



**KAUNO TECHNOLOGIJOS UNIVERSITETAS**

**INFORMATIKOS FAKULTETAS**

**Rimantas Baranauskas**

**BŪSTO INTELEKTUALIOSIOS APLINKOS TAIKYMŲ LYGMENS  
SAUGOS SISTEMOS KŪRIMAS IR TYRIMAS**

Baigiamasis magistro darbas

**Vadovas**

Prof. Raimundas Jasinevičius

**KAUNAS, 2015**

# **KAUNO TECHNOLOGIJOS UNIVERSITETAS**

## **INFORMATIKOS FAKULTETAS**

### **KOMPIUTERIŲ KATEDRA**

TVIRTINU

Katedros vedėjas

(parašas) Prof. dr. Algimantas Venčkauskas

(data)

## **BAIGIAMOJO PROJEKTO PAVADINIMAS**

Baigiamasis magistro darbas

**Informacijos ir informacinių technologijų sauga (kodas 621E10003)**

### **Vadovas**

(parašas) Prof. Raimundas Jasinevičius

(data)

### **Recenzentas**

(parašas) Doc. dr. Agnius Liutkevičius

(data)

### **Projektą atliko**

(parašas) Rimantas Baranauskas

(data)

**KAUNAS, 2015**



KAUNO TECHNOLOGIJOS UNIVERSITETAS

Informatikos fakultetas

---

(Fakultetas)

Rimantas Baranauskas

---

(Studento vardas, pavardė)

Informacijos ir informacinių technologijų sauga, 621E10003

---

(Studijų programos pavadinimas, kodas)

Baigiamojo projekto

„Būsto intelektualiosios aplinkos taikymų lygmens saugos sistemos kūrimas ir tyrimas“

**AKADEMINIO SAŽININGUMO DEKLARACIJA**

20 \_\_\_\_\_ m. \_\_\_\_\_ d.

\_\_\_\_\_ Kaunas \_\_\_\_\_

Patvirtinu, kad mano, **Rimanto Baranausko**, baigiamasis projektas tema „.....“ yra parašytas visiškai savarankiškai, o visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

Aš suprantu, kad išaiškėjus nesąžiningumo faktui, man bus taikomos nuobaudos, remiantis Kauno technologijos universitete galiojančia tvarka.

---

(vardą ir pavardę įrašyti ranka)

---

(parašas)

Baranauskas, R. „Būsto intelektualiosios aplinkos taikymų lygmens saugos sistemos kūrimas ir tyrimas“. Informatikos inžinerijos magistro baigiamasis projektas / vadovas prof. Raimunas Jasinevičius; Kauno technologijos universitetas, informatikos fakultetas, kompiuterių katedra.

Kaunas, 2015. 48 psl.

## **SANTRAUKA**

Magistrantūros studijų baigiamajame darbe atliktas būsto intelektualiosios aplinkos taikymų lygmens saugos sistemos kūrimas ir tyrimas.

**Darbo objektas** – būsto intelektualiosios aplinkos modelis realizuotas miglotąja logika bei šios aplinkos taikymų lygmens saugos sistema.

**Darbo metodai:** intelektualumas formuojamas miglotąja logika su modifikuotu Wang ir Mendel taisyklių generavimo metodu. Sprendimo analizė atliekama palyginant BIA veikimą be saugos sistemos ir su ja įvertinant taisyklių kiekį bei sistemos sugebėjimą prisitaikyti prie kintančio naudotojo elgesio.

**Darbo rezultatai.** Apibrėžta intelektualumo sąvoka, atlikta panašių tyrimų analizė. Pasirinkus intelekto formavimo būdą (miglotoji logika) sukurtas BIA modelis. Remiantis šiuo modeliu suformuota saugos sistemos funkcinė organizacija. Apsimokymo procesui įgyvendinti panaudotas Wang ir Mendel taisyklių generavimo metodas su papildomu taisyklių įsiminimo koeficientu. Nenaudojant saugos sistemos taisyklių buvo generuojama daugiausiai, dėl to mažėjo efektyvumas. Nešalinant nenaudojamų taisyklių sistema nustojo tenkinti naudotojo poreikius nes vadovavosi ir senomis ir naujomis taisyklėmis. Įdiegus saugos sistemą taisyklių kiekis tapo priklausomas nuo naudotojo periodiškai atliekamų veiksmų kiekio – buvo naudojamos tik aktualiausios taisyklės. Pastebėta, jog sistemos prisitaikymas prie pakitusio elgesio tiesiogiai priklauso nuo atliekamų veiksmų periodiškumo, bei kaip reikšmingu laikomas naudotojo elgesys.

**Raktažodžiai:** miglotoji logika, intelektika, sauga, būsto intelektualioji aplinka, mašininis mokymasis.

Baranauskas, R. ,, Design and implementation of intellectual home environment application layer security system“. Master's thesis / supervisor Prof. Raimundas Jasinevičius; Department of Computer Science, Faculty of Informatics, Kaunas University of Technology.

Kaunas, 2015. 48 p.

## SUMMARY

The master thesis presents development and implementation of intellectual home environment application layer security system.

**Object of the work** – intellectual home environment and its application layer security system.

**Method of the work** – intellect of system entities is implemented using fuzzy logic and modified Wang and Mendel fuzzy rule generation method. Results are presented by comparing performance of intellectual home environment model vs. the same model with security system. Key characteristics used in comparison are rule count and ability to adjust itself to match the changing user behavior.

**The results of the work.** The concept of intellect in such system is defined by analyzing similar research documents. Intellectual home environment model is implemented using fuzzy logic. Machine learning when using fuzzy logic is implemented using modified Wang and Mendel method. This method is enhanced by adding rule memorization coefficient. The worst system performance was achieved when not using security system. Large amount of new and old (deprecated) rules were generated which resulted in system actions being incorrect and failure to adapt to changing user behavior. System results were better after adding the security system. There were less rules and only actually important and used rules were kept in the memory. System ability to adapt to changing user behavior depends on periodically performed actions and importance of these actions to the system.

**Key words:** fuzzy logic, intellectics, security, intelligent home environment, machine learning.

## TURINYS

Įvadas .....	8
1. Probleminės srities analizė.....	10
1.1. Tiriamoji aplinka, taikymai.....	10
1.2. Taikymų sauga.....	13
1.3. Intelektualumas .....	18
1.4. Teorinio problemos nagrinėjimo išvados.....	26
2. Sistemos projektavimas.....	27
2.1. Virtualaus BIA modelio projektavimas .....	27
2.2. BIA taikymų lygmens saugumo sistemos projektavimas .....	34
2.3. Projektavimo išvados .....	36
3. BIA taikymų lygmens saugumo sistemos tyrimas.....	38
Išvados .....	45
4. Literatūra.....	47
5. Priedai .....	50
5.1. priedas. Straipsnis .....	50
5.2. priedas. Kompaktinė plokštelė.....	60

## PAVEIKSLŲ SĄRAŠAS

1 pav. Būsimos sistemos koncepcinis modelis .....	12
2 pav. Principinė sistemos aktorių ir jų sąsajų schema [12] .....	17
3 pav. Miglotosios priklausomybės funkcijos.....	22
4 pav. Verbalizavimas.....	22
5 pav. Būsto intelektualiosios sistemos modelio principinė schema .....	27
6 pav. Hierarchinis BIA modelis.....	28
7 pav. Koncepcinis veiklos objektų modelis.....	29
8 pav. Mąstymo agento būsenų modelis .....	30
9 pav. BIA komponentų modelis .....	31
10 pav. BIA esybių principinė schema .....	32
11 pav. Veiksmų registro sandara .....	32
12 pav. Elgsenų tipai.....	33
13 pav. Kintamojo hod miglotoji išraiška.....	33
14 pav. BIA diegimo diagrama .....	38
15 pav. Kompiuterinio modelio grafinė sąsaja .....	39
16 pav. FCL aprašyto scenarijaus sandara .....	40
17 pav. Paros meto grafikas .....	41
18 pav. Savaitės dienos grafikas .....	41
19 pav. Rezultato grafikas.....	41
20 pav. Pirmojo eksperimento taisyklių generavimas .....	42
21 pav. Antrojo eksperimento taisyklių generavimas .....	43
22 pav. Trečiojo eksperimento taisyklių generavimas.....	44

## IVADAS

Baigiamasis magistro darbas „Būsto intelektualiosios aplinkos taikymų lygmens saugos sistemos tyrimas“ priklauso informatikos ir informacinių technologijų saugos studijų programai.

Interneto ir informacinių technologijų išplitimas ir vystymas privedė prie daiktų interneto (DI) atsiradimo. DI apibrėžiamas kaip daiktų, egzistuojančių išmaniojoje erdvėje, turinčių tapatybes ir virtualias asmenybes, visuma, kurie naudoja intelektualiąsias sąsajas komunikacijai socialinėje, erdvės ir naudotojo aplinkose. Manoma, jog tai yra sekantis interneto evoliucijos žingsnis. Prognozuojama, jog DI susiformuos per artimiausius 20 metų ir taps dar viena išplitusia, visur esančia technologija. Ši koncepcija apima ateities įrenginių ir sistemų integravimą į vieną sistemą. Šiuo metu tai yra viena iš svarbių tyrimo kryptų pasaulyje, kuri akcentuojama „Horizon 2020“ programoje Europos sąjungoje, „Smart grid“ ir „Smarter planet“ JAV, „1-Japan“ Japonijoje, „U-Korea“ Pietų Korėjoje, „Sensing China“ Kinijoje. [1 – 4].

Europos programoje Horizon 2020 DI vystymo skyriuje akcentuojamas nepakankamas technologinis lygis, kad būtų sukurta tokia sistema, esminės problemos – komunikacijos infrastruktūra ir energijos ištekliai. Energijos išteklių problemas planuojama spręsti naudojant mažai energijos reikalaujančias mikroschemas, bei kuriant tobulesnes energijos talpyklas ir išgavimo būdus. Į komunikacijos problemas įeina ir naujų vieningų komunikacijos protokolų sukūrimas ir techninių informacijos perdavimo priemonių kūrimas. Nepaisant to, vyksta mažesnės apimties DI tyrimai ir taikymai. Tuomet išryškėja daugiau kliūčių – privatumas, saugumas. Yra teigiama, jog tam, kad DI taptų realybe, visų pirma turi būti įdiegtos privatumą ir saugumą užtikrinančios priemonės, kitaip ši technologija tiesiog nebus priimta visuomenės. Numatoma, jog tai būtų galima spręsti naudojant programinių ir technologinių sprendimų derinį. Šiuo metu nėra sukurta priežiūros institucijų, atsakingų už DI, nėra vieningų standartų, kurių laikantis būtų galima taikyti DI globaliu mastu. Panaši situacija buvo ir interneto atveju – kol taikomoji sritis buvo pakankamai maža, nereikėjo rūpintis saugumu ir privatumu. Problemos išryškėjo tik jam išplitus, o tuomet teko skubiai taisyti atsiradusias spragas. Skiriamos trys charakteristikos, kuriomis pasižymi DI: aplinkos duomenų surinkimas, gebėjimas jais apsikeisti ir intelektualus operavimas duomenimis [5]. Nors intelektika ir skamba kaip vienas iš būsimų privalumų, galinčių sukelti teigiamus pokyčius saugumo, rizikos valdymo ir finansiniu atžvilgiu tačiau kartu nepamirštama ir tai, jog šio tipo sistemas yra itin sudėtinga kurti.

Europos DI darbų programoje skiriamos dvi saugos plėtojimo kryptys – kriptografinė sauga ir sistemos architektūros sauga. Kriptografijoje informacijos saugai užtikrinti orientuojamasi į tokius aspektus kaip konfidencialumas, integralumas, bei duomenų ar esybių autentifikacija [30]. Sistemos architektūros saugoje orientuojamasi į šiuos aspektus:



- nepriklausymas nuo platformos ir saugumo užtikrinimas visuose techniniuose ir programiniuose sistemos lygmenyse;
- veikimo konteksto suvokimas ir gebėjimas adaptuotis kintant kontekstui;
- apsaugos nuo dabartinių grėsmių teikimas, bei apsauga nuo netinkamo panaudojimo ar netinkamos konfigūracijos.

Apibendrinant – yra siekiama, kad naudotojai ne tik būtų apsaugoti nuo grėsmių, bet ir įgautų pasitikėjimą šio tipo sistemomis nuo pat jų įdiegimo momento [1].

Svarbu atskirti dvi sąvokas – išmanumas ir intelektualumas. Šias sąvokas galima iliustruoti šildymo sistemos pavyzdžiu. Išmanios šildymo sistemos pavyzdys – programuojamas termostatas – šildoma kai vėsu, pasiekus norimą temperatūros ribą, šildymas išjungiamas, naudotojas gali sudaryti grafiką, kada temperatūra turėtų būti aukštesnė, o kada žemesnė. Intelektualios sistemos pavyzdys – šildymo sistema stebi ir įsimena naudotojo įpročius, kada jis būna patalpose, seka temperatūros pokyčius, naudotojo finansinę padėtį, ir remdamasi sukaupta informacija autonomiškai priima sprendimus, susijusius su šildymo reguliavimu. Būsto intelektualioji aplinka minima kaip viena iš potencialių DI taikymo sričių [1, 2. 11 - 15]. Akcentuojama tokių funkcijų, kaip energijos naudojimo sekimo ir optimizavimo svarba [15]. Fizinė aplinka, kurioje duomenims surinkti yra įdiegta technologinės priemonės dar yra vadinama išmaniaja aplinka [1]. Nėra tikslaus apibrėžimo kas yra intelektualumas, tačiau literatūroje išskiriami šie žmogaus veiksmai, kurie yra laikomi intelektualiais [30]: situacijų atpažinimas ir klasifikavimas, veikimas vadovaujantis miglotomis taisyklėmis, veikimas vadovaujantis suformuota tendencija. Darbe pasiūlytas sprendimas yra skirtas pagerinti būsto intelektualiosios sistemos taikymų lygmens saugumą.

Darbą sudaro trys dalys. Probleminės srities analizėje atlikta literatūros šaltinių analizė. Analizė buvo atlikta šiose darbu aktualiose srityse: tiriamoji aplinka bei jos komponentų taikymai, taikymų sauga, bei intelektualumas ir jo formavimas. Sistemos projektavimo dalis atlikta dviem etapais – pirmiausia buvo suprojektuotas virtualus BIA modelis. Remiantis šiuo modeliu įgyvendintas BIA taikymų lygmens saugos sistemos projektavimas. Trečiojoje dalyje atliktas BIA taikymų lygmens saugumo sistemos tyrimas.

## 1. PROBLEMINĖS SRITIES ANALIZĖ

Šioje dalyje atliekama literatūros ir esamų sprendimų analizė, kurios tikslas yra suformuoti pakankamą dalykinės srities žinių bazę, kad būtų galima apibrėžti tyrimo tikslą bei iškelti tolimesnius darbo uždavinius. Šios dalies uždaviniai yra:

- atlikti tiriamosios srities analizę;
- atlikti panašių tyrimų ir sprendimų analizę;
- suformuoti darbo temą ir uždavinius.

### 1.1. Tiriamoji aplinka, taikymai

Darbo tiriamoji aplinka yra būsto intelektualiosios aplinkos taikymų lygmuo. Literatūroje intelektualusis būstas išskiriamas kaip viena iš svarbių DI ir Aml panaudojimo sričių. DI architektūrą yra siūloma skaidyti į tris lygmenis [7, 8, 11, 13 - 18]:

- suvokimo lygmuo, sudarytas iš daiktų, jutiklių ir vykdyklių;
- tinklo lygmuo, sudarytas iš duomenų perdavimo kanalų ir jungiamųjų mazgų;
- taikomasis lygmuo, sudarytas iš paslaugų.

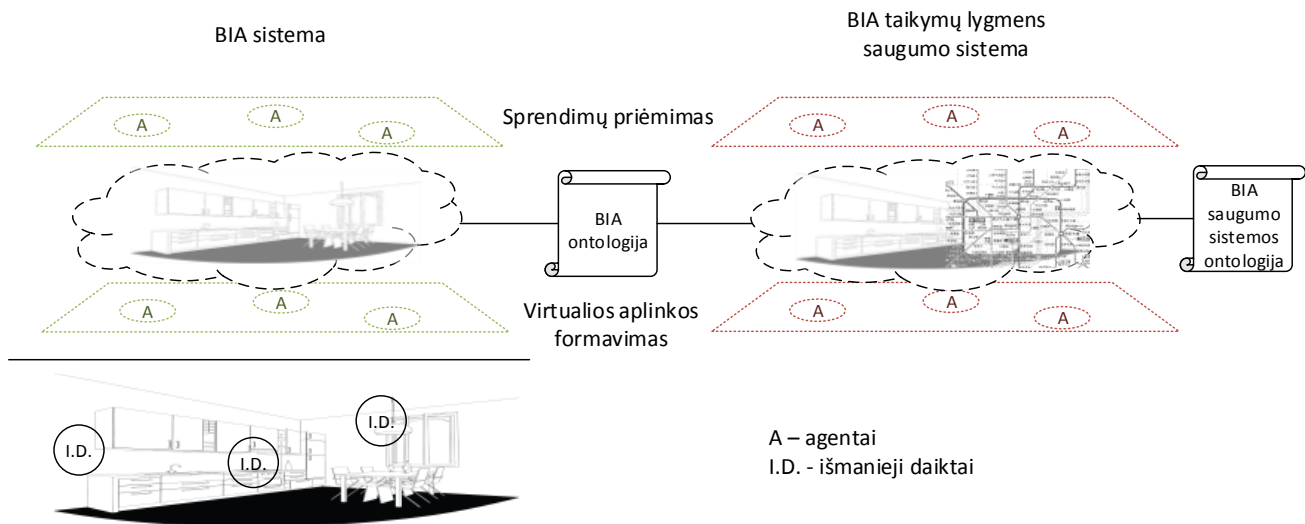
DI aplinką sudaro fizinė dalis – būsto aplinka ir virtualioji dalis – būsto aplinkos intelektika. Jungiamoji fizinės ir virtualiosios aplinkos dalis yra jutikliai, kurių dėka fiziniai veiksniai yra transformuojami į skaitmeninę formą ir vykdykliai, kuriais galima keisti fizinių objektų būseną. Komunikaciją užtikrina tinklo lygmuo. Šio lygmens paskirtis yra atskirų elementų sujungimas, užtikrinant galimybę efektyviai ir ekonomiškai keisti informacija [8]. Fizinė dalis yra sudaryta iš išmaniųjų prietaisų, kurie suteikia sąsają, per kurią galima gauti informaciją ir valdyti šiuos prietaisus. Taigi išmanusis daiktas yra bazinis fizinės sistemos blokas, kurio duomenimis formuojama virtualioji aplinka, ir kurie suteikia galimybę ją įtakoti. Būsto aplinkos intelektika veikia suvokdama situacijas, kurias generuoja būsto aplinka ir dalyvauja naujų situacijų generavime paveikdama fizinę aplinką.

Virtualioji aplinka yra sudaryta iš esybių – agentų, kurie šioje sistemoje bendradarbiauja ir atlieka jiems suformuotus tikslus. Aukščiausias sistemos tikslas – padėti žmogui. Nėra vieningo apibrėžimo, kas yra intelektualusis agentas. Literatūroje jis apibūdinamas kaip esybė, pasižyminti tokiomis savybėmis kaip autonomiškumas, socialumas, reaktyvumas, proaktyvumas [31], ar paprasčiau – esybė, jaučianti aplinką jutikliais, bei įtakojanti ją vykdykliais. Fizinius – virtualius agentus pagal turimas savybes siūloma grupuoti į 5 kategorijas [23] (žemesniojo lygio savybės galioja aukštesniuose):

1. bazinės savybės – egzistavimas fizinėje ir virtualioje formoje, tapatybės turėjimas, galimybė keistis informacija, naudojant numatytus protokolus;
2. įprastinės savybės – naudojimas paslaugomis per suprantamas sąsajas, varžymasis su kitais dėl paslaugų ar resursų bei sąveika su aplinka;
3. socialumas – bendravimas tarpusavyje ir su naudotojais, bendradarbiavimas kuriant grupes ir naujus tinklus, komunikacijos iniciavimas;
4. autonomiškumas – autonomiškas veiksmų atlikimas, aplinkos suvokimas ir prisitaikymas prie jos, gebėjimas formuoti aplinkos modelį, arba istorijos kaupimas ir mokymasis iš jos, autonominis sprendimų priėmimas, informacijos suvokimas – vertimas žiniomis;
5. gyvavimo laiko valdymas naikinant, valdant kitas esybes ir dauginantis.

Kiekviena šių grupių savo savybėmis reikalauja vis aukštesnio intelekto lygio – didėja ir sudėtingumas, ir atsakomybė, kuri atsiranda dėl galimybės atlikti tam tikrus veiksmus. Darbe bus laikoma, jog intelektualusis agentas yra agentas, pasižymintis savybėmis, kurios žmoguje laikomos intelektualiomis [30]: gebėjimas atpažinti situacijas ir jas klasifikuoti į kategorijas; veikimas vadovaujantis miglotomis taisyklėmis; veikimas vadovaujantis suformuota tendencija.

Virtualioji aplinka yra aprašoma ontologija. Ontologija yra apibrėžiama kaip darinys, kurio paskirtis yra leisti modeliuoti žinias apie aplinką [10]. Tam, kad gauti žinias yra reikalinga informacija, o iš fizinės aplinkos yra gaunami tik duomenys. Visi aktualūs fiziniai objektai virtualioje aplinkoje yra informacijos rinkiniai apie dabartinę būseną, ar būsenų istoriją. Žvelgiant hierarchiškai, žemiausio lygmens agentai transformuoja duomenis į informaciją – atlieka virtualių objektų formavimą. Aukštesniajame lygmenyje agentai remdamiesi ontologija ir esama informacija atlieka žinių modeliavimą, kurių dėka geba priimti sprendimus. Šiame darbe virtualioji aplinka turi du sluoksnius – būsto aplinkos intelektika ir saugumo intelektika.



**1 pav. Būsimos sistemos koncepcinis modelis**

Nėra aiškaus apibrėžimo ką DI elementai turėtų atlikti – tai vienareikšmiškai priklauso nuo konteksto, tačiau išskiriami keli bruožai, kurie turėtų būti būdingi visiems – tai žinių apie save ir kitus turėjimas ir gebėjimas jomis dalintis bei operuoti. Fiziniai daiktai vienu metu egzistuoja ir fiziniėje ir elektroninėje erdvėje, todėl jie tuo pačiu metu gali dalyvauti įvairiuose veikimo scenarijuose, būti skirtingose būsenose, ar netgi turėti skirtingas tapatybes.

Literatūroje DI taikymo galimybės pagal paskirtį išskiriamos į dvi dalis: informacijos rinkimas ir analizė, bei valdymas ir automatizacija [4, 5]. Informacijos rinkimo ir analizės kategorija skaidoma į tris taikymų sritis:

- elgesio (būsenos kitimo) sekimą – stebimi gali būti daiktai, žmonės, ar informacija;
- augmentedotą situacijos suvokimą – aplinka yra suvokiama artimu realiam laikui;
- sprendimų priėmimą remiantis surinktomis ir suvoktomis žiniomis.

Valdymo ir automatizacijos kategorija taip pat skaidoma į tris taikymų sritis:

- procesų optimizaciją – jų automatizuotą valdymą uždarose sistemose;
- sunaudojamų resursų optimizaciją;
- sudėtingas autonomines sistemas.

Būsto intelektualioji sistema patenka į abi didžiąsias kategorijas – informacijos rinkimo metu stebima aplinka, aplinka yra suvokiama artimu realiam laikui ir remiantis surinkta informacija yra priimami sprendimai. Priimtų sprendimų rezultatas gali būti būste esančių prietaisų valdymas. Be to, atsiranda galimybė optimizuoti valdomus procesus jų vykdymo, išteklių sunaudojimo atžvilgiu.

Konkrečius taikymo atvejus [1, 2, 6, 14] galima sugrupuoti į tris sritis:

- pramonės – finansinės ir komercinės operacijos tarp organizacijų ir asmenų;

- aplinkos – operacijos susiję su gamtiniais ištekiais;
- visuomenės – operacijos susiję su žmonėmis, jų bendruomenėmis.

Į visuomenės sritį patenka būsto intelektualiosios aplinkos sistema.

