

RĪGAS TEHNISKĀ UNIVERSITĀTE

Egons LAVENDELIS

**ATVĒRTA DAUDZAGENTU ARHITEKTŪRA UN
METODOLOĢIJA INTELEKTUĀLU MĀCĪBU SISTĒMU
IZSTRĀDEI**

Promocijas darba kopsavilkums

Rīga 2009

RĪGAS TEHNISKĀ UNIVERSITĀTE
Datorzinātnes un informācijas tehnoloģijas fakultāte
Lietišķo datorsistēmu institūts

Egons LAVENDELIS
Datorsistēmu doktora programmas doktorants

**ATVĒRTA DAUDZĀĢENTU ARHITEKTŪRA UN
METODOLOĢIJA INTELEKTUĀLU MĀCĪBU SISTĒMU
IZSTRĀDEI**

Promocijas darba kopsavilkums

Zinātniskais vadītājs
Dr.habil.sc.ing., profesors
J. GRUNDSPEŅĪS

Rīga 2009

UDK 004.89(043.2)

La 943 a

Lavendelis Egons. Atvērta daudzāģentu arhitektūra un metodoloģija intelektuālu mācību sistēmu izstrādei. Promocijas darba kopsavilkums. RTU, 2009. - 49 lpp.

Iespiests saskaņā ar DITF LDI padomes 2009. gada 15. jūnija lēmumu, protokols Nr. 64.

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

ISBN 987-9984-32-317-6

**PROMOCIJAS DARBS
IZVIRZĪTS RĪGAS TEHNISKAJĀ UNIVERSITĀTĒ
INŽINIERZINĀTŅU (datorsistēmu)
DOKTORA GRĀDA IEGŪŠANAI**

Promocijas darbs inženierzinātņu (datorsistēmu) doktora grāda iegūšanai tiek publiski aizstāvēts 2009.gada 21. decembrī plkst. 14:30 Rīgas Tehniskās universitātes Datorzinātnes un informācijas tehnoloģijas fakultātē, Meža ielā 1/3, 202. auditorijā.

OFICIĀLIE RECENZENTI:

Profesors, Dr.habil.sc.ing. Leonīds Novickis
Rīgas Tehniskā universitāte

Asoc. profesors, Dr.sc.comp. Guntis Arnicāns
Latvijas Universitāte

Profesors, PhD (Datorzinātne) Jaak Tepandi
Tallinas Tehniskā universitāte, Igaunija

APSTIPRINĀJUMS

Es apstiprinu, ka esmu izstrādājis šo promocijas darbu, kas iesniegts izskatīšanai Rīgas Tehniskajā universitātē inženierzinātņu doktora grāda iegūšanai. Promocijas darbs nav iesniegts nevienā citā universitātē zinātniskā grāda iegūšanai.

Egons Lavendelis..... (paraksts)

Datums:.....

Promocijas darbs rakstīts latviešu valodā, un sastāv no ievada, 5 nodaļām un secinājumiem. Darbā ir 30 tabulas un 49 attēli, kopā 222 lappuses. Bibliogrāfiskais saraksts satur 164 nosaukumus.

Saturs

| | |
|---|----|
| Ievads | 5 |
| 1. Aģentos sakņotas intelektuālas mācību sistēmas | 12 |
| 2. Aģentos sakņotu intelektuālu mācību sistēmu izstrāde..... | 16 |
| 2.1. Daudzaģentu sistēmu arhitektūras..... | 16 |
| 2.2. Aģentu realizācijas platformas | 17 |
| 2.3. Aģentu izstrādes metodoloģijas | 19 |
| 3. Atvērta daudzģentu arhitektūra intelektuālas mācību sistēmas realizācijai | 22 |
| 4. MASITS metodoloģija un rīks | 26 |
| 4.1. MASITS metodoloģija | 26 |
| 4.2. MASITS metodoloģijas automatizācijas algoritmi..... | 33 |
| 4.3. MASITS rīks | 34 |
| 5. Intelektuālas mācību sistēmas praktiska piemēra izstrāde, izmantojot MASITS metodoloģiju un rīku | 36 |
| 6. Rezultāti un secinājumi | 40 |
| Bibliogrāfiskais saraksts..... | 43 |

Ievads

Mūsdienās aizvien lielāku nozīmi sabiedrībā ieņem zināšanas, un daudzas valstis, tajā skaitā Latvija savā Nacionālajā attīstības plānā [NAP 2006], ir deklarējušas virzību uz zināšanu sabiedrību. Zināšanu sabiedrības izveidi atbalsta arī UNESCO [BIN 2005] un Eiropas Komisija kopā ar Eiropas Savienību [BOL 1999], [LIS 2000], [CCU 2005].

Lai izveidotu zināšanu sabiedrību, ir jāpaplašina izglītības pieejamība, jāatbalsta mūžizglītība un jārealizē elastīgas izglītības iespējas, kas veicina zināšanu apgūšanu jebkurā laikā un jebkurā vietā, adaptējot mācīšanās procesu katra individuāla apmācāmā spējām un vēlmēm. Taču mācību procesā, kurā tiek apmācītas lielas studentu grupas (piemēram, datorzinātņu studenti Rīgas Tehniskajā universitātē), nav iespēju pielāgoties katra apmācāmā vēlmēm par apmācības vietu un laiku. Lai spētu apmierināt zināšanu sabiedrībai nepieciešamās prasības pēc izglītības, bez tradicionālās klātienē apmācības tiek izmantotas dažādas datorizētas tehnoloģijas, tajā skaitā e-apmācības un m-apmācības tehnoloģijas, kas ļauj iegūt zināšanas jebkurā laikā un praktiski jebkurā vietā.

No datorizētām apmācības tehnoloģijām mūsdienās visplašāk tiek izmantotas e-apmācības sistēmas, kas apmācāmajam piedāvā mācību materiālus un novērtē apmācāmā zināšanas, izmantojot objektīvus vai subjektīvus testus [ANO 2007]. Taču šīs sistēmas visiem apmācāmajiem lieto vienus un tos pašus mācību materiālus un kontroles uzdevumus, tādējādi neveicot individualizētu apmācību. Līdz ar to tās ir grūti izmantojamas, ja apmācāmajiem ir dažāds zināšanu līmenis. Bez tam apmācības laikā netiek ņemti vērā apmācāmo mācīšanās stili un dažādas uztveres un psiholoģiskās īpašības. Šajās sistēmās parasti trūkst arī intelektuāla atbalsta. Tās nespēj izmantot problēmsfēras zināšanas, lai apmācāmajam sniegtu jebkādu papildus paskaidrojumu vai atgriezenisko saiti. Visi mācību materiāli, kontroles uzdevumi ir jāizveido pasniedzējam, un sistēma spēj tikai tos attēlot.

Lai novērstu minētās e-apmācības problēmas, veido intelektuālas mācību sistēmas (IMS), kurās iekļauj dažādus intelektuālus mehānismus, kas atdarina pasniedzēja darbības, lai veiktu individualizētu apmācību, izmantojot problēmsfēras zināšanas. IMS-as spēj adaptēt mācību materiālus, ģenerēt kontroles uzdevumus no problēmsfēras zināšanām, patstāvīgi novērtēt apmācāmā zināšanas un sniegt pietiekami informatīvu atgriezenisko saiti [ANO 2007].

Tēmas aktualitāte

Neskatoties uz to, ka dažādi IMS-u prototipi eksistē jau vairākus gadu desmitus un tie realizē adaptīvu un intelektuālu apmācību ar visām iepriekš minētajām IMS-u priekšrocībām, IMS-as nav kļuvušas par industriālu risinājumu un vēl joprojām tiek veidotas tikai zinātnisku pētījumu rezultātā. Iemesls ir fakts, ka IMS-ām veltītie pētījumi galvenokārt koncentrējas uz sistēmu funkcionalitāti un konkrētu sistēmu realizēšanu, nevis tieši uz IMS-u izstrādi. Tas ir visai sarežģīts process, kura rezultātā ir jārealizē intelektuāli mehānismi, kas nepieciešami, lai IMS spētu aizstāt cilvēku-skolotāju. Bez tam IMS-as bieži vien ir nepieciešams modificēt, lai vai nu attīstītu mācāmo kursu, vai tās pielāgotu citam mācību kursam. Lai vienkāršotu sistēmu izstrādi un uzturēšanu, ir nepieciešama tehnoloģija, kas palielina IMS-u modularitāti un veicina izmaiņu ieviešanu tajās.

Pēdējos gados, lai realizētu intelektuālus mehānismus IMS-ās un veicinātu IMS-u modularitāti, aizvien biežāk tiek izmantoti intelektuālie aģenti. Ir izstrādāti vairāki desmiti aģentos sakņotu IMS-u. ABITS sistēma attālinātai apmācībai [CAP 2000b], kompilatoru apmācības sistēma WADIES [GEO 2003], intelektuāla virtuāla apmācības vide [DEA 2003] un medicīnas māsu apmācības sistēma Ines [HOS 2003] ir tikai daži šādu sistēmu piemēri. Taču zināmās sistēmu realizācijas neļauj esošās sistēmas viegli pielāgot jaunam pielietojumam un atkārtoti izmantot to moduļus vai tās veidojošos aģentus. Lai risinātu šīs problēmas, IMS-u arhitektūru ir nepieciešams modificēt tā, lai tajā būtu atkarājami izmantojamas komponentes un tās būtu atvērtas jaunas funkcionalitātes pievienošanai.

Aģentos sakņotām IMS-ām ir nozīmīgas specifiskas īpašības, kas ierobežo dažādu aģentorientētas programmatūras izstrādes metodoloģiju izmantošanu IMS-u izstrādei. Lai arī pētījumi par IMS-as veidojošiem aģentiem un IMS-u arhitektūru ir noderīgi IMS-u izstrādē, tos nav iespējams efektīvi izmantot kopā ar kādu vispārīgu aģentorientētas programmatūras izstrādes metodoloģiju, jo metodoloģijas neļauj pievienot izstrādājamo sistēmu pētījumu rezultātus. Tādēļ ir nepieciešams izveidot specifisku aģentos sakņotu IMS-u izstrādes metodoloģiju, kas atbilstu visām IMS-u īpašībām un iekļautu IMS-u pētījumu rezultātus, tādējādi atvieglojot praktisku IMS-u izstrādi.

Promocijas darba mērķis

Darba mērķis ir izstrādāt pieeju IMS-u modularitātes, atvērtības un atkārtotas izmantojamības nodrošināšanai, kas sevī ietver piemērotu arhitektūru un specializētu IMS-u izstrādes metodoloģiju, ko atbalsta rīks.

Darba uzdevumi

Lai sasniegtu izvirzīto darba mērķi, ir definēti šādi darba uzdevumi:

- Izpētīt aģentu pielietojumu IMS-u izstrādē un identificēt neatrisinātos uzdevumus.
- Izanalizēt dažādu daudzāģentu arhitektūru piemērotību IMS-u realizēšanai.
- Izstrādāt atvērtu daudzāģentu arhitektūru, kas nodrošina aģentos sakņotu IMS-u izstrādi un izmaiņu ieviešanu tajās.
- Izanalizēt aģentu realizācijas platformu piemērotību aģentos sakņotu IMS-u izstrādei.
- Izanalizēt vispārīgo aģentorientētas programmatūras izstrādes metodoloģiju un tajās lietoto tehniku piemērotību IMS-u izstrādei.
- Izstrādāt aģentos sakņotu IMS-u izstrādes metodoloģiju, kas apvieno aģentorientētas programmatūras izstrādes un IMS-u pētījumu rezultātus.
- Realizēt rīku, kas atbalsta izstrādāto metodoloģiju un realizē tādas funkcijas, kā diagrammu zīmēšana, nepretrunīguma nodrošināšana un šķērspārbaude starp diagrammām un koda ģenerēšana.
- Pārbaudīt izstrādātās arhitektūras piemērotību, kā arī metodoloģijas un rīka lietojamību, veicot praktisku IMS-as izstrādi.

Pētījuma objekts

Darba pētījumu objekts ir aģentos sakņotas intelektuālas mācību sistēmas.

Pētījuma priekšmets

Darba pētījumu priekšmets ir aģentos sakņotu intelektuālu mācību sistēmu arhitektūra, to izstrādes metodoloģijas un realizācijas iespējas.

Darba zinātniskais jauninājums ir šāds:

- Ir izstrādāta holoniska daudzāģentu arhitektūra aģentos sakņotu IMS-u izstrādei, kas veicina visu nepieciešamo aģentu izstrādi, kā arī izmaiņu ieviešanu realizētajās IMS-ās.
- Ir definēts aģentorientētas programmatūras izstrādes dzīves cikls. Dzīves cikls izstrādāts, modificējot ūdenskrituma modeli, lai tas atbilstu populārākajām aģentorientētas programmatūras izstrādes metodoloģijām.
- Ir izstrādāta aģentorientētas programmatūras izstrādes metodoloģiju klasifikācija pēc to izcelsmes.

- Ir izstrādāta aģentos sakņota IMS-u izstrādes metodoloģija MASITS, kas aģentorientētas programmatūras izstrādes procesā integrē zināšanas no IMS-u pētījumiem un iekļauj tikai IMS-u īpašībām atbilstošas tehnikas, kuras vai nu pārņemtas no vispārīgām aģentorientētas programmatūras izstrādes metodoloģijām, vai definētas no jauna, ja tādu nav vispārīgajās metodoloģijās.
- Daļai no IMS-u izstrādes gaitā veicamajiem soļiem definētas jaunas tehnikas. Nozīmīgākās no tām ir: (1) uzdevumu piešķiršana aģentiem, izmantojot IMS-u pētījumu rezultātos sakņotus likumus, (2) aģentu iekšējās struktūras projektēšana, izmantojot MASITS aģenta iekšējo skatu un (3) holonu projektēšana, izmantojot holonu hierarhijas diagrammu un mijiedarbības diagrammu.
- Ir izstrādāti algoritmi aģentos sakņotu IMS-u izstrādes automatizēšanai. Algoritmi nodrošina nepretrunīgumu un šķērspārbaudes starp diagrammām, kā arī daļēju diagrammu un koda ģenerēšanu.
- Ir izstrādāts MASITS metodoloģiju atbalstošs rīks, kas nodrošina metodoloģijā iekļauto diagrammu zīmēšanu, to nepretrunīguma nodrošināšanu un šķērspārbaudes starp tām, kā arī diagrammu daļu un JADE platformai atbilstoša Java koda ģenerēšanu no izveidotajām diagrammām.
- Ir realizēta intelektuāla mācību sistēma MIPITS pirmo trīs kursa „Mākslīgā intelekta pamati” moduļu mācīšanai, kas apmācāmā raksturojumiem adaptē zināšanu novērtēšanā izmantotos kontroles uzdevumus.

Pētījumu praktiskā nozīmība

Darba praktiskā nozīme ir saistīta ar MASITS rīku un MIPITS sistēmu. MASITS rīks nodrošina praktisku IMS-u izstrādi, izmantojot MASITS metodoloģiju. Savukārt, MIPITS sistēma realizē kursa „Mākslīgā intelekta pamati” mācīšanu. Darbā izstrādātās arhitektūras pielietošana MIPITS sistēmā ļauj to pielāgot arī citu līdzīgu kursu apgūšanai, jo tās piedāvātos uzdevumus un mācību materiālus ir iespējams vienkārši izmainīt.

Darba aprobācija. Darba rezultāti ir prezentēti četrās starptautiskās konferencēs Latvijā, Nīderlandē, Spānijā un Portugālē:

- 2009. gada 21.-23. jūnijs. IADIS International Conference „Intelligent Systems and Agents 2009”, Algarve, Portugāle.

- 2009. gada 25.-27. marts. 7th International Conference on Practical Applications of Agents and Multi-Agent Systems, Salamanka, Spānija.
- 2008. gada 13.-15. oktobris. RTU 49. Starptautiskā zinātniskā konference, Rīga, Latvija.
- 2008. gada 22.-24. jūlijs. IADIS International Conference „Intelligent Systems and Agents 2008”, Amsterdama, Nīderlande.

Darba rezultāti publicēti septiņās publikācijās starptautiskos un Latvijas Zinātnes Padomes atzītos zinātniskos izdevumos:

- Lavendelis E., Grundspenkis J. Open Holonic Multi-Agent Architecture for Intelligent Tutoring System Development. Proceedings of IADIS International Conference „Intelligent Systems and Agents 2008”, Amsterdam, The Netherlands, 22 - 24 July 2008, pp. 100-108.
- Lavendelis E., Grundspenkis J. MASITS - A Tool for Multi-Agent Based Intelligent Tutoring System Development. Proceedings of 7th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 2009), Salamanca, Spain, 25-27 March 2009. Advances in Intelligent and Soft Computing Vol. 55., Springer, 2009, pp. 490-500.
- Lavendelis E., Grundspenkis J. Design of Multi-Agent Based Intelligent Tutoring Systems. RTU Zinātniskie raksti, 5. sērija, Datorzinātne, Lietišķās datorsistēmas, 38. sējums. Izdevniecība “RTU”, Rīga, 2009, pp. 48-59.
- Lavendelis E., Grundspenkis J. Requirements Analysis of Multi-Agent Based Intelligent Tutoring Systems. RTU Zinātniskie raksti, 5. sērija, Datorzinātne, Lietišķās datorsistēmas, 38. sējums. Izdevniecība “RTU”, Rīga, 2009, pp. 37-47.
- Lavendelis E., Grundspenkis J. MASITS – A Multi-Agent Based Intelligent Tutoring System Development Methodology. In Proceedings of IADIS International Conference „Intelligent Systems and Agents 2009”, 21-23 June 2009, Algarve, Portugal, pp. 116-124.
- Lavendelis E., Grundspenkis J. MIPITS - An Agent Based Intelligent Tutoring System. Submitted at the Second International Conference on Agents and Artificial Intelligence (ICAART 2010), Spain, Valencia, 22-24 January 2010.
- Lavendelis E., Grundspenkis J. Multi-Agent Based Intelligent Tutoring System Source Code Generation Using MASITS Tool. Submitted at 50th RTU International Scientific Conference, Riga, Latvia, October 13-15 2009.

