pēdējā pievienotā freima vai slotu identifikatora iegūšana), kopēšanas (piemēram, freimu, slotu kopēšana) un ievietošanas funkcijas (piemēram, ievieto freimu, slotu nosaukumus, identifikatorus DB).
2. Palīgfunkciju modulis, kas ļauj realizēt ievadīto datu pārbaudes un iestatīt dažādus parametrus sistēmas I4S formās. Piemēram, I4S tiek veiktas šādas palīgfunkcijas:
- a. formās redzamo sarakstu parametru (piemēram, kolonu platums) uzstādīšana un mainīšana;
 - b. ievadei paredzēto lauku pārbaude (piemēram, vai ir ievadīta vērtība un vai tā atbilst sagaidāmai vērtībai);
 - c. vairāku ievadīto vērtību pārbaude (piemēram, pārbauda vai ievadītās vērtības nedublējas);
 - d. ievadītās simbolu virknes sadalīšana (piemēram, no eksperta norādītām plūsmu kombinācijām, sistēma I4S iegūst atsevišķus plūsmu nosaukumus);
 - e. loģisko operatoru iegūšana. Sistēma I4S pārveido eksperta ievadītos simbolus: (a) „,” par AND, (b) „;” par OR un (c) „:” par XOR;
 - f. ievadītās loģikas pārbaude – tiek noteikts vai atbilst iekavu skaits, vai ievadīti pareizi plūsmu nosaukumi, vai nav ievadīti lieki simboli;
 - g. iezīmētās vērtības kārtas numura noteikšana hierarhijā. Piemēram, nosaka izvēlētā freima kārtas numuru freimu hierarhijā.
 - h. pirmā līmeņa pēcteču vai visu pēcteču identifikatoru noteikšana. Piemēram, izmanto kopējot un dzēšot freimus.
3. Lietotāju sesiju modulis, kas kontrolē ar sistēmas lietotāju saistītos parametrus:
- a. lietotāja datu pārbaude (piemēram, lietotāja pieslēgšanās laiks sistēmai I4S, lietotājam pieejamās izpētes sistēmas);

- b. ierobežojumu uzstādīšana. Administrators var uzstādīt ievades ierobežojumus (piemēram, maksimālo alternatīvu skaitu, ko var ievadīt izpētes sistēmai) konkrētam lietotājam.
4. Datu dzēšanas modulis, kas pēc lietotāja norādītās izvēles īsteno datu/informācijas/zināšanu dzēšanu. Piemēram, lietotājs izvēlas dzēst slotus. Sistēma I4S nosaka kādam freimam sloti pieder, kā izvēlēto slotu dzēšana ietekmēs (citus slotus un/ vai freimus) un paziņo par to lietotājam. Ja lietotājs apstiprina savu izvēli, tad I4S īsteno dzēšanu lietojumā, zināšanu un datu bāzē.
 5. Lietojuma vadības modulis nodrošina mijiedarbību starp sistēmu I4S un tās lietotāju. Modulis kontrolē un pārvalda arī saskarnē veiktās darbības, kas saistītas ar sistēmas I4S funkcionalitātes īstenošanu, piemēram:
 - a. zināšanu atspoguļošanu, labošanu, dzēšanu un saglabāšanu (piemēram, atsevišķa objekta aprakstu);
 - b. datu atjaunošanu (piemēram, hierarhijās, tabulās);
 - c. pārvietošanos pa formas elementiem un ekspertam nepieciešamo zināšanu atspoguļošanu;
 - d. automātisku identifikatoru piešķiršanu (piemēram, uzvedības stāvokļiem);
 - e. ar kontaktu slotiem saistīto elementu izveidi. Brīdī, kad eksperts pievieno kontaktu slotu, sistēma I4S, ievērojot SM pieejas pamatprincipus, izveido kontaktu slotu arī saistītam freimam, kā arī plūsmu, funkciju, uzvedības stāvokļu un parametru kopu nosaukumus.
 6. Modeļu un matricu ģenerēšanas modulis, kurā ir implementēti visi transformācijas algoritmi no freimu kopas uz struktūras modeļiem, kā arī uz notikumu kokiem (skatīt 3.2.2 un 3.2.3 nodaļas). Modulis izmanto eksperta atspoguļotās zināšanas, lai izpētes sistēmai izvēlētajā detalizācijas līmenī, automatizēti konstruētu struktūras modeļus un matricas. Modulī tiek īstenotas šādas funkcijas:
 - a. zināšanu iegūšana, kas nepieciešama visu struktūras modeļu, kā arī matricu konstruēšanai. Transformāciju veikšanai nepieciešamo sarakstu izveide. I4S ņemt vērā arī to vai ir nepieciešams atspoguļot plūsmu nosaukumus, vai modeli veidot sistēmai vai apakšsistēmai;

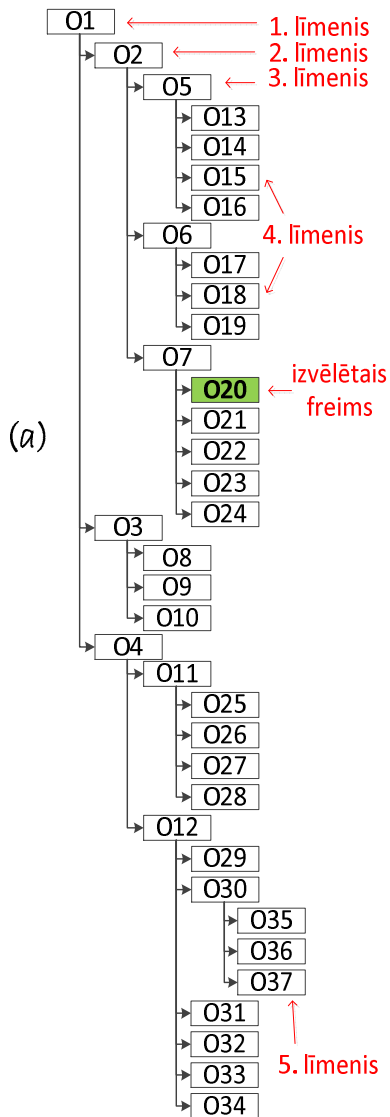
- b. modeļu un matricu klases objektu⁸ izveide. Piemēram, ņemot vērā eksperta izvēlēto struktūras modeļa tipu, sistēma I4S izvēlas atbilstošos sarakstus un izveido virsotnes un saites starp tām (bet vēl tos nezīmē). Sistēma I4S izveido modeļa klases objektus atbilstoši izvēlētajā struktūras modeļa tipa sintaksei;
- c. izveidoto klases objektu zīmēšana. Sistēma I4S lietotājam redzamajā formā vizuāli attēlo izveidotos klases objektus;
- d. lokālo datu (attiecībā uz izveidoto modeli) iegūšana, lai īstenotu: (a) notikumu koku izveidi, (b) saistību meklēšanu starp diviem sistēmas elementiem; (c) matricu reizināšanu; (d) sistēmas struktūras topoloģisko un kvalitatīvo analīzi.

3.2.2. Transformācija no freimu kopas uz struktūras modeļiem

Lai veiktu transformāciju no freimu kopas uz morfoloģiskās struktūras modeli, lieto freimu sistēmas hierarhiju, klases un kontaktu freimus (skatīt 3.16. att.). Transformācijas procesā iegūst informāciju, kas ir nepieciešama modeļa izveidei, vizualizē un sasaista morfoloģiskā modeļa pamatelementus un izveido MSM. Transformācija no freimu kopas uz MSM sastāv no 7 soļiem:

- 1) Izvēlas noteiktu dekompozīcijas līmeni freimu sistēmas hierarhijā, ko attēlo koka veidā (skatīt 3.16. att. (a) gadījumu). Katrs freims un tajā atspoguļotais objekts atrodas noteiktā dekompozīcijas līmenī. Apskatītajā piemērā pirmajā līmenī atrodas O1, otrajā O2, O3 un O4, bet izvēlētais freims O20 atrodas ceturtnā dekompozīcijas līmenī;
- 2) Nosaka kādu struktūras modeli veidot (MSM vai kādu no FSM veidiem), un vai to konstruēt visai sistēmai izvēlētajā detalizācijas līmenī, vai arī tikai noteiktai apakšsistēmai. Piemērā izvēlētais modelis ir MSM visai izpētes sistēmai ceturtnajā detalizācijas līmenī. Ekspertam ir jānorāda arī kādu no MSM veidiem attēlos;

⁸ Klases objekts ir realitāte, kas tiek lietota, lai atspoguļotu vai modelētu kādu izstrādātās sistēmas daļu [WEI 2009].



(b)

Klases freims

Nosaukums:	O20	Priekštecis:	O7
Sloti			
Kontakti		Īpašības	
Nosaukums	Vērtība	Nosaukums	Vērtība
KO 20_1	KI 22_1		
KO 20_2	KI 21_1		
KI 20_3	KO 36_1		
Alternatīvas		Procedūras	
Nosaukums	Vērtība	Nosaukums	Vērtība
		IF	THEN

(c)

Kontaktu freims

Nosaukums:	F1	Priekštecis:	F80
Sloti			
Plūsma		Īpašības	
Nosaukums	Vērtība	Nosaukums	Vērtība
f1	matērija		
F1	eļļas piegāde		
leejas	(f3)		
Izejas	(f1,f2)		
Uzvedība		Procedūras	
Nosaukums	Vērtība	Nosaukums	Vērtība
US 20_1_1	US 22_1_1	IF	THEN

3.16. att. Freimu sistēmas hierarhija un klases, kontaktu freimi

- 3) No freimu hierarhijas iegūst sistēmas objektu nosaukumus un ievietoto **objektu sarakstā** (skatīt 3.17. att. (a)), un pievieno atzīmi „+” kolonā „Detalizēt” (šo kolonu algoritmā izmanto, lai izveidotu pareizu FSM FT virsotņu kopu, kas atbilst izvēlētam sistēmas detalizācijas līmenim). No klases freimiem, kuriem ir **kopējs priekštecis** ar sākotnēji izvēlēto freimu, tiek iegūti objektu nosaukumi. Piemērā izvēlētais freims ir O20 un tā priekštecis ir O7, tāpēc objektu sarakstā tiek ievietoti objekti O20, O21, O22, O23 un O24.

(a)

Objekts	Apskatīts	Detalizēt
O20	+	+
O21	-	+
O22	-	+
O23	-	+
O24	-	+

(b)

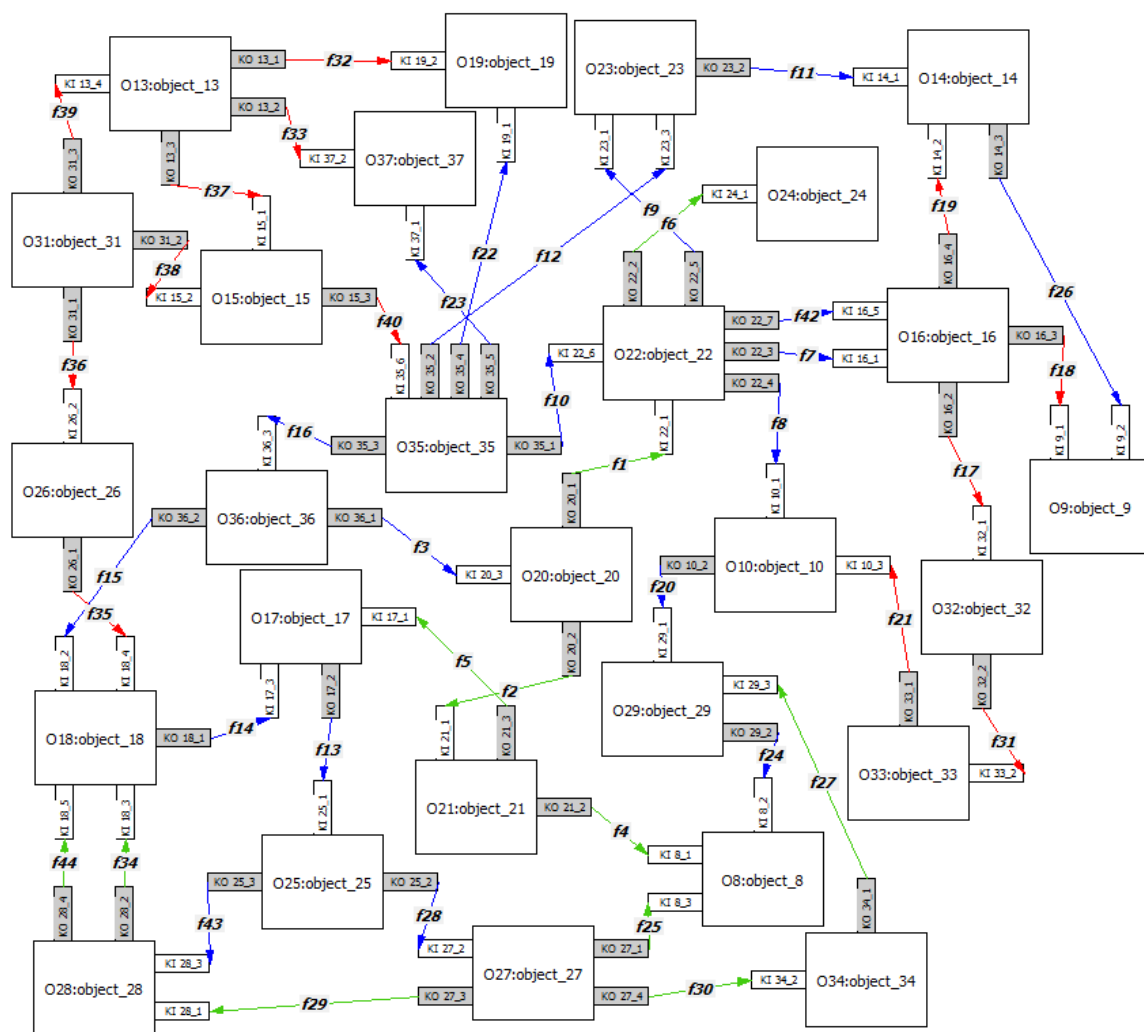
Nosaukums	Vērtība	Plūsma	1-enerģija; 2-informācija; 3-matērija	Ieejas	Izejas
KO 20_1	KI 22_1	f1	3	f3	(f1,f2)
KO 20_2	KI 21_1	f2	3	f3	(f1,f2)
KI 20_3	KO 36_1	f3	2	f3	(f1,f2)

3.17. att. Objektu un saistību saraksti transformācijai uz MSM

- 4) Pārbauda saistības (kontakts, plūsmas un loģiskos operatorus) pirmajam objektam objektu sarakstā, kuram nav atzīme „+” kolonā „Apskatīts”, bet ir atzīme „+” kolonā „Detalizēt”. Secīgi apskata klases freimu, kurā atspoguļots objekts, un kontaktu slotus (skatīt 3.16. att. (b) gadījumu):
- ja 2. solī eksperts ir izvēlējies veidot MSM visai sistēmai, tad pārbauda vai saistītais objekts (ko nosaka, apskatot kontakta slotu vērtību) ir pievienots objektu sarakstā. Tad veic pārbaudi vai objekts nav priekštecis vai pēctecis, kādam no objektiem, kas jau ir sarakstā. Ja nav, tad to pievieno objektu saraksta beigās un ievieto atzīmi „+” kolonā „Detalizēt”. Ja izvēlēts atspoguļot MSM apakšsistēmai, tad objektam pieliek atzīmi „-” kolonā „Detalizēt”. Ja veido modeli apakšsistēmai, tad detalizē tikai tos objektus, kas ievietoti objektu sarakstā, izpildot transformācijas algoritma 3. soli.
 - informāciju par kontaktu slotu ievieto saistību sarakstā (skatīt 3.17. att. (b) gadījumu). Pirmā kolonā norāda kontakta slotu nosaukumu un otrā kolonā slotu vērtību;
 - apskata ar kontaktu slotu saistīto kontaktu freimu un iegūst informāciju par plūsmas nosaukumu un vērtību, ko ievieto saistību saraksta trešajā un ceturtajā kolonā. Iegūst informāciju par loģiskiem operatoriem un ievieto saistību saraksta kolonās „Ieejas” un „Izejas”. Ja nav apskatīti visi freima kontakta sloti, pāriet pie nākamā slotu un atkārto soļus, sākot no 4.a. Ja ir apskatīti kontakta sloti, pāriet pie nākamā soļa izpildes;

d. apskatītajam objektam pievieno atzīmi „+” objektu saraksta kolonā „Apskatīts”. Kamēr objektu sarakstā ir objekti, kuriem nav atzīme „+” 2. kolonā, atkārto 4. soļa izpildi. Ja visiem objektiem ir atzīmes, tad pāriet pie piektā soļa izpildes.

* Izpildot pirmos 4 soļus, ir iegūts objektu saraksts, kurā ir visi saistītie objekti izvēlētajā dekompozīcijas līmenī, kas tieši vai netieši ir saistīti ar pirmo izvēlēto objektu O20. Izmantojot iegūto informāciju, sistēma I4S automātiski konstruē izvēlēto morfoloģiskās struktūras modeli (skatīt 3.18. att.).



3.18. att. MSM ar kontaktiem atspoguļojums sistēmā I4S

- 5) No objektu saraksta izvēlas objektu, kas vēl nav atspoguļots modelī un to vizuāli attēlo, ievērojot MSM lietotos apzīmējumus;
- 6) Pārbauda saistību sarakstu attiecībā uz atspoguļoto objektu:
 - a. nosaka ieejošo un izejošo kontaktu skaitu;

- b. ja izvēlēts struktūras modelis ar kontaktiem, tad vizuāli attēlo kontaktus. Uzzīmētajiem kontaktiem pievieno kontaktu nosaukumus. Kamēr nav attēloti visi objekti objektu sarakstā, atkārti 5. soļa izpildi.
- 7) Savieno objektus – vizuāli attēlojot plūsmas (bultiņa, plūsmas nosaukums un bultiņas krāsa, atbilstoši plūsmas tipam).
- a. pārbauda saistību sarakstu, ievērojot pirmo kolonu (nosaukums). Ja ir izejas kontakts, tad savieno (uzzīmē bultiņu) ar saistīto kontaktu. Ja ir ieejas kontakts, tad nesavieno. Uz bultiņas attēlo plūsmas nosaukumu;
 - b. loģiskās saistības starp plūsmām, kas apskatītas 3.17. att. (b) attēla tabulas kolonās „Ieejas” un „Izejas”, sistēma I4S izvada atsevišķā formā produkciju likumu formā. Kamēr nav apskatīti visi kontakti, atkārti 7. soļa izpildi.

Transformācijā no freimu kopas uz funkcionālās struktūras modeli funkciju telpā (FSM FT), izmanto freimu sistēmas hierarhiju, klases un kontaktu freimus. Pirmos četrus soļus veic līdzīgi kā transformācijā no freimu kopas uz MSM, lai iegūtu objektu un saistību sarakstus. Objektu saraksts ir tieši tāds pats, bet saistību sarakstā norāda funkciju nosaukumus (skatīt 3.19. att.), ko iegūst no kontaktu freima, nevis īstenoto plūsmu nosaukumus.

Nosaukums	Vērtība	Funkcijas	ieejas	Izejas
KO 20_1	KI 22_1	F1	F3	(F1,F2)
KO 20_2	KI 21_1	F2	F3	(F1,F2)
KI 20_3	KO 36_1	F3	F3	(F1,F2)

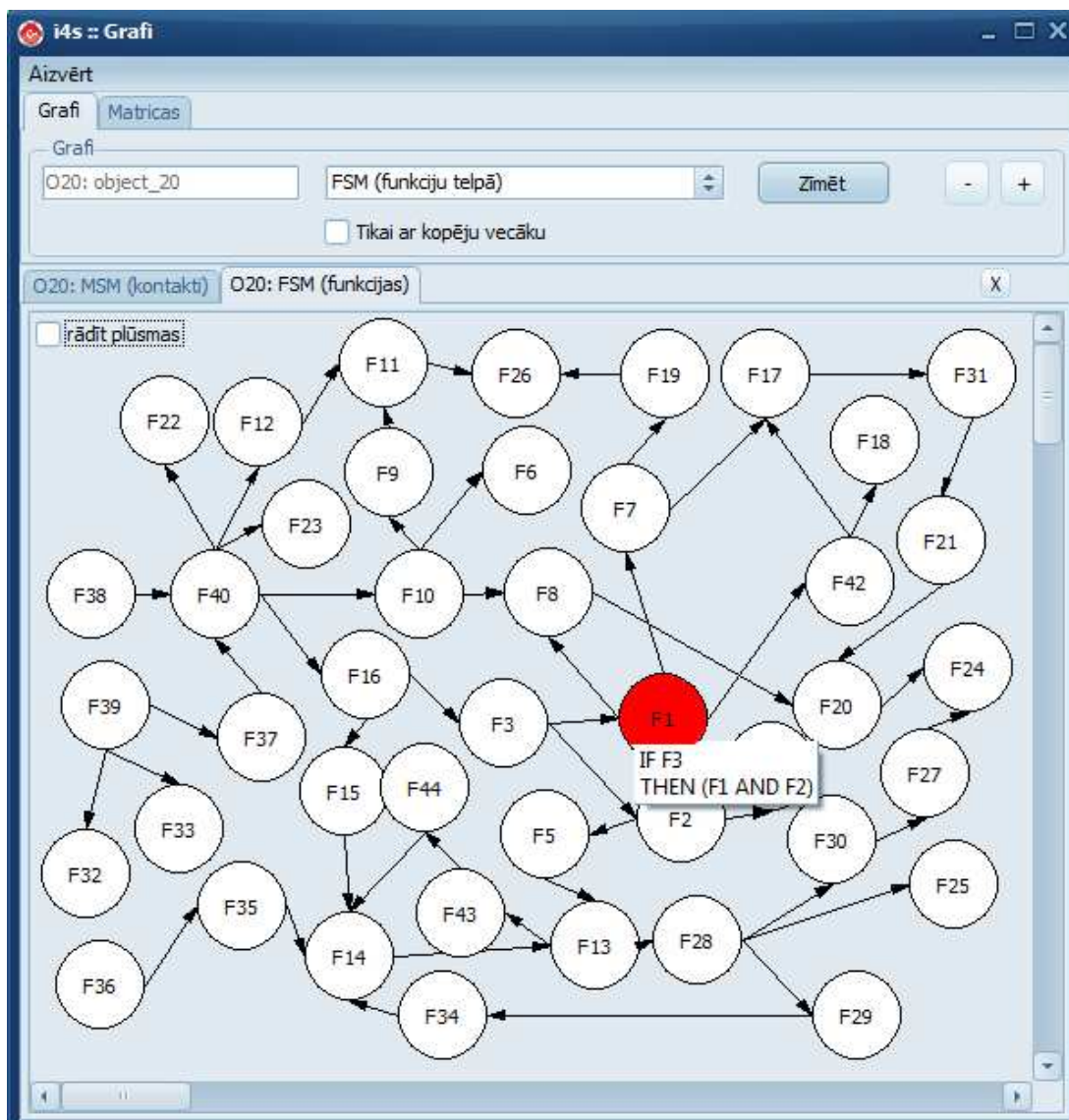
3.19. att. Saistību saraksta transformācijai uz FSM FT piemērs

Lai izveidotu FSM FT, tālāk transformācijas algoritmā lieto saistību sarakstu un veic šādus soļus:

- 1) No saistību saraksta izvēlas pirmo kontaktu, kas nav apskatīts. Ja funkcija vēl nav attēlota modelī, tad to vizualizē kā virsotni un norāda tās nosaukumu. Soli atkārti līdz ir atspoguļotas visas funkcijas.
- 2) Pārbauda saistību sarakstu. Ja apskatīts izejas kontakts, tad, ievērojot vērtības kolonās „Funkcijas” un „Ieejas”, savieno virsotnes. Ja kolonu vērtības ir vienādas, bultiņu nevelk. Ja nav vienādas, tad bultiņu velk no virsotnes, kas atbilst kolonā „Ieejas” apskatītajam funkcijas nosaukumam, uz virsotni, kas atbilst kolonā „Funkcijas” norādītajam funkcijas nosaukumam. Kolonā „Ieejas” var būt vairāki

funkciju nosaukumi un šādā gadījumā savieno visus uzskaitītos. Otro soli atkārtoti līdz ir apskatīti visi izejas kontakti saistību sarakstā.

Veicot transformācijas algoritma soļus, sistēma I4S izveido FSM FT (skatīt 3.20. att.). I4S vizualizē ne tikai eksperta izvēlēto modeli, bet arī, pietuvinot kursoru noteiktai funkcijai, norāda saistības starp funkcijām produkciju likumu formā (skatīt 3.20. att. virsotni *F1*).



