

Vilniaus universiteto  
Komunikacijos fakulteto  
Informacijos ir komunikacijos katedra

Lukas Ramašauskas  
Informacijos sistemų vadybos magistro studijų programos studentas

**DIRBTINIO INTELEKTO SISTEMŲ NAUDOJIMAS E. MOKYMOSE  
SISTEMOSE**

Magistro darbas

Vadovas doc. dr. Povilas Abarius

Vilnius, 2011

# MAGISTRO BAIGIAMOJO DARBO LYDRAŠTIS

<i>Pildo bakalauro / magistro baigiamojo darbo autorius</i>
_____
(bakalauro / magistro baigiamojo darbo autoriaus vardas, pavardė)
_____
(bakalauro / magistro baigiamojo darbo pavadinimas lietuvių kalba)
_____
(bakalauro / magistro baigiamojo darbo pavadinimas anglų kalba)
<b>Patvirtinu, kad bakalauro / magistro baigiamasis darbas parašytas savarankiškai, nepažeidžiant kitiems asmenims priklausančių autorių teisių, visas baigiamasis bakalauro / magistro darbas ar jo dalis nebuvo panaudoti kitose aukštosiose mokyklose.</b>
_____
(bakalauro / magistro baigiamojo darbo autoriaus parašas)
<b>Sutinku, kad bakalauro / magistro baigiamasis darbas būtų naudojamas neatlygintinai 5 metus Vilniaus universiteto Komunikacijos fakulteto studijų procese.</b>
_____
(bakalauro / magistro baigiamojo darbo autoriaus parašas)
<i>Pildo bakalauro / magistro baigiamojo darbo vadovas</i>
<b>Bakalauro / magistro baigiamąjį darbą ginti _____</b>
(įrašyti – leidžiu arba neleidžiu)
_____
(data) (bakalauro / magistro baigiamojo darbo vadovo parašas)
<i>Pildo instituto / katedros, kuriojančios studijų programą, reikalų tvarkytoja</i>
<b>Bakalauro / magistro baigiamasis darbas įregistruotas</b>
_____
(instituto / katedros, kuriojančios studijų programą, pavadinimas)
_____
(data) (instituto / katedros reikalų tvarkytojos parašas)
<i>Pildo instituto / katedros, kuriojančios studijų programą, vadovas</i>
<b>Recenzentu skiriu _____</b>
(recenzento vardas, pavardė)
_____
(data) (instituto / katedros vadovo parašas)
<i>Pildo recenzentas</i>
<b>Darbą recenzuoti gavau. _____</b>
(data) (recenzento parašas)

## REFERATO LAPAS

**Ramašauskas, Lukas**

Ra166 Dirbtinio intelekto sistemų naudojimas e. mokymosi sistemose: Magistro darbas / Lukas Ramašauskas; mokslinis vadovas Povilas Abarius; Vilniaus universitetas. Komunikacijos fakultetas. Informacijos ir komunikacijos katedra. – Vilnius, 2011. – 70 [1] lap. : lent. – Maš. inr. – santr. Angl. – Bibliogr.: lap. 66–69 (47 pavad.).

UDK indeksas 519.7 + 37.01 + 005

**Reikšminiai žodžiai:** e. mokymasis, dirbtinis intelektas, informacijos ir žinių vadyba, agentinės sistemos, agentų komunikacija, algoritmas, modelis, programa.

Magistrinio *darbo objektas* – dirbtinio intelekto sistemų naudojimas e. mokymosi sistemose. *Darbo tikslas* – ištirti dirbtinio intelekto sistemų naudojimo e. mokymosi sistemose galimybes ir optimalius taikymus. *Darbo uždaviniai:* atlikti nuotolinio mokymosi sistemų ir jų aplinkų mokslinę analizę informacijos ir žinių vadybos požiūriu bei sudaryti tose sistemose vykstančių informacijos ir žinių kaitos procesų modelį; ištirti algoritmų su dirbtinio intelekto įskiepiais panaudojimą e. mokymui ir juos naujai pritaikyti agentinėse e. mokymosi sistemose; sukurti vartotojui patogias taikomas programas, kurios emuliuotų agentų dirbtinio intelekto vykdomus skaičiavimus e. mokymosi sistemose ir apibendrinti gautuosius rezultatus.

Išanalizavus mokslinę literatūrą, taikant analogijų, palyginimų ir ekstrapoliacijos mokslinius metodus, prieita išvados, kad galima sudaryti e. mokymosi sistemose vykstančių informacijos ir žinių kaitos modelį bei jį realizuojančius algoritmus. Ištirtos naujos algoritmų su dirbtinio intelekto įskiepiais panaudojimo galimybės, taikant sukurtąjį konceptualų informacijos ir žinių kaitos procesų modelį e. mokymui, taip pat agentinėse e. mokymosi sistemose. Magistro darbe yra sukurtos taikomosios programos, kurios emuliuoja agentų dirbtinio intelekto vykdomus skaičiavimus e. mokymosi sistemose ir pagerina mokinio ir sistemos komunikaciją.

Magistro darbas gali būti naudingas mokslo ir švietimo institucijoms, informacijos mokslų, matematikos bei informatikos disciplinų dėstytojams ir studentams, susijusiems su e. mokymosi sistemų pritaikymu ir naudojimu.

# TURINYS

TURINYS .....	4
SANTRUMPŲ SĄRAŠAS .....	5
ĮVADAS .....	6
1. NUOTOLINIO, E. MOKYMO SI ANALIZĖ, PROCESŲ MODELIAI .....	8
1.1 Mokymosi proceso samprata .....	8
1.1.1 E. mokymosi koncepcija ir technologiniai reikalavimai .....	9
1.1.2 E. mokymosi modeliai .....	11
1.2 E. mokymosi aplinkos .....	11
1.2.1 VMA klasifikacija .....	14
1.2.2 Adaptyviosios e. mokymosi aplinkos .....	16
1.3 Žinių vadybos aspektai VMA sistemose .....	18
1.4 Informacijos vadybos aspektų analizė e. mokymosi sistemose .....	25
1.5 Apibendrinimas .....	33
2. DIRBTINIO INTELEKTO ALGORITMŲ E. MOKYMO SI SISTEMOSE TYRIMAS .....	34
2.1. E. mokymosi sistemų intelektualizacija ir modeliavimas .....	34
2.2 Dirbtinio intelekto agentų realizavimo algoritmai .....	37
2.2.1 Dirbtinio intelekto algoritmų skirstymas .....	37
2.2.1 Uždelsto apdovanojimo modelis .....	38
2.2.2 Markovo savybių integravimo metodai .....	39
2.3 Q-learning algoritmas ir jo analizė .....	40
2.3.1 Adaptuotas Q-learning algoritmas .....	40
2.3.2 Mokymo proceso planas .....	41
2.3.3 Mokymo proceso rezultatas .....	42
2.4 Parametro $r$ vertės įtaka e. patarėjo apmokymo procese .....	43
2.4.1 Parametro $r$ samprata .....	43
2.4.2 Skirtingų reikšmių parametro $r$ tyrimas .....	44
2.4.3 Parametro $r$ tyrimo rezultatų $Q(s, a)$ lentelių palyginimas .....	48
2.5 Apibendrinimas .....	50
3. PROBLEMINIŲ UŽDAVINIŲ SPRENDIMAS Q-LEARNING ALGORITMU .....	51
3.1. Statinės aplinkos modelio analizė ir taikymas .....	51
3.1.1 Algoritmo inicializacija .....	51
3.1.2 Duomenų apdorojimo metodas .....	54
3.2 Mišrios aplinkos modelio realizavimo programa .....	56
3.2.1 Skirtumų apžvalga ir realizavimas .....	56
3.2.2 Atsitiktinis strategijų generatorius .....	57
3.2.3 Programos kontrolė ir informatyvesnė vartotojo sąsaja .....	58
3.3 Sukurtų programų duomenų ir realizavimo analizė .....	58
3.3.1 Programos v100 rezultatų palyginimas .....	58
3.3.2 Eksperimento paklaidų įvertinimas .....	60
3.3.3 Programos v200 aprašymas .....	61
3.4 Apibendrinimas .....	62
IŠVADOS .....	64
SANTRAUKA ANGLŲ KALBA .....	65
LITERATŪROS SĄRAŠAS .....	66
PRIEDAI .....	70

## SANTRUMPŲ SĄRAŠAS

AHC	Klasifikavimo metodas (iš angl. <i>Agglomerative Hierarchical Clustering</i> ), panaudojantis grupuojamų objektų panašumo skirtumus. Metodo algoritmas sukuria dvejetainių grupių medį ir trunka tol, kol visi objektai bus sugrupuoti.
AIES	Adaptyvios intelektinės mokymosi sistemos (angl. <i>Adaptive and Intelligence Educational Systems</i> ).
ANN's	Dirbtinių neuronų tinklai (angl. <i>Artificial Neural Networks, ANN's</i> ), dirbtinio intelekto srities dalykas.
BIRCH	Hierarchinis duomenų gavybos algoritmas (angl. <i>Balanced Iterative Reducing and Clustering using Hierarchies</i> ).
CLIP3	Mašininio mokymo taisyklių klasifikavimo algoritmas (angl. <i>Cover Learning using Integer Programming, v.3</i> ).
CMS	Kursų tvarkymo sistema (angl. <i>Course Management System</i> ).
CURE	Didelių duomenų bazių klasterizacijos algoritmas (angl. <i>Clustering Using REpresentatives</i> ).
E. mokymas(is)	Elektroninis mokymas, elektroninis mokymasis.
HTML	Hiperteksto žymėjimo kalba (angl. <i>Hyper Text Markup Language</i> ).
IKT	Informacinės komunikacinės technologijos.
IS	Informacinė sistema.
IT	Informacinės technologijos.
ITERATE	Konceptualus duomenų gavybos klasterizacijos algoritmo pavadinimas
LMS	Mokymosi tvarkymo sistema (angl. <i>Learning Management System</i> ).
ML	Mašininis mokymas(is) (angl. <i>Machine Learning</i> )
PAC	Mokymosi iš pavyzdžių algoritmas (angl. <i>Probably Approximately Correct</i> ).
RL	Sustiprintas (skatinamasis) mokymas (angl. <i>Reinforcement Learning</i> ).
SARSA	Sustiprinto mokymosi (skatinamojo) algoritmas – prototipas
TSVM algoritmas	Tikslus aklosios steganoanalizės duomenų klasifikavimo algoritmas (angl. <i>Transductive Support Vector Machines Algorithm</i> ).
UML	Vieningoji modeliavimo kalba (angl. <i>Unified Modelling Language</i> ).
VMA	Virtuali mokymosi aplinka.
ŽV	Žinių vadyba.

# ĮVADAS

## **Darbo temos aktualumas ir naujumas**

Nuotolinio mokymosi koncepcija formavosi kartu su komunikacijų technologijų tobulėjimu ir siekiu taupyti žmogiškuosius išteklius. Ypač spartus elektroninio ryšio ir kompiuterių mokslo progresas XX a. pabaigoje leido panaudoti naujas, inovatyvias informacines technologijas (toliau tekste – IT) nuotoliniam mokymuisi: internetą, elektroninį paštą, telefoną, faksą, video skambučius, telekonferencijas ir t.t. Nuotolinio mokymosi sistemoms priskirtinas naujas e. mokymosi sistemų poaibis. E. mokymosi sistemos per paskutinįjį dešimtmetį intensyviai tobulinamos, vystomos jų funkcijos, taikant dirbtinio intelekto, duomenų gavybos, miglotosios (angl. *fuzzy*) ir patikimosios (angl. *robust*) logikos mokslų metodus. Jos turi geras pritaikymo perspektyvas visose edukacinėse, galbūt, ir mokslui priskiriamose srityse mokyklose, kolegijose, universitetuose. Diegiant modernias e. mokymosi sistemas aiškėja ir iššūkiai, sunkumai, susiję su žmogaus ir kompiuterio sąveikos, Web semantikos, edukologinėmis, psichologinėmis, bendravimo problemomis, kurias reikia spręsti fizinių (ypač informatikos), socialinių ir humanitarinių mokslų kontekste. Taigi, su e. mokymuisi ir su IT taikymais susietos mokslinių tyrimų temos yra aktualios dabar, o ateityje bus dar svarbesnės.

## **Tyrimo objektas**

Viena iš e. mokymosi atšakų yra mokymasis su paskatinimu, kurio ištakos siekia kibernetikos, statistikos, psichologijos, neurologijos ir informatikos mokslų sankirtas, grindžiamas intelektualių programinių agentų sąveika per bandymus ir klaidas dinamiškoje aplinkoje. Darbe tiriamos dirbtinio intelekto sistemų naudojimo e. mokymosi sistemose galimybės, naudojant teorinius analogijų ir palyginimų bei empirinius patikros metodus.

## **Darbo tikslas**

Ištirti dirbtinio intelekto sistemų naudojimo e. mokymosi sistemose galimybes ir optimalius taikymus.

## **Darbo uždaviniai**

1. Atlikti nuotolinio mokymosi sistemų ir jų aplinkų mokslinę analizę informacijos ir žinių vadybos požiūriu bei sudaryti tose sistemose vykstančių informacijos ir žinių kaitos procesų modelį.
2. Ištirti algoritmų su dirbtinio intelekto įskiepiams panaudojimą e. mokymuisi ir juos naujai pritaikyti agentinėse e. mokymosi sistemose.

3. Sukurti vartotojui patogias taikomąsias programas, kurios emuliuotų agentų dirbtinio intelekto vykdomus skaičiavimus e. mokymosi sistemose ir apibendrinti gautuosius rezultatus.

### **Darbo mokslinio tyrimo hipotezė**

Dirbtinio intelekto sistemų įskiepai e. mokymosi sistemose turėtų pagerinti tų sistemų veikimą, o naujai sukurti ir įdiegti programiniai agentai greičiau ir tiksliau emuliuoti žmogaus – patarėjo funkciją, sumažinti žmogiškojo faktoriaus klaidų skaičių sistemoje.

### **Tyrimo metodai**

Darbe taikomi įvairūs teoriniai ir empiriniai tyrimo metodai. Pirmojoje (analitinėje) darbo dalyje reikšmingesnę vietą užima kokybiniai žvalgomieji tyrimai, skirti idėjų generavimui ir gilesnei informacijos, kuri reikalinga uždavinio modeliavimui, paieškai mokslinėje literatūroje. Antrojoje ir trečiojoje dalyse dominuoja kiekybiniai priežastiniai tyrimai, kurių pagrindą sudaro skaitmeninis eksperimentas, atliekamas su sukurtuoju modeliu (algoritmais ir programomis), siekiant gauti platesnį duomenų spektrą ir detalesnę informacijos ir žinių kaitos procesų modelio įvertinimą.

### **Mokslinė ir praktinė darbo vertė**

Darbe aprašomi nauji e. mokymosi sistemų tyrimai ir teikiami intelektikos algoritmų taikymai, turintys praktinę mokslinę vertę bei realų praktinį pritaikymą, gerinant e. mokymo sistemos darbą švietimo ir mokymo įstaigose. Naudojant sukurtąsias programas taupomas sistemos naudotojų laikas, parenkant mokiniui optimalią medžiagos pateikimo formą.

### **Darbo struktūra**

Darbą sudaro antraštinis lapas, magistro darbo lydraštis, referato lapas, turinys, santrumpų sąrašas, įvadas, 3 pagrindiniai darbo skyriai, išvados, santrauka anglų kalba, bibliografinių nuorodų sąrašas, iš viso 70 psl. Priedas su elektronine darbo dokumentų ir programų laikmena – 1 vnt.

# 1. NUOTOLINIO, E. MOKYMOSI ANALIZĖ, PROCESŲ MODELIAI

## 1.1 Mokymosi proceso samprata

Nuo pat senovės viena iš žmonijos funkcijų yra sukaupto patyrimo, įgūdžių ir žinių perdavimas naujoms kartoms. Iš pradžių mokymosi procesas buvo realizuojamas tik praktinės veiklos metu. Vėliau kūrėsi pirmosios akademijos, mokyklos, kolegijos, universitetai, formuojantys nuoseklus mokymosi kultūrą. Tobulėjant komunikacijų technologijoms ir taupant žmoniškųjų išteklių resursus, XIX a. viduryje pradeda formotis nuotolinio mokymosi koncepcija, panaudojant paštą. Informacinių technologijų progresas leidžia panaudoti naujas inovatyvias informacines technologijas nuotoliniam mokymuisi: elektroninį paštą, telefoną, faksą, vaizdo skambučius, vaizdo konferencijas. (Rutkauskienė, 2003, 2007).

Mokymasis yra labai sudėtingas procesas. Žinomos trys pagrindinės mokymosi paradigmos: instrukcinė (angl. *instructural*), atskleidimo (angl. *revelatory*), numanymo (angl. *conjectural*) ir ketvirtoji, papildanti visas tris – išlaisvinimo (angl. *emancipatory*) paradigma, apimanti kompiuterines priemones, leidžiančias besimokančiajam atsikratyti nenaudingo rutininio darbo, tiesiogiai nesusijusio su mokymosi procesu (Denisovas, 2000).

Instrukcinė paradigma atspindi biheivoristinį požiūrį į mokymosi procesą, akcentuojantį dalyko turinio svarbą bei jo pateikimo tvarką, pavidalą, žinių testavimo techniką ir pan. Pagal šį požiūrį, mokyme ir mokymesi svarbiausi veiksniai yra mokymas, tai yra stimulus, ir mokymosi rezultatas, tai yra reakcija. Taigi, mokymusi (arba išmokimu) yra laikomas pasikeitęs individo išorinis elgesys, kurį galima sustiprinti apdovanojimais arba bausmėmis. Anot biheivoristų, studijuoti besimokančiojo mąstymą ir informacijos apdorojimą keblu, nes tokių dalykų negalima tiesiogiai stebėti. Programuoto mokymosi, pratybų, elektroninių vadovėlių ir įvairių testų programos priklauso šiai paradigmai.

Atskleidimo paradigma mokymosi subjektu laiko ne dalyko turinį, bet patį besimokantį. Jos pagrindinės nuostatos grindžiamos „mokymosi atrandant“ teorijomis, kur svarbiausia yra besimokančiojo intuicija. Virtualių laboratorijų, duomenų apdorojimo ir kompiuterinių modelių programos gerai iliustruoja šią paradigmą.