Papildus publikācijām par promocijas darba rezultātiem, darba autors ir publicējis šādas ar promocijas darba tematiku saistītas publikācijas:

- Anohina A., Lavendelis E., Grundspenkis J. Concept Map Based Knowledge Assessment System with Reduction of Task Difficulty. Information Systems Development “Challenges in Practice, Theory and Education” (Proceedings of the 16th International Conference on Information Systems Development (ISD`2007), August 29-31, 2007, Galway, Ireland), Vol.2, Springer, 2009, pp. 853-866.
- Lavendelis E., Stankeviča L., Atvare D., Pozdņakovs D., Kirikova M. Development of the Information System’s Concept for Supporting Scientific Activities at Riga Technical University. RTU Zinātniskie raksti, 5. sērija, Datorzinātne, Lietišķās datorsistēmas, 30. sējums. Izdevniecība “RTU”, Rīga, 2007. 125.-137. lpp.
- Lanka A., Grundspenkis J., Anohina A., Pozdnyakov D., Lavendelis E. Knowledge assessment and self-assessment in the process-oriented learning, using intelligent system based on concept maps. Starpaugstskolu Zinātniski praktiskā un mācību metodiskā konference “Mūsdienu izglītības problēmas”, Transporta un sakaru institūts, 2006. gada 23.-24. februārī, Rīga, Latvija, 65.-68. lpp.
- Lavendelis E., Grundspenkis J. Simulation Tool for Multicriteria Auctions in Transportation and Logistics Domain. Proceedings of the International Conference on Computer Systems and Technologies – CompSysTech`06, June 15-16, 2006, Veliko Tarnovo, Bulgaria, ACM Bulgaria, 2006, pp. IIIB.17-1 - IIIB.17-6.
- Grundspenkis J., Lavendelis E. Multiagent Based Simulation Tool for Transportation and Logistics Decision Support. Proceedings of the 3rd International Workshop on Computer Supported Activity Coordination – CSAC 2006, May 16-19, 2006, Cyprus, Paphos, pp. 45-54.

Darba autors ir bijis izpildītājs šādos zinātniskos projektos:

- Latvijas Republikas Izglītības un Zinātnes ministrijas un Rīgas Tehniskā universitātes pētnieciskais projekts „Jēdzienu tīklu un ontoloģiju balstīta intelektuāla sistēma studentu zināšanu pašvērtēšanai un procesu orientētai zināšanu pārbaudei” (no 2006. gada jūnija līdz decembrim).
- Latvijas Republikas Izglītības un Zinātnes ministrijas un Rīgas Tehniskā universitātes pētnieciskais projekts „Informācijas sistēmas koncepcijas izstrāde RTU zinātniskās darbības atbalstam” (no 2006. gada jūnija līdz decembrim).
- Eiropas Savienības 6. ietvarprogrammas projekts eLOGMAR-M (Web-based and Mobile Solutions for Collaborative Work Environment with Logistics and Maritime Applications) (no 2005.g. septembra līdz 2006. gada septembrim).

- Latvijas Republikas Izglītības un Zinātnes ministrijas un Rīgas Tehniskā universitātes pētnieciskais projekts „Intelektuāla sistēma procesu orientētas studiju efektivitātes analīzes atbalstam” (no 2005. gada septembra līdz decembrim).

Darba struktūra

Promocijas darbā ir ievads, 5 nodaļas, secinājumi, bibliogrāfiskais saraksts un 5 pielikumi. Promocijas darba pamatteksts satur 222 lappuses, 49 attēlus un 30 tabulas.

Ievadā analizēta izglītības nozīme zināšanu sabiedrībā, uzsverot e-apmācības un m-apmācības nozīmi apmācības piemērošanai zināšanu sabiedrībai. Ir identificētas galvenās e-apmācības problēmas un ir konstatēts, ka piemērotākā tehnoloģija šo problēmu novēršanai ir intelektuālas mācību sistēmas.

Darba *pirmajā nodaļā* ir izanalizētas IMS-as, to galvenās funkcijas, struktūra un intelektuālu aģentu izmantošana to realizēšanai. Nodaļas sākumā aprakstīti e-apmācības sistēmu trūkumi, ko iespējams novērst, tām pievienojot adaptivitāti un intelektu, tādējādi iegūstot IMS-as. Nodaļas turpinājumā izanalizētas IMS-u galvenās funkcijas un arhitektūra, kā arī definētas darbības, ko IMS-ām jāveic, lai imitētu cilvēku-pasniedzēju, un analizēti intelektuālie mehānismi, kas tiek izmantoti šo darbību realizēšanai IMS-ās. Pēdējās divas apakšnodaļas veltītas intelektuāliem aģentiem un to pielietojuma IMS-u realizācijā analīzei.

Darba *otrajā nodaļā* ir veikta aģentos sakņotu IMS-u izstrādei nepieciešamo tehnoloģiju analīze. Nodaļā analizētas eksistējošās daudzāģentu arhitektūras, aģentu realizācijas platformas un izstrādes metodoloģijas. No arhitektūru analīzes (2.1. apakšnodaļa) secināts, ka IMS-u izstrādei piemērotākā daudzāģentu arhitektūra ir holoniskas daudzāģentu sistēmas. 2.2. apakšnodaļā veikta aģentu realizācijas platformu analīze, kuras rezultātā secināts, ka piemērotākā aģentos sakņotu IMS-u realizācijas platforma ir JADE. 2.3. apakšnodaļā analizētas aģentorientētas programmatūras izstrādes metodoloģijas un tajās izmantotās tehnikas ar mērķi noteikt to piemērotību aģentos sakņotu IMS-u izstrādei. Pētot eksistējošās metodoloģijas, definēts arī aģentorientētas programmatūras dzīves cikls.

Darba *trešajā nodaļā*, balstoties uz 2. nodaļā veikto analīzi par aģentu pielietojumu IMS-u izstrādē un holoniskiem aģentiem, izstrādāta atvērta holoniska daudzāģentu arhitektūra IMS-u realizēšanai. Izstrādātā arhitektūra nodrošina IMS-u modularitāti un atvērtību, tādējādi atvieglojot aģentos sakņotu IMS-u izstrādi un izmaiņu ieviešanu tajās.

Darba *ceturtajā nodaļā* ir aprakstīta MASITS metodoloģija aģentos sakņotu IMS-u izstrādei. Katrai metodoloģijas dzīves cikla fāzei ir definēti veicamie soļi un izstrādāts fāzes

izpildes algoritms. Papildus metodoloģijas aprakstam izstrādāti izstrādes automatizācijas algoritmi un izveidots rīks, kas šos algoritmus realizē.

Darba *piektajā nodaļā* veikta IMS-as izstrāde, izmantojot 3. nodaļā definēto atvērto holonisko arhitektūru un 4. nodaļā izstrādāto MASITS metodoloģiju. IMS-a domāta mācību kursa „Mākslīgā intelekta pamati” mācīšanai. Nodaļā definētas prasības sistēmai, veikta sistēmas izstrāde, izmantojot MASITS metodoloģiju un rīku, un aprakstīta izstrādātā sistēma.

Darba *noslēguma daļā* doti galvenie darba rezultāti un praktiskais pielietojums, kā arī sniegti iespējamie turpmāko pētījumu virzieni.

Darbam pievienoti pieci pielikumi: pielikums P1 „Atvērta daudzāģentu arhitektūra intelektuālu mācību sistēmu realizēšanai”, pielikums P2 „MASITS metodoloģijā izmantoto aprakstu šabloni”, pielikums P3 „MASITS rīka lietotāja ceļvedis”, pielikums P4 „MIPITS sistēmas izstrādes artefakti” un pielikums P5 „MIPITS sistēmas lietotāja ceļvedis”.

1. Aģentos sakņotas intelektuālas mācību sistēmas

Pirmā nodaļa veltīta intelektuālām mācību sistēmām. Nodaļā konstatēta datortehnoloģiju balstītas apmācības, it īpaši e-apmācības, kas ir visplašāk lietotā no datorizētām apmācības tehnoloģijām, nozīme un priekšrocības mūsdienās. E-apmācības priekšrocības ir šādas [SIR 2007], [ZHA 2003]: tā ir pieejama jebkurā vietā un jebkurā laikā, taupa laiku un līdzekļus, piedāvā ērtu komunikāciju ar pasniedzēju un dod neierobežotu pieeju mācību materiāliem.

Balstoties uz literatūras avotu analīzi, ir konstatēti nozīmīgi e-apmācības trūkumi. E-apmācības sistēmām trūkst adaptivitātes un intelekta. E-apmācības sistēmas parasti visiem apmācāmajiem dod vienus un tos pašus mācību materiālus un kontroles uzdevumus. Tādējādi šāda pieeja neļauj realizēt individualizētu apmācību un kļūst neefektīva apmācāmajiem ar dažādiem zināšanu līmeņiem un mācīšanās stiliem. E-apmācības sistēmas nespēj izmantot problēmsfēras zināšanas, lai sniegtu paskaidrojumus un ģenerētu jaunus materiālus. Līdz ar to viss, ko sistēma dod apmācāmajam, ir jāizveido pasniedzējam, kas ir apjomīgs darbs. Minēto trūkumu dēļ e-apmācības sistēmas nevar sekmīgi aizstāt cilvēku-pasniedzēju.

Lai novērstu e-apmācības trūkumus, tiek veidotas intelektuālas mācību sistēmas (IMS), kas ir datorizētas sistēmas, kuras lieto apmācības mērķiem un kam piemīt šādas intelektuālām sistēmām raksturīgas īpašības [ANO 2007]: (1) tās pielieto mākslīgā intelekta principus un metodes (tādas kā dabīgās valodas apstrāde, zināšanu atspoguļošana un mašīnu apmācība), (2) tās ir adaptīvas sistēmas, jo pielāgo savu darbību katram apmācāmajam,

tādējādi, realizējot individuālu apmācību, (3) tās imitē cilvēku-pasniedzēju un (4) tās balstās uz apmācības un izziņas teoriju.

Pēdējo 40 gadu laikā ir izstrādāti vairāki desmiti IMS-u, piemēram, pirmā intelektuālā mācību sistēma SCHOLAR, kas paredzēta ģeogrāfijas mācīšanai [CAR 1970], Ines sistēma medicīnas māsu apmācībai [HOS 2003], FLUTE sistēma formālo valodu un automātu apmācībai [DEV 2000], MATHTUTOR [POS 2004] un AlgeBrain [ALP 1999] sistēmas dažādu matemātikas jautājumu apmācībai un Passive Voice Tutor [VIR 2001] angļu valodas apmācībai.

Lielākā daļa minēto IMS-u ir realizētas, balstoties uz modulāru arhitektūru, kas sastāv no trim moduļiem, kurus parasti sauc par „tradicionālo trīsvienību” [SMI 1998], [GRU 2005]:

- **Eksperta modulis** iekļauj eksperta zināšanas un problēmu risināšanas raksturojumus. Moduļa uzdevums ir ar mācību priekšmetu saistītu problēmu un uzdevumu risināšana. Tas kalpo par standartu, ar kuru tiek salīdzinātas apmācāmā zināšanas un darbība [ANO 2007].
- **Studenta diagnosticēšanas modulis** vāc informāciju par studenta zināšanām un kļūdainiem priekšstatiem par problēmsfēru [SMI 1998].
- **Pedagoģiskais modulis** satur mācību stratēģijas un svarīgākās instrukcijas. Šī moduļa rīcībā esošās stratēģijas ir jāpielāgo katra studenta vajadzībām bez cilvēka iejaukšanās. Moduļa mērķis ir samazināt starpību starp studenta un eksperta zināšanām līdz minimumam vai to vispār novērst [SMI 1998].

Mūsdienīgām IMS-ām izmanto arī ceturto komponenti – **komunikāciju moduli**, kas ir atbildīgs par komunikāciju ar apmācāmo. Nodaļā doti visu moduļu veicamie uzdevumi.

Balstoties uz pasniedzēja veiktajām darbībām mācību procesā, ir analizētas IMS-ai veicamās darbības, lai IMS spētu aizstāt cilvēku-pasniedzēju un realizēt adaptīvu apmācību. Rezultātā ir secināts, ka ideālā gadījumā IMS-ām visas darbības ir jāadaptē apmācāmajam. Katras darbības adaptīvai veikšanai ir nepieciešams realizēt savus intelektuālus mehānismus. IMS-ās vajadzīgie intelektuālie mehānismi ir salīdzināti ar intelektuālu aģentu īpašībām un secināts, ka aģentu īpašības labi atbilst IMS-u vajadzībām (skat. 1.1. tabulu). Līdz ar to intelektuāli aģenti un daudzāģentu sistēmas ir piemērota tehnoloģija IMS-u realizēšanai.

Pēdējo gadu laikā aizvien vairāk IMS-u tiek realizētas, izmantojot aģentus. Lielākoties šīs sistēmas izmanto vienu pieeju – iepriekš aprakstīto IMS-as veidojošo moduļu realizāciju aģentu veidā. Grundspenķis un Anohina [GRU 2005] ir piedāvājuši aģentu kopu, kas realizē katru moduli. Moduļus realizējošie aģenti un to veicamie uzdevumi ir doti 1.2. tabulā.

1.1. tabula

Aģentu īpašību atbilstība IMS-ām

| Aģentu īpašības | Vajadzības IMS-ās |
|--------------------------------|---|
| Reaktivitāte | IMS-ai visu laiku jānovēro apmācāmā darbības. |
| Proaktivitāte | IMS-ai var būt nepieciešams iejaukties mācību procesā arī gadījumā, ja apmācāmais neveic nekādas darbības. |
| Spēja komunicēt un sadarboties | IMS-ās komponentes atbild par dažādiem aspektiem, un tādēļ tām ir jāspēj komunicēt un sadarboties, lai sasniegtu mācību mērķi. IMS-ās aktuāla ir nepieciešamība pēc komponentu integrācijas un sadarbības starp dažādām realitātēm. |
| Spēja spriest | IMS-ām ir jāpieņem dažādi lēmumi, balstoties uz zināšanām par apmācāmo, problēmsfēras zināšanām un pedagoģiskajām zināšanām. |
| Spēja apmācīties | Ne visi IMS-ām vajadzīgie intelektuālie mehānismi ir iebūvējami sistēmu izstrādes laikā, tādēļ tām jāapmācās vai jāadaptējas mācību procesa gaitā. |
| Daudzaģentu sistēmu atvērtība | IMS-ām ir nepieciešama atvērta un modulāra arhitektūra, lai izmaiņu ieviešana tajās būtu pēc iespējas vienkāršāka. |

1.2. tabula

IMS-u realizēšanai izmantojamo aģentu veicamie uzdevumi

| Aģents | Uzdevumi |
|--|--|
| Studenta diagnosticēšanas modulis | |
| Zināšanu novērtēšanas aģents | Apmācāmā zināšanu un prasmju novērtēšana |
| Psiholoģiskais aģents | Apmācāmā psiholoģisko raksturojumu noteikšana |
| Kognitīvās diagnosticēšanas aģents | Apmācāmā kļūdu un to iemeslu noteikšana |
| Mijiedarbības reģistrēšanas aģents | Mijiedarbības ar sistēmu reģistrēšana |
| Pedagoģiskais modulis | |
| Mācību stratēģiju aģents | Mācību materiālu ģenerēšana |
| Mācību plāna aģents | Mācību plāna veidošana, novērtēšana un izmainīšana |
| Atgriezeniskās saites aģents | Atgriezeniskās saites ģenerēšana |
| | Iejaukšanās mācību procesā un mājienu sniegšana |
| | Ieteikumu, kāda apmācāmā zināšanu un nepareizo priekšstatu korekcija ir jāveic, ģenerēšana |
| Uzdevumu ģenerēšanas aģents | Kontroles uzdevumu, problēmu un jautājumu ģenerēšana |
| Eksperta modulis | |
| Eksperta aģents | Problēmsfēras uzdevumu risināšana |
| Komunikāciju modulis | |
| Interfeisa aģents | Komunikācija ar apmācāmo |
| | Lietotāja saskarnes realizēšana |
| | Apmācāmā darbību uztveršana |
| | Komunikācija ar pasniedzēju |
| Animēts pedagoģiskais aģents | Visi interfeisa aģenta uzdevumi |
| | Emociju attēlošana un apmācāmā motivēšana |

Papildus IMS-u moduļu veidojošo aģentu kopai, ir zināmas vairākas IMS-u izstrādes arhitektūras, kas pielāgo un paplašina aplūkoto aģentu kopu, piemēram, daudzāģentu

arhitektūra attālinātām mācību sistēmām [DOR 2003], IVET arhitektūra [DEA 2005], ABITS arhitektūra [CAP 2000a], JADE arhitektūra mācību sistēmu izstrādei [SIL 2002] un X-GENITOR ietvars [TRI 2004]. Šīs arhitektūras atvieglo IMS-u veidojošu daudzāģentu sistēmu projektēšanu, piedāvājot konkrētus āģentus, to veicamos uzdevumus un mijiedarbību.

Neskatoties uz to, ka āģentu izmantošana IMS-u moduļu realizācijai atvieglo intelektuālo mehānismu ieviešanu un uzlabo modularitāti, āģentos sakņotu IMS-u izstrādē ir identificētas šādas problēmas:

- Izstrādātās daudzāģentu sistēmas, kas izmantotas IMS-u realizācijai, nav atvērtas. Taču atvērtība ir nepieciešama, lai varētu IMS-ām pievienot jaunu funkcionalitāti, nemainot esošos āģentus. Jāpiezīmē, ka daļa no minētajām arhitektūrām (IVET, JADE un X-GENITOR) tiek uzskatītas par atvērtām, bet tām var pievienot tikai viena veida āģentus – jaunam apmācāmajam atbilstošu interfeisa āģentu.
- IMS-as tiek veidotas no lielizmēra āģentiem, kam ir vairāki uzdevumi un to izstrāde ir sarežģīta salīdzinājumā ar nelieliem āģentiem.
- Lielizmēra āģentu atkārtota lietošana ir ierobežota, jo ir visai maz sistēmu, kurās ir iespējams atkārtoti izmantot visu āģenta funkcionalitāti.

Tādējādi var secināt, ka pašreiz zināmās daudzāģentu arhitektūras IMS-u realizēšanai nespēj izmantot visas daudzāģentu tehnoloģijas priekšrocības, galvenokārt tās, ko sniedz daudzāģentu sistēmu atvērtība un augstā modularitāte. Tādēļ, ir nepieciešams izveidot atvērtu arhitektūru, kas sastāv no nelieliem āģentiem.

Lai arī ir zināms liels skaits āģentos sakņotu IMS-u un daudzāģentu arhitektūru IMS-u realizēšanai, šādas sistēmas nav kļuvušas par industriālu risinājumu. Kā secināts darbos [BOR 2005b] un [BER 2004], lai āģentu tehnoloģija kļūtu par industriālu risinājumu, ir nepieciešamas praktiski pielietojamas tehnoloģijas āģentu izstrādei. Svarīgākās no šādām tehnoloģijām ir āģentu izstrādes metodoloģijas un rīki, kā arī āģentu realizācijas platformas. Tādēļ ir nepieciešams atrast vai no jauna izstrādāt tieši IMS-ām piemērotu āģentu izstrādes metodoloģiju un realizācijas platformu.