3.20. att. FSM FT atspoguļojums I4S

Arī transformācijā no freimu kopas uz funkcionālās struktūras modeli uzvedības telpā (FSM UT), lieto freimu sistēmas hierarhiju, klases un kontaktu freimus. Tas norāda uz struktūras modeļu savstarpējo saistību un freimu kopas iespējām, vienuviet atspoguļot zināšanas par dažādiem izpētes sistēmas aspektiem (morfoloģiskiem, funkcionāliem). Transformācijā pirmajos soļos: (1) izvēlas dekompozīcijas līmeni, (2) nosaka, kuru FSM UT

konstruēs, norādot uzvedību, vai atsevišķus uzvedības stāvokļus, (3) iegūst objektu sarakstu (skatīt 3.17. att. (a)) un (4) veido saistību sarakstu (skatīt 3.21. att.):

4. Pārbauda saistības pirmajam objektam objektu sarakstā, kuram nav atzīme „+” kolonā „Apskatīts”, bet ir atzīme „+” kolonā „Detalizēt”:

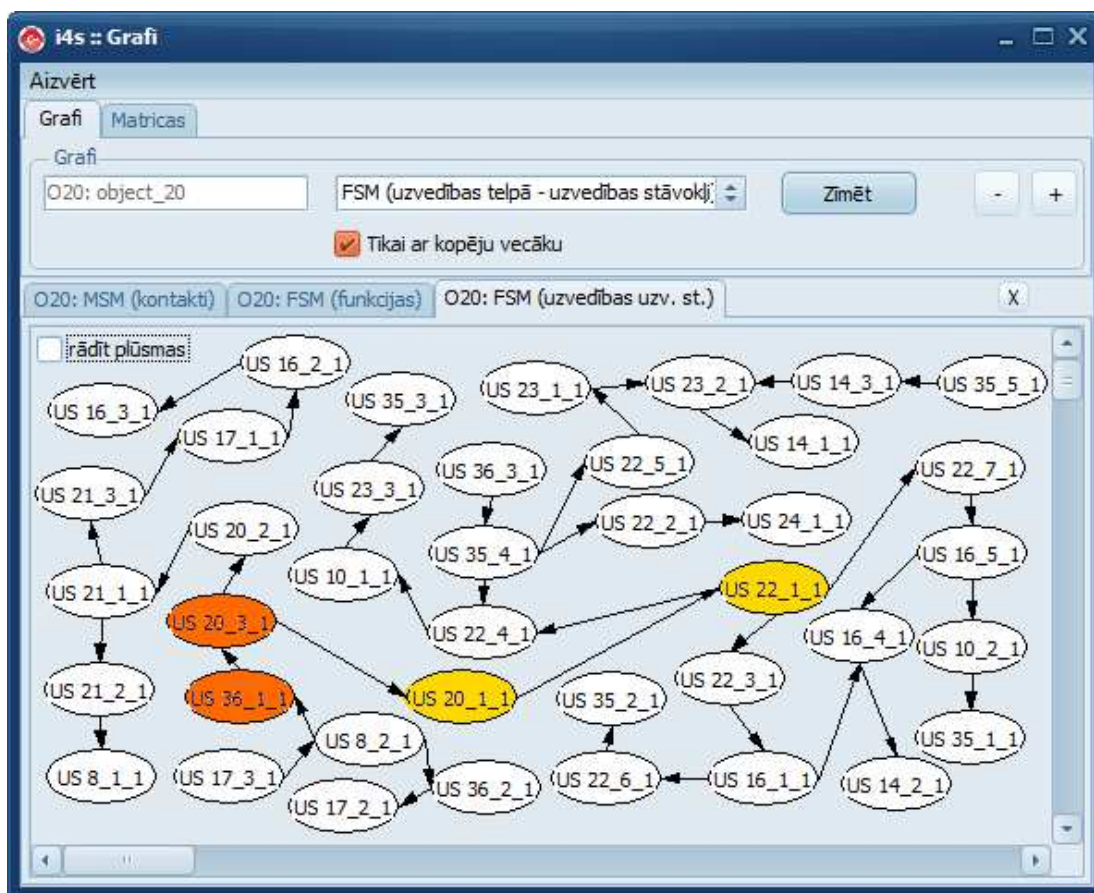
 - a) secīgi apskata klases freimu, kurā atspoguļots objekts, un kontaktu slotus;
 - b) pārbauda vai saistītais objekts ir pievienots objektu sarakstā. Ja nav, tad to pievieno objektu saraksta beigās (līdzīgi kā transformācijā uz MSM);
 - c) kontaktu slota nosaukumu attēlo saistību saraksta pirmajā kolonā.
 - d) apskata ar kontaktu slotu saistīto kontaktu freimu un saistību sarakstā ievieto informāciju, kas atspoguļota uzvedības slotos. Otrajā kolonā norāda uzvedības slota nosaukumu, bet trešajā kolonā slota vērtību;
 - e) kontaktu freima plūsmas slotos norādīto plūsmas nosaukumu un saistības starp ieejas un izejas plūsmām ievieto saistību saraksta 4., 5. un 6. kolonā. Kamēr nav apskatīti visi izvēlēta objekta kontakti, atkārti soļus, sākot no 4.a
 - f) apskatītam objektam pievieno atzīmi „+” objektu saraksta kolonā „Apskatīts”. Kamēr objektu sarakstā ir objekti, kuriem nav atzīme „+” 2. kolonā, atkārti 4. soļa izpildi. Ja visiem objektiem ir atzīmes, tad pāriet pie piektā soļa izpildes.

Kontaktu slots	Nosaukums	Vērtība	Plūsma	Ieejas	Izejas
KO 20_1	US 20_1_1	US 22_1_1	f1	f3	(f1,f2)
KO 20_2	US 20_2_1	US 21_1_1	f2	f3	(f1,f2)
KI 20_3	US 20_3_1	US 36_1_1	f3	f3	(f1,f2)

3.21. att. Saistību saraksta transformācijai uz FSM UT piemērs

5. Saistību sarakstā apskata izejas kontaktiem atbilstošos uzvedības stāvokļus. Soli a vai b veic līdz apskatīti visi izejas kontakti:
 - a) ja izvēlēts atspoguļot uzvedību, tad secīgi attēlo virsotnes – nosaukumā norādot divus uzvedības stāvokļus (otrā un trešā kolona);
 - b) ja izvēlēts vizualizēt uzvedības stāvokļus, tad atspoguļo divas virsotnes, katrā virsotnē norādot vienu uzvedības stāvokli. Vienai virsotnei atbilst otrajā kolonā norādītais uzvedības stāvoklis, otrai virsotnei atbilst trešajā kolonā norādītais uzvedības stāvoklis.
6. Sasaista virsotnes (skatīt 3.22. att.):

- a) ja virsotnēs norādīta uzvedība, tad novelk bultiņu, ņemot vērā 4. un 5. kolonu – plūsmas, kas ir saistītas ar uzvedības stāvokļiem. Bultiņu velk no virsotnes, kurai atbilst plūsma, kas apskatīta kolonā „Ieejas”, uz virsotni, kurai atbilst plūsma, kas apskatīta kolonā „Plūsma”.
- b) ja virsotnēs norādīti uzvedības stāvokļi, tad vienu bultiņu novelk no virsotnes, kuras nosaukums atbilst vērtībai, kas ir apskatīta otrajā kolonā (piemēram, US 20_1_1), uz virsotni, kuras nosaukums atbilst vērtībai, kas apskatīta trešajā kolonā (piemēram, US 22_1_1). Otru bultiņu velk līdzīgi kā 6.a. solī, bet savieno virsotnes, kas ir apskatītas 2. un 3. kolonā, ievērojot saistītās plūsmas un kontakta tipu. Piemēram, novelk bultiņu no virsotnes *US 36_1_1* uz virsotni *US 20_3_1*.



3.22. att. FSM UT atspoguļojums I4S

Transformācijas rezultātā sistēma I4S izveido FSM UT, kurā apskatīti visi uzvedības stāvokļi izpētes sistēmas atsevišķai komponentei, jo ir norādīta izvēle apskatīt sistēmas objektus „Tikai ar kopēju vecāku” (skatīt 3.22. att.).

Transformācijā no freimu kopas uz funkcionālās struktūras modeli parametru telpā (FSM PT), izmanto freimu sistēmas hierarhiju, klases, kontaktu un uzvedības freimus. Transformācijas soļus uz FSM PT, norādot parametru kopas, darbā neapskata, jo dotais modelis ir izomorfs FSM UT (norādot uzvedības stāvokļus), un saistības starp virsotnēm modelī nosaka identiski, kā iepriekš apskatītajā transformācijas algoritmā uz FSM UT.

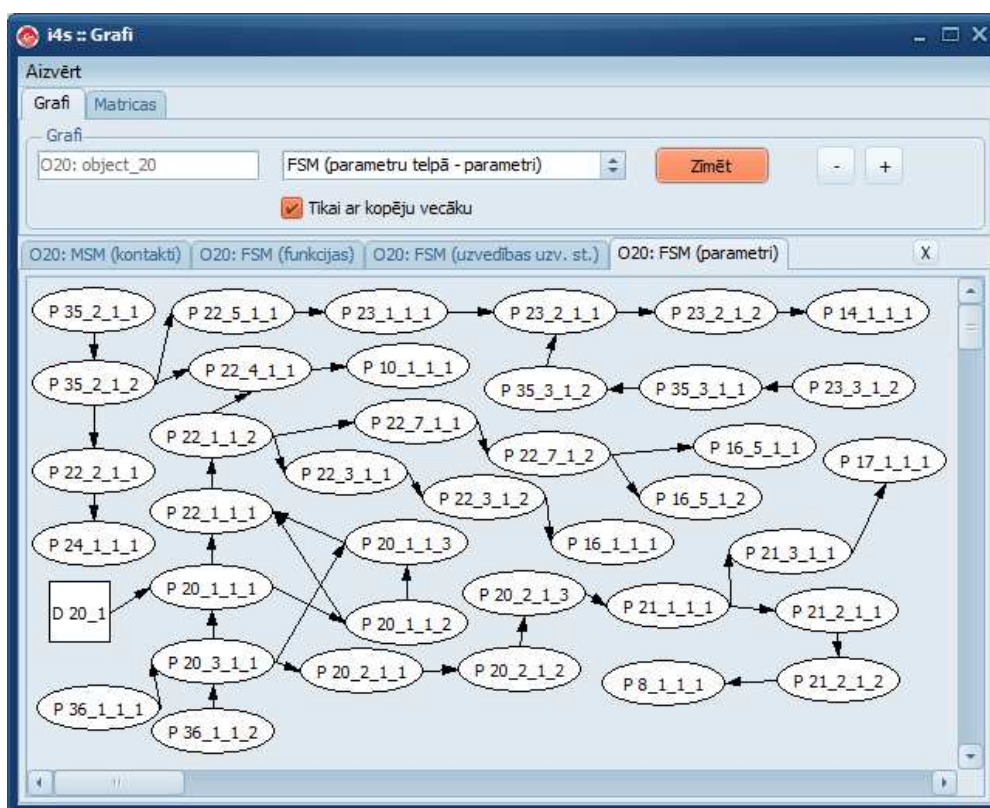
Transformācijā uz parametru telpu: (1) izvēlas dekompozīcijas līmeni, (2) nosaka, ka FSM PT konstruēs, norādot parametrus, (3) iegūst objektu sarakstu (skatīt 3.17. att. (a)) un (4) veido saistību sarakstu (skatīt 3.23. att.):

4. Pārbauda saistības pirmajam objektam objektu sarakstā, kuram nav atzīme „+” kolonā „Apskatīts”, bet ir atzīme „+” kolonā „Detalizēt”:
 - a) secīgi apskata klases freimu, kurā atspoguļots objekts, un kontaktu slotus;
 - b) pārbauda vai saistītais objekts ir pievienots objektu sarakstā. Ja nav, tad to pievieno objektu saraksta beigās (līdzīgi kā transformācijā uz MSM);
 - c) apskata ar kontaktu slotu saistīto kontaktu freimu un tad atbilstoši kontaktu freimā norādītajam uzvedības slotam piesaistīto uzvedību freimu. Saistību tabulā ievieto saišu slotos attēloto informāciju:
 - i. ja slotā nosaukumā un vērtībā ir norādīts tikai parametra nosaukums, tad to ievieto attiecīgi saistību saraksta pirmajā un otrajā kolonā;
 - ii. ja slotā nosaukumā ir loģiskais operators, tad atbilstoši slotā vērtībai, to ievieto saistību sarakstā pie tiem parametriem, kas ir saistīti ar loģisko operatoru. Soli 4.h. atkārto līdz ir apskatīti visi kontaktu sloti.
 - e. apskatītajam objektam pievieno atzīmi „+” objektu saraksta kolonā „Apskatīts”. Kamēr objektu sarakstā ir objekti, kuriem nav atzīme „+” 2. kolonā, atkārto 4. soļa izpildi. Ja visiem objektiem ir atzīmes, tad pāriet pie piektā soļa izpildes.
5. Pārbaudot saistību sarakstu, attēlo virsotnes (apskata saistību saraksta pirmo kolonu) – virsotnes nosaukumā norādot parametra vai defekta identifikatoru. Piekto soli atkārto līdz atspoguļotas visas virsotnes.
6. Sasaista virsotnes, izmantojot saistību saraksta pirmo un otro kolonu. Bultiņu velk no virsotnes, kuras nosaukums atbilst vērtībai, kas ir norādīta pirmajā kolonā, uz virsotni, kuras nosaukums atbilst vērtībai, kas ir norādīta otrajā kolonā. Soli atkārto līdz attēlotas visas saistības.

Nosaukums	Vērtība	Loģiskais operators	Saistības
P 20_1_1_1	P 20_1_1_2	AND	P 20_1_1_1, D 20_1
P 20_1_1_2	P 20_1_1_3		
P 20_1_1_3	P 22_1_1_1		
D 20_1	P 20_1_1_2	AND	P 20_1_1_1, D 20_1
P 20_1_1_3	US 36_1_1		

3.23. att. Saistību saraksta transformācijai uz FSM PT piemērs

Veicot aprakstītos transformācijas soļus, sistēma I4S konstruē FSM PT (skatīt 3.24. att.)



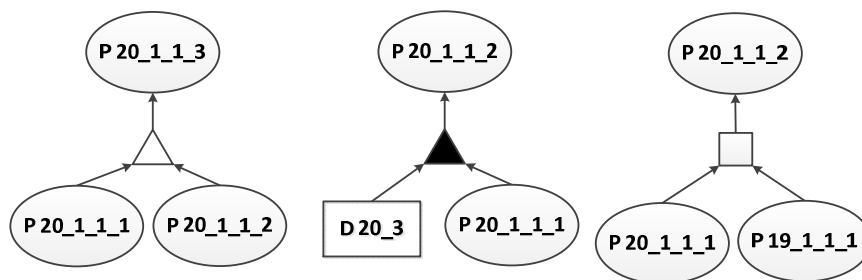
3.24. att. FSM PT atspoguļojums I4S

3.2.3. Transformācija no freimu kopas uz notikumu kociem

Sistēma I4S papildus transformācijām no freimu kopas uz struktūras modeļiem veic arī transformācijas no freimu kopas uz notikumu kociem. Katram notikumam atbilst viens parametrs vai defekts [GRU 1997b]. Transformāciju īsteno, sākotnēji iegūstot informāciju par parametriem un defektiem izvēlētajā detalizācijas līmenī, līdzīgi kā transformācijā no freimu kopas uz FSM PT. Divu notikumu kauzālā secība un tai atbilstošā loģika tiek interpretēta kā

kauzālas saites. Kausālu saišu virkne ļauj veidot notikuma koka struktūru. Notikumu kokā pamatelementi ir [GRU 1993, GRU 1997a, GRU 1997b]:

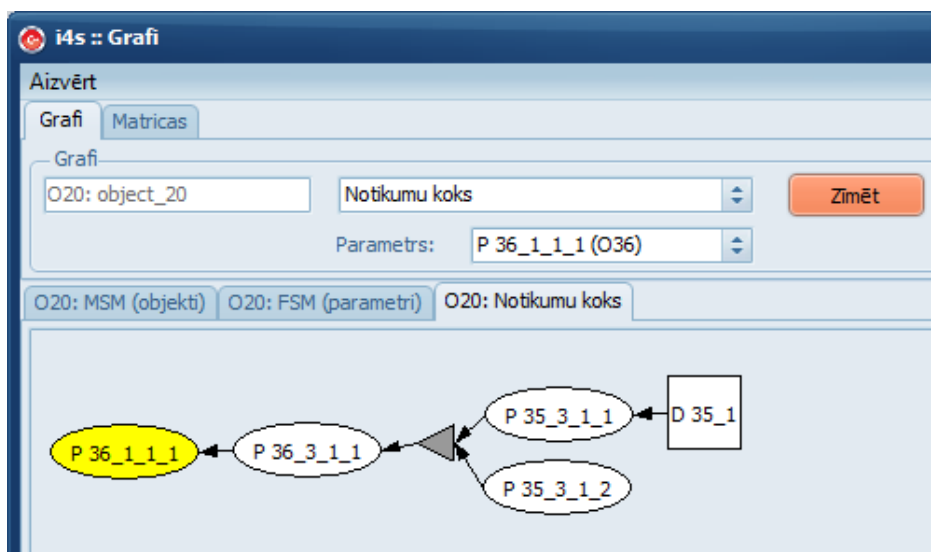
- notikumi, ko attēlo, norādot parametra vai defekta identifikatoru. Notikumu kokā izdala: galīgo notikumu, starpnotikumus un primāros notikumus.
 - Galīgais notikums ir saistīts ar novēroto (izmērīto) parametra vērtības izmaiņu. Galīgo notikumu izraisa izmaiņas, tas ir kļūdas kādā saistītā elementā. Gala notikumu atspoguļo kā notikumu koka sakni, kurai ir pakārtoti notikuma iestāšanās cēloņi.
 - Primārais notikums ir galīgā notikuma iestāšanās cēlonis, kas var būt defekts vai citi mērāmi parametri. Katrs primārais notikums rada vienu vai vairākus secīgus starpnotikumus.
- stāvokļu pārejas tiek interpretētas kā cēloņseku saites, ko attēlo orientētu loku veidā;
- loģiskie operatori, ko vizualizē, ņemot vērā to pieraksta veidu notikumu kokos (skatīt 3.25. att.).



3.25. att. Loģiskā operatoru atspoguļojums notikumu kokā

Notikumu kokos loģiskos operatorus atspoguļo tāpat kā MSM. Loģiskā operatora VAI gadījumā izmanto trīsstūri, izslēdzošā VAI gadījumā aizpildītu trīsstūri un loģiskā operatora UN gadījumā – kvadrātu. Loģiskos operatorus notikumu kokā lieto kā saistošos elementus starp parametriem un/vai defektiem [GRU 1999]. Notikumu koku veido, izvēloties virsotni (parametru), kurā ir konstatētas izmaiņas. Lai iegūtu parametru sarakstu noteiktā detalizācijas līmenī, īsteno transformācijas algoritmu soļus no freimu kopas uz FSM PT. Notikumu koku konstruē, izmantojot parametru un saistību sarakstus un procedūru „pārmeklēšana un atkāpšanās” (no angļu val. backtrack) [GRU 1997b, RUS 2010], ņemot vērā eksperta izvēlēto parametru. Procedūra īsteno meklēšanu grafā, kāpjoties atpakaļ no izvēlētajā virsotnes (galīgais notikums). Saistītās virsotnes tiek atspoguļotas kokā, ievērojot starp tām norādītos loģiskos operatorus. Procedūrā kāpjjas atpakaļ un pievieno virsotnes līdz ir sasniegti primārie

notikumi. Piemēram, izvēloties parametru *P 36_1_1_1* transformācijas rezultātā iegūst notikumu koku (skatīt 3.26. att.)



3.26. att. Notikumu koka atspoguļojums I4S

FSM PT attēlo visus notikumus un saistības starp tiem, izvēlētajā sistēmas detalizācijas līmenī. Notikumu koks ir grafisks attēlojums noteiktai cēloņseku virknei, kas ļauj identificēt izvēlētajā parametrā konstatēto izmaiņu cēloņus. Mainot sistēmas parametrus, var iegūt secinājumus par to, kāda ir to ietekme uz sistēmas funkcionalitāti. Notikumu kokiem ir būtiska loma diagnostikā, jo tie ļauj spriest par sistēmas uzvedības maiņas cēloņiem. Lietojot notikumu kokus var ģenerēt produkciju likumus, kas atbalsta dažādus spriešanas mehānismus (strukturālo, diagnostikas, kauzālo) [GRU 1997a, GRU 1997b, ZEL 2008b]. Produkcijas likumu iegūšana un saglabāšana zināšanu bāzē ir ārpus promocijas darba uzdevumiem.

3. nodaļas kopsavilkums un secinājumi

Trešās nodaļas mērķis ir aprakstīt darba autores izveidoto zināšanu atspoguļošanas shēmu – freimu kopu, ko lieto sistēmā I4S, lai iegūtu zināšanas no problēmsfēras eksperta, kā arī izveidotās sistēmas I4S uzbūves un darbības pamatprincipus. Trešās nodaļas galvenie rezultāti ir šādi:

- skaidroti jēdzieni „freims”, „freimu kopa” un „freimu sistēma” un sniegts to detalizēts apraksts;

- izanalizēti dažādi freimu atspoguļošanas gadījumi, aprakstīta darba ietvaros izveidotā freimu kopa un sniegti freimu kopā izmantoto freimu vizualizācijas piemēri;
- apskatīta realizētā sistēmas I4S arhitektūra un funkcionalitātes īstenošanai izveidotie moduļi;
- aprakstītas jaunizveidotās transformācijas no freimu kopas uz struktūras modeļiem, kas ļauj īstenot automatizētu struktūras modeļu izveidi sistēmā I4S, kā arī transformācija no freimu kopas un uz notikumu kokiem;

Galvenie trešās nodaļas secinājumi ir šādi:

- jaunizveidotā zināšanu izgūšanas un atspoguļošanas shēma freimu kopa ļauj vienuviet atspoguļot zināšanas par dažādiem izpētes sistēmas aspektiem (morfoloģiskiem, funkcionāliem), kā arī veidot objektu un funkciju hierarhijas;
- transformācijās ir būtiski ievērot ne tikai loģisko operatoru lietojumu, bet arī veikt papildus pārbaudes, lai, veidojot struktūras modeli, nepieļautu vienlaicīgi gan priekšteču, gan pēcteču attēlošanu (tas ir nepieciešams, lai automatizēti izveidotu pareizu struktūras modeli);
- izstrādātā intelektuālā sistēma I4S ļauj īstenot zināšanu izgūšanu, atspoguļošanu un saglabāšanu;
- sistēma I4S iekļauj 3 pamatkomponentes, kurās ir implementēta freimu kopa, un 6 moduļus, kas ļauj īstenot struktūrmodelēšanas mērķus. Lietojot iegūtās zināšanas un darba ietvaros izveidotās transformācijas, I4S nodrošina struktūras modeļu un notikumu koku automatizētu konstruēšanu.

4. PRAKTISKA PIEMĒRA REALIZĀCIJA SISTĒMĀ I4S

Lai praktiski pārbaudītu promocijas darba ietvaros izstrādāto sistēmu I4S un tās funkcionalitāti, tajā ir īstenots sarežģītas tehniskas sistēmas – robota AGR8 atspoguļojums. AGR8 ir mobila robota platforma, kas ir izstrādāta Rīgas Tehniskajā universitātē mehānikas un mākslīgā intelekta pētniecības nolūkos. Robota iespējamās pielietojuma sfēras, piemēram, ir izlūkošana un nelīdzenas zonas pārmeklēšana. Izstrādātā platforma sastāv no korpusa un astoņiem piedziņas riteņiem, kas ir sadalīti pa pāriem. Riteņi ir savienoti ar riteņu pāru balstiem. Katrs riteņu pāris rotē ap savu asi neatkarīgi no citiem riteņu pāriem, un katru no tiem kontrolē viens mikroprocesors. Kad platforma pārvietojas, tās korpuss tiek līdzsvarots ar atsperu un amortizatoru palīdzību. Lai palielinātu platformas mobilitāti, tiek darbināts katrs no astoņiem riteņiem, tādējādi nodrošinot maksimālu elastību. Motori ir aprīkoti ar ierīci, kas ļauj noteikt faktisko motora ātrumu. Šos datus izmanto, lai kontrolētu platformas kustības laikā, kad tā pārvar šķēršļus. Lai noteiktu faktisko riteņu pāru pozīciju attiecībā pret platformas korpusu, ir integrēti augstas precizitātes sensori [NIK 2010]. Robots sastāv no daudzām heterogēnām daļām, kas ir savstarpēji saistītas ar dažāda veida saitēm (informācijas, matērijas un enerģijas). Robota AGR8 struktūrmodelēšana ir īstenota ar sistēmu I4S, kurā implementēta zināšanu atspoguļošanas shēma, kas ļauj iegūt modeļu izveidei nepieciešamās zināšanas, un algoritmi, kas ļauj atspoguļotās zināšanas lietot automatizētu struktūras modeļu izveidei. Atbilstoši SM pieejas koncepcijai sistēmā I4S ir implementēti arī risinājumi, kas atbalsta jaunu zināšanu izgūšanu par izpētes sistēmu un tās analīzi. Lai īstenotu izpētes sistēmas struktūrmodelēšanu, sistēma I4S veic šādus uzdevumus:

- atbalsta zināšanu izgūšanu, atspoguļošanu un saglabāšanu tādā veidā, lai zināšanas var koplietot, atkārtoti izmantot un lietot SM mērķu sasniegšanai;
- iegūtās zināšanas sistēma I4S izmanto, lai, lietojot transformācijas algoritmus, automatizēti konstruētu SM pieejā aprakstītos struktūras modeļus un notikumu kokus dažādos sistēmas detalizācijas līmeņos;
- pamatojoties uz atspoguļoto morfoloģiskās struktūras modeli, sistēma I4S izveido matricas un īsteno sistēmas topoloģisko un kvalitatīvo sistēmas struktūras analīzi, kas nepieciešama, lai veiktu sistēmas izpēti.

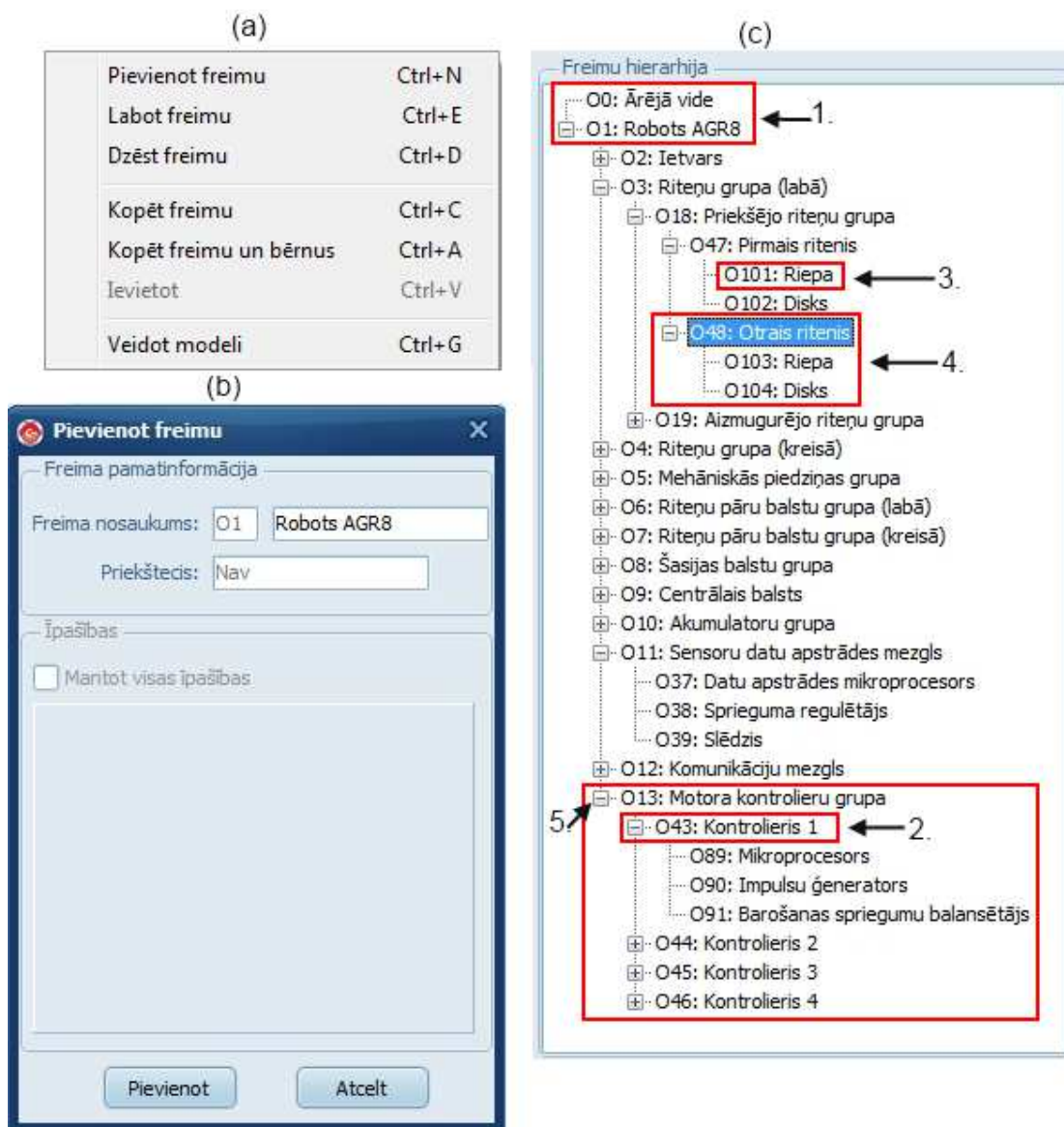
4.1. Zināšanu iegūšana un atspoguļošana sistēmā I4S

Strādājot ar sistēmu I4S, eksperta galvenais uzdevums ir izveidot izpētes sistēmas formālu atspoguļojumu, norādot struktūrmodelēšanai būtiskos aspektus: kādu sistēmu apskata, kādi objekti ir sistēmā un kādas ir starp tiem pastāvošās saistības, kādas ir sistēmas un atsevišķu objektu īpašības un uzvedība. Sistēma I4S var iegūt un atspoguļot eksperta zināšanas attiecībā uz izvēlēto izpētes sistēmu dažādos veidos:

- aprakstot sistēmas objektus, saites ar citiem objektiem, īpašības (piemēram, krāsa svars, utt.), alternatīvas, ko var izmantot apskatītā objekta vietā;
- detalizējot saistības, plūsmas nosaukumu un tipu, funkcijas nosaukumu un realizēto uzvedību, norādot iespējamās plūsmu kombinācijas, kā arī plūsmas īpašības (piemēram, matērijas veids: eļļa);
- specificējot parametrus katram uzvedības stāvoklim un to savstarpējās saites;
- izveidojot un attēlojot sistēmas daļu, funkciju un īpašību hierarhijas;
- transformējot no eksperta iegūtās zināšanas struktūras modeļos.