Numanymo paradigma apima kognityvinę pažinimo teoriją, probleminį mokymą ir akcentuoja hipotezių kėlimą, jų tikrinimą ir įgyvendinimą. Dalyko suvokimas, mokymasis traktuojamas kaip aktyvus žinių konstravimo procesas, labiau pabrėžiamas informacijos apdorojimas, o ne rezultatas. Programų pavyzdžiais čia galėtų būti ne atskiri jau sudaryti modeliai, bet modeliavimo sistemos,



leidžiančios besimokančiajam pačiam realizuoti jo idėjas, konstruojant kompiuterinius modelius, modifikuojant ir testuojant juos, organizuojant su jais įvairius kompiuterinius eksperimentus.

Pasak Denisovo (2000), naudojant informacines technologijas ir ženkliai išaugus mokymosi poreikiui, savaime pakito ir mokymosi procesas bei mokymo metodai. Galima išskirti keletą esminių bruožų, kurie įtakoja mokymosi metodų ir pačios sistemos kitimą pastaraisiais dešimtmečiais (Lipeikienė, 2003):

- augantis mokinių skaičius;
- augantis neakivaizdinių ir pavienių mokinių klasių skaičius;
- mokinių, mokančių už mokslą, reikalavimas lanksčios mokymosi sistemos;
- išaugęs mokinių kompiuterinis raštingumas;
- kompiuterių tinklų atsiradimas;
- ypač greitai besikeičiantis dėstomų dalykų turinys;
- lėtas ir brangus knygų leidybos procesas.

Lipeikienės pateiktus argumentus galima papildyti ir A. Haapala išskiriamu požymiu, kad kiekvienas mokinys turi savitą, jam priimtina, mokymosi būdą – stilių. Haapala (2006) remiasi Felder'io (1993) suformuluotais penkiais klausimais, kurių atsakymai leidžia nusakyti konkretaus mokinio mokymosi stilių.

1. Kokią informaciją mokinys priima ir supranta geriausiai.
2. Kokiais kanalais mokinys geriausiai priima informaciją.
3. Koks informacijos pateikimo būdas mokiniui priimtinausias.
4. Kaip mokinys linkęs apdoroti gautą informaciją.
5. Kaip mokinys pasiekia pateiktos informacijos esmę (Felder, 1993).

Šiomis dienomis ypač didelę įtaką turi technologinis aspektas, kuris yra beveik neišvengiamas mokymosi procese. Kompiuterinės technologijos įgalina mokiniui geriau įsisavinti pateikiamą medžiagą (pagal poreikį pateikiamą garsiniu, video ar tekstiniu pavidalu), ją peržiūrėti keletą kartų ar trumpam atitrūkti nuo mokymosi proceso, padarant reikiamą pertrauką, ir taip efektyviau įsisavinant žinias bei planuojant mokymosi laiką. Galime teigti, kad informacinės technologijos yra esminis „kaltininkas“, kuris standartinį mokymosi proceso modelį papildė elektroninio mokymosi modelio savybėmis, taip sukurdamas naują mokymosi koncepciją.

### 1.1.1 E. mokymosi koncepcija ir technologiniai reikalavimai

Terminai „nuotolinis švietimas“ bei „nuotolinis mokymasis“ suprantami kaip tam tikra mokymosi forma, besiskirianti nuo tradicinių mokymosi būdų. Neatsiejamas šios mokymosi formos atributas yra šiuolaikinės informacinės komunikacinės technologijos (toliau tekste – IKT).

*Nuotolinis mokymas(is) – tai bet koks mokymas(is), kai mokymo paslaugas dažniausiai valdo pats besimokantis asmuo, o paslaugos teikėjas jam suteikia paramą ir pagalbą* (Rutkauskienė, 2007).

Mokymui, naudojant tokią mokymosi formą, neribojamas žmonių amžius ar socialinė padėtis. Studijuojant nuotoliniu būdu galima įgyti išsilavinimą, kelti profesinę kvalifikaciją bei užsiimti savivugda (Rutkauskienė, 2003). Dabar nuotolinio mokymosi samprata yra neatsiejama nuo elektroninio mokymosi (toliau tekste – e. mokymasis) koncepcijos dėka informacinės visuomenės formavimosi ir informacinių technologijų raidos. Toks mokymasis apibrėžia naujas žinių gavimo galimybes ir yra suprantamas kaip informacijos gavimo, kaupimo ir perdavimo būdas (Ričkutė, 2005).

E. mokymas šiomis, IT progreso dienomis, yra labai plati sąvoka, kuri apima įvairias mokymosi metodikas ir pagalbinius įrankius (internetą, kompaktinius diskus, video konferencijas ir pan.). E. mokymasis turi akivaizdų pranašumą, ypač panaudojant multimedijos priemones prieš tiesioginį mokytojo ir mokinio dialogą (Tankelevičienė, 2009). Mokinys gali pakartotinai susipažinti su mokomąja medžiaga, nereikalaudamas papildomo laiko resurso iš mokytojo. Taipogi mokinys gali individualiai studijuoti pateiktą medžiagą ir bandyti ją gauti pagal jam priimtina formatą – stilių.

Dauguma e. mokymosi sistemų turi įgyvendintą tinklo pagrindu grįstą mokymosi metodą – technologiją. Tinklo technologija (dažniausiai sutinkama kaip Web 2.0) įgalina e. mokymosi koncepciją išplėsti iki nuotolinio mokymosi koncepcijos, kadangi Web 2.0 naudojami įrankiai (tokie kaip internetas, intranetas, nutolę duomenų serveriai, wiki, telekonferencijų įranga) leidžia sėkmingai vykdyti mokymosi procesą, kai mokinys ir mokytojas nėra prisijungę vienu metu.

Galime daryti prielaidą, kad IKT dėka e. mokymosi procesui vykdyti reikia mažiau fizinių išteklių, nei standartiniui mokymosi procesui. D. Rutkauskienė kaip tik ir pabrėžia esminius technologinius reikalavimus. Norint studijuoti nuotoliniu būdu, reikia turėti kompiuterinę įrangą ir internetinį ryšį (Rutkauskienė, 2003). Kokybiškam e. mokymosi proceso vykdymui yra keliami ne tik išoriniai technologiniai, bet ir vidiniai kokybiniai reikalavimai visai mokymosi medžiagai.

P. Abarius šių dienų e. mokymosi medžiagai išskiria šiuos būdingus bruožus, kurie ir lemia nuotolinio mokymosi sistemos išbaigtumą ir profesionalumą:

- aukštas interaktyvumo lygis;
- geras imitacijų ir animacijų lygis;
- geras vaizdo, garso, ir kitų daugialypės terpės priemonių lygis;
- integruota su tradiciniu mokymu ar pateikta pakankama palaikymo medžiaga;
- bendravimo priemonių (su mokytojais ir kolegomis) buvimas;
- Web technologija grįsta valdymo sistema (virtuali mokymosi terpė) (Abarius, 2010).

Be esminių technologinių aspektų nuotolinio mokymosi sistemoms realizuoti yra ne ką mažiau svarbus informacijos ir žinių valdymo aparatas. Siekiant užtikrinti, kad bet kokia sistema ar organizacija veiktų profesionaliai ir gerai atliktų savo funkcijas, ji turi pati gebėti priimti, apdoroti, pateikti ir saugoti gautąją informaciją (Ruževičius, 2007).

### 1.1.2 E. mokymosi modeliai

E. mokymosi modeliai yra skirti du pagrindinius tipus: sinchroninį ir asinchroninį.

*Sinchroninis modelis* – toks mokymosi būdas, kai užsiėmimai vyksta pagal grafiką, mokytojai ir mokiniai palaiko ryšį techninėmis priemonėmis: telefono, radijo, televizijos arba kompiuterinio ryšio kanalais, o taip pat elektroniniu paštu, pokalbiais. Šis modelis naudojamas sukurtose kompiuterių programose, kurios vadinamos virtualiomis mokymosi aplinkomis (toliau tekste – VMA) (Rutkauskienė, 2003, 2007). Taikant pastarąjį modelį yra nesunku organizuoti mokinių grupinio darbo užsiėmimus, kadangi mokytojas gali perteikti tiek vaizdinę, tiek garsinę informaciją mokiniams, o mokiniai gali komunikuoti tarpusavyje.

*Asinchroninis modelis* – tai toks būdas, kai mokiniui perduodama paruošta studijų medžiaga, kontrolinės užduotys ir nustatomas atsiskaitymų grafikas (Rutkauskienė, 2003). Kada ir kaip mokytis, paliekama mokinio nuožiūrai. Šiame modulyje gana sunku organizuoti mokinių grupinį darbą, kadangi nėra tiesioginio ir nuolatinio bendravimo tarp komunikuojančių pusių. Viena iš asinchroninio mokymo modelio rūšių yra sudėtingosios mokymo sistemos, kurios dar dažnai vadinamos adaptiviomis mokymosi sistemomis (Gubaidulin, 2007).

Šiuo metu efektyviausiu yra laikomas sinchroninio ir asinchroninio modelių kombinacijos mokymo modelis. Jis dar vadinamas mišriuojamu modeliu. Pastarąjį modelį įprasmina atvejis, kai paskaitos skaitomos, grupinis darbas ir diskusijos vyksta sinchroniškai, pasitelkus informacines technologijas, o mokinio individualus mokymasis vyksta asinchroniniu būdu, pateikus jam medžiagą individualiam darbui ir individualias užduotis (Rutkauskienė, 2003).

Nuotolinio švietimo sinchroninio, asinchroninio ir mišriojo modelių principų taikymas su informacinėmis technologijomis suartino tradicinių ir nuotolinių studijų galimybes bei sudarė sąlygas pagerinti tradicinių studijų efektyvumą bei kurti e. mokymosi sistemas.

### 1.2 E. mokymosi aplinkos

Informacinės technologijos didina švietimo galimybes, kurios gali turėti dvejopą vaidmenį:

1. Palaiko ir išplečia esamų mokymosi metodų galimybes.
2. Leidžia sukurti naujas mokymosi metodikas.

Tai gali būti virtualios mokymosi aplinkos arba adaptyvios mokymosi sistemos, turinčios hiperteksto, programavimo kalbų bei interneto galimybes (Rutkauskienė, 2003).

Tam tikros technologinės priemonės, tokios kaip elektroninis paštas, nereikalauja, kad mokiniai dirbtų vienu metu, o naudojant elektroninius pokalbius jie privalės būti bent prisijungę prie mokymosi sistemos.

Mokymosi medžiagai pateikti gerai tinka interneto teikiamos galimybės. Hipertekstas leidžia mokymo medžiagą pateikti pakankamai vaizdžiai: panaudojant grafikus, paveikslus, garso ir vaizdo įrašus. Šias galimybes apima e. mokymosi sistemos, kurios dar vadinamos virtualiomis mokymosi aplinkomis. Jos taipogi suteikia sistemos vartotojui galimybes tvarkyti kursų turinį ir medžiagą bei gauti ir pasirinkti priimtinausią pateikimo formą.

Virtuali mokymosi aplinka – tai sąvoka, kuri sudaryta iš trijų žodžių: virtualus, mokymas, aplinka. Žodis „virtualus“ yra kildinamas iš lotynų kalbos (lot. *virtualis*), kuris paprastai apibūdina numanomą, tariamą reiškinį, imituojantį tikrovę (pvz.: kompiuterio ekrane vaizduojami objektai). Tradiciškai mokymas suprantamas kaip mokinio ir mokytojo tiesioginis bendravimas dirbant toje pačioje aplinkoje, dalyviams matant ir girdint vienas kitą. Atsiradus nuotoliniam mokymui, atsirado ir virtualumo sąvoka, kai mokytojas ir mokinys bendrauja internetu. Žodis „aplinka“ suprantamas įvairiai, tačiau šiuo atveju įvardijami visi mokymosi procesai, kurie vyksta virtualioje erdvėje (Balbieris, 2005).

Paprastas ir bendrinis VMA apibrėžimas yra formuluojamas Dillenbourg:

*Virtuali mokymosi aplinka – tai visapusiška programinė įranga kompiuterių tinklo teikiamam mokymosi procesui valdyti.* (Dillenbourg, 2000).

VMA leidžia naudoti įvairius mokymosi metodus, panašiai, kaip ir realioji mokymosi aplinka (pvz.: klasė, auditorija), tačiau VMA siekiama padėti mokiniams mokytis, stebėti mokymosi procesą ir tobulinti jo turinį. Taigi, VMA galėtume apibūdinti kaip mokomosios medžiagos, užduočių, bendravimo ir vertinimo priemonių sistemą, leidžiančią lanksčiai valdyti ugdymo procesą.

Išskiriamos šios bendriausios virtualiųjų mokymosi aplinkų funkcijos (1 lentelė):

1 lentelė. Bendriausios VMA funkcijos (Balbieris, 2005; Brusilovsky, 2004)

<b>Funkcija</b>	<b>Technologija / realizacija</b>
Bendravimas	Elektroninis paštas, diskusijų forumai, pokalbiai, garso – vaizdo konferencijos
Bendradarbiavimas	Elektroninis paštas, mokinių grupių kūrimo ir valdymo priemonės, grupinio darbo priemonės
Vartotojų registracija	Mokinių, mokytojų, administratorių, stebėtojų kūrimas ir valdymas virtualioje mokymosi aplinkoje
Ugdymo turinio tvarkymas	Kompiuterinių priemonių komplektas
Užduočių vykdymas	Kompiuterinės programos užduočių vykdymui
Stebėjimas	Mokinių ir kitų sistemos vartotojų veiksmų ir pažangumo stebėjimo įrankiai
Sąsajos keitimas	Lanksti programinė įranga su sąsajos personalizavimo funkcija
Informacijos teikimas apie VMA	Žinynų (angl. <i>Help</i> ) įdiegimas

Virtualiosios mokymosi aplinkos viena nuo kitos skiriasi savo funkcijomis. Vienos jų gali turėti daugiau priemonių ugdymo procesui organizuoti, kitos – mažiau. Remiantis P. Dillbourg, G. Balbieriaus ir P. Brusilovsky analizuota medžiaga, galima susidaryti bendrą vaizdą, kas sudaro ar turi sudaryti virtualiąją mokymosi aplinką (2 lentelė):

2 lentelė. Virtualios mokymosi aplinkos priemonės ir jų pavyzdžiai

<b>VMA priemonė</b>	<b>Priemonių pavyzdžiai</b>
Bendravimo ir bendradarbiavimo priemonės	Sinchroninės priemonės – vaizdo konferencijos, pokalbiai tinkle, skelbimų lenta; Asinchroninės priemonės – elektroninis paštas, elektroniniai žurnalai, diskusijų forumai
Vartotojų (mokinių, mokytojų) prisistatymo sritis – priemonė	Informacinis (reprezentacinis) profailas su pagrindiniais vartotojo duomenimis
Vartotojų registracija	Administraciniai įrankiai, skirti vartotojų sukūrimui ir valdymui, kuriuos valdo sistemos administratoriai
Ugdymo turinio tvarkymo priemonės	Mokymosi tvarkymo sistema (angl. <i>Learning Management System</i> , toliau tekste – LMS), kursų tvarkymo sistema (angl. <i>Course Management System</i> , toliau tekste – CMS) ir kitos.
Užduočių rengimo ir apklausos organizavimo priemonės	Tai posistemės, kurios yra atsakingos už užduočių kūrimo ir vykdymo automatizavimą, kuris palengvina mokytojų darbą

Mokinių mokymosi ir pažangos stebėjimo ir vertinimo priemonės	Automatizuotos vertinimo sistemos ir foninės (angl. <i>background</i> ) posistemės, kurios stebi vartotojų elgseną.
Aplinkos sąsajos keitimo priemonės	Papildomos techninės sistemos priemonės, leidžiančios personalizuoti kiekvieno vartotojo sesiją, pagal jo pageidavimus (fono spalvos keitimas, turinio ir valdymo pulto išdėstymo tvarka ekrane ir pan.).

Pastebime, kad išvardintos komponentės yra atitinkamas VMA funkcijų objektas. Bendras komponentių vaizdas galėtų būti toks (1 paveikslas).

1 paveikslas. Virtualios mokymosi aplinkos sudėtinės dalys



### 1.2.1 VMA klasifikacija

Terminas *virtualioji mokymosi aplinka* bendrąja prasme vartojamas norint įvardyti bet kurią iš toliau išvardytų sistemų ar jų derinį.

**Mokymosi tvarkymo sistema.** Tai gali būti ir paprasta mokinių ir mokytojų registravimo sistema, leidžianti prieiti prie mokymosi medžiagos kompiuterių tinkle, ir sudėtinga sistema, stebinti mokinių mokymosi eigą ir pagal tai atliekanti kitas ugdymo proceso organizavimo funkcijas (Balbieris, 2005). Dauguma LMS tipo sistemų yra pagrįstos tinklinėmis technologijomis, kadangi jos leidžia naudoti ir tvarkyti medžiagą bet koku laiku, nepriklausomai nuo vietos ir naudojimosi trukmės.

**Mokymosi turinio tvarkymo sistema** (angl. *Learning Content Management System, LCMS*). Tai individualiems poreikiams pritaikoma mokymosi tvarkymo priemonė. Pavyzdžiui, ji gali atlikti mokymosi medžiagos dalių, kurias mokinys naudojo, stebėjimą, sekti besimokančiojo atsiskaitymus bei pagal tai parinkti ar pritaikyti atitinkamą ugdymo turinį (Balbieris, 2005).

Šio tipo sistemos dar gali būti vadinamos individualizuoto e. patarėjo sistemomis.

**Kursų tvarkymo sistema.** Ši priemonė leidžia mokytojui, nenaudojant hiperteksto žymėjimo kalbos (toliau tekste – HTML) arba kitos programavimo kalbos, parengti mokymosi kursą ir kompiuterių tinkle pateikti mokomąją medžiagą bei kitą su mokymusi susijusią informaciją. Vienos iš žinomiausių šio tipo sistemų yra WebCT ir BlackBoard (Meerts, 2003). Tokios sistemos veikia panašiu į duomenų saugyklos darbo principu, kai serveryje yra talpinama informacija.

**Kolektyvinio kompiuterizuoto mokymosi aplinka** (angl. *Computer Supported Collaborative Learning Environment*). Tai skirta konstruoti žinias ir mokytis kognityviniais metodais bei dirbant grupėmis sistema, kurioje susipynę psichologijos, informatikos, edukologijos mokslai.