Galvenie nodaļas secinājumi ir šādi:

- Plaši izmantotajām e-apmācības sistēmām ir nozīmīgi trūkumi: tās nespēj veikt individualizētu un uz apmācāmo centrētu apmācību. Tāpat šīm sistēmām nav pietiekams intelekts, lai tās spētu kaut vai daļēji aizstāt pasniedzēju. Minētos trūkumus novērš IMS-as, kuras spēj adaptēties katram apmācāmajam, piedāvājot tieši viņam piemērotu

mācību materiālu un/vai vingrinājumu, un/vai atgriezenisko saiti, un/vai tēmu secību u.c. veidu adaptāciju.

- IMS-as tiek veidotas ar mērķi imitēt cilvēku-pasniedzēju. Šim nolūkam ir jārealizē dažādi, lielākajā daļā adaptīvi intelektuāli mehānismi.
- IMS-as tradicionāli sastāv no četriem lieliem moduļiem, kas atvieglo to izstrādi salīdzinājumā ar monolītām IMS-ām. Taču, izstrādājot IMS-as ar plašu funkcionalitāti, aprakstītajai arhitektūrai ir būtiski trūkumi, jo tajās minētie moduļi kļūst par lielizmēra komponentēm, kuru izstrāde ir sarežģīta un atkārtota izmantošana praktiski neiespējama. Arī izmaiņu ieviešana lielizmēra moduļos un intelektuālu mehānismu realizēšana IMS-ā ir sarežģīta.
- Aģentu īpašību analīze pierāda, ka aģenti un daudzāģentu sistēmas ir piemērota tehnoloģija IMS-u izstrādei, jo veicina intelektuālu mehānismu realizēšanu un daļēji novērš izstrādes grūtības.
- Lielākā daļa no daudzāģentu sistēmām, kas izmantotas IMS-u izstrādei, ir slēgtas daudzāģentu sistēmas. Arī atvērtās arhitektūras IMS-u realizācijai neļauj vienkārši mainīt sistēmas funkcionalitāti. Tādējādi nav iespējams pilnībā izmantot atvērtu daudzāģentu sistēmu priekšrocības.
- Lai aģentos sakņotas IMS-as varētu izmantot kā industriālu risinājumu, ir nepieciešamas atbilstošas izstrādes metodoloģijas un rīki, kā arī aģentu realizācijas platformas.

2. Aģentos sakņotu intelektuālu mācību sistēmu izstrāde

Promocijas darba otrajā nodaļā veikta aģentos sakņotu IMS-u izstrādei nepieciešamo tehnoloģiju – daudzāģentu arhitektūru, aģentu realizācijas platformu un aģentu izstrādes metodoloģiju analīze.

2.1. Daudzāģentu sistēmu arhitektūras

Kā secināts darba pirmajā nodaļā, zināmās IMS-u specifiskās daudzāģentu arhitektūras nenodrošina pietiekamu IMS-u modularitāti un atvērtību, tādēļ veikta vispārīgo daudzāģentu arhitektūru analīze ar mērķi novērtēt to pielietojamību IMS-u izstrādei. Nodaļā analizētas trīs daudzāģentu arhitektūras – holoniskas daudzāģentu sistēmas [FIS 2003], daudz-daudzāģentu sistēmas [NIM 2004] un AgentCities aģentu tīkls [DAL 2002]. Pēdējā no šīm arhitektūrām atzīta par nepiemērotu, jo tā piedāvā tikai atsevišķu daudzāģentu sistēmu integrāciju, nevis organizētu struktūru.

Holonisku daudzāģentu sistēmu piemērotību IMS-ām pamato to atbilstība visiem galvenajiem šīs arhitektūras autoru piedāvātajiem kritērijiem, kas izmantojami, lai noteiktu, vai problēmsfēra ir holoniska [GER 1999]. Galvenie kritēriji ir: (1) Hierarhiska struktūra. IMS-as sastāv no moduļiem, kas sastāv no aģentiem, ko var veidot no apakšāģentiem, utt. (2) Operatoru abstrakcija. Katram IMS-as modulim un aģentiem atbilst savi uzdevumi, kas sastāv no apakšuzdevumiem. (3) Daļējas dekompozīcijas iespējas. Daļa IMS-ās veicamo uzdevumu ir sadalāmi sīkāk, bet daļai ir nepieciešams viens pārējiem aģentiem zināms veicējs. (4) Kopīgi mērķi. IMS-ām ir viens kopīgs mērķis – veikt apmācību.

Salīdzinot holoniskas daudzāģentu sistēmas un daudz-daudzāģentu sistēmas, secināts, ka to pamatidejas ir līdzīgas, taču to atšķirības ļauj secināt, ka holoniskas daudzāģentu sistēmas ir piemērotākā arhitektūra IMS-u izstrādei. Galvenās atšķirības starp abām arhitektūrām ir šādas:

- Holoniskās daudzāģentu sistēmās holona galva var veikt ne tikai starpnieka funkcijas, bet arī citu aģentu darbību koordinēšanas un vadīšanas uzdevumus, kas ļauj izmantot holonu galvas aģentus par vadošajiem aģentiem IMS-u moduļos.
- Daudz-daudzāģentu sistēmās mijiedarbība starp atsevišķām daudzāģentu sistēmām ir vāja, turpretim IMS-u moduļi nav vāji saistīti.
- Holoniskas daudzāģentu sistēmas ļauj veidot vairāk nekā divus hierarhijas līmeņus, kas nodrošina hierarhisku uzdevumu dekompozīciju praktiski vajadzīgajā detalizācijas līmenī.
- Holoni var tikt dinamiski mainīti, realizējot IMS-u atvērtību un izmaiņu ieviešanu tajās.

2.2. Aģentu realizācijas platformas

Aģentu realizācijas platformas tiek veidotas, lai aģentu projektiem nepieciešamā infrastruktūra būtu atkārtoti izmantojama un katra projekta ietvaros to nebūtu jāizstrādā no jauna. Darbā analizētas šādas platformas: JADE [BEL 2007], JADEX [POK 2005], Jason [BOR 2005a], ZEUS [COL 2000], JACK [WIN 2005], AgentBuilder [AGE], ABLE [BIG 2002], FIPA OS [LAU 2001], [BUC 2002], Aglets [CLE 1997], RETSINA [SYC 2001], JatLite [JEO 2000] un IMPACT [DIX 2005]. Platformu piemērotība IMS-u izstrādei analizēta, izmantojot šādus kritērijus:

- Heterogēnu aģentu atbalsts. Platformai jāļauj ērti realizēt heterogēnus aģentus.
- Ontoloģiju izmantošana. Lai aģenti varētu efektīvi apmainīties ar zināšanām, ir jāizmanto ontoloģijas.
- Atbilstība FIPA standartiem. Lai varētu realizēt atvērtas sistēmas, platformai jāatbilst FIPA standartiem [BEL 2007], [WIL 2004].

- Balstīšanās programmēšanas valodā. Lai būtu iespējams izveidot rīku, kas realizē visu izstrādes procesu un no projektējuma ģenerē sistēmas kodu, ir nepieciešams, lai aģentu izstrādi būtu iespējams veikt, balstoties uz kodu.
- Aģentu vadības rīku esamība, kas atvieglo izstrādi.
- Aģentu testēšanas rīku pieejamība, kas nodrošina aģentorientētai programmatūrai atbilstošu testēšanas procesu.
- Brīva pieejamība un popularitāte, kas nodrošina to, ka, sākot lietot platformu, nebūs lieku izdevumu saistītu ar tās apgūšanu un iegādi. Platformu popularitāte analizēta, balstoties uz literatūrā pieejamo informāciju par tajās realizētajiem projektiem.
- Attīstības turpināšanās, kas nodrošina to, ka jaunākās tendences aģentu izstrādē un jauna funkcionalitāte arī turpmāk tiks iekļautas platformā.

2.1. tabulā apkopota platformu atbilstība izvirzītajiem kritērijiem, izmantojot šādus apzīmējumus: pilna atbilstība kritērijam ir apzīmēta ar „+”, daļēja atbilstība ir apzīmēta ar „+/-”, „-” nozīmē, ka platforma kritērijam neatbilst, bet tukša tabulas šūna nozīmē, ka informācija par rīka atbilstību kritērijam nav pieejama. No tabulas var secināt, ka JADE platforma vislabāk atbilst izvirzītajiem kritērijiem, tādēļ tā ir vispiemērotākā aģentos sakņotu IMS-u izstrādei.

2.1. tabula

Aģentu izstrādes rīku piemērotība IMS izstrādes metodoloģijai

| Platforma | Kritēriji | | | | | | | | |
|--------------|-------------------|-------------|----------------|------------------------|---------------------|-----------------|----------|----------------|----------------------|
| | Heterogēni aģenti | Ontoloģijas | FIPA standarti | Progr. valodā sakņotas | Aģentu vadības rīki | Testēšanas rīki | Populārs | Brīvi pieejams | Turpinās attīstīšana |
| ABLE | + | - | - | + | - | - | - | + | - |
| Aglets | + | - | - | + | | | | + | - |
| AgentBuilder | - | - | - | - | + | - | - | - | - |
| FIPA OS | + | + | + | + | +/- | - | - | + | - |
| IMPACT | +/- | | +/- | +/- | + | + | - | - | - |
| JACK | - | - | +/- | +/- | + | + | + | - | + |
| JADE | + | + | + | + | + | + | + | + | + |
| JADEx | - | + | + | + | + | +/- | +/- | + | + |
| JatLite | + | +/- | +/- | + | | | +/- | + | - |
| Jason | - | - | - | + | + | + | +/- | + | + |
| RETSINA | +/- | + | - | + | + | +/- | +/- | - | |
| ZEUS | - | + | + | + | + | +/- | +/- | + | - |

2.3. Aģentu izstrādes metodoloģijas

Promocijas darbā ir secināts, ka aģentos sakņotām IMS-ām ir specifiskas īpašības un tām ir veltīti pētījumi, kuru rezultāti izstrādes laikā ir jāņem vērā. Līdz ar to, metodoloģijā, ko izmanto IMS-u izstrādei, ir jāievēro, ka:

- IMS-as ir vāji integrētas organizācijā. Tas nozīmē, ka nav pielietojamas nekādas organizācijas analīzē sakņotas metodes.
- Nepieciešams atbalsts IMS-ām atbilstošiem aģentiem un daudzāģentu sistēmām, t.i., reaktīviem, proaktīviem, loģiskiem aģentiem un atvērtām un sadarbojošām daudzāģentu sistēmām.
- IMS-u moduļi un līdz ar to aģenti tiek veidoti atbilstoši zināšanu veidiem, ko IMS-as izmanto, nevis atbilstoši lietotājiem, organizācijām, lietošanas gadījumiem vai funkcionalitātēm.
- Daudzos pētījumos ir iegūtas dažādas zināšanas, kas atvieglo IMS izstrādi. Svarīgākie no pētījumu rezultātiem ir IMS-u arhitektūra, aģentos sakņotu IMS-u realizācijai definētā aģentu kopa un katra šīs kopas aģenta veicamie uzdevumi. Visus šos pētījumu rezultātus ir nepieciešams integrēt metodoloģijā.

Analizējot aģentorientētas programmatūras izstrādes metodoloģijas, ir secināts, ka tās lielākoties ir veidotas, balstoties uz aģentiem līdzīgus pētījumu virzienos izmantotām metodoloģijām. Dažādos pētījumu virzienos sakņotās metodoloģijas ievērojami atšķiras, tādēļ promocijas darbā, balstoties uz Vuldridža [WOO 2000] piedāvāto klasifikāciju, ir izveidota metodoloģiju klasifikācija pēc to izcelsmes, kurā ietilpst šādas klases:

- Metodoloģijas, kas cēlušās no objektorientētām pieejām, piedāvā objektorientētās metodoloģijas paplašināt vai adaptēt, lai tās būtu iespējams sekmīgi izmantot aģentu izstrādē. Šī ir visplašākā metodoloģiju klase, kurā ietilpst AURL [ODE 2000], MESSAGE [CAI 2001], INGENIAS [PAV 2005], OPEN procesu ietvars [DEB 2002], MaSE [DEL 2001], PASSI [BUR 2002], RAP [TAV 2005], AOMT [STU 2002] un ADELFE [BER 2005].
- No zināšanu inženierijas cēlušās metodoloģijas, kas piedāvā modelēšanas valodas, kuras ir piemērotas sistēmas struktūras un funkciju verifikācijai. Šai grupai pieder Tropos [BRE 2002], OPEN procesu ietvars [DEB 2002], MAS-CommonKADS [IGL 1998], DESIRE [BRA 1997] un COMOMAS [GLA 1998].
- Formālās valodās balstītas metodoloģijas. Šai grupai no darbā analizētajām pieder DESIRE metodoloģija [BRA 1997].

- Organizācijas projektēšanā balstītas metodoloģijas. Šīs klases metodoloģijas galveno uzvaru liek uz organizācijas modelēšanu, kurā ir jāiekļaujas izstrādājamai sistēmai. Šai klasei pieder Styx [BUS 2002], metodoloģija daudzāģentu sistēmu izstrādei organizācijas integrācijai [KEN 1995] un ARCHON ietvars [VAR 1994].
- Speciāli aģentu izstrādei izveidotās metodoloģijas, kas pamatojas uz mākslīgā intelekta atziņām par intelektuāliem aģentiem. Šādas metodoloģijas ir Gaia [WOO 2000], Cassiopeia [COL 1996], augsta līmeņa un vidēja projektējuma pieeja [ELA 1999], BDI aģentu modelēšana [KIN 1997], MASSIVE [LIN 2000] un Prometheus [PAD 2005].

Analizējot vispārīgās aģentorientētas programmatūras izstrādes metodoloģijās realizētos dzīves ciklus, secināts, ka tikai dažas no metodoloģijām (MESSAGE [CAI 2001] un ADELFE [PIC 2004]) izmanto precīzi definētus dzīves ciklus, kas ir RUP modifikācijas [RUP]. Pārējās metodoloģijas precīzi nedefinē izmantoto dzīves ciklu. Lielākā daļa no tām realizē dažādus iteratīvus ūdenskrituma dzīves cikla modeļus. Promocijas darbā, pamatojoties uz Seidža [SAG 1992] aplūkotajiem ūdenskrituma modeļiem, ir secināts, ka ūdenskrituma modeļi ir pielāgojami dažādām sistēmām. Tādēļ, balstoties uz metodoloģijās realizētajiem izstrādes procesiem, ir definēts aģentorientētas programmatūras izstrādes dzīves cikls, kas ir ūdenskrituma modeļa modifikācija. Izstrādātais dzīves cikla modelis ir iteratīvs un sastāv no šādām fāzēm: analīze, aģentu ārējā projektēšana, aģentu iekšējās funkcionalitātes projektēšana, realizēšana, testēšana, ieviešana un uzturēšana. Projektēšanas fāze sadalīta divās, jo aģentu projektēšana parasti tiek veikta, vispirms projektējot to uzvedību un/vai funkcijas, un pēc tam projektējot to iekšējo struktūru. Rezultātā tiek noteikts, kā aģenti sasniegs savu funkcionalitāti. Definētais dzīves cikla modelis dots 2.1. attēlā.

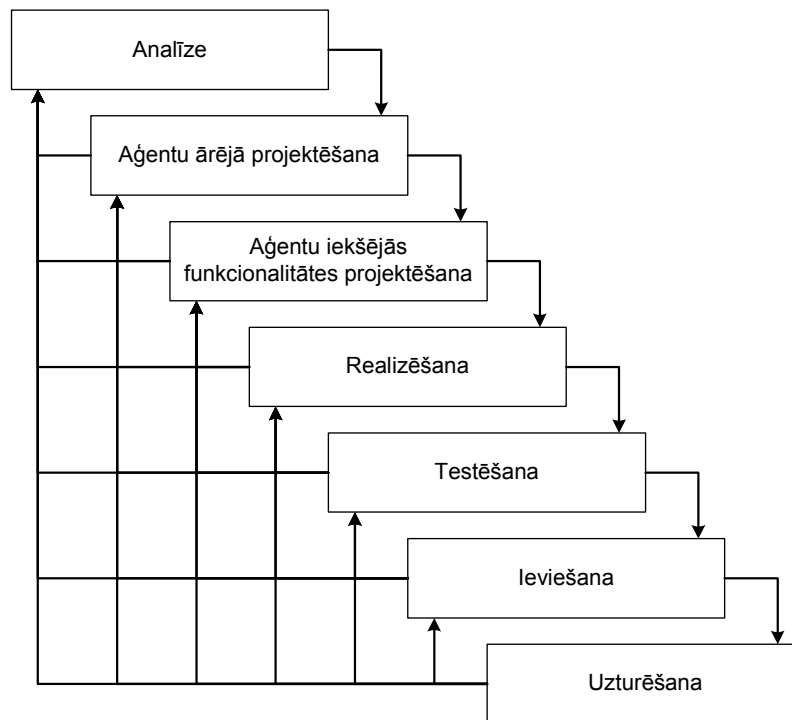
Katrā no aģentorientētas programmatūras izstrādes dzīves cikla fāzēm ir veikta tajā izmantoto tehniku analīze un izdarīti secinājumi par tehniku piemērotību IMS-u izstrādei. Tāpat ir secināts, ka trūkst visu dzīves ciklu atbalstošas metodoloģijas, jo lielākā daļa no tām neatbalsta visu dzīves ciklu. Daļa metodoloģiju vāji atbalsta realizēšanas fāzi, bet īpaši vāji atbalstītas ir uzturēšanas, testēšanas un ieviešanas fāzes.

Metodoloģiju piemērotības IMS-u izstrādei analīzes rezultātā ir izvirzītas prasības un ir secināts, ka neviena metodoloģija pilnībā tām neatbilst. Tādēļ ir nepieciešams izstrādāt specifisku aģentos sakņotu IMS-u izstrādes metodoloģiju.

Galvenie otrās nodaļas secinājumi ir šādi:

- JADE platforma ir vispiemērotākā, lai veidotu tai atbilstošu aģentos sakņotu IMS-u izstrādes metodoloģiju un rīku, jo šī platforma ļauj vienkārši realizēt daudzāģentu sistēmās sakņotas IMS-as, iekļauj nepieciešamos aģentu izstrādes rīkus, atbilst FIPA

standartiem un ir gan populāra (un līdz ar to izstrādātājiem zināma), gan brīvi pieejama, gan tiek turpināta tās izstrāde.