## 1.2. Taikymų sauga

DI yra kuriamas ant esamų interneto pamatų, taigi, išlieka esamos saugos problemos, bei atsiranda naujų. Saugumo sistemos uždavinys yra ne tik apsaugoti naudotojus nuo grėsmių, bet ir pasiekti, jog naudotojai įgautų pasitikėjimą sistemomis nuo pat jų įdiegimo momento [1]. Pastebima, kad DI tradiciniai saugos metodai neapima viso kylančio grėsmių spektro [1, 2]. Bendruoju atveju yra keliamos trys kliūtys, kurios trukdo užtikrinti saugumą [7, 11, 14]:

- srities, kurios saugumą reikia užtikrinti, plėtra (fiziniai objektai yra atvaizduojami virtualioje erdvėje, plečiasi tinklai, atsiranda naujos sąsajos tarp vidinių ir išorinių objektų);
- DI elementų dinamiškumas (kiekvienas elementas vienu metu gali dalyvauti keliose skirtingose veiklose, būti skirtingose būsenose jų atžvilgiu);
- veiksmai yra atliekami tarp įvairialyčių elementų.

CIA (confidentiality, integrity, availability) trikampio, kuris buvo kompiuterių saugos pramonės standartas nuo universaliųjų kompiuterių laikų [9] nebepakanka [12]. Atsiradus intelektualiams ir autonomiškiems prietaisams reikalingi nauji sprendimai, galintys užtikrinti saugumą, patikimumą ir privatumą [11, 12]. Saugumo grėsmės DI pagal atliekamus veiksmus yra grupuojamos į keturias sritis [7]:

- duomenų srauto analizė, pasiklausymas, duomenų modifikavimas – prarandamas konfidencialumas, vientisumas;
- impersonacija, pakartotiniai užklausų siuntimai – gali būti apeinama identifikacija, autentifikacija ir autorizacija, prarandamas konfidencialumas;
- paslaugų tiekimo sutrikdymas (įvairios DoS atakos), kenkėjiškos programos – prarandamas prieinamumas, gali būti prarandamas konfidencialumas;
- privatumo atakos – prarandamas konfidencialumas.

Šiuo atveju yra atsižvelgiama vien tik į informacijos saugą, ir visiškai ignoruojama funkcionavimo sauga. Kadangi grėsmės yra klasikinės, joms sumažinti yra siūlomi klasikiniai metodai, tokie, kaip kriptografija ar specialios programų sistemos.

Saugumo grėsmių analizė pagal architektūrinius DI lygmenis yra geresnis pasirinkimas, kadangi galima detaliai analizuoti atakos paviršių kiekvienam lygmeniui [7, 8, 9, 11, 14, 18]. Pagrindinis suvokimo lygmens elementas yra išmanusis daiktas, turintis jutiklius ir vykdyklius. Šio

lygmens problemos – objektų apsauga nuo nesankcionuoto fizinio poveikio, bei informacijos rinkimo proceso saugumo užtikrinimas [13]. Informacijos rinkimo proceso metu galima įterpti papildomų žingsnių, ar modifikuoti esamus. Taip informacija gali būti prarasta, modifikuota ar generuojama iš šalies.

Dėl elementų daugialypiškumo ir standartizavimo nebuvimo yra sudėtinga sukurti daugeliui atvejų tinkančią saugos politiką. DI elementai gali būti diegiami sunkiai prieinamose vietose, arba, priešingai, lengvai prieinamose vietose. Pirmuoju atveju juos gali būti sunku arba neįmanoma aptarnauti tiek fiziškai tiek virtualiai. Galimas sprendimo būdas – autonomiškai veikiantys išmanieji prietaisai, tačiau šiuo atveju, jei atliekami veiksmai yra sudėtingi ir reikalaujantys informacijos analizės bei sprendimų priėmimo, reikalinga visa aibė naujų saugos užtikrinimo priemonių, kad nebūtų nukrypta nuo siekiamo tikslo. Jei DI elementas bus lengvai prieinamoje vietoje, jį gali būti neįmanoma apsaugoti nuo fizinių tyčinių ar netyčinių pažeidžiamumų. Šiuo atveju yra svarbu, jog globali sistema, kurios dalis jis yra, sugebėtų atpažinti netinkamai veikiančią prietaisą ir atlikti savo konfigūracijos pakeitimus, kurie sistemai leistų veikti teisingai.

Tam, kad užtikrinti fizinį saugumą, priklausomai nuo situacijos, gali reikėti atsparumo drėgmei ar temperatūros pokyčiams. Yra bendrų bruožų [8], kuriais turėtų pasižymėti visos DI esybės: jos turi būti atsparios kopijavimui, modifikavimui, bei pakartotiniam atskirų dalių panaudojimui. Pavyzdys – pakartotinai panaudojamos egzistuojančios sąsajos, bet pakeičiama veikimo logika. Palyginus su programiniais apsaugos metodais, fiziniai apsaugos metodai yra brangūs, ir juos ne visada galima pritaikyti [13].

Yra svarbu apsaugoti informacijos rinkimo procesą. Suvokimo lygmenyje galimas neautorizuotas priėjimas prie duomenų su tikslu juos sužinoti, ar pakeisti [7, 8, 18]. Sprendimas identifikacijos – autentifikacijos – autorizacijos veikimo užtikrinimas. Tačiau problema tampa sudėtingesnė, kai komunikacija vyksta tarp DI esybių – reikalingas būdas, kaip jos galėtų autentifikuotis viena kitai. Tam, kad būtų pastebėtas duomenų modifikavimas, ar duomenų srauto modifikavimas yra siūloma vientisumo kontrolė. Apibendrinant – turi būti diegiamos kriptografinės saugumo užtikrinimo priemonės. Tokios grėsmės, kaip DoS sumažinimo sprendimai yra abstraktūs. Be tradicinių priemonių, tokių kaip ugniasienės, įvairūs filtravimo metodai ir IDS bei IPS sprendimai [7, 8, 14], siūloma dar du variantai – padaryti sistemą, kuri būtų jiems atspari, arba sukurti mechanizmą, gebantį aptikti šaltinius ir eliminuoti arba sumažinti jų daromą įtaką [8]. Tai yra panašu į sistemos koncepciją, kai saugumas yra įkomponuojamas projektavimo metu [1]. Šiuos sprendimus gali būti sudėtinga įgyvendinti dėl fizinių elementų savybių – fiziniai įrenginiai gali neturėti pakankamai skaičiavimo ir energijos resursų.

Tinklo lygmenyje vykdomas apsikeitimas duomenimis. Šiame lygmenyje galima neteisėta prieiga prie duomenų, jų rinkimas, modifikavimas. Taip pat, piktavališkas gali apsimesti suvokimo lygmens objektu ir perimti duomenis, gauti priėjimą prie kitų elementų, ar naudotojų. Šiame lygmenyje atsiranda ir programinės grėsmės, tokios, kaip virusai ir kitos piktavališkos programos. Yra siūloma taikyti kriptografinius mechanizmus grėsmėms sumažinti – šifravimą, skaitmeninius parašus, vientisumo ir autentiškumo užtikrinimo algoritmus [8]. Į DI elementus galima įdiegti antivirusinę sistemą [8], tačiau akivaizdi problema – elementai dažnai turi mažai resursų, todėl antivirusinėms sistemoms jų gali nepakakti, be to, turi būti užtikrinamas būdas, kaip ją nuolatos atnaujinti, o tai taip pat gali būti neįmanoma – elementai nebūtinai visuomet yra aktyvūs dėl energijos sąnaudų sumažinimo.

Taikomajame lygmenyje yra realizuojamos paslaugos. Jos sudarytos iš pavienių elementų teikiamų paslaugų, ar atliekamų funkcijų. Šis lygmuo yra labai įvairiai suprantamas, priklausomai nuo žemesnių lygmenų sandaros. Pavyzdžiui, siūloma atsižvelgti vien tik į duomenų saugumą ir privatumo užtikrinimą [13]. Grėsmės šioms sritims siūloma spręsti panaudojant prieigos valdymą ir šifravimą ir programinėje ir techninėje įrangoje. Tačiau požiūris privalo būti platesnis, kadangi būtent šiame lygmenyje vyksta operavimas duomenimis ir priimami sprendimai atlikti funkcijas, susijusias su tiekiamomis paslaugomis. Svarbu ne vien sistemos apsauga nuo neautorizuoto išorinio poveikio – žalos šaltiniu gali tapti ir autorizuotas išorinis poveikis, bei sistemos situacijos, kylančios iš sistemos vidaus jos veikimo metu. Abiem atvejais gali būti sutrikdytas norimas paslaugų tiekimas. Literatūroje dažniausiai daroma prielaida, jog autorizuota įtaka yra teisinga, o neįtakojama sistema standartiškai taip pat veiks korektiškai. Eksperimentinių DI taikymų metu komponentai yra kruopščiai parenkami, kad būtų suderinami tarpusavyje. Teigiama, jog yra svarbu sukurti tokius elementus, kurie „įtraukti“ į sistemą susikonfigūruotų patys [2]. Bendroju atveju konfigūracija gali būti trejopa: gamyklinė, naudotojo ir daikto savikonfigūracija.

Gamyklinė konfigūracija turi būti tokia, kad ja būtų galima pasitikėti. Likusiems atvejams turi būti sukurtas būdas, leidžiantis išvengti netinkamos konfigūracijos būsenos, kuri gali trikdyti tiek individualaus elementu, tiek visos sistemos funkcionavimą, ko pasekoje būtų iškreiptas ar visai nutrauktas paslaugos tiekimas.

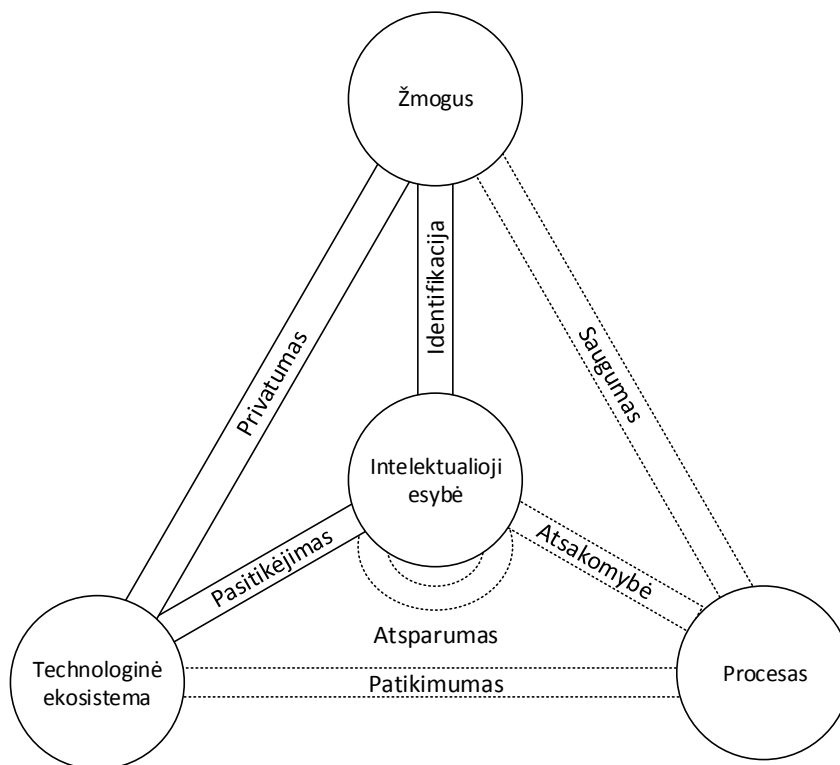
Klaidingai veikiantys daiktai, nepriklausomai nuo priežasties – dar viena grėsmė [2]. Klaidingai veikiantis daiktas gali ir kenkti sistemos funkcionavimui, ir „užkrėsti“ kitus, perduodamas jiems klaidingą informaciją. Reikalingos priemonės, kuriomis galima būtų identifikuoti netinkamą veikimą ir atitinkamai reaguoti, pavyzdžiui, ignoruoti, perspėti žmogų ar kitus įrenginius.

Dar viena grėsmė, kuri iškyla, kai kalbama apie intelektualiujų aplinkų sukūrimą tai jų esybių patikimumas. Ši sąvoka intelektualumo kontekste apima tikslo „gerumą“, kai sprendimai yra priimami bendradarbiaujant, bei prioriteto teikimas globaliems tikslams prieš lokalius tikslus. T.y. kiekviena DI esybė turi savo veikimo tikslą, tačiau kartais jam pasiekti yra reikalinga atlikti veiksmus, nesusijusius su numatytais, kad būtų galima tęsti numatytus veiksmus.

Blogos konfigūracijos ir klaidingai veikiančių esybių aptikimas tampa labai sudėtingu dėl jų įvairovės. Naudotojai įprastai nesirenka daiktų specialiai tam, kad būtų idealiai suderinti. Kitas faktorius – laikas. Viena lokali sistema gali būti sudaryta ir esybių, sukurtų skirtingu metu. Seniau sukurtos esybės gali nieko nenumanyti apie tas, kurios buvo sukurtos vėliau. Būtent dėl šios priežasties tampa neįmanoma numatyti visų galimų jų derinių, veikimo scenarijų ir situacijų, kurios gali susidaryti. Viską dar labiau apsunkina tai, jog ta pati esybė vienu metu gali egzistuoti skirtinguose kontekstuose, todėl ją keičiant, kad tiktų vienam kontekstui gali sugriūti jos veikimas kitame kontekste. Intelektualumu pasižyminčių sistemų pranašumas prieš tradicines [6] – joms yra nebūtinai detalus išankstinis aplinkos ir joje vykstančių procesų aprašas. Sprendimai yra priimami remiantis susidarančiomis situacijomis, pavyzdžiui, laikantis miglotų taisyklių, ar vertinant pagal iš anksto suformuotus situacijų pavyzdžius. Taigi, nors intelektualias sistemas ir sudėtinga sukurti, ilgainiui jos gali atnešti naudos, kadangi „griežtai“ suprogramuotos sistemos sukurti dėl daug nežinomųjų paprasčiausiai yra neįmanoma, vienintelė galimybė – ją nuolat atnaujinti.

Trečiasis saugumo analizavimo būdas yra išskirti sistemos aktorius ir įvertinti jų tarpusavio sąsajų problemas (2 pav).





**2 pav. Principinė sistemos aktorių ir jų sąsajų schema [12]**

Žmogaus (žmogiškųjų resursų) atsakomybės yra apibrėžti saugumo politiką, atlikti auditą ir remiantis jo rezultatais suformuoti reikalingus pakeitimus, bei teikti sistemos palaikymą eksploataavimo metu. Kad tai atliktų žmogus turi turėti priėjimą prie DI, jo veikimo konteksto ir gebėti atlikti veikimo analizę. Procesu apibrėžiama sistemos užduočių vykdymas. Proceso sauga turi būti reglamentuota standartais, politikomis ir kitais dokumentais. Literatūroje pabrėžiama, jog ypatingai svarbu pasiekti balansą tarp saugumo procesų sudėtingumo ir reikiamo saugos lygio. Technologinė ekosistema tai sprendimų visuma, kurie buvo priimti užtikrinti saugai (fiziniai, konfigūraciniai, identifikacija, autorizacija ir t.t.). Intelektualusis objektas tai DI esybė, kuri operuoja informacija, bendrauja ir bendradarbiauja su kitais objektais, reaguoja į aplinkos būsenos pokyčius, priima sprendimus. Šio elemento saugumas turi būti įdiegtas intelektikos lygmenyje.

Anksčiau nagrinėtuose šaltiniuose buvo orientuojamasi į tamsesnėmis linijomis pažymėtas sąsajas. To priežastis gali būti tai, jog ir intelektualumas ir DI vykstantys procesai nėra griežtai apibrėžti standartais ir normomis. Nepaisant to, teigiama jog yra būtinas platesnis požiūris. Anksčiau nagrinėtuose šaltiniuose tai taip pat nėra pamirštama, tačiau apsiribojama užuominomis, jog šis aspektas yra svarbus. Blankesnės spalvos linijomis pažymėti ryšiai, kurie yra nepakankamai išnagrinėti [12]:

- atsakomybė – prieigos kontrolė, autorizacijos mechanizmas, bei galimybė lanksčiai valdyti taisykles;

- atsparumas – intelektualaus objekto sugebėjimas suvokti kas yra „gerai“, ką objektas yra autorizuotas daryti, remiantis konteksto situacija, į kurią jis pateko;
- saugumas – procesų atsparumas kai vienas ar daugiau elementų pradeda neveikia ar veikia netinkamai;
- patikimumas – duomenų ir komunikacijos valdymas, automatinės priemonės, didinančios procesų patikimumą;

Šiuo požiūriu darbe bus orientuojamasi į atsparumą – kuriama saugumo sistema, gebanti suvokti kas yra gerai, remiantis aplinkos situacija. Tai yra mažai nagrinėjama, tačiau svarbi sritis.

### 1.3. Intelektualumas

Literatūroje yra apibrėžiamos tokios DI elementų savybės kaip autonomiškumas, prisitaikymas prie aplinkos, duomenų rinkimas ir jų interpretavimas, žinių formavimas, sprendimų priėmimas, mokymasis iš istorijos, socialumas. DI elementų intelektualumas yra laikomas prioritetine tyrimų kryptimi [2]. Plėtojant šią temą teigiama, jog ilginiui turėtų atsirasti ir tapti įprastomis šios funkcijos:

- veiksmų sekos (elgsenos) atlikimas suvokus iškilusią situaciją;
- prisitaikymas prie besikeičiančio naudotojo elgesio;
- bendradarbiavimas tarp intelektualių daiktų siekiant suformuoto tikslo.

Šiomis savybėmis pasižyminčią esybę galima sutapatinti su anksčiau suformuota agento koncepcija – remdamasis įsitikinimais agentas vykdo užduotis, kad būtų pasiektas tikslas [19, 20]. Įprasta, jog ilgalaikį tikslą suformuoja kūrėjas, o įsitikinimai ir užduotys gali susiformuoti tiek veikimo metu, tiek būti numatytos iš anksto. Siūloma skirti šias sprendimų priėmimo strategijas [19, 20]:

- sprendimas konkrečiu momentu priklauso tik nuo tuo metu suvoktos situacijos;
- prieš priimant sprendimą situacija sumodeliuojama virtualiame aplinkos modelyje;
- kiekvienu priimtu sprendimu yra siekiama galutinio tikslo;
- kiekvienu priimtu sprendimu yra siekiama didžiausios naudos, net jei taip ir nukrypstama nuo galutinio tikslo.

Pirmasis būdas – paprasčiausias, nes neatsižvelgiama į aplinkos ir priimtų sprendimų istoriją. Sprendimas yra priimamas remiantis gautu aplinkos atvaizdu. Esminė problema – šis atvaizdas turi būti pakankamai teisingas ir pilnas, kad priimtas sprendimas būtų teisingas. Priešingu atveju rezultatas gali būti nepageidaujamas ar net žalingas.

Aplinkos modeliavimas teikia privalumą – kiekvienu momentu galima negauti pilnos informacijos apie aplinką, o atsiradusias spragas užpildyti iš modelio. Aplinkos modelis susideda iš dviejų komponentų – informacijos, kaip aplinkos būsenos keičiasi nepriklausomai nuo agento įtakos

ir informacijos, kaip agento įtaka pakeis aplinkos būseną. Problema yra aplinkos išreiškimas forma, kurią būtų galima naudoti sprendimų priėmimo procese. Tai galima laikyti mokymusi iš istorijos, kadangi modelyje yra informacija ir apie dabartinę būseną ir jų istoriją – kaip kito aplinkos būseną, kokie veiksmai buvo atlikti, ir koks buvo jų poveikis. Literatūroje [25] mokymasis iš istorijos yra reikalaujama intelektualumo savybė, kuri kitur nėra taip sureikšminama [24].

Siekiant galutinio tikslo esybei yra suformuojamas tikslas, kurio ji sieks savo egzistavimo metu. Šis apibūdinimas yra panašus į didžiausios naudos siekimą, tačiau naudą galima įsivaizduoti kaip trumpesnio laiko intervalo tikslą [28]. Tai aiškiau pasireiškia situacijose, kuriose iškyla poreikis bendradarbiauti. Siekiant tikslo bendradarbiauti galima atsisakyti, kadangi tai nesusiję su siekiamu tikslu. Siekiant naudos bendradarbiavimui gali būti suteiktas prioritetas, kadangi esybė, su kuria buvo bendradarbiaujama gali suteikti informacijos, ar atlikti reikiamas užduotis, taip suteikdama naudos konkrečioje situacijoje. Kitoje situacijoje nauda jau gali būti kitokios išraiškos, o tikslo bus siekiama to paties, nenukrypstant į šalutines galimybes. Kadangi nėra konkrečių apibrėžimų, šie du dalykai dažnai yra sutapatinami, todėl viskas priklauso nuo realizacijos ir panaudojimo.

Kaip ir būsto intelektualiosios aplinkos atveju, kalbant apie agentus svarbu atskirti išmanumo ir intelektualumo sąvokas. Siūloma [26] agentą vadinti išmaniuoju, kai jo sprendimų priėmimas yra pagrįstas griežta logika, o intelektualiuoju – tokį, kurio sprendimų priėmimas yra grindžiamas savybėmis, kurios yra laikomos intelektualiomis. Formaliai agentą  $A$  galima apibrėžti sekančiai:

$$A = \{X, K, Y, R, T, \Phi, \Psi\} \quad (1)$$

Čia:  $X$  – įtakojantys veiksniai;  
 $K$  – agento vidinė būsena;  
 $Y$  – sprendimas;  
 $R$  – nepriklausoma erdvės įtaka;  
 $T$  – nepriklausoma laiko įtaka.

Sprendimų priėmimas yra vykdomas atliekant transformacijas:

$$\Phi: X \times K \times R \times T \rightarrow K; \quad (2)$$

$$\Psi: K \times R \times T \rightarrow Y; \quad (3)$$

$\Phi$  transformacija keičiama agento vidinė būseną, o  $\Psi$  suformuojama sprendimo išraiška. Iš šio apibrėžimo išplaukia, jog agento sprendimų priėmimo procesas priklauso nuo to, kaip yra realizuotos minėtos transformacijos.

Vienas iš intelekto formavimo būdų – nurodyti pavyzdžius. Pavyzdžius turėtų formuoti ekspertas, žinantis situacijas ir jų reikšmes. Ekspertu gali būti dalykinės srities specialistas ar sistemos galutinis naudotojas. Šiais pavyzdžiais remdamasis agentas gali įvertinti aplinkoje susidarancias situacijas ir priimti sprendimus. Tai yra atliekama dabartinę situaciją palyginant su pavyzdinėmis situacijomis ir nusprendžiant į kurią pavyzdinę situaciją dabartinė yra panašiausia. Tarkime, jog svarbūs erdvės veiksniai yra  $n=1, 2, \dots, S$ , o juos įvertinus erdvę galima įtakoti atliekant veiksmus  $p=1, 2, \dots, N$ . Kiekvieno veiksnio suvokimo metu yra atliekama jo išmatavimas ir normalizavimas [14]. Normalizavimas yra reikalingas, kad kiekvienas veiksnys turėtų vienodą vertę, ir sprendimų priėmimas priklausytų nuo  $\Phi$  ir  $\Psi$  transformacijų, o ne nuo to, kaip buvo atliktas veiksmių matavimas. Tai atlikus gaunama situacijos  $l$ , kuriai įvykus bus atliekamas veiksmas  $p$ , veiksmių  $n$  būsenų  $\alpha_{pn}$  rinkinio  $\vec{\alpha}_p^l$  išraiška:

$$\vec{\alpha}_p^l = (\alpha_{p1}, \alpha_{p2}, \dots, \alpha_{pi}, \dots, \alpha_{pS}) \quad (4)$$

Yra teigiama [27], jog atlikus situacijos centravimą, sprendimų priėmimo procesas atliekamas teisingiau. Agento apmokymas vykdomas nurodant pavyzdžius – etaloninių situacijų  $\vec{\alpha}_p^{0l}$ , kurios turėtų iššaukti kurį nors veiksmą  $p$ , rinkinį. Remiantis pavyzdžiais suformuojama galimų veiksmių svarbos koeficientai kiekvieno veiksmo atžvilgiu:

$$\vec{K}_p = (\vec{K}_{p1}, \vec{K}_{p2}, \dots, \vec{K}_{pi}, \dots, \vec{K}_{pN}) \quad (5)$$

Remiantis šių koeficientų rinkiniu yra vertinama kiekviena susidariusi situacija, ir pagal tai priimamas sprendimas. Koeficientų apskaičiavimui yra naudojama tiesinis programavimas, sprendžiant sekantį uždavinį:

$$\Phi_p(\vec{\alpha}_p^{0k}) = \sum_{i=1}^S \alpha_{pi}^{0k} K_{pi} \rightarrow \max \quad (6)$$

Su šiais apribojimais:

$$\sum_{i=1}^N \alpha_{pi}^{0l} K_{pi} \geq \gamma \sum_{i=1}^N \alpha_{pi}^{0k} K_{pi}, \quad \forall l \quad (7)$$

$$\sum_{i=1}^N \alpha_{ri}^{0l} K_{pi} \leq \kappa \sum_{i=1}^N \alpha_{pi}^{0k} K_{pi}, \quad \forall l \quad (8)$$

$$0 \leq \vec{K}_p \leq A \quad (9)$$

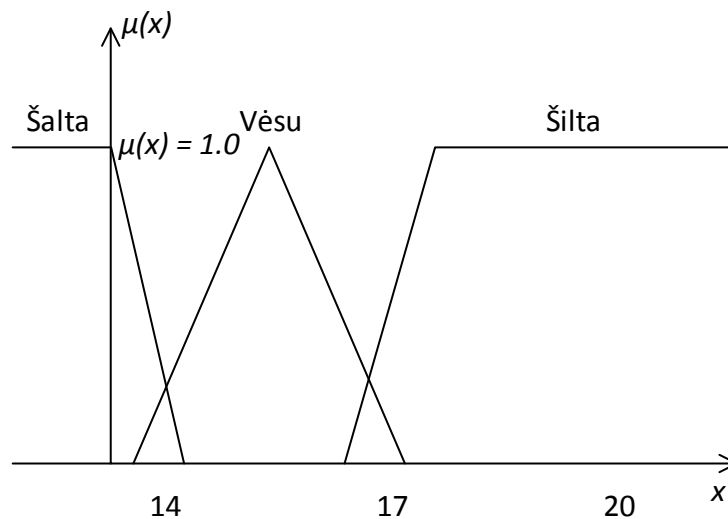
Čia:  $\vec{\alpha}_p^{0k}$  – pasirinkta „pagrindinė“ situacija veiksmo  $p$  atžvilgiu;  
 $\alpha_{pi}^{0l}$  – situacija, panaši į pagrindinę, kuriai iškilus taip pat reikėtų atlikti veiksmą  $p$ ;  
 $\alpha_{ri}^{0l}$  – situacija, nepanaši į pagrindinę;  
 $A$  – bet koks patogus skaičius;  
 $\gamma$  ir  $\kappa$  panašumo lygio įvertinimo koeficientai.