**Turinio tvarkymo aplinka** (angl. *Content Management System*). Tai programinė įranga, kuri yra naudojama mokymosi medžiagai tvarkyti. Mokymosi medžiaga gali būti ir elektroniniai failai, nuotraukos, garso bei vaizdo dokumentai. Ji suteikia galimybes mokytojui lanksčiai tvarkyti mokomąją medžiagą: kurti naujus modulius, paimiti informaciją iš kitų šaltinių, pertvarkyti, pateikti ją vairiais būdais (Balbieris, 2005).

Išvardytų komponentų tarpusavio ryšiai pateikiami 2 paveiksle.

2 paveikslas. Virtualios mokymosi aplinkos komponentų diagrama



Virtualias mokymosi aplinkas galima klasifikuoti ne tik pagal santykį tarp vartotojo ir sistemos, bet ir pagal veikimo lokalizaciją:

1. *Lokali VMA*. Tai aplinka, kuria naudojama tik tuomet, kai kompiuteryje yra įdiegta specialios programinės įranga. Bendrauti arba bendradarbiauti galima tik su tais vartotojais, kurie yra prisijungę prie to paties tinklo bei kurių kompiuteriuose yra įdiegta tokia pati virtualioji mokymosi aplinka.

2. *Internetinė VMA*. Jai nereikia jokios specialios programinės įrangos, galima naudotis bet kuriuo kompiuteriu, kuriame įdiegta interneto naršyklė ir standartiniai grafinio vaizdavimo paketai. Pastaruoju metu šios aplinkos yra labai populiaros, iš jų – komercinės: EduCMS, BlackBoard, WebCT; atvirojo kodo – Moodle.

E. mokymosi sistemos pagal sudėtingumą gali būti suskirstytos į paprastas ir sudėtingas. Anksčiau aptarti modeliai yra priskiriami paprastųjų tipui, o sudėtingųjų tipui yra priskiriamos adaptyviosios e. mokymosi aplinkos.

### 1.2.2 Adaptyviosios e. mokymosi aplinkos

Per pastaruosius du dešimtmečius labai išpopuliarėjo adaptyviųjų e. mokymosi aplinkų mokslinis tyrinėjimas ir kūrimas, kadangi sinchroninio mokymosi sistemos tapo neefektyvios.

Adaptyviosios e. mokymosi sistemos (angl. *Adaptive Intelligent Educational System*, toliau tekste – AIES) dar dažnai yra vadinamos intelektualiosiomis, kadangi šių aplinkų gebėjimą prisitaikyti prie mokinio – mokytojo nulemia naudojami dirbtinio intelekto metodai. Terminas „adaptyvus“ (angl. *adaptation, adaptive*) yra siejamas su įvairiomis sistemų galimybėmis ir charakteristikomis, kurių savybė yra pasiūlyti individualų mokymosi planą kiekvienam vartotojui, sudaryti vartotojo žinių lygį atitinkančią mokymosi programą, nusakyti temas, kurios turėtų būti išstudijuotos kituose mokinio mokymosi žingsniuose, analizuoti vartotojo sprendimus (VMA požiūriu). Sistema turėtų gebėti identifikuoti mokymosi medžiagą, kuri turėtų būti pakartota, jei sprendimas buvo klaidingas. Taip pat gebėti pateikti tas mokymosi medžiagos dalis, kurios tinka pagal mokymosi tikslą bei kitas sistemos žinias apie vartotoją, kurios saugomos sistemos mokinio modulyje. Sistema taip pat gali priimti naujus vartotojus, kuriuos burtų į grupes, skirtas įvairioms problemoms spręsti. Taip galėtų būti apibūdinamas pilnas adaptyvių priemonių rinkinys (Paramythis, 2003).

Literatūroje yra minima dar viena adaptyvios virtualios mokymosi sistemos samprata. Pagal Paramythis tai yra tokios sistemos, kurios gali „*stebėti savo vartotojų veiklą; interpretuoti pagrindinius ir specialius sistemos modelius; nustatyti vartotojo reikalavimus ir prioritetus iš gautų veiklų rezultatus; atitinkamai pateikti rezultatus į asocijuotus modelius; ir galutinis svarbus reikalavimas, veikti prie turimų, prieinamų vartotojo žinių tam, kad palengvinti besimokančiojo mokymosi procesą*“ (Paramythis, 2003).

#### 1.2.2.1 Adaptyvių intelektinių mokymosi sistemų moduliai

AIES yra sudarytos iš keturių skirtingų komponentų:

1. *Mokinio modulio*, kuris yra orientuotas į mokinių skirstymą klasėmis pagal jų gebėjimus, galimybes įsisavinti žinias ar kitus parametrus, kurie apsprendžia mokinio aplinką. Mokinio modulyje būna integruota ir mokinio veiksmų stebėjimo sistema (Iglesias, 2003; Virvou, 2004);
2. *Domeno modulio*, kuris yra atsakingas už mokymosi turinio laikymą, klasifikavimą ir struktūrizavimą. Dažniausiai sutinkama medžio tipo hierarchinė įrašų saugojimo struktūra (Iglesias, 2003). E. mokymosi sistemos, kuriose daugiausia išplėtotas domeno modulis, gali būti



papildytos informacija apie darbo eigą, įvykius, vartotojus, jų vaidmenis ir pan. (Njike, 2005);

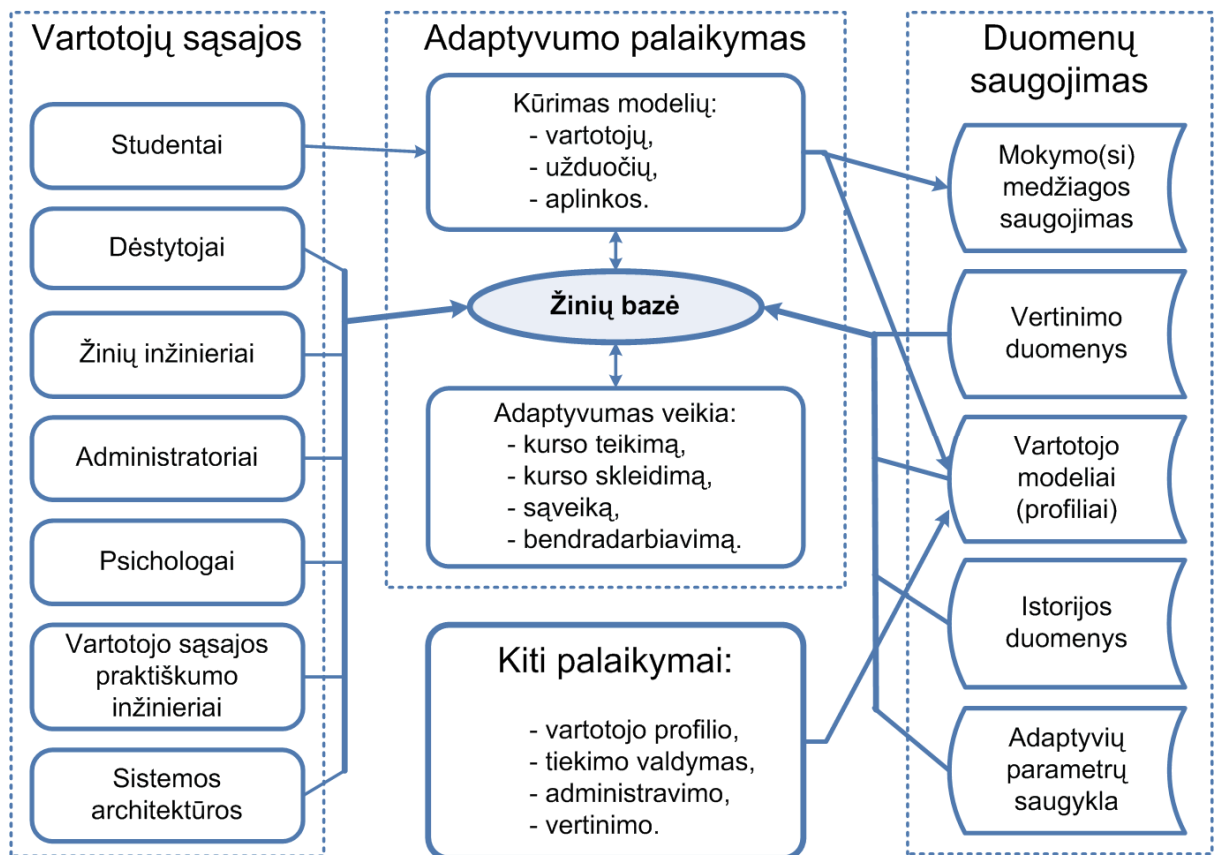
3. *Sąsajos modulis*, kuris leidžia AIES parinkti optimalų informacijos ir žinių perteikimo pavidalą mokiniui;
4. *Pedagoginio modulio*, kuris yra atsakingas už optimalaus mokymosi proceso strategijos radimą ir jos taikymą mokiniui. Šis modulis dažniausiai realizuojamas kaip elektroninis patarėjas. Pastarasis modulis taip pat yra naudojamas proceso apibūdinimui ir nustatymui, ar mokiniai mokosi panašiai, ar skirtingai (Iglesias, 2003; Galeev, 2003).

Remiantis šiais AIES formuojančiais moduliais, galime išvelgti esminį žinių vadybos poreikį tokiose e. mokymosi sistemose. Tai patvirtina ir P. Brusilovsky: „*Žinių panaudojimas apie tam tikrą sritį, mokinį ir mokymo procesą yra vienas iš pagrindinių intelektualiuju mokymo sistemos tikslų, kuriant individualizuotą mokymosi aplinką*“ (Brusilovsky, 1999).

### 1.2.2.2. Adaptyvių intelektinių mokymosi sistemų bendroji architektūra

Bendroji adaptyvios virtualios mokymosi sistemos architektūra pagal E. Vasilyevą pavaizduota 3 paveiksle.

3 paveikslas. AIES architektūra pagal E. Vasilyeva



### 1.3 Žinių vadybos aspektai VMA sistemose

Terminas žinių vadyba (angl. *knowlege management*, toliau tekste – ŽV) naudojamas apibūdinant naujų technologijų taikomąsias programas, kuriose įdiegti įrankiai, palaikantys organizacijos intelektualų turtą. Tai nevienakryptė disciplina, turinti įvairių panaudojimo sričių (Berry, 2006).

Remiantis Ubon (2002) apibūdinimu, ŽV yra susijusi su organizacijos vertės eksploatacijos siekiais ir žinios gali būti sudarytos iš aiškių arba dokumentuotų, neaiškių bei subjektyvių žinių. Vadyba sukelia visus šiuos procesus, susijusius su žinių identifikacija, padalinimu ir kūrimu. Tai reikalauja technologinių įgyvendinimų (žinių saugyklos) sukūrimo, palaikymo ir žinių apdorojimo procesų (Ubon, 2002).

Apibendrinant žinių vadybos sąvoką galime teigti, kad tai yra vadybos procesas, kuris kontroliuoja žinių kūrimą, platinimą ir panaudojimą, apjungiant technologijas, organizacijos struktūras bei žmones efektyvesniam mokymuisi.

Virtualiose mokymosi sistemose yra apdorojami dideli kiekiai skirtingų duomenų, informacijos ir žinių. Siekiant efektyvaus sistemos darbo tai būtina analizuoti.

Realybėje esantys daiktai yra susieti su esybėmis. Įrašai yra esybių atributai, kurie atstovauja duomenis. Informacija – tai jau tam tikra metodika apdoroti duomenys (pavyzdžiui – surikiuoti). Žinios yra informacijos apdorojimo veiksmų rezultatas, atsakantis į klausimą „kaip“. Adaptyviosiose mokymosi aplinkose išsiplečia požiūris į duomenis, metaduomenis, informaciją, žinias bei metažinias (Vasilyeva, 2005). Lyginamojo metodo rezultatus matome 3 lentelėje.

3 lentelė. Įprastosios ir adaptyviosios VMA naudojamų objektų palyginimai (Ubon, 2002; Vasilyeva, 2005)

	<b>Įprastosios VMA</b>	<b>Adaptyviosios VMA</b>
<b>Duomenys</b>	Pagrindinis esybių atributas	Mokinio profilis, jo charakteristikos ir sistemos vartotojo veiksmų istorija
<b>Informacija</b>	Sistemos vartotojai, mokinio įvertinimų rezultatai	Modelio struktūra
<b>Žinios</b>	Mokymosi medžiaga sistemoje	Mokymosi medžiaga kaip informacija
<b>Metažinios</b>	Raktiniai žodžiai	Mokymosi medžiaga su tam tikromis charakteristikomis

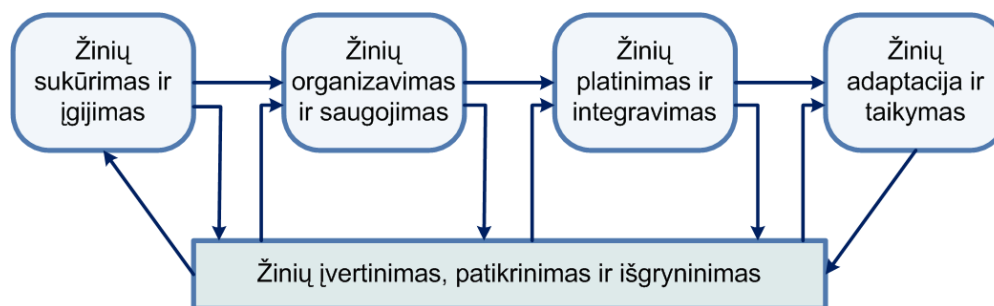
Remiantis E. Vasilyeva tyrimais, galima išskirti penkis esminius žinių vadybos procesus, kurie yra susiję su AIES:

1. Žinių sukūrimas ir įgyjimas (angl. *knowledge creation and acquisition*).
2. Žinių saugojimas ir organizavimas (angl. *knowledge storage and organisation*).

3. Žinių platinimas ir integravimas (angl. *knowledge distribution and integration*).
4. Žinių adaptacija ir taikymas (angl. *knowledge adaptation and application*).
5. Žinių vertinimas, patikrinimas ir išgryninimas (angl. *knowledge evaluation, validation and refinement*).

Pastarieji procesai yra tarpusavyje priklausomi ir juos galima pavaizduoti taip, kaip parodyta 4 paveiksle.

4 paveikslas. Žinių vadybos procesai pagal Vasilyeva (2005)



Galime pastebėti, kad adaptuoti žinių vadybos procesai yra neatsiejami komponentai virtualios mokymosi aplinkos moduluose, kadangi žinios ir metažinios įgalina AIES veikti efektyviai.

Efektyviam AIES darbui yra keliami trys funkciniai reikalavimai:

1. Plano sudarymas.
2. Mokinio mokymas.
3. Mokinio kontrolė (Brusilovsky, 1999).

Remiantis D. Baziukaitės dėstomomis mintimis, plano sudarymą galime apibūdinti kaip procesą, kurio metu sistema parenka atitinkamus mokymosi kursus, vertinimo kriterijus, informacijos pateikimo būdus, kuriuos mokinys ir mokytojas gali keisti ir adaptuoti pagal specialius savo poreikius. Mokymosi proceso metu sistema atlieka esminius agentinius skaičiavimus. Po jų sistema mokiniui pateikia atitinkamą mokymosi kelią pagal mokinio gebėjimus (pavyzdžiui, mokiniui užduodami papildomi klausimai, pakartotinis egzamino laikymas, pateikiama detalesnė informacija dalomojoje medžiagoje ir pan.).

Mokinio kontrolės funkcija yra panaši į mokymosi proceso esybę. Kontrolės funkcija iš dalies ją papildo. Kontrolės procesas yra atsakingas už mokinio vertinimo veiksmus: tarpinių ir galutinių kontrolinių klausimų pateikimą, aktyvumo stebėjimą ir kitus faktorius, kurie turi didelės įtakos mokymosi procesui (Baziukaitė, 2007). Išsamesnė šių dalių analizė pateikiama D. Baziukaitės disertacijoje.

Taikant įvairius matematinius modelius sistema gali pati parinkti mokiniui „patogiausią“ pagal jo poreikius mokymosi aplinką, pateikti atitinkamu formatu informaciją ar parinkti optimalią mokymosi strategiją. Sinchroninio mokymosi aplinkoje šiuos uždavinius turėtų spręsti „gyvas“ mo-

kytojas, kuriam tektų nelengvas darbas apdoroti kiekvieno mokinio specialius poreikius bei gebėjimus. Asinchroninio mokymosi sistemose šį darbą atlieka dirbtinio intelekto agentai (Baziukaitė, 2007).

Tam, kad sistema galėtų vykdyti pagrindines tris funkcijas (plano sudarymas, mokymas, mokinio kontrolė), reikia nustatyti mokinio žinių lygį. Tai yra viena iš sudėtingiausių procedūrų e. mokymosi sistemose. Egzistuoja trys bendri principai, kuriais galima bandyti nustatyti mokinio žinių lygį:

1. Mokinio savęs paties įsivertinimas (mokinys nusistato savo žinių lygį pagal sistemos pateikiamus jo mokymosi rezultatus ir mokymosi istoriją).
2. Mokytojo ar mokytojų atsiliepimai (tai mokytojų parinkti kursai, pagal kurių rezultatus mokytojai gali spręsti apie mokinio žinių lygį).
3. Automatizuotas egzaminavimas (kai pati sistema parenka kursus ir pagal mokinio rezultatus nustato jo žinių lygį).

Žinių lygio nustatymas yra vienas iš svarbiausių mokymosi proceso etapų, kadangi rezultatai gali būti naudojami iteratyviai: gilinant žinias silpnose vietose arba tobulinant mokymosi turinį. Mokinio žinių lygį galima nustatyti įvairiais metodais, kuriuos galime suskirstyti į automatizuoto ir neautomatizuoto (4 lentelė) tikrinimo tipus:

4 lentelė. Mokinio žinių lygio nustatymo metodų pavyzdžiai

<b>Automatizuoto tikrinimo metodai</b>	<b>Neautomatizuoto tikrinimo metodai</b>
Patikrinimo testas (angl. <i>Quiz</i> ), kai pateikiami pasirinkimo variantai	Rašinys, projektas
Patikrinimo testas, kai reikalaujama tikslaus rašytinio atsakymo (pavyzdžiui, aritmetinių veiksmų sekos rezultatas ar tam tikra anglų kalbos veiksmažodžio gramatinė forma)	Problemos aiškinimo testas

Neautomatizuoto tikrinimo metodai yra parankūs socialinės pakraipos mokslams, kadangi intelektinės sistemos dar nepajėgia pakankamai efektyviai analizuoti ir interpretuoti rašytinių šaltinių. Naudojant pastaruosius metodus galima patikrinti mokinio kūrybiškumą, kurį gali įvertinti tik mokytojas, o ne sistema.