2.1. att. Aģentorientētas programmatūras izstrādes dzīves cikls

- Aģentos sakņotām IMS-ām ir specifiskas īpašības, kas ietekmē to izstrādes procesu. Bez tam šī tipa IMS-as ir atsevišķs pētījumu virziens, kurā ir veikti intensīvi pētījumi un iegūti nozīmīgi rezultāti, kas var atvieglot IMS-u izstrādi. Taču neviena no zināmajām aģentorientētas programmatūras izstrādes metodoloģijām neņem vērā šo pētījumu rezultātus un neļauj efektīvi izstrādāt sistēmas ar tādām specifiskām īpašībām, kādas piemīt IMS-ām. Tādēļ ir jāizstrādā specifiska aģentos sakņotu IMS-u izstrādes metodoloģija, kas ļautu izmantot IMS-u pētījumu rezultātus un adekvāti aprakstītu sistēmas ar tādām īpašībām, kādas ir IMS-ām.
- Lielākā daļa aģentorientētas programmatūras izstrādes metodoloģiju precīzi nedefinē izmantoto dzīves ciklu. Taču, pētot tajās realizēto izstrādes procesu, var secināt, ka tajā tiek izdalītas ūdenskrituma modelim atbilstošas fāzes, un tas lielākoties ir iteratīvs. Pamatojoties uz šo secinājumu, ir izveidota ūdenskrituma modeļa modifikācija aģentorientētas programmatūras izstrādei. Nozīmīgākā izmaiņa šajā modelī ir tā, ka projektēšanas fāze parasti tiek sadalīta divās stadijās. Pirmajā no šīm stadijām projektē daudzāģentu sistēmu, specificējot, no kādiem aģentiem tā sastāvēs, kāda būs katra aģenta funkcionalitāte un kāda būs mijiedarbība starp aģentiem. Savukārt, otrajā stadijā projektē

aģentu iekšējo struktūru, respektīvi, tiek specificēts, kādā veidā aģenti veic savas funkcijas.

- Lielākā daļa metodoloģiju neatbalsta visu dzīves ciklu, jo tās neaplūko visas dzīves cikla fāzes. Daļa metodoloģiju vāji atbalsta realizēšanas fāzi, bet īpaši vāji atbalstītas ir uzturēšanas, testēšanas un ieviešanas fāzes. Metodoloģijai, kas atbalsta visu dzīves ciklu, ir priekšrocība, jo tās lietotājam nav vajadzīgs meklēt nekādus citus papildus rīkus vai kādu fāzi veikt, izmantojot tikai savas prasmes.
- Katrā aģentorientētas programmatūras izstrādes dzīves cikla fāzē ir iespējams pielietot dažādas tehnikas. Taču tikai neliela daļa no tām ir piemērota IMS-u izstrādei, jo IMS-as ir specifiskas sistēmas ar nozīmīgām atšķirībām no citām aģentorientētām sistēmām. Līdz ar to, liela daļa tehniku vai nu apraksta neinformatīvus IMS-u aspektus (piemēram, organizāciju), vai nespēj ņemt vērā IMS-u pētījumu rezultātus (piemēram, IMS-u veidojošo aģentu kopu). Tādēļ, veidojot metodoloģiju aģentos sakņotu IMS-u izstrādei, ir nepieciešams tajā iekļaut tikai tādas tehnikas, kas ir izmantojamas IMS-u izstrādei un ir savstarpēji savietojamas.

3. Atvērta daudzāģentu arhitektūra intelektuālas mācību sistēmas realizācijai

Darba trešajā nodaļā izstrādāta holoniska daudzāģentu arhitektūra IMS-u realizēšanai, kurā IMS no ārpuses ir viens holons. Šo holonu pārstāv interfeisa aģents, kas tādējādi ir holona galva, un ir vienīgais, kurš realizē mijiedarbību ar studentu un visas komunikāciju moduļa sastāvdaļas.

Pārējie moduļi tiek realizēti apakšholonu veidā un ir iekļauti holona ķermenī. Katram modulim ir definēti viens vai vairāki holoniski aģenti, kas to realizē. Studenta diagnosticēšanas moduli veido studenta modelēšanas aģents un zināšanu novērtēšanas aģents. Pedagoģisko moduli veido mācību plāna aģents, mācību stratēģijas aģents, uzdevumu ģenerēšanas aģents un atgriezeniskās saites aģents. Eksperta modulis ir realizēts ar vienu aģentu – eksperta aģentu. Augstākā līmeņa apakšholoniem promocijas darbā ir definēti veicamie uzdevumi, piemēram, eksperta aģentam jārisina vingrinājumi un problēmas attiecīgajā problēmsfērā, kā arī jānosūta risinājumi atgriezeniskās saites aģentam un zināšanu novērtēšanas aģentam.

Izveidotās arhitektūras augstākajā līmenī aģentu mijiedarbība ir realizēta kā atsevišķu ziņojumu nosūtīšana. Netiek definēti un izmantoti sarežģītāki protokoli, jo mijiedarbība sastāv tikai no trīs veidu ziņojumiem: servisa pieprasījuma, servisa izpildes rezultāta paziņošanas un

informatīva ziņojuma. Promocijas darbā dota aģentu pazīšanās diagramma, kas norāda, starp kuriem aģentiem eksistē mijiedarbība, un mijiedarbības diagramma, kas satur visu nosūtāmo informāciju starp aģentiem.

Visi augstākā līmeņa aģenti var tikt realizēti kā holoni, kas sastāv no viena galvas aģenta un neierobežota skaita apakšholoniem. Galvas aģents pārstāv holonu ārpus tā robežām, veic ķermeņa aģentu koordināciju, izmantojot centralizētu plānošanu un uzdevumu piešķiršanu aģentiem. Tam ir jāzina visu holonā ietilpstošo aģentu spējas, lai veiktu uzdevumu sadali un plānošanu. Šādiem mērķiem izmanto globālu dzelteno lapu servisa aģentu, kurš zina visu aģentu piedāvātos servisos. Papildus jau minētajiem galvas aģentu uzdevumiem tiem jāveic arī holona ķermeņa aģentu iegūto rezultātu apkopošana un gala rezultāta noteikšana.

Izstrādātā arhitektūra ir atvērta, un tajā ir divu veidu holoni: slēgti holoni un atvērti holoni. Slēgtos holonos ietilpst tikai konkrēti aģenti. Atvērtos holonos ietilpst galvas aģents un noteikta veida ķermeņa aģenti, taču ķermeņa aģentu skaits un to precīzas funkcijas ir brīvi maināmas sistēmas izstrādes un darbības laikā. Atvērtos holonos katrs holona ķermeņa aģents veic konkrētu holonam veicamā uzdevuma veidu. Atvērta holona ķermeņa aģenti ir brīvi pievienojami un izslēdzami no sistēmas, tādējādi mainot sistēmas funkcionalitāti.

Holona ķermeņa aģenti veic konkrētus uzdevumus, kas ir holona veicamo uzdevumu apakšuzdevumi. Slēgtu holonu ķermeņa aģenti parasti veic principiāli atšķirīgus uzdevumus. Turpretim atvērtos holonos ķermeņa aģenti veic viena un tā paša uzdevuma veidus. Līdz ar to ir iespējams pievienot jaunu aģentu, kas realizē vēl kādu uzdevuma veidu. Piemēram, katrs atvērta holona aģents var ģenerēt sava veida mācību materiālu. Tādā gadījumā holonam var pievienot jauna materiāla veida ģenerēšanas aģentu. Katram šādam holona ķermeņa aģentam, sākot darbu, ir pienākums pierēģistrēt savus servisos pie dzelteno lapu servisa aģenta. Piedāvātā arhitektūra ļauj holona ķermenī ietilpstošos aģentus realizēt nākamā līmeņa holonu veidā. Tādējādi, tā pieļauj neierobežota dziļuma hierarhijas veidošanu.

Promocijas darbā dots tipisks atvērta holona galvas aģenta algoritms. Atvērtā holonā mijiedarbība notiek šādā veidā. Vispirms holona galvas aģents saņem pieprasījumu veikt kādu uzdevumu. Tad galvas aģents atrod visus ķermeņa aģentus, izmantojot dzelteno lapu servisu, un nosūta tiem pieprasījumus. Ķermeņa aģenti veic savus uzdevumus un to rezultātus nosūta galvas aģentam, kas izvēlas piemērotāko no rezultātiem pēc problēmsfērā sakņotiem kritērijiem un nosūta to atbilstošajiem augstākā līmeņa aģentiem.

Atvērti pirmā līmeņa holoni nevar nodrošināt pilnībā atvērtu arhitektūru, jo jauna sistēmas funkcionalitāte ir jāsniedz arī tās lietotājam, ko veic interfeisa aģents jeb pirmā

līmeņa holona galvas aģents. Tādēļ ir izveidots holoniskas daudzāģentu sistēmas paplašinājums, kurš paredz pirmā līmeņa holona galvu (interfeisa aģentu) realizēt atvērta holona veidā. Rezultātā, interfeisa aģenta holona galvas aģents mijiedarbojas ar pārējiem aģentiem un tam ir fiksētas spējas mijiedarbībai ar lietotāju.

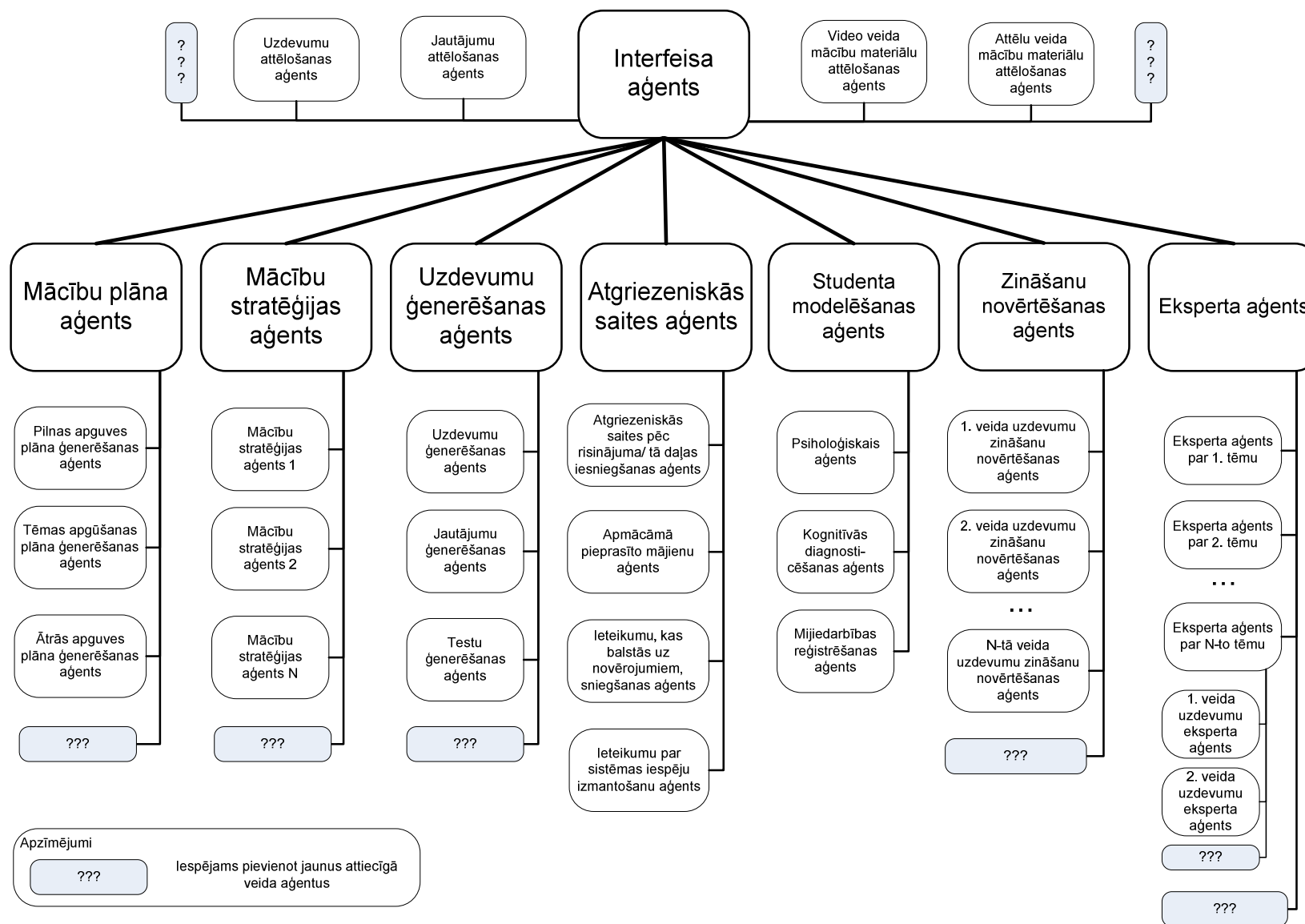
Kad nepieciešams veikt mijiedarbību ar lietotāju, interfeisa aģenta otrā līmeņa holona galvas aģents rīkojas pēc šāda algoritma: ja pašam galvas aģentam ir nepieciešamās spējas veikt mijiedarbību, tad tā tiek veikta. Pretējā gadījumā tiek izmantots dzelteno lapu serviss un holonā meklēts aģents, kurš spētu veikt konkrēto mijiedarbību. Atrastajam aģentam nosūta ziņojumu ar pieprasījumu veikt mijiedarbību.

Izstrādātā arhitektūra ir augsta līmeņa arhitektūra, ko var pielāgot dažādu IMS-u izstrādes vajadzībām. Promocijas darbā aplūkotos holonus var realizēt vienkāršu aģentu veidā un otrādi. Respektīvi, aprakstītā arhitektūra uzskatāma par karkasu, ko iespējams izmantot gan bez izmaiņām, gan veicot izmaiņas tajā, lai vienkāršotu vai detalizētu kādu sistēmas daļu.

3.1. attēlā dota atvērtās arhitektūras uzbūve ar paplašināšanas mehānismiem. Šajā attēlā doti visi augstākā līmeņa holoni. Bez tam visiem slēgtiem holoniem parādīti visi apakšholoni, bet atvērtiem holoniem pievienoti daži aģentu piemēri un norādīts, ka ir iespējams vēl pievienot jaunus attiecīgā veida aģentus (to norāda pelēkie taisnstūri ar „???”).

Trešās nodaļas ietvaros izstrādātā arhitektūra publicēta darbā [LAV 2008]. Galvenie nodaļas secinājumi ir šādi:

- Izveidotā arhitektūras koncepcija rekomendē holonisko aģentu izmantošanu IMS-u izstrādei.
- Arhitektūras atvērtība atvieglo izmaiņu ieviešanu strādājošā IMS-ā, jo lielu daļu izmaiņu ir iespējams ieviest tikai pievienojot vai izslēdzot aģentus no sistēmas.
- Holonu izmantošana atvieglo izmaiņu veikšanu IMS-ā, jo izmaiņas var ietekmēt aģentus tikai tajā holonā, kur tās tiek veiktas.
- Nodaļā ir piedāvāti holonu galvas aģentu darbības algoritmi, kas atvieglo galvas aģentu izstrādi.
- Ja IMS ir realizēta, izmantojot izstrādāto arhitektūru, tā sastāv no vairāk aģentiem, nekā sistēmas, kas realizē dažādas 1.4. nodaļā aplūkotās aģentu kopas modifikācijas. Līdz ar to izstrādājami aģenti ir mazāki un tos ir vieglāk izstrādāt. Nelielus aģentus, kas realizē vienu konkrētu servisu, ir arī vienkāršāk atkārtoti izmantot citās sistēmās.
- Holonu veidošana zemākos līmeņos dod iespēju pielietot dažādas aģentu mijiedarbības, piemēram, dažādus konfliktu risināšanas mehānismus.



3.1. att. Atvērta holoniska daudzāģentu arhitektūra intelektuālas mācību sistēmas realizācijai

4. MASITS metodoloģija un rīks

Ceturtnā nodaļa ir veltīta promocijas darbā izstrādātajai IMS-u izstrādes metodoloģijai MASITS, kas publicēta darbā [LAV 2009d], MASITS metodoloģijas automatizācijas algoritmiem, kas publicēti darbā [LAV 2009e] un MASITS rīkam, kas publicēts darbā [LAV 2009a].

4.1. MASITS metodoloģija

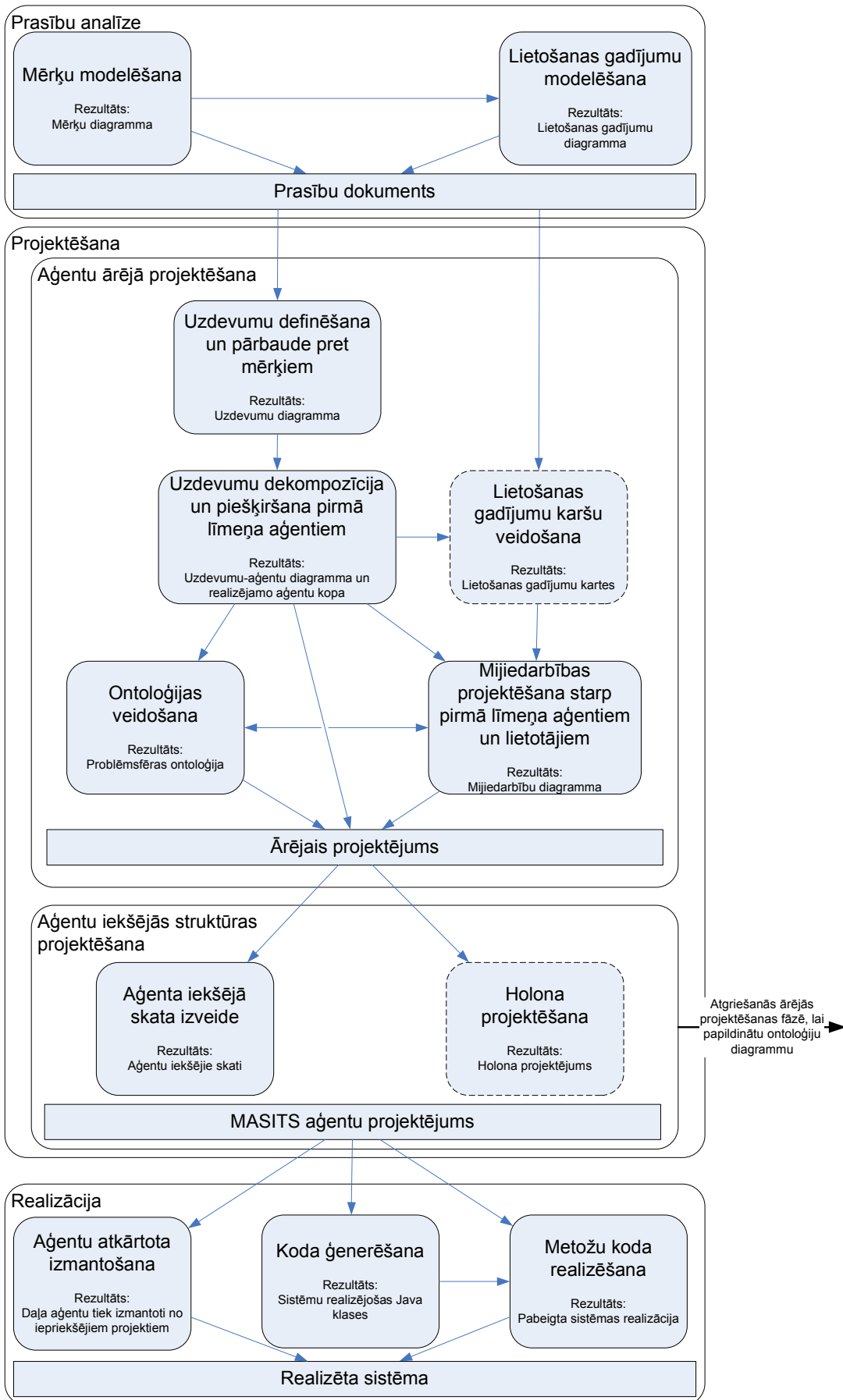
MASITS metodoloģija ir radīta kā specifiska aģentorientēta programmatūras izstrādes metodoloģija, kas paredzēta IMS-u izstrādei. Tā ir piemērota jebkuras sarežģītības sistēmām, jo izmanto holonisku daudzāģentu arhitektūru un atbalsta holonu projektēšanu, kas nodrošina iespēju projektēt jebkuras sarežģītības sistēmu, sadalot to pa atsevišķiem holoniem. Lai atvieglotu izstrādes procesu un izvairītos no papildus projektēšanas laikā izmantoto konceptu transformācijām izstrādes vidē izmantotajos konceptos, MASITS metodoloģija ir piesaistīta vienai konkrētai aģentu realizācijas platformai – JADE, kura darba 2.2 apakšnodaļā atzīta par vispiemērotāko IMS-u izstrādei. Līdz ar to aģentu iekšējā projektēšana notiek atbilstoši JADE platformas konceptiem.