Rekomenduojama [23], kad šie koeficientai būtų iš intervalo [0 - 1], ir  $\gamma > \kappa$ . Uždavinys yra sudaromas ir išsprendžiamas kiekvienam veiksmui. Išsprendus šį uždavinį randami  $\vec{K}_p$  ir  $\Phi_{pmax}$ . Pastarasis yra naudojamas suvienodinti apskaičiuojamų situacijų įverčių svarbą sekančiame žingsnyje, kai yra nusprendžiama koks veiksmas bus atliekamas. Agento gyvavimo metu pastebėtų situacijų  $\vec{x}^0$  vertinimas atliekamas sekančiai:

$$\Phi_p(\vec{x}^0) = \sum_{i=1}^N x_i^0 K_{pi} \quad (10)$$

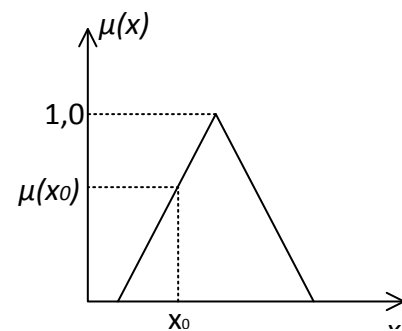
Taip realizuojama nuomonės formavimo logika. Sprendimų priėmimo logika priklauso nuo konkrečių taikymų.

Kitas galimas būdas formuoti intelektą – suformuoti miglotas taisykles, kuriomis remdamasis agentas priimtų sprendimus. Šio būdo pagrindas yra miglotoji logika. Tai logika, kurioje kiekviena reikšmė  $x$  turi tikrumo įvertį  $\mu(x) \in [0, 1]$ . Paveiksle (3 pav) pavaizduotas miglotosios logikos lingvistinio kintamojo grafinė išraiška. Šis kintamasis turi tris narius – [Šalta, Vėsu, Šilta]. Šie nariai yra išreikšti priklausomybės funkcijomis.



3 pav. Miglotosios priklausomybės funkcijos

Pirmasis intelekto formavimo žingsnis – suformuoti kintamuosius kiekvienam aktualiam aplinkos veiksniai: veiksnys yra identifikuojamas ir padalijamas į pakankamo detalumo narius, pagal kuriuos suformuojamas lingvistinis kintamasis. Remiantis sudarytomis priklausomybės funkcijomis kiekvienas sensoriaus įvertis  $x_i$  bus paverčiamas į miglotąją formą (verbalizavimas)  $(x_i, \mu(x_i))$  (4 pav). Tokiu pat principu aprašomi ir rezultato lingvistiniai kintamieji,



4 pav. Verbalizavimas

pavyzdžiui, kas gali būti atlikta. Literatūroje [33] žinių bazei suformuoti siūloma naudoti šiuos šaltinius: dalykinės srities eksperto žinios, rėmimasis sudarytu modeliu bei saviorganizacija. Teigiama, jog populiariausias ir plačiausiai naudojamas yra pirmasis. Jis yra paprasčiausias – galima tiesiog parinkti reikšmes ir jas tobulinti, kol bus pasiekti tenkinantys rezultatai. Antrasis žingsnis – miglotųjų taisyklių sudarymas. Literatūroje [26, 33] paminėta dvi tokių taisyklių išraiškos:

$$IF x \text{ is } A \text{ AND } y \text{ is } B \text{ THEN } z \text{ is } C \quad (11)$$

$$IF x \text{ is } A \text{ AND } y \text{ is } B \text{ THEN } z = F(x, y) \quad (12)$$

Pirmoji taikoma Mamdani miglotuosiuose modeliuose, o antroji – Takagi-Sugeno miglotuosiuose modeliuose. Taisyklės yra sudarytos iš dviejų dalių – sąlygos ir rezultato. Pirmuoju atveju rezultatas yra miglotasis įvertis, antruoju – funkcija. Taisyklių sudarymas miglotąja forma leidžia jų sudaryti daug mažiau, nei reikėtų tą patį dalyką atliekant tradicine forma, bei, galima atsižvelgti į veiksmus kurie įvyksta retai, ar iš viso neįvyksta ir juos sugrupuoti, taip dar labiau sumažinant taisyklių kiekį. Rezultato transformaciją iš miglotosios logikos į tradicinę (nuspręstis)

galima atlikti įvairiais metodais, pavyzdžiui apskaičiuojant gravitacijos centrą (CoG) Mamdani atveju. Takagi-Sugeno atveju rezultatas yra tradicinės logikos funkcija, todėl nuspręstis yra nereikalinga [34].

Trečiasis būdas – gali būti formuojamas tikslas, kurio siekdamas agentas atliks veiksmus. Agentą, turintį šio tipo intelektą galima priskirti prie agentų, siekiančio tikslo, kategorijos. Tendencija yra formuojama naudojant stochastinės aproksimacijos pagrindus. Realizacija pradedama identifikuojant veiksmus, kuriuos gali atlikti agentas, bei jų reikšmę siekiant tikslo. Vieni jų gali trukdyti, kiti – padėti. Tai atlikus yra suformuojami elgseną apibūdinančios lygtys  $Q(K, X)$ . Jų vidurkiai yra maksimizuojami arba minimizuojami, priklausomai nuo įtakos tikslui. Tarkime, kad agentas veikia pagal nežinomą tikslo funkciją  $V(K, X)$ . Norint pakeisti šią funkciją elgsenų išraiška yra siūloma panaudoti multiplikatyvią arba adityvią funkciją  $\aleph\{*\}$ . Rezultate, tikslo funkcija gali turėti sekančią išraišką:

$$\aleph\{\mathcal{M}\{Q(K, X)\}\} \rightarrow \min \quad (13)$$

Svarbu pabrėžti, jog šiuo atveju apmokymas yra vykdomas sistemos veikimo metu – įverčiai yra nuolat atnaujinami. Tai gali turėti žalingų pasekmių eksploatacijos metu. Kad to išvengti yra siūloma sukurti virtualų aplinkos modelį, kuriame sistema galėtų apsimokyti, ir pasiekus tenkinantį veikimą nebeleisti keisti vidinės būsenos. Tuomet ji tampa statine ir nekinta veikimo metu, tačiau ir toliau sėkmingai naudojama priimant sprendimus.

Ketvirtasis literatūroje išskirtas būdas [28] – siekti didžiausios naudos. Nauda yra įvertis kaip labai agentui patinka situacija. Jai įvertinti yra naudojama funkcija  $u$ , kuri bet kokiam situacijų aibės nariui  $\alpha \in A$  geba priskirti skaitinį įvertį iš realiųjų skaičių aibės  $\aleph$ :

$$u: A \rightarrow \aleph \quad (14)$$

Kadangi erdvė yra nedeterministinė, situacijų gali būti be galo daug. Praktikoje yra suformuojamos situacijų kategorijos, kurioms pagal išskirtus požymius priskiriama situacija  $\alpha$ . Dažniausiai naudos funkcijų nėra galima numatyti iš anksto, nes aplinka keičiasi ne vien dėl agento veiksmų bet ir nepriklausomai nuo jo veiksmų, todėl yra taikomi tikimybiniai metodai, ar vykdomas apmokymas iš situacijų istorijos [28]. Iškilus situacijai  $s$  yra žinoma, jog atlikus veiksmą  $p$  bus iššaukta situacija  $\alpha'$  su tam tikra tikimybe  $T(\alpha, p, \alpha')$ . Tuomet naudą  $E$ , kurią agentas gaus atlikęs veiksmą  $p$  galima apskaičiuoti naudojant formulę:

$$E[u, p, \alpha] = \sum_{\alpha' \in A} T(\alpha, p, \alpha')u(\alpha') \quad (15)$$

Iš to seka jog bet kuriuo momentu agentas žino, jog būdamas tam tikroje būsenoje  $\alpha$  jis turėtų atlikti veiksmą  $p$ , nes iššaukta situacija  $\alpha'$  suteiks daugiausia naudos. Tuomet uždavinys tampa surasti tokią veiksmų seką  $\pi' \in \pi$ , kad būtų pasiekta didžiausia nauda:

$$\pi'(\alpha) = \arg \max_{p \in P} E[u, p, \alpha] \quad (16)$$

Tam, kad naudos būtų siekiama greičiau, tolimesnius atlygius už situacijas  $r(\alpha)$  galima mažinti įvedus koeficientą  $\gamma$ , tuomet agento nauda  $u$  yra apskaičiuojama naudojant šią formulę:

$$u(\alpha) = r(\alpha) + \gamma \max_{p \in P} \sum_{\alpha' \in A} T(\alpha, p, \alpha') u(\alpha') \quad (17)$$

Kadangi gaunama nauda iš anksto nėra žinoma, yra siūloma parinkti bet kokius naudos funkcijos  $u(\alpha)$  įverčius ir juos pastoviai atnaujinti. Teigiama, jog įverčiai sąlyginai greitai konverguoja į tikruosius [28]:

$$u^{t+1}(s) \leftarrow r(s) + \gamma \max_{p \in P} \sum_{\alpha' \in A} T(\alpha, p, \alpha') u^t(\alpha') \quad (18)$$

Yra ir daugiau intelekto formavimo būdų, kurie bus neapžvelgti, pavyzdžiui, neuroniniai tinklai. Šio darbo kontekste buvo aktualūs šie, kadangi laikomasi nusistatymo, jog intelektualiai esybė turėtų pasižymėti savybėmis, kurios žmoguje yra laikomos intelektualiomis – mokymasis iš pavyzdžių, vadovavimasis miglotomis taisyklėmis, tikslo ar naudos siekimas.

Literatūroje rasta, jog yra atliekami panašūs elementai. Šaltinyje [19] yra siūlomas metodas kaip sumažinti energijos sąnaudas būste iki tokio lygio, kad būste išgaunamas energijos kiekis būtų pakankamas jį naudojančioms įrenginiams. Darbe yra orientuojamasi į įrenginius, kurie nenaudojami persijungia į „laukimo“ režimą ir vis vien tęsia energijos naudojimą. Energijos sąnaudų mažinimo sistema yra siūloma diegti ne į pačius įrenginius, bet į maitinimo lizdus, prie kurių šie yra prijungti. Taip yra išsprendžiamas įrenginių suderinamumo su projektuojama sistema problema – valdomi ne patys įrenginiai, o maitinimo lizdai, prie kurių jie yra prijungti. Valdymas yra atliekamas remiantis šiomis taisyklėmis:

- įtaisai gali būti pažymėtas kaip valdomas ar nevaldomas;
- valdomi įtaisai gali turėti hierarchinę struktūrą – išjungus hierarchiškai aukščiau esantį elementą visi žemiau esantys elementai taip pat bus išjungti;
- atsižvelgiama į naudotojo elgesį.



Darbe yra pastebėta, jog didžiausia problema įrenginio grąžinimas į „laukimo“ režimą. Tai gali būti pavadama atlikti sistemos naudotojui, tačiau sprendimas yra nepatogus. Pastebėtas ir patogus būdas – tai atlikti automatiškai, remiantis naudotojo elgesio nuspėjimu naudojant atitinkamus jutiklius. Pabrėžiama, jog šiuo atveju reikalinga būdas kaip išvengti klaidų dėl klaidingo spėjimo. Nepaisant nepatogių sprendimų naudotojo atžvilgiu, sistemos prototipo kūrėjams pavyko sumažinti energijos sąnaudas iki 10%.

Hui Hong [20] sprendime yra siekiama sumažinti intelektualiojo būsto energijos sąnaudas užtikrinant naudotojų komfortą. Sistema yra sudaryta iš jutiklių, valdiklio ir vykdyklių. Valdiklyje yra iš anksto suformuotos taisyklės, kuriomis remiantis vykdomas valdymas. Jutikliai yra periodiškai apklausiami ir pagal surinktus duomenis įvykdomi atitinkami veiksmai. Pavyzdžiui, jei dūmų jutiklis yra aktyvuotas, atidaromi langai ir įjungžiama ventiliacija, kad pašalinti dūmus. Tokiai sistemai realizuoti reikalinga žinoti visus galimus įvykius, taikomoje aplinkoje ir jiems sudaryti taisykles. Šiuo atveju galbūt ir išvengiama poreikis diegti apsaugos priemones nuo blogų sprendimų, nes turint pilną „gerų“ taisyklių aibę atliekami veiksmai taip pat bus „geri“, tačiau bendruoju atveju, būsto aplinka nėra deterministinė, todėl į daugiau veiksmų atsižvelgiančią sistemą (pavyzdyje buvo temperatūra, dūmai ir lietus) realizuoti gali būti sunku arba neįmanoma. Šio tipo sistema yra vadinama tradicine [21], kadangi yra remiamasi tik jutiklių parodymais.

Saugumo užtikrinimas gali būti realizuojamas naudojant įterptines sistemas [22]. Įvertinus taikymų kainą ir atnešamą naudą autorių yra nuspręsta naudoti hibridinį techninį – programinį sprendimą. Kuriant saugumo sistemą autoriai siūlo taikyti krioklio stiliaus metodologiją. Yra siūloma naudoti sąlyginai mažai skaičiavimų reikalaujančius kriptografinius algoritmus kartu su TPM. Komunikacija turėtų būti vykdoma saugiais standartizuotais protokolais, o konkretūs taikymai realizuojami saugioje operacinėje sistemoje.

A. Kamilaris ir A. Pitsillides [23] yra siūloma naudoti egzistuojančius metodus. Įrenginiams siūloma bendrauti HTTP protokolu, savo paslaugas teigti REST WS priemonėmis, komunikacijai naudoti XML ir JSON. Sistema agreguoja įvairius duomenis, naudotojui jie yra pateikiami lentelių ir grafikų forma. Remiantis tuo naudotojas sudaro taisykles, pagal kurias atliekamas įrenginių valdymas. Be to, siūloma daiktus „dalintis“ socialiniuose tinkluose. Galima išvelgti, jog šiuo atveju ir daiktai ir juos valdanti sistema turi turėti pakankamus resursus, kad galėtų funkcionuoti – neatsižvelgiama į realizavimo kainą. Konfigūraciją turi atlikti naudotojas, numatydamas kada ir kaip sistema turėtų elgtis. Sprendime yra visiškai neatsižvelgiama į saugumo aspektus. Apibendrinat, tam, kad realizuoti saugą tokioje sistemoje turi būti užtikrinti skaičiavimo resursai, spartūs ryšio kanalai dideliems duomenų srautams perduoti, pakankami energijos ištekliai, kad įrenginiai galėtų funkcionuoti. Be

abejo, atsižvelgus į tai, kokios problemos ištinka dabartinius internetines programų sistemas, tokiai sistemai reikalinga ir pastovi naudotojų bei administratorių priežiūra, atnaujinimai ir t.t.

#### **1.4. Teorinio problemos nagrinėjimo išvados**

Atlikus literatūros analizę galima teigti, jog būsto aplinka yra viena iš svarbių DI taikymo sričių. Pirmoji DI taikymo stadija yra informacijos rinkimas, agregavimas ir pateikimas sistemos naudotojams. Pastebėta kryptingas taikymų judėjimas link automatizacijos – kai surinkta informacija yra naudojama atlikti numatytoms funkcijoms. Kadangi tiriamoji fizinė aplinka yra nedeterministinė, numatyti galimas situacijas gali būti pernelyg sunku. Intelektualiųjų metodų pranašumas yra tai, jog jiems nėra būtinas pilnas situacijų aprašas. Funkcionavimą galima realizuoti darant prielaidas apie aplinką, remiantis turima informacija ar spėjimais. Keliamas autonomiškumo reikalavimas – realizavus mokymąsi galima pasiekti, jog sistema prisitaikytų prie aplinkos pokyčių ir atsirandančių naujų reikalavimų. To pasekmė – nereikėtų žmogiškojo įsikišimo, sistema galėtų veikti ekonomiškai. Dėl intelektualumo koncepcijos populiarėjimo IT, bei daiktų interneto plėtros, ir galimos naudos yra tikslinga plėtoti šią tyrimų kryptį.

Literatūroje pabrėžiama, jog šio tipo sistemoje sauga yra ypatingai svarbu. Siekiama, kad naudotojai ne tik būtų apsaugoti nuo grėsmių, bet ir įgautų pasitikėjimą sistemomis nuo pat jų įdiegimo momento. Skiriami keturi sistemos aktoriai – žmogus, intelektualusis objektas, technologinė ekosistema ir procesas. Galima teigti, jog ryšiuose tarp žmogaus, technologinės ekosistemos ir intelektualaus objekto yra saugos atžvilgiu yra nuveikta daug, esami metodai yra pritaikyti naujoms struktūros, bei sukurti nauji (pav. 1). Ryšiuose tarp proceso ir kitų aktorių saugumo atžvilgiu yra atlikta per mažai tyrimų. Viena iš esminių priežasčių – neaiškus proceso apibrėžimas. Nuspręsta, jog sritis, kuri yra nepakankamai ištyrinėta, tačiau svarbi ateities IT plėtojime, yra racionalus tiriamosios srities pasirinkimas. Remiantis šiais pastebėjimais buvo suformuotas darbo tikslas:

#### **sudaryti ir iširti būsto intelektualiosios aplinkos taikomojo lygmens saugumo intelektikos sistemos modelį.**

Numatyti šie uždaviniai:

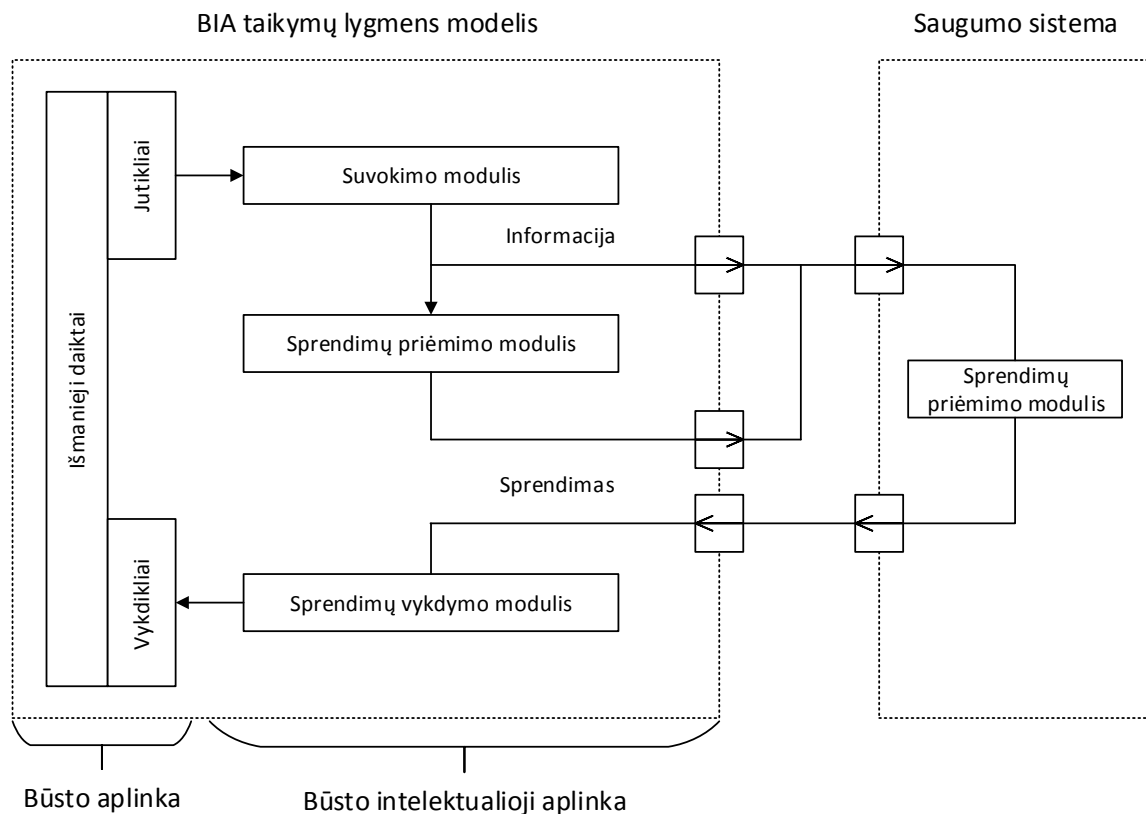
- atlikti literatūros analizę tiriamosios srities klausimais;
- sukurti BIA modelį;
- sudaryti BIA modelio aplinkos situacijų aibės intelektualiuosius apdorojimo metodus;
- sudaryti BIA saugumo sistemos funkcinę organizaciją;
- atlikti kompiuterinio BIA modelio eksperimentą ir rezultatų analizę.

## 2. SISTEMOS PROJEKTAVIMAS

Šiame skyriuje aprašytas darbo projektavimo etapas. Projektavimas susideda iš dviejų dalių:

- būsto intelektualiosios aplinkos taikymų lygmens sistemos modelio projektavimo;
- būsto intelektualiosios aplinkos taikymų lygmens saugos sistemos projektavimo.

Pirmosios dalies paskirtis yra modeliuoti būsto intelektualiojoje aplinkoje vykstančius procesus, taip sukuriant virtualią platformą, kurioje galima kurti ir išbandyti antrosios dalies veikimą.



5 pav. Būsto intelektualiosios sistemos modelio principinė schema

### 2.1. Virtualaus BIA modelio projektavimas

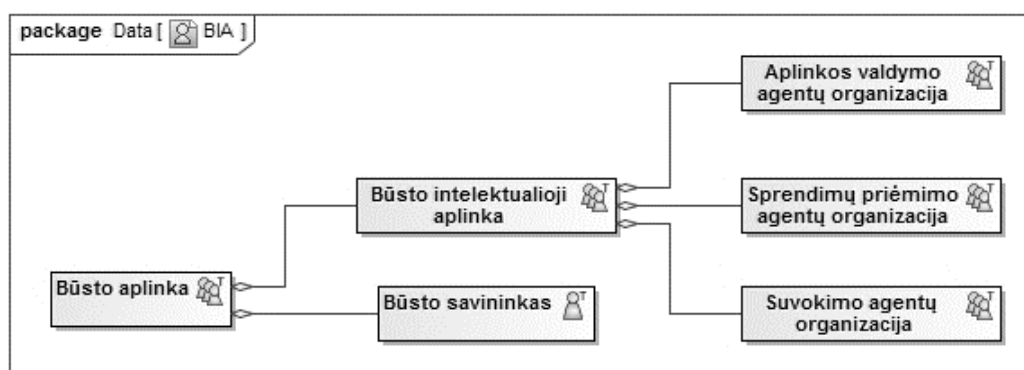
Projektuojant būsto intelektualiąją aplinką buvo naudojamas smulkinamasis projektavimas. Būsto intelektualiojoje aplinkoje skiriami trys intelektualiujų esybių (agentų) lygmenys:

- suvokimo lygmuo – surenka rodmenis iš išmaniųjų prietaisų jutiklių ir transformuoja juos į kitoms esybėms suprantamą informaciją;
- sprendimų priėmimo lygmuo – esybės remiasi suvokimo lygmenyje suformuota informacija, kad priimtų sprendimus;

- valdymo lygmuo – esybės priėmę sprendimą paveikia būsto aplinką išmaniųjų prietaisų vykdikliais.