4.1.1. Robota AGR8 daļu un to saistību atspoguļojums

Sistēmas apraksta izveidi uzsāk, pievienojot pirmo freimu (skatīt 4.1. att. (b)), kurā norāda sistēmas nosaukumu, kas promocijas darba ietvaros ir *robots AGR8*. Freimu pievieno, lietojot izvēlnes iespēju „Pievienot freimu” vai taustiņu „Ctrl+N” kombināciju (4.1. att. (a)). Kad pirmais freims ir pievienots, sistēma I4S automātiski pievieno vēl vienu freimu, ar nosaukumu *00: Ārējā vide* (4.1. att. 1. (c)). Tas tiek darīts ar nolūku, dot ekspertam iespēju aprakstīt izpētes sistēmas saistību ar tās ārējo vidi, līdz ar to arī plūsmas, kuras ietekmē sistēmas funkcijas un uzvedību vai kuras īsteno sistēma iedarbojoties uz ārējo vidi. Kad ir norādīts sistēmas nosaukums, tālāk secīgi apraksta sistēmā esošos objektus, freimu hierarhijā pievienojot jaunus freimus ar sistēmas daļu nosaukumiem. Darba autore ir sistēmā I4S atspoguļojusi zināšanas par robota AGR8 struktūru, norādot sistēmā esošo objektu nosaukumus (skatīt 4.1. att. (c)). Robots AGR8 sastāv no 13 pamatkomponentēm, bet kopumā I4S ir attēloti 148 objekti (objektu skaitu sistēmā ļauj identificēt pēdējā pievienotā objekta numurs).



4.1. att. Freima pievienošana un freimu hierarhija sistēmai *robots AGR8*

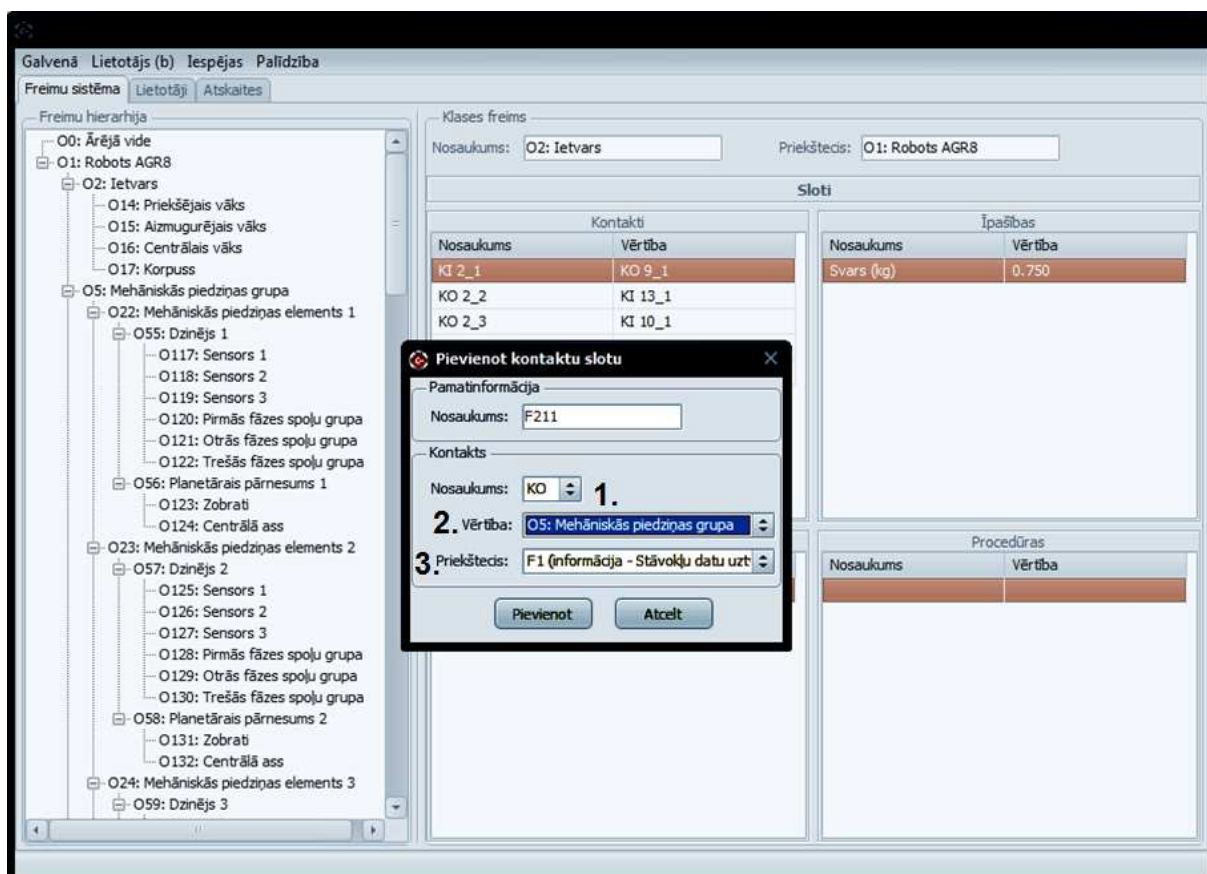
Freimu hierarhijā var aplūkot robota dekompozīciju, kas apstiprina I4S iespējas, atspoguļot zināšanas par dažādām sarežģītām sistēmas daļām:

- **komponentēm** – piemēram, *O43: Kontrolieris 1* (4.1. att. 2. (c)) ir sistēmas komponente, kas iekļauj sistēmas elementus: *O89: Mikroprocesors*, *O90: Impulsu ģenerators* un *O91: Barošanas spriegumu balansētājs*;
- **elementiem**, piemēram, *O101: Riepa* ir sistēmas elements (4.1. att. 3. (c));
- **heterogēnām daļām**. Piemēram, *O48: Otrais ritenis* (skatīt 4.1. att. 4. (c)) ir sistēmas komponente, kas iekļauj 2 dažādu elementus: *O103: Riepa* un *O104: Disks*;

- **homogēnām daļām.** Kā piemērs apskatīta *O13:Motora kontrolieru grupa* (skatīt 4.1. att. 5. (c)), kas iekļauj četras komponentes – kontrolierus, kuru uzbūve un darbības principi ir identiski.

Sistēma I4S ļauj izveidot saites tikai starp freimu hierarhijā aprakstītiem objektiem, tāpēc vispirms noteikti ir jābūt aprakstītām visām sistēmas daļām. Tālāk darba autore ir apskatījusi katru sistēmas daļu atsevišķi, kontaktu slotos norādot saistības starp objektiem (skatīt 4.2. att.):

- kontakta tipu (4.2. att. 1.), kas atbilstoši struktūrmodeļēšanas sintaksei var būt KO (izejas) vai KI (ieejas).
- objektu (4.2. att. 2.) ar ko ir saistīts apskatītais objekts. Saistību pievienošanu atvieglo tas, ka visi objekti ir uzskatāmi redzami freimu hierarhijā, un vērtību sarakstā, no kura izvēlas saistīto objektu, tie ir sakārtoti identiskā secībā kā freimu hierarhijā.



4.2. att. Ieejas izejas kontaktu attēlošana I4S

- funkcijas priekšteci (skatīt 4.2. att. 3.). Ja funkcija ir pakārtota kādai citai funkcijai sistēmā, tad tai norāda priekšteci (skatīt 4.4. att. 4.). Priekšteču sarakstā

tiek attēlotas apskatītā objekta priekšteča funkcijas un plūsmas tips. Priekšteci var mainīt jebkurā brīdī kontaktu slotos, kuros nosaukums atbilst izejas tipa kontaktam (KO). Izmantojot šo I4S iespēju tiek veikta sistēmas funkciju dekompozīcija (skatīt 4.4. att. 5.).

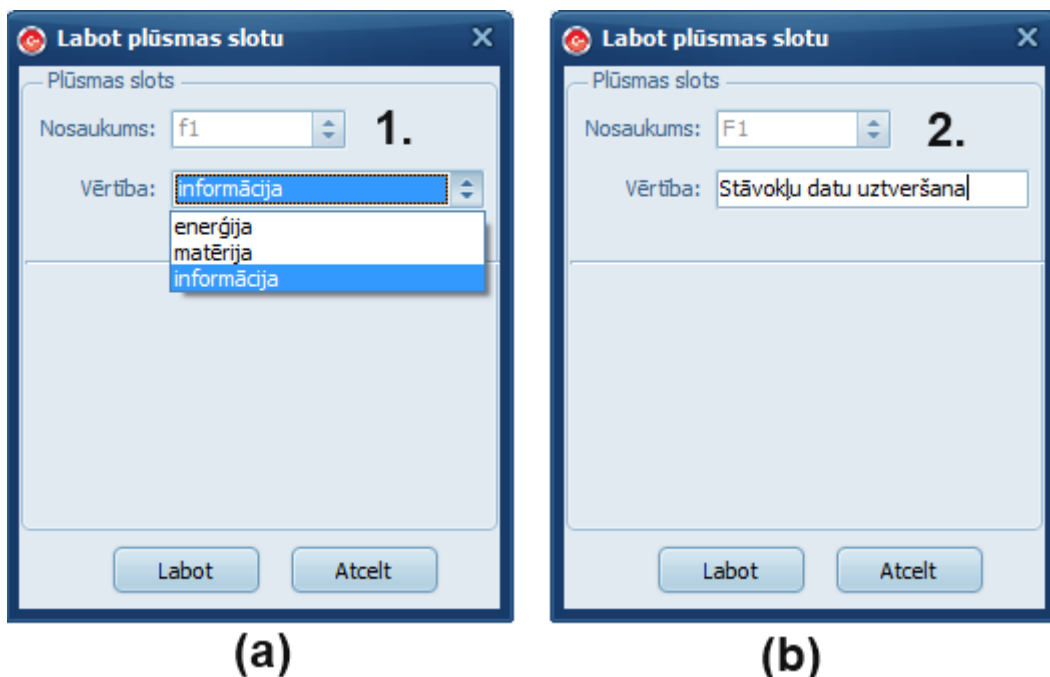
Nospiežot pogu „Pievienot”, I4S automātiski izveido kontaktu slotu saistītam objektam (skatīt 4.3. att.). Izveidotai saistībai sistēma automātiski piešķir plūsmu (4.4. att. 1.) un funkciju (4.4. att. 2.) identifikatorus, ko ievieto kontaktu freima plūsmu slotos.

Klases freims		Klases freims	
Nosaukums: O17: Korpuss		Nosaukums: O43: Kontrolieris 1	
Kontakti		Kontakti	
Nosaukums	Vērtība	Nosaukums	Vērtība
KO 17_1	KI 14_1	KI 43_1	KO 17_4
KO 17_2	KI 15_1	KI 43_2	KO 22_1
KO 17_3	KI 16_1	KO 43_3	KI 22_2
KO 17_4	KI 43_1	KI 43_4	KO 55_3
KO 17_5	KI 44_1	KO 43_5	KI 55_4
KO 17_6	KI 45_1		
KO 17_7	KI 46_1		
KO 17_8	KI 35_1		
KO 17_9	KI 36_1		

4.3. att. Ieejas un izejas kontaktu attēlošana I4S

4.4. att. Plūsmas, funkcijas un uzvedības stāvokļu identifikatori

Tālāk darba autore ir norādījusi plūsmas un funkcijas nosaukumus, pievienojot kontaktu freima plūsmu slotiem vērtības (skatīt 4.5. att. (a) un (b)). Plūsmas nosaukuma vērtība atbilst SM definētiem plūsmas tiptiem, kas var būt enerģija, matērija vai informācija. Funkcijas nosaukumu ievada brīvā formā, piemēram, „Stāvokļu datu uztveršana”. Plūsmas un funkciju identifikatorus eksperts nevar mainīt (skatīt 4.5. att. 1. (a) un 2. (b)).



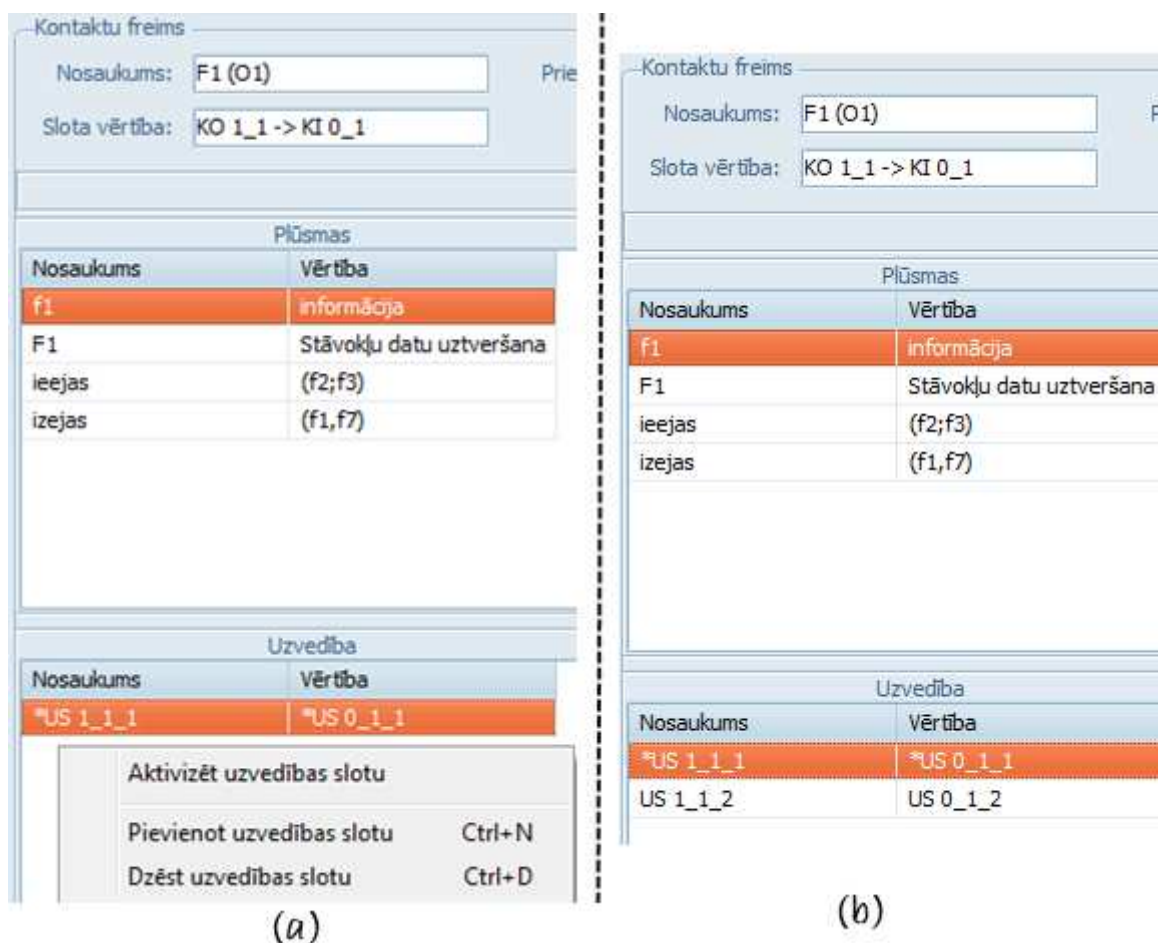
4.5. att. Plūsmas slotu labošana

Kad ir aprakstīti visi sistēmas objekti un starp tām pastāvošās saistības, sistēma I4S pēc eksperta pieprasījuma izveido MSM. Tomēr, lai konstruētu arī FSM, sistēmā I4S ir jānorāda loģiskie operatori un atbilstošās plūsmu kombinācijas katrā kontaktu freimā. Līdz ar to ir ieteicams sākotnēji atspoguļot visas modelēšanai nepieciešamās zināšanas attiecībā uz sistēmu robots AGR8 un tikai tad pāriet pie modeļu izveides.

4.1.2. Robota AGR8 uzvedības un parametru atspoguļojums

Sistēma I4S, izveidojot jaunu kontaktu slotu, automātiski izveido arī divus uzvedības stāvokļus (skatīt 4.4. att. 3.), kas atbilst ieejas un izejas kontaktiem, kā arī katram uzvedības stāvoklim atbilstošo parametru kopu. Tomēr eksperts jebkurā brīdī var pievienot papildus uzvedības stāvokļus, kā tas ir parādīts 4.6. att. (a) attēlā uzvedības slotos, nospiežot labo taustiņu un izvēloties iespēju „Pievienot uzvedības slotu”. Sistēma I4S pārbauda uzvedības

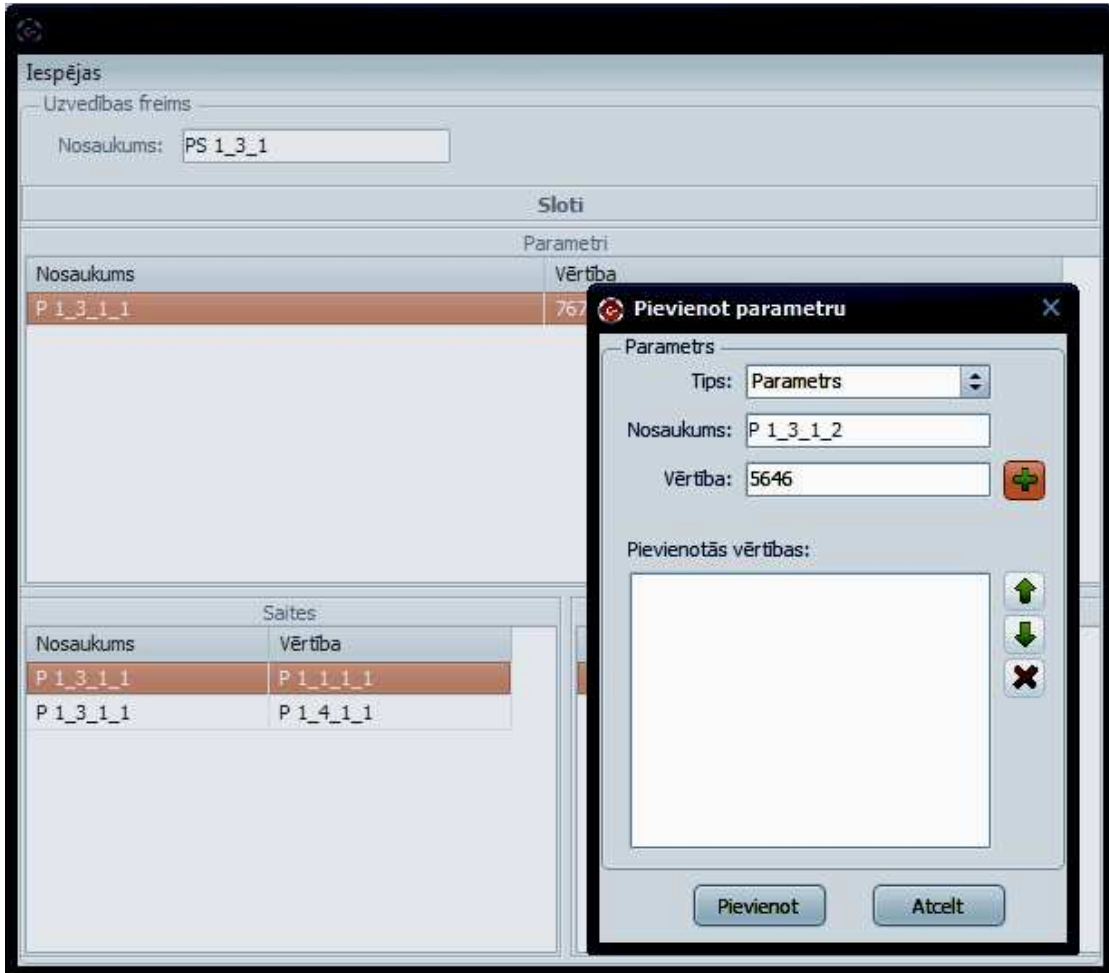
slotu skaitu noteiktā kontaktā un izveido uzvedības stāvokļu nosaukumus, ko pēc eksperta akcepta pievieno uzvedības slotiem (skatīt 4.6. att. (b)).



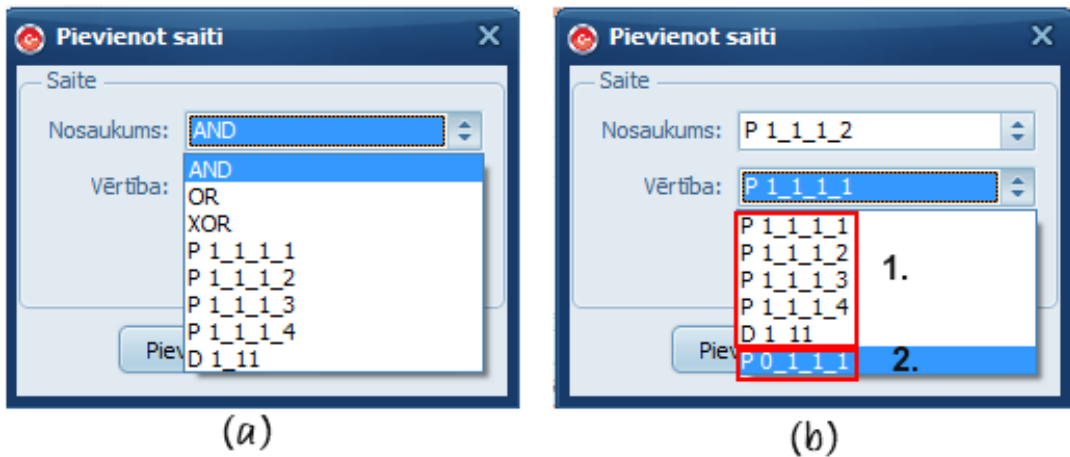
4.6. att. Uzvedības slotu atspoguļojums

Turpmāk tiek aprakstīti parametru kopā iekļautie parametri un defekti, ja tādi pastāv. Atspoguļošanu veic, pievienojot parametru slotus uzvedības freimā. Parametra un defekta identifikatorus sistēma I4S piešķir automātiski, bet vērtības ir jāievada ekspertam. Parametra un defekta vērtības var būt dažādas – viens skaitlis, noteikts diapazons, vairākas atsevišķas skaitliskas vērtības vai simbolu virkne (skatīt 4.7. att.).

Kad ir pievienoti parametri un/vai defekti, tad saišu slotos ir jānorāda saistības starp tiem. Pievienojot saiti, pie nosaukuma var izvēlēties kādu no iekšējiem parametriem vai defektu, kā arī loģiskos operatorus (skatīt 4.8. att. (a)). Pie vērtības norāda iekšējo (4.8. att. 1.(b)) vai ārējo (4.8. att. 2.(b)) parametru, vai defektu ar ko ir saistīts parametrs, kas attēlots pie nosaukuma (skatīt 4.8. att. (b)). Ja pie nosaukuma izvēlas kādu no loģiskiem operatoriem, tad pie vērtības ir jāievada divi saistītie parametri.