Automatizuoto tikrinimo metodai yra labiausiai parankūs tikslųjų mokslų mokymams, kadangi tiksluosiuose moksluose „nėra vietos“ interpretacijoms ir reikalingas tikslus bei konkretus atsakymas.

Yra išskiriami penki metodai, kuriuos naudojant gaunamas tam tikras mokinio žinių lygio įvertinimas. Tai yra:

1. Teisingo / neteisingo atsakymo metodas.
2. Klaidų ieškojimo metodas.
3. Teisingų atsakymų ieškojimo metodas.
4. Kombinuotas metodas.
5. Statistinis metodas.

*Teisingo / neteisingo* atsakymo metodas kiekvieną klausimą laiko atsakytu arba ne. Sistema paprasčiausiai suskaičiuoja gerai atsakytus klausimus. Tuomet sistema, suskaičiavusi tik teisingai atsakytų klausimų skaičių, gali pateikti tikslų mokinio žinių lygį. Šis metodas teisingiausiai pateikia mokinio žinių lygį, kai yra užduodama pakankamai daug klausimų. Jei pateiksime mokiniui tik vieną klausimą ir jis bus blogai atsakytas, tuomet jis bus įvertintas pačiu blogiausiu įvertinimu.

*Klaidų ieškojimo* metodas susumuoja maksimaliai didžiausią galimų surinkti taškų kiekį ir po kiekvieno blogo atsakymo atima tam tikrą balų skaičių. Tai nėra geras metodas automatinėms įvertinimo sistemoms, kadangi mokinys, visiškai neatsakinėdamas į klausimus, gaus aukščiausią įvertinimą, nes mokinys paprasčiausiai taip ir nepadarys nė vienos klaidos.

*Teisingų atsakymų ieškojimo* metodas yra priešingas klaidų ieškojimo metodui. Šis pradžioje turi minimalų (dažniausiai 0) balų skaičių. Atsakius į klausimą teisingai yra sumuojami balai. Testams, kur pateikiami klausimai su keletu pasirinkimo variantų, automatizuotoms sistemoms tai nėra tinkamas metodas, nes mokinys gali paprasčiausiai pažymėti visus atsakymus ir jis gaus maksimalų įvertinimą.

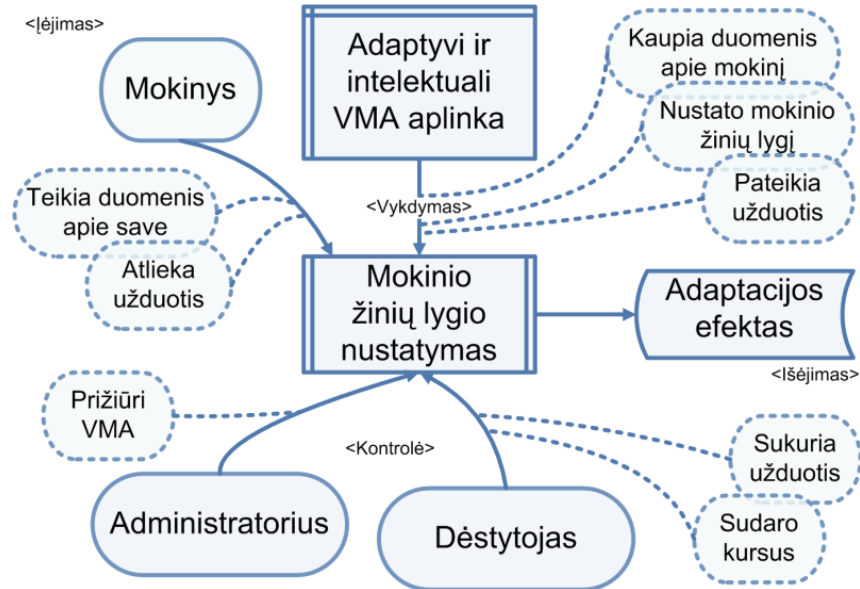
*Kombinuotas* modelis pirmiausia gražina minimalų balų skaičių, o vėliau už kiekvieną teisingą atsakymą prideda tam tikrą balų skaičių, o už neteisingą – nuima. Tai yra *klaidų ieškojimo* ir *teisingų atsakymų ieškojimo* mišinys. Tai tinkamas metodas automatinėms vertinimo sistemoms. Šis metodas neleidžia mokiniui surinkti visų įmanomų balų, pažymint visus atsakymų variantus.

*Statistinis* modelis – tai modifikuotas *kombinuotasis* modelis, kuris skaičiuoja balus pagal proporciją. Paimamas visų teisingų atsakymų skaičius ir visų neteisingų atsakymų skaičius ir kiekvienas įvertinamas balu, proporcingu visų galimų atsakymų skaičiui (Bota, 2000).

Galime daryti prielaidą, kad efektyviausiai nustatyti mokinio žinių lygį galime pasitelkę statistinį arba kombinuotąjį modelius. Savaiame suprantama, kad bet kokia AIES, turinti žinių nustatymo sistemą, turi ne tik nustatyti mokinio žinių lygį, bet ir jį išsaugoti tam tikroje aplinkos vietoje. Dažniausiai tai yra atliekama mokinio modulyje, kaip mokinio aplinkos personalizacijos dalis. Informacinių sistemų kūrimą galima pavaizduoti grafiškai, kaip adaptuotos e. mokymosi sistemos mokinio žinių nustatymo procesą (komponentes ir jų sąveikas) bei vieningos modeliavimo kalbos

(angl. *Unified Modelling Language*, toliau tekste - UML) standartu pavaizduoti vykstančius procesus (5 paveikslas).

5 paveikslas. Bendra mokinio žinių nustatymo sistema

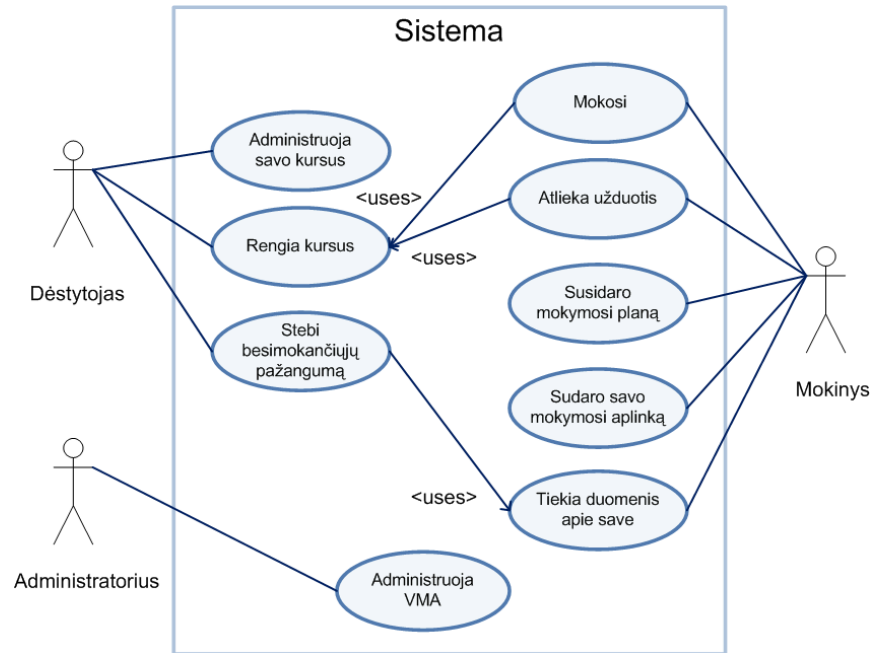


Iš 5 paveikslo mes matome pagrindinius veikėjus, kurie įtakoja sistemą savo veiksmiais:

- mokinys: teikia pradinis duomenis („metaduomenis“) apie save ir atlikdamas užduotis atnaujina savo profailą sistemoje;
- administratorius: prižiūri VMA sisteminių funkcionalumą, taip užtikrindamas sistemos kokybišką darbą;
- mokytojas (dėstytojas, mentorius): sudaro mokymosi kursus mokiniui ir, sukurdamas užduotis, paruošia informacinę bazę mokinio duomenims kaupti.

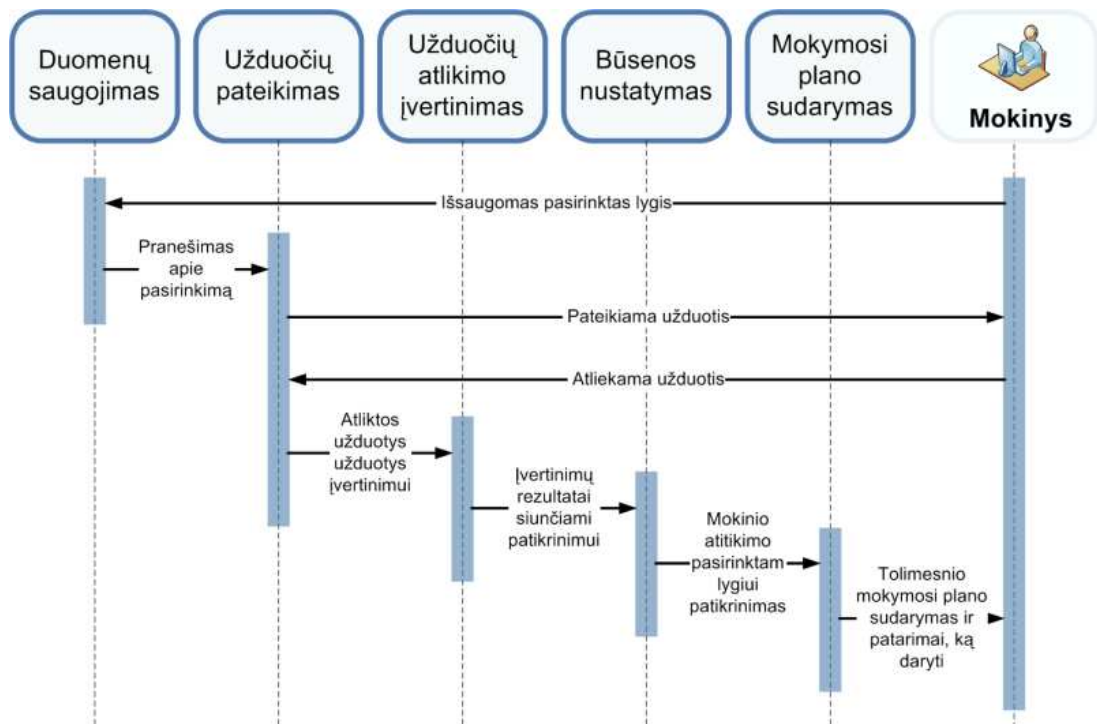
Žmones, kurie veikia sistemoje, UML diagramų pavyzdžiu galima įvardyti aktoriais. Aktorių ir sistemos ryšius galime pavaizduoti UML diagrama (6 paveikslas). Joje vidiniams funkciniam ryšiams pabrėžti palikta UML žymė <uses>.

6 paveikslas. AIES sistemos galima UML diagrama



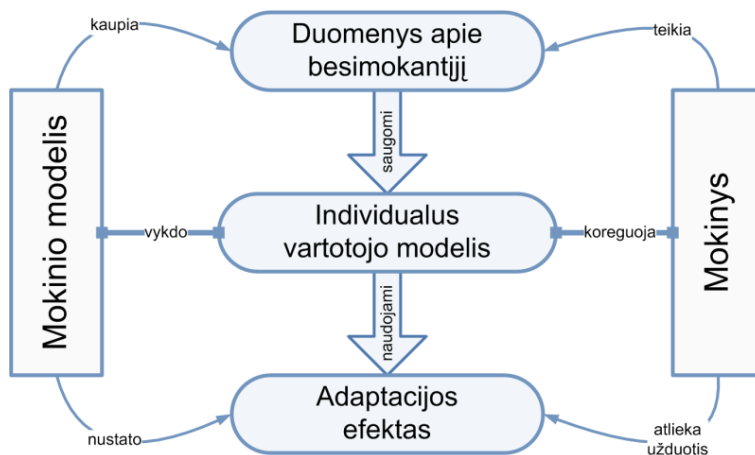
UML diagramoje yra matomi išplėstiniai mokinio veiksmai, kurie įtakoja sistemą – sudaro mokymosi planą ir sudaro savo mokymosi aplinką. Tokios sistemos pranašumas, lyginant su kitomis, yra tas, kad pats mokinys gali pasirinkti ar teikti pageidavimą, kad jam būtų sukurta personalizuota mokymosi aplinka bei parinktas individualus mokymosi planas. Mokinio žinių lygio nustatymo procesus galime pavaizduoti UML sekų diagramomis (7 paveikslas).

7 paveikslas. AIES sistemos UML procesų diagrama



Tokios VMA naudą galima išvelgti ir iš mokytojo pusės, kadangi kokybiškai sukūrus ir tinkamai administruojant mokymosi kursą, mokytojui nebereikia „budėti“ prie sistemos ir stebėti mokinio progresą, mokymosi procesą. Jis tai gali padaryti jam patogiu metu ir patogioje vietoje. Mokytojui tenka vienintelė prievolė – administruoti paruoštus kursus ir juos pateikti atitinkamiems mokiniams. Apibendrinant mokinio žinių nustatymo procesą, jį būtų galima pavaizduoti grafiškai, kur pateikiami mokinio, sistemos, duomenų ir jų tarpusavio ryšiai (8 paveikslas).

8 paveikslas. Mokinio žinių lygio nustatymo apibendrintas vaizdas.



Glaustai apžvelgę adaptuotos mokymosi aplinkos funkcinius reikalavimus, galime išskirti tokių sistemų trūkumus ir privalumus (5 lentelė).

5 lentelė. VMA sistemų trūkumų ir privalumų palyginimas

Privalumai	Trūkumai
Kiekvienas mokinys gali turėti personalizuotą mokymosi planą pagal savo žinias ir pageidavimus	Daugiausia laiko atimantis faktorius yra tas, kad mokytojai turi kruopščiai parengti medžiagą ir ją kokybiškai patalpinti sistemoje
E. patarėjo egzistavimas sistemoje palengvina mokytojų darbą ir suteikia greitesnį grįžtamąjį ryšį mokiniui iš sistemos	Nekokybiškai įdiegtas parinktas mokinio žinių nustatymo metodas gali sumenkinti pačios AIES darbo efektyvumą, nes atsiranda galimybė mokiniams sukčiauti (angl. <i>Workaround</i> ).
Tiek mokytojas, tiek mokinys yra nepriklausomas nuo darbo vietos (lokalizacijos) ir socialinių faktorių (išsilavinimo lygio, amžiaus, socialinės padėties).	

Pasak D. Baziukaitės, tai vyksta todėl, kad adaptyvių mokymosi sistemų naudojimas distanciniame mokymesi yra naudingas, nes suteikia besimokančiajam virtualią galimybę dirbti su savo „individualiu mokytoju“ (Baziukaitė, 2003). Tokia mokymo technika yra vadinama sustiprinto mo-



kymosi (angl. *Reinforcement Learning*, toliau tekste – RL), kai dirbtinio intelekto sistema yra pajėgi rasti sistemos optimalią veiksmų tvarką, paremtą tikrai sistemos patirtimi (Iglesias, 2002).

Nagrinėjant informacijos ir žinių vadybos aspektus AIES sistemose, yra pastebima ne tik žinių vadybos elementų, bet randama nemažai AIES sistemų panašumų su informacijos procesų modeliais, kuriuos detalizuosime sekančiuose skyreliuose.

#### 1.4 Informacijos vadybos aspektų analizė e. mokymosi sistemose

L. Markevičiūtė pastebi, kad informacijos vadybos, kaip savarankiško reiškinių formavimasis, prasidėjo apie 1970 – 1980 metus. Galime daryti prielaidą, kad informacinės technologijos ir jų spartus vystimasis taipogi turėjo didelės įtakos kaip ir žinių bei mokslo poreikis to meto visuomenėje (Markevičiūtė, 2008). Informacijos ir žinių vadyba yra suvokiama kaip informacijos ir žinių gavimo metodų, išteklių ir įrašų vadyba. Dažnai informacijos ir žinių vadybos sąvoka yra papildoma informacijos turinio, procesų, kontrolės, sklaidos ir organizavimo metodų sąvokomis (Raudeliūnienė, 2010; Elskytė, 2010).

Susipažinę su VMA struktūra ir atliekamomis komponentų funkcijomis, galime išvelgti informacijos ir žinių vadybos bruožų. Remiantis Z. Atkočiūnienės pateikta informacijos ir žinių vadybos lyginamąja lentele (6 lentelė), galime teigti, kad nuotolinio mokymosi sistemos, turinčios savyje intelektualizuotų agentų, yra linkusios veikti žinių vadybos lygmenyje (metodais) – tai parodė ankstesnis eksperimentas, o paprastosios sistemos (gebančios apdoroti informaciją ir priimti elementarius sprendimus) tik informacijos vadybos metodais.