Pamatojoties uz vispārīgo aģentorientētas programmatūras izstrādes metodoloģiju analīzi, MASITS ir veidota, no tām izmantojot piemērotākās tehnikas, kas ir papildinātas ar jaunām tehnikām un teorētiskām atziņām par aģentu tehnoloģiju pielietojumu IMS-u realizācijā.

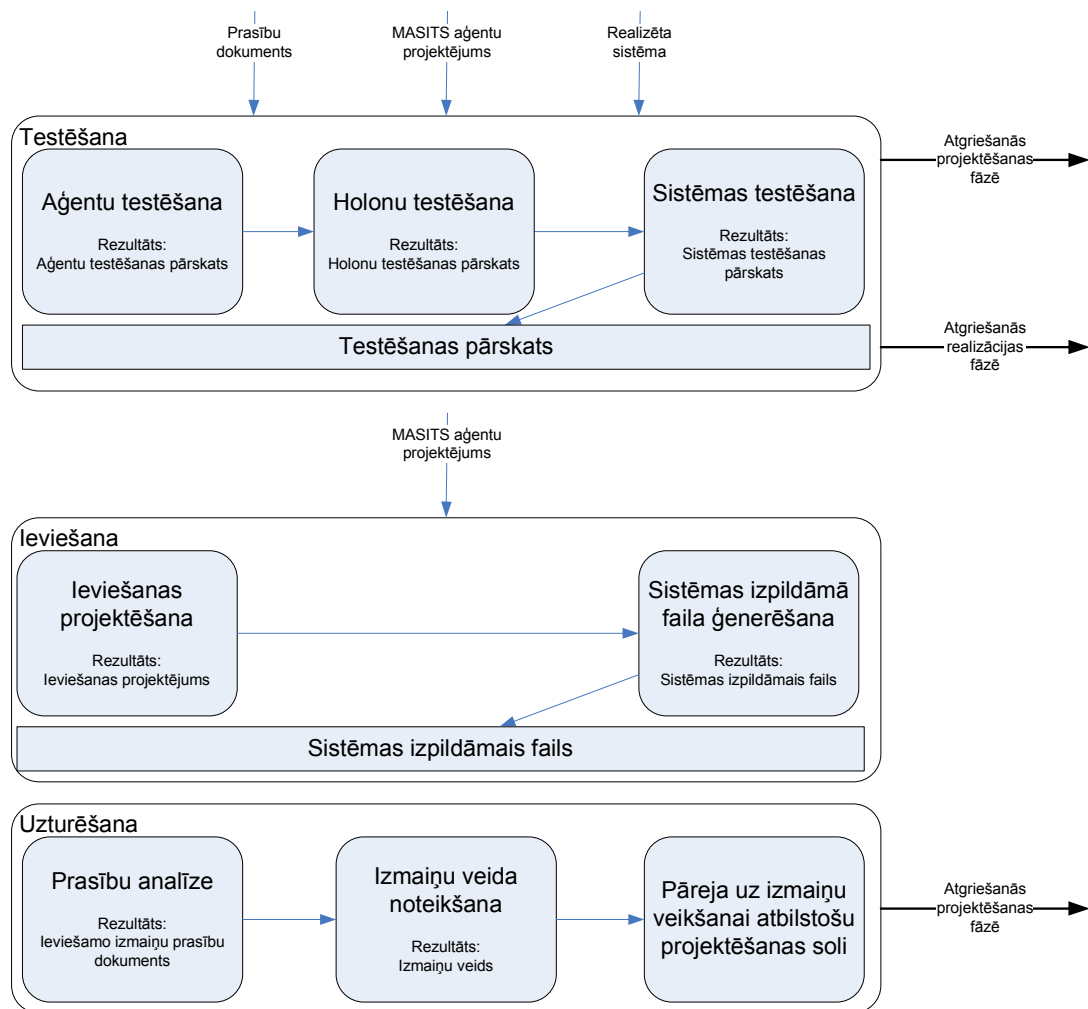
MASITS metodoloģija atbalsta visu 2. nodaļā definēto aģentorientētas programmatūras izstrādes dzīves ciklu. Metodoloģijas struktūra atspoguļota 4.1. un 4.2. attēlos, kur ar pārtrauktu līniju apzīmēti soļi, kas ir veicami tikai gadījumos, ja ir nepieciešama sīkāka analīze.

Promocijas darbā iekļautais metodoloģijas apraksts satur katras fāzes detalizētu aprakstu. Katrai fāzei ir aprakstīti nepieciešamie iepriekšējo soļu produkti, zināšanas un prasmes, veicamie soļi un tajos izmantotās tehnikas, kā arī fāzes rezultātā veidotie dokumenti. Katram izstrādes gaitā iekļautajam solim MASITS metodoloģijā dots tā veikšanai izmantojamais algoritms.

Prasību **analīzes fāzē** ir secīgi jāveic divi galvenie soļi: mērķu modelēšana un lietošanas gadījumu modelēšana. IMS-u prasību analīzes fāze aprakstīta darbā [LAV 2009b].



4.1. att. Metodoloģijas struktūra (analīzes, projektēšanas un realizācijas fāzes)



4.2. att. Metodoloģijas struktūras turpinājums (testēšanas, ieviešanas un uzturēšanas fāzes)

Mērķu modelēšanas solī vispirms izvirza vienu vai vairākus sistēmas mērķus un veic šo mērķu dekompozīciju līdz brīdim, kad zemākā līmeņa mērķi ir sasniedzami ar konkrētām aģenta darbībām. Katram mērķim veido aprakstu, kas ietver mērķa nosaukumu, īsu aprakstu, nosacījumu, kas nosaka, vai mērķis ir sasniegts, un aktieri, kam mērķis nepieciešams. Šajā solī tiek izveidota mērķu diagramma.

Lietošanas gadījumu modelēšanas laikā vispirms ir jāidentificē visi sistēmas aktieri un tad zemākā līmeņa mērķiem no mērķu diagrammas jāveido atbilstoši lietošanas gadījumi. Katram lietošanas gadījumam ir jāveido apraksts, kura ietvaros veido veiksmes scenāriju, identificē iespējamus izņēmumus un veido tiem atbilstošos scenārijus.

Projektēšanas fāze MASITS metodoloģijā ir iedalīta divās stadijās – **aģentu ārējā projektēšanā** un **aģentu iekšējās struktūras projektēšanā**. Vispirms veic aģentu ārējo projektēšanu, kuras laikā specificē aģentu ārējo funkcionalitāti un mijiedarbību. Savukārt, aģentu iekšējās struktūras projektēšanas laikā veido mehānismus, ar kuru palīdzību aģents

sasniedz nepieciešamo funkcionalitāti. Aģentos sakņotu IMS-u projektēšanas veikšana, izmantojot MASITS metodoloģiju, publicēta darbā [LAV 2009c].

Aģentu **ārējās projektēšanas stadijā**, izmantojot analīzes fāzē izveidotos mērķu un lietošanas gadījumu modeļus, veic šādus soļus:

- uzdevumu definēšana atbilstoši lietošanas gadījumiem,
- uzdevumu pārbaude attiecībā pret mērķiem,
- uzdevumu koka (dekompozīcijas) izveide un uzdevumu piešķiršana pirmā līmeņa aģentiem,
- aģentu mijiedarbības projektēšana, kas sastāv no šādiem soļiem:
 - mijiedarbības starp aģentiem veidošana un attēlošana mijiedarbības diagrammā,
 - gadījumos, kad mijiedarbība ir pārāk sarežģīta, lai to varētu uzreiz izveidot mijiedarbības diagrammā, vispirms jāizveido lietošanas gadījumu kartes,
 - mijiedarbības, kurās ir pietiekami daudz ziņojumu, lai tie mijiedarbības diagrammu padarītu neuzskatāmu, ir jādetalizē ar atsevišķu diagrammu,
 - sākotnējā ontoloģijas veidošana (nepieciešams veikt mijiedarbības projektēšanas laikā, lai varētu specificēt mijiedarbību saturu ar ontoloģijā iekļautajiem predikātiem).

Pirmajā solī notiek *uzdevumu definēšana* atbilstoši lietošanas gadījuma scenārijos iekļautajiem soļiem. Katram scenārijam jāizveido atbilstoši uzdevumi, kas to realizē. Taču vispārīgā gadījumā uzdevumu atbilstība scenārija soļiem nav tieša, t.i., katram solim neveido tieši vienu uzdevumu, bet vairāki soļi var tikt apvienoti vienā uzdevumā, kā arī, ja nepieciešams, vienam solim var tikt izveidoti vairāki uzdevumi.

Pēc uzdevumu definēšanas veic *lietošanas gadījumu pārbaudi pret mērķiem*, t.i., pārbaudi, vai visiem zemākā līmeņa mērķiem no mērķu diagrammas ir izveidoti atbilstoši uzdevumi, kuru veikšanas rezultātā tiks sasniegti visi mērķi. Ja kādam no mērķiem neatrod atbilstošu uzdevumu, tad ir jāpārskata izveidotie uzdevumi, kā arī atbilstošie lietošanas gadījumu scenāriji.

Nākamais solis ir *uzdevumu apvienošana hierarhijā un uzdevumu piešķiršana aģentiem*. Vispirms veido uzdevumu hierarhiju, apvienojot uzdevumus, kas ir līdzīgi no to veikšanas viedokļa un ir piešķirami vienam aģentam. Uzdevumu hierarhiju attēlo uzdevumu diagrammā, ko izmanto kā sagatavi uzdevumu-aģentu diagrammai, kas atšķiras ar to, ka tajā katram uzdevumam ir norādīts aģents. Uzdevumu-aģentu diagrammu veido, piešķirot uzdevumus aģentiem, ko realizē, izmantojot IMS-u pētījumos sakņotus likumus. Katram no 3.

nodaļā definētās arhitektūras augstākā līmeņa aģentiem ir izveidoti neformāli likumi teksta formā, kas apraksta, kādi uzdevumi tam jāpiešķir. Piemēram, studenta modelēšanas aģentam atbilstošais likums ir šāds:

„Studenta modelēšanas aģentam jāpiešķir:

- a) uzdevumi, kas saistīti ar informācijas uzskaiti un apstrādi par to, kam apmācāmais dod priekšroku apmācībā, mācīšanās stilu, emocijām mācību seansā u.c. tamlīdzīgi uzdevumi,
- b) uzdevumi, kas saistās ar apmācāmā darbību un sistēmas iespēju izmantošanas uzskaiti,
- c) pilna studenta modeļa veidošanas uzdevumi,
- d) studenta progresu mācību kursā uzglabāšanas uzdevums”.

Pēc tam, kad ir specificēti aģentiem veicamie uzdevumi, jāuzsāk aģentu *mijiedarbības projektēšana*. Mijiedarbības projektēšanas laikā caurskata lietošanas gadījumu scenārijus. Sarežģītākiem scenārijiem vispirms veido *lietošanas gadījumu kartes*, kas norāda visus secīgus uzdevumu pārus, starp kuriem ir nepieciešama viena konkrēta ziņojuma nosūtīšana. Lai attiecīgo ziņojumu ievietotu mijiedarbības diagrammā, pēc lietošanas gadījuma kartes izveides projektētājam ir tikai jāspecificē ziņojuma saturs.

Mijiedarbības projektēšanas solī aģentu mijiedarbība ir definējama, nosakot secīgi izpildāmus uzdevumus, kurus veic dažādi aģenti. Atrodot šādu uzdevumu pāri, projektētājam jādefinē ziņojuma saturs, kas jānosūta starp atbilstošajiem aģentiem. Ziņojuma saturs atbilst nepieciešamajai informācijas plūsmai starp secīgi veicamajiem uzdevumiem. Soļa rezultātā ir izveidota aģentu mijiedarbības diagramma.

Ja veidojamā mijiedarbība satur daudz ziņojumu starp diviem aģentiem, tad diagramma kļūst neuzskatāma. Tādā gadījumā jāveic mijiedarbības detalizācijas solis, kurā mijiedarbība starp diviem aģentiem ir jāattēlo atsevišķā diagrammā – mijiedarbības detalizācijā, bet mijiedarbības diagrammā šo mijiedarbību jāaizstāj ar vienu detalizācijas saiti.

Strādājot ar MASITS rīku, pirms ziņojuma pievienošanas mijiedarbības diagrammai, *ontoloģiju diagrammas veidošanas solī* nepieciešams definēt ziņojuma saturam atbilstošo predikātu un tā definēšanai vajadzīgos konceptus ontoloģiju diagrammā. Koncepti ontoloģiju diagrammā pievienojami kā klases „Koncepts” apakšklases, bet predikāti kā klases „Predikāts” apakšklases.

Aģentu **iekšējās struktūras projektēšanas stadijā** jāspecificē, kā katrs aģents sasniegs savu funkcionalitāti, kas projektēta iepriekšējā stadijā. Lai aģentam veiktu iekšējo projektēšanu, projektētājam jārikojas pēc šāda algoritma:

1. Jāizveido aģenta iekšējais skats, kurā ietilpst aģenta diagramma, aģenta pārlicības, ziņojumu apstrādes likumi, notikumu apstrādes likumi, darba uzsākšanas likumi, kā arī darbību apraksti.
2. Jāpapildina ontoloģiju diagramma ar aģenta pārlicībām atbilstošiem konceptiem.
3. Ja aģents tiek realizēts kā holons, tad:
 - 3.1. Jāpapildina holonu hierarhijas diagramma.
 - 3.2. Jāveic holona projektēšana:
 - 3.2.1. Jāizveido aģentu mijiedarbības diagramma holona iekšējai uzbūvei.
 - 3.2.2. Jāpapildina ontoloģiju diagramma ar holona aģentu mijiedarbībā izmantotajiem predikātiem.
 - 3.2.3. Katram holona aģentam jāveic visi šajā soļu secībā aprakstītie soļi.

Aģenta iekšējā skata izveide notiek iteratīvi. Katrā iterācijā var iekļaut vai nu kādai uztverei, vai darbībai atbilstošos projektējuma elementus. Katras iterācijas ietvaros projektēšanu ieteicams veikt šādā secībā: (1) definēt nepieciešamās aģenta pārlicības; (2) izveidot darbību, attēlot to aģenta diagrammā, izveidot nepieciešamās saites un darbību sarakstā attēlot darbības īpašības; (3) izveidot likumu, ar kura palīdzību tiek apstrādāts kāds notikums un izpildīta attiecīgā darbība vai veiktas izmaiņas aģenta pārlicībās.

Ja aģents tiek realizēts kā holons, tad tas tiek pievienots holonu hierarhijas diagrammai un tam tiek veikta *holona projektēšana*, kuras laikā, izmantojot mijiedarbības diagrammu, projektē tajā esošo aģentu mijiedarbību. Holonā ietilpstošo aģentu iekšējās struktūras projektēšanu veic, izmantojot aģenta iekšējo skatu.

Realizācijas fāzē notiek sistēmas realizēšana JADE platformā, veicot šādus soļus:

1. Jācaurskata iepriekšējos projektos realizētie aģenti (ja tādi ir) un jāatrod tie, kurus iespējams atkārtoti izmantot jaunveidojamā sistēmā.
2. Jāveic koda ģenerēšana no MASITS projektējuma, izmantojot MASITS rīku.
3. Programmētājam jāpievieno metožu kods.
4. Jāizstrādā sistēmas lietotāja saskarne.

Testēšanas fāzē veic, izmantojot JADE platformā iebūvētos rīkus (JADE Test Suite).

Testēšanas fāzē ir šādi soļi:

1. Atsevišķu aģentu testēšana, kuras laikā testē visus sistēmā esošos aģentus.
2. Holonu testēšana, kuras laikā testē katru holonu, uzskatot to par vienotu veselu.
3. Sistēmas testēšana, kuras laikā testē sistēmas funkcionalitāti.

Ieviešanas fāzē MASITS metodoloģija iekļauj divus soļus:

1. Ieviešanas diagrammā izvēlas sistēmu veidojošos JADE konteinerus un tajos ieviešamās aģentu instances.
2. Izmantojot MASITS rīku, ģenerē sistēmas izpildāmo failu, kas palaiž izvēlētos konteinerus un aģentus.

Uzturēšanas fāzē MASITS metodoloģija, balstoties uz 3. nodaļā izstrādāto daudzāģentu arhitektūru, vienkāršo atsevišķu izmaiņu veidu ieviešanu sistēmā. Lai ieviestu izmaiņas, ir jāveic šādi soļi:

1. Jāveic prasību analīze, lai noskaidrotu, kādas izmaiņas sistēmā ir vajadzīgas.
2. Jānoskaidro izmaiņu veids.
3. Jāprojektē izmaiņu ieviešana atbilstoši izmaiņu veidam.
4. Jāveic aģentu realizācija, testēšana un ieviešana.

MASITS metodoloģija izšķir šāda veida izmaiņas:

1. Atvērtam holonam atbilstošas funkcionalitātes izmaiņas. Šajā gadījumā atbilstošajam holonam ir iespējams pievienot (vai izslēgt) aģentus.
2. Izmaiņas, kas atbilst tikai vienam aģentam vai vienam holonam. Šajā gadījumā ir jāmaina tikai atbilstošais aģents vai holons.
3. Izmaiņas, kas atbilst vairāk kā vienam holonam. Šāda veida izmaiņu ieviešanu MASITS metodoloģija nevienkāršo.