4.7. att. Parametru pievienošana I4S



4.8. att. Parametru saišu pievienošana

4.1.3. Papildus iespējas sistēmā I4S

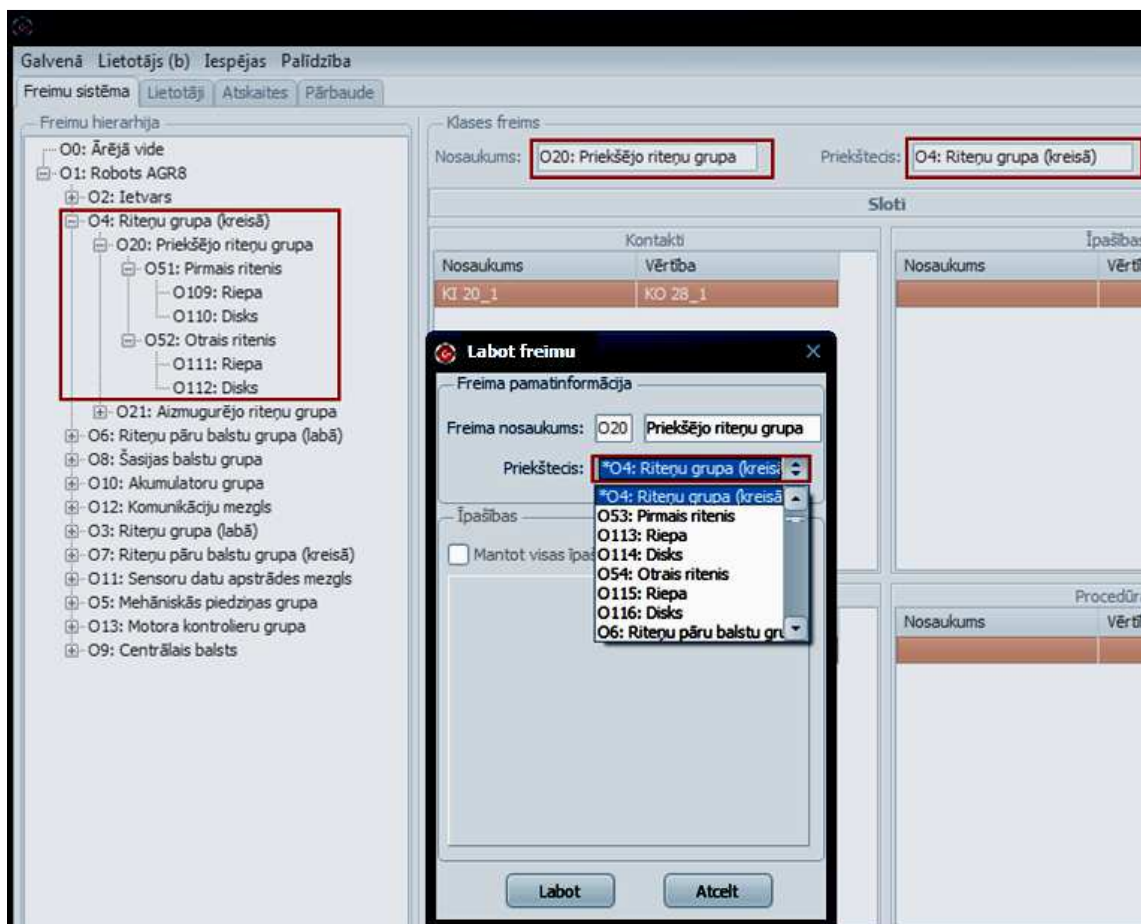
Sistēma I4S ļauj manipulēt ar atspoguļotajām zināšanām, par objektiem, saitēm, īpašībām un uzvedību. Ekspertam ir iespēja ne tikai pievienot objektu aprakstu, bet arī to mainīt un dzēst, kā arī īstenot reorganizēšanu sistēmas atspoguļojumā. Reorganizēšanas iespēja ir ieviesta, lai atdarinātu pašorganizēšanās procesu, kas ir būtisks sarežģītās sistēmās. Reorganizēt nozīmē to, ka I4S var pārvietot izvēlētu sistēmas daļu (ar visiem tās pēcteciem) uz citu vietu sistēmā, tas ir, mainīt daļas priekštecī. Mainot priekštecī, nemainās definētās saistības un tiek saglabāta objekta funkcionalitāte. Sistēmā ir mainīta struktūra, bet ne organizētība. Pārveidot sistēmas struktūras atspoguļojumu ir būtiski ne tikai, ja mainās reālā sistēma, bet arī gadījumos, kad eksperts ir pieļāvis kļūdas, veicot sistēmas aprakstu. Piemēram, ja divām sistēmas daļām, kas attiecībā viena pret otru sistēmā ir kā priekštecis un pēctecis, vienlaicīgi pastāv saite uz citu objektu sistēmā, tad sistēmas aprakstā varbūt ir kļūda. Iespējams, kāda no abām daļām reāli nepastāv un ir mākslīgi izveidota, kas iekļauta atspoguļojumā. Tomēr tas nenozīmē, ka sarežģītās sistēmās nepastāv minētās saistības, tāpēc I4S ļauj izveidot struktūras modeļus izvēlētam detalizācijas līmenim arī pie nosacījumiem, kad priekštecim un pēctecim ir saistība ar vienu un to pašu objektu sistēmā.

Veidojot robota AGR8 aprakstu, tika attēlotas dažas liekas komponentes, jo darba autore aprakstīja objektus, kas robotam reāli neeksistē. Piemēram, daļa *O4: riteņu grupa (kreisā)* principā nav nepieciešama (skatīt 4.9. att.) un pietiktu, ja būtu norādītas daļas: *O20: priekšējo riteņu grupa* un *O21: aizmugurējo riteņu grupa*, to nosaukumos pievienojot vārdu (*kreisā*). Tāpēc tika pārveidota izpētes sistēmas struktūra, pārvietojot objektus O20 un O21 zem objekta O1, veicot šādus soļus:

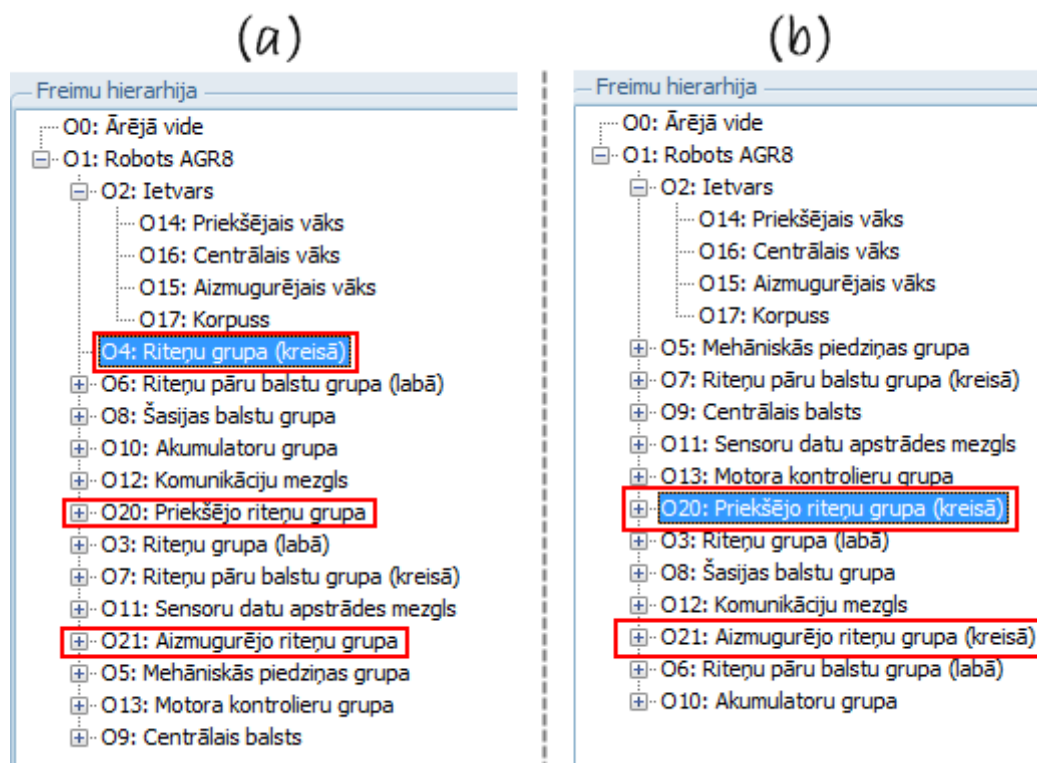
1. izvēlas objektu un nospiež izvēlnē iespēju „Labot freimu”;
2. I4S atvērtajā formā (skatīt 4.9. att.) nomaina priekštecī un nospiež pogu „Labot”.

I4S ar simbolu *, norāda daļas priekštecī pirms izmaiņām.

Pēc tam autore izlaboja objektu O20 un O21 nosaukumus, izmantojot pogu „Labot freimu”. Attēlā 4.10. att. ir apskatīta freimu hierarhija pēc priekšteča izmaiņas (O4 vairs nav pēctecī) un pirms O20, O21 nosaukumu labošanas (a gadījums) un pēc labošanas un objekta O4 dzēšanas (b gadījums).

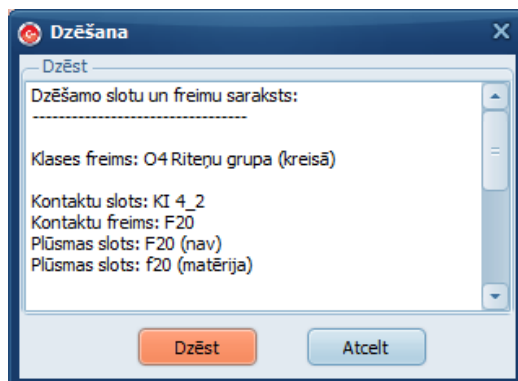


4.9. att. Objekta priekšteča labošana



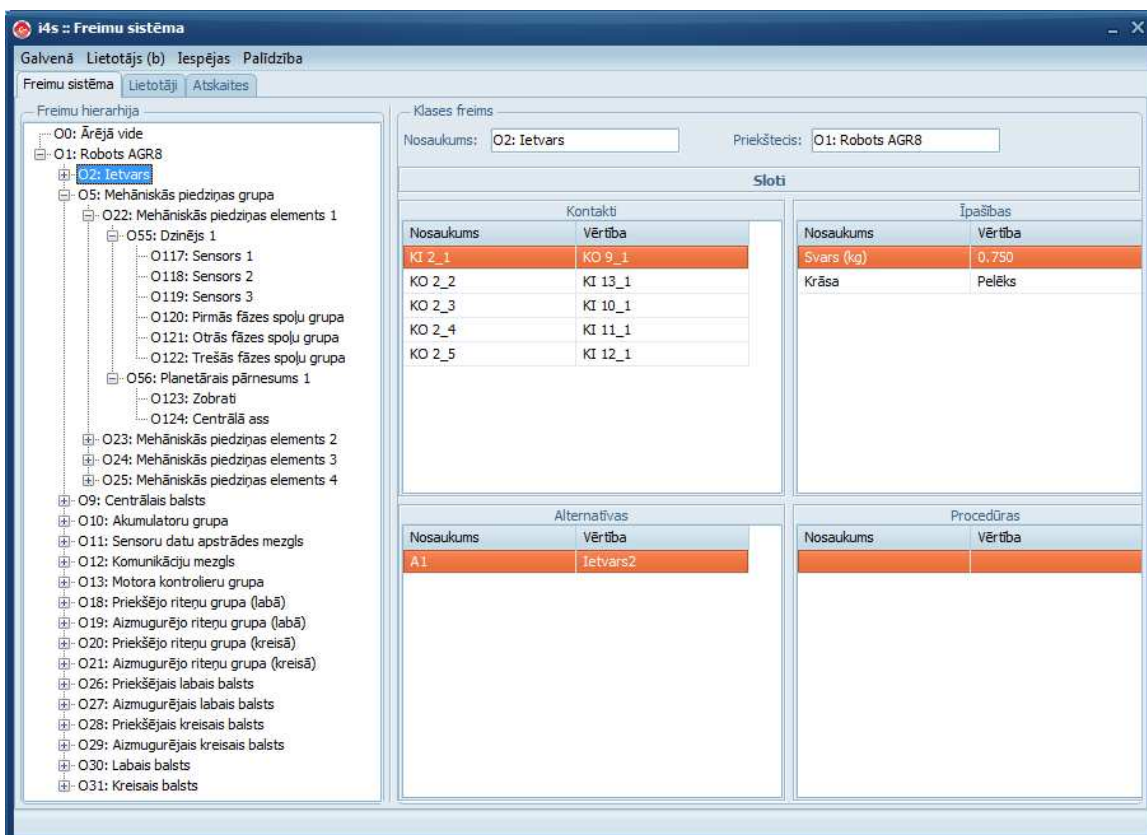
4.10. att. Objektu nosaukumu labošana

Pirms objekta dzēšanas I4S paziņo par tā dzēšanu, kā arī par saitēm un citām īpašībām, kas tiks dzēstas līdz ar objektu (skatīt 4.11. att.).



4.11. att. Sistēmas daļu dzēšana

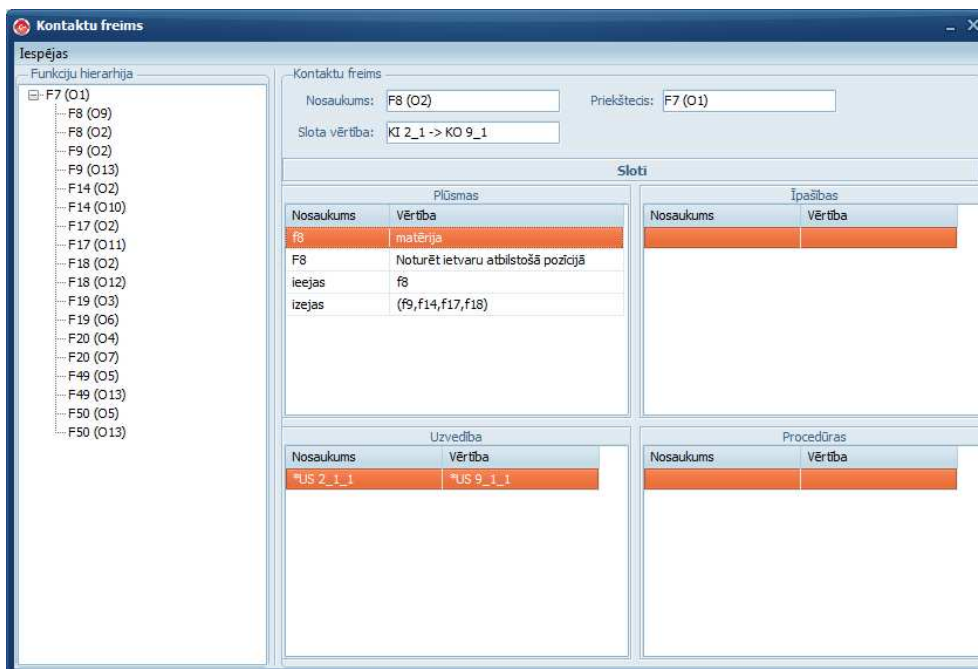
Pēc veiktajām izmaiņām (reorganizēšanas, objektu nosaukumu maiņas un lieko objektu dzēšanas), sistēmas Robots AGR8 aprakstā ir norādītas 17 komponentes: O2, O5, O9, utt. (skatīt 4.12. att.)



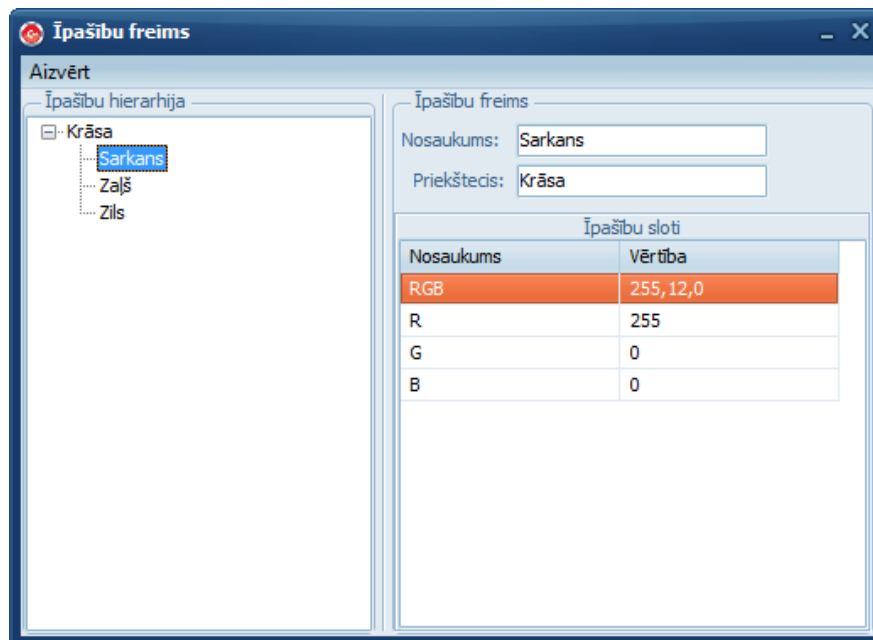
4.12. att. Robota AGR8 atspoguļojums freimu hierarhijā

Sistēmā I4S, izmantojot freimu hierarhijas iespējas, attēlo dažāda veida hierarhijas: sistēmas daļu (skatīt 4.1. att. (c)), funkciju (skatīt 4.13. att.), īpašību (skatīt 4.14. att.). Pēc

eksperta norādes, sistēma I4S konstruē daļu un funkciju hierarhijas arī vizualizēti, izveidojot atbilstošos modeļus (skatīt 4.2.1. nodaļu).



4.13. att. Kontaktu freima un funkciju hierarhijas atspoguļojums



4.14. att. Īpašību freima un īpašību hierarhijas atspoguļojums

Kad sistēmā I4S ir izveidots izpētes sistēmas apraksts freimu kopā, un ir atspoguļotas visas modelēšanai nepieciešamās sistēmas daļas, to saistības, uzvedība un parametri, kā arī norādīti loģiskie operatori, tad īsteno dažādu struktūras modeļu konstruēšanu, dažādos sistēmas detalizācijas līmeņos (skatīt 4.2.1. nodaļu).

4.2. Robota AGR8 struktūrmodelēšana un analīze ar I4S

Pētot sarežģītas tehniskas sistēmas nepilnīgas informācijas apstākļos, to modeļiem ir jāapmierina šāda prasība [GRU 1993, GRU 1997a]:

- modeli jāspēj izveidot pamatojoties tikai uz pieejamām zināšanām;
- modelim jāaptver visa sistēma kopumā, neskatoties uz elementu fizisko daudzveidību;
- modelim jābūt viegli koriģējamam sistēmas konstruktīvo izmaiņu gadījumā;
- modelim ‘‘jāstrādā’’ nepilnīgas informācijas apstākļos un jādod jaunas zināšanas par pētāmo sistēmu.

Īstenojot robota AGR8 struktūrmodelēšanu, ir iegūts apstiprinājums, ka struktūrmodelēšanas pieeja, implementēta intelektuālā datorsistēmā I4S, ļauj konstruēt struktūras modeļus, ievērojot iepriekš uzskaitītos kritērijus.

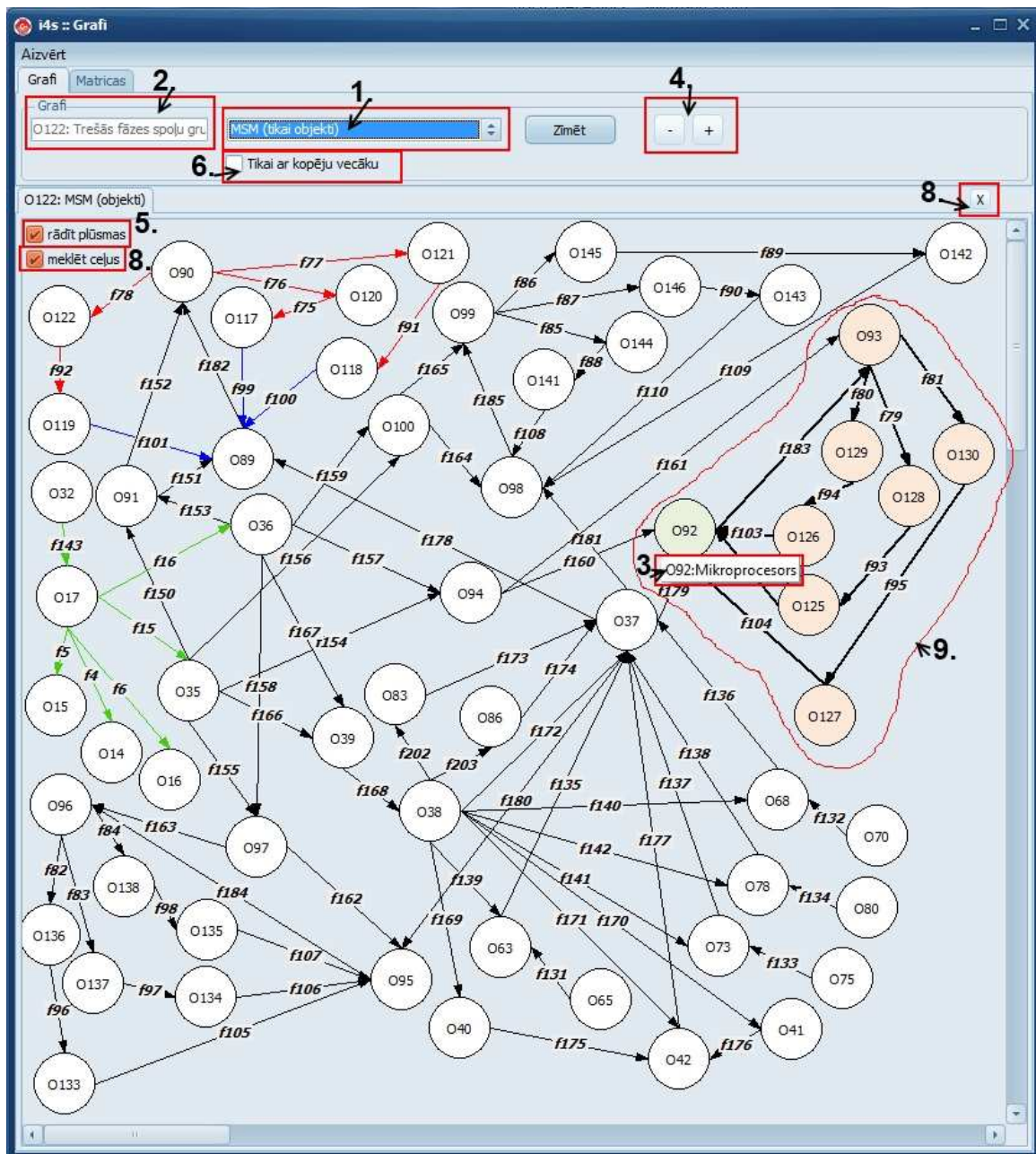
4.2.1. Struktūras modeļu izveide

Iezīmējot noteiktu freimu freimu hierarhijā un nospiežot peles labo taustiņu, eksperts dod komandu sistēmai I4S veidot struktūras modeļus. Nospiežot izvēli ‘‘Veidot modeli’’ vai taustiņu ‘‘Ctrl+G’’ kombināciju (skatīt 4.1. att. (a)), I4S atver jaunu formu, kurā iespējams izvēlēties, kādus struktūras modeļus konstruēt:

1. MSM plūsmas attēlo dažādās krāsās atbilstoši plūsmu tipam, kā arī pēc eksperta izvēles var norādīt vai nenorādīt plūsmu nosaukumus. Objektus atspoguļo, ievērojot to, vai daļa ir komponente vai elements. MSM sistēmā I4S apzīmēts kā:

- a. *MSM (tikai objekti)* (skatīt 4.15. att. 1.) ir morfoloģiskās struktūras modelis, kurā attēlo tikai objektus bez kontaktiem un starp tiem esošās plūsmas;

Katru izveidoto struktūras modeli, eksperts var saglabāt kā failu (bmp faila formātā) vai kopēt kā attēlu un ievietot, piemēram, programmā Microsoft Word. Ja struktūras modelī (jebkurā) eksperts uz virsotnes noklikšķina divreiz ar peles kreiso pogu, tad sistēma I4S ar gaišāku krāsu iezīmē izvēlēto virsotni, bet ar tumšāku visas virsotnes, kuras ir tieši vai netieši saistītas ar izvēlēto virsotni. Sistēma I4S meklē ceļus no izvēlētas virsotnes. Ceļš grafā ir virsotņu secība, kurā visas virsotnes ir dažādas [GRO 2003]. Šī sistēmas I4S iespēja, piemēram, apskatot MSM, ļauj ekspertam izprast kā katrs no sistēmas objektiem var ietekmēt citus objektus sistēmā.