6 lentelė. Informacijos ir žinių lyginamoji lentelė (Atkočiūnienė, 2006; Raudys, 2008)

<b>Informacija</b>	<b>Žinios</b>
Apdoroti duomenys	Į veiksmą nukreipta informacija
Dažniausiai pateikia faktus	Leidžia daryti prognozes, asociacijas ar pranašiškus sprendimus
Aiški, glausta, struktūruota ir paprasta	Painios, neapibrėžtos, iš dalies nestruktūruotos
Formalizuota (užfiksuota ir išreikšta, gali būti suteikta daugkartinio naudojimo forma)	Susidaro žmonių galvose įgyjant patirties
Išgaunama iš duomenų (formalizuota duomenų bazėse, knygoje, dokumentuose).	Formuojasi kolektyvinės žinios (nuolat kaupiant patirtį, apibendrinant sėkmę ir klaidas, mokantis)

Agentą, kaip sistemos dalį, mes suprantame, kaip tam tikrą programinės įrangos neatsiejamą komponentą, kuris gali įtakoti sistemos darbą, priimdamas tam tikrus sprendimus. Sprendimai gali būti paprasti ir sudėtingi. Paprastiems sprendimams apibrėžti yra pasitelkiami elementarūs sąlyginio

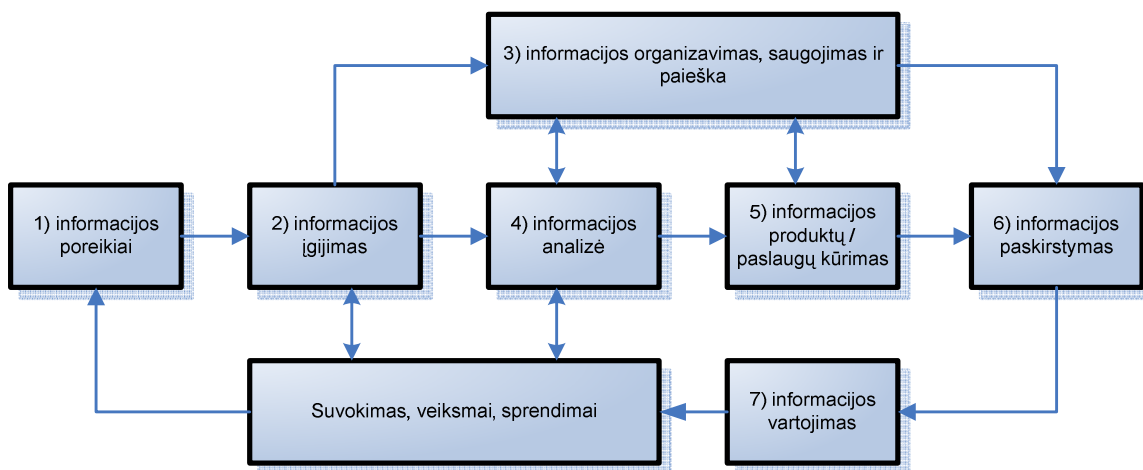
perėjimo (pvz.: „jeigu – tai“, „kol – tol“) metodai. Sudėtingiems sprendimams priimti yra suprogramuojami užduočių algoritmai, kurie savyje turi ypač sudėtingas matematinės išraiškas su netiesioginėmis priklausomybėmis. Sudėtingų sprendimų agentai yra vadinami dirbtinio intelekto agentais (Ramašauskas, 2010).

AIES agentai yra į veiksmą nukreipti objektai, kurie priima sprendimą pagal iš aplinkos gautą informaciją ir/ar žinias, sukurtas kitų agentų. Agentai, dirbantys su dideliais žinių ir informacijos kiekiais, aukštame intelektualizacijos lygmenyje (neteisinių matematinių algoritmų realizacija), susiduria su informacijos ekonomikos problema. Tokie agentai turi gebėti atskirti vertingą nuo bevertės, kokybišką ir nekokybišką informaciją. Kokybiškos informacijos naudojimas ir generavimas leidžia efektyviau paskirstyti sistemos darbą, o tai užtikrina ir bendro kuriamo produkto (mokslo) kokybę (Ruževičius, 2007). Galime teigti, kad nuotolinio mokymosi sistemos turi atsižvelgti ne tik į informacijos ir žinių vadybos metodiką, bet ir atlikti informacijos ekonomikos analizę.

Remiantis informacijos ir žinių vadybos procesų pamatiniu modeliu, galime sukurti nuotolinio mokymosi sistemos informacijos vadybos modelį. E. mokymosi sistemai L. Markevičiūtės apžvelgtas informacijos procesų modelis yra labai detalus. Analizuojant ir modeliuojant informacinę sistemą (toliau tekste – IS), skirtą nuotoliniam mokymuisi, galime sujungti panašius elementus į vieną, kadangi pastarųjų atliekama funkcija yra panaši ir iš dalies dubliuojasi realizavimo požiūriu.

L. Markevičiūtė savo straipsnyje (Markevičiūtė, 2008) remiasi kitų mokslininkų sudarytu informacijos procesų modeliu (9 paveikslas), kurio pagrindu galima kurti ir pamatinį nuotolinio mokymosi, ir dirbtinio intelekto sistemų informacijos procesų sąveikos modelį.

9 paveikslas Informacijos procesų modelis (Markevičiūtė, 2008)

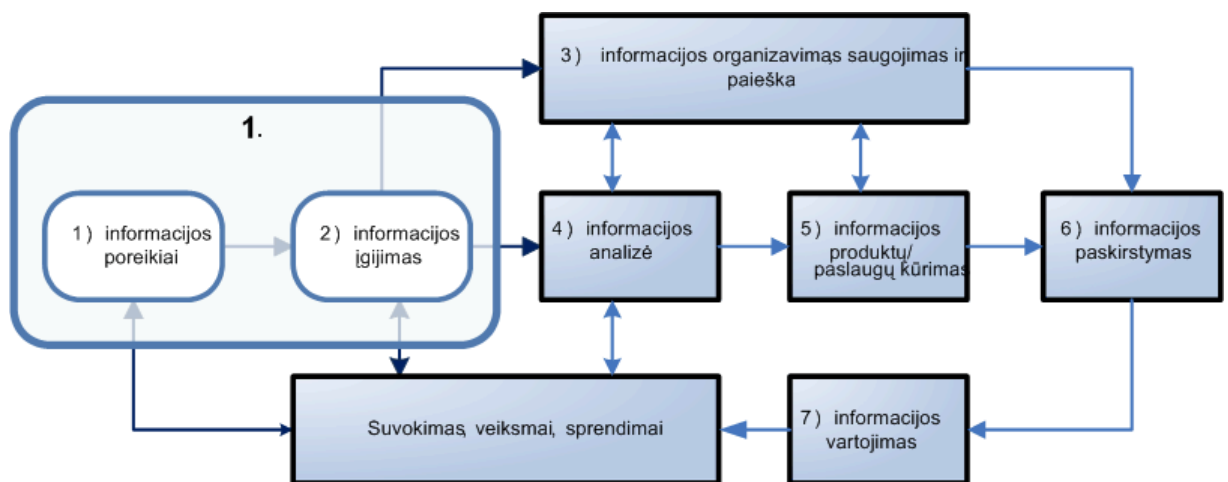


Pastebime, kad nuotolinio mokymosi sistemų pagrindinis funkcinis elementas, kurį apdoroja, yra informacija. Žinios – tai jau sekančio lygmens informacija, turinti savyje papildomos informacijos (pripažintos sistemos ir turinčios didesnės galios sprendimų priėmimo). Nuotolinio mokymosi sistemos daugiausiai informacijos gauna iš aplinkos (tiek vidinės, tiek išorinės). Informacija

turi savyje surinktus duomenis apie mokinį ar jų grupę. Gautą informaciją sistema turi atitinkamai apdoroti ir atsižvelgiant į gautus rezultatus pateikti grįžtamąjį ryšį. Visa tai apima informacijos rinkimo, saugojimo, apdorojimo ir perteikimo procesus, kuriuos detalčiai nagrinėja informacijos vadyba.

Pirmiausia, galime optimizuoti modelį (Eidukas, 2002), apjungdami *informacijos poreikių* ir *informacijos įgijimo* elementus į vieną, kadangi sistema yra užprogramuota surinkti tam tikrą informaciją ir ją atpažinti. Tai parodyta 10 paveikslo pirmame (1) laukelyje.

10 paveikslas. Informacijos procesų modelis po pirmojo optimizavimo etapo, remiantis L. Markevičiūtės (2008) pateiktu modeliu

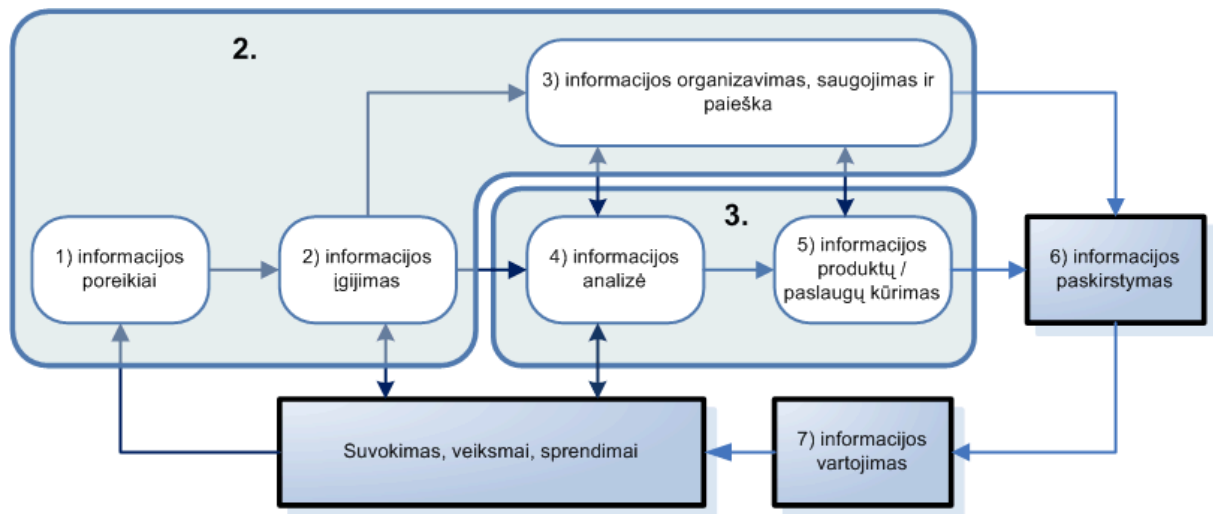


Kadangi informacijos gavimas yra siejamas su informacinėmis technologijomis, tai suprantama, kad bet kokia gauta informacija yra iš karto saugoma tam tikru skaitmeniniu pavidalu (pvz.: įrašoma į laikiną atmintį, įvedama į duomenų bazes ar „skaitymo buferį“).

Pastarasis technologinis aspektas leidžia eliminuoti ir *informacijos organizavimo, saugojimo ir paieškos* elementą sujungiant su *informacijos gavimu*. Tai atitinka atrąjį modelio optimizavimo etapą.

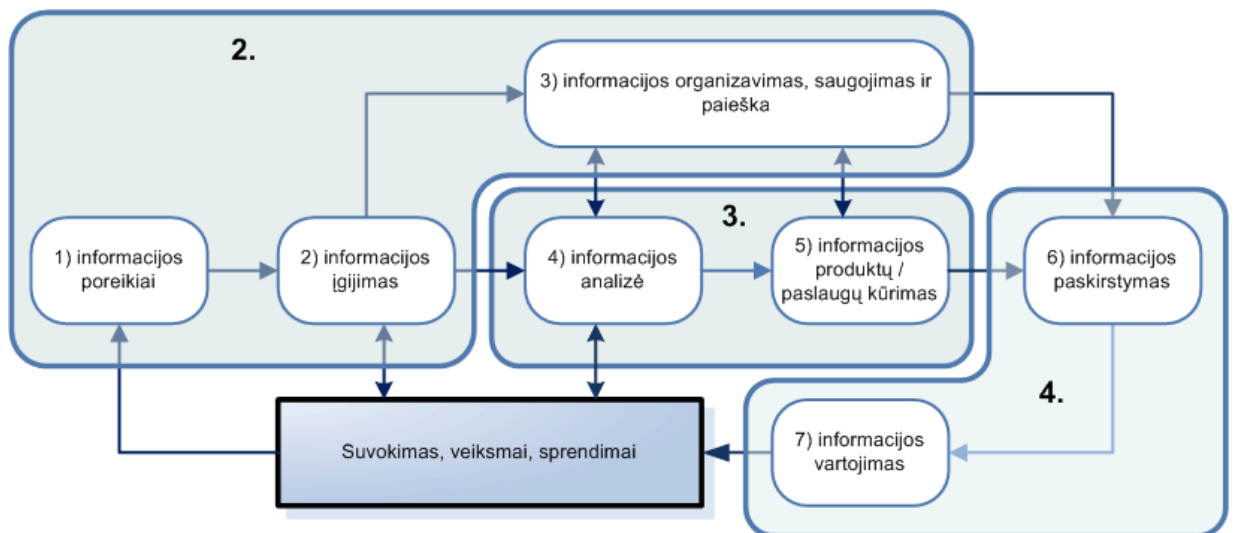
Po to, nuosekliai tirdami L. Markevičiūtės analizuotą informacijos procesų modelį, susiduriame su *informacijos apdorojimo – analizės* elementu. Pastarąjį galime apjungti su *informacijos produktų ir paslaugų kūrimo* elementu (11 paveikslas), kadangi šiuolaikiniame moksle ir jo taikyme informacijos betikslė analizė yra bevertė. Analizės rezultatu (produktu) gali būti ataskaita apie informaciją (vidinis auditas, kontrolė) ar tam tikro modelio sukūrimas (naujos informacijos – žinių kūrimas). Trečiojo optimizavimo etapo iteracinius žingsnius akivaizdžiai matome 11 paveikslo antroje (2) ir trečiame (3) laukeliuose.

11 paveikslas. Informacijos procesų modelis po trečiojo optimizavimo etapo, remiantis L. Markevičiūtės (2008) pateiktu modeliu



*Informacijos paskirstymas ir informacijos vartojimas* gali būti sujungtas. Tai parodyta 12 paveikslas ketvirtame (4) laukelyje. Nagrinėjamas kontekstas – nuotolinio mokymosi sistema – jau savaime siejasi su klasifikuota ir atrinkta šiai sistemai tinkančia informacija. Jei informacija sistemai yra „neaktuali“, tuomet duomenys „nepraeina“ pirmojo – *informacijos poreikių / gavimo* etapo, kuris apsaugo sistemą ir jos komponentus nuo netinkamos ir nereikalingos informacijos.

12 paveikslas. Informacijos procesų modelis po ketvirtojo optimizavimo etapo, remiantis L. Markevičiūtės (2008) pateiktu modeliu



Pagal informacijos vadybos modelį *informacijos suvokimas, veiksmai ir sprendimai* yra atliekami nepriklausomai nuo pagrindinių elementų. Kadangi mūsų nagrinėjama sritis yra siauresnė ir konkretizuota, tai šis etapas yra jau savaime integruotas į *informacijos analizės ir produktų kūrimo* etapą. Bendruoju pavidalu gautąjį supaprastintą modelį galime pavaizduoti trimis etapais, kaip parodyta 13 paveiksle.

13 paveikslas. Gautasis supaprastintas informacijos procesų modelis po visų optimizavimo etapų, vadovaujantis 10–12 paveiksluose pateiktaisiais modeliais.



Punktyrine linija yra žymimas nebūtinai, tačiau galimas ryšys naujai sukurtajame informacijos procesų modelyje galimoms e. mokymosi sistemoms. L. Markevičiūtės pateikto (Markevičiūtė, 2008) ir naujai sukurto modelių palyginimas pagal turimus elementus (7 lentelė):

7 lentelė. Tirtų informacijos procesų modelių palyginimai

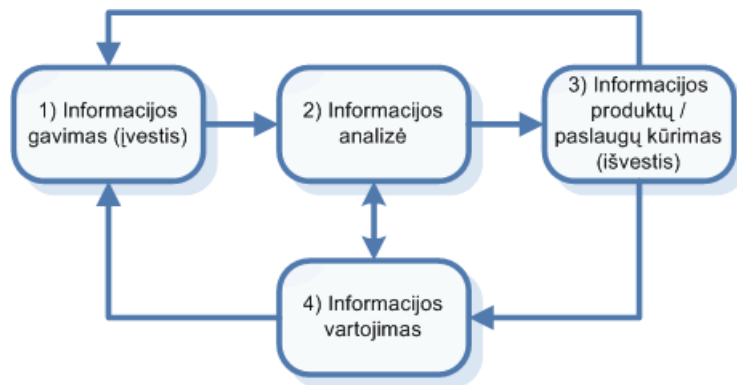
L. Markevičiūtės pateikto modelio etapai	Naujai sukurtojo modelio etapai
1. Informacijos poreikiai	1. Informacijos gavimas (įvestis)
2. Informacijos įgijimas	
3. Informacijos <ul style="list-style-type: none"> <li>• organizavimas</li> <li>• saugojimas</li> <li>• paieška</li> </ul>	
4. Informacijos analizė	2. Informacijos analizė
5. Informacijos produktų / paslaugų kūrimas	
6. Informacijos paskirstymas	3. Informacijos vartojimas
7. Informacijos vartojimas	

Sukurtasis modelis atitinka dirbtinio intelekto sistemų modelio elementą – agentą. Agentai ir agentinės sistemos yra nagrinėjamos tolimesniuose skyriuose. Remiantis atliktais naujausiais tyrimais (Ramašauskas, 2010), pastarajai sistemai atitiktų agentas, kuris neturi pakankamo intelektualizacijos lygio, o tik geba surinkti informaciją, ją apdoroti ir „aklai“ grąžinti rezultata.

Intelektualizuotas ir dinamiškas agentas gali būti realizuotas sistemoje, kurioje egzistuoja atskirtis tarp *informacijos vartojimo* ir *informacijos produktų – paslaugų kūrimo*, kadangi informacijos vartojimą traktuojame kaip sukurtų rezultatų pakartotinį panaudojimą to paties arba kitų, sistemoje egzistuojančių, agentų. Taip pat tokioje sistemoje informacijos vartojimas yra apibrėžiamas kaip dinamiųjų duomenų (informacijos produktų) įvestis, kuri realiame laike priklauso nuo kitų agentų darbo eigos. Tarkime, kad sistemoje veikia keletas intelektualizuotų agentų, kuriems pagrindinę informaciją teikia mokiniai (pavyzdžiui, testo klausimų atsakymai) per informacijos įvesties kanalą, o per informacijos vartojimo kanalą analizei yra pateikiami duomenys, gauti iš kitų agentų

(pavyzdžiui, informacija apie kitų mokinių mokymąsi ta pačia tema, tuo pačiu metu). Toks modelis gali būti realizuotas, siekiant sumažinti mokinio nesąžiningo mokymosi apraiškas (kai mokiniai, prisijungę prie nuotolinio mokymosi sistemos, siekia išlaikyti egzaminą ir, gavę klausimus, naudoja kitas parankines komunikavimo priemones, nesąžiningai dalinasi informacija ir žiniomis). Agentai, pateikdami klausimus, gali tarpusavyje keistis turima informacija apie užduotis ir taip paveikti sistemą, kad toks komunikavimas tarp mokinių taptų jų pačių „priešu“ laiko atžvilgiu, parenkant kiekvienam mokiniui vis kitus klausimus, kurie nesikartotų su kitų mokinių klausimais. Aptarto intelektualizuoto nuotolinio mokymosi sistemos agento modelis pagal informacijos procesus gali atrodyti taip (14 paveikslas).