MASITS metodoloģija izmanto šādas **diagrammas**:

- Mērķu diagrammu, kas attēlo sistēmas mērķu hierarhiju. Mērķu diagramma sastāv no virsotnēm (mērķiem) un divu veidu saitēm: „UN” un „VAI” dekompozīcijas.
- Lietošanas gadījumu diagrammu, kas attēlo sistēmas līmeņa lietošanas gadījumus.
- Uzdevumu diagrammu, kas attēlo aģentiem veicamo uzdevumu hierarhiju. Uzdevumu diagrammu papildinot ar aģentiem, kas veic katru uzdevumu, iegūst uzdevumu-aģentu diagrammu.
- Lietošanas gadījumu karti, kas aģentorientētā projektēšanā attēlo aģentus, to veicamos uzdevumus un ziņojumu secību starp tiem [BUH 1998]. Lietošanas gadījumu kartes attēlo lietošanas gadījuma scenārija secīgu izpildīšanos iesaistītajos aģentos.
- Aģentu mijiedarbības diagrammu, kas satur divus pamatelementus: aģentus un saites starp tiem (ziņojumus). Saite nosaukums ir nosūtāmais predikāts. Šajā diagrammā attēlo arī mijiedarbību ar lietotāju, kas aprakstīta ar interfeisa metožu izsaukumiem un notikumiem, ko aģents spēj uztvert.

- Ontoloģiju diagrammu, kas ir modificēta klašu diagramma, kura apraksta aģentu darbības problēmsfēru, izmantojot konceptus (klases „Koncepts” apakšklase) un predikātus (klases „Predikāts” apakšklase).
- Aģenta iekšējo skatu, kas sastāv no šādiem elementiem: (1) MASITS aģentu diagrammas, kas attēlo aģenta uztveres, nosūtītos ziņojumus un veicamās darbības; (2) ziņojumu apstrādes likumiem, kas apraksta ieeju apstrādi; (3) darba uzsākšanas likumiem, kas apraksta, kādas darbības aģentam ir jāveic, uzsākot darbu; (4) aģenta pārlicībām; (5) darbību saraksta, kurš satur visu aģenta darbību nosaukumus un to īpašības.
- Holonu hierarhijas diagrammu, kas attēlo sistēmu veidojošo holonu hierarhiju.
- Ieviešanas diagrammu, kas specificē palaižamos JADE konteinerus un tajos veidojamās aģentu instances.

4.2. MASITS metodoloģijas automatizācijas algoritmi

Lai varētu izveidot rīku, kas realizē praktisku izstrādi ar MASITS metodoloģiju, ir definēti metodoloģijai atbilstoši izstrādes automatizācijas algoritmi, kas automatizē vairākas izstrādes darbības.

Pirmkārt, ir izveidoti algoritmi saišu veidošanai starp dažādu diagrammu elementiem, lai realizētu šķērspārbaudi un nepretrunīguma nodrošināšanu starp diagrammām. Šķērspārbaudes nodrošināšanai starp diagrammām veido loģiskas saites starp dažādu modeļu elementiem, kas viens otru atbalsta, piemēram, no mērķu diagrammas veido saites uz uzdevumiem, kas nodrošina mērķu sasniegšanu. Izmantojot šādas saites, var veikt analīzi, vai visi kāda modeļa elementi ir atbalstīti citā modelī. Nepretrunīguma nodrošināšanu veic tādējādi, ka katra veida elementu var dzēst vai izmainīt tikai vienā diagrammā, piemēram, aģentu var dzēst tikai mijiedarbības diagrammā. No elementa, kuru var dzēst vai izmainīt, veido saites uz elementiem, kuri ir attiecīgi jāmaina vai jādzēš, ja tiek mainīts vai dzēsts konkrētais elements. Šķērspārbaudei nepieciešamās saites jādefinē MASITS rīka lietotājam, kamēr nepretrunīguma nodrošināšanas saites veido automātiski brīdī, kad lietotājs definē atbilstošo elementu vai tā īpašību.

Otrkārt, ir izstrādāti algoritmi, kas ģenerē atsevišķas diagrammu daļas vai elementus, pamatojoties uz iepriekš izveidotajām diagrammām. Brīdī, kad lietotājs izveido kādu diagrammas elementu, MASITS rīks izveido visus elementus visās citās diagrammās, kuri viennozīmīgi seko no lietotāja izveidotā elementa. Piemēram, pievienojot jaunu aģentu, uzdevumu-aģentu diagrammai, rīks pievieno attiecīgo aģentu mijiedarbības diagrammai un

izveido tam atbilstošu iekšējo skatu. MASITS rīks automātiski ģenerē arī jaunas diagrammas – holonu hierarhijas diagrammu un uzdevumu aģentu diagrammu.

Treškārt, ir izveidoti algoritmi, kas no MASITS projektējuma ģenerē JADE platformai atbilstošo klašu kodu. Koda ģenerēšanas laikā tiek izveidotas pakotnes, klases un sistēmas palaižamais fails. Pakotņu veidošana notiek, izveidojot jaunu mapi atbilstošā ceļā. Rīks veido projekta galveno pakotni, ontoloģijas pakotni un katram holonam atbilstošu pakotni. Java klašu kodu ģenerē, balstoties uz rīkā iebūvētiem šabloniem. Promocijas darbā iekļauti šabloni un algoritmi koda ģenerēšanai no tiem. Algoritmi izveidoti šādu klašu ģenerēšanai: ontoloģijas bāzes klase, visu konceptu un predikātu klases, aģentu klases un aģentu izpildīto uzvedību klases. Bez tam piedāvāts šablonos balstīts sistēmas palaižamā faila ģenerēšanas algoritms.

4.3. MASITS rīks

Lai būtu iespējams MASITS metodoloģiju praktiski pielietot IMS-u izstrādē, ir realizēts MASITS rīks, kas pilnībā realizē visus 4.2. apakšnodaļā dotos automatizācijas algoritmus. Rīka galvenās funkcijas ir šādas:

- Funkcijas visu MASITS metodoloģijā iekļauto diagrammu zīmēšanai un rediģēšanai.
- Projektējuma saglabāšana, nolasīšana no faila un drukāšana.
- Starpdigrammu saišu veidošana un šķērspārbaude starp diagrammām.
- Nepretrunīguma nodrošināšana starp diagrammām.
- JADE platformai atbilstoša koda ģenerēšana no projektējuma.

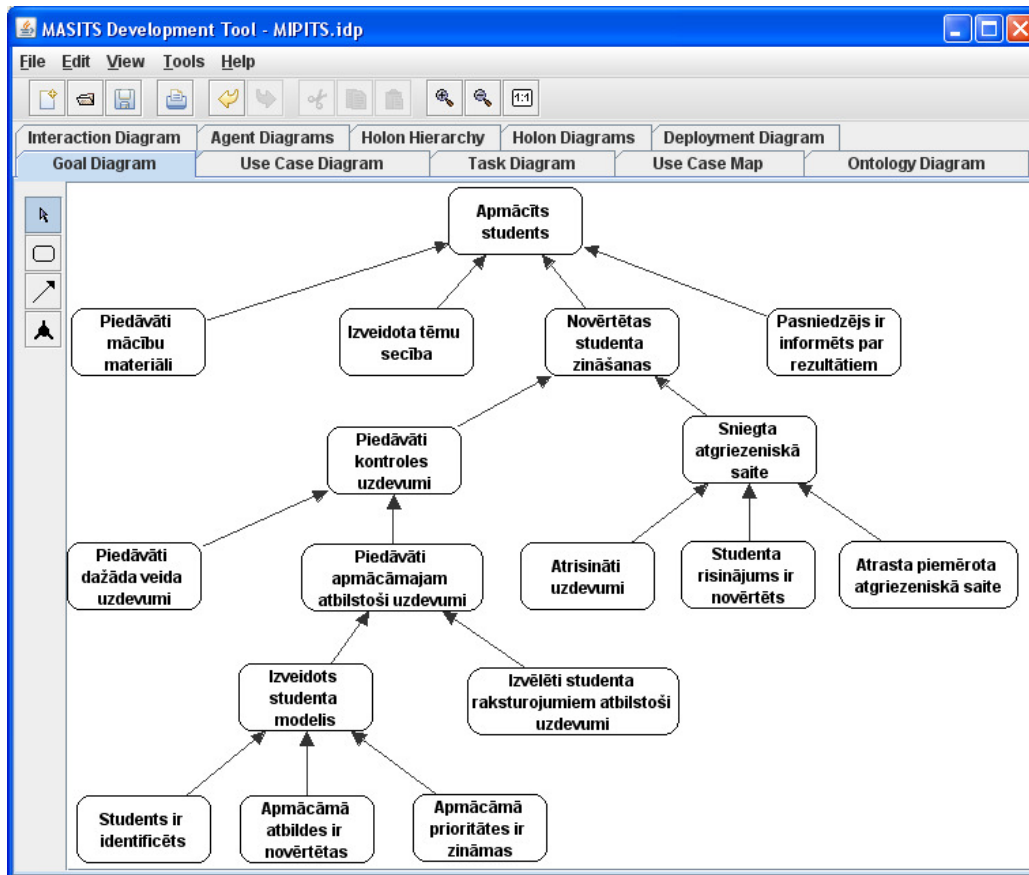
MASITS rīks publicēts darbā [LAV 2009a]. Promocijas darba 4.3. apakšnodaļā dota MASITS rīka funkcionalitāte un lietotāja saskarnes apraksts, savukārt, pielikumā P3 iekļauts MASITS rīka lietotāja ceļvedis. MASITS rīka lietotāja saskarne dota 4.3. attēlā.

MASITS rīka lietotājs var jebkurā brīdī pāriet uz jebkuru no izstrādes gaitā veidojamām diagrammām. Tas ļauj gan veikt nepieciešamos labojumus iepriekš izveidotajās diagrammās, gan realizēt iteratīvu izstrādes procesu, iteratīvi veidojot vienu diagrammu vai visu projektējumu.

Galvenie ceturtās nodaļas secinājumi ir šādi:

- MASITS metodoloģijā iekļautās zināšanas no IMS-u pētījumiem vienkāršo izstrādes procesu, atvieglojot vairākus izstrādes soļus. Vispārīgās aģentorientētas programmatūras izstrādes metodoloģijās viens no sarežģītākajiem soļiem ir aģentu definēšana un funkcionalitāšu piešķiršana aģentiem. Tādēļ par nozīmīgāko ieguvumu no IMS-u pētījumu

integrācijas uzskatāma tehnika uzdevumu piešķiršanai aģentiem, kas izmanto IMS-u pētījumos sakņotus likumus.



4.3. att. MASITS rīka lietotāja saskarne ar mērķu diagrammu

- Aģentu iekšējās struktūras projektēšanā izmantoti vienai konkrētai aģentu izstrādes platformai (JADE) atbilstoši koncepti. Šāda risinājuma galvenais trūkums ir tāds, ka metodoloģija ļauj realizēt IMS-as tikai šajā platformā. Taču tam ir arī nozīmīga priekšrocība – MASITS metodoloģijai ir vienkārša realizācijas fāze, kurā ir iespējams veikt sistēmas koda ģenerēšanu un nav nepieciešamas nekādas projektēšanā izmantoto konceptu transformācijas aģentu platformā izmantotajos konceptos.
- MASITS metodoloģiju atbalstoša rīka izstrāde veicina praktisku IMS-u izstrādi ar šo metodoloģiju, piedāvājot visu nepieciešamo diagrammu zīmēšanu, daļēju izstrādes procesa automatizāciju un kļūdu novēršanu projektējumā, veicot šķērspārbaudes un uzturot diagrammu nepretrunīgumu.
- Definētās atvērtās holoniskās daudzāģentu arhitektūras izmantošana MASITS metodoloģijai ļauj izveidot atbalstu arī uzturēšanas fāzei, kas nav iekļauts nevienā citā no analizētajām metodoloģijām. Uzturēšanas fāzē galvenais ieguvums no šīs arhitektūras

izmantošanas ir vienkāršota izmaiņu ieviešana, jo daļu no izmaiņām var ieviest tikai pievienojot vai izslēdzot aģentus no sistēmas.

- Tā kā MASITS metodoloģija atšķirībā no vispārīgajām aģentorientētas programmatūras izstrādes metodoloģijām atbalsta pilnu dzīves ciklu, to ir iespējams izmantot visa izstrādes procesa gaitā, sākot no prasību analīzes, beidzot ar ieviešanu un uzturēšanu. Tas novērš tādu vispārīgo metodoloģiju trūkumu, kā atsevišķu fāžu neatbalstīšanu, kuras ir vai nu jāveic bez metodoloģijas atbalsta, vai papildus jāizmanto kāda cita metodoloģija.

5. Intelektuālas mācību sistēmas praktiska piemēra izstrāde, izmantojot MASITS metodoloģiju un rīku

Lai praktiski pārbaudītu MASITS metodoloģiju un rīku, ir izstrādāta intelektuāla mācību sistēma MIPITS. Sistēma izveidota pirmajiem trim priekšmeta „Mākslīgā intelekta pamati” moduļiem: „Ievads”, „Neinformēta pārmeklēšana” un „Informēta pārmeklēšana” [LUG 2005] ar mērķi realizēt individuālu pieeju katram no vairāk nekā 200 studentiem, kas vienlaicīgi mācās šajā kursā. Promocijas darba piektā nodaļa ietver sistēmas izstrādes procesa ar MASITS metodoloģiju un rīku aprakstu, kas iekļauj katra soļa veikšanu un iegūtos artefaktus. Izstrādātās sistēmas apraksts iekļauts promocijas darba pamattekstā un publicēts darbā [LAV 2010].

Sākot darbu ar MIPITS sistēmu, apmācāmais veic reģistrāciju sistēmā, norādot savus personas datus un prioritātes attiecībā uz kontroles uzdevumiem. Saņemot studenta datus, sistēma izveido apgūstamo tēmu secību. Katrā mācību tēmā MIPITS sistēma vispirms piedāvā studentam atkārtot mācību vielu, iepazīstoties ar konspektīviem mācību materiāliem par katru no kursa ietvaros apgūtajām tēmām. Pēc tam tiek piedāvāts kontroles uzdevums attiecīgajā tēmā, pie tam tas tiek darīts gan katras tēmas beigās, gan noslēdzot katra moduļa apgūšanu. Pēc uzdevuma izpildīšanas sistēma sniedz atgriezenisko saiti, kurā norāda apmācāmā kļūdas.

Sistēma piedāvā kursa „Mākslīgā intelekta pamati” pirmajiem trim moduļiem atbilstošus uzdevumu veidus, kā arī ļauj pievienot jaunus uzdevumu veidus. Minētajos kursa moduļos un līdz ar to MIPITS sistēmas sākotnējā versijā ir iekļauti šādi uzdevumu veidi:

- Dažādu veidu testi, kas satur gan tādus, kuros apmācāmais var izvēlēties tikai vienu atbildi, gan tādus, kuros var izvēlēties vairākas atbildes, gan tādus, kuros apmācāmajam atbilde jāieraksta pašam. Papildus paredzēta iespēja testu jautājumiem pievienot attēlus un ģenerētas stāvokļu telpas, par kurām tiek uzdoti jautājumi.

- Pārmeklēšanas algoritmu realizācijas uzdevumi, kuros students veic stāvokļu telpas pārmeklēšanu, izmantojot sarakstus OPEN un CLOSED [LUG 2005].
- Divpersonu spēļu algoritmu (minimaksa algoritma un alfa-beta procedūras) realizācijas uzdevumi.

Kontroles uzdevumus sistēma ģenerē atbilstoši katra apmācāmā īpašībām un vēlmēm. Vispirms visiem uzdevumu ģenerēšanas aģentiem pieprasa ģenerēt uzdevumus nepieciešamajā tēmā, kas maksimāli atbilst studenta prioritātēm un zināšanu līmenim. Pēc uzdevumu ģenerēšanas pabeigšanas uzdevumu ģenerēšanas holona galvas aģents nosaka piemērotāko no šiem uzdevumiem un nosūta to interfeisa aģentam attēlošanai. Piemērotākā uzdevuma izvēlei aģenti izmanto četrus kritērijus, kuriem vispirms aprēķina vēlamo vērtību:

- Uzdevuma grūtības pakāpe. Reģistrējoties sistēmā, students norāda sākotnējo uzdevuma grūtības pakāpi skalā no 1 (ļoti vienkārši uzdevumi) līdz 5 (vissarežģītākie uzdevumi). Tas ir studenta subjektīvs vērtējums par savu zināšanu līmeni, kas var neatbilst reālajai situācijai. Tādēļ sistēma vēlamo uzdevumu grūtības pakāpi nosaka, ņemot vērā gan studenta izvēlēto sākotnējo grūtības pakāpi, gan līdzšinējos studenta rezultātus, izmantojot šādu formulu:

$$\text{grPak}_{\text{vel}} = \frac{s \cdot \text{grPak}_{\text{sāk}} + \text{max} \cdot l}{s + \text{max}},$$

kur

s – koeficients, kas norāda, cik iegūstamiem punktiem par uzdevumu atrisināšanu atbilst studenta sākotnējais uzdevumu grūtības pakāpes vērtējums. Tā vērtība noteikta empīriski un ir vienāda ar 50. Salīdzinājumam 1 testa jautājums ir 2 līdz 4 punktus vērts.

$\text{grPak}_{\text{sāk}}$ – studenta sākotnēji izvēlēta uzdevuma grūtības pakāpe.

max – maksimālais punktu skaits par uzdevumiem, ko students līdz šim risinājis.

l – līmenis, kas atbilst studenta līdz šim uzrādītajiem rezultātiem. Šī līmeņa noteikšanai vispirms atrod, cik punktus procentuāli students ir ieguvis no maksimāli iespējamajiem punktiem par tiem uzdevumiem, ko viņš ir izpildījis. Tad, izmantojot empīriski iegūtu sakarību, aprēķina līmeni: 0-34% atbilst 1. līmenim, 35-49% atbilst 2. līmenim, 50-64% atbilst 3. līmenim, 65-80% atbilst 4. līmenim, bet virs 80% atbilst 5. līmenim.

- Uzdevuma apjoms. Šo parametru students izvēlas, reģistrējoties sistēmā. Tam ir trīs iespējamās vērtības: mazs, vidējs un liels, kam attiecīgi atbilst skaitļi 1, 2 un 3.