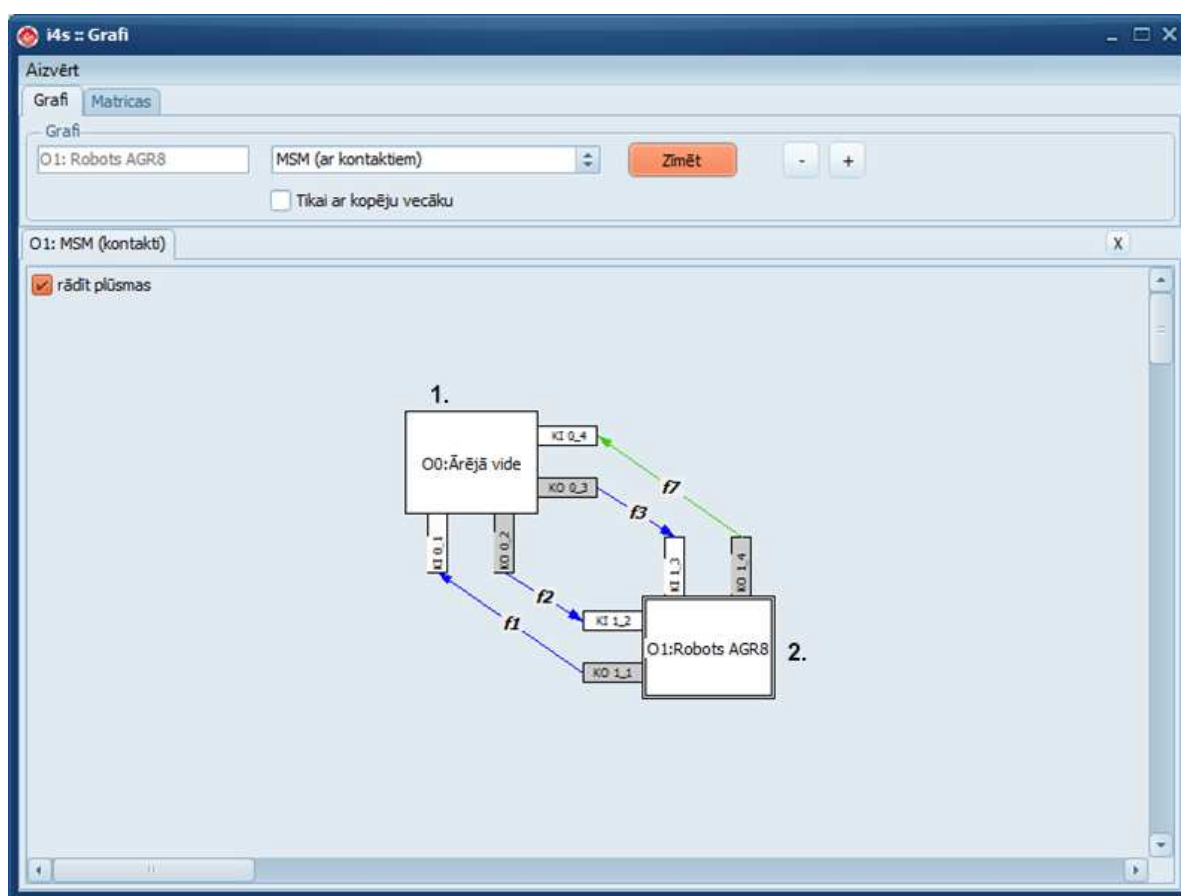


4.15. att. MSM (tikai objekti) atspoguļojums

Attēlā 4.15. att. pie atzīmes:

2. ir norādīts objekts, kas iepriekš (pirms modeļu ģenerēšanas loga) tika izvēlēts freimu hierarhijā;
3. ir redzams objekta nosaukums, kuru I4S atspoguļo, kad eksperts pietuvina kursoru virsotnei. Piemēram, virsotnei O92 atbilst objekts *O92: Mikroprocesors*;
4. ir pogas, kuras eksperts var lietot, lai palielinātu vai samazinātu struktūras modeļa elementus;
5. ir apskatīta iespēja, kas ļauj modelī attēlot plūsmu nosaukumus;

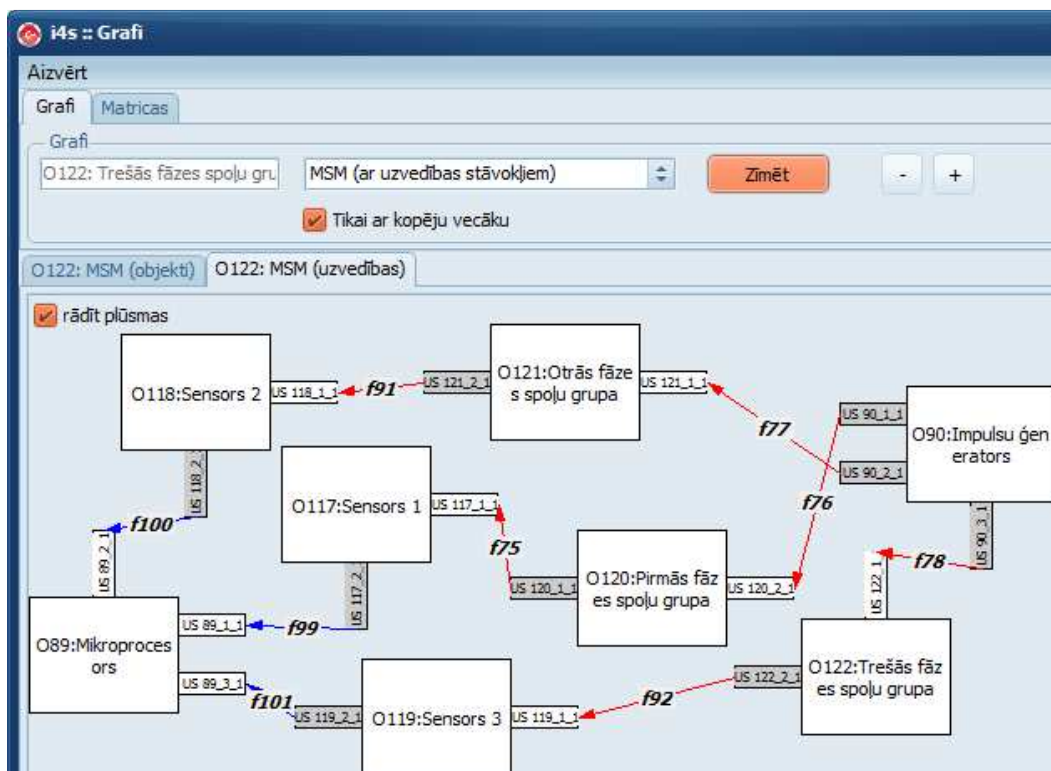
6. ir iespēja, kas ļauj ekspertam izvēlēties izveidot struktūras modeli kādai sistēmas apakšsistēmai;
7. ir poga, kuru nospiežot tiek aizvērts struktūras modeļa atspoguļojums;
8. ir apskatīta iespēja, kas ļauj modelī meklēt ceļus;
9. ir redzama izvēlētā un ar to saistītās virsotnes;
 - b. *MSM (ar kontaktiem)* – modelī atspoguļo objektus kopā ar kontaktiem un starp objektiem pastāvošās plūsmas. Attēlā 4.16. att. ir apskatīts robota AGR8 MSM, detalizācijas līmenī, kurā ir tikai objekti O0: Ārējā vide un O1: Robots AGR8;



4.16. att. MSM (ar kontaktiem) attēlojums

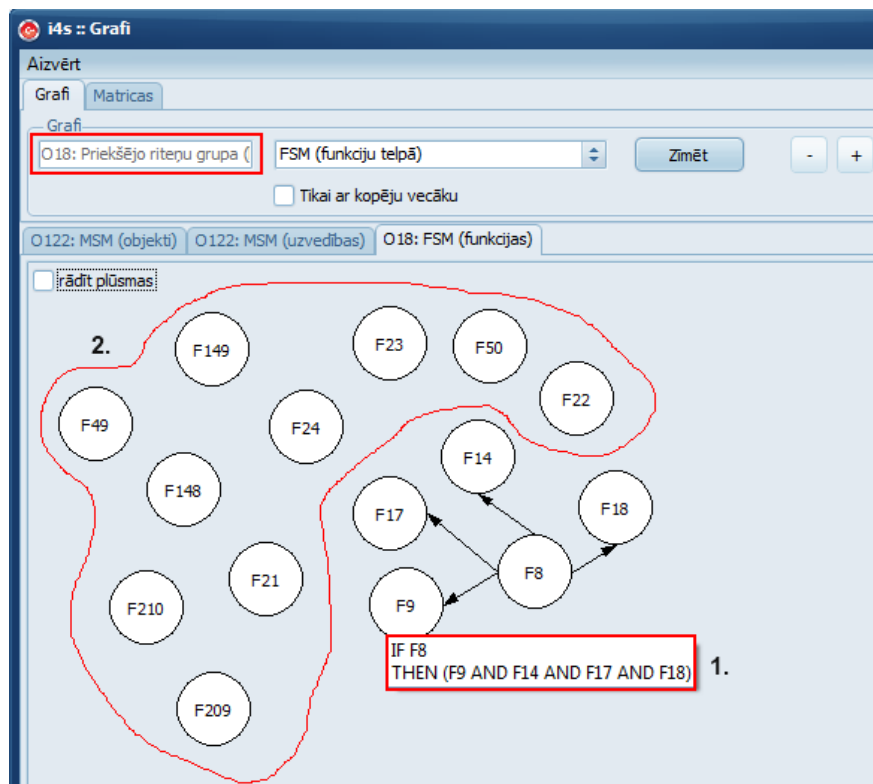
Eksperts var mainīt struktūras modelī atspoguļoto objektu novietojumu sistēmas I4S logā un I4S automātiski maina saišu un kontaktu novietojumu. MSM katram objektam tiek norādīts vai tas atbilst elementam (skatīt 4.16. att. 1.), vai komponentei (4.16. att. 2.).

- c. *MSM (ar uzvedības stāvokļiem)* – modelī attēlo objektus kopā ar uzvedības stāvokļiem un starp objektiem pastāvošās plūsmas (skatīt 4.17. att.);



4.17. att. MSM (ar uzvedības stāvokļiem) atspoguļojums

2. FSM FT sistēmā I4S ir apskatīts kā *FSM (funkciju telpā)*. Modelī atspoguļo funkcijas un starp tām pastāvošās cēloņseku saites (skatīt 4.18. att.).

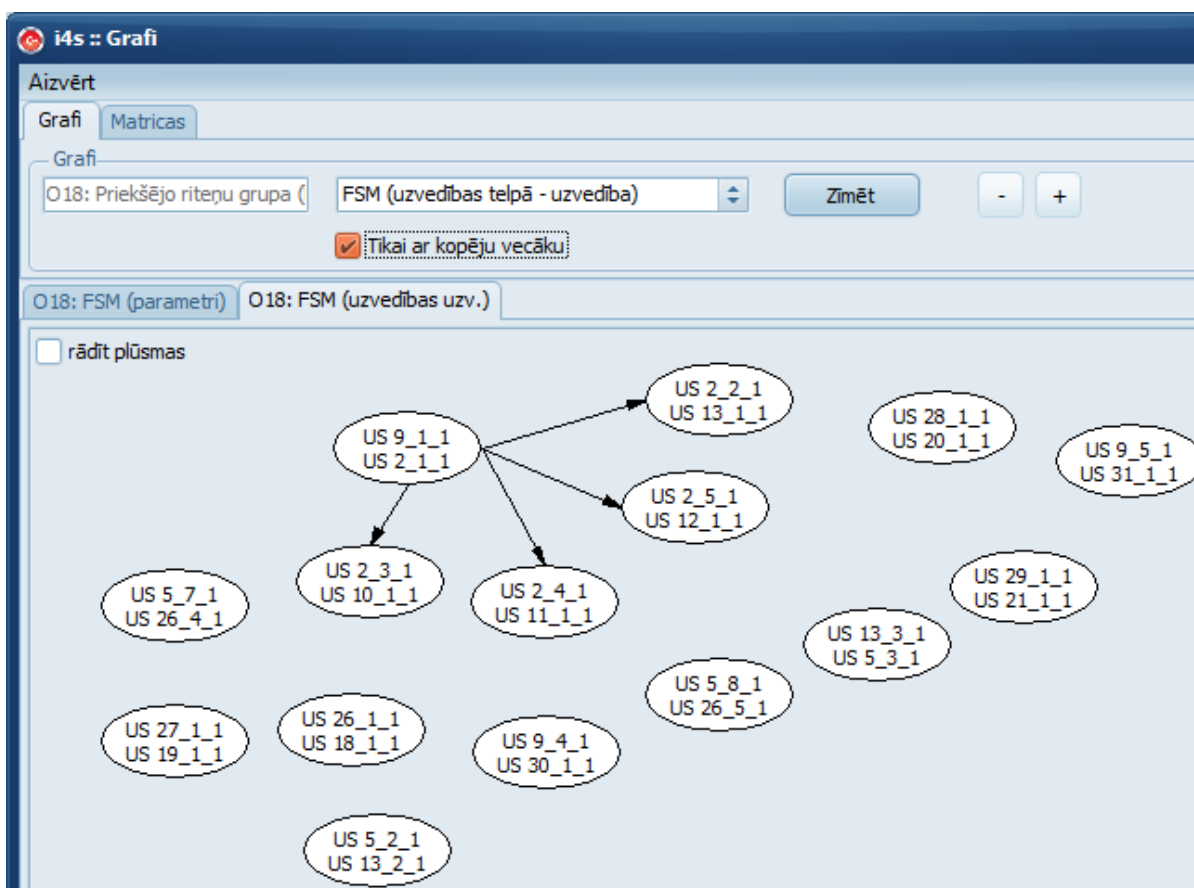


4.18. att. FSM (funkciju telpā) attēlojums

Ja eksperts freimu kopā nav aprakstījis visas saistības starp objektiem, arī tad sistēma I4S ļauj izveidot FSM FT, bet modelī tiek attēloti nesaistīti elementi (skatīt 4.18. att. 2.). Tas pierāda to, ka sistēma I4S spēj strādāt „nepilnīgas informācijas” apstākļos un izmantot visas pieejamās zināšanas, lai konstruētu struktūras modeļus. Ja modelī ir redzami nesaistīti elementi, tas nozīmē to, ka ekspertam ir jāpapildina sistēmas apraksts. Katrs no 4.18. att. attēlā apskatītiem FSM FT elementiem noteikti ir saistīts ar vismaz vienu citu elementu. Novietojot kursoru uz virsotnes, sistēma I4S atspoguļo produkcijas likumu, kas norāda uz funkciju savstarpējo saistību;

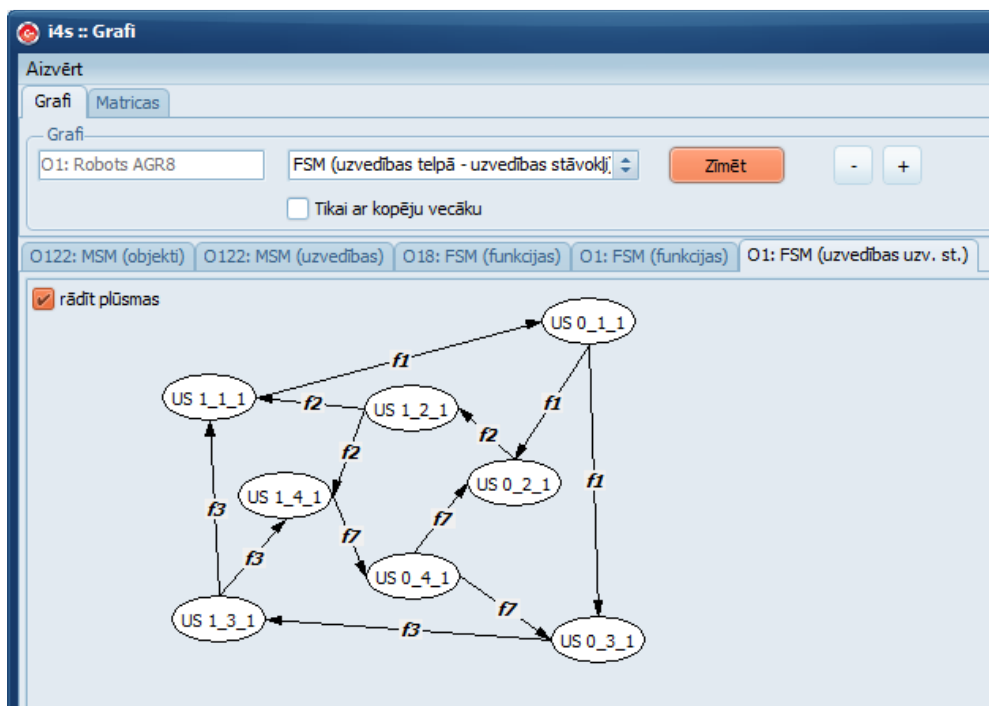
3. FSM UT sistēmā I4S apzīmēts kā:

- a. *FSM (uzvedības telpā-uzvedība)*. Struktūras modelī (skatīt 4.19. att.) attēlo uzvedību (vienkopus divi uzvedības stāvokļi) un cēloņseku saites. Attēlā 4.19. att. atspoguļotais modelis ir izomorfs FSM FT (4.18. att.);



4.19. att. FSM (uzvedības telpā - uzvedība) atspoguļojums

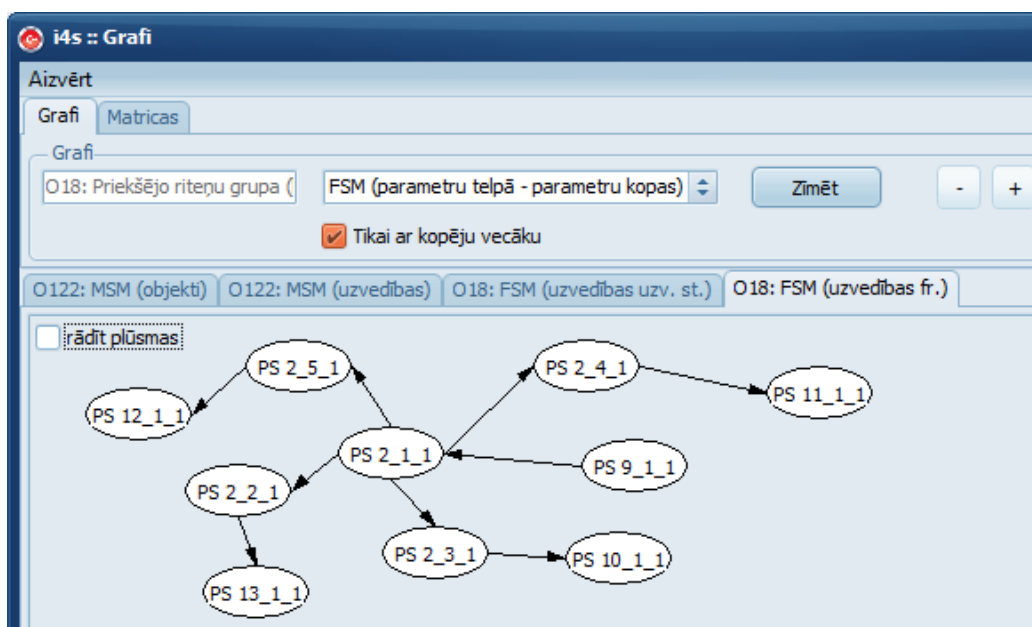
- b. *FSM (uzvedības telpā-uzvedības stāvokļi)* – modelī atspoguļo uzvedības stāvokļus un starp tiem esošās cēloņseku saites (skatīt 4.20. att.);



4.20. att. FSM (uzvedības telpā – uzvedības stāvokļi) attēlojums

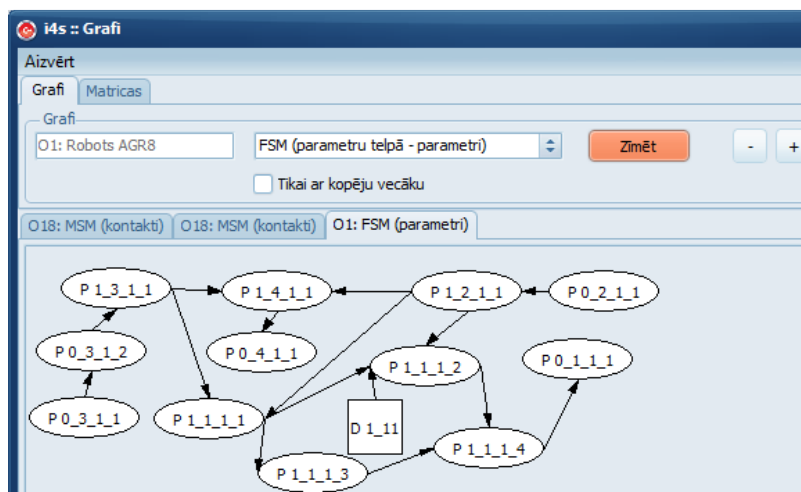
4. FSM PT sistēmā I4S apskatīts kā:

- a. *FSM (parametru telpā-parametru kopas)* – modelī attēlo parametru kopas un cēloņseku saites (skatīt 4.21. att.);



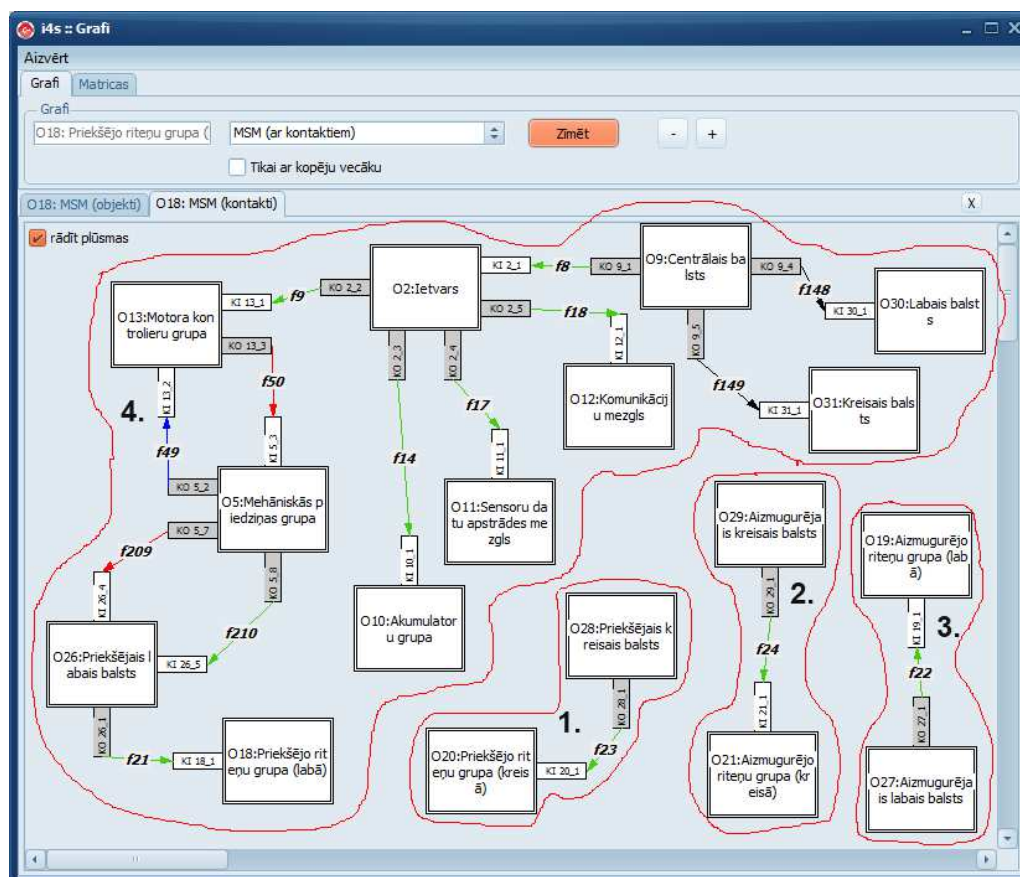
4.21. att. FSM (parametru telpā-parametru kopas) atspoguļojums

- b. *FSM (parametru telpā-parametri)* – modelī attēlo parametrus un defektus un starp tiem pastāvošās cēloņseku saites (skatīt 4.22. att.);



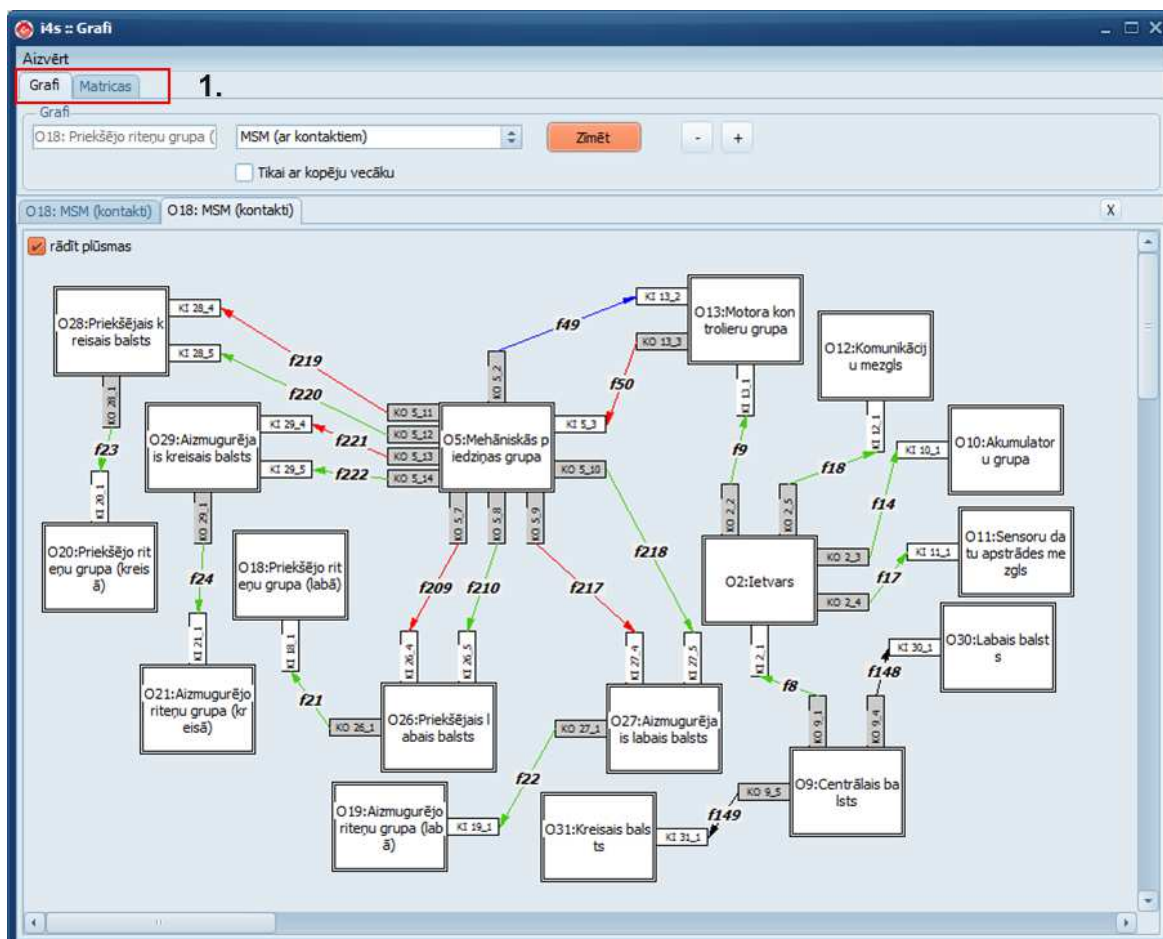
4.22. att. FSM (parametru telpā-parametri) atspoguļojums

Izvēloties kādu no konstruētiem struktūras modeļiem un nospiežot pogu „Zīmēt”, sistēma I4S attēlo struktūras modeli. Struktūras modeļi atspoguļo visu sistēmu vai izvēlēto komponenti, neskatoties uz sistēmas daļu fizisko daudzveidību, dažādo uzvedību un īpašībām. Modeļi ir viegli maināmi (skatīt 4.23. att. un 4.24. att.), un, ekspertam veicot izmaiņas freimu kopās, sistēma I4S uzkonstruē jaunu modeli.



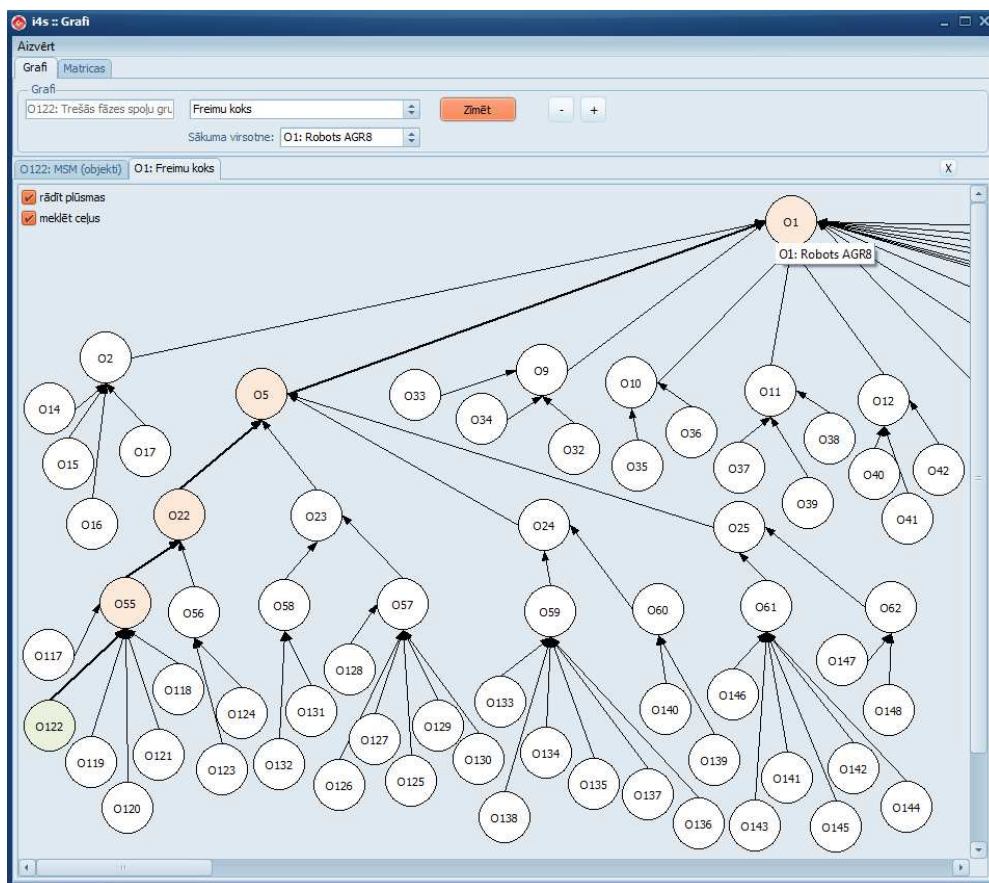
4.23. att. MSM atspoguļojums pirms izmaiņām

Piemēram, 4.23. att. attēlā apskatītajam MSM ir četras daļas, kas nav savstarpēji savienotas (norādītas ar atzīmēm 1-4). Darba autore papildināja sistēmas aprakstu, norādot saistības starp objektu O5 un objektiem O27, O28, O29 un sistēma I4S izveidoja MSM, attēlojot tajā aprakstītās saites (skatīt 4.24. att.).