Atsižvelgiant į šiuos lygmenis buvo sudarytas hierarchinis BIA modelis, atskleidžiantis kuriamos organizacijos struktūrą (6 pav). Sistemos modelis sudarytas iš dviejų dalių – BIA taikymų lygmens modelio ir saugumo sistemos. BIA taikymų lygmens modely sudaro būsto aplinka ir būsto intelektualioji aplinka. Būsto aplinka yra fizinis būstas, kuriame yra išmanieji daiktai. Šie daiktai turi jutiklius, kuriais gali surinkti informaciją apie susidariusią situaciją bei vykdiklius, kuriais gali ją įtakoti. Būsto intelektualioji aplinka yra virtuali aplinka, kuri komunikuoja su išmaniaisiais daiktais. Išmanieji daiktai vienu metu egzistuoja ir virtualioje aplinkoje ir fizinėje. Per suteikiamas sąsajas iš jutiklių yra gaunami duomenys, kuriuos interpretuoja suvokimo modulis. Šis modulis suformuoja sistemai suprantamą ir aktualią informaciją, kuri yra perduodama į sprendimų priėmimo modulį. Neesant saugumo sistemai priimtas sprendimas būtų perduodamas sprendimų vykdymo moduliui ir per sąsajas su išmaniaisiais daiktais būtų įvykdytas jų vykdiklių (jei reikia). Esant saugumo sistemai grandis tarp sprendimų priėmimo modulio bei sprendimų vykdymo modulio yra nutraukiama, ir toje vietoje įterpiama sąsaja su saugumo sistema. BIA taikymų lygmens saugumo sistemoje yra tik sprendimų priėmimo modulis. Informacija yra gaunama iš BIA, sprendimai įvykdomi taip pat per ją. Saugumo sistema naudoja ir fizinės ir virtualios būsto aplinkos informaciją.

Būsto aplinka – aukščiausiam hierarchiniame lygmenyje esanti struktūra, apimanti būsto intelektualiąją aplinką ir būsto savininką (6 pav). Būsto intelektualiojoje aplinkoje išskiriami trijuose lygmenyse esantys agentai – aplinkos valdymo, skirti valdyti išmaniuosius daiktus; sprendimų priėmimo, bei suvokimo – iš išmaniųjų daiktų jutiklių gauna duomenis ir transformuoja į suprantamą informaciją.



6 pav. Hierarchinis BIA modelis

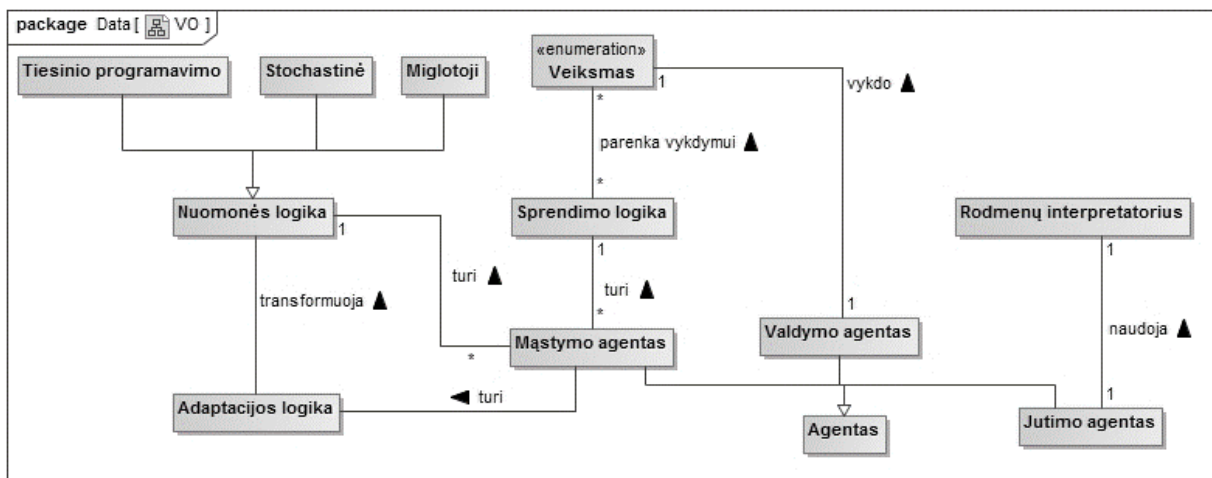
Sudarius hierarchinį būsto intelektualiosios aplinkos modelį buvo suformuotas supaprastintas koncepcinis veiklos objektų – agentų modelis (7 pav). Išskirtos trys agentų grupės: mąstymo, valdymo ir jutimo. Mąstymo agentas sudarytas iš trijų logikos rinkinių – nuomonės formavimo, adaptacijos ir

sprendimų priėmimo. Nuomonės formavime yra naudojama miglotoji logika. Nuomonė yra formuojama remiantis jutimo agento surinktais duomenis iš išmaniųjų prietaisų, bei transformuoja juos į sistemai suvokiamą informaciją. Transformacijai atlikti yra naudojamas rodmenų interpretatorius.

Remiantis suformuota nuomone sprendimo logika parenka reikalingus veiksmus, kad įgyvendinti numatytą tikslą. Veiksmus vykdo valdymo agentas. Adaptacijos logika yra skirta prisitaikyti prie pokyčių aplinkoje, naujų poreikių atsiradimo. Sprendimų priėmimo ir adaptacijos logika yra sudaromos agento kūrimo metu ir jo gyvavimo metu nekinta. Nuomonės formavimo logika yra kintantis komponentas, kurio transformaciją atlieka nuomonės logika. Adaptacijos logika nuomonės logikos pakeitimus atlieka remdamasi:

- gautomis žiniomis;
- sistemos esybių atliekamais veiksmais;
- naudotojo atliekamais veiksmais;

Šį procesą dar galima vadinti agento apsimokymo procesu – agentas autonomiškai keičia sukūrimo metu įdiegtas veikimo taisykles ir prisitaiko prie realios aplinkos, kad patenkintų naudotojo poreikius.

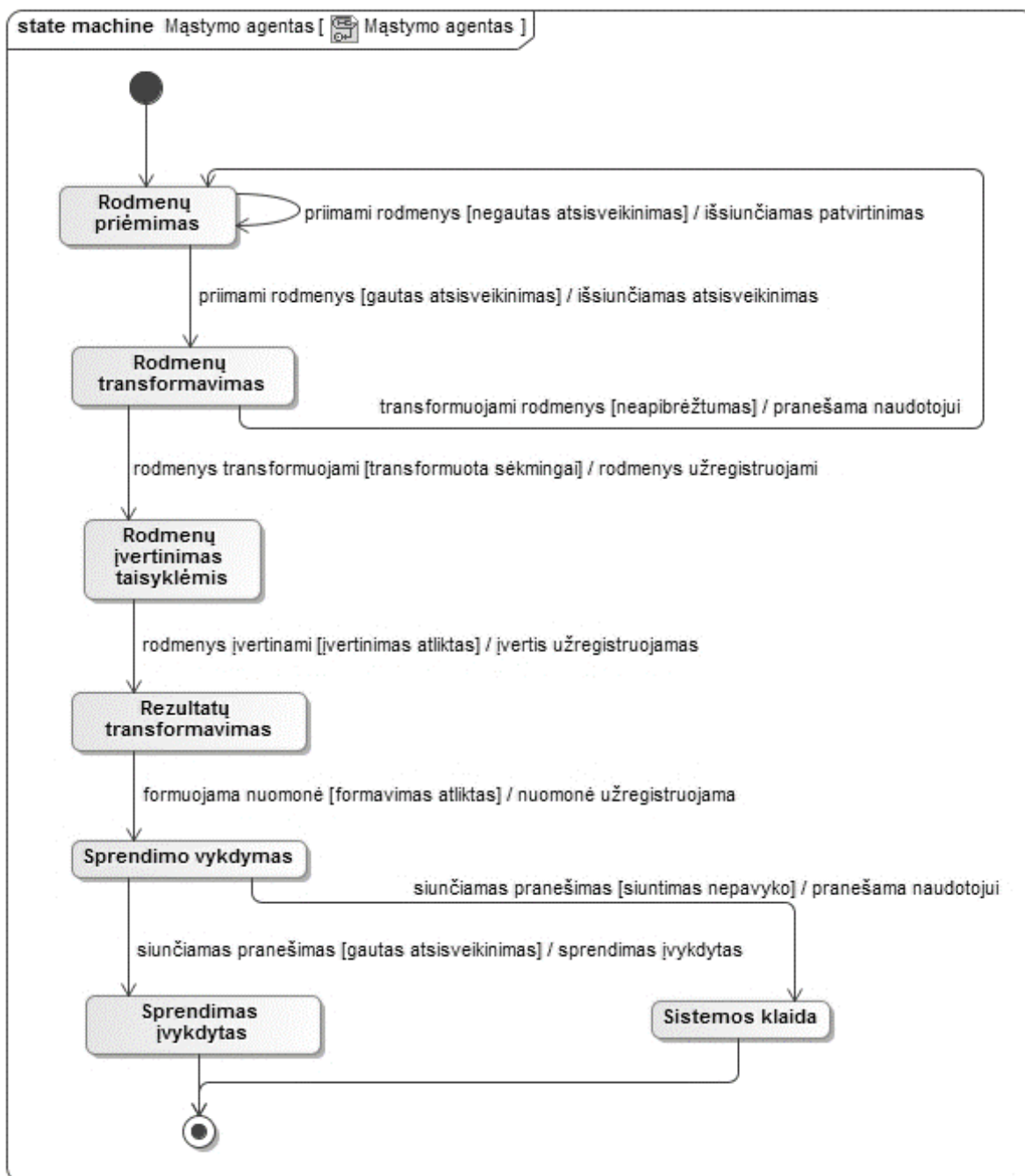


7 pav. Konceptinis veiklos objektų modelis

Buvo sudarytas apibendrintas mąstymo agento būsenų modelis (8 pav). Šiame modelyje buvo išskirtos šios būsenos:

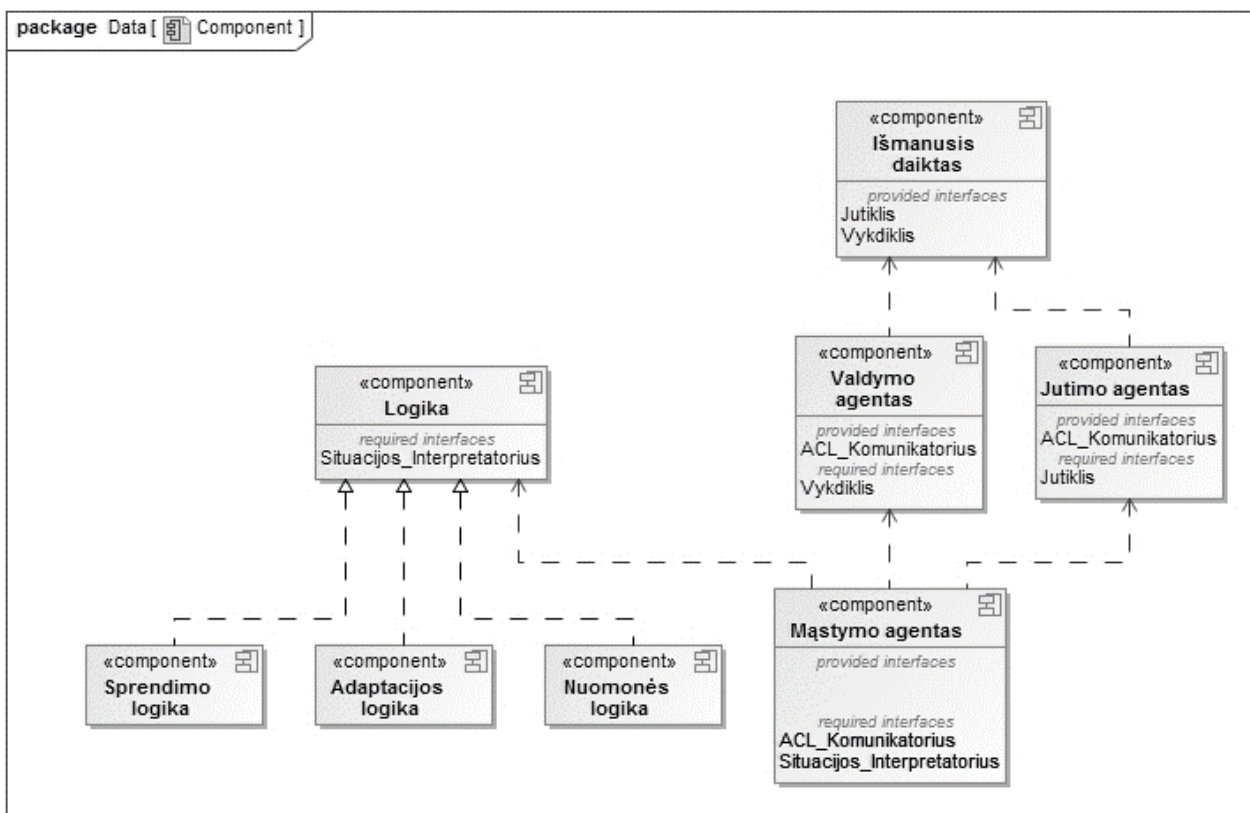
- 1) Rodmenų priėmimas – pokalbio su jutimo agento metu gaunamos aplinkos būseną apibūdinančios žinios.
- 2) Rodmenų transformavimas – pokalbio metu gautos žinios yra transformuojamos į BIA agentų logikai suprantamas žinias – miglotąją formą.

- 3) Rodmenų įvertinimas taisyklėmis – žinios yra įvertinamos remiantis agento nuomonės formavimo taisyklėmis.
- 4) Rezultatų transformavimas – įvertinimo rezultatai yra transformuojami į veiksmą ar jų seką, atliekamas sprendimo priėmimas.
- 5) Sprendimo vykdymas – sprendimą vykdo valdymo agentas, tačiau mąstymo agentas yra atsakingas nurodymo perdavimą, o tai taip pat yra sprendimo vykdymo dalis.
- 6) Sprendimas įvykdytas / sistemos klaida – sprendimas yra įvykdomas, arba jo vykdymas nepavyksta. Rezultatas „aptariamasis“ tarp mąstymo agento ir vykdymo agento.



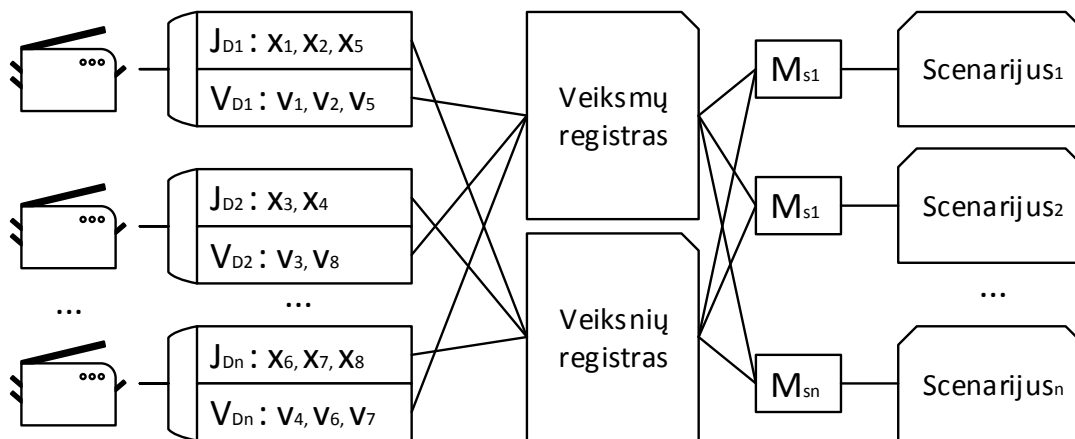
8 pav. Mąstymo agento būsenų modelis

Būsto intelektualioji aplinka naudotojui teikia paslaugas  $P = \{p_1, p_2, \dots, p_n\}$ . Šios paslaugos yra sudarytos iš veiksmų sekos  $V = \{v_1, v_2, \dots, v_m\}$ . Konkrečia eilės tvarka išdėstyti veiksmai sudaro scenarijus  $S = \{s_1, s_2, \dots, s_n\}$ . Tam, kad būtų suteikta paslauga mąstymo agentas įgyvendina scenarijų, sudarytą iš veiksmų sekos. Paslauga yra realizuojamos per būste esančius išmaniuosius daiktus  $D = \{d_1, d_2, \dots, d_m\}$ . Kiekvienas daiktas gali atlikti kelis veiksmus, šie veiksmai gali kartotis. Kylanti problema – daiktai gali būti naudojami ne tolygiai, laukiama kol vienas galės atlikti veiksmą, kai jį gali atlikti kitas. Tai stabdytų visos sistemos veikimą. Dėl šios priežasties nuspręsta su kiekvienu daiktu susieti du agentus – valdymo ir jutimo (9 pav). Išmanusis daiktas teikia jutiklio ir vykdiklio sąsajas. Jutimo agentas naudoja jutiklio sąsają, valdymo – vykdiklio. Visi agentai bendrauja FIPA organizacijos apibrėžta ACL kalba. Tam yra naudojama ACL\_komunikatoriaus sąsaja. Mąstymo agentas naudoja sprendimo, adaptacijos ir nuomonės logikas.



9 pav. BIA komponentų modelis

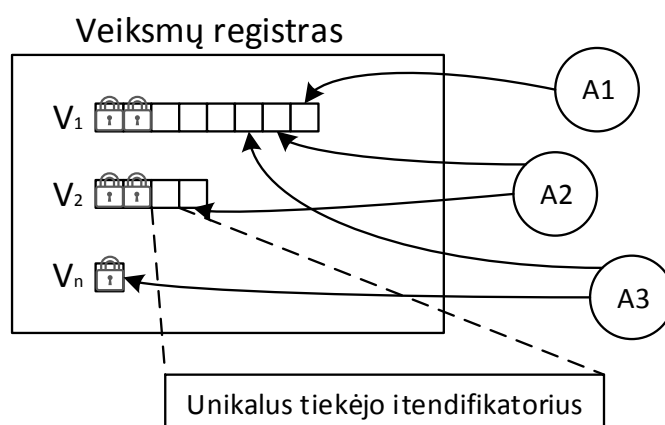
Įtakojantys veiksniai yra gaunami komunikavimo su jutimo agentu metu, sprendimai yra vykdomi perduodant nurodymus vykdymo agentui. Visi veiksmai, kuriuos gali atlikti valdymo agentai ( $V_{Dn}$ ) yra saugoma veiksmų registre, o aplinkos veiksniai, kuriuos gali išgauti jutimo agentai ( $J_{Dn}$ ) – veiksmų registre (10 pav). Mąstymo agentai yra priskiriami sistemoje sukurtiems scenarijams įgyvendinti. Kiekvienas scenarijus turi su juo susietą mąstymo agentą, kuris remdamasis veiksniais nusprendžia kada reikia suteikti paslaugą.



**10 pav. BIA esybių principinė schema**

Scenarijai yra nekintantys sistemos komponentai, apibrėžiantys reikiamus veiksmus, kurie turi būti atlikti, kad būtų suteikta paslauga. Norint, kad sistema suteiktų naują paslaugą yra pridamas scenarijus, su kuriuo susiejamas mąstymo agentas. Kada scenarijus bus įgyvendintas yra nustatoma keičiant agento logiką.

Agentas, susietas su daiktu užregistruoja veiksmų ir veiksnių registre, pateikdamas savo unikalų identifikatorių. Veiksmų registras yra skirtas kaupti informacijai, kokie veiksmai gali būti atlikti ir kas juos gali atlikti. Suformavęs nuomonę mąstymo agentas kreipiasi į veiksmų registrą norėdamas sužinoti kas gali atlikti reikiamą scenarijaus veiksmą. Šiame registre veiksmų tiekėjai yra sugrupuojami į eiles pagal atliekamus veiksmus. Kiekvienas tiekėjas gali būti prieinamoje arba užrakintoje būsenoje. Užrakinta būsena reiškia jog šiuo metu tiekėjas veiksmo atlikti negali.

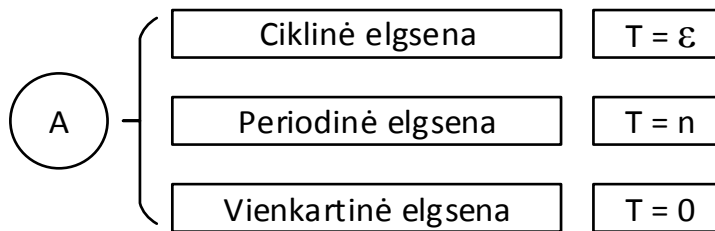


**11 pav. Veiksmų registro sandara**

Klientas, pasiimdamas tiekėjo adresą nustato užrakinimo požymį. Tiekėjas, atlikęs veiksmą šį požymį pašalina, kad jo teikiamas veiksmas vėl galėtų būti atliktas. Veiksnių registro veikimas yra analogiškas veiksmų registro veikimui.



Esybių veikla neapsiriboja vien scenarijų įgyvendinimu – yra vykdomi pokalbiai su kitais agentais, tikrinama registrų būsenos, stebima situacija aplinkoje. Šios visos elgsenos turi vykti nepriklausomai viena nuo kitos. Literatūroje [38] siūloma išskirti tris elgsenų tipus, priklausomai nuo jų vykdymo periodo  $T$  (12 pav).

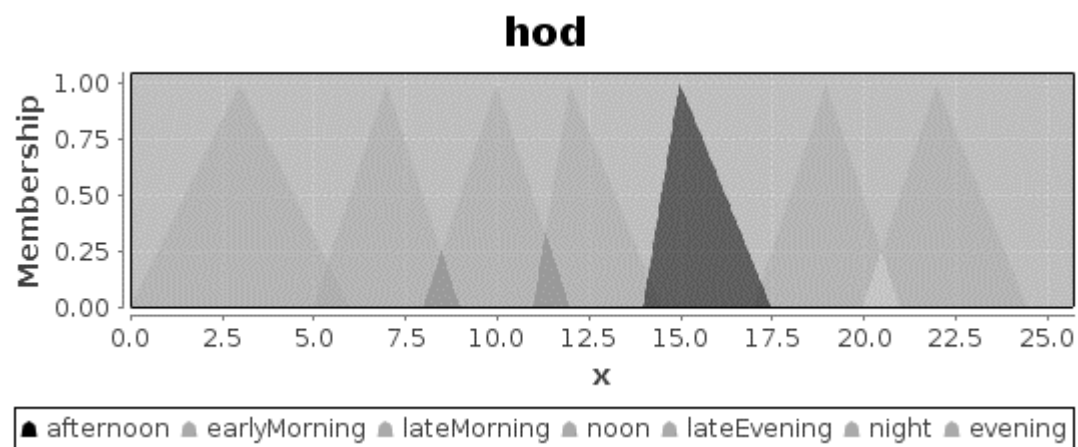


12 pav. Elgsenų tipai

Čia:  $\varepsilon$  – nereikšmingai mažas dydis, priklausantis nuo realizacijos,

$n$  – elgsenos vykdymo periodas,  $n > \varepsilon$ .

Agento vidinė būsena apibrėžiama miglotųjų aibių ir taisyklių rinkiniu. Veiksniai  $X$  yra transformuojami (verbalizuojami) į miglotąją formą suteikiant jiems žodinius įverčius ir apibrėžiant jų tikrumą tam tikrame kategoriškame įvertyje (13 pav). Pavyzdyje pateiktas miglotasis kintamasis *hod*, vaizduojantis paros valandų atvaizdavimą paros metu. Termų persidengimo vietose kintamasis vienu metu patenka į kelis terminus, tačiau yra priskiriamas tam, kuriame jo tikrumas (angl. *Membership*) yra didžiausias. Pavyzdžiui 20,1 bus priskirtas vakarui, kadangi jo tikrumas yra didesnis šiame terme, lyginant su vėlyvu vakaru.