- Uzdevuma praktiskums. Šo parametru arī izvēlas students, reģistrējoties sistēmā. Tam ir trīs iespējamās vērtības: „Vairāk teorētisku uzdevumu”, „Sabalansēti” un „Vairāk praktisku uzdevumu”, kam attiecīgi atbilst skaitļi 1, 2 un 3.
- Katra uzdevumu veida biežums. Lai, pamatojoties uz pārējiem parametriem, studentam netiktu pārāk bieži doti viena veida uzdevumi, tiek aprēķināts attiecīgā uzdevumu veida biežums. Ja students nav saņēmis nevienu uzdevumu, tad biežums ir 0, pretējā gadījumā biežumu aprēķina šādi:

$$b_i = \frac{s_i}{s},$$

kur

b_i – i-tā veida uzdevuma biežums;

s_i – i-tā veida uzdevumu skaits, kas izsniegti studentam;

s – kopējais uzdevumu skaits, kas izsniegti studentam.

Pēc visu kritēriju vērtību aprēķināšanas ir jāminimizē starpība starp vēlamajiem raksturojumiem un uzdevuma raksturojumiem. To veic uzdevumu ģenerēšanas holona galvas aģents, kas uzdevumu piemērotību aprēķina, izmantojot šādu formulu:

$$U = -(|\text{grPak}_{\text{vēl}} - \text{grPak}_{\text{uzd}}| * k_g + |\text{apj}_{\text{vēl}} - \text{apj}_{\text{uzd}}| * k_a + |\text{prakt}_{\text{vēl}} - \text{prakt}_{\text{uzd}}| * k_p + b_v * k_b),$$

kur

$\text{grPak}_{\text{vēl}}$ – vēlamā grūtības pakāpe;

$\text{grPak}_{\text{uzd}}$ – uzdevuma grūtības pakāpe;

k_g – grūtības pakāpes svara koeficients;

$\text{apj}_{\text{vēl}}$ – vēlamais uzdevuma apjoms;

apj_{uzd} – reālais uzdevuma apjoms;

k_a – uzdevuma apjoma svara koeficients;

$\text{prakt}_{\text{vēl}}$ – vēlamais uzdevuma praktiskums;

$\text{prakt}_{\text{uzd}}$ – reālais uzdevuma praktiskums;

k_p – uzdevuma praktiskuma svara koeficients; b_v – uzdevuma veida biežums;

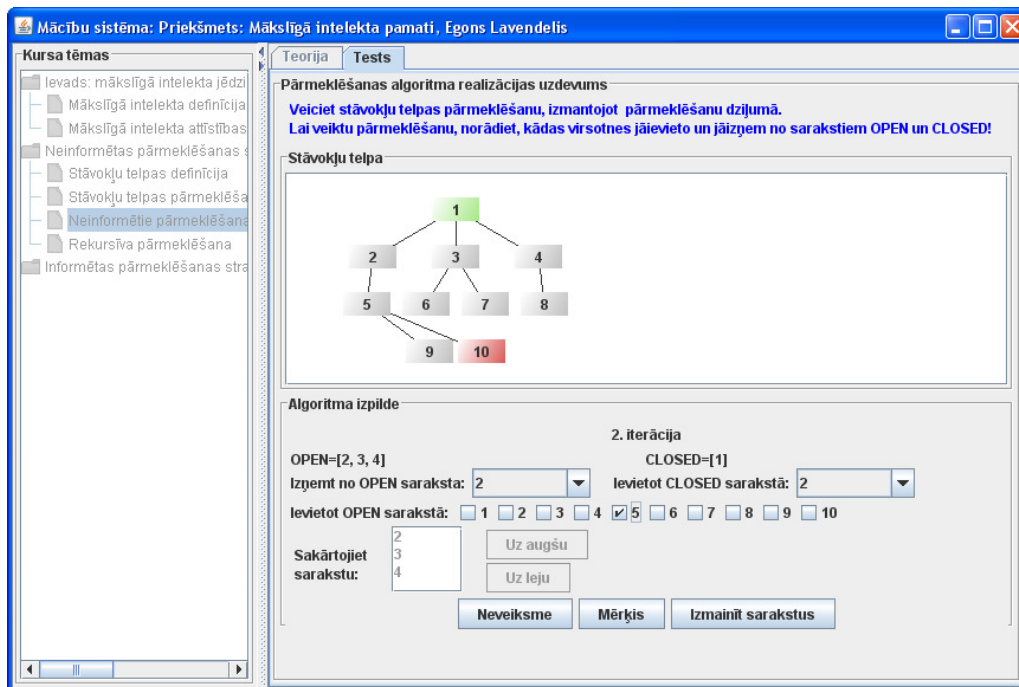
k_b – uzdevuma biežuma svara koeficients.

MIPITS sistēmā šiem svaru koeficientiem empīriski izvēlētas tādas vērtības, lai visi parametri reāli ietekmētu uzdevuma izvēli: $k_g=2$, $k_a=3$, $k_p=3$, $k_b=6$.

MIPITS sistēmas lietotāja saskarne sastāv no tēmu saraksta (kreisajā pusē) un vietas mācību materiālu un uzdevumu attēlošanai. Lietotāja saskarne ar attēlotu pārmeklēšanas uzdevumu dota 5.1. attēlā.

MIPITS sistēma ir realizēta, izmantojot 3. nodaļā izstrādāto daudzāģentu arhitektūru. Augstākā līmeņa holons sastāv no interfeisa aģenta, mācību plāna aģenta, mācību stratēģijas aģenta, studenta modelēšanas aģenta, uzdevumu ģenerēšanas aģenta, zināšanu novērtēšanas aģenta, eksperta aģenta un atgriezeniskās saites aģenta. Lai būtu iespējams sistēmai vienkārši

pievienot jaunu uzdevuma veidu, uzdevumu ģenerēšanas aģents, eksperta aģents, atgriezeniskās saites aģents un interfeisa aģents ir realizēti atvērta holonu veidā, kur katrs ķermeņa aģents veic sava uzdevuma veida apstrādi. Šāda izvēle ļauj pievienot jaunus uzdevumu veidus, nemainot esošo sistēmas kodu. Savukārt, uzdevumu veida izņemšana no sistēmas ir vienkārša, jo no sistēmas palaišanas faila ir tikai jāizdzēš attiecīgo aģentu palaišanas komandas.



5.1. att. MIPITS sistēmas lietotāja saskarne, kur attēlots pārmeklēšanas uzdevums

Lai sistēmai pievienotu jaunu uzdevuma veidu, jāveic trīs uzdevumi:

1. Jāpapildina ontoloģija ar uzdevuma un tā atrisinājuma klasēm un citām klasēm, ja tādas ir nepieciešamas.
2. Katrā atvārtā holonā jāpievieno uzdevuma veidam atbilstošs aģents.
3. Jāpapildina datu bāze ar informāciju par jauno uzdevuma veidu.

Galvenie nodaļas secinājumi ir šādi:

- Izstrādātais MIPITS sistēmas prototips apliecina, ka 3. nodaļā definētā arhitektūra un 4. nodaļā piedāvātā MASITS metodoloģija un rīks ir izmantojami aģentos sakņotu IMS-u izstrādei.
- MASITS metodoloģijas pielietošana MIPITS sistēmas izstrādei pierādīja, ka metodoloģijā iekļautās tehnikas ir pietiekamas, lai veiktu visus aģentu izstrādei nepieciešamos soļus. Bez tam nevienai no izmantotajām tehnikām nebija nepieciešama tāda informācija, kas

IMS-ām nav svarīga, tādēļ var secināt, ka MASITS metodoloģijā iekļautās tehnikas ir atbilstošas IMS-u izstrādei.

- MASITS metodoloģijā un rīkā iekļautās saites starp dažādām diagrammām palīdz uzturēt diagrammu nepretrunīgumu un ļauj ērti izmainīt iepriekš izveidotās diagrammas. Šķērspārbaude starp diagrammām ir efektīvi izmantojama un palīdz atrast kļūdas projektējumā (MIPITS sistēmas izstrādes gaitā, izmantojot šķērspārbaudi, tika noskaidrots, ka vienam mērķim ir aizmirsts izveidot atbilstošu lietošanas gadījumu un dažiem uzdevumiem nav izveidotas atbilstošas darbības aģentu iekšējos skatos).
- MASITS metodoloģijā iekļautās zināšanas no IMS-u pētījumiem atvieglo vairākus izstrādes soļus. Vispārīgās aģentorientētas programmatūras izstrādes metodoloģijās viens no sarežģītākajiem soļiem ir aģentu definēšana un funkcionalitāšu piešķiršana aģentiem, tādēļ MASITS metodoloģijā ir iekļauta aģentu kopa, kuriem uzdevumi tiek piešķirti, balstoties uz konkrētiem likumiem, kas, realizējot MIPITS sistēmu, sevi pilnībā attaisnoja.
- MASITS rīks no projektējuma spēj ģenerēt 20-60% no dažādu klašu koda un līdz ar to ievērojami samazina programmētāja veicamā darba apjomu.
- Izvēlētā atvērtā arhitektūra ļauj izmantot nelielus aģentus, kuru izstrāde ir salīdzinoši vienkārša. Lielākā daļa izstrādāto aģentu klašu bija 100-300 koda rindiņu, bet atsevišķās klasēs līdz 500 rindiņām, kas ir vairākas reizes mazāk nekā klasēs, ko izmanto 1.4. nodaļā aprakstītās aģentu kopas realizēšanai.
- Izveidotā MIPITS sistēma demonstrē, ka izstrādātā arhitektūra ļauj izveidot atvērtas sistēmas, kurās jaunu funkcionalitāti ir iespējams pievienot, nemainot esošo kodu, bet pievienojot jaunus aģentus un ontoloģijas klases. Izstrādātajā sistēmā šādi var pievienot jaunus uzdevumu veidus.

6. Rezultāti un secinājumi

Lai izanalizētu pašreizējo situāciju aģentos sakņotu IMS-u izstrādē darba ietvaros ir veiktas šādas aktivitātes:

- Aģentu pielietojumu IMS-u realizēšanai izpēte, kuras rezultātā secināts, ka:
 - Eksistējošās aģentos sakņotās IMS-as un aģentu arhitektūras IMS-u realizēšanai nav pietiekami atvērtas, un līdz ar to izmaiņu ieviešana tajās ir apgrūtināta.
 - IMS-as sastāv no lielizmēra aģentiem, kam ir daudz uzdevumu. Līdz ar to sistēmas modularitāte ir nepietiekama, kas sarežģī izstrādi un aģentu atkārtotu izmantošanu.
 - Lai aģentus varētu sekmīgi pielietot kā industriālu risinājumu IMS-u izstrādei, ir nepieciešama atbilstoša izstrādes metodoloģija.

- Aģentu realizācijas platformu piemērotības IMS-u izstrādei analīze, kuras gaitā izvirzīti galvenie kritēriji, kam jāatbilst platformai, lai tā būtu piemērota IMS-u izstrādei, un analizēta platformu atbilstība šiem kritērijiem. Analīzes rezultātā secināts, ka vispiemērotākā platforma IMS-u realizācijai ir JADE.
- Vispārīgo daudzāģentu arhitektūru analīze ar mērķi novērtēt to piemērotību IMS-u modularitātes un atvērtības paaugstināšanai, kā rezultātā ir secināts, ka vispiemērotākās ir holoniskas daudzāģentu sistēmas. Holonisku daudzāģentu sistēmu piemērotību IMS-u izstrādei apliecina arī IMS-u atbilstība holonisku problēmsfēru kritērijiem.
- Vispārīgo aģentorientētas programmatūras izstrādes metodoloģiju analīze, ar mērķi noskaidrot metodoloģiju un tajās izmantoto tehniku piemērotību IMS-u izstrādei. Analīzes rezultātā noskaidrots, ka nav IMS-u īpašībām atbilstošu metodoloģiju un, izmantojot vispārīgās metodoloģijas, nav iespējams sekmīgi izmantot IMS-u pētījumu rezultātus. Secināts arī, ka tikai daļa no vispārīgās metodoloģijās izmantotajām teknikām ir pielietojamas IMS-u izstrādē.

Lai atrisinātu analīzes gaitā atklātās problēmas, darbā ir sasniegti šādi galvenie rezultāti:

- Ir izstrādāta atvērta holoniska arhitektūra aģentos sakņotu IMS-u izstrādei, kas veidota, balstoties uz tradicionālo aģentu kopu IMS-u realizācijai. Augstākā līmeņa holons ir tradicionālās aģentu kopas modifikācija. Katram augstākā līmeņa holonā ietilpstošajam aģentam definēti iespējamie apakšholoni.
- Ir definēta MASITS metodoloģija aģentos sakņotu IMS-u izstrādei. Metodoloģija ir izveidota, izmantojot gan aģentorientētas programmatūras inženierijas, gan IMS-u pētījumu rezultātus. Līdz ar to, tā iekļauj gan IMS-u īpašībām piemērotākās tehnikas no vispārīgām aģentorientētas programmatūras izstrādes metodoloģijām, gan jaunas īpaši IMS-u izstrādei veidotas tehnikas, gan zināšanas no IMS-u pētījumiem.
- Nozīmīgākās no MASITS metodoloģijai izstrādātajām jaunajām teknikām ir šādas:
 - Uzdevumu piešķiršana aģentiem, balstoties uz IMS-u pētījumos sakņotiem likumiem.
 - Aģenta iekšējā skata veidošana atbilstoši IMS-u īpašībām un JADE platformai.
 - Mijiedarbības definēšana atbilstoši lietošanas gadījumu scenārijiem un lietošanas gadījumu kartēm.
 - Holonu kā atsevišķu daudzāģentu sistēmu projektēšana.

- Ir izstrādāti aģentos sakņotu IMS-u izstrādes ar MASITS metodoloģiju automatizācijas algoritmi, kas nodrošina nepretrunīguma uzturēšanu un šķērspārbaudi starp diagrammām, kā arī diagrammu un koda ģenerēšanu.
- Ir izstrādāts MASITS rīks, kas realizē visu MASITS metodoloģijā izmantoto diagrammu zīmēšanu un visus aprakstītos automatizācijas algoritmus.
- Izmantojot MASITS metodoloģiju un rīku, ir realizēta MIPITS sistēma priekšmeta „Mākslīgā intelekta pamati” mācīšanai. Sistēma piedāvā apmācāmajam mācību materiālus un kontroles uzdevumus, kas tiek adaptēti atbilstoši studenta modelim.

Darba izstrādes laikā ir iegūti arī šādi papildus rezultāti:

- Balstoties uz ūdenskrituma modeli un eksistējošām aģentorientētas programmatūras izstrādes metodoloģijām, ir definēts aģentorientētas programmatūras izstrādes dzīves cikls.
- Izveidota aģentorientētas programmatūras izstrādes metodoloģiju klasifikācija pēc to izcelsmes.

Darba praktiskais pielietojums:

- Izstrādātais MASITS rīks ļauj pielietot gan MASITS metodoloģiju, gan atvērto arhitektūru praktiskai IMS-u izstrādei.
- Izstrādātā MIPITS sistēma ir izmantojama kursa „Mākslīgā intelekta pamati” mācīšanai.

Darba galvenie secinājumi ir:

- Aģentos sakņotu IMS-u praktiskai izstrādei ir nepieciešamas atbilstošas tehnoloģijas. Svarīgākās no tām ir izstrādes metodoloģija un realizācijas platforma. Savukārt, sistēmu modularitātes un atvērtības uzlabošanai ir nepieciešama specifiska arhitektūra.
- Izstrādātā holoniskā arhitektūra ļauj veidot atvērtas IMS-as, kā arī samazina izstrādājamo aģentu izmēru un līdz ar to vienkāršo izstrādi. Pateicoties arhitektūras atvērtībai un holonu izmantošanai, tā ļauj vienkāršāk ieviest izmaiņas IMS-ās, lai vai nu izmainītu to funkcionalitāti, vai tās pielāgotu jauniem mācību kursiem. Tās piemērotību atvērtām IMS-ām pierāda izstrādātā atvērtā MIPITS sistēma, kura ir viegli paplašināma ar jauniem uzdevumu veidiem, nemainot esošo kodu, bet pievienojot jaunus aģentus. Izstrādātā arhitektūra ir augsta līmeņa arhitektūra, ko var pielāgot dažādu IMS-u izstrādes vajadzībām, izvēloties, kuri aģenti ir realizējami kā holoni un kuri kā vienkārši aģenti.
- Specifiskas IMS-u izstrādes metodoloģijas izveide vienkāršo izstrādes procesu, izmantojot:

- IMS-u pētījumu rezultātus, kas atvieglo vairāku izstrādes soļu veikšanu, piemēram, uzdevumu piešķiršana aģentiem no IMS-as realizējošo aģentu kopas notiek, izmantojot IMS-u pētījumos sakņotus likumus, kas aizstāj vienu no sarežģītākajiem soļiem vispārīgās aģentorientētas programmatūras izstrādes metodoloģijās – aģentu definēšanu.
- IMS-u īpašībām piemērotas tehnikas, kas analizē tieši IMS-ām svarīgus aspektus un ļauj rīkoties saskaņā ar IMS-u pētījumiem.
- MASITS rīks vienkāršo praktisku IMS-u izstrādi, nodrošinot diagrammu zīmēšanu, izstrādes procesa automatizāciju, nepretrunīguma starp diagrammām uzturēšanu, kā arī palīdzot novērst projektējuma kļūdas, kas saistītas ar to, ka veidojamais artefakts neatbilst iepriekš izstrādātajiem.
- MIPITS sistēmas izstrāde pierādīja, ka MASITS rīks un metodoloģija ir piemēroti aģentos sakņotu IMS-u izstrādei, un IMS-u pētījumu rezultāti kopā ar IMS-u īpašībām piemērotām tehnikām atvieglo aģentos sakņotu IMS-u izstrādi.

Turpmākie pētījumi ir saistīti ar:

- Lielāku atbalstu intelektuālo mehānismu projektēšanai. Patlaban MASITS metodoloģija aģentos ļauj veidot jebkādu intelektuālu mehānismu, taču to izveidei nav paredzēti speciāli rīki. Šādu rīku izveide atvieglotu konkrētu mehānismu (mašīnu apmācība, plānošana, u.c. tamlīdzīgu mehānismu) projektēšanu.
- Atkārtoti izmantojamu zemāko līmeņu aģentu bibliotēkas izveidi, lai vienkāršotu jaunu sistēmu realizācijas fāzi.
- Vispārīgas IMS-u ontoloģijas izstrādi, kas iekļautu vispārīgus IMS-ās izmantotus konceptus un predikātus un varētu tikt papildināta ar katrai sistēmai specifiskiem konceptiem un predikātiem. Šādas ontoloģijas izstrāde gan ļautu atvieglot katras sistēmas ontoloģijas izveidi, gan veicinātu aģentu atkārtotu lietošanu, jo dažādās sistēmās izmantotajās ontoloģijās būtu iekļauts lielāks skaits kopīgu konceptu.