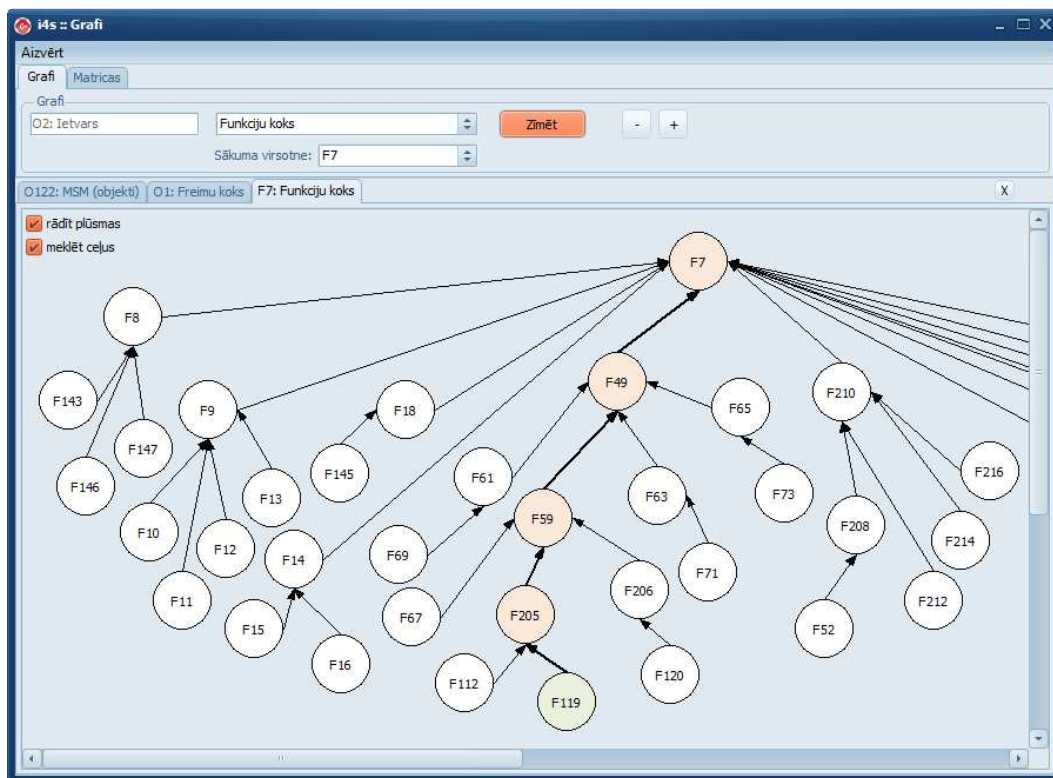


4.24. att. MSM atspoguļojums pēc izmaiņām

Sistēmā I4S tiek vizuāli atspoguļota arī objektu hierarhija, kas atbilst sistēmas uzbūvei, un funkciju hierarhija, kas atbilst sistēmas funkcionālai struktūrai. Funkcionālā struktūra ir sistēmas funkciju detalizācija (4.26. att.), ievērojot to, kā funkcijas ir pakārtotas cita citai [POL 1981]. Hierarhijas pēc eksperta izvēles var konstruēt gan visai sistēmai, gan izvēlētai sistēmas komponentei. Attēlā 4.25. att. ir apskatīta neliela daļa no visas izpētes sistēmas robots AGR8 uzbūves. Objektu hierarhiju sistēma I4S konstruē, izmantojot freimu hierarhijā atspoguļotās eksperta zināšanas par izpētes sistēmu (skatīt 4.12. att.).



4.25. att. Sistēmas robots AGR8 uzbūve



4.26. att. Funkcionālā struktūra sistēmā robots AGR8

Izstrādājot sistēmu I4S, tajā ir iekļauta arī agrāk struktūrmodelēšanā aprakstītā teorētiskā grafu analīzes daļa [GRU 1993]. Līdz ar to izveidotajam MSM vai FSM FT sistēma I4S veic topoloģisko un kvalitatīvo struktūras analīzi. Tajā pašā sistēmas I4S formā, kur ir redzami modeļi (skatīt 4.24. att. 1.), pēc eksperta izvēles, var attēlot matricas, kas atbilst struktūras modelim, izvēlētajā detalizācijas līmenī. Visas izveidotās matricas var iezīmēt, nokopēt un pārnest uz citām datora programmām, piemēram uz Microsoft Word vai Microsoft Excel. Sistēma I4S automatizēti var izveidot šādas matricas:

1. **Blakusvirsoņu matricu** (no angļu val. adjacency matrix). Tā ir kvadrātveida matrica A, kuras izmērs ir $|i| \times |j|$ (kur $i=j$ ir virsoņu skaits grafā). Divas virsoņus sauc par blakusvirsoņiem, ja starp tām eksistē loks. Matricas elementi A [i,j] orientētā grafā, kurā var būt gan paralēli loki, gan cikli ir [GRO 2003, KNO 2005]:

- „0”, ja no virsoņus i uz virsoņi j neaiziet neviens loks;
- „n”, kur n ir loku skaits kas iziet no virsoņus i un ieiet virsoņē j. No virsoņus i uz virsoņi j, apskatot sarežģītu sistēmu, var aiziet vairāk nekā viens loks.

Cikls ir noslēgts maršruts, kurā var atkārtoties virsoņus un loki. Maršruts ir mainīga virsoņu un loku secība. I4S izveidotās matricas tālāk izmanto izpētes sistēmas topoloģiskai, kvalitatīvai un kvantitatīvai analīzei [GRO 2003].

2. **Sasniedzamības matricu** (no angļu val. reachability matrix). Sasniedzamības matricu sistēma I4S iegūst, lietojot Varšalla (no angļu val. Warshall) algoritmu. Tā ir kvadrātveida matrica R, kuras izmērs ir $|i| \times |j|$ (kur $i=j$ ir virsoņu skaits grafā). Matricas elementi R [i,j] orientētā grafā ir [GRO 2003, KNO 2005]:

- „0”, ja no virsoņus i nevar sasniegt virsoņi j;
- „1”, ja no virsoņus i virsoņē j ir sasniedzama.

Ja no virsoņus i uz virsoņi j pastāv ceļš, tad virsoņē j ir sasniedzama no virsoņus i [GRO 2003].

3. **Incidenču matricu** (no angļu val. incidence matrix). Tā ir taisnstūra veida matrica I ar izmēru $|v| \times |q|$, kur rindas atbilst grafa virsoņiem, bet kolonas atbilst grafa lokiem. Ja virsoņē v ir loka q sākuma vai gala virsoņē, tad virsoņē v ir incidenta lokam q. Matricas elementi I [v,q] orientētā grafā ir [GRO 2003, KNO 2005]:

- „+1”, ja virsoņē v ir incidenta lokam q un šis loks iziet no virsoņus v;
- „-1”, ja virsoņē v ir incidenta lokam q un šis loks ieiet virsoņē v;
- „0”, ja virsoņē v nav incidenta lokam q;

4. **Attālumu matricu** (no angļu val. distance matrix). Tā ir kvadrātveida matrica D , ar izmēru $|i| \times |j|$, kur $i=j$ ir virsotņu skaits grafā. Matricas elementi $D [i,j]$ ir vienādi ar attālumu no virsotnes i līdz virsotnei j . Attālums ir īsākais ceļa garums starp virsotnēm. [GRO 2003, KNO 2005]. Attālumu matrica I4S tiek iegūta izmantojot algoritmu: (1) diagonālmatrixas (visi elementi, izņemot galveno diagonāli, ir vienādi ar nulli) diagonālē ievieto nulles un iegūst $D [i,j]$; (2) no blakusvirsotņu matricas $D [i,j]$ ievieto vieniniekus; (3) kāpina blakusvirsotņu matricu kvadrātā un $D [i,j]$ ievieto divniekus vietās, kur atbilstošā vērtība ir nulle; (4) atkārti 3 soli kāpinot blakusvirsotņu matricu kubā, utt., līdz kamēr aizpilda visus $D [i,j]$.

Sistēmā I4S eksperts var apskatīt arī blakusvirsotņu matricu reizinājumus, izvēlētā pakāpē (no 1 līdz n , kur n ir virsotņu skaits MSM). Reizinājums ir būtisks, lai noteiktu ceļu un ciklu skaitu starp struktūras modelī attēlotiem objektiem, noteiktā garumā, kā arī attālumus starp virsotnēm.

4.2.2. Sistēmas struktūras analīze

Struktūras topoloģiskā analīze ir analīzes veids, kurā pēta struktūras elementu savstarpējo saistību [НИК 1985, GRU 1993]. Sistēmā I4S ir implementēta iespēja veikt tiešo saišu analīzi un topoloģisko šķirošanu apskatītajai izpētes sistēmas struktūrai, ko attēlo MSM vai FSM FT. Lai realizētu tiešo saišu analīzi, izmanto blakusvirsotņu matricu (skatīt 4.27. att.). Blakusvirsotņu matrica ir izveidota 4.24. att. attēlā redzamajam struktūras modelim.

Matricā, apskatot katru virsotni, nosaka ieejošo un izejošo loku lokālās pakāpes. Par virsotnes v lokālo pakāpi $\rho(v)$ (grafa pakāpi virsotnē v) sauc loku skaitu, kas ir incidenti virsotnei. Orientētā grafā izdala divas lokālās pakāpes $\rho+(v)$, kas atbilst izejošo loku skaitam un $\rho-(v)$, kas atbilst ieejošo loku skaitam virsotnei v . Kad ir noteiktas lokālās pakāpes $\rho+(v)$ un $\rho-(v)$, tad izrēķina summāro lokālo pakāpi (sistēmā I4S apzīmēts ar SUM). I4S lokālās pakāpes nosaka automātiski un atspoguļo sarakstā (skatīt 4.31. att.), ko tālāk lieto, lai iegūtu elementu rangus (struktūras kvalitātes rādītājus) [GRU 1993].

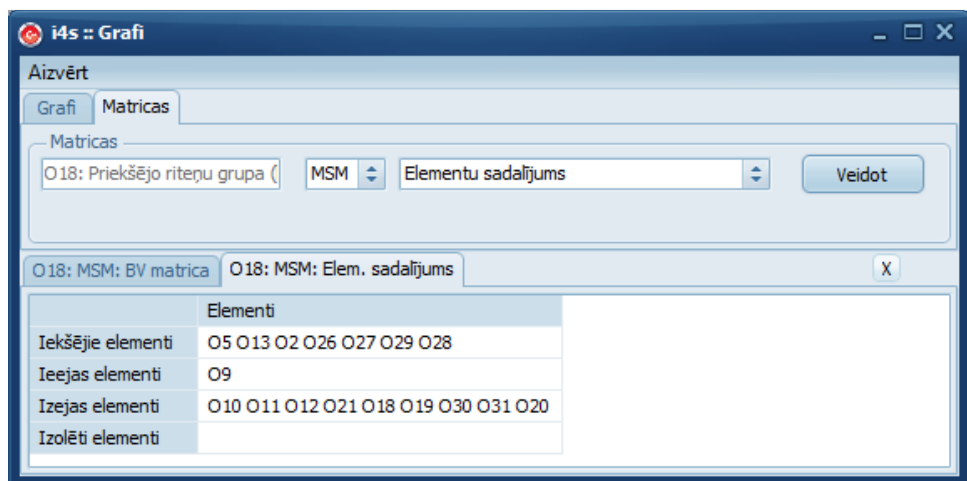
	O05	O09	O10	O11	O12	O13	O02	O21	O18	O19	O26	O27	O29	O30	O31	O20	O28	Ieejas
O05	0	0	0	0	0	1	0	0	0	0	2	2	2	0	0	0	2	9
O09	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	3
O10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
O11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
O12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
O13	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
O02	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	4
O21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
O18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
O19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
O26	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1
O27	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
O29	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1
O30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
O31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
O20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
O28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
Ieejas	1	0	1	1	1	2	1	1	1	1	2	2	2	1	1	1	2	0

4.27. att. Blakusvirсотņu matricas atspoguļojums

Lai īstenotu topoloģisko šķirošanu, izmantojot blakusvirсотņu matricu (vai lokālās pakāpes), veic klasifikāciju, sadalot apskatītos elementus četrās apakškopās [KNO 2005]:

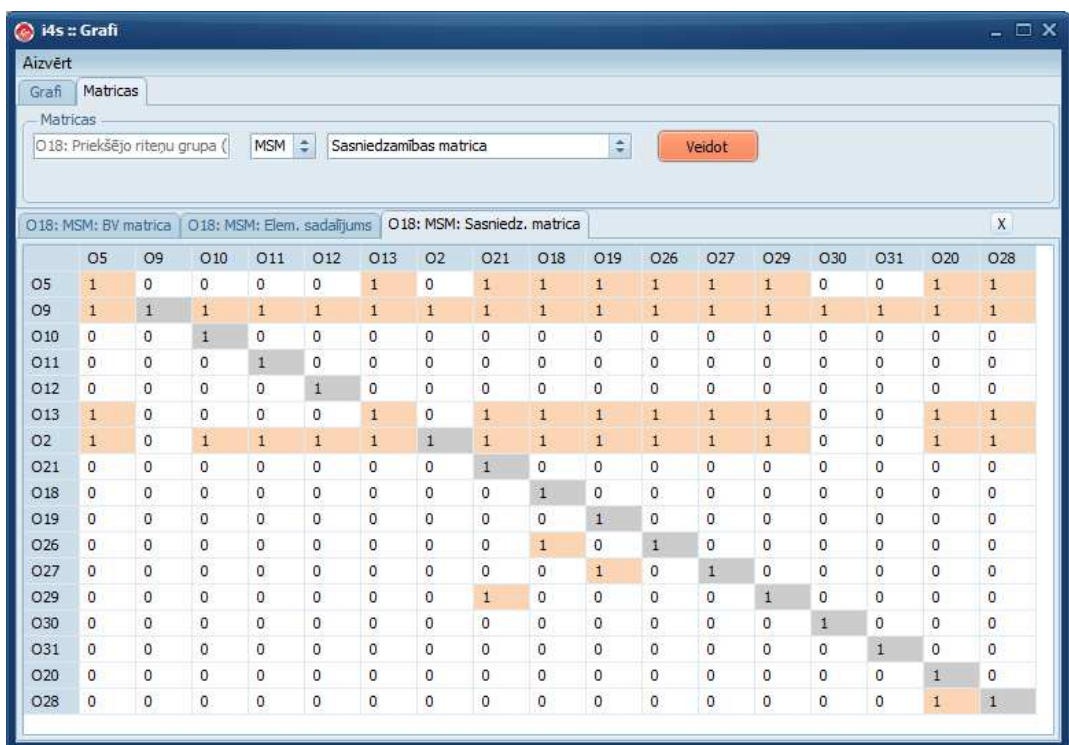
- **Ieejas elementos.** Ja elementam ieejošo loku skaits (4.27. att. „Ieejas”) blakusvirсотņu matricā ir vienāds ar 0, bet izejošo loku skaits (4.27. att. „Izejas”) ir lielāks par 0, tad tas ir ieejas elements.
- **Iekšējais elementos.** Ja elementam ieejošo loku skaits un izejošo loku skaits ir lielāks par 0, tad tas ir iekšējais elements.
- **Izejas elementos.** Ja elementam ieejošo loku skaits ir lielāks par 0, bet izejošo loku skaits ir vienāds ar 0, tad tas ir izejas elements.
- **Izolētais elementos.** Ja elementam ieejošo un izejošo loku skaits ir vienāds ar 0, tad tas ir izolētais elements.

Sistēma I4S atspoguļo iegūto elementu sadalījumu sarakstā (skatīt 4.28. att.). Ja ir uzrādīti izolēti elementi, tas nozīmē, ka eksperts nav norādījis visas saites starp izpētes sistēmas daļām.



4.28. att. Elementu sadalījuma atspoguļojums

Tālāk sistēma I4S izveido strukturizāciju pa līmeņiem, kur algoritmā lieto sasniedzamības matricu (skatīt 4.29. att.).



4.29. att. Sasniedzamības matricas atspoguļojums

Līmenis	Virsošne Si	Priekšteču kopa p-(Si)	Pēcteču kopa p+(Si)	p- UN p+
1	O9	O9	O5 O9 O10 O11 O12 O13 O2 O21 O18 O19 O26 O27 O29 O30 O31 O20 O28	O9
2	O2	O2	O5 O10 O11 O12 O13 O2 O21 O18 O19 O26 O27 O29 O20 O28	O2
2	O30	O30	O30	O30
2	O31	O31	O31	O31
3	O5	O5 O13	O5 O13 O21 O18 O19 O26 O27 O29 O20 O28	O5 O13
3	O10	O10	O10	O10
3	O11	O11	O11	O11
3	O12	O12	O12	O12
3	O13	O5 O13	O5 O13 O21 O18 O19 O26 O27 O29 O20 O28	O5 O13
4	O26	O26	O18 O26	O26
4	O27	O27	O19 O27	O27
4	O29	O29	O21 O29	O29
4	O28	O28	O20 O28	O28
5	O21	O21	O21	O21
5	O18	O18	O18	O18
5	O19	O19	O19	O19
5	O20	O20	O20	O20

4.30. att. Elementu strukturizācija pa līmeņiem

Sistēma I4S aplūko virsotnes, ko var sasniegt no apskatītās virsotnes (virsotnei atbilstošā rinda), kā arī tās kuras var sasniegt virsotni (virsotnei atbilstošā kolona), un ievieto virsotnes atbilstoši priekšteču un pēcteču kopās (skatīt 4.30. att.). Tālāk iegūst kopu šķēlumu (kolona „p- un p+”) un, ja iegūtais šķēlums atbilst apskatītai virsotnei, tad tai piešķir strukturizācijas līmeni (kolona „Līmenis”) un tālāk neapskata. Algoritms tiek īstenots līdz ir sašķirotas visas virsotnes. Strukturizācijas rezultātā iegūtos datus eksperts izmanto, lai uzzīmētu grafu, ievērojot kādos līmeņos atrodas katrs no elementiem. Strukturizāciju pa līmeņiem, lai sakārtotu virsotnes, veic tad, ja struktūra satur ciklus [GRU 1993].

Sistēma I4S ļauj īstenot izpētes sistēmas struktūras kvalitatīvo analīzi, kas ir analīzes veids, kurā tiek noteikti struktūras kvalitatīvie rādītāji – elementu rangi. Rangu var noteikt pēc dažādiem kritērijiem [GRU 1993]:

- Pēc lokālās pakāpes. Tad rangu apzīmē kā R(LP);
- Pēc ceļu un ciklu skaita ar ierobežotu garumu (sistēmā I4S eksperts var norādīt garumu no 1 līdz n, kur n ir virsotņu skaits), kas iziet no apskatītās virsotnes. Rangu apzīmē kā R(CE);
- Pēc sasniedzamo virsotņu skaita. Rangu apzīmē kā R(S).

Nosakot rangu pēc lokālās pakāpes, lieto katra elementa summāro lokālo pakāpi (skatīt 4.31. att. (a)). Jo lielāka summārā lokālā pakāpe, jo augstāks ir rangs. Augstākā ranga vērtība ir vienāda ar 1, pārējās vērtības attiecīgi 2,3 utt. Izmantojot noteikto rangu, sistēma I4S izrēķina arī elementa nozīmību $N(v)$, pēc formulas:

$$N(v) = 1 + \left(\frac{1 - R(LP)}{\sum R} \right) \quad (1.)$$

, kur:

- $R(LP)$ ir rangs pēc lokālās pakāpes
- $\sum R$ ir maksimālais $R(LP)$

Nosakot rangu pēc ceļu skaita (skatīt 4.31. att. (b)), lieto blakusvirsošņu matricas reizinājumus. Ja eksperts ir norādījis ceļus ar ierobežotu garumu 5, tad sistēma I4S veic matricu reizināšanu līdz 5. pakāpei un pēc katras kāpināšanas saskaita izejošos ceļu garumus dotajai virsotnei n (skatīt 2. formulu).

$$SUM(n) = A(v)^1 + A(v)^2 + \dots + A(v)^n \quad (2.)$$

, kur $A(v)^n$ ir summārais izejošo loku skaits virsotnei v , blakusvirsošņu matricā, kas celta pakāpē n

	p-	p+	SUM	R(LP)	Nozīmība	SUM (3)	R(CE)	SUM	Vērtība	R(S)	R(SUM)	R.TOT	Nozīmība	Rangu dispersija
O05	1	9	10	1	1.00	27	1	10	0.59	3	5	1	1.00	17.20
O09	0	3	3	3	0.50	8	4	16	0.94	1	8	3	0.60	
O10	1	0	1	4	0.25	0	6	0	0.00	5	15	5	0.20	
O11	1	0	1	4	0.25	0	6	0	0.00	5	15	5	0.20	
O12	1	0	1	4	0.25	0	6	0	0.00	5	15	5	0.20	
O13	2	1	3	3	0.50	19	2	10	0.59	3	8	3	0.60	
O2	1	4	5	2	0.75	14	3	13	0.76	2	7	2	0.80	
O21	1	0	1	4	0.25	0	6	0	0.00	5	15	5	0.20	
O18	1	0	1	4	0.25	0	6	0	0.00	5	15	5	0.20	
O19	1	0	1	4	0.25	0	6	0	0.00	5	15	5	0.20	
O26	2	1	3	3	0.50	1	5	1	0.06	4	12	4	0.40	
O27	2	1	3	3	0.50	1	5	1	0.06	4	12	4	0.40	
O29	2	1	3	3	0.50	1	5	1	0.06	4	12	4	0.40	
O30	1	0	1	4	0.25	0	6	0	0.00	5	15	5	0.20	
O31	1	0	1	4	0.25	0	6	0	0.00	5	15	5	0.20	
O20	1	0	1	4	0.25	0	6	0	0.00	5	15	5	0.20	
O28	2	1	3	3	0.50	1	5	1	0.06	4	12	4	0.40	

(a) lokālās pakāpes (b) (c) (d) (e)

4.31. att. Elementu rangi

Apskatītajā gadījumā (skatīt 4.31. att. (b)), rezultātā iegūst summāro ceļu skaitu SUM (3), kur skaitlis 3 ir maksimālais ceļa garums. Augstāko rangū R(CE) piešķir virsotnei, kurai ir lielākais SUM(n).

Aprēķinot trešo rangū (skatīt 4.31. att. (c)) izmanto sasniedzamības matricu (skatīt 4.29. att.). Katrai virsotnei saskaita virsotņu skaitu, ko tā var sasniegt (rindas summa matricā) un iegūst vērtību SUM, ko izdala ar kopējo virsotņu skaitu grafā. Iegūto vērtību lieto, lai noteiktu virsotnes rangū. Jo lielāka ir vērtība, jo augstāks ir rangū.