13 pav. Kintamojo hod miglotoji išraiška

Sprendimų priėmimas vykdomas atliekant transformacijas:

$$\Phi: X \times K \times R \times T \rightarrow K; \quad (19)$$

$$\Psi: K \times R \times T \rightarrow Y; \quad (20)$$

Naudojant miglotąją logiką  $\Psi$  yra išreiškiama taisyklių rinkiniu:

$$IF\ x\ is\ A\ AND\ y\ is\ B\ THEN\ z\ is\ C \quad (21)$$

Čia:  $x, y, z$  yra žodinės miglotojo kintamojo išraiškos.

$\Phi$  yra apibrėžiama agento vidinės būsenos kitimas. Vidinė būsena turi kisti priklausomai nuo aplinkos ir naudotojo norų, kitaip taisyklės bus statinės, o agentas nepasižymės anksčiau apibrėžtais intelektualumo požymiais.

## 2.2. BIA taikymų lygmens saugumo sistemos projektavimas

Kadangi aplinka yra nuolat stebima jutimo agentų, kiekvienu metu yra žinoma jos būsena. Kaip jau minėta, su kiekvienu scenarijumi yra susieti jam aktualūs veiksniai. Šie veiksniai yra žinomi iš anksto ir sistemos eksploatacijos metu nekinta. Sistemai taip pat yra prieinama informacija apie scenarijaus rezultata, todėl galima nustatyti, kai naudotojas įgyvendina scenarijų, ir sužinoti tuo metu buvusių aktualių veiksmių būsenas. Turint šią informaciją galima taikyti Wang ir Mendel [35, 36, 39] miglotųjų taisyklių generavimo metodą. Šis metodas remiasi išankstinių aktualių veiksmių ir rezultatų žinojimu. Tarkime jog tam, kad būtų įgyvendintas scenarijus reikia, jog aktualūs veiksniai  $\{x_1, \dots, x_k\}$  įgautų tam tikras reikšmes. Kaip bus vykdomi jį sudarantys veiksmai priklauso nuo šių veiksmių. Kadangi sistemos kūrimo metu agentus sudarančios logikos jau buvo realizuotos, dalis taisyklių  $F(x_1, \dots, x_k) \rightarrow y$  yra žinoma iš anksto. Iš anksto yra žinomos ir lingvistinių kintamųjų išraiškos. Jeigu aplinkoje įgyvendinamas scenarijus  $s = \{v_1, v_2, \dots, v_n\}$  sistema padaro aplinkos situacijos atvaizdą. Šį atvaizdą sudaro aibė scenarijui aktualių veiksmių  $\{x_1, \dots, x_m\}$ . Šie veiksniai ir scenarijų sudarantys veiksniai yra įvertinami naudojant nuomonės logikoje apibrėžtus miglotuosius aprašus ir gaunami jų miglotieji įverčiai:

$$(x_1, \mu(x_1)) \in Lingvar_1, \dots, (x_k, \mu(x_k)) \in Lingvar_k, (y, \mu(y)) \in Lingvar_y \quad (22)$$

Pagal šiuos įverčius yra generuojamos taisyklės:

$$Jei\ x_1 = Lingvar_1\ ir\ \dots\ ir\ x_k = Lingvar_k\ tai\ y = Lingvar_y \quad (23)$$

Saugumo sistema yra skirta apsaugoti naudotoją nuo neteisingo paslaugų teikimo. Tai gali įvykti dėl kelių priežasčių – neteisingai priimtų sprendimų arba dėl neteisingų aktualių aplinkos veiksmių. Kadangi sistemoje yra įdiegtas mokymasis, neteisingi veiksniai įtakoja agento vidinę būseną.

Ši įtaka ilginiui (priklausomai nuo realizacijos) priveda prie netinkamos konfigūracijos. Atstačius teisingą veiksmų tiekimą sistema veiktų neteisingai dėl blogos konfigūracijos. Dėl šių priežasčių nagrinėjamo saugos aspektu tapo konfigūracija. Konfigūraciją sistemoje atitinka sprendimo ir nuomonės logika. Adaptacijos logika yra statinė, įdiegiama sistemos kūrimo metu ir eksploatacijos laikotarpyje nekinta.

Nuomonės logika yra sudaryta iš scenarijui aktualių veiksmų aprašų miglotąja forma. Jie gali būti iškreipti dėl neteisingų veiksmų. Padidinti tikimybę, jog veiksniai yra teisingi galima keliais būdais:

**lentelė 1. Veiksmų teisingumo nustatymo būdai**

<b>Būdas</b>	<b>Privalumai</b>	<b>Trūkumai</b>
naudoti kelių daiktų parodymus	šaltinis sistemos viduje – neįnešamos papildomos grėsmės; patogu identifikuoti neteisingus duomenis – didžiausio nuokrypio parodymai eliminuojami;	didesnė sistemos apkrova – daugiau matavimų, pokalbių; sprendimai priimami lėčiau dėl jutimo agentų užimtumo; veiksnio rodmenis gali teikti tik vienas agentas – sprendimas neįgyvendinamas;
atsižvelgti į istoriją	šaltinis sistemos viduje; sistema papildomai neapkraunama; visada egzistuojantys duomenys, nepriklausomai nuo parodymus teikiančių agentų kiekio;	duomenys istoriniai – situacija galėjo pasikeisti;
remtis išoriniais šaltiniais	sistema papildomai neapkraunama (į resursus reikalingus komunikacijai neatsižvelgiama)	šaltinis sistemos išorėje – įnešamos papildomos grėsmės (gali būti pažeistas vientisumas, konfidencialumas, prieinamumas)

Sprendimo logiką sudaro taisyklės, kuriomis remiantis yra priimami sprendimai. Šios taisyklės yra generuojamos naudojant Wang ir Mendel pasiūlytą metodą. Tikėtina, jog gali atsirasti taisyklių, kurios prieštarauja vienos kitoms, bei taisyklių, turinčių vienodas įėjimų reikšmes, tačiau skirtingus rezultatus. Taisyklių teisingumo problemos sušvelninimas yra naudoti jų kintamųjų tikrumo reikšmes – kuo kintamieji yra tikresni, tuo taisyklė yra tikresnė:

$$T(r_i) = \mu(x_{i,1}) \times \mu(x_{i,2}) \times \mu(y_i) \quad (24)$$

Čia:  $x_{i,j}$  –  $i$  taisyklės  $j$  veiksnio kintamasis;

$T$  – taisyklės tikrumas;

$\mu$  – kintamojo tikrumas;

$y_i$  – rezultato kintamasis.

Šiuo būdu yra sudaromas taisyklių rinkinys, kur kiekviena taisyklė turi jai priskirtą tikrumo įvertį. Šis rinkinys yra išskaidomas į grupes pagal vienodas aktualių veiksnių reikšmes:

$$\{r_1: \text{jei } x_1 = \text{Lingvar}_1 \text{ ir ... ir } x_k = \text{Lingvar}_k \text{ tai } y = \text{Lingvar}_{y_1}; T(r_1)\} \quad (25)$$

$$\{r_1: \text{jei } x_1 = \text{Lingvar}_1 \text{ ir ... ir } x_k = \text{Lingvar}_k \text{ tai } y = \text{Lingvar}_{y_1}; T(r_1)\} \quad (26)$$

$$\{r_2: \text{jei } x_1 = \text{Lingvar}_1 \text{ ir ... ir } x_k = \text{Lingvar}_k \text{ tai } y = \text{Lingvar}_{y_2}; T(r_2)\} \quad (27)$$

...

$$\{r_i: \text{jei } x_1 = \text{Lingvar}_1 \text{ ir ... ir } x_k = \text{Lingvar}_k \text{ tai } y = \text{Lingvar}_{y_j}; T(r_i)\} \quad (28)$$

Iš šios grupės parenkama viena taisyklė, kurios tikrumo reikšmė  $T(r_i)$  yra didžiausia. Literatūroje [37] yra siūloma įvesti papildomą tikrumo dedamąją – eksperto nuomonę  $h$ . Tai galima panaudoti taisyklėms, įdiegtoms sistemos sukūrimo metu:

$$T(r_i) = \mu(x_{i,1}) \times \mu(x_{i,2}) \times \mu(y_i) \times h_i \quad (29)$$

Po taisyklės sukūrimo tikrumas tampa nekintančiu įverčiu. Tikrumą susiejus su taisyklės panaudojimo kiekiu būtų suformuota savotiška „atmintis“ – kiekvieną kartą, kai scenarijaus veiksmas atliekamas su kitokiomis veiksnių reikšmėmis taisyklės tikrumas mažinamas, atliekant su tomis pačiomis – didinamas naudojant koeficientą  $m$ :

$$T(r_i) = \mu(x_{i,1}) \times \mu(x_{i,2}) \times \mu(y_i) \times h_i \times (1 \pm m) \quad (30)$$

### 2.3. Projektavimo išvados

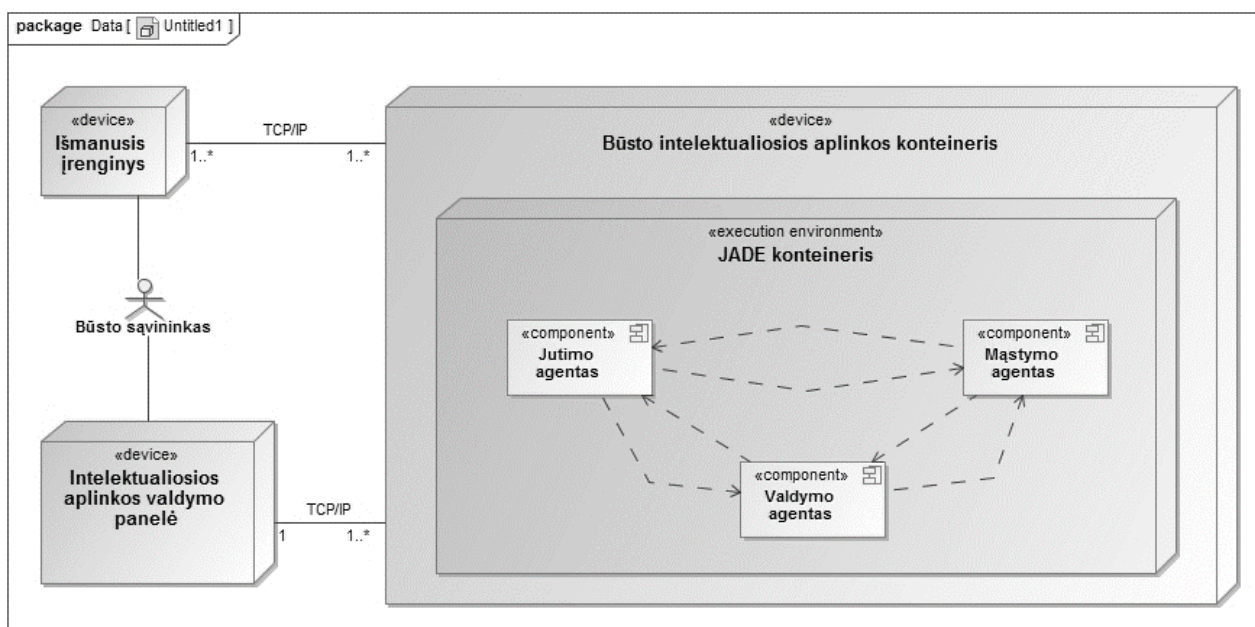
Šiame skyriuje buvo atliktas BIA taikymų lygmens sistemos modelio ir BIA taikymų lygmens saugos sistemos modelio projektavimas. Siekiant išgryninti virtualaus BIA modelio struktūrą buvo išskirti trys intelektualųjų esybių lygmenys – suvokimo lygmuo, sprendimų priėmimo lygmuo ir valdymo lygmuo. Atsižvelgiant į šiuos lygmenis buvo suformuotas hierarchinis BIA modelis, atskleidžiantis kuriamos sistemos aktorių ryšius. Svarbu pabrėžti, jog aktoriais laikomi ir sistemos naudotojai ir sistemoje realizuotos intelektualiosios esybės – agentai. Sudarius hierarchinį BIA modelį buvo apibrėžtos skirtingų lygmenų esybių funkcijos. Remiantis funkcijomis buvo sudarytas koncepcinis veiklos objektų modelis, atskleidžiantis vidinę esybių struktūrą, reikalingą šioms funkcijoms realizuoti virtualiame modelyje. Pastebėta, jog mąstymo agentas yra sudėtingiausias komponentas. Buvo sudaryta šio komponento būsenų modelis. Remiantis sukurtais modeliais suformuotas BIA komponentų modelis.

Suformavus architektūrinę dalį buvo pereita prie veikimo projektavimo. Nuspręsta, jog mąstymo agentų veikimas ir sistemos teikiamos sąlygos bus apibrėžtos scenarijais. Tam, kad šiuos scenarijus būtų galima įgyvendinti, buvo apibrėžta sprendimų priėmimo įeiga – veiksniai, bei išeiga – veiksmi. Nuspręsta, jog tam, kad sistemos esybės gebėtų funkcionuoti yra reikalingi du tarpiniai elementai – registrai, kuriuose bus saugomi veiksniai – tai, kokią informaciją gali suteikti suvokimo lygmuo, bei veiksmi – tai, kokius veiksmus gali atlikti valdymo lygmuo. Išskirti trys agento elgsenų tipai – ciklinė, periodinė bei vienkartinė.

Sprendimų priėmimui realizuoti buvo pasirinkta miglotoji logika. Šios logikos naudojimo savybės buvo apjungtos su analizės dalyje apibrėžta funkcinė agento organizacija ir suformuota BIA intelektualiosios esybės funkcinė organizacija. Šiuo žingsniu buvo baigtas BIA projektavimas. BIA taikymų lygmens saugumo sistemos projektavimo dalyje buvo orientuotasi į intelektikos apsaugojimą nuo neteisingos konfigūracijos. Išskirti trys būdai padedantys sustiprinti suvokimo lygmens saugą – remtis keliais informacijos šaltiniais, atsižvelgti į veikimo istoriją, bei įtraukti išorinių šaltinių informaciją. Sistemos taisyklių generavimo saugai pagerinti buvo nuspręsta naudoti Wang ir Mendel metodą, nes jis leidžia įvertinti taisyklių „gerumą“ pagal jų tikrumą. Buvo nuspręsta modifikuoti šį būdą įvedant koeficientą priklausantį nuo taisyklių naudojimo sistemoje kiekio.

### 3. BIA TAIKYMŲ LYGMENS SAUGUMO SISTEMOS TYRIMAS

Tyrimas buvo pradėtas nuo kompiuterinio būsto intelektualiosios aplinkos sukūrimo. Sistemos diegimo diagramoje yra nurodytas būsto savininkas (14 pav.). Tai sistemos aktorius, galintis valdyti ją naudodamas valdymo panelę ar susijusius išmaniuosius įrenginius. Kaip apibrėžta anksčiau, sistema gali veikti autonomiškai. Modeliui sukurti buvo naudota Java programavimo kalba. Agentų logikai realizuoti buvo naudojama jFuzzyLogic biblioteka, agentai realizuoti naudojant JADE platformą. Diagramoje išskirtas JADE platformos konteineris, kuriame veikia sistemos esybės – agentai.



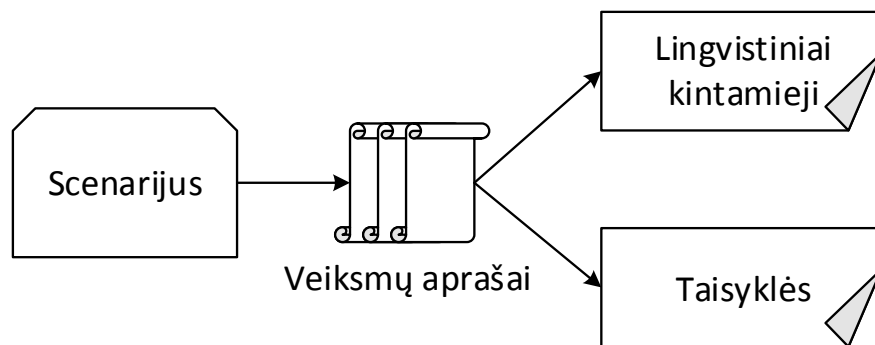
14 pav. BIA diegimo diagrama

Valdymo panelės grafinė sąsaja buvo sukurta naudojant JavaFX platformą (15 pav.). Ją naudodamas naudotojas gali rankiniu būdu įsiterpti į sistemos autonomišką veikimą ir valdyti būste esančius išmaniuosius daiktus.



**15 pav. Kompiuterinio modelio grafinė sąsaja**

JADE platformoje sukurti agentai susieti su šia valdymo panele naudojant Spring platformą. Naudojant panelę sistema reaguoja lyg būtų pakeista išmaniojo įtaiso būsena. Sistemos inicializacijos metu esybės įkelia išorinio eksperto suformuotą logiką. Ši logika yra aprašoma FCL [40] kalba. Sistemos scenarijai yra suformuojami šiais sąrašais (16 pav)



**16 pav. FCL aprašyto scenarijaus sandara**

Scenarijus yra sudarytas iš veiksmų rinkinio. Šie veiksmai yra aprašomi FCL kalba. Vienas aprašas atitinka vieną veiksmą. Aprašą sudaro du komponentai – lingvistiniai kintamieji ir taisyklės. Lingvistinių kintamųjų aprašais suformuojama agento nuomonės logikos duomenų bazė, taisyklėmis – sprendimų priėmimo duomenų bazė. Pavyzdyje pateiktas supaprastintas kavos gaminimo sprendimo priėmimo veiksmo aprašas. Lingvistinių kintamųjų aprašai suformuoti šiais blokais:

```

VAR_INPUT
  dow : REAL;
  hod : REAL;
END_VAR

VAR_OUTPUT
  coffee : REAL;
END_VAR

FUZZIFY dow
  TERM mon := trian 0.5 1 1.6;
  TERM tu := trian 1.4 2 2.6;
  TERM we := trian 2.4 3 3.6;
  TERM thu := trian 3.4 4 4.6;
  TERM fri := trian 4.4 5 5.6;
  TERM sat := trian 5.4 6 6.6;
  TERM sun := trian 6.4 7 7.6;
END_FUZZIFY

FUZZIFY hod
  TERM earlyMorning := trian 5 7 9;
  TERM lateMorning := trian 8 10 12;
  TERM noon := trian 11 12 14.5;
  TERM afternoon := trian 14 15 17.5;
  TERM evening := trian 17 19 21;
  TERM lateEvening := trian 20 22 24.5;
  TERM night := trian 0 3 6;
END_FUZZIFY

DEFUZZIFY coffee
  TERM sure := trian 0 1 2;
  TERM nah := trian 1 2 3;
  METHOD: COG;
  
```

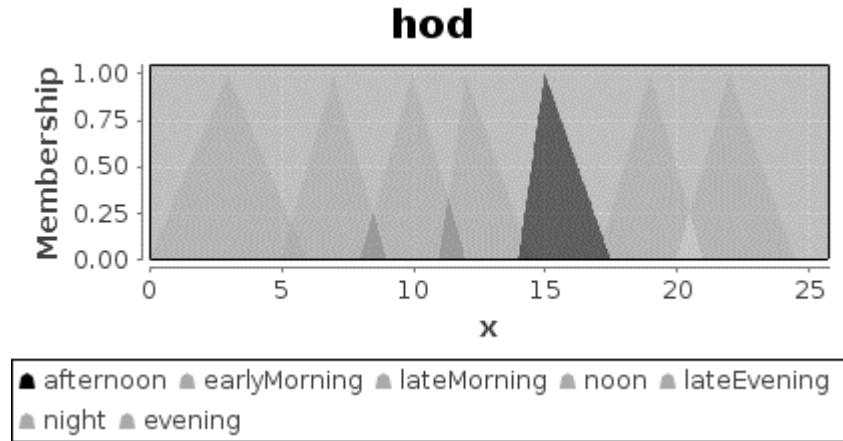


```

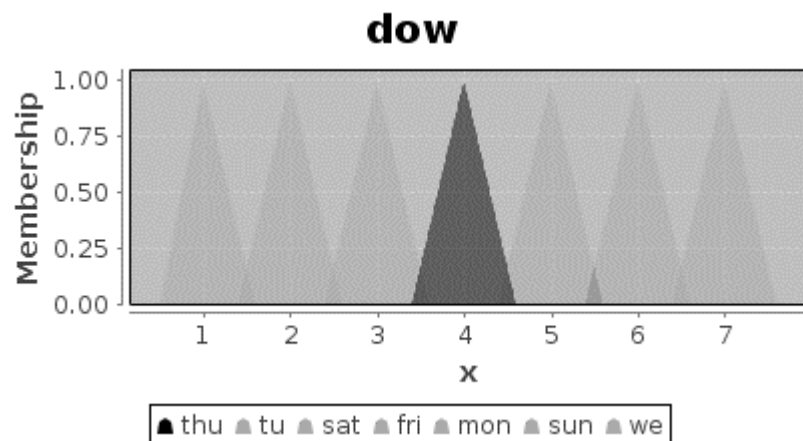
    DEFAULT := 0;
END_DEFUZZIFY

```

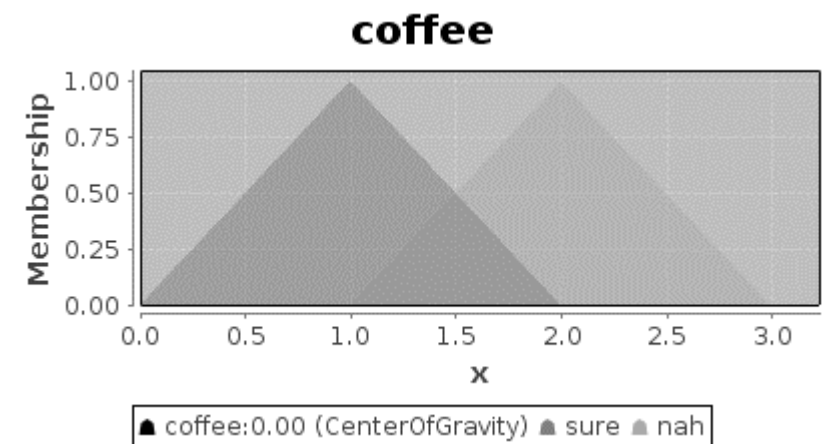
Aktualūs aplinkos veiksnaiai – savaitēs diena (*dow*) bei paros metas (*hod*). Rezultatas – ar priimtas sprendimas gaminti kava.