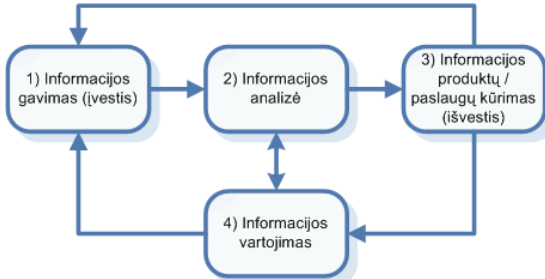
14 paveikslas. Intelektualizuoto agento informacijos procesų modelis



Tai, kad sukurtieji modeliai yra panašūs savo struktūra į L. Ramašausko nagrinėtas agentų struktūras, galime įsitikinti pateiktoje lentelėje (8 lentelė).

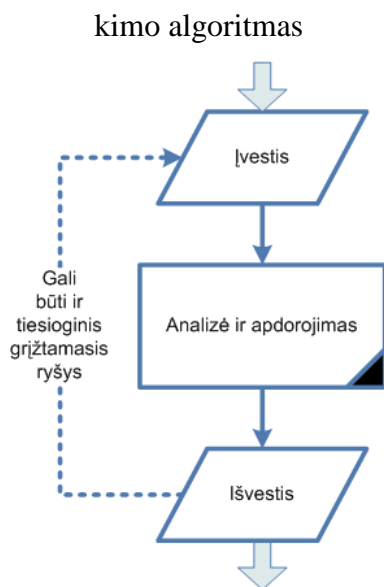
8 lentelė. Agentų ir informacijos procesų modelių palyginimas skirtingose sistemose

	<b>Mažai intelektualizuota sistema</b>	<b>Intelektualizuota sistema</b>
Agento modelis tiriamoje sistemoje	<p>15a paveikslas. Neintelektualizuoto agento ryšys su aplinka</p>	<p>15b paveikslas. Intelektualizuoto agento ryšys su aplinka</p>

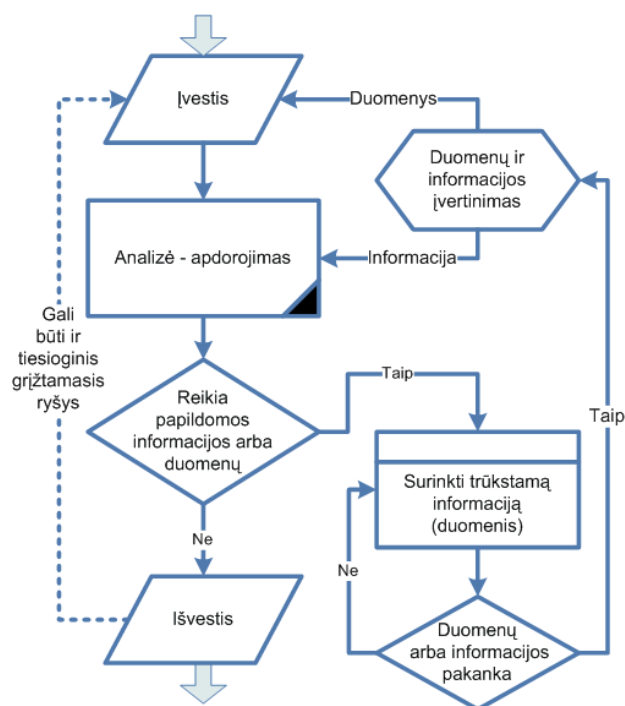
Informacijos procesų modelis	<p>15c paveikslas. Sukurtasis neintelektualizuoto agento informacijos procesų modelis (iš 13 paveikslo)</p> 	<p>15d. Sukurtasis intelektualizuoto agento informacijos procesų modelis (iš 14 paveikslas)</p> 
------------------------------	---	--

Pagal gautuosius (15a, 15b, 15c, 15d paveikslai) paprastojo ir intelektualizuoto agentų informacijos procesų modelius galima sudaryti atitinkamų agentų veikimo algoritmus (16, 17 paveikslai), kurie gali būti taikomi kaip modeliniai algoritmai agentų techninėje realizacijoje.

16 paveikslas. Neintelektualizuoto agento veikimo algoritmas



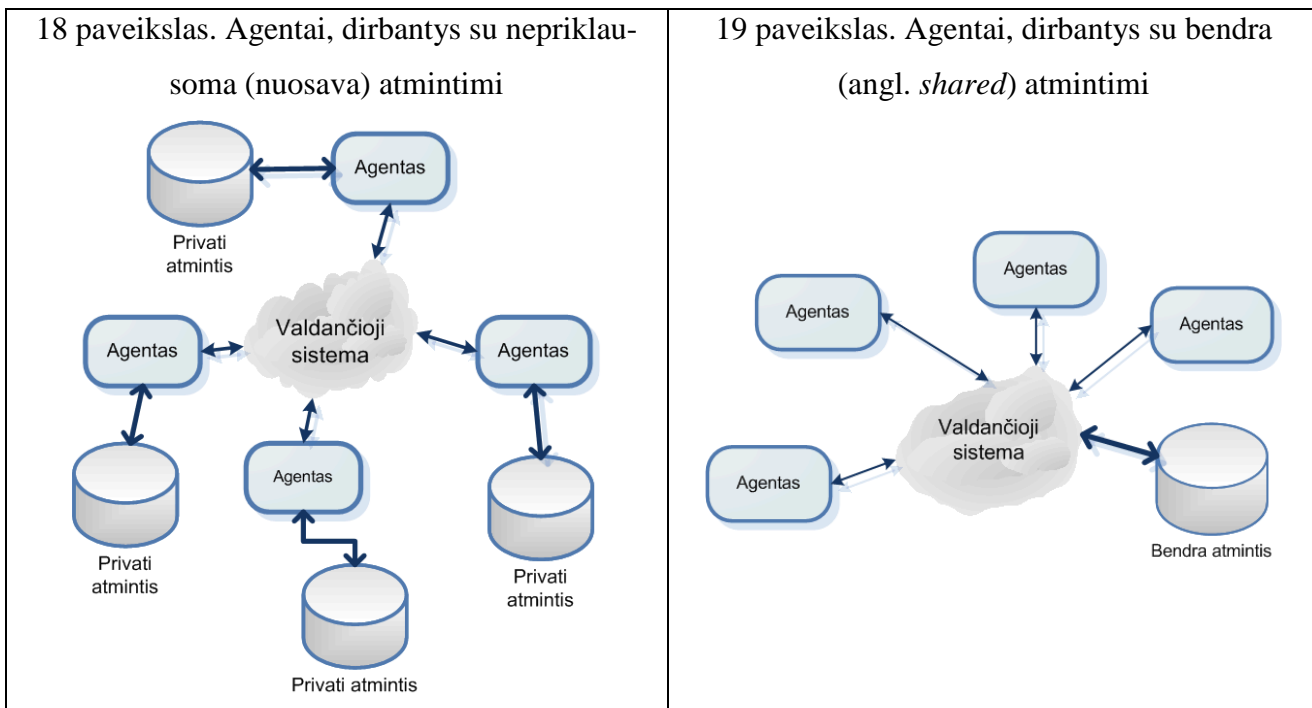
17 paveikslas. Intelektualizuoto agento veikimo algoritmas



Tam, kad intelektualizuotieji agentai galėtų efektingai atlikti savo funkcijas, jie privalo kaupti tam tikrą informaciją ir kurti žinias. Remiantis O. Vasileco informacinių sistemų kūrimo metodais ir technikomis, duomenys gali būti saugomi įvairiais būdais ir turėti skirtingą priėjimą prie jų. Išskirkime du pagrindinius agentų duomenų pasiekimo metodus:

1. Kai agentai turi vidinę, nepriklausomą nuo valdančiosios sistemos atmintį ir bendrus (dalinimuisi skirtus) duomenis, saugomus privačiose bibliotekose (18 paveikslas).

2. Kai agentai yra tam tikros bendros atminties naudotojai ir dirba su bendra adresine erdve (19 paveikslas).



Nesunkiai galime išvelgti tiek privalumus, tiek trūkumus abiejų agentų architektūrų sistemoje. Centralizuotos atminties privalumų ir trūkumų palyginimas yra pateiktas 9 lentelėje.

9 lentelė. Agentų atminties naudojimo architektūrų palyginimų lentelė

Privalumai	Trūkumai
Centralizuoti ir lengvai prieinami duomenys įgalina nesunkią ir greitą informacijos paiešką bendro naudojimo atmintyje, kadangi adresinė erdvė yra visiems agentams vienodai suprantama, apskaičiuojama (naudojami tie patys matematiniai algoritmai duomenų paieškai).	Rizika prarasti visus duomenis, esant sistemos nesklandumams, kai duomenys saugomi centralizuotai. Norint to išvengti, privaloma daryti atsargines kopijas, kurios gali užimti daug papildomos vietos sistemos išorinėje atmintyje.
Centralizuotų duomenų vienodumas formato atžvilgiu – agentai naudoja suvienodinto formato duomenis tam, kad nerėiktų naudoti papildomų algoritmų duomenų konvertavimui.	Sistemos darbo sulėtėjimas, priimant globalius sprendimus (skaitant ar rašant į pagrindinę atmintį). Jei atmintis artėja prie visiško jos užpildymo, atsiranda uždelsimas (angl. <i>lag</i> ), kadangi agentai, norintys rašyti / nuskaityti duomenis iš tam tikros adresinės erdvės, kurią jau naudoja rašymui kitas agentas, turi laukti, kol atsilaisvins. Taip užtikrinamas duomenų vientisumas ir



	korektiškumas.
Greita komunikacija tarp agentų, kadangi agentai pirmiausia komunikuoja tarpusavyje ir, esant reikalui, gali persiųsti tam tikrą informaciją. Jie gali persiųsti ne visą jos kiekį, o tik adresą su nuoroda į bendrosios atminties dalį, kur yra patalpinta reikiama informacija. Taip neapkraunamos komunikacinės magistralės.	Greitas neekonomiškos informacijos ir žinių plitimas gali atsirasti tuomet, kai agentas arba agentai pradeda veikti nekorektiškai ir skleisti „melagingą“ informaciją sistemos viduje. Šis atvejis yra mažai tikėtinas, tačiau įmanomas, jei suprojektuota sistema turi spragų saugumo ir išbaigtumo požiūriais.

## 1.5 Apibendrinimas

Pirmajame skyriuje buvo apžvelgta e. mokymosi koncepcija, e. mokymuisi keliami technologiniai reikalavimai, pristatyti virtualių mokymosi aplinkų tipai, jų sandara ir komponentų atliekamos funkcijos. Šiame skyriuje atskleistas informacijos ir žinių vadybos procesų panaudojamumas ir modelių panašumai su e. mokymosi sistemų techninio realizavimo produktais. Pagilintas adaptyviųjų virtualiųjų mokymosi aplinkų supratimas ir pastebėta, kad viena iš esminių užduočių tokioms aplinkoms yra mokinio žinių lygio nustatymas bei mokinio mokymosi modelio generavimas.

Parodyta, kad kol kas nėra sukurta universali intelektualizuota nuotolinio mokymosi sistemos agentų architektūra. Kuriant informacinę sistemą, kuri įvykdytų visus e. mokymosi reikalavimus, reikėtų pasirinkti tokią architektūrą, kuri tenkintų svarbiausius aplinkai keliamus reikalavimus (saugumo, duomenų apdorojimo greičio, patikimumo, paslaugumo ir pan.) ir atitiktų reikalaujamą informacijos ir žinių valdymo būdą.

Įžvelgti supaprastinto informacijos procesų modelio ir e. mokymosi sistemos technologinių elementų panašumai operaciniame lygmenyje leidžia manyti, kad pritaikius dirbtinio intelekto įskiepius, e. mokymosi sistema geriau tenkintų išskeltus reikalavimus. Šių uždavinių sprendimas atskleidžiamas sekančiuose šio darbo skyriuose.

## 2. DIRBTINIO INTELEKTO ALGORITMŲ E. MOKYMOSE SISTEMOSE TYRIMAS

Parama mokantis yra svarbus elementas, kadangi, vykdant nuotolinį mokymąsi (nesilaikant tradicinio mokymosi etalono), dažnai besimokančiajam iškyla nemažai klausimų, į kuriuos jis negali pats atsakyti, o reikia kreiptis į paslaugų teikėją ar atitinkamą asmenį, susijusį su tam tikra problema. Kadangi visuomenės tobulėjimo ir užimtumo mastai yra dideli, tai kiekviena minutė vertinama ir laikas tampa vertingu turtu. E. mokymasis yra vykdomas virtualioje mokymosi aplinkoje, prie kurios bet kada gali prisijungti mokinys ir mokytojas. Taupant mokytojų laiką (nuolatinis prisijungimas prie sistemos reikalauja finansinių kaštų) yra ieškoma naujų kelių, kaip būtų galima pakeisti – pavaduoti mokytojus. Čia į pagalbą ateina dirbtinio intelekto elementai (agentai), kurių pagalba galime sutaupyti brangų mokytojų laiką (suteikti daugiau laiko mokslinei veiklai, nei mokinių konsultacijoms), realizuojant e. patarėjo modelį. Taip pat galime nešališkai vertinti mokinio žinias ir pagal jo žinių lygį parinkti atitinkamą kurso sudėtingumą bei taikytinus metodus mokymosi procese, pasitelkus e. planuotojo modelį.

### 2.1. E. mokymosi sistemų intelektualizacija ir modeliavimas

Neatsiejama RL aplinkos modelio dalis yra agentai, kurie yra jungiami prie aplinkos per įvesties ir išvesties kanalus. Agentai pagal atitinkamą įvesties reikšmę pateikia tam tikrą rezultatą išvestyje, prieš tai atlikus sudėtingus matematinius skaičiavimus. Dažnai sutinkama, kad agento gebėjimas priimti sprendimą yra vadinamas agento elgsena (Ramašauskas, 2010; Kaebling, 1996). Standartinių agentų elgsena apsiriboja iteraciniais sumų skaičiavimais ir elementariaisiais sąlyginiais perėjimais tarp aritmetinių veiksmų (Iglesias, 2003; Kaebling, 1996).

Dirbtinio intelekto agento apibrėžimų yra ne vienas. A. A. Bielskis pateikia paprastą agento apibrėžimą:

*„Dirbtinio intelekto agentas – tai aktyvi nepriklausoma programa arba programa, esanti tam tikros programų sistemos dalis“* (Bielskis, 2004).

Detalesnis ir tikslesnis apibrėžimas, kuriuo ir vadovausimės, yra pateiktas G. Weiss:

*„Agentai – yra nepriklausomos nuo žmogaus, skaičiuojamosios esybės, kurias galime vertinti kaip sensorinius aplinkos stebėtojus ir poveikio darytojus tai aplinkai. Skaičiuojamosios esybės reiškia tai, kad agentai gali fiziškai egzistuoti kokioje nors formoje (pvz.: kompiuterių programos dalis). Agentų esminis darbas yra nuoseklus įvairių užduočių vykdymas“* (Weiss, 1999).

Remiantis Denisovo (2000) ir Bielskio (2004) darbais išskiriami svarbiausi agento atributai:

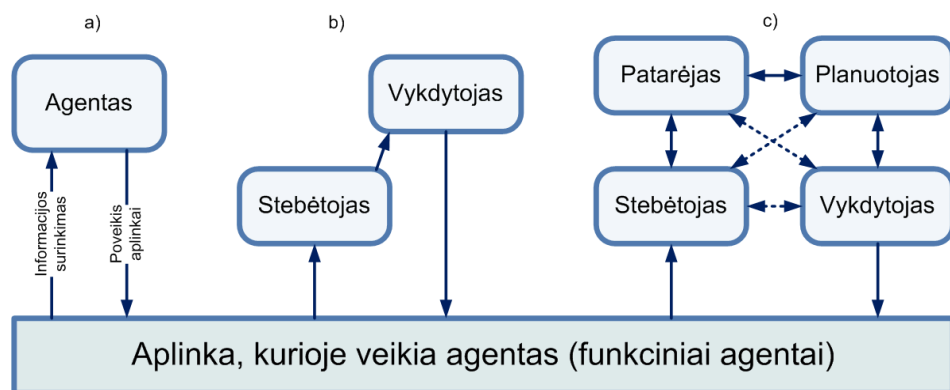
- autonomiškumas – gebėjimas veikti be žmogaus įsikišimo;

- atvirumas – gebėjimas komunikuoti su kitais objektais (pvz.: agentais ir/ar žmogumi);
- jautrumas – gebėjimas suvokti aplinką ir priimti atitinkamus sprendimus;
- aktyvumas – gebėjimas perimti iniciatyvą, kai to reikalauja sąlygos;
- mokymasis – gebėjimas tobulėti, gerinti atliekamą darbą laike.

Agentai, kurie veikia pavieniui, nepriklausomai nuo kitų aplinkų, dažniausiai yra ypač komplikuoti, kadangi vienas agentas privalo atlikti daug funkcijų. Tokių agentų kokybiškas sukūrimas ir realizavimas labai brangiai kainuoja bei atima laiko. Taipogi padidėja rizika sistemos saugumo aspektu, kadangi vienas agentas yra atsakingas už daug funkcijų ir klaidos – nelaimės atveju, agentui nustojus veikti, sistema praranda funkcionalumą. Siekiant išvengti tokių nemalonumų, sudėtingi agentai yra skaidomi į smulkesnius funkcinius agentus, kurie jau įprasmina atvirumo atributą – gebėjimą bendradarbiauti.