Summāro rangū aprēķina, pēc formulas, kur R(LP), R(CE) un R(S) ir iepriekš promocijas darbā aprakstītie trīs rangū:

$$R(SUM) = R(LP) + R(CE) + R(S) \quad (3.)$$

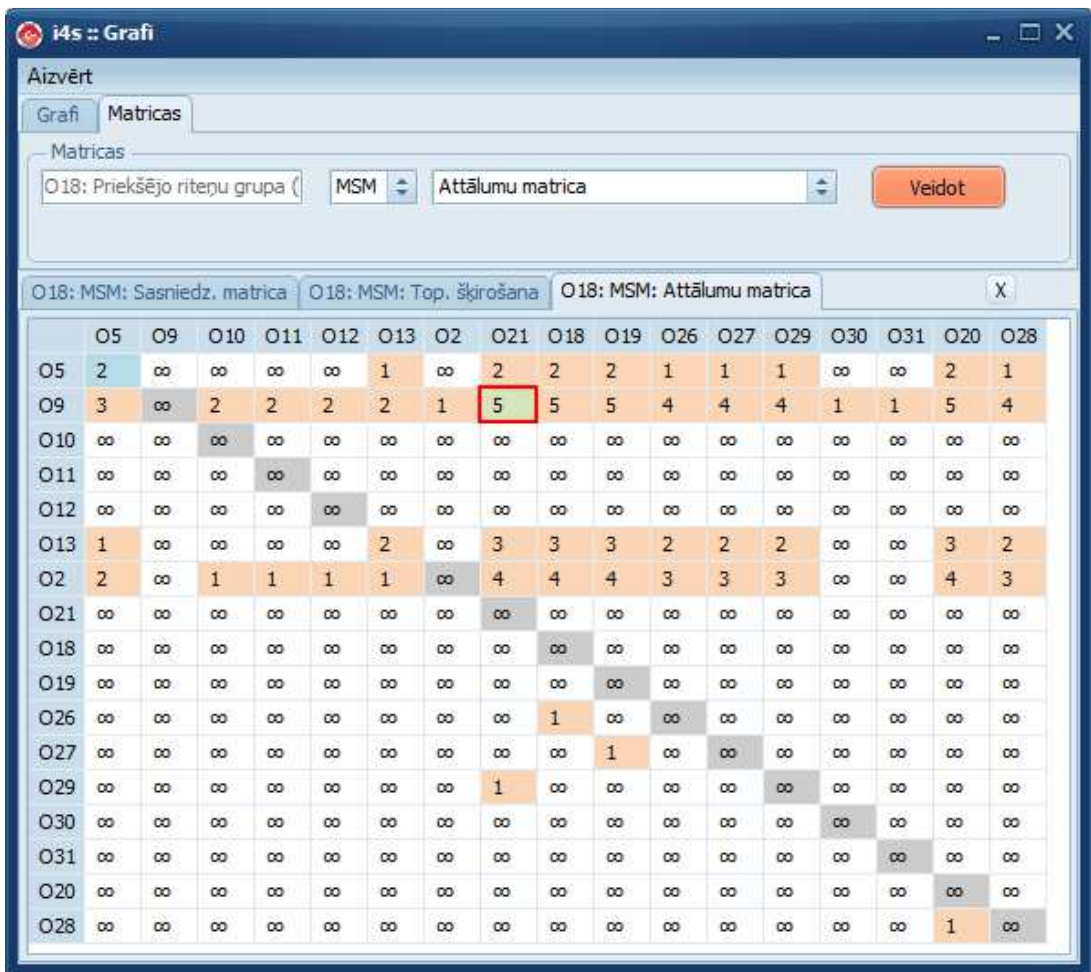
Izmantojot summāro rangū (skatīt 4.31. att. (d)), nosaka totālo rangū (R(TOT)). Jo mazāks ir R(SUM), jo augstāks ir elementa rangū. Totālo rangū lieto elementa nozīmības aprēķināšanā (skatīt 1.formulu). Jo augstāks ir elementa rangū, jo stingrāk elements ir sasaistīts ar citiem elementiem. Ja sistēmas daļai, kurai ir augsts rangū, maina tās funkcijas vai izslēdz to no sistēmas, tad tas kritiski ietekmē visas sistēmas funkcionalitāti (sistēma funkcionēt kļūdaini vai pat pārtrauc darbību). Rezultātā ir iegūts novērtējums katrai virsotnei, izmantojot 3 dažādus rangū. Maksimālā vērtība elementa nozīmībai N(v) ir vienāda ar viens. Sistēma I4S, ņemot vērā virsotņu totālos rangū, aprēķina rangū dispersiju (skatīt 4.31. att. (e)) pēc formulas:

$$D = \left(\frac{1}{R_{max}}\right) * \sum_{i=1}^{R_{max}} r_i^2 - \left(\frac{V}{R_{max}}\right)^2 \quad (4.)$$

, kur:

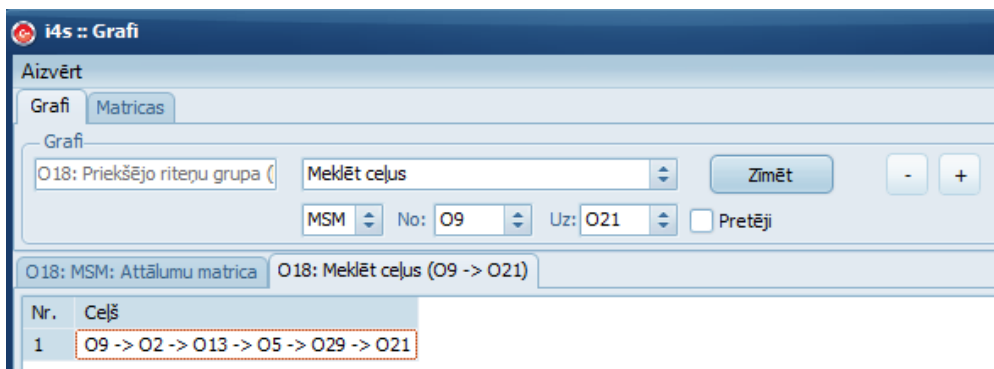
- R_{max} ir kopējais totālo rangū skaits;
- r_i ir elementu skaits ar noteiktu rangū;
- V ir kopējais virsotņu skaits grafā.

Sistēmā I4S tiek noteikts arī struktūras kvantitatīvs rādītājs – struktūras diametrs, kas tiek izcelts attālumu matricā (skatīt 4.32. att.). Diametrs ir maksimālais attālums (īsākais ceļš) starp jebkuru virsotņu pāri, kas nav vienāds ar ∞ .



4.32. att. Attālumu matrica

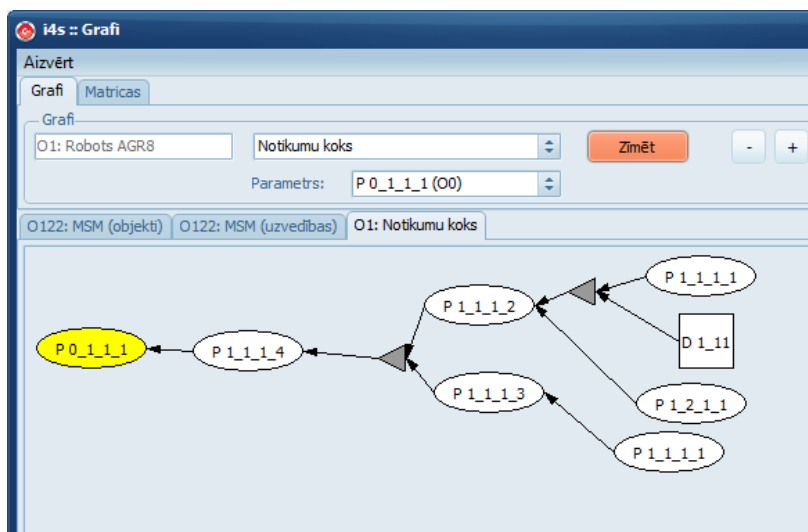
Bez matricu analīzes sistēma I4S ļauj meklēt arī ceļus izveidotajā MSM un FSM FT, starp jebkurām divām virsotnēm. Eksperts norāda sākuma virsotni, no kuras ir jāsāk meklēšana (piemēram, O9) un beigu virsotni, līdz kurai ir jāatrod ceļš (piemēram, O21). Rezultātā I4S izvada sarakstu ar ceļiem (skatīt 4.33. att.), kas pastāv starp virsotnēm.



4.33. att. Ceļu saraksts starp virsotnēm

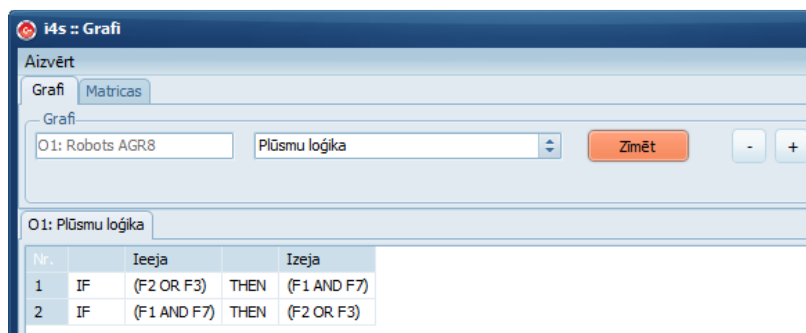
Ceļu meklēšanas mērķi ir noteikt vai viens izvēlētais objekts var ietekmēt otru objektu sistēmā, vai objekti ir saistīti, kāds ir ceļu skaits starp tiem un kāds ir šo ceļu garums. Ja no objekta O9 var sasniegt O21, tad tas nozīmē arī to, ka O9 var ietekmēt O21 funkcionalitāti. Ceļu meklēšana palīdz identificēt izpētes sistēmas struktūras „vājās” vietas, t.i., gadījumus, kad starp diviem objektiem sistēmā pastāv tikai viens ceļš. Ekspertam ir jāpievērš īpaša uzmanība šiem objektiem un starp tiem pastāvošām saitēm, jo šo objektu kļūdainas darbības rezultātā, var tikt kritiski ietekmēta visas sistēmas funkcionalitāte.

Sistēma I4S ekspertam ļauj veidot arī notikumu kokus, ko tālāk var izmantot, lai spriestu par defektu seku izplatīšanos sistēmā. Eksperts izvēlas parametru, kurā konstatētas izmaiņas, un nospiež pogu „Zīmēt”. Sistēma I4S automātiski konstruē notikumu koku (skatīt 4.34. att.), kurā izvēlētais parametrs ir notikumu koka virsotne. Notikumu koki tāpat kā FSM PT, ļauj spriest par sistēmas uzvedības izmaiņām, kā arī prognozēt, kādas sekas var izraisīt defekti vai atsevišķu elementu izslēgšana. Notikumu kokā vizuāli atspoguļotā notikumu virkne ļauj pētīt atsevišķu notikumu iestāšanās cēloņus [GRU 1993].



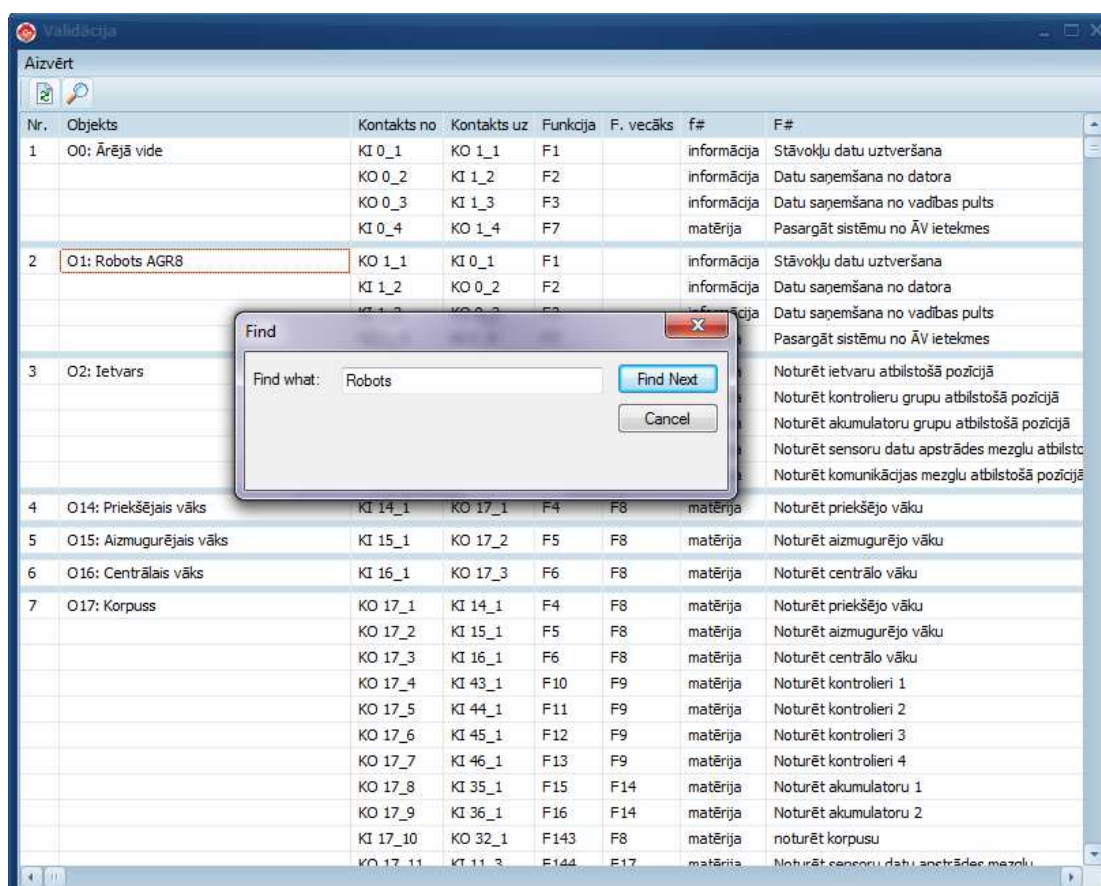
4.34. att. Notikumu koka atspoguļojums

Lai atbalstītu eksperta spriešanu par sistēmu, sistēma I4S izvada arī produkciju likumu tabulu (skatīt 4.35. att.), kas atbilst sistēmas funkcijām un to saistībām izvēlētajā sistēmas detalizācijas līmenī. Likumi ļauj noteikt funkciju izpildes secību, kā arī funkciju savstarpējo ietekmi. Iegūto produkciju likumu kopu var izmantot ekspertu sistēmā, kas ļauj realizēt spriešanu [GRU 1993, GRU 1997b], bet šis uzdevums ir ārpus promocijas darba ietvariem. Piemēram, izpētītai sistēmai *robots AGR8* pirmajā detalizācijas līmenī ir divi likumi attiecībā uz 4 funkcijām.



4.35. att. Produkciju likumu tabula

Sistēma I4S atspoguļo arī visu sistēmas daļu un tām atbilstošo kontaktu un funkciju tabulu (skatīt 4.36. att.). Tā ir izveidota, lai eksperts varētu pārbaudīt sistēmas aprakstu un konstatēt nepilnības. Tabulā ir iespējams meklēt objektus, kontaktus, funkcijas, kā arī funkciju un plūsmu vērtības. Visu atspoguļoto informāciju var nokopēt un līdz ar to izmantot arī citu veidu analīzei pēc eksperta ieskatiem.



4.36. att. Sistēmas pārbaudes tabula

Promocijas darbā ir apskatīta struktūrmodelēšanai būtiskā sistēmas I4S funkcionalitāte, bet sistēmas lietojuma apraksts ir dots 3.2. pielikumā.

4. nodaļas kopsavilkums un secinājumi

Ceturtais nodaļas mērķis ir aprakstīt praktiskās realizācijas piemēru, īstenojot sistēmā I4S zināšanu izgūšanu un atspoguļošanu attiecībā uz sarežģītu tehnisku sistēmu robots AGR8.

Ceturtais nodaļas galvenie rezultāti ir šādi:

- Ir aprakstīta izpētes sistēma robots AGR8, atspoguļojot zināšanas par objektiem, no kuriem tas sastāv, saitēm starp šiem objektiem, plūsmām, funkcijām un uzvedību, kas tiek īstenota sistēmā, kā arī parametriem un iespējamajiem defektiem;
- Ir veikta astoņu struktūras modeļu izveide robotam AGR8 un norādītas modelēšanas iespējas sistēmā I4S;
- Ir aprakstītas matricas, ko sistēma ļauj automātiski izveidot eksperta izvēlētas struktūras modelim noteiktā sistēmas detalizācijas līmenī;
- Ir skaidrotas sistēmā I4S iespējamās struktūras analīzes iespējas un doti analīzes piemēri attiecībā uz sistēmu robots AGR8.

Galvenie ceturtais nodaļas secinājumi ir šādi:

- Sistēmā I4S var:
 - realizēt zināšanu izgūšanu, atspoguļošanu, labošanu un saglabāšanu attiecībā uz izvēlēto izpētes sistēmu;
 - atspoguļot dažādas hierarhijas (objektu, funkciju, īpašību), kas ļauj izprast izpētes sistēmas uzbūvi;
 - konstruēt 8 dažādus struktūras modeļus un notikumu kokus dažādos eksperta izvēlētos sistēmas detalizācijas līmeņos gan visai sistēmai, gan atsevišķām sistēmas komponentēm.
- Sistēma I4S, izmantojot eksperta atspoguļotās zināšanas, ļauj īstenot sarežģītas tehniskas sistēmas struktūrmodelēšanu, kā arī veikt izveidotās struktūras topoloģisko un kvalitatīvo analīzi (gan MSM, gan FSM FT).
- Īstenojot struktūras modeļu automatizētu izveidi sistēmā I4S, salīdzinot ar modeļu manuālu izveides procesu, ekspertam nav nepieciešams ieguldīt daudz laika, lai izveidotu nepieciešamos struktūras modeļus (nospiežot pogu, tiek izveidots modelis).

5. REZULTĀTI UN SECINĀJUMI

Promocijas darba mērķis ir izstrādāt intelektuālu sistēmu, kas paredzēta sarežģītu sistēmu struktūrmodelēšanai, un pārbaudīt to, konstruējot sarežģītas tehniskas sistēmas struktūras modeļus. **Lai noteiktu prasības pret sistēmu, darba ietvaros ir veikti šādi uzdevumi:**

- Izanalizētas sarežģītas sistēmas un noteiktas to būtiskākās īpašības, ko nepieciešams ievērot, izstrādājot sarežģītas sistēmas modeli:
 - daudz un dažādas (homogēnas, heterogēnas, vienkāršas un saliktas) daļas, kas mijiedarbojas dažādos veidos;
 - daudzveidīga (tīklveida un hierarhiska) un sadalāma (var veikt dekompozīciju) struktūra;
 - katrai sistēmas daļai piemīt īpašības un uzvedība un sistēmai kopumā piemīt sistēmīpašības un sistēmuzvedība;
 - sistēma sadarbojas ar ārējo vidi, pašorganizējas un attīstās laikā.

Izpētes rezultātā secināts, ka sarežģītu tehnisku sistēmu specifika un pieejamais informācijas apjoms ir noteicošie faktori modelēšanas pieejas izvēlē un šiem kritērijiem atbilst struktūrmodelēšanas pieeja.

- Aplūkota struktūrmodelēšanas pieeja, noskaidrotas tās iespējas un konstatētas nepilnības, ko ir būtiski, novērst īstenojot struktūrmodelēšanu datorsistēmā;
- Veikta intelektuālu sistēmu uzbūves un darbības principu analīze un sniegta intelektuālas sistēmas definīcija, kas ir saistoša promocijas darba ietvaros;
- Izpētīts freima lietojums dažādās pieejās un noteiktas tā galvenās iezīmes, kā arī aprakstīti struktūrmodelēšanā iepriekš izmantotās zināšanu izgūšanas un atspoguļošanas shēmas trūkumi. Analīze ļāva noskaidrot prasības pret zināšanu izgūšanas un atspoguļošanas shēmu, kas ir jāimplementē intelektuālā sistēmā struktūrmodelēšanas mērķu īstenošanai.

Lai novērstu struktūrmodelēšanas pieejā konstatētās nepilnības un realizētu intelektuālas sistēmas izstrādi programmatūrā, promocijas darba ietvaros **ir iegūti šādi jauni teorētiskie rezultāti:**

- Struktūras modeļiem izstrādāti jauni pamatelementi (loģiskie operatori) un skaidrotas to pielietošanas iespējas;

- Pilnveidota struktūras modeļu sintakse un semantika un transformācijas algoritmi starp struktūras modeļiem;
- Izstrādāta zināšanu izgūšanas un atspoguļošanas shēma freimu kopa un transformācijas algoritmi no tās uz struktūras modeļiem;
- Izstrādāta intelektuālas sistēmas arhitektūra sarežģītu sistēmu struktūrmodelēšanai.

Iegūtie teorētiskie rezultāti ir praktiski realizēti intelektuālā sistēmā I4S, īstenojot šādus uzdevumus:

- Zināšanu izgūšanas un atspoguļošanas shēma – freimu kopa ir implementēta intelektuālas sistēmas arhitektūrā. Shēma ļauj izgūt un atspoguļot zināšanas no eksperta un tās saglabāt tā, lai varētu koplietot, atkārtoti pielietot, kā arī lietot automatizētai struktūras modeļu konstruēšanai un struktūras analīzei;
- Izveidotā intelektuālas sistēmas arhitektūra ir praktiski realizēta programmatūras veidā;
- Pārbaudīta izstrādātās intelektuālas sistēmas I4S darbība un atbilstība struktūrmodelēšanas mērķiem, atspoguļojot tajā zināšanas par sarežģītu tehnisku sistēmu robots AGR8 un veicot tās struktūras modeļu ģenerēšanu, kā arī struktūras analīzi.

Promocijas darba galvenais rezultāts ir izstrādāta intelektuāla sistēma I4S struktūras modeļu automatizētai konstruēšanai, kurā tiek ievērota SM definētā modeļu notācija.

Darba praktiskais pielietojums:

- Izstrādāto intelektuālo sistēmu I4S var izmantot sarežģītu tehnisku sistēmu struktūras modeļu konstruēšanai un struktūras topoloģiskās un kvalitatīvās analīzes īstenošanai;
- Izstrādātā intelektuālā sistēma I4S ir pielietojama mācību priekšmetu „Sistēmu un procesu teorija” un „Struktūrmodelēšana” apmācībā.

Darba galvenie secinājumi ir:

- Sarežģītu sistēmu izpētes pamata līdzeklis ir modeļi, kas ļauj izprast izpētes sistēmas uzbūvi un darbības principus. Lai īstenotu šādu sistēmu struktūrmodelēšanu ir nepieciešami atbilstoši rīki un pieejas;

- Freimu kopas izmantošana intelektuālā sistēmā ļauj izgūt, atspoguļot un saglabāt eksperta zināšanas par izpētes sistēmu un ar tām manipulēt;
- Automatizēta struktūras modeļu konstruēšana salīdzinājumā ar manuālu izveidi, ļauj ietaupīt eksperta laiku, veicot sistēmas izpēti;
- Izstrādātā intelektuālā sistēma nodrošina struktūras modeļu automatizētu konstruēšanu un ir piemērota sarežģītu tehnisku sistēmu struktūras modelēšanai un analīzei.

Turpmākie pētījumi ir saistīti ar:

- Papildus grafisku modeļu un algoritmu realizēšana intelektuālā sistēmā I4S, kas atbalstītu detalizētāku izpētes sistēmas analīzes īstenošanu, kā arī ļautu veikt sistēmu projektēšanu;
- Mehānismu izstrādi, kas ļauj interpretēt un pielietot sistēmā I4S iegūtos produkciju likumus.

LITERATŪRAS SARAKSTS

- [ABU 1994] Abu-Hanna A. and Jansweier W. N. H., 1994, Modelling Domain Knowledge Using Explicit Conceptualization. *IEEE-Expert* 9(5), pp. 53-64.
- [ACK 1971] Ackoff R.L., 1971, Towards a system of system concepts, *Management science*, Vol 17, No 11, pp. 661- 671.
- [ADE 1988] Adeli H., 1988, Expert systems in construction and structural engineering., Chapman and Hall, New York, NY10001, USA., p. 330 (pp.16, 229).
- [AMA 2004] Amatriain X., 2004, An Object-oriented Metamodel for digital signal processing, Thesis (PhD), Universitat Pompeu Fabra, p 399.
- [APP 2011] Jurgen Appelo, 2011, Management 3.0: Leading Agile Developers, Developing Agile Leaders, (Addison-Wesley Signature Series (Cohn)), p. 464.
- [ASH 1956] Ashby W.R., 1956, An Introduction to Cybernetics, First Edition, Chapman and Hall: London, UK, John Wiley and Sons: New York, p. 296.
- [ASH 1981] Ashby W.R., 1981, Mechanisms of Intelligence: Ashby's Writings on Cybernetics by Conant R., p. 442.
- [ASH 2004] Ashby W.R., 1962, Principles of self – organizing system, Principles of self – organization: Transactions of the University of Illinois Symposium, (Von Foerster, Zopf G.W. Eds.), Pergamon Press: London, UK, pp. 255-278. Reprinted in E:CO Special Double Issue Vol. 6 No. 1/2 Fall by Goldstein J., **2004**, pp.102-126.
- [ASH 2007] Ashby W.R. Requisite variety and its implications for the control of complex systems, *Cybernetica Vol.1*: No. 2, 1958 pp. 83-99. Reprinted in Mechanisms of Intelligence: Ashby's Writings on Cybernetics by Conant R., 1981; Reprinted on the web Principia Cybernetica Project by Heylighen F. 2007.
- [AST 1996] Astudillo H.R., 1996, Reorganizing Split Objects, In proceedings of the ACM SIGPLAN Conference on Object-Oriented Programming Systems, Languages & Applications (OOPSLA '96), pp. 138-149.
- [BAR 1997] Bar-Yam Y., 1997, Dynamics Of Complex Systems. Addison-Wesley, Reading, Massachusetts, pp. 848.
- [BEE 1995] Beer S., 1995, Platform for Change, Published by Wiley J.& Sons Ltd., reprinted from 1978, pp. 468.

- [BER 1969] Von Bertalanffy, L., 1969, General System Theory: Foundations, Development, Applications. George Braziller: NY, Revised Edition, p. 296.
- [BIE 1991] Bielawski L., Lewland R., 1991, Intelligent system design: integrating expert systems, hypermedia, and database Technologies, New York, John Wiley & Sons, p. 302
- [ВОР 2008] Воройский Ф. С., 2008, Энциклопедический систематизированный словарь-справочник ИНФОРМАТИКА, Электронное издание №5, ГПНТБ России, с. 842.
- [BOU 2004] Boulding K.E., 2004, General systems theory: E:CO Special Double Issue Vol. 6 No. 1/2 Fall, pp.127-139, reprinted by Richardson K. from The skeleton of science, Management Science, 2, 1956, pp. 197-208.
- [BRA 1984] Brachman R. J., Levesque H., J., 1984, The Tractability of Subsumption in Frame-Based Description Languages. In Proceedings of the 4th National Conference of the American Association for Artificial Intelligence (AAAI-84). Austin, TX, pp. 34-37.
- [BRA 2000] Bratko I., 2000, 15.7.2. Frames, Prolog Programming for Artificial Intelligence, 3rd edition, published by Addison Wesley, p. 678, pp. 374-379.
- [BRO 1998] Brown D.C., 1998, Intelligent Computer-Aided Design, AI in Design Group, Encyclopedia of Computer Science and Technology, (Williams J.G., Sochats K., Eds.), p. 19.
- [BRO 1999] Brown D. E., Pomykalski J. J. and Truszkowski W.F., 1999, Expert Systems, Wiley Encyclopedia for Electrical and Electronics Engineering, p. 66.
- [CAP 1996] Capra F., 1996, The Web of Life: A New Scientific Understanding of Living Systems, Anchor Books, Doubleday, New York, p. 368, pp. 3 – 112.
- [CAR 2004] Carpenter S. A., Cannady, J., 2004, Tool for Sharing and Assessing Models of Fusion-Based Space Transportation Systems, Proceedings of the 40th AIAA/SAE/ASME/ASEE Joint Propulsion Conference and Exhibit, Paper AIAA-2004-3535, p. 9.
- [CAR 2008] Carpenter S., 2008, A Primer: Enterprise Wisdom Management and the Flow of Understanding, p. 6.
- [CHA 1978] Charniak E., 1978, With A Spoon In Hand This Must Be The Eating Frame, Theoretical Issues In Natural Language Processing, Proceedings of the 1978 workshop on Theoretical issues in natural language processing, p. 187.
- [CHU 1979] Churchman C. W., 1979, The Systems Approach, Published by Delacorte Press, Paperback edition Dell Publishing, New York, p. 243.