17 pav. Paros meto grafikas



18 pav. Savaitēs dienos grafikas



19 pav. Rezultato grafikas

Rezultato gavimo taisyklės aprašytos šia forma:

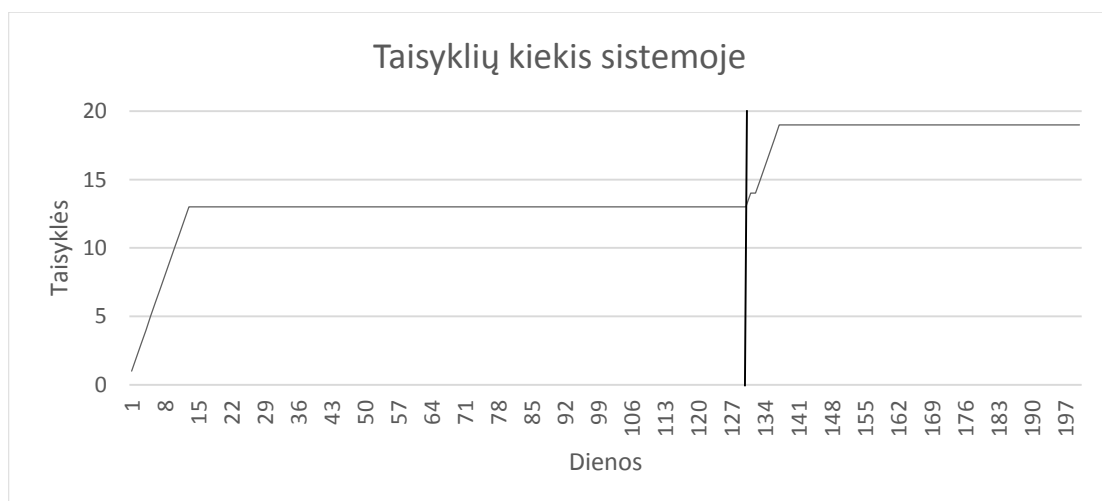
```

RULEBLOCK rules
  AND: MIN;
  ACT: MIN;
  ACCU: MAX;

  RULE 1: IF dow IS mon AND hod IS earlyMorning THEN coffee
IS sure;
  RULE 2: IF dow IS tu AND hod IS earlyMorning THEN coffee
IS sure;
  RULE 3: IF dow IS we AND hod IS earlyMorning THEN coffee
IS sure;
  RULE 4: IF dow IS thu AND hod IS earlyMorning THEN coffee
IS sure;
  RULE 5: IF dow IS fri AND hod IS earlyMorning THEN coffee
IS sure;
  RULE 6: IF dow IS sat AND hod IS noon THEN coffee IS sure;
  RULE 7: IF dow IS sun AND hod IS lateMorning THEN coffee
IS sure;
END_RULEBLOCK

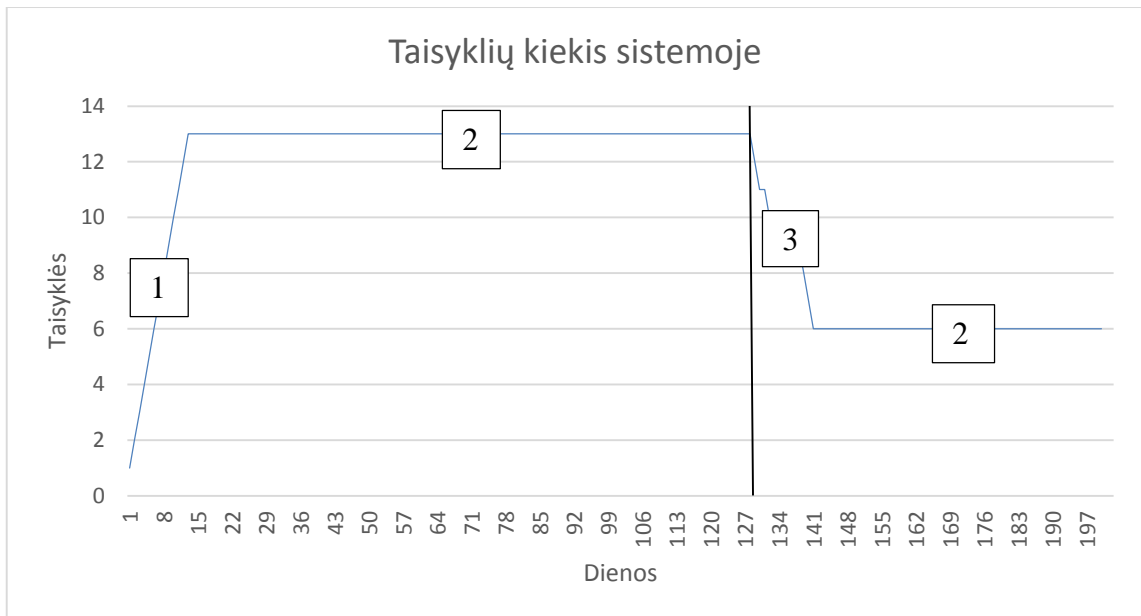
```

Kad atlikti eksperimentą buvo suformuoti du scenarijai kuriais buvo modeliuojama naudotojo sąveika su sistema. Pirmuoju atveju atliktas tik būsto intelektualiosios aplinkos ir naudotojo sąveikos modeliavimas. Scenarijus: naudotojas 10 savaitių geria kavą (pirmoji dalis), po dešimties savaitių jo įpročiai pasikeičia – kavą geria kitu metu, arba iš viso negeria (antroji dalis). Būsto intelektualiosios aplinkos tikslas – išmokti reikiamu metu naudotojui pagaminti kavą. Atlikus modeliavimą buvo sugeneruota 19 taisyklių. Išliko abu taisyklių rinkiniai – pirmosios ir antrosios dalies. Galima teigti jog sistema prisitaikė prie naudotojų įpročių, tačiau didelė dalis taisyklių yra senos ir netenkinančios naudotojų norų. Punktyrine linija pažymėtas perėjimas tarp scenarijaus dalių (20 pav).



**20 pav. Pirmojo eksperimento taisyklių generavimas**

Antrojo tyrimo atveju buvo naudojama aprašytas BIA taikymų lygmens saugumo sistemos modelis.

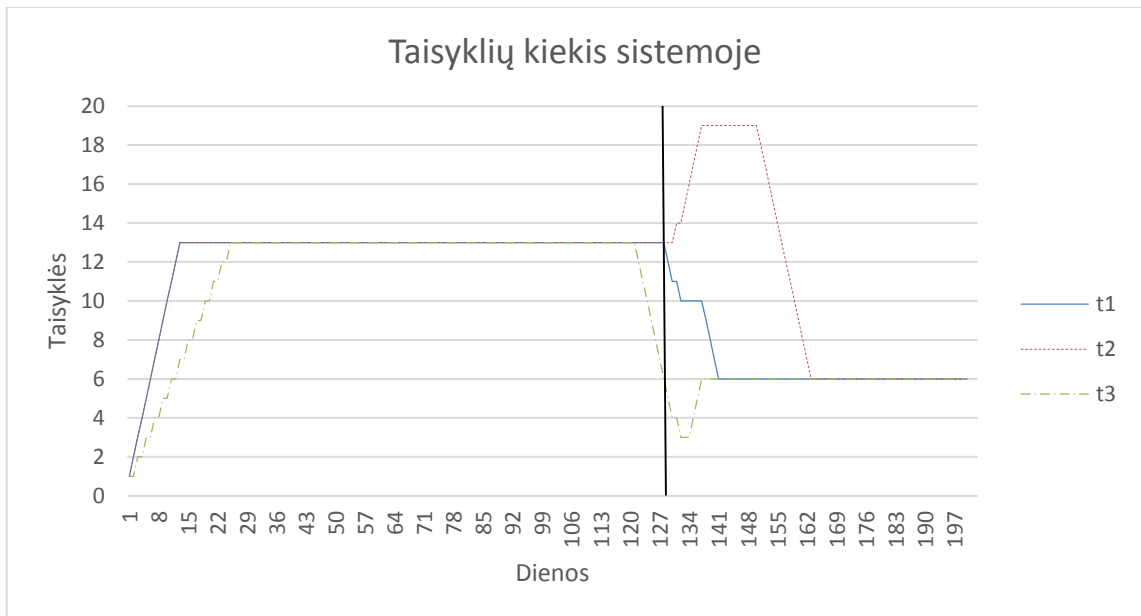


**21 pav. Antrojo eksperimento taisyklių generavimas**

Įdiegus šį modelį pastebėta, jog taisyklių kiekis sistemoje kinta priklausomai nuo naudotojo elgesio. Galima išskirti šiuos veikimo etapus (21 pav):

- 1) stebima naudotojo elgsena, formuojamos naujos veikimo taisyklės;
- 2) naudotojo elgsenos atžvilgiu sistema veikia optimaliai – sėkmingai nuspėjami jo veiksmai;
- 3) naudotojas pakeitė elgseną – sistema prisitaiko, netinkančios taisyklės yra pašalinamos iš atminties, tuo pat metu pridedamos naujos veikimo taisyklės.

Tiriamų scenarijų atveju sistemos prisitaikymo laikas yra ~ 2 savaitės. Tai paaiškinama tuo, jog naudotojo vykdomi veiksmai laiko atžvilgiu yra reti, ir tam, kad jų poveikis būtų reikšmingas reikalinga jog jie įvyktų tam tikrą kiekį kartų. Šio perėjimo periodą galima įtakoti pakeitus reikšmingo tikrumo režį, nuo kurio taisyklė laikoma galiojančia arba negaliojančia (22 pav).



**22 pav. Trečiojo eksperimento taisyklių generavimas**

Buvo naudojami trys taisyklių tikrumo režiai:  $t3 > t1 > t2$ . Esant per mažam tikrumo režiu taisyklių nusistovėjimas užtrunka – jos yra lėčiau pašalinamos iš sistemos. Didinant tikrumo režį ( $t3$ ) taisyklės sistemoje yra lėčiau kuriamos, tačiau greičiau iš jos pašalinamos.

## IŠVADOS

Šio darbo tikslas buvo sudaryti ir ištirti būsto intelektualiosios aplinkos taikymų lygmens saugos sistemos modelį. Buvo suformuotas koncepcinis būsimos sistemos modelis. Išskirti du darbo vykdymo etapai – būsto intelektualiosios aplinkos kūrimas bei būsto intelektualiosios aplinkos taikymų lygmens saugos sistemos kūrimas. Panašios sistemos literatūroje analizuojamos įvairiais pjūviais, pavyzdžiui, daiktai yra sugrupuojami pagal atliekamus veiksmus ir analizuojami šių veiksmų pažeidžiamumai, ar, aplinka yra suskirstoma į architektūrinius lygmenis ir analizuojami šių lygmenų paviršių pažeidžiamumai. Darbe buvo pasirinkta remtis sistemos aktorių išskyrimu ir jų sąsajų pažeidžiamumų analize. Projektavimo etape buvo nuspręsta naudoti smulkinamąjį projektavimą. Išskirti trys intelektualijų esybių lygmenys – suvokimo, sprendimų priėmimo bei valdymo lygmuo. Sistema buvo modeliuojama naudojant UML. Buvo apibrėžta hierarchija, veiklos objektai, komponentai, komunikacija bei funkcionavimas. Projektuojant saugos sistemą buvo orientuojamasi į intelektualumo formavimo metodus naudojant miglotąją logiką bei apsimokymo procesą. Nuspręsta jog sistemos „gerumas“ bus matuojamas tiriant taisyklių kiekį bei sistemos prisitaikymą prie kintančios naudotojo elgsenos.

1. Atlikus tiriamosios aplinkos ir taikymų analizę buvo išgryninta panašių sistemų plėtojimo kryptys. Pastebėta jog ryškėjantys panašių sistemų bruožai – esybių socialumas, autonomiškumas, bei gyvavimo laiko valdymas.
2. Atlikus panašių taikymų saugos analizę pastebėta jog literatūroje neapsiribojama vien konfidencialumu integralumu ir prieinamumu. Pabrėžiama, jog intelektualijų ir autonomiškų prietaisų atsiradimas sąlygoja naujų saugos užtikrinimo sprendimų poreikį.
3. Analizuojant tai, kas literatūroje yra laikoma intelektualumu pastebėta problema – nėra vieningo nusistatymo kas tai yra. Tai paaiškinama ir tuo, jog ši sritis nėra standartizuota ir laikoma eksperimentine. Nuspręsta, jog darbe bus remiamasi šiuo intelektualumo apibrėžimu: intelektualumas tai žmogaus veiksmas, kurie yra laikomi intelektualiais, t.y: situacijų atpažinimas ir klasifikavimas, veikimas vadovaujantis miglotomis taisyklėmis bei veikimas vadovaujantis suformuota tendencija.
4. Atlikus intelektualumo realizavimo būdų analizę buvo nuspręsta, jog darbe bus naudojama miglotoji logika. Šis sprendimas priimtas todėl, kad miglotoji logika leidžia supaprastinti aplinkoje vykstančius procesus bei išvengti neapibrėžtų situacijų.
5. Miglotoji logika yra lengvai suvokiama žmogui. Sistemos kūrėjams jis padeda supaprastinti tiriamąją sritį ir išgryninti paslaugas bei joms svarbius aktualius veiksnius. Sistemos naudotojai gali būti ekspertais, kurie nurodys kaip sistema turėtų veikti. Šiuo

atveju miglotosios logikos pranašumas – jų norus be didesnių pastangų galima tiesiogiai perkelti į sistemą.

6. Sukūrus BIA modelį pastebėta, jog sistema veikė su trūkumais, dėl priklausomybės nuo naudojamų įrankių – operacinės sistemos bei Java virtualios mašinos. Remiantis atlikta analize galima teigti jog kuriant sistemos prototipą ir toliau ją vystant svarbu atsisakyti papildomų architektūrinių sluoksnių, kurie didina riziką bei paverčia sistemą nestabilia.
7. BIA modelio bei BIA saugumo sistemos intelektualumo formavimas buvo įgyvendintas taikant Wang ir Mendel miglotųjų taisyklių generavimo metodą. Pastebėta, jog šis metodas netinkamas naudoti šioje situacijoje, kadangi buvo neatsižvelgiama į negaliojančias taisykles ir sistema nustodavo funkcionuoti pagal naudotojo elgseną.
8. Modifikavus Wang ir Mendel miglotojų taisyklių generavimo metodą pridėdant intelektualųjų esybių atminties efektą pastebėta jog saugos sistemą taisyklių kiekis tapo priklausomas nuo naudotojo periodiškai atliekamų veiksmų kiekio. Buvo naudojamos tik aktualesios taisyklės, o sistema prisitaikydavo prie kintančio naudotojo elgesio.
9. Atlikus palyginamąją taisyklių kiekio ir sistemos prisitaikymo prie kintančio naudotojo elgesio greičio analizę pastebėta jog nenaudojant saugos sistemos taisyklių buvo generuojama daugiausiai, dėl to mažėjo veikimo efektyvumas.
10. Pastebėta, jog sistemos prisitaikymas prie pakitusio elgesio tiesiogiai priklauso nuo atliekamų veiksmų periodiškumo, bei kaip reikšmingu laikomas naudotojo elgesys.

Plėtojant darbus šioje srityje būtų reikšminga apibrėžti intelektualumo „gerumo“ požymius ir pagal juos atlikti intelektikos formavimo būdų efektyvumo palyginamąją analizę. Literatūroje siūlomi įvairūs būdai kaip galima valdyti intelektualųjų esybių apsimokymo procesą, tačiau nėra šių metodų analizės panašioje taikymų srityje. Žinant konkrečius šių metodų privalumus ir trūkumus būtų galima paprasčiau priimti sprendimą kurį iš jų naudoti. Darbe sistemos modeliavimui buvo naudojama JADE platforma. Ši platforma veikia virtualioje JAVA mašinoje, kuri įdiegta suderinamoje operacinėje sistemoje. Visi šie papildomi komponentai sumažina efektyvumą bei padidina pavojų riziką. Kadangi panašius eksperimentus stengiamasi taikyti realiose aplinkose svarbu atlikti tyrimą iš naudotojų bei gamintojų pozicijos bei pasiūlyti realius realizavimo būdus. Ši sritis nėra standartizuota – realių taikymų pagrindas yra vieningos tokių sistemų kūrimo žinių bazės suformavimas.

#### 4. LITERATŪRA

- [1] Horizon 2020 work programme 2014-2015
- [2] Bassi, A., and G. Horn. "Internet of Things in 2020." In *Joint European Commission/EPoSS expert workshop on RFID/Internet-of-Things*. 2008
- [3] Bari, Nima, Ganapathy Mani, and Simon Berkovich. "Internet of things as a methodological concept." In *Computing for Geospatial Research and Application (COM. Geo), 2013 Fourth International Conference on*, pp. 48-55. IEEE, 2013.
- [4] Coetzee, Louis, and Johan Eksteen. "The Internet of Things-promise for the future? An introduction." In *IST-Africa Conference Proceedings, 2011*, pp. 1-9. IEEE, 2011
- [5] Gou, Quandeng, Lianshan Yan, Yihe Liu, and Yao Li. "Construction and Strategies in IoT Security System." In *Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (iThings/CPSCoM), IEEE International Conference on and IEEE Cyber, Physical and Social Computing*, pp. 1129-1132. IEEE, 2013.
- [6] Lau, Raymond YK, Yunqing Xia, and Yunming Ye. "A probabilistic generative model for mining cybercriminal networks from online social media." *Computational Intelligence Magazine*, IEEE 9, no. 1 (2014): 31-43.
- [7] Ning, Huansheng, Hong Liu, and Laurence T. Yang. "Cyberentity security in the Internet of Things." *Computer* 4 (2013): 46-53.
- [8] Jiang, Du, and Chao ShiWei. "A study of information security for M2M of IOT." In *Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on*, vol. 3, pp. V3-576. IEEE, 2010.
- [9] Venčkauskas, A.; Kazanavičius, E. *Informacinių technologijų saugos metodai*, 2011, p. 11-16
- [10] Tom Gruber, *Ontology [Tinkle]* Available: <http://tomgruber.org/writing/ontology-definition-2007.htm> - [Kreiptasi 2013 gruodžio 12]
- [11] Suo, Hui, Jiafu Wan, Caifeng Zou, and Jianqi Liu. "Security in the internet of things: a review." In *Computer Science and Electronics Engineering (ICCSEE), 2012 International Conference on*, vol. 3, pp. 648-651. IEEE, 2012.
- [12] Riahi, Arbia, Yacine Challal, Enrico Natalizio, Zied Chtourou, and Abdelmadjid Bouabdallah. "A systemic approach for IoT security." In *Distributed Computing in Sensor Systems (DCOSS), 2013 IEEE International Conference on*, pp. 351-355. IEEE, 2013.
- [13] McKinsey Quarterly. *The Internet of Things [Tinkle]* Available: [http://www.mckinsey.com/insights/high\\_tech\\_telecoms\\_internet/the\\_internet\\_of\\_things](http://www.mckinsey.com/insights/high_tech_telecoms_internet/the_internet_of_things) - [Kreiptasi 2013 lapkričio 16]

- [14] Zhang, Baoquan, Zongfeng Zou, and Mingzheng Liu. "Evaluation on security system of internet of things based on fuzzy-AHP method." In E-Business and E-Government (ICEE), 2011 International Conference on, pp. 1-5. IEEE, 2011.
- [15] Khan, Rafiullah, Sarmad Ullah Khan, Rifaqat Zaheer, and Shahid Khan. "Future Internet: The Internet of Things Architecture, Possible Applications and Key Challenges." In FIT, pp. 257-260. 2012.
- [16] Jun, Yu, and Wang Yang. "Internet of things: System framework, applications and attentions in program operation." In Computational Problem-Solving (ICCP), 2011 International Conference on, pp. 15-18. IEEE, 2011.
- [17] Li, Lan. "Study on security architecture in the Internet of Things." In Measurement, Information and Control (MIC), 2012 International Conference on, vol. 1, pp. 374-377. IEEE, 2012.
- [18] Xiaohui, Xu. "Study on security problems and key technologies of the internet of things." In Computational and Information Sciences (ICCIS), 2013 Fifth International Conference on, pp. 407-410. IEEE, 2013.
- [19] Murad, Adil, Salim Ullah, Arif Ali Shah, Laiq Hasan, Kiramat Ullah Khan, and Adnan Ahmed Khan. "Design and implementation of an intelligent energy saving system." In Emerging Technologies (ICET), 2014 International Conference on, pp. 135-138. IEEE, 2014.
- [20] Hong, Hui, Weikang Wu, and Kaiyuan Wang. "Design and implementation of environmental monitoring system in intelligent home." (2011): 522-526.
- [21] Suh, Changsu, and Young-Bae Ko. "Design and implementation of intelligent home control systems based on active sensor networks." *Consumer Electronics, IEEE Transactions on* 54, no. 3 (2008): 1177-1184.
- [22] Babar, Sachin, Antonietta Stango, Neeli Prasad, Jaydip Sen, and Ramjee Prasad. "Proposed embedded security framework for internet of things (iot)." In Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology (Wireless VITAE), 2011 2nd International Conference on, pp. 1-5. IEEE, 2011.
- [23] Sundmaeker, Harald, Patrick Guillemin, Peter Friess, and Sylvie Woelfflé. "Vision and challenges for realising the Internet of Things." (2010).
- [24] Russell, Stuart, Peter Norvig, and Artificial Intelligence. "A modern approach." *Artificial Intelligence*. Prentice-Hall, Egnlewood Cliffs 25 (1995)
- [25] Wooldrige, Michael. "An introduction to multiagent systems." (2002).
- [26] Baranauskas, Rimantas, Audrone Janaviciute, Raimundas Jasinevicius, Vaidas Jukavicius, Egidijus Kazanavicius, Vytautas Petrauskas, and Arunas Vrubliauskas. "On Multi Agent Systems Intellectics." *Information Technology And Control* 44, no. 1 (2015): 112-124.



- [27] Jasinevicius, Raimundas, and Vytautas Petrauskas. "Fuzzy expert maps: The new approach." In FUZZ-IEEE, pp. 1511-1517. 2008.
- [28] Vidal, José M., and José M. Vidal. "Fundamentals of multiagent systems with NetLogo examples." (2007).
- [29] Menezes, Alfred J., Paul C. Van Oorschot, and Scott A. Vanstone. Handbook of applied cryptography. CRC press, 1996.
- [30] Sanayei, Ali, and Otto E. Rössler. ISCS 2014: Interdisciplinary Symposium on Complex Systems. Edited by Ivan Zelinka. Vol. 14. Springer, 2014.
- [31] Jennings, Nicholas R., and Michael Wooldridge. "Applying agent technology." Applied Artificial Intelligence an International Journal 9, no. 4 (1995): 357-369.
- [32] Available: [http://all-free-download.com/free-vector/vector-misc/interior\\_line\\_art\\_drawing\\_vector\\_295306.html](http://all-free-download.com/free-vector/vector-misc/interior_line_art_drawing_vector_295306.html) [Kreiptasi 2015 kovo 25]
- [33] Takagi, Tomohiro, and Michio Sugeno. "Fuzzy identification of systems and its applications to modeling and control." *Systems, Man and Cybernetics, IEEE Transactions on* 1 (1985): 116-132.
- [34] Tomohiro Takagi, Michio Sugeno. Fuzzy Identification of Systems and Its Applications to Modelling and Control. IEEE Transactions on systems, man, and cybernetics, vol smc-15 no1 jan/feb 1985 p 116-133
- [35] Wang, L-X., and Jerry M. Mendel. "Generating fuzzy rules by learning from examples." *Systems, Man and Cybernetics, IEEE Transactions on* 22, no. 6 (1992): 1414-1427.
- [36] Alcalá, R., J. Casillas, O. Cordón, F. Herrera, and I. Zwir. "Techniques for learning and tuning fuzzy rule-based systems for linguistic modeling and their applications." *Knowledge Engineering Systems, Techniques and Applications* 3 (1999): 889-941.
- [37] Adriano Oliveira Cruz. ANFIS: Adaptive Neuro-Fuzzy Inference. Available: <http://equipe.nce.ufrj.br/adriano/fuzzy/transparencias/Learning/Learninghandout.pdf> [Kreiptasi 2015 balandžio 11]
- [38] Bellifemine, Fabio Luigi, Giovanni Caire, and Dominic Greenwood. Developing multi-agent systems with JADE. Vol. 7. John Wiley & Sons, 2007.
- [39] Sakita, Mauro T., Joao GS Figueiroa, and Homero AF Souza. "Vibration Satellite Qualification Testing by Employing Wang & Mendel Learning Algorithm and Takagi-Sugeno Fuzzy Modelling." (2009).
- [40] IEC 1131 - PROGRAMMABLE CONTROLLERS, Part 7 - Fuzzy Control Programming [Tinkle] Available: <http://www.fuzzytech.com/binaries/ieccd1.pdf> [Kreiptasi: 2015-04-12]

## 5. PRIEDAI

### 5.1. priedas. Straipsnis

#### On Multi-Agent Systems Intellectics

Rimantas Baranauskas, Audrone Janaviciute, Raimundas Jasinevicius, Vaidas Jukavicius,  
Egidijus Kazanavicius, Vytautas Petrauskas, Arunas Vrubliauskas

*Centre of Real Time Computer Systems, Kaunas University of Technology,  
K. Barsausko st. 59-A313, LT-51423 Kaunas, Lithuania*

*e-mail: rimantas.baranauskas@ktu.lt, audrone.janaviciute@ktu.lt, raimundas.jasinevicius@ktu.lt,  
vaidas.jukavicius@ktu.lt, egidijus.kazanavicius@ktu.lt, vytautas.petrauskas@ktu.lt,  
arunas.vrubliauskas@ktu.lt*

**crossref** <http://dx.doi.org/10.5755/j01.itc.44.1.8768>

**Abstract.** This paper intends to precisiate the well-known and widespread definitions of both smart and intelligent agent (SA; IA), as well as the smart and intelligent multi-agent system (SS/II\_MAS). The use of a unified and standardized agent and multi-agent system description based on definitions of the general systems theory is delivered and proposed as well. The intellectics of multi-agent systems is considered as a kind of an extension of the agent intelligence. Three typical features of human intellectual activities are proposed to be implemented and simulated in an agent/multi-agent system as the basic paradigms for agent and multi-agent system intellectics. As underlined in the paper, operation according to those paradigms (recognition and classification, behavior according to a set of fuzzy rules, and operation according to some prescribed tendency) is solidly mathematically based (correspondingly: mathematical programing, fuzzy logic and stochastic approximation). Finally, results of computerized modeling and simulation are delivered demonstrating the practical vitality and efficiency of the theoretical approach to the realization of the intelligent environment of the Internet of Things and Services (IoT&S) for user's comfort in two projects: "Research and Development of Internet Infrastructure for IoT& S in the Smart Environment (IDAPI)" and "Research on Smart Home Environment and Development of Intelligent Technologies (BIATech)".

**Keywords:** Multi-agent systems; smart agents; intelligent agents; intelligent environment; fuzzy systems.

#### 1. Introduction, Related Work and Motivation

An agent-based approach to various engineering applications, especially to those which involve IT-enabled technologies, is very popular nowadays for two reasons. The first one – the term "agent" itself is very attractive because it appears mysterious and for this reason is suitable for the purpose of advertising new products, and it sounds scientific enough for the researcher circles. The second reason is the fuzziness of agent definitions.