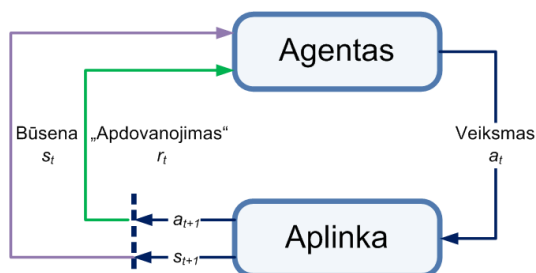
Didelėse, atvirose, dinamiškose ir neprognozuojamose aplinkose paprastai naudojamos *daugiaagentės sistemos* (angl. *MultiAgent Systems*, toliau tekste – MAS). MAS patrauklumas sparčiai auga akademinuose ir pramoniniuose tyrimuose. Sukurti jas per trumpą laiką minėtose aplinkose yra labai sunku, o kartais iš viso neįmanoma. Labiausiai tikėtinas kelias tai padaryti būtų agentų aprūpinimas sugebėjimais mokytis, padidinančiais jų individualų vaidmenį ir gerinant visos sistemos veikimą. Besimokantiems agentams svarbu nustatyti mokymosi strategiją, kuri leistų agentams sėkmingai save adaptuoti dinamiškai kintančioje aplinkoje. Agentų tipų palyginimas matomas (20 paveikslas) pateiktame paveiksle.

20 paveikslas. Agentai: (a) vienišiai, (b) vienafunkciniai ir (c) bendradarbiaujantys



Standartinio agento vizualizavimas RL sistemoje pavaizduotas 21 paveiksle, kur laiko parametrą atitinka  $t$ . Kiekvieno laiko momentu  $t$  agentas būna tam tikroje  $s_t \in S$  būsenoje, vienoje iš visų galimų  $S$  būsenų. Agentas būsenai  $s_t$  parenka atitinkama veiksmą  $a_t$  iš visų galimų  $s_t$  būsenos veiksmų aibės  $a_t \in A(s_t)$ . Iškart gaunamas apdovanojimas (angl. *Reward*)  $r_{t+1} \in R$  ir nustatoma sekanti būseną  $s_{t+1}$  (Sutton, 1998).

21 paveikslas. Agento ryšiai su aplinka RL sistemoje



Pastebime, kad agentai yra tam tikros programos, kurios veikia tam tikroje aplinkoje ir pagal savo sudėtingumą ir funkcionalumą gali būti skirstomos į bendro ir mąstančiojo tipo agentus. Bendrinio ir „mąstančio“ agentų padėtys ir santykis su aplinka yra pavaizduoti 15a paveikslas ir 15b. paveiksluose. Intelektika suteikia agentams galimybę atlikti paskirtas užduotis optimaliu keliu.

Agentai yra kuriami programavimo kalbomis (pvz.: LISP, JAVA, C++), nes jos pasižymi geromis tinklinėmis, nepriklausomumo nuo platformos savybėmis, kurios yra esminės efektyviam agentų darbui nuotolinio mokymosi sistemose. Paprastai dirbtinio intelekto agentai veikia sistemose (agentinėse programų sistemose), atlikdami tam tikrą užduotį. Agentinės programų sistemos yra plačiai taikomos gamybos, telekomunikacijų, finansų, elektroninio verslo, transporto valdymo ir elektroninio mokymosi sistemose (Bielskis, 2004; Lakami, 1998). Elektroninio mokymosi sistemoje yra išskiriami du dirbtinio intelekto agentų tipai: mokymo (angl. *educational*) ir sąsajos (angl. *interface*) agentai (Brusilovsky, 1999; Murray, 1999).

### 2.1.1.1 Mokymo agentai

Tai agentai, kurie įprasmina du modelius.

1. Informacijos sklaidos modelį.
2. Optimalaus komunikacijos būdo pateikimo per IKT tam tikra tema vartotojui modelį.

Mokymo tipo agentams yra svarbiausios mokymo turinio ir komunikacijos strategijos savybės.

### 2.1.1.2 Sąsajos agentai

Sąsajos agentai yra skaičiavimų apdorojimo komponentai, kurie elgiasi kaip žmonių asistentai. Šių agentų veikimo principas yra toks: stebint vartotojo aplinką, sukurti atitinkamą vartotojo profailą ir pagal jį prognozuoti vartotojo tolimesnius veiksmus (pavyzdžiui, sprendžiant problemą, pateikti reikiamą informaciją konkrečiu klausimu, numatant problemos vystymosi kryptį). Tokie agentai, surinkę pakankamai informacijos apie vartotoją, gali pradėti atlikinėti užduotis, kurias šiaip atliktų kitas gyvas asmuo (pavyzdžiui, virtualios sekretorės arba pagalbos linijos operatorės rolės) (Amandi, 2003).

### 2.1.1.3 E. patarėjas

E. patarėjas (angl. *Learning Assistant*) – tai dirbtinio intelekto agentas, kuris atlieka tiek sąsajos, tiek mokymo agento roles. Šio agento užduotis yra stebint vartotoją ir jo aplinką, pateikti vartotojui optimalią informaciją, kurią jis turėtų gauti pagal savo gebėjimus ir žinių lygį (žinių lygio nustatymo problemos buvo nagrinėtos 1 skyriuje). Klausimai, kurie apsunkina nuotolinio mokymosi e. patarėjo realizavimą būtų šie:

1. Kokias žinias turi vartotojas (mokinys).
2. Kokiam lygmenyje mokinys išmano dėstomą dalyką (teoriniame ar praktiniame).

3. Kaip mokinys geba suvokti koncepcijas ir neapibrėžtumus bei kokia forma yra priimtinausia informacijos pateikimui (grafikai, apibrėžimai, detalūs aprašymai, mišrūs metodai) (Amandi, 2003).

Iš šių problemų išplaukia tai, kad e. patarėjai turi būti individualizuojami kiekvienam vartotojui ir informacija apie vartotoją turi būti saugoma tam tikrame profilyje, pagal kurį agentas priiminėja sprendimus. Šios problemos sprendimui yra plačiai naudojamas specialus greito mokymosi (angl. *Q-learning*) algoritmas, kurio taikymo sritys yra intelektinė robotika (robotų apmokymai), realaus laiko intelektualios sistemos (atominų reaktorių valdymo sistemos, oro uosto sistemų valdymas) ir taip pat intelektualiųjų agentų apmokymas (paruošimas) jų pagrindinių užduočių vykdymui. Toliau tiriamas Q-learning algoritmas ir jo pritaikymo galimybės realizuoti e. patarėjuje, kuris iš aplinkos gauna tiesioginę informaciją savo funkcijoms atlikti (Iglesias, 2003; Bielskis, 2004).

## 2.2 Dirbtinio intelekto agentų realizavimo algoritmai

Be Q-learning algoritmo praktikoje yra naudojami ir sudėtingesni algoritmai, kaip uždelsto apdovanojimo algoritmas, modifikuotų Markovo grandinių modelių algoritmai ir pan. Keletas jų bus aprašyti tolimesniuose poskyriuose.

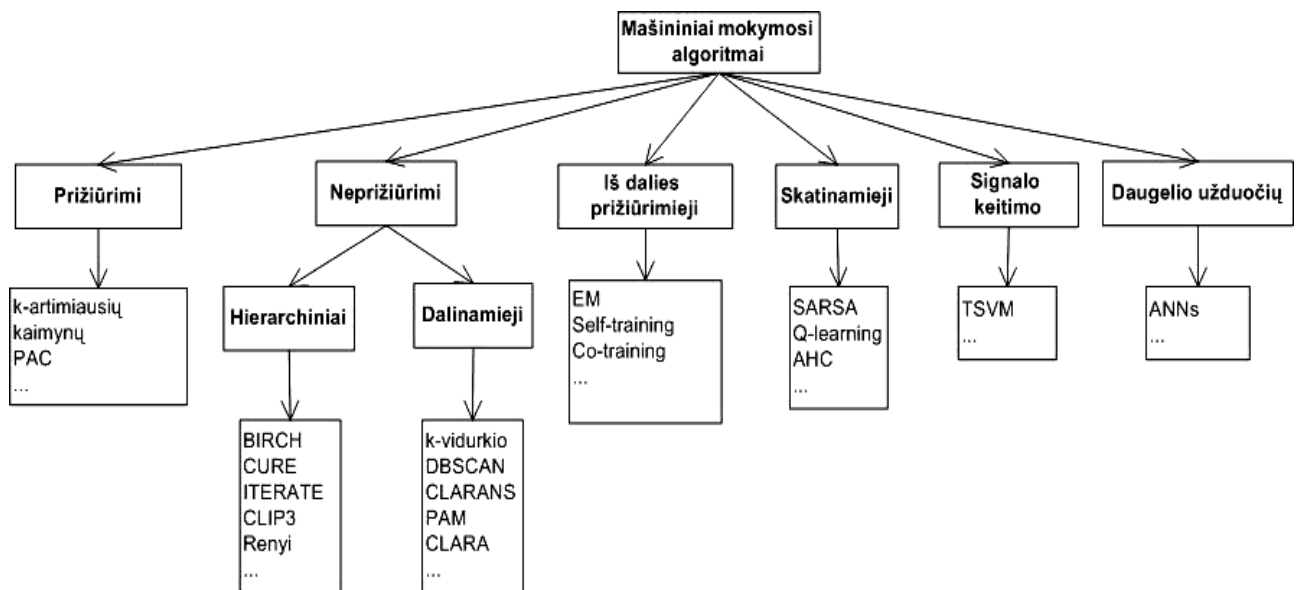
### 2.2.1 Dirbtinio intelekto algoritmų skirstymas

Mašininio apmokymo (angl. *Machine Learning*, toliau tekste – ML) algoritmai yra skirstomi į tipus pagal jų veikimo principus ir kontrolės aparatą (22 paveikslas):

- **Iš dalies prižiūrimieji** mokymosi algoritmai (angl. *Semi Supervised Learning*) – sujungia pažymėtus ir nepažymėtus pavyzdžius atitinkamai funkcijai arba klasifikatoriui sugeneruoti.
- **Skatinamieji** mokymosi algoritmai, kurie keičia elgsenos strategiją, priklausomai nuo pateiktų žinių apie aplinką. Kiekvienas veiksmas turi įtakos aplinkai, o aplinka gražina informaciją, pagal kurią vadovaujasi besimokantis agentas.

- **Signalų keitimo** mokymosi algoritmai (angl. *Transduction Learning*) – algoritmai, panašūs į prižiūrimą mokymąsi, bet skirtingai nuo jo, nesiekiantys sukonstruoti tikslią funkciją. Grindžiama bandymais nuspėti naujus išvedimus pagal jau turimų įvedimų ir išvedimų informaciją.
- **Daugelio užduočių** (angl. *Multitask Learning*) mokymosi algoritmai – algoritmai, nagrinėjantys savo induktyvia, tendencingumu pagrįsta ankstesne patirtimi.
- **Prižiūrimas** (angl. *Supervised*) – tai algoritmas, dar vadinamas kaip „mokymasis iš patirties“, kai klasifikavimo sistema gauna įvedimo duomenis ir atsakymus (mokymosi duomenų rinkinį (angl. *Training Set*)).
- **Neprižiūrimas** (angl. *Unsupervised*), „mokymasis be pavyzdžių“, mokymosi algoritmas negauna mokymosi duomenų rinkinio ir turi pats suklasifikuoti įvedimo duomenis. Klasifikavimas – tai neprižiūrimo mokymosi algoritmo uždavinys (Golouchova, 2008).

22 paveikslas. ML algoritmų taksonomija (Golouchova, 2008)



### 2.2.1 Uždelsto apdovanojimo modelis

Pastarasis modelis vadinamas uždelsto apdovanojimo modeliu (angl. *Delayed Reward*), kadangi aplinkoje veikiantis agentas gali nustatyti ne tik turimą paskatinimą, bet gali tikimybiniais metodais pabandyti nuspėti sekančią ateities būseną. Tokie agentai turi turėti galimybę mokytis iš ateities (gebėjimas skaičiuoti tikimybes ateities būsenoms) ir turi gebėti atskirti, kokie jo veiksmai yra priimtini, o kurie - nepageidaujami. Uždelsto apdovanojimo modeliai savo veikimo principu yra iteraciniai, generuojantys būsenų sekas, o kadangi remiamasi tikimybiniais skaičiavimais, todėl gauname Markovo grandinės proceso kontekstą (Kaelbling, 1996; Ramašauskas, 2009).

### 2.2.2 Markovo savybių integravimo metodai

RL sistemoje agentas priima sprendimus, remdamasis iš aplinkos gautais signalais – aplinkos būsenomis. Turimas mokymosi sistemos modelis yra vadinamas Markovo, jei jis tenkina Markovo proceso savybę: jei perėjimo būsenos yra nepriklausomos nuo bet kurios prieš tai buvusios aplinkos būsenos ar bet kurio agento veiksmo praeityje (Kaelbling, 1996; Sutton, 1998).

Markovo procesai gali turėti baigtinį arba begalinį būsenų bei apdovanojimų skaičių. Dažniausiai apsiribojama baigtinėmis aibėmis, siekiant išlaikyti paprastus matematinius skaičiavimus, leidžiančius greičiau atlikti tikimybių teorijos skaičiavimus (Sutton, 1998). Šio modelio esmę sudaro tikimybės, kurios padeda atsakyti į klausimą, į kur yra linkusi vystytis aplinka laiko momentu  $t + 1$ , kai sistemą stebime  $t$  momentu.

Toks sistemos gebėjimas matematiškai užrašomas kaip tikimybinius skirstinys (1)

$$P_r \{s_{t+1} = s', r_{t+1} = r \mid s_t, a_t, r_t, s_{t-1}, a_{t-1}, \dots, r_1, s_0, a_0\} \quad (1)$$

visiems  $s', r, (s_t, a_t, r_t, \dots, r_1, s_0, a_0)$ .

Jeigu būsenos signalas turi Markovo savybę (aplinkos reagavimas laiko momentu  $t + 1$  priklauso tik nuo būsenos ir veiksmo atvaizdavimų laike  $t$ ) tai dinamiką galime užrašyti taip (2)

$$P_r \{s_{t+1} = s', r_{t+1} = r \mid s_t, a_t\} \quad (2)$$

visiems  $s_t, a_t, r, s'$ .

Aptartos (1) ir (2) Markovo savybės yra svarbios RL procese, kadangi sprendimai ir įverčiai yra priimami kaip dabartinės būsenos funkcijos. Tai leidžia mums bandyti nuspėti ateities būseną, remiantis tikimybinais skaičiavimais, dažnai vadinamais Markovo sprendimų procesu (angl. *Markov Decision Process*) (Sutton, 1998; Ramašauskas, 2009).

Jei sistema nėra pilnai apibrėžta kaip Markovo proceso dalis, labai tikėtina, kad sistemoje veikia dažnai sutinkami bazinio Q-learning algoritmo žingsniai (Baziukaitė, 2007; Watkins, 1992).

Toliau darbe bus tiriami Q-learning algoritmo veikimo principai ir bus bandomi sukurti efektyvūs algoritmai bei taikomosios programos, realizuojančios sukurtuosius algoritmus.

## 2.3 Q-learning algoritmas ir jo analizė

Algoritmas yra pagrįstas tam tikros reikšmės ir veiksmo funkcijomis  $Q(s,a)$ , kuri apibrėžia tolimesnę veiksmą (Watkins, 1992). Algoritmo parametrai ir veikimo aprašymas pateikti 10 ir 11 lentelėse.

10 lentelė. Q-learning algoritmo parametrų reikšmės (Bielskis, 2004)

Parametras	Paaiškinimas
$s$	Būsena (angl. <i>State</i> )
$a$	Veiksmas (angl. <i>Action</i> )
$\alpha$	Mokymosi greičio koeficientas
$r$	Paskatinimo reikšmė – „premija“ (angl. <i>Reward</i> )
$\lambda$	Pasitikėjimo būsenos reikšmėmis koeficientas
$s'$	Ateities būsena
$a'$	Veiksmas būsenoje, turintis didžiausią $Q$ reikšmę

11 lentelė. Q-learning algoritmo veikimo planas (Iglesias, 2003)

<ul style="list-style-type: none"> <li>• Kiekvienai porai , inicijuojamos lentelės <math>Q(s,a)</math> reikšmės;               <ul style="list-style-type: none"> <li>○ Imama esamoji būsena <math>s</math></li> <li>○ Kartoti:                   <ul style="list-style-type: none"> <li>▪ Pasirinkti ir įvykdyti veiksmą <math>a</math> ;</li> <li>▪ Gaunama iškart „premija“ – grįžtamasis ryšys <math>r</math> ;</li> <li>▪ Stebima nauja būsena ;</li> <li>▪ Atnaujinamas <math>Q(s,a)</math> lentelės įrašas pagal tokį sąryšį:  <math display="block">Q(s,a) = (1 - \alpha)Q(s,a) + \alpha(r + \lambda \max_a Q(s', a'))</math> </li> <li>▪ Pereinama iš <math>s</math> į <math>s'</math> būseną.</li> </ul> </li> </ul> </li> </ul>
--

### 2.3.1 Adaptuotas Q-learning algoritmas

Remiantis A. Iglesias, adaptuotas Q-learning algoritmas (12 lentelė) naudojamas adaptyviose ir intelektinėse mokymo sistemose.

12 lentelė. Adaptuotas Q-learning algoritmas AIES sistemai

<ul style="list-style-type: none"> <li>• Kiekvienai porai , inicijuojamos lentelės <math>Q(s,a)</math> reikšmės;               <ul style="list-style-type: none"> <li>○ Imama esamoji būsena <math>s</math></li> <li>○ Kartoti:</li> </ul> </li> </ul>
--



- Pasirinkti ir įvykdyti veiksmą  $a$  ;
- Gaunama iškart „premija“ – grįžtamasis ryšys  $r$  ; premija lygi vienetui, kai mokinys pasiekia AIES tikslą (surenka tam tikrą rezultatą patikros metu) ir premija lygi nuliui visais kitais atvejais;
- Tikrinamos mokinio žinios būsenoje  $s'$  ;
- Atnaujinamas  $Q(s,a)$  lentelės įrašas pagal tokį sąryšį:

$$Q(s, a) = (1 - \alpha)Q(s, a) + \alpha(r + \lambda \max_a Q(s', a'))$$

- Pereinama iš  $s$  į  $s'$  būseną.

Šio adaptuoto algoritmo parametrus ir veikimo principą taikome, modeliuodami e. patarėją.