Bibliogrāfiskais saraksts

- [AGE] Agentbuilder Home Page. Pieejams: <http://www.agentbuilder.com/> (apmeklēts: 07.04.2009).
- [ALP 1999] Alpert, S.R., Singley, M.K., Fairweather, P.G. Deploying Intelligent Tutors on the Web: An Architecture and an Example. In: International Journal of Artificial Intelligence in Education, 1999, Vol. 10, No. 2, pp.183-197.
- [ANO 2007] Anohina A. Adaptīvas apmācības un zināšanu vērtēšanas intelektuāla atbalsta sistēmas izstrādāšana. Promocijas darbs, Rīga, 2007.

- [BEL 2007] Bellifemine F., Caire G., Greenwood, D. Developing Multi-Agent Systems with JADE. John Wiley & Sons, West Sussex, p. 286, 2007.
- [BER 2005] Bernon, C., et al. Engineering Adaptive Multi-Agent Systems: The ADELFE Methodology. In Agent-Oriented Methodologies, Henderson-Sellers B., Giorgini P. (Eds.) Idea Group Publishing, London, 2005, pp. 172-202.
- [BER 2004] Bergenti, F., Gleizes, M.P., Zambonelli, F. (Eds.). Methodologies and Software Engineering for Agent Systems. The Agent-Oriented Software Engineering Handbook, Kluwer Academic Publishers, London p. 505, 2004.
- [BIG 2002] Bigus, J. P., et al. ABLE: A Toolkit for Building Multiagent Autonomic Systems. In IBM Systems Journal, 41(3), 2002, pp. 350-371.
- [BIN 2005] Bindé, J. et al. Towards Knowledge Societies. UNESCO Publishing, 2005, p. 220. Pieejams: <http://unesdoc.unesco.org/images/0014/001418/141843e.pdf> (apmeklēts: 11.05.2009)
- [BOL 1999] The European Higher Education Area. Joint Declaration of the European Ministers of Education Convened in Bologna on the 19th of June 1999, p. 2 .
- [BOR 2005a] Bordini, R.H., Hubner, J.F., Vieira, R. Jason and the Golden Fleece of Agent-Oriented Programming. In Multi-Agent Programming. Languages, Platforms and Applications (Bordini, R.H. et al. Eds.), Springer, New York, 2004, pp. 3-37.
- [BOR 2005b] Bordini, R.H., Dastani, M., Dix, J., Seghrouchni, A.E.F. Multi-Agent Programming. Languages, Platforms and Applications, Springer, New York, 2004. p. 295.
- [BRA 1997] Brazier, F.M.T., Dunin-Keplicz, B.M., Jennings N.R., Treur J. DESIRE: Modelling Multi-Agent Systems in a Compositional Formal Framework. In International Journal of Cooperative Information Systems", 6(1), 1997, pp. 67-94.
- [BRE 2002] Bresciani, P., Giorgini, P., Giunchiglia, F., Mylopoulos, J. and Perini, A. TROPOS: An Agent-Oriented Software Development Methodology. Technical Report DIT-02-015, Informatica e Telecomunicazioni, University of Trento. 2002.
- [BUC 2002] Buckle, P., et al. Scalability in Multi-Agent Systems: the FIPA-OS Perspective. In Foundations and Applications of Multi-Agent Systems LNCS Vol. 2403, Springer, 2002, pp. 110-130.
- [BUH 1998] Buhr, R.J.A., Elammari, M., Gray, T., Mankovski, S. Applying Use Case Maps to Multi-Agent Systems: A Feature Interaction Example. In HICSS(6), 1998, pp. 171-179.
- [BUR 2002] Burrafato, P., Cossentino, M. Designing a multi-agent solution for a bookstore with the PASSI methodology. 4th International Bi-Conference Workshop on Agent-Oriented Information Systems (AOIS-2002) at CAiSE'02. Toronto, Ontario, Canada, 2002.
- [BUS 2002] Bush, G., Cranefield, S., Purvis, M. The Styx Agent Methodology, 2002. Pieejams: <http://citeseer.ist.psu.edu/cache/papers/cs/22112/http:zSzzSzdivcom.otago.ac.nzSzinfosicizSzpublctnszSzcompletezSzpaperszSzdp2001-02.pdf/bush01styx.pdf> (Apmeklēts: 10.10.2007)

- [CAI 2001] Caire, G., et al. Agent Oriented Analysis Using MESSAGE/UML. In: AOSE 2001.
- [CAP 2000a] Capuano, N., De Santo, M., Marsella, M., et al. A Multi-Agent Architecture for Intelligent Tutoring. Proceedings of the International Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet, SSGRR 2000, L'Aquila, July 31 - August 06, 2000.
- [CAP 2000b] Capuano, N., Marsella, M., Salerno, S. ABITS: An Agent Based Intelligent Tutoring System for Distance Learning. In Proceedings of the International Workshop on Adaptive and Intelligent Web-Based Education Systems ITS 2000, Montreal, Canada, 2000.
- [CAR 1970] Carbonell, J.R. AI in CAI: An Artificial Intelligence Approach to Computer-Assisted Instruction. IEEE Transactions on Man-Machine Systems, Vol. 11, No. 4, 1970, pp.190-202.
- [CCU 2005] Council Conclusions on Education and Training in the Framework of the Mid-Term Review of the Lisbon Strategy. Doc.6604/05 EDUC 29 SOC 76. Brussels, 21 February 2005. Council of the European Union, p. 9.
- [CLE 1997] Clements, P.E., Papaioannou, T., Edwards, J. Aglets: Enabling the Virtual Enterprise. In Proceedings of the International Conference 'Managing Enterprises - Stakeholders, Engineering, Logistics and Achievement' (ME-SELA '97), 1997, pp. 425-431.
- [COL 2000] Collis, J., Ndumu, D., van Buskirk, C. The Zeus Technical Manual. 2000. Pieejams: <http://labs.bt.com/projects/agents/zeus/techmanual/TOC.html> (apmeklēts: 07.04.2009).
- [COL 1996] Collinot, A., Drogoul, A., Benhamou, P. Agent Oriented Design of a Soccer Robot Team. In Proceedings of ICNAS'96. Pieejams: <http://www-poleia.lip6.fr/~drogoul/papers/Drogoul.icmas96.ps.gz> (apmeklēts: 30.09.2007).
- [DAL 2002] Dale, J., Willmott, S., Burg, B. Agentcities: Building a Global Next-Generation Service Environment, 2002. Pieejams: <http://citeseer.ist.psu.edu/dale02agentcities.html> (apmeklēts: 13.02.2008).
- [DEA 2005] de Antonio, A., Ramírez, J., Imbert, R., Méndez, G. Intelligent Virtual Environments for Training: An Agent-based Approach. In Proceedings of 4th International Central and Eastern European Conference on Multi-Agent Systems (CEEMAS'05). Budapest, Hungary, September 15-17, 2005.
- [DEB 2002] Debenham, J., Henderson-Sellers, B. Full lifecycle methodologies for agent-oriented systems the extended OPEN process framework. In Proceedings of Agent Oriented Information Systems (AOIS-2002), 2002.
- [DEL 2001] DeLoach, S. Analysis and Design Using MaSE and agentTool, In Proceedings of the 12th Midwest Artificial Intelligence and Cognitive Science Conference, Oxford OH, March 31 - April 1 2001, pp. 1-7.
- [DEV 2000] Devedzic, V., Debenham, J., Popovic, D. Teaching Formal Languages by an Intelligent Tutoring System. Educational Technology & Society, 2000, Vol. 3, No. 2. - http://ifets.ieee.org/periodical/vol_2_2000/devedzic.html (apmeklēts 18.04.2005)

- [DIX 2005] Dix, J., Zhang, Y. IMPACT: A Multi-Agent Framework with Declarative Semantics. In Multi-Agent Programming. Languages, Platforms and Applications (Eds. Bordini R.H. et al.), Springer, 2004, pp. 69-94.
- [DOR 2003] Dorça, F.A., Lopes, C.R., Fernández, M.A. A multiagent architecture for distance education systems. Proceedings of the 3rd IEEE International Conference on Advanced Learning Technologies, ICALT'03, 368.
- [ELA 1999] Elammari, M., Lalonde, W. An agent-oriented methodology: High-level and intermediate models. In Proceedings of the 1st International Workshop on Agent-Oriented Information Systems, 1999.
- [FIS 2003] Fischer, K., Schillo, M., Siekmann, J. Holonic Multiagent Systems: A Foundation for the Organisation of Multiagent Systems, LNCS 2744, Springer, 2003.
- [GEO 2003] Georgouli, K., Paraskakis, I., Guerreiro, P. A Web Based Tutoring System for Compilers. Proceedings of the 14th EAEEIE Annual Conference on Innovation in Education for Electrical and Information Engineering (EIE), Gdansk, Poland, June 16-18, 2003.
- [GER 1999] Gerber, C., Siekmann, J., Vierke, G. Holonic multi-agent systems, Technical Report R-99-03, DFKI GmbH, 1999.
- [GLA 1998] Glaser, N. The CoMoMAS Approach: From Conceptual Models to Executable Code. Pieejams: <http://citeseer.ist.psu.edu/cache/papers/cs/981/http:zSzzSzwww.loria.frzSz~glaserzSzexternzSzPUBLICATIONSzSzmaamaw97-1.pdf/the-comomas-approach-from.pdf> (apmeklēts: 25.09.2007).
- [GRU 2005] Grundspenkis, J., Anohina, A. Agents in Intelligent Tutoring Systems: State of the Art. Scientific Proceedings of Riga Technical University, 5th series, Computer Science, Applied Computer Systems, Vol. 22, Riga, RTU Publishing, 2005, pp. 110-121.
- [HOS 2003] Hospers, M., et al. An Agent-based Intelligent Tutoring System for Nurse Education. In Applications of Intelligent Agents in Health Care (J. Nealon, A. Moreno Eds.)- Basel, Switzerland: Birkhauser Publishing Ltd, 2003. – pp. 141-157.
- [IGL 1998] Iglesias C. et al. Analysis and design of multiagent systems using MAS-CommonKADS. In Intelligent Agents IV (ATAL97), LNAI 1365, Springer-Verlag, 1998, pp. 313-326.
- [JEO 2000] Jeon, H., Petrie, C., Cutkosky, M.R., JATLite: A Java Agent Infrastructure with Message Routing, IEEE Internet Computing Mar/Apr 2000.
- [KEN 1995] Kendall, E.A., Malkoun, M.T., Jiang, C.A Methodology for Developing Agent Based Systems for Enterprise Integration, IFIP Working Conference of TC5 Special Interest Group on Architectures for Enterprise Integration, Queensland, Australia, November 1995.
- [KIN 1997] Kinny, D., Georgeff, M. Modeling and Design of Multi-Agent Systems. In Agent Theories, Architectures, and Languages (ATAL), 1996. LNCS, Vol. 1193, Springer, 1997.

- [LAU 2001] Laukkanen, M., Tarkoma, S., Leinonen, J. FIPA-OS Agent Platform for Small-Footprint Devices. In Revised Papers from the 8th International Workshop on Intelligent Agents VIII. LNCS Vol. 2333, Springer, 2001. pp. 447 – 460.
- [LAV 2008] Lavendelis, E., Grundspenkis, J. Open Holonic Multi-Agent Architecture for Intelligent Tutoring System Development. In Proceedings of IADIS International Conference „Intelligent Systems and Agents 2008”, Amsterdam, The Netherlands, July 22 - 24, 2008, pp. 100-108.
- [LAV 2009a] Lavendelis, E., Grundspenkis, J. MASITS - A Tool for Multi-Agent Based Intelligent Tutoring System Development. Proceedings of 7th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 2009), Salamanca, Spain, 25-27 March 2009. Advances in Intelligent and Soft Computing Vol. 55., Springer, 2009. pp. 490-500.
- [LAV 2009b] Lavendelis, E., Grundspenkis, J. Requirements Analysis of Multi-Agent Based Intelligent Tutoring Systems. RTU Zinātniskie raksti, 5. sērija, Datorzinātne, Lietišķās datorsistēmas, 38. sējums. Izdevniecība “RTU”, Rīga, 2009, pp. 37-47.
- [LAV 2009c] Lavendelis, E., Grundspenkis, J. Design of Multi-Agent Based Intelligent Tutoring Systems. RTU Zinātniskie raksti, 5. sērija, Datorzinātne, Lietišķās datorsistēmas, 38. sējums. Izdevniecība “RTU”, Rīga, 2009, pp. 48-59.
- [LAV 2009d] Lavendelis, E., Grundspenkis, J. MASITS – A Multi-Agent Based Intelligent Tutoring System Development Methodology. In Proceedings of IADIS International Conference „Intelligent Systems and Agents 2009”, 21-23 June 2009, Algarve, Portugal, pp. 116-124.
- [LAV 2009e] Lavendelis, E., Grundspenkis, J. Multi-Agent Based Intelligent Tutoring System Source Code Generation Using MASITS Tool. Submitted at 50th RTU International Scientific Conference, Riga, Latvia, October 13-15 2009.
- [LAV 2010] Lavendelis, E., Grundspenkis, J. MIPITS - An Agent Based Intelligent Tutoring System. Submitted at the Second International Conference on Agents and Artificial Intelligence (ICAART 2010), Valencia, Spain, January 22-24, 2010.
- [LIN 2000] Lind, J. A development method for multiagent systems. In Cybernetics and Systems: Proceedings of the 15th European Meeting on Cybernetics and Systems Research, Symposium "From Agent Theory to Agent Implementation", 2000.
- [LIS 2000] Presidency conclusions. Lisbon European Council. March 23-24, 2000, 17 p.
- [LUG 2005] Luger, G.F. Artificial Intelligence: Structures and Strategies for Complex Problem Solving. Addison-Wesley, Harlow, England, 2005, 903 p.
- [NAP 2006] Latvijas nacionālais attīstības plāns 2007-2013. LR Reģionālās attīstības un pašvaldību lietu ministrija. 2006., 56 lpp. Pieejams: <http://www.nap.lv> (apmeklēts: 10.05.2009)
- [NIM 2004] Nimis, J., Stockheim, T. The Agent.Enterprise Multi-Multi-agent System In Proceedings of the Conference on Agent Technology in Business Applications (ATeBA 2004), part of the Multi-Conference on Business Information Systems (MKWI 2004) Essen, Germany, 2004.

- [ODE 2000] Odell, J., Parunak, H.V.D., Bauer, B. Extending UML for Agents. In Proceedings of the Agent-Oriented Information Systems Workshop at the 17th National conference on Artificial Intelligence, 2000.
- [PAD 2005] Padgham, L., Winikoff, M., A Practical Agent-Oriented Methodology. In Agent-Oriented Methodologies, Henderson-Sellers, B., Giorgini, P. (Eds.) Idea Group Publishing, London, 2005, pp. 107-135.
- [PAV 2005] Pavon, J., Gomes-Sanz, J.J., Fuentes, R. The INGENIAS Methodology and Tool. In Agent-Oriented Methodologies, Henderson-Sellers, B., Giorgini, P. (Eds.) Idea Group Publishing, London, 2005, pp. 236-276.
- [POK 2005] Pokahr, A., Braubuch, L., Lamersdorf, W. Jadex: A BDI Reasoning Engine. In Multi-Agent Programming. Languages, Platforms and Applications (Bordini, R.H. et al. Eds.), Springer, 2004. pp. 149-174.
- [POS 2004] Postal, A., et al. MathTutor: A Multi-agent Intelligent Tutoring Systems. In Proceedings of the 1st IFIP Conference on Artificial Intelligence Applications and Innovations, Toulouse, France, August 22-27, 2004. - Vol.1. - pp. 231-242.
- [RUP] Rational Unified Process. Best Practices for Software. Development Teams Rational Software White Paper TP026B, pieejams: http://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf (apmeklēts: 10.04.2009)
- [SAGE 1992] Sage, A.P. Systems Engineering. John Wiley&Sons, 1992, p. 608.
- [SIL 2002] Silveira, R.A., Vicari, R.M. Developing Distributed Intelligent Learning Environment with JADE – Java Agents for Distance Education Framework. In ITS 2002, Cerri, S.A., Gouardères, G., Paraguaçu, F. (Eds.), LNCS 2363, 2002, pp. 105–118.
- [SIR 2007] Sirohi, V. E-learning: An Overview. In Towards Next Generation E-government. 2007, pp. 236-240. Pieejams: http://www.csi-sigegov.org/2/24_385_2.pdf (apmeklēts: 22.05.2009).
- [SMI 1998] Smith, A.S.G. Intelligent Tutoring Systems: personal notes. - School of Computing Science at Middlesex University. – 1998, pieejams: <http://www.cs.mdx.ac.uk/staffpages/serengul/table.of.contents.htm> (apmeklēts 18.04.05).
- [STU 2002] Sturm, A., Shehory, O. Towards industrially applicable modeling technique for agent-based systems. In Proceedings of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems, Bologna, Italy, 2002, pp. 39-40.
- [SYC 2001] Sycara, K., Paolucci, M., Van Velsen, M., Giampapa, J. The RETSINA MAS Infrastructure. In the Special Joint Issue of Autonomous Agents and MAS, Vol. 7, N. 1 and 2, 2001.
- [TAV 2005] Taveter, K., Wagner, G. Towards Radical Agent-Oriented Software Engineering Process. In Agent-Oriented Methodologies. Henderson-Sellers, B., Giorgini, P. (Eds.) Idea Group Publishing, London, 2005, pp. 277-316.

- [TRI 2004] Triantis, T., Pintelas, P. An Integrated Environment for Building Distributed Multi-agent Educational Applications. In AIMS 2004, LNAI 3192, pp. 351–360, 2004.
- [VAR 1994] Varga, L.Z., Jennings, N.R., Cockburn, D. Integrating intelligent systems into a cooperating community for electricity distribution management. In International Journal of Expert Systems with Applications 7 (4) , 1994, pp. 563-579.
- [VIR 2001] Virvou, M., Tsiriga, V. Web Passive Voice Tutor: an Intelligent Computer Assisted Language Learning System over the WWW. In Proceedings of the IEEE International Conference on Advanced Learning Technology: Issues, Achievements and Challenges, Madison, WI, USA, 6-8 August, 2001. - pp.131-134.
- [WIL 2004] Willmott, S., Cortes, U., Cabanillas, D. Standards of Agent development. In Agent-Based Software Development (Luck, M., Ashri, R., D'Iverno, M., Eds.), Artech House, Norwood, MA 2004. pp. 133-166.
- [WIN 2005] Winikoff, M. Intelligent Agents: An Industrial Strength Platform. In Multi-Agent Programming. Languages, Platforms and Applications (Bordini R.H. et al. Eds.), Springer, 2004, pp. 175-196.
- [WOO 2000] Wooldridge, M., Jennings, N.R., Kinny, D. The Gaia methodology for agent-oriented analysis and design. In Journal of Autonomous Agents and Multi-Agent Systems, 2000.
- [ZHA 2003] Zhang, D., Nunamaker, J.F. Powering E-Learning In the New Millennium: An Overview of E-Learning and Enabling Technology. In Information Frontiers Vol. 5(2), 2003, pp. 207-218.