Starting with encyclopaedical fundamentals of the multi-agent systems approach delivered in [1] and [7], researchers can find neither a precise definition on agent nor even a scientific concept to be used for the construction of agent definitions.

So, according to these popular definitions, an agent is an entity, a piece of software or a computer system that functions in an environment in order to meet its design objectives. And if this behavior is autonomous, the agent is called intelligent.

According to the literature [2-6], as well as [8] and [9], an agent is:

- "an entity that senses its environment and acts upon it";
- "an entity that functions continuously and autonomously in an environment in which other processes take place and other agents exist";
- "a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives";
- "Intelligent agent is a computer system capable of flexible autonomous actions in order to meet its design objectives";
- "Intelligent agents are software entities that carry out some set of operations on behalf of a user or another program, with some degree of independence and autonomy, and in doing so, employ some knowledge or representation of the user's goals or desires".

Such uncertainty in this field of knowledge served as a strong motivation for our research.

We are fully aware that even a mere attempt to precisiate any term is dangerous, but we will take the

risk of disclosing the meaning of agent intellectics, or at least put it in a clearly understandable content and attempt to demonstrate its practicability.

Thus, the goal of this paper is threefold: 1) to present our point of view concerning the definition under discussion; 2) to deliver our ideas covering the problem of agent intelligence and 3) to show the intellectics of multi-agent systems as a kind of an extension of the agent intelligence.

Naturally, in a case of several interacting agents present we have a multi-agent system. And it must be emphasized that the intellectics of agent or multi-agent system is hidden in those two mysterious terms: "autonomous" and "design objectives". We intend to cover those terms by putting some contemporary mathematics and soft computing modeling of three most simple features of human beings widely considered as his/her intellectual activity: 1) recognition and classification (of patterns, processes, situations); 2) behavior according to a set of fuzzy rules and 3) operation according to some prescribed tendency. We emphasize the novelty of such an approach. Roughly speaking, the models of those three activities mentioned above, from a mathematical point of view, are covered by mathematical programming, fuzzy logic and soft computing, and stochastic approximation respectively.

The last part of this paper contains results of simulations demonstrating the features of agent intellectics.

## 2. Definitions

It is important to note that, for example, the <http://scholar.google.it> presents more than 208000 entries according to the item "computerized intelligent agent definition pdf" (2015.01.15). An existence of numerous different agent definitions found in literature suggests the necessity to unify and propose a more precise and systemic one. Here, we deliver our own system of agent definitions based on the properties of its functional organization.

So, an agent (A) is a software/hardware entity which interacts with the environment in a prescribed way and as such - inherits a functional organization based on a deterministic or stochastic (fuzzy) description of its external activity and internal operations. If an agent can act without any programming activity coming from a user, we have a smart or an intelligent agent.

When agent's external interaction with the environment and its internal operations are based on a crisp algorithmic logic, we have a *smart agent* (SA).

When agent's external interaction with the environment and its internal operations are based on fuzzy algorithmic logic, we have an *intelligent agent* (IA).

A set of interacting agents is usually considered as a multi-agent system (MAS).

The level of MAS intellectics depends on these types: 1) the type of agents in the MAS and 2) the type of interactions between agents in the MAS. There are four types of MAS formed from possible combinations. These types are presented in Table 1.

Table 1. MAS types

Type of agent \ Type of interaction	SMART	INTELLIGENT
SA	SS_MAS	SI_MAS
IA	IS_MAS	II_MAS

Both smart and intelligent agent interaction in the MAS may be performed either in a relatively static or a dynamic way. In the case of a relatively static interaction, the graph of information flow is defined and prescribed in advance; only the timing and the content of the information flow is subjected to smart or intelligent changes. Examples of a relatively static MAS are presented in Fig. 1 and Fig. 2, where interconnected agents  $A_1$ - $A_6$  act upon environmental entities  $E_1$ - $E_4$ , and agents  $A_1$ - $A_{10}$  act upon entities  $E_1$ - $E_{10}$ .

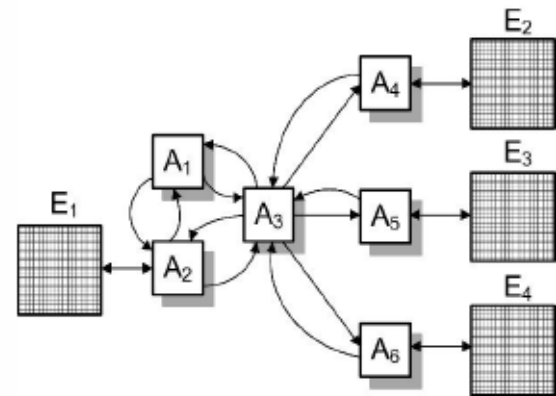


Figure 1. Specialized MAS

Here, the circles correspond to certain zones of agents' interacting activity. The size of each zone depends on the distance between all possible pairs of agents. Here, the term "distance" is used not in a geometrical or geographical but rather in a physical way, determining the distance between situations in which the agent  $A_A$  and the agent  $A_B$  function. So, if the situation of agent  $A_A$  is described by a set of measurable characteristics  $a_i, i=1, \dots, N$  and the situation of agent  $A_B$  - by a set of  $b_i, i=1, \dots, N$ , then the Minkovski distance is

$$D_{AB} = (\sum_{i=1}^N |a_i - b_i|^p)^{\frac{1}{p}} \quad (1)$$

for  $p \in \mathbb{R}$  ( $\mathbb{R}$  is a set of real numbers).

Usually, the value of  $p$  is determined by practical considerations.

If  $D_{AB} > D_0$ , where  $D_0$  is a prescribed minimum accepted distance, then  $A_A$  or  $A_B$  communicates and interacts. It must be emphasized that  $D_0$  changes

intelligently according to the MAS and the environmental context.

Such an approach permits us to consider MAS as a smartly or intelligently augmented agent and to use the same general description for both entities.

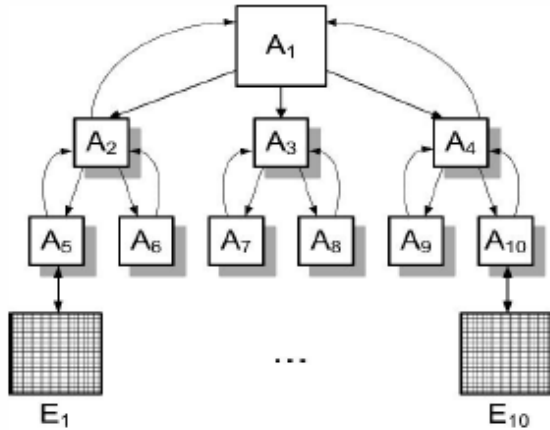


Figure 2. Hierarchical MAS

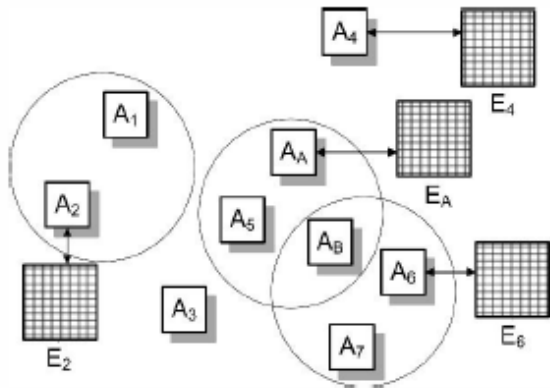


Figure 3. MAS Based On Distance and/or Context Description

### 3. General Description

The general description of an agent (and/or MAS) must be based on a formalized description of the general system theory. Moreover, when this formalization is performed, the fact that the contemporary agent is a space-time entity must be taken into account.

The best attempt to formalize the description of space – time dependent systems was proposed by prof. Gerhard Wunch from Dresden University in 1975 [10]. Using a similar approach, the agent  $A$  is presented by a row consisting of five sets of variables  $X, K, Y, R, T$  and of two functional transformations  $\Phi, \Psi$ :

$$A = \{X, K, Y, R, T, \Phi, \Psi\}; \quad (2)$$

Here,  $X$  is a set of inputs  $I$ ,  $K$  is a set of agent internal states  $S$ ,  $Y$  is a set of agent outputs  $O$ ,  $R$  is an independent space variable, and  $T$  is an independent time variable;

$$\Phi: X \times K \times R \times T \rightarrow K \quad (3)$$

and

$$\Psi: K \times R \times T \rightarrow Y \quad (4)$$

is a transformation of the internal states of the agent and a transformation of agent outputs respectively (see Fig. 4). Here we mean that all dependent variables are functions of the space and time coordinates  $X(\rho, t)$  and  $Y(\rho, t)$ .

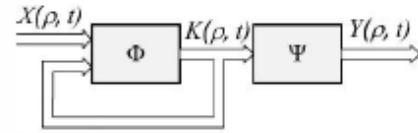


Figure 4. Transformations of Agent's (Or MAS's) Inputs and Internal States into Its Internal States and Outputs

The dimensionality of space usually depends on the problem under investigation. For example, in world financial activity we use two – dimensional space, while problems of environmental and/or marine modeling and simulation are tackled in three – dimensional space.

It must be emphasized that such an agent (or MAS) is able to represent and monitor global, real-life situations adequately only if both transformations ( $\Phi, \Psi$ ) are performed not only crisply but fuzzily, softly and/or using verbal computing mechanisms as well, along with a mixture of real-life variables comprising numerical, crisp, quantitative, deterministic, as well as fuzzy, verbal, qualitative, soft, stochastic, and uncertain (or even erroneous) information.

So, the intellectics of an agent ( $A$ ) or a multi-agent system (MAS) is determined by the operations of variables and a logic type implemented in the transformations  $\Phi$  and  $\Psi$ .

### 4. $\Phi$ and $\Psi$ Intellectics

As it was mentioned in the introduction, we intend to construct the  $\Phi$  and  $\Psi$  transformations (3) and (4) according to the three most simple features of human beings that are widely considered as his/her intellectual activity: 1) recognition and classification, 2) behavior according to a set of fuzzy rules, and 3) operation according to some prescribed tendency.

#### 4.1. MAS Intellectics Based On Situation Recognition

The approach to the problem mentioned in the title is based on the theoretical considerations and practical experience delivered in [11].

Let us imagine that our environment, in order to be intellectualized, must perform a certain action  $p$  that is adequate to the situation that has arisen in the environment. Moreover, similar situations must trigger the same action. And this is the reason why we call a group of similar situations that require the  $p$ -th action as a class of situations (let's say,  $p$ -class) generated

when they correspond to the  $p$ -th pattern. In the case of multiple actions (and multiple environmental situation patterns), we have  $p=1, 2, \dots, r, \dots, S$ .

Usually, each concrete situation is described by  $N$  features numbered as  $n=1, 2, \dots, i, \dots, N$ . In the case of certain feature extraction, measurement and normalization procedures are performed [14], the  $i$ -th feature of a situation that belongs to the  $p$ -th class (corresponds to the  $p$ -th pattern) can be represented by a real number  $\alpha_{pi}$  that expresses a degree of intensity of this particular feature. It is convenient to use a vector-row notation to describe the whole situation  $\vec{\alpha}_p = (\alpha_{p1}, \alpha_{p2}, \dots, \alpha_{pi}, \dots, \alpha_{pN})$ . If we have several situations (indexed by  $l=1, 2, \dots, k, \dots, L$ ) and know in advance that they are similar according to a certain one and belong to class  $p$  (they are originated by the  $p$ -th pattern), then we can say that the situations of class  $p$  are represented by a set of vectors  $\vec{\alpha}_p^{ol}$  ( $l=1, 2, \dots, k, \dots, L$ ).

As a rule, better reasoning results are achieved when features of situations are not only normalized but centered as well [12, 13]. It means that the whole situation is represented as a vector

$$\vec{\alpha}_p^{ol} = (\alpha_{p1}^{ol}, \alpha_{p2}^{ol}, \dots, \alpha_{pi}^{ol}, \dots, \alpha_{pN}^{ol}) \quad (5)$$

with components calculated according to the following formula:

$$\alpha_{pi}^{ol} = \alpha_{pi}^l - \frac{1}{N} \sum_{j=1}^N \alpha_{pj}^l \quad (6)$$

So, all available information about the patterns of situations is hidden in the set of  $\vec{\alpha}_p^{ol}$ , for  $\forall p$ , where  $p=1, 2, \dots, r, \dots, S$  and  $l=1, 2, \dots, k, \dots, L$ . Now, the main task is to determine or extract the significance of the pattern's features of each situation group (let's say, group  $p$ ) and to present them in a vector form known as the generalized situation's pattern (GSP)

$$\vec{K}_p = (K_{p1}, K_{p2}, \dots, K_{pi}, \dots, K_{pN}). \quad (7)$$

The problem can be easily solved if the corresponding linear programming problem (LPP) is formulated in the following way.

Let us select randomly one representative of the situation class  $p$ , for example  $\vec{\alpha}_p^{ok}$  (we will call it "central" in order to make it easier to understand). Let's say we need to find such  $K_{pi}$  for  $\forall i$  so that the measure of degree of certainty  $\Phi_p(\vec{\alpha}_p^{ok})$  of belonging of the selected situation  $k$  to the pattern  $p$  would be maximum:

$$\Phi_p(\vec{\alpha}_p^{ok}) = \sum_{i=1}^N \alpha_{pi}^{ok} K_{pi} \rightarrow \max \quad (8)$$

and it must be reached under the following constraints:

$$\sum_{i=1}^N \alpha_{pi}^{ol} K_{pi} \geq \gamma \sum_{i=1}^N \alpha_{pi}^{ok} K_{pi}, \text{ for } \forall l \quad (9)$$

and

$$\sum_{i=1}^N \alpha_{ri}^{ol} K_{pi} \leq \kappa \sum_{i=1}^N \alpha_{pi}^{ok} K_{pi}, \text{ for } \forall r, r \neq p$$

$$\text{and } \forall l. \quad (10)$$

It is recommended to choose optimal values of real numbers  $\gamma$  and  $\kappa$  from interval  $[0-1]$ , and  $\gamma > \kappa$  [12]. Specific values of those coefficients depend on the experts' knowledge or guess concerning the structure (internal connections and dispersion of patterns' features) of the pattern (or class). Physical meaning of (9) is tightly connected with the understanding of "positive similarities" inside the class  $p$ . The physical meaning of (10) corresponds to the concept of dissimilarities between certain patterns of situations (in our case – the pattern of class  $p$ ) and all other classes  $r$  (or "negative similarities") for  $\forall r, r \neq p$ . Even a quick overview of the problem described above shows that the problem really belongs to the class of linear programming problems (LPP) where inequalities (9) and (10) need additional constraints:

$$0 \leq \vec{K}_p \leq A \quad (11)$$

where  $A$  is any practically convenient real number, serving as maximum degree of importance and informativeness of features that describe the pattern of the  $p$ -th class of situations.

Naturally, the solution of the LPP (8)-(11) for the pattern of situations (class)  $p$  consists of the obtained value for

$$\max \Phi_p(\vec{\alpha}_p^{ok}) = \Phi_{p\max} \quad (12)$$

and the generalized pattern of situations for class  $p$ :

$$\vec{K}_p = (K_{p1}, K_{p2}, \dots, K_{pi}, \dots, K_{pN}). \quad (13)$$

The procedure must be repeated for all classes of situation patterns ( $\forall p$ ). In this way, a set of  $S$  solutions will be generated. The recognition procedure for the situation must be performed taking into account the need of fulfilling proportionality condition that guarantees the same numerical degree of certainty to the same qualitative evaluation of the situation using verbal definitions of the similarity between situations:

$$\begin{aligned} c_1 \Phi_{1\max} = \dots = c_p \Phi_{p\max} = \\ \dots = c_S \Phi_{S\max} = B \end{aligned} \quad (14)$$

where  $B$  and  $c_p$  are real numbers.

When the unknown situation  $\vec{x}^0$  is under consideration, its degree of belonging to the pattern  $p$  can be evaluated by

$$\Phi_p(\vec{x}^0) = \sum_{i=1}^N x_i^0 K_{pi} \text{ for } \forall p. \quad (15)$$

The maximum value can be considered as an argument for the environment's action.

A complex of such procedures enables us to construct a situation recognition instrument capable of assigning any unknown but properly described situation  $\vec{x}$  to one of the possible patterns (or classes) and perform the corresponding environmental action 1, 2, ...,  $r$ , ...,  $S$ .

Descriptions of real situations, as well as descriptions collected from different environmental models, standards or user's requirements are usually

collected in a certain data base (DB in Fig. 5(a)) and used to evaluate  $\vec{K}_p = (K_{p1}, K_{p2}, \dots, K_{pi}, \dots, K_{pN})$  – the significance of each parameter in the description of a situation.

Evaluation is performed according to the LPP procedure described above. Obtained results are used in the structure of a generalized description of the MAS (see Fig. 4) to perform the transformation of inputs (situation's description) into agent (or MAS) internal states, as well as the transformation of its internal states into outputs that determine environment's actions (Fig. 5(b)).

#### 4.2. MAS Intellectics Based On the Fuzzy Rules Approach

The approach to the problem mentioned in the title is based on the theoretical considerations and practical experience delivered in [13-15]. As it was stated in the previous subsection, the environment to be intellectualized will have to perform a certain action  $p$  which is adequate to the situation that has arisen in the environment. And as before, similar situations must trigger the same action. This similarity must be described fuzzily and the action must be performed by the MAS behaving as a certain fuzzy system.

The inference of ordinary fuzzy systems is usually based on: 1) derivation of verbal (linguistic) or parametric consequents by preprocessing lists of fuzzy rules that contain verbal or parametric antecedents connected by certain fuzzy logic operations and 2) a defuzzification process based on some compositional rule or formula.

Types of rules can be presented as follows:

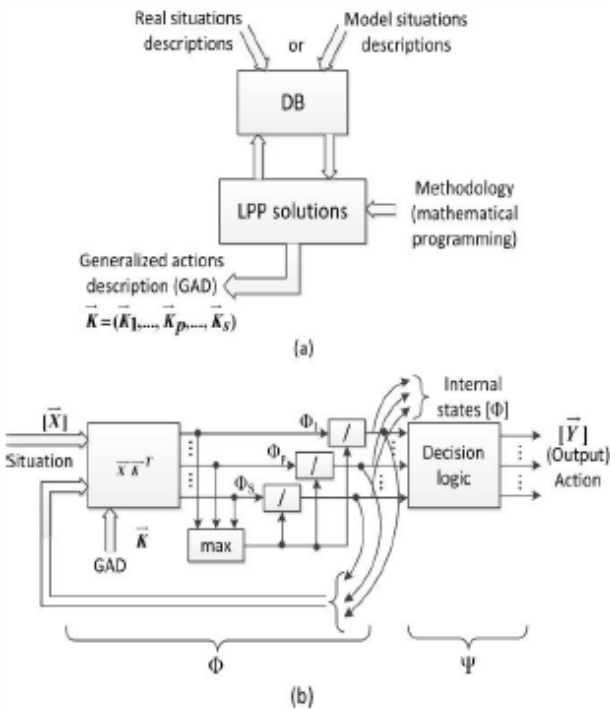


Figure 5. Functional Organization of the MAS Corresponding to the Type 1 Agents

*IF x is A AND y is B THEN z is C* (for Mamdani fuzzy models),

*IF x is A AND y is B THEN z=F(x,y)* (for Takagi-Sugeno fuzzy models).

Defuzzification procedures for the two cases mentioned above can be described as reasoning based on a set of consequents  $C$  using the CoG (center of gravity) or MoM (mean of maximum) methods for Mamdani type systems [14, 15], and MF (fuzzy mean) method as reasoning by evaluation of all results  $z$  included and processed according to the formula  $F(x,y)$  for Takagi-Sugeno systems.

A block-diagram of an ordinary fuzzy system corresponding to both cases is presented in Fig. 6.

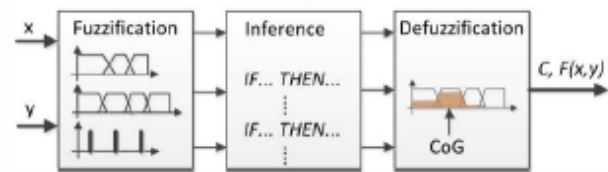


Figure 6. Ordinary Fuzzy System

As a matter of fact, the task of recognizing action's pattern, as it is noted in the introduction, belongs to the class of fuzzily described problems and requires a defuzzified answer.

Functional organization of an agent (or MAS) based on the fuzzy rules approach is delivered in Fig. 7(a) and 7(b).

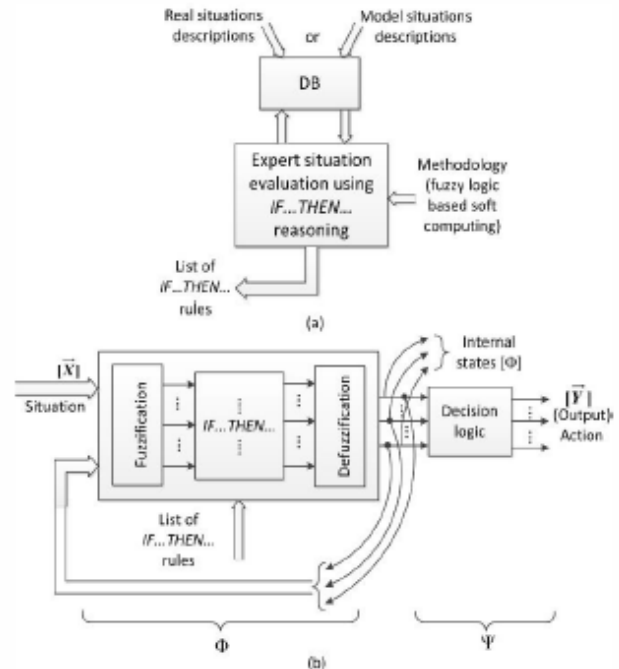


Figure 7. Functional Organization of the MAS Corresponding to the Type 2 Agents

Here, data from a DB are used to construct a list of fuzzy rules that serve as an inference engine for the transformation of inputs (situation's description) into internal states, as well as the transformation of its

internal states into outputs that determine environment's actions according to the generalized description of the agent or MAS (see Fig. 4).

### 4.3. MAS Intellectics Based On Prescribed Behavioral Tendencies

According to our approach, MAS is able to organize a successful intellectual behavior of the environment while possessing information only on a general tendency prescribed by experts or possible users. In such case, a decentralized adaptive control process aimed at intellectualizing our environment is involved [16, 17]. The backbone of this approach is seen in a space-time extension of the well-known stochastic approximation procedure combined with computerized fuzzy verbal and perceptual reasoning [18]. The idea of extending and using the stochastic approximation procedure is evoked by the fact that each system acts according to its inherent internal potential function  $V(K, X)$ . Sometimes, this function itself cannot be determined explicitly in terms of classic mathematics. Instead, we are able to measure or evaluate some decisively important characteristics of behavior  $Q(K, X)$  depending on the situation that occurs in the environment. Usually, the situation requires to minimize or maximize the averages of those characteristics  $\mathcal{M}\{Q(K, X)\}$  (here  $\mathcal{M}$  stands for mathematical expectation). When substituting unknown potential  $V$ -function for the available set of characteristics  $Q$ , it is convenient to use a certain additive or multiplicative function  $\mathfrak{R}\{*\}$  that permits to form only one function  $\mathfrak{R}\{Q(K, X)\}$ , for example- to be minimized stochastically around a local minimum:

$$\mathfrak{R}\{\mathcal{M}\{Q(K, X)\} \rightarrow \min. \quad (16)$$

The convergence of  $K$  towards a desirable state  $K_C$  that can determine the environment's action as its reaction to the current situation during the process of stochastic approximation can be performed according to the gradient procedure:

$$\frac{dK(\rho, t)}{d\rho} = \Gamma_\rho(\rho, t) \text{grad } \mathfrak{R}\{Q(K, X)\} \quad (17)$$

$$\frac{dK(\rho, t)}{dt} = \Gamma_t(\rho, t) \text{grad } \mathfrak{R}\{Q(K, X)\}. \quad (18)$$

This procedure can be performed by the algorithm shown in Figure 8.

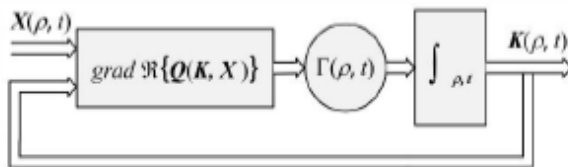


Figure 8. The Stochastic Approximation Performed According to the Space-Time Gradient Procedure

Actual implementation of all the operations of this algorithm is based on space-time integration procedures discussed and elaborated in [10, 12, 19].