- [COO 2001] Cook S. C., Kasser J.E., Asenstorfer J., 2001, A Frame-Based Approach to Requirements Engineering, 11th International Symposium of the INCOSE, Melbourne, Australia, p. 9.
- [COR 2003] Cornet R., Abu-Hanna, A., 2003, Using description logics for managing medical terminologies. In: Dojat, M., Keravnou, E., Barahona, P. (Eds.), Artificial Intelligence in Medicine. Proceedings of the Ninth Conference on Artificial Intelligence in Medicine in Europe (AIME), pp. 61-70.
- [CZO 1991] Czogala E., Cholewa W., 1991, Management of Statements in Frame Interpreter of CC_SHELL, Busefal journal 49 (october), LISTIC, p. 10.
- [DUR 1994] Durkin J., 1994, Expert Systems Design and Development, Macmillan Publishing, ISBN0-02-330970-9, p. 800.
- [EDM 1999] Edmonds B., 1999, Syntactic Measures of Complexity, Doctor thesis, The University of Manchester, United Kingdom, p. 245.
- [EUR 2012] European Commission, 2012, Theme 3, ICT - information and communications Technologies, Work programme 2013, p. 170.
- [EVE 1991] Evertsz R., Motta E., 1991, The abstract interpretation of hybrid rule/frame-based systems, Lecture Notes in Computer Science, Trends in artificial intelligence : 2nd Congress of the Italian Association for Artificial Intelligence, AI/IA, In Ardizzone, E. Gaglio, S. & Sorbello F. (eds.), published by Springer-Verlag Palermo, Italy, October, pp. 147 – 156.
- [FAN 1994] Fang M., 1994, MFM Model-Based Diagnosis and Implementation, *Report 94-ID-712*, Institute of Automatic Control systems, Technical University of Denmark, p. 14.
- [FAR 1997] Farkas J., Jármay K., 1997, Chapter 7: expert systems, Analysis and Optimum Design of Metal Structures (1st edition), Rotterdam-Brookfield, Balkema, p. 350, pp. 151-166.
- [FAY 1996] Fayyad U., Piatetsky-Shapiro G., Smyth P., 1996, From Data Mining to Knowledge Discovery in Databases, AI Magazine, pp. 37-54.
- [FER 2001] Ferreira P., 2001, Tracing Complexity Theory, For ESD.83 – Research Seminar in Engineering Systems, MIT Course Notebook, p. 26.
- [FIK 1985] Fikes R., Tom Kehler, 1985, The Role Of Frame-Based Representation In Reasoning, Communications of the ACM, v.28 n.9, pp. 904-920.
- [FRI 2011] Friedenbergh J.D., Silverman G., 2011, Cognitive Science: An Introduction to the Study of Mind, Second Edition, Sage Publications Inc., p. 544.

- [GAN 1993] Gan H., 1993, Script and Frame: Mixed Natural Language Understanding System with Default Theory, Methodologies for Intelligent Systems, In proceedings of 7th International Symposium (ISMIS '93), Trondheim, Norway, pp. 466-475.
- [GAR 2001] Garcia R.M., 2001, Simulation for system fault diagnosis, Master thesis, Delft university of Technology, Netherlands, p. 89.
- [GLO 2002] Glouberman S., Zimmerman B., 2002, Complicated and Complex Systems: What Would Successful Reform of Medicare Look Like? Catalogue No. CP32-79/8-2002E-IN, Discussion paper No. 8, p. 37.
- [GOL 1999] Goldenfeld N., Kadanoff L.P., 1999, Simple Lessons from Complexity, Science 2: Complex Systems, Vol. 284 no 5411, pp. 87-89.
- [GRE 1980] Greiner R., 1980, RLL-1: A Representation Language Language, Proceedings of the First National Conference of the American Association of Artificial Intelligence, p.46.
- [GRE 1996] Greenberg M., Westbrook D.L., 1996, The Frame System, Experimental Knowledge Systems Laboratory, Computer Science Department, University of Massachusetts Amherst, p.40.
- [GRO 2003] Gross J.L., Yellen J., 2003, Handbook of Graph Theory, Series: Discrete Mathematics and Its Applications Volume: 25, p. 1192.
- [GRU 1972] Grundspenkis J., 1972, Disertācijas autoreferāts, p. 132.
- [GRU 1993] Grundspenkis J., 1993, Sarežģītu tehnisku sistēmu struktūrmodelēšana nepilnīgas informācijas apstākļos, zinātnisko darbu apskats habilitētā zinātņu doktora grāda iegūšanai, p. 33.
- [GRU 1997a] Grundspenkis J., 1997. Structural Modelling of Complex Technical Systems in Conditions of Incomplete Information: A Review. In: Modern Aspects of Management Science, No 1. Riga, Latvia, pp. 111-135.
- [GRU 1997b] Grundspenkis J., 1997, Causal Domain Model Driven Knowledge Acquisition for Expert Diagnosis System Development. Applications of AI to Production Engineering. Lecture Notes of Nordic - Baltic Summer School '97. K.Wong and H.Pranevicius (Eds.), Kaunas University of Technology Press, Kaunas, Lithuania, pp.251-268.
- [GRU 1999] Grundspenkis J., 1999, Reasoning Supported by Structural Modelling. Intelligent Design, Intelligent Manufacturing and Intelligent Management. Lecturer notes of the Nordic-Baltic Summer School on Applications of AI to Production Engineering.

K.Wang and H.Pranevicius (Eds.), Kaunas University of Technology Press, Technologija, pp.57-100.

[GRU 2001] Grundspenkis J., Jekabsons J., 2001, Model Transformation for Knowledge Base Integration within the Framework of Structural Modelling. In: Databases and Information Systems. Fourth International Baltic Workshop, Baltic DB&IS 2002, Vilnius, Lithuania, May 1-5, 2000, Selected Papers, (Barzdins J., Caplinskas A., Eds.), Kluwer Academic Publishers, The Netherlands, pp. 261-274.

[GRU 2002] Grundspenkis J., 2002, Reasoning in Structural Model-Based Diagnosis. In: Proceedings of the 4th International Conference of Quality, Reliability and Maintenance, QRM2002, Oxford, March 21-22, (G. J. McNulty, Ed.), Professional Engineering Publishing, London, UK, pp. 295-298.

[GRU 2004] Grundspenkis J., 2004. Automated Transformation of the Functional Model into the Diagnosis Knowledge Base. In: Proceedings of the 5th International Conference on Quality, Reliability and Maintenance, QRM 2004, Oxford, 1-2 April, 2004, (McNulty G.J., ed.), Professional Engineering Publishing, London, UK, 2004, pp. 295-298.

[HAK 2006] Haken H., 2006, Information and Self-Organization: A Macroscopic Approach to Complex Systems, Springer Verlag, 3rd enlarged edition, pp. 262.

[HAL 1968] Hall A. D., Fagen R. E., 1968, Definition of System, in Modern Systems Research for the Behavioral Scientist: A Sourcebook (Ed. Buckley W.), Aldine Publishing Company (reprinted from *General Systems I* 1956), pp. 18-28

[HAR 1990] Harandi M. T., Lange R., 1990, Model-Based Knowledge Acquisition, In: Adeli, H. ed., *Knowledge Engineering, Vol. 1*, McGraw Hill, New York, pp. 103 – 129.

[HEY 1990] Heylighen F., 1990, Representation and Change, A Metarepresentational Framework for the Foundations of Physical and Cognitive Science, Communication & Cognition, Gent. Belgium, p. 200.

[HEY 2001] Heylighen F., 2001, The Science of Self-organization and Adaptivity, Knowledge Management, Organizational Intelligence and Learning, and Complexity (Kiel L. D. ed.), in: The Encyclopedia of Life Support Systems, (Eolss Publishers, Oxford), pp. 253-280.

[HEY 2008] Heylighen F., 2008, Five questions on complexity, Complexity: 5 questions, (In Gershenson C. Eds.), Automatic Press/ VIP, pp. 156.

[HIL 2009] Hillmann V., 2009, An Introduction to Embarcadero® C++Builder® 2010, Embarcadero Technologies whitepaper, p. 106.

- [HOP 2012] Hopgood A. A., 2012, Intelligent Systems for Engineers and Scientists, Third Edition, p. 451.
- [HOR 1995] Horgan J., 1995, From complexity to perplexity, Scientific American, Volume 272, Issue 6, pp. 74–79.
- [INF 2011] Information Society Technologies Advisory Group, 2011, 10 Key Recommendations, Vision and Needs, Impacts and Instruments, Orientations for EU ICT R&D & Innovation beyond 2013, Report for European Commission, p.61.
- [JAR 2006] Jaramillo C. M. Z., Gelbukh A. F., Isaza F. A., 2006, Pre-conceptual Schema: A Conceptual-Graph-Like Knowledge Representation for Requirements Elicitation, Lecture Notes in Computer Science, Vol. 4293, Advances in Artificial Intelligence, p. 1232, pp. 27-37.
- [JOH 2009] Johnson N.F., 2009, Simply complexity, A Clear guide to complexity theory, Oneworld, p. 256.
- [JOR 1999] Jordan M. I., Ghahramani Z., Jaakkola T. S., Saul L. K., 1999, An Introduction to Variational Methods for Graphical Models, Machine Learning, Vol 37. n.2, Nov.1., p.183-233.
- [JOS 2000] Joslyn C., Rocha L., 2000, Towards semiotic agent-based models of socio-technical organizations, Proc. AI, Simulation and Planning in High Autonomy Systems (AIS 2000) Conference, Tucson, Arizona, pp. 70-79.
- [KAK 2003] Kak A.C., 2003, Programming with Objects: A Comparative Presentation of Object-Oriented Programming with C++ and Java., John Wiley & Sons, ISBN:0471268526, p. 1115, pp. 29-36.
- [KAR 1993] Karp P.D., 1993, The Design Space of Frame Knowledge Representation Systems, Artificial Intelligence Center, SRI International, SRI AI Center Technical Note Nr. 520, p. 54.
- [KAR 1995] Karp P.D., Myers K., Gruber T., 1995, The generic frame protocol, in proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95), Canada, pp. 768-774.
- [KHA 2010] Khalil A. Yaghi K.A., Barakat S., 2010, Predicting the Life Cycle of Complex Technical Systems (CTS), In Proceedings of World Academy of Science, Engineering and Technology, Issue 69, pp. 316 – 319.
- [KNO 2005] Knorn Florian, 2005, Ranking and Importance in Complex Networks, B.Sc. thesis, Otto von Guericke Universität Magdeburg, p. 128.

- [KRI 1986] Krippendorff K. A., 1986, Dictionary of Cybernetics, NorfolkVA: The American Society for Cybernetics, pp. 81.
- [KUH 1974] Kuhn A., 1974, The logic of social systems: A unified, deductive, system-based approach to social science. Foreword by Boulding K.E., Publishers: San Francisco: Jossey-Bass., pp. 534.
- [KUI 1994] Kuipers B., 1994 Algernon for Expert Systems, Draft document in Computer Science Department University of Texas at Austin, p.38.
- [KUR 1995] Kurki M., 1995, Model-Based Fault Diagnosis for Mechatronic Systems. Thesis (PhD), Technical Research Centre of Finland, VTT Publications 223, Espoo. p. 116.
- [KUS 1997] Kusiak A., 1997, Knowledge – based systems, Lecturer notes of the Nordic-Baltic Summer School on Applications of AI to Production Engineering, K. Wang, H.Pranevicius (Ed.), Kaunas, Lithuania, pp. 45-79.
- [LAS 1990] Lassila O., 1990, Frames or Objects, or Both?., Workshop Notes from the Eight National Conference on Artificial Intelligence (AAAI-90): Object-Oriented Programming in AI, Boston (Massachusetts, U.S.A.); also published as report HTKK-TKO-B67, Otaniemi (Finland), Department of Computer Science, Helsinki University of Technology, p. 8.
- [LEE 1999] Lee F., and Heyworth, R., 1999, Errors Due to Misperception and the Default- Value Model. Advanced Research in Computers and Communications in education. G.Gumming et.al., (Eds), IOS Press, p. 6.
- [LIE 1997] Liebowitz J. (Ed.), 1997, The Handbook of Applied Expert Systems, CRC Press; 1 edition, p. 736.
- [LIN 1994] Lindland O.I., Sindre G., Solvberg., 1994, Understanding Quality in Conceptual Modeling. IEEE Softw. 11, 2, pp. 42-49.
- [LUG 2005] Luger G.F, 2005, Artificial Intelligence. Structures and Strategies for Complex Problem Solving, Fifth edition, Addison-Wesley, Harlow, England, p. 903.
- [MAG 2004] Magee C., de Weck O.L., 2004, Complex System Classification, Proceedings of the 14th Annual International Symposium of the International Council on Systems Engineering (INCOSE), France, pp.1- 18.
- [MAR 2006] Martins J. P., 2006, Foundations of knowledge representation and reasoning, Chapter 7: Frames, Technical University of Lisbon (Portugal), pp. 270 – 299.

- [MAT 1974] Maturana H.R., 1974, Cognitive strategies, in Von Foester, Heinz (ed.), Cybernetics of Cybernetics, UrbanaIL: Biological Computer Laboratory, University of Illinois, pp. 457-469.
- [MCC 1997] McCawley P. F., 1997, The Logic Model for Program Planning and Evaluation, CIS 1097, University of Idaho Extension Program, p. 5.
- [MCC 2000] McCarthy I.P., Rakotobe J.T., Frizelle G., 2000, Complex system theory: implications and promises for manufacturing organisations, *Int. J. Manufacturing Technology and Management*, Vol. 2, Nos. 1-7, pp. 559 – 579.
- [MEN 2010] Mens T., Magee J., Rumpe B., 2010, Evolving Software Architecture Descriptions of Critical Systems, *IEEE Computer* 43: 5, pp. 42-48.
- [MER 2003] Merritt D., 2003, Rules, *AI Expert Newsletter, Architecture & Design, Dr. Dobb's Journal*, Vol 11, p. 6.
- [MIN 1975a] Minsky M., 1975, A Framework for representing knowledge, (P.Winston Ed.), *The Psychology of Computer Vision*. New York: McGraw-Hill, pp. 211–277.
- [MIN 1975b] Minsky M., 1975, Minsky's frame system theory, *Proceedings of the 1975 workshop on Theoretical issues in natural language processing*, pp. 104-116.
- [MOR 2009] Morgan B., 2009, Funk2: A Distributed Processing Language for Reflective Tracing of a Large Critic-Selector Cognitive Architecture, *Proceedings of the Metacognition Workshop at the Third IEEE International Conference on Self-Adaptive and Self-Organizing Systems, California, USA*, p. 6.
- [MOU 2009] Mouchart M., Russo F., Wunsch, G., 2009, Structural modelling, exogeneity, and causality, *Causal Analysis in Population Studies: Concepts, Methods, Applications* (Engelhardt H., Kohler H. P., Fürnkranz - Prskawetz A., Eds.), *The Springer Series on Demographic Methods and Population Analysis*, Springer, Dordrecht, Netherlands, pp. 59-82.
- [NEG 2004] Negnevitsky M., 2004, *Artificial intelligence: a guide to intelligent systems*, Addison Wesley; 2 edition p. 440, pp. 131-162.
- [NGU 2010] Nguyen-Thi T.U., Martin L., 2010, Impact of R&D and ICT on Innovation and Productivity. Empirical evidence from micro data, In *Proceedings of DRUID Summer Conference 2010*, „Opening Up Innovation: Strategy, Organization and Technology”, p. 27.
- [NIE 1989] Nierstrasz O., 1989, A Survey of Object-Oriented Concepts, *Object-oriented concepts, databases and applications* (Kim W., Lochovsky F.H., Eds.), ACM Press and Addison Wesley, pp. 3 – 21.

- [NIK 2010] Nikitenko A., Kulikovskis G., 2010, Eight wheel robotic platform and its Fuzzy control system, Proceedings of International conference on automation, robotics and control systems, Orlando, USA, pp. 16. – 23.
- [OKA 2007] Okafor E.C., Osuagwu C., 2007, Issues in Structuring the Knowledge-base of Expert Systems Electronic Journal of Knowledge Management (EJKM), Volume 5 Issue 3, pp. 313-323.
- [OSI 1969] Осис Я., 1969, Топологическая модель функционирования систем. Автоматика и вычисл. техника. АН Латв. ССР, № 6, с. 44-50.
- [OXF 2009] Oxford Dictionaries, 2009, Concise Oxford English Dictionary: Luxury Edition, 11th edition, OUP Oxford, p. 1728.
- [PAL 1978] Palmer S. E., 1978, Fundamental aspects of cognitive representation, Cognition and categorization, (Eds. Rosch E., Lloyd B. L.), Hillsdale, N.J.: Erlbaum., pp. 259-302.
- [PAR 1989] Parunak H. V.D., 1989, A linguistic approach to the problem of slot semantics. In Proceedings of Eleventh Annual Conference of the Cognitive Science Society, Lawrence Erlbaum Associates, pp. 797-804.
- [PEA 2002] Pease A., Liuzzi R.A., Gunning D., 2002, Knowledge bases, Encyclopedia of Software Engineering, 2nd ed., Volume I (J. Marciniak, Ed.), USA, New York: Wiley & Sons, pp. 696-704.
- [PEI 2011] Peine A., 2011, Challenging Incommensurability: What We Can Learn From Ludwik Fleck for the Analysis of Configurational Innovation, Minerva, Volume 49, Issue 4, pp. 489-508.
- [POD 2012] Podnieks K., 2012, Datizrases metodes (I, II), Datizrace (Data Mining) kurss, p. 47.
- [POL 1981] Половинкин А.И., Бобков Н.К., Буш Г.Я., 1981, Автоматизация поискового конструирования (Искусственный интеллект в машинном проектировании), Под ред. А.И. Половинкина, Радио и связь, p. 344.
- [POL 2002] Polyakov L.M., 2002, Structure Approach to the Intelligent System Design, Proceedings of the 2002 PerMIS Workshop, p. 10.
- [POL 2004] Polyakov L.M., 2004, Agent with Reasoning and Learning: The Structure Design, Performance Metrics for Intelligent Systems, Gaithersburg, MD., p. 9.
- [POS 2012] PostgreSQL Global Development Group, 2012, PostgreSQL 9.1.5 Documentation, University of California, p.2652.

- [RAM 1997] Ramirez C., 1997, Schemata, Frames, and Dynamic Memory Structures. Technical Report No. 7-97, Computing Laboratory - University of Kent at Canterbury, p.10.
- [RIC 2000] Richardson K. A., Mathieson G., Cilliers P., 2000, The Theory and Practice of Complexity Science: Epistemological Considerations for Military Operational Analysis, *SystemeMexico*, Vol 1: pp. 25-66.
- [RIC 2001] Richardson K.A., Cilliers P., 2001, What is Complexity Science? A View from Different Directions, *Emergence*, Vol. 3 (1), pp. 5-23.
- [ROB 1977] Roberts R. B. and Goldstein, I. P., 1977, The FRL Primer. Technical Report. UMI Order Number: AIM-408, Massachusetts Institute of Technology. p. 28.
- [ROB 1994] Robinson G.P., Colchester, A.C.F., Griffin, L.D., 1994, Model-Based Recognition of Anatomical Objects from Medical Images, *Image and Vision Computing* 12 (8), pp. 499-507.
- [ROE 1969] Roe P.H., 1969, Soulis G.N., Handa V.K. The discipline of design, published by University of Waterloo, p. 284.
- [ROS 1979] de Rosnay J., 1979, The macroscope: a new world scientific system, Publisher Harper & Row, pp. 247, reprinted in *Principia Cybernetica Web*, 1999, by Vranckx A., Heylighen F.
- [RUD 2008] Rudas I.J., Fodor J., 2008, Intelligent Systems, *International Journal of Computers Communications and Control*, Volume: 3, Suppl. Issue: Proceedings of ICCCC, pp. 132-138.
- [RUS 2010] Russell S., Norvig P., 2010, *Artificial Intelligence: A Modern Approach* (3rd Edition), Pearson Education, Upper Saddle River, New Jersey, p. 1152.
- [SHA 2003] Sharma A.K., Kumar C., Mustafa K., & Kumar A., 2003, A Fuzzy Frame Based Expert Shell, In *Proceedings of the National workshop on Information Technology Services and Applications (WITSA-2003)*, New Delhi, p. 6.
- [SHE 2000] Shedroff N., 2000, An Overview of Understanding, *Information Anxiety 2* (Wurman R.S. author), (Hayden/Que), 2nd edition, p. 350, pp. 27-29.
- [SIM 1962] Simon H.A., 1962, The Architecture of Complexity. *Proceedings of the American Philosophical Society*, Vol. 106, No. 6., republished by JSTOR, 2007, pp. 467-482.
- [SIM 1987] Simon H.A., 1987, *Models of man: social and rational*, published by Garland Pub., New York, pp. 287 (p. 198), Reprint. Originally published: New York : Wiley, 1957.

- [SKY 2006] Skyytner L., 2006, General Systems Theory: Problems, Perspective, Practice. World Scientific Publishing Company; 2 edition, pp.536.
- [SNO 2005] Snowden D., 2005, Multi-ontology sense-making; a new simplicity in decision making, Management Today, Volume: 13, Issue: 1, Publisher: Radcliffe Publishing Ltd., pp. 1 – 13.
- [SNO 2010] Snowden D., 2010, The Origins of Cynefin, Cognitive Edge Pte Ltd., pp. 17.
- [SOK 2010] Sokolowski, J. A. and Banks, C. M. (eds), 2010, Modeling and simulation fundamentals: theoretical underpinnings and practical domains, John Wiley & Sons, Inc.Publication, USA, p. 437.
- [SON 2008] Sonar S.G, 2008, Urban Sprawl A System Dynamic Approach, Proceedings of the 44th ISOCARP Congress, p. 10.
- [SPE 2004] Spenser C., 2004, WIN-Prolog 4.9 Flex Tutorial, Logic Programming Associates Ltd, p. 68.
- [STA 2006] Stahl T., Voelter M., 2006, Chapter 2: MDSD – Basic Ideas and Terminology, Model-Driven Software Development: Technology, Engineering, Management, p. 444, pp. 11-29.
- [STE 1978] Steels L., 1978, Frame-Based Knowledge Representation, MIT Artificial Intelligence Laboratory Working Papers, WP-170, MIT Artificial Intelligence Laboratory, p.29.
- [STE 1979] Stefik M., 1979, An examination of a frame-structured representation system. Proceedings of the International Joint Conference on Artificial Intelligence, Tokyo, Japan (IJCAI-79), pp. 845-852.
- [STO 2001] Stockburger D. W., 2001, Introductory Statistics: Concepts, Models, and Applications, Atomic Dog Publishing, 2nd edition, p. 287.
- [TUT 2001] Tutte W.T., 2001, Graph Theory, Cambridge University Press, New York, p. 360.
- [UEN 1991] Ueno h., Yamamdto Y., Fukuda H., 1991, Knowledge modeling and model-based problem solving-towards a multi-use engineering knowledge base, Applications of Supercomputers in Engineering II, (Brebbia C.A., Howard D., Peters A., Eds.), Elsevier Applied Science, pp. 215-231.
- [VAL 2005a] Valkovska I., Grundspenkis J., 2005, Development of Frame Systems Shell for Learning of Knowledge Representation Issues. In Proceedings of the International Conference on Computer Systems and Technologies – CompSysTech’05, Varna,

Bulgaria, June 16-17, (Ed. Rochev B., Smrikarov A.), The Bulgarian Chapter of ACM, pp. IV.11.-1 – IV.11.-6.

[VAL 2005b] Valkovska I., Grundspenkis J., 2005, Representation of Complex Agents by Frames for Simulation of Internal Relationships in Structural Modelling. In Proceedings of the 19th European Conference on Modelling and Simulation (ECMS 2005), pp. 151-157.

[VAL 2005c] Valkovska I., Graudina V., Grundspenkis J., 2005, Usage of Frame System for Modelling of Intelligent Tutoring System Architecture. In Annual Proceedings of Vidzeme University College. ICTE in Regional Development. Valmiera, pp. 105-109.

[VAZ 2009] Vazquez, F. and Gonzalez-Avella, J. and Eguiluz, V. and Miguel, M.S., 2009, Collective Phenomena in Complex Social Networks, In journal Applications of Nonlinear Dynamics Model and Design of Complex Systems, (In, Visarath; Longhini, Patrick; Palacios, Antonio, eds) XII , Springer Verlag, pp. 189-200.

[VIC 2002] Vicsek T., 2002, Complexity: The bigger picture. Nature, Volume 418, Issue 6894, pp. 131.

[WEA 2004] Weaver W., 2004, Science and complexity, E:CO Special Double Issue Vol. 6 No. 3, pp. 65-74, reprinted by Wirth K from American scientist, 36: 536, 1948, pp. 536-544.

[WEI 2009] Weisfeld M.A., 2003, The Object-Oriented Thought Process, Addison-Wesley Professional, 3rd edition, p.360, pp. 10- 50.

[WHE 1993] Wheeler, L. A., Shapiro, A. E., 1993, Method and apparatus for storing information about and associating slot behaviors of a slot in a frame-based semantic network, United States Patent US5208899, (Pittsburgh, PA), p. 33.

[WHI 1999] Whitesides G.M., Ismagilov R.F., 1999, Complexity in Chemistry, Science 2: Complex Systems, Vol. 284, pp. 89-92.

[WOO 2004] Woog R. Attachment F, 2004, Contemporary Theories and Thinking, The Knowing of Knowledge, p. 19.

[YOU 2007] Young B.J., Booch G., Conallen J., Engel M.W., Houston K. A., Maksimchuk R.A., 2007, Chapter 1: Complexity, Object-Oriented Analysis and Design with Applications, 3rd Edition, Addison-Wesley, p. 717, pp. 3-29.

[ZEL 2007] Zeltmate, I., 2007, Frame system development for structural modelling knowledge representation purpose, Master thesis, RTU, Riga, Latvia, p. 214.

[ZEL 2008a] Zeltmate I., Grundspenkis J., Kirikova M., 2008, The Challenges in Knowledge Representation for Analysis of Inter - Institutional Knowledge Flows. In Proceedings of

the IADIS International Conference on Cognition and Exploratory Learning in Digital Age (CELDA 2008), Freiburg, Germany, pp. 145-152.

[ZEL 2008b] Zeltmate I., Grundspenkis J., 2008, Formal Method of Functional Model Building Based on Graph Transformations. In Proceedings of the 5th International Mediterranean and Latin American Modeling Multiconference, The international Workshop on Modelling & Applied Simulation, Italy, pp. 140-147.

[ZEL 2010a] Zeltmate I., Grundspenkis J., 2010, An Extension of Frame-Based Knowledge Representation Schema, In proceedings of the International Multi-Conference on Complexity, Informatics and Cybernetics. Vol.1, USA, Orlando, pp. 401-406.

[ZEL 2010b] Zeltmate I., Kirikova M., Grundspenkis J., 2010, Prototype for the Knowledge Representation Supporting Inter-Institutional Knowledge Flow Analysis. Learning And Instruction In The Digital Age, Part II - Knowledge Representation and Mental Models, Chapter 6 (Spector J.M., Ifenthaler D., Isaías P., Kinshuk., Sampson D.G. Ed.), New York, Springer, pp. 87-99.

[ZEL 2011] Zeltmate I., 2011, Logical Operator Usage in Structural Modelling, In Proceedings of the 10th International Conference on Modeling and Applied Simulation, Italy, p. 338 -346.

[НИК 1985] Николаев В.И., Брук В.М., 1985, Системотехника: методы и приложения, „Машиностроение”, с.199.

[ЮР 2007] Юревич Е. И., 2007, Основы робототехники 2-е издание (ISBN 978-5-94157-473-5), БХВ-Петербург, 408 стр.