It is well known [14, 18] that such a procedure converges only probabilistically:

$$P \left\{ \lim_{\rho, t \rightarrow \infty} [K(\rho, t) - K_C] = 0 \right\} = 1 \quad (19)$$

This convergence is shown in Fig. 9.

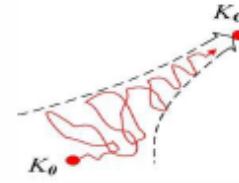


Figure 9. Probabilistic Convergence Process

The probabilistic convergence of the procedure is guaranteed under sufficient (but not necessary) requirements imposed on the functions of characteristics  $Q(K, X)$  and sequences of coefficients of proportionality  $\Gamma(\rho, t)$  [16, 17]:  $Q$  must vary slower than quadratic parabola, and coefficients  $\Gamma$  must decrease approximately according to the law  $\Gamma/n$ , where  $n$  is a number of iteration.

So, using these assumptions and procedures, MAS is able to adapt itself to the prescribed tendency of environment's behavior, and do so in an online fashion.

Corresponding functional organization of the agent (or MAS) based on the behavioral tendency prescription is shown in Fig. 10(a) and 10(b).

It is important to emphasize that the whole procedure of agent (or MAS) behavior converging towards a proper environmental action is realized in the online regime.

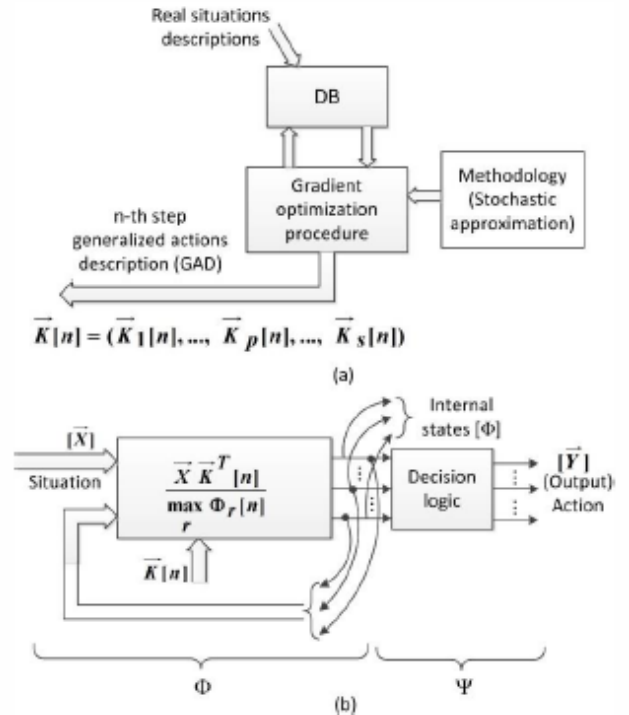


Figure 10. Functional Organization of MAS Corresponding to the Type 3 Agents

## 5. On Preliminary Application

This research was performed under the methodological philosophy developed according to the results of COST Action IC0702 “Combining Soft Computing Techniques and Statistical Methods to Improve Data Analysis Solutions (SOFTSTAT)”.

Different aspects of a concrete implementation and application of intellectics of the multi-agent systems (MAS) were used in two projects under support of the EU Structural Funds: 1) “Research and Development of Internet Infrastructure for IoT& S in the Smart Environment (IDAPI)” (project VP1-3.1-ŠMM-08-K-01-018) and 2) “Research on Smart Home Environment and Development of Intelligent Technologies (BIATech)” (project VP1-3.1-ŠMM-10-V-02-020). Both projects are supervised by the Ministry of Education and Science (MES) of the Republic of Lithuania.

The main research task specified in the framework of those projects is to develop an infrastructure, its functional organization, technologies and design methodology suitable for the implementation of the Internet of Things and Services (IoT&S) environment based on MAS intellectics for the comfort of users.

In both projects, the models of functional organization of MAS were developed according to the three approaches delivered and investigated in Sections 2-4 of this paper.

Computer simulation of the modeled environment actions was performed following the approach based on MAS intellectics. Said approach is delivered in Subsections 4.1-4.3. Simplified illustrative examples to help better understand the movement actions activated by intelligent agents in the smart home environment are delivered in this Section. A virtual environment was created to test the agent decision-making. It consists of a plane made of blocks. Dimensions of the plane are  $30 \times 30$  blocks (Fig. 11).

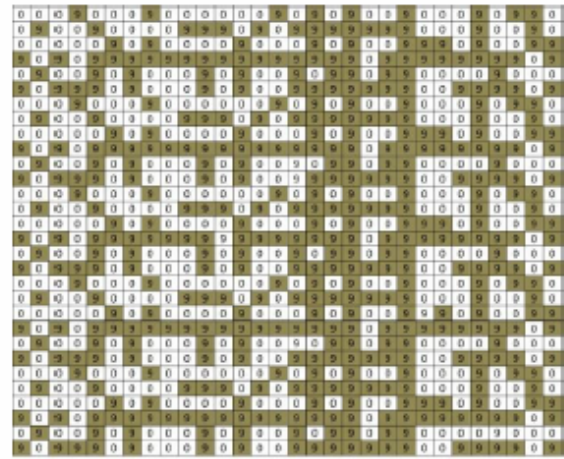


Figure 11. Generated Initial Plane

Each block has a property that we called a repulsion value. It ranges from 0 to 9. Block repulsion pattern was generated from the initial set of 20 block groups sized  $3 \times 3$  (Fig. 12).

The initial plane can be altered with noise. The noise has a uniform distribution in the interval  $[0-9]$ . At each given moment, the agent could sense the repulsion of  $N = 9$  blocks (square) with the agent in the center (circle) (Fig. 13). The higher the repulsion of the block, the more the agent “does not like” that block.



Figure 13. Environment of the Agent

There are  $S = 4$  actions the agent can make to affect the environment: it can walk by one block to the chosen direction (left, up, right, down). Agent decision-making process was performed in three phases:

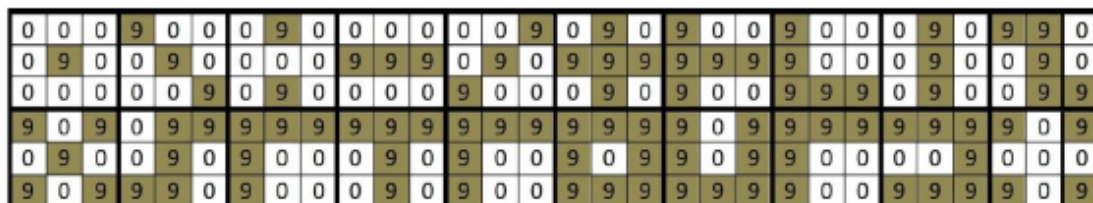


Figure 12. Initial Set of Twenty  $3 \times 3$  Blocks

Table 2. LPP solutions

$p$	$K_{p1}$	$K_{p2}$	$K_{p3}$	$K_{p4}$	$K_{p5}$	$K_{p6}$	$K_{p7}$	$K_{p8}$	$K_{p9}$	$\Phi_{pmax}$
1	9.00	6.80	2.20	0	5.30	9.00	0	4.50	7.10	50.62
2	0	0	1.00	4.00	2.00	7.00	9.00	9.00	4.00	90.00
3	9.00	5.50	0	9.00	1.20	0	7.40	3.10	0	123.26
4	6.90	7.30	7.60	9.00	3.10	9.00	0	0	3.8	109.04



1. Agent forms an opinion about the situation ( $\Phi$  transformation according to (3) is performed). The result of this phase is an opinion about each direction he could go to.
2. Agent forms intentions based on its opinion. In this phase, the agent considers if it has reached the border in order to eliminate the possibility of stepping over it. The result of this phase is a list of actions the agent wants to perform.
3. Agent makes a decision based on his intentions by following these rules:
  - If agent wants to perform one action – it decides to perform it;
  - If agent wants to perform two or three actions (if these are to move left and right or up and down at the same time), it performs a summarized action (even a diagonal movement) and so on. The second and the third phases together form a  $\Psi$  transformation according to (4).

Finally, after deciding what to do, the agent performs an action or actions and then arrives at a new situation. Then, the process is repeated.

```

RULEBLOCK blockUp
  AND : MIN;
  ACT : MIN;
  ACCU : MAX;
  RULE 0 : IF x0 IS low AND x3 IS low AND x6 IS low THEN LEFT IS high;
  RULE 1 : IF (x0 IS low AND x3 IS low AND x6 IS high) OR (x0 IS low AND x3 IS high AND x6
IS low) OR (x0 IS high AND x3 IS low AND x6 IS low) THEN LEFT IS mid;
  RULE 2 : IF (x0 IS low AND x3 IS high AND x6 IS high) OR (x0 IS high AND x3 IS low AND x6
IS high) OR (x0 IS high AND x3 IS high AND x6 IS low) THEN LEFT IS low;
  RULE 3 : IF x0 IS low AND x1 IS low AND x2 IS low THEN UP IS high;
  RULE 4 : IF (x0 IS low AND x1 IS low AND x2 IS high) OR (x0 IS low AND x1 IS high AND x2
IS low) OR (x0 IS high AND x1 IS low AND x2 IS low) THEN UP IS mid;
  RULE 5 : IF (x0 IS low AND x1 IS high AND x2 IS high) OR (x0 IS high AND x1 IS low AND x2
IS high) OR (x0 IS high AND x1 IS high AND x2 IS low) THEN UP IS low;
  RULE 6 : IF x2 IS low AND x5 IS low AND x8 IS low THEN RIGHT IS high;
  RULE 7 : IF (x2 IS low AND x5 IS low AND x8 IS high) OR (x2 IS low AND x5 IS high AND x8
IS low) OR (x2 IS high AND x5 IS low AND x8 IS low) THEN RIGHT IS mid;
  RULE 8 : IF (x2 IS low AND x5 IS high AND x8 IS high) OR (x2 IS high AND x5 IS low AND x8
IS high) OR (x2 IS high AND x5 IS high AND x8 IS low) THEN RIGHT IS low;
  RULE 9 : IF x6 IS low AND x7 IS low AND x8 IS low THEN DOWN IS high;
  RULE 10 : IF (x6 IS low AND x7 IS low AND x8 IS high) OR (x6 IS low AND x7 IS high AND x8
IS low) OR (x6 IS high AND x7 IS low AND x8 IS low) THEN DOWN IS mid;
  RULE 11 : IF (x6 IS low AND x7 IS high AND x8 IS high) OR (x6 IS high AND x7 IS low AND x8
IS high) OR (x6 IS high AND x7 IS high AND x8 IS low) THEN DOWN IS low;
END_RULEBLOCK

```

A method capable of achieving a prescribed behavioral tendency proposed in Subsection 4.3 and based on the stochastic approximation approach was used to form the intellectics of the type 3 agent. The behavioral tendency was constructed in the following form:

$$\mathfrak{R}\{-\mathcal{M}_1\{Q_1(K, X)\} + \mathcal{M}_2\{Q_2(K, X)\} - \mathcal{M}_3\{Q_3(K, X)\} - \mathcal{M}_4\{Q_4(K, X)\}\} \rightarrow \min. \quad (20)$$

It means that four partial behavior types were constructed for each action. Important characteristics

Three types of agents were implemented for each approach proposed in Section 4. First of all, the LPP approach was used to form the intellect. There were  $S = 4$  situation classes, one for each action. 12 situations from the initial movement plane were selected randomly as representatives for these classes: 3 for each of the classes, one being picked as “central” ( $k$ ) as it is stated in subsection 4.1. LPPs were constructed ((8) – (11) for each  $p$ ) with  $\gamma = 0.8$  and  $\kappa = 0.2$ .

Obtained solutions are shown in Table 2. The values of generalized patterns  $\vec{K}_p$  and  $\Phi_{pmax}$  were used for the type 1 agent.

Fuzzy rules according to Subsection 4.2 were used to form the intellectics of the type 2 agent. jFuzzyLogic library was used [20, 21]. First of all, linguistic variables were defined for input and output variables. There were 9 input variables for each block the agent sensed, and 4 output variables for each action.

Fuzzy rules were created to evaluate the situation. Because jFuzzyLogic library was used, the rules were defined by FCL [22]. The list of those 11 rules is presented below:

for each partial behavior were selected from the area that the agent sensed (Table 3):

Since agent learning (or self-training) is performed online, two implementations of opinion phase logic were made for the agent of this type. One was for learning, and the other was for an already known internal state  $K$  that has been formed when learning online.

All implemented agents (agents of three types) were put to different tests. Agents of the first and second type were tested by going through the following steps:

Table 3. Description of Partial Actions

Behavior	Mathematical representation	Visual presentation									
Left	$Q_1(\vec{K}, \vec{X}) = x_0k_0 + x_3k_3 + x_6k_6$	<table border="1"><tr><td>0</td><td>9</td><td>0</td></tr><tr><td>9</td><td>9</td><td>0</td></tr><tr><td>9</td><td>0</td><td>0</td></tr></table>	0	9	0	9	9	0	9	0	0
0	9	0									
9	9	0									
9	0	0									
Up	$Q_2(\vec{K}, \vec{X}) = x_0k_0 + x_1k_1 + x_2k_2$	<table border="1"><tr><td>0</td><td>9</td><td>0</td></tr><tr><td>9</td><td>9</td><td>0</td></tr><tr><td>9</td><td>0</td><td>0</td></tr></table>	0	9	0	9	9	0	9	0	0
0	9	0									
9	9	0									
9	0	0									
Right	$Q_3(\vec{K}, \vec{X}) = x_2k_2 + x_5k_5 + x_8k_8$	<table border="1"><tr><td>0</td><td>9</td><td>0</td></tr><tr><td>9</td><td>9</td><td>0</td></tr><tr><td>9</td><td>0</td><td>0</td></tr></table>	0	9	0	9	9	0	9	0	0
0	9	0									
9	9	0									
9	0	0									
Down	$Q_4(\vec{K}, \vec{X}) = x_6k_6 + x_7k_7 + x_8k_8$	<table border="1"><tr><td>0</td><td>9</td><td>0</td></tr><tr><td>9</td><td>9</td><td>0</td></tr><tr><td>9</td><td>0</td><td>0</td></tr></table>	0	9	0	9	9	0	9	0	0
0	9	0									
9	9	0									
9	0	0									

1. Agent was forced to perform an action sequence containing 100 actions in the initial environment. These action sequences were recorded.
2. Noise was applied and the agent performed another action sequence of the same length. This was repeated for every noise level. Each sequence was recorded.
2. Performance was evaluated by comparing the action sequence with the initial one in a noisy environment. The comparison of these actions has shown the reliability of agent training; it means that to some degree the agent is noise -proof.
4. First three steps were performed 10,000 times, with a random starting position of the agent each time. Performance results were summed up and divided by the run count. In this way, the average agent performance was calculated.

An example of one run of the type 1 agent is shown in Fig. 14(a).

Performance results show that even though the choices an agent makes degrade with higher noise level, it still makes similar choices in 30% of cases:

Noise	1	2	3	4	5	6	7	8	9
Result	38.7	35.6	32.0	31.0	30.4	30.0	29.6	29.2	27.8

Examples of the movement paths for one run taken by the type 2 agent, show that it constantly gets stuck in a movement loop between a few tiles of the environment (Fig. 14 (b)). This, however, does not mean that the agent is acting wrong. Its actions are merely consequences of a situation he got himself into, and for him they are always the right thing to do.

The performance of the type 2 agent is more or less the same as of the first one:

Noise	1	2	3	4	5	6	7	8	9
Result	35.3	34.2	34.9	32.5	30.6	29.8	26.3	28.0	28.2

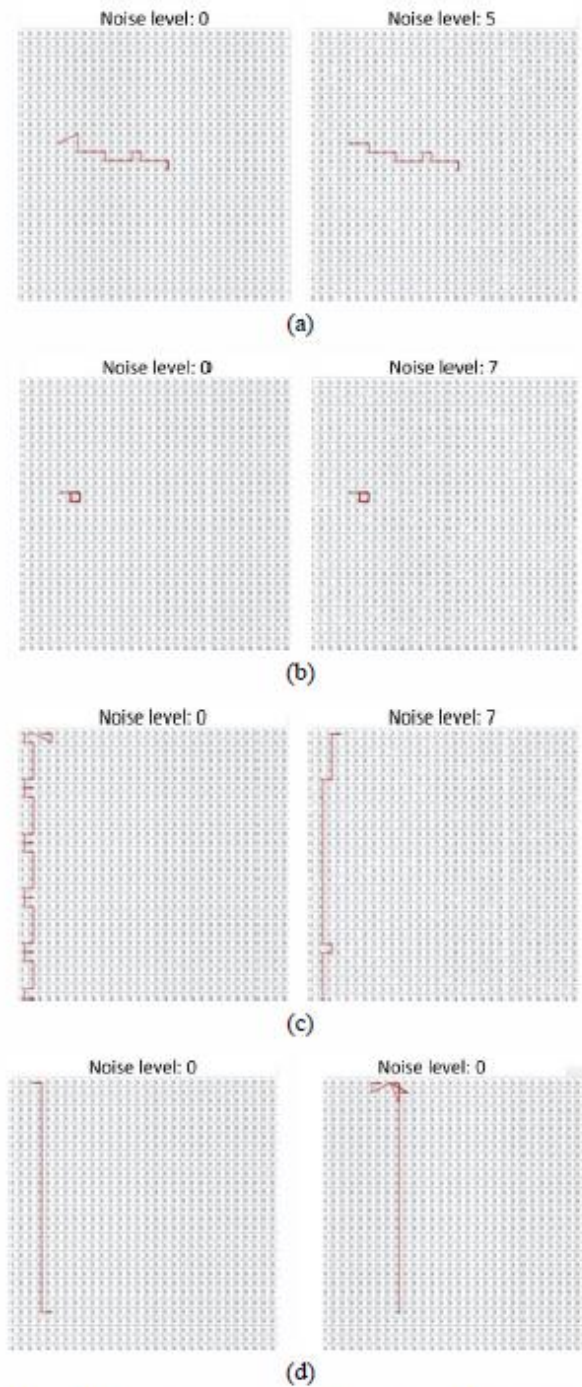


Figure 14. Examples of testing runs performed by agents of the type 1, 2 and 3 subjected to different noise level

A comparison of the performance of agents in a graphical form is presented in Fig. 15.

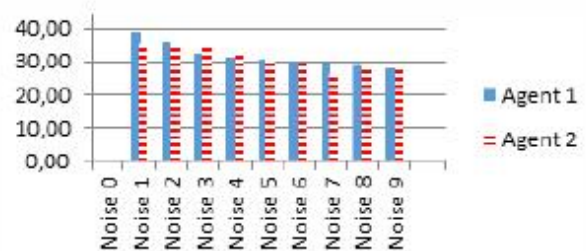


Figure 15. Performance comparison of the Agents

For the type 3 agent, the tendency to “move up” was formulated. The movement paths of the third agent (Fig. 14 (c)) show that it develops a tendency to move up successfully, but, after reaching the top, often gets stuck in two blocks.

5. After the online self-training was performed, the type 3 agent, now possessing a newly-gained knowledge, was put in random places on the initial field. The examples of its behavior after the self-training process (Fig. 14 (d)) show relatively good (or even excellent) performance results.

## 6. Final Remarks

After summarizing the theoretical and experimental research results, we can state that:

- the well-known and widely spread definitions of the smart and intelligent agent (SA; IA), as well as the smart and intelligent multi-agent system (SS/II\_MAS) were precisiated;
- the use of a unified and standardized agent/multi-agent system description based on the definitions of the general systems theory was delivered;
- three typical features of human intellectual activities were proposed to be implemented and simulated in the agent/multi-agent system as basic paradigms for the intellectics of agent and multi-agent systems; it must be underlined that operation according to those paradigms (recognition and classification, behavior according to a set of fuzzy rules, and operation according to some prescribed tendency) is solidly mathematically based (correspondingly: mathematical programming, fuzzy logic and stochastic approximation);
- results of computerized modeling and simulation have demonstrated the practical vitality and efficiency of the theoretical approach for the realization of an intelligent environment of IoT&S for user's comfort;
- a lot of technical and social problems still remain and need to be solved in order to successfully implement a user-friendly environment based on the intellectics of multi-agent systems; some of them are still being researched and will serve as authors' investments in further publications

## References

[1] M. Wooldridge. An Introduction to Multi-Agent Systems. John Wiley & Sons, Ltd., 2008.

[2] Y. Shoham, M. Tennenholtz. On the emergence of social conventions: modeling, analysis, and simulations. *Artificial Intelligence*, 1997, Vol. 94, No. 1-2, 139-166.

[3] N. R. Jennings, C. Sierra, L. Sonenberg, M. Tambe. In: *Proceedings of the Third International Joint Conference on Autonomous Agents and Multi Agent Systems*, 2004, N.Y. ACM Press.

[4] S. Russel, P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2003

[5] G. Weiss, S. Sen. *Adaptation and Learning in Multi-agent Systems*. Springer Verlag, Berlin, 1996.

[6] Y. Shoham, K. Leyton-Brown. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, 2009.

[7] J. Ferber. *Multi-agent systems; an introduction to distributed artificial intelligence*. Addison Wesley, 1999.

[8] B. Hayes-Roth, L. Brownston, R. van Gent. Multi-agent collaboration in direct improvisation. In: *Proceedings of the First International Conference on Multi-Agent Systems*, Menlo Park, Ca, AAAI Press, 1995, pp. 148-154.

[9] I. J. Dickinson. *Agent Standards. Living of the Vision* HP Laboratories. Bristol, HPL-97-156, 1997.

[10] G. Wunch. *Systemtheorie*. Akademische Verlagsgesellschaft Geest&Porting, K.-G. Leipzig, 1975.

[11] M. Guarracino, R. Jasinevicius, R. Krusinskiene, V. Petrauskas. Fuzzy Hyperinference-Based Pattern Recognition. In: *Towards Advanced Data Analysis by Combining Soft Computing and Statistics*, eds. C. Borgelt, M. A. Gil, J.M.C. Sousa, M. Verleysen. Springer, 2013, pp. 223-240.

[12] R. Jasinevicius. *Parallel Space-Time Computing Structures (in Russian)*. Moksas, Vibnius, 1988.

[13] R. Jasinevicius, V. Petrauskas. Fuzzy expert maps: the new approach. In: *WCCI 2008 Proceedings: 2008 IEEE World Congress on Computational Intelligence*, June 1-6, 2008, Hong Kong: 2008 IEEE International Conference on Fuzzy Systems. 2008 IEEE International Joint Conference on Neural Networks. 2008 IEEE Congress on Evolutionary Computation. Piscataway: IEEE, 2008. ISBN 978-1-4244-1819-0. pp. 1511-1517.

[14] A. Konar. *Computational Intelligence (Principles, Techniques and Applications)*. Springer, Berlin, 2005.

[15] B. Kosko. *Fuzzy Engineering*. Prentice Hall, 1997.

[16] M. A. Aizerman, E. M. Braverman, L. I. Rozonoer. Theoretical Foundations of the Potential Function Method in Pattern Recognition Learning. *Automation and Remote Control*, 1964, Vol. 25, 821-837.

[17] Ya. Z. Tsyplkin. *Adaptation and Learning in Automatic Systems*. Academic Press, 1971.

[18] V. S. Borkar. *Stochastic Approximation: A Dynamical Systems Viewpoint*. Cambridge University Press, 2008.

[19] R. Jasinevichius. Parallel space-time structure for computer vision systems. *Informatica*, 1992, Vol. 3, No. 3, 418-431.

[20] P. Cingolani, J. Alcalá-Fdez. jFuzzyLogic: a Java Library to Design Fuzzy Logic Controllers According to the Standard for Fuzzy Control Programming. *International Journal of Computational Intelligence Systems*, 2013, Vol. 6, Supplement 1, 61-75.

[21] P. Cingolani, J. Alcalá-Fdez J. jFuzzyLogic: A Robust and Flexible Fuzzy-Logic Inference System Language Implementation. *WCCI 2012 IEEE World Congress on Computational Intelligence*, June, 10-15, 2012 - Brisbane, Australia, pp. 1090-1097.

[22] INTERNATIONAL STANDARD IEC 61131-7. Programmable controllers – Part 7: Fuzzy control programming. International Electrotechnical Commission (IEC) 2000. [http://www.plcopen.org/pages/tc1\\_standards/iec\\_61131\\_7](http://www.plcopen.org/pages/tc1_standards/iec_61131_7).

Received November 2014.

## **5.2. priedas. Kompaktinė plokštelė**

Kompaktinės plokštelės turinys:

1. elektroninė šio darbo versija;
2. BIA modelio programinė įranga;
3. elektroninė straipsnio versija.