### 2.3.2 Mokymo proceso planas

Įsivaizduokime mokymosi aplinką, kurioje mokiniai sprendžia tam tikrus uždavinius pagal jiems pateiktas sąlygas, kur uždaviniai yra suskirstyti pagal sunkumą į keletą klasių. Iš pradžių kiekvienam mokiniui yra priskiriama lengviausių uždavinių klasė. Mokiniui teisingai išsprendus pakankamai tam tikro sudėtingumo uždavinių yra laikoma, kad mokinys įsisavino ir išmoko pateiktą medžiagą. Tuomet mokinys gauna leidimą spręsti sudėtingesnio lygio uždavinius. Mokinys, kuris neišsprendžia tam tikro lygio uždavinių, yra paliekamas prie to paties sunkumo uždavinių rinkinio, tačiau e. patarėjas tuos pačius uždavinius pateikia mokiniui jau kitu stiliumi. Siekiant tokio e. patarėjo darbo, pastarąjį reikia apmokyti, kad būtų žinoma, koks stilius tam tikrame uždavinių sudėtingume mokiniams yra palankesnis. Po apmokymo e. patarėjas renka stilius pagal savo turimus (sukauptus) duomenis apmokymo metu iš  $Q(s, a)$  lentelės reikšmių.

#### 2.3.2.1 Pradiniai duomenys

Pabandykime sumodeliuoti e. patarėjo agentą, kurio užduotis ir yra uždavinių pateikimo stilius mokiniui, pritaikant adaptuotą Q-learning algoritimą. Tegul būsenos atitinka sprendžiamų uždavinių sunkumo lygius (13 lentelė).

13 lentelė. Būsenų klasifikacija ir jų aprašymai

Būsenos $s$	Paaškinimas
B1	Pateikiami lengvi uždaviniai.
B2	Pateikiami vidutinio sunkumo uždaviniai.
B3	Pateikiami sunkūs uždaviniai.

Veiksmai – tai uždavinių pateikimas tam tikru stiliumi (14 lentelė) mokiniui. Naudojamas elementarus maksimalios reikšmės parinkimo būdas po apmokymo proceso.

14 lentelė. Galimų veiksmų aibė ir paaiškinimai

Veiksmai $a$	Paaiškinimas
A1	Pateikiamos pilnos uždavinių sąlygos su panašiais pavyzdžiais, iliustracijomis, mokomąja video medžiaga.
A2	Pateikiamos pilnos uždavinių sąlygos su minimaliomis užuominomis (uždavinio formuluotė video pavidale).
A3	Pateikiamos tikslios uždavinių sąlygos be papildomos informacijos, be paveikslukų, be užuominų.

### 2.3.3 Mokymo proceso rezultatas

Taigi, taikant Q-Learning algoritimą su standartinėmis sąlygomis (15 lentelė), mes galime sužinoti, kuris stilius yra palankesnis tam tikro sunkumo uždaviniams spręsti.

15 lentelė. Q-Learning algoritmo dažniausiai naudojamos parametrų reikšmės

Parametras	Standartinė reikšmė
$a$	0,9
$r$	0 arba 1
$\lambda$	0,1

Naudojant standartinės sąlygas mes negalime sužinoti, kuris stilius yra dažniau pateikiamas ir ar iš viso yra pateikiamas, kadangi  $Q(s, a)$  lentelėje yra saugomos nulinės reikšmės tiek neišsprendus, tiek net nesprendus uždavinių, pateiktų tam tikru stiliumi. Galime stebėti tik palankiausio veiksmo kitimą tam tikroje būsenoje (16 lentelė.).

16 lentelė. Q reikšmių lentelė po antro bandymo spręsti visų lygių klausimus (Bielskis, 2004).

$s \setminus a$	Stilius1	Stilius2	Stilius3
1	0,9	0,9891	0
2	0,0891	0,0899991	0,9981
3	0,0891	0,981	0,0981
4	0,9	0	0,981
5	0	0	1,071

Vykdamt algoritimą buvo pastebėta, kad nulinės reikšmės atsiranda tada, kai mokinys neišsprendžia uždavinių tam tikru stiliumi ir tam stiliui yra suteikiama nulinė premija. Ji neleidžia mums aiškiai sužinoti, ar sprendžiant uždavinius stilius buvo panaudotas, ar ne.

Tolimesniuose tyrimuose šis parametras ir yra nagrinėjamas. Naudojamos standartinės koeficientų reikšmės, išskyrus premijos vertes.

## 2.4 Parametro $r$ vertės įtaka e. patarėjo apmokymo procese

### 2.4.1 Parametro $r$ samprata

Vykdamas Q-Learning adaptuotąjį algoritmą pirmąjį kartą (po inicializacijos proceso, kurio metu visoms  $Q(s, a)$  reikšmėms yra priskiriamos nulinės vertės) didžiausią svarbą turi mokymosi greičio koeficientas  $\alpha$  ir „premijos“ parametras  $r$ , kadangi pirmo e. patarėjo apmokymo metu tik pastarųjų parametrų sandauga yra nenulinė (3), kurią galime išreikšti iš standartinio algoritmo užrašymo (4).

$$\begin{aligned} Q(s, a) &= (1 - \alpha)Q(s, a) + \alpha(r + \lambda \max_a Q(s', a')) \\ Q(s, a) &= (1 - \alpha)Q(s, a) + \alpha r + \alpha \lambda \max_a Q(s', a') \end{aligned} \quad (3) \text{ ir } (4)$$

Nagrinėjant apmokymo procesą, mokymosi koeficiento reikšmė visada bus  $\alpha = 0,9$  (tai reiškia, kad sistema mokosi sparčiai) ir  $\lambda = 0,1$  (tai nurodo, kad mažai pasitikima būsimomis ateities būsenomis). Tuomet (4) formulę galime perrašyti įstatę žinomas koeficientų reikšmes (5):

$$Q(s, a) = \frac{1}{10}Q(s, a) + \frac{9}{10}r + \frac{9}{100} \max_a Q(s', a') \quad (5)$$

Yra įprasta, kad realizuojant Q-Learning algoritmą e. patarėjo sistemose parametro  $r$  reikšmė būna arba 0 arba 1, priklausomai nuo uždavinio išsprendimo sąlygų. Tokios reikšmės leidžia sekti e. patarėjo apmokymo procesą ir analizuoti pateikiamos medžiagos kokybę pagal uždavinių išsprendimą (analizuojamos  $Q(s, a)$  lentelės reikšmės). Tokių sąlygų rezultatų lentelės pavyzdys yra pateiktas apžvalginėje dalyje (16 lentelė).

Siekiant patikrinti  $r$  parametro įtaką algoritmui, buvo pasirinktos trupmeninės teigiamos, santykinai didelio tarpusavio teigiamo skirtumo ir skirtingo ženklų parametro  $r$  vertės, prie tokių pat pradinių sąlygų.

Tolimesnių tyrimų metu naudosime tokį parametro  $r$  žymėjimą nuo sąlyginio jo parinkimo: kai mokinys įvykdo keliamą reikalavimą sprenddamas uždavinius, tuomet paskiriama premija su didesne verte ir ją žymėsime  $r_1$ , kitu atveju –  $r_0$ .

Taip pat remsimės apžvalginėje dalyje nagrinėtais Q-Learning algoritmo veikimo principais, tačiau paprastumo dėlei naudosime tik tris veiksmus (tris pateikiamus stilių tipus) ir tris būsenas (tris sunkumo lygius). Realioje sistemoje galima naudoti daugiau veiksmų ir naudoti didesnę būsenų aibę. Nagrinėjamu atveju  $Q(s, a)$  inicializacijos metu atrods taip, kaip pateikta 17 lentelėje.

17 lentelė. Inicializuota nagrinėjama  $Q(s, a)$  reikšmių lentelė

$s \setminus a$	<b>A1</b>	<b>A2</b>	<b>A3</b>
B1	0	0	0
B2	0	0	0
B3	0	0	0

Tyrimo metu naudosime tas pačias apmokymo sąlygas su tikslu palyginti apmokymų skirtumus. Tegul apmokymo procese dalyvauja 5 mokiniai, kurie sprendžia uždavinius, ir tegul pirmasis mokinys galutinį rezultatą pasiekia tokia uždavinių išsprendimo – neišsprendimo ir stilių – būsenų seka: B1 (A1, **A3**), B2 (A2, A2, **A2**), B3 (A1, **A3**), kur B nurodo būseną (kuriame lygyje sprendžiami uždaviniai), skaičius greta B nurodo uždavinių sunkumo lygį (13 lentelė), skliaustuose pateikiamas stilių „atsitiktinis“ parinkimas, o paryškintas stilius žymi, kad mokinys išsprendė einamojo lygio užduotis atitinkamu stiliumi ir toliau sprendžia sekančio lygio uždavinius. Visų penkių mokinių atsitiktinai sugeneruotus sprendimus matome 18 lentelėje.

18 lentelė. Agento apmokymo procese dalyvaujančių mokinių sprendimų strategijos

1-as mokinys	B1 (A1, <b>A3</b> )	B2 (A2, A2, <b>A2</b> )	B3 (A1, <b>A3</b> )
2-as mokinys	B1 ( <b>A2</b> )	B2 (A1, <b>A2</b> )	B3 ( <b>A1</b> )
3-ias mokinys	B1 (A1, A1, <b>A3</b> )	B2 (A1, A3, <b>A1</b> )	B3 (A2, <b>A1</b> )
4-as mokinys	B1 (A3, <b>A2</b> )	B2 ( <b>A2</b> )	B3 ( <b>A2</b> )
5-as mokinys	B1 (A1, A2, <b>A1</b> )	B2 (A2, <b>A3</b> )	B3 (A3, <b>A3</b> )

Skaičiavimus palengvinkime taip, kad ieškodami ateities veiksmo (galimo pateiktino uždavinių stiliaus) su maksimalia reikšme ieškome tik einamojoje būsenoje, tuomet mūsų algoritmo formulė atrodys taip (6):

$$Q(s, a) = (1 - \alpha)Q(s, a) + \alpha r + \alpha \lambda \max_a Q(s, a') \quad (6)$$

Kitaip sakant, mokinys sprendžia to lygio uždavinius įvairiais stiliais tol, kol išsprendžia ir pereina į sekantį lygį (nukristi į žemesnį jau nebegali).

Nusistatę pradinės reikšmes ir skaičiavimo principus, analizuojame tris situacijas, kai premijos koeficientai yra pateikti vis kita forma.

#### 2.4.2 Skirtingų reikšmių parametro $r$ tyrimas

Parametro  $r$  reikšmė Q-learning algoritmo veikimui, kai  $r_1 = 0,8$  ir  $r_0 = 0,2$ .

Pirmajam mokiniui išsprendus užduotis pagal numatytą strategiją (18 lentelė), bus apmokomas e. patarėjas ir  $Q(s, a)$  lentelėje reikšmės pasikeistų, taip, kaip pateikta 19 lentelėje.

19 lentelė.  $Q(s, a)$  lentelės reikšmės po pirmojo apmokymo etapo

$s \setminus a$	A1	A2	A3
B1	0,18	0	0,7362
B2	0	0,760698	0
B3	0,18	0	0,7362

Naudodamiesi jau sugeneruota  $Q(s, a)$  lentele po pirmojo mokinio leidžiamame antrajam mokiniui spręsti uždavinius (taip pat pagal anksčiau apibrėžtą strategiją), kurio rezultatai modifikuoja (sisteminą) e. patarėjo turimus duomenis (20 lentelė).

20 lentelė. Antras e. patarėjo apmokymas

$s \setminus a$	A1	A2	A3
B1	0,18	<b>0,786258</b>	0,7362
B2	0,248463	0,864558	0
B3	<b>0,804258</b>	0	0,7362

Paryškintos tos reikšmės 20 lentelėje, kurios santykinę vertę daugiau nei pusę karto skiriasi nuo anksčiau turėtos prieš tai sprendusio mokinio. Tokiu apmokymo principu (mokiniam „sprendžiant“ uždavinius su jau žinoma jų sprendimo strategija) leidžiamame e. patarėjui būti apmokytam iš viso penkis kartus. Trečiojo, ketvirtojo ir penktojo mokinių uždavinių sprendimo įtaką  $Q(s, a)$  lentelės reikšmėms matome 21, 22 ir 23 lentelėse.

21 lentelė Trečiojo mokinio sprendimo rezultato suformuota  $Q(s, a)$  lentelė

$s \setminus a$	A1	A2	A3
B1	0,277639	0,786258	0,864383
B2	<b>0,826076</b>	0,864558	0,25781
B3	0,87281	0,252383	0,7362

22 lentelė Ketvirtojo mokinio sprendimo rezultato suformuota  $Q(s, a)$  lentelė

$s \setminus a$	A1	A2	A3
B1	0,277639	0,869389	<b>0,344232</b>
B2	0,826076	0,884266	0,25781
B3	0,87281	<b>0,767952</b>	0,7362

23 lentelė Penktojo mokinio sprendimo rezultato suformuota  $Q(s, a)$  lentelė

$s \setminus a$	A1	A2	A3
B1	<b>0,779668</b>	<b>0,345184</b>	0,344232
B2	0,826076	<b>0,348011</b>	<b>0,745781</b>
B3	0,87281	0,767952	0,831765

E. patarėjo apmokymo etapuose akivaizdžiai galime pastebėti, kaip kito  $Q(s, a)$  lentelės reikšmės. Pvz.: po ketvirto apmokymo B1 eilutėje A3 stulpelyje, lyginant su trečiojo mokinio sprendimo rezultatais iš esmės pasikeitė reikšmė, todėl palankus stilius pasikeitė į nepalankų. Tokį pasikeitimą nulėmė tai, kad ketvirtasis mokinys nesugebėjo išspręsti uždavinių, pateiktų A3 stiliu- mi, o B1 sunkumo įveikimas A2 stiliu tik sustiprino pastarojo stiliaus tinkamumą uždaviniams spręsti.

Po paskutinio (penktojo) apmokymo matome, kad  $Q(s, a)$  lentelėje yra beveik pusė ženkliai pasikeitusių reikšmių, nors po trečiojo ir ketvirtojo apmokymų tokių reikšmių buvo mažiau. Daro- me prielaidą, kad, norint apmokyti e. patarėją efektingam darbui, reikia žymiai daugiau nei penkių apmokymo etapų, kad susidarytų tam tikri dėsningumai.

Pasirinktų atlygio (premijos) parametru vertės mums leidžia stebėti rečiau naudojamus stilius tam tikrame uždavinių sunkumo lygmenyje, o tai parodo itin mažos  $Q(s, a)$  lentelės reikšmės (nely- gios nuliui). Nepanaudoti stiliai uždavinių sprendimo metu išlaiko nulines reikšmes.

Parametro  $r$  reikšmė Q-learning algoritmo veikimui, kai  $r_1 = 100$  ir  $r_0 = 0,01$ .

Tuo pačiu principu, kaip ir ankstesniame skyriuje, užpildome  $Q(s, a)$  lentelę, pasitelkę penkių mokinių sprendimų strategiją (8 lentelė) ir šio apmokymo  $Q(s, a)$  lentelės reikšmių kitimas mato- mas bendroje lentelėje (24 lentelė).

24 lentelė E. patarėjo apmokymo  $Q(s, a)$  bendra lentelė

Pirmas mokinys	A1	A2	A3
B1	0,009	0	90,00171
B2	0	90,002035	0
B3	0,009	0	90,00171
Antras mokinys			
B1	0,009	<b>98,10015</b>	90,00171
B2	8,109183	107,100387	0
B3	<b>98,101054</b>	0	90,00171
Trečias mokinys			
B1	9,721905	107,829185	90,00171
B2	<b>100,68493</b>	107,100387	9,648035
B3	108,6392	8,919093	90,00171
Ketvirtas mokinys			
B1	9,721905	110,487545	<b>18,713798</b>

B2	100,68493	110,349073	9,648035
B3	108,6392	<b>100,668621</b>	90,00171
Penktas mokinys			
B1	<b>102,815108</b>	<b>21,000424</b>	18,713798
B2	100,68493	<b>20,975324</b>	<b>109,130017</b>
B3	108,6392	100,668621	102,427976

Pastebime, kad lentelės reikšmės skiriasi nuo prieš tai nagrinėto modelio  $Q(s, a)$  reikšmių, nes esminis premijos parametras taip labai skiriasi. Esminių skirtumų tarp šių dviejų modelių nėra, nes jų  $Q(s, a)$  lentelės reikšmės kinta tuo pačiu principu (esminių kitimų vietos sutampa). Tačiau vienintelis skirtumas yra rekomendacinių stilių pasiūloje, kai po penktųjų apmokymų pirmuoju atveju siūloma B2 lygio uždavinius pateikti A1 stiliumi, o ką tik nagrinėtu – A3 stiliumi. Šis nesutapimas galimas dėl skaičiavimo operacijų. Kitų nesutapimų nepastebėta.

Parametro  $r$  reikšmė Q-learning algoritmo veikimui, kai  $r_1 = 1$  ir  $r_0 = -1$ .

Kitas įdomus modelio variantas yra tuomet, kai įvedama neigiama premija. Pirmiausia yra vertinamas tam tikros  $Q(s, a)$  lentelės reikšmės ženklas, o tik paskui konkreti skaitinė vertė. Kadangi norima akcentuoti ženklą (teigiami, neigiami skaičiai), o ne premijos dydį, buvo pasirinktos vienetinės premijos  $r_1 = 1$  ir  $r_0 = -1$ . E. patarėjo apmokymo strategija išlieka tokia pati kaip ir anksčiauose tyrinėjimuose (18 lentelė).

Gauti apmokymo rezultatai yra atvaizduoti bendroje 25 lentelėje.

25 lentelė  $Q(s, a)$  lentelės reikšmių kitimas naudojant  $r_1 = 1$  ir  $r_0 = -1$  reikšmes

Pirmas mokinys	<b>A1</b>	<b>A2</b>	<b>A3</b>
B1	-0,9	0	0,9
B2	0	0,801	0
B3	-0,9	0	0,9
Antras mokinys			
B1	-0,9	0,981	0,9
B2	-0,009	1,8711	0
B3	<b>1,8</b>	0	0,9
Trečias mokinys			
B1	-0,901881	0,981	1,07829
B2	<b>0,760825</b>	<b>-0,544491</b>	-0,731601

B3	1,242	-0,738	0,9
Ketvirtas mokinys			
B1	-0,901881	1,08639	<b>-0,695124</b>
B2	0,760825	<b>0,914025</b>	-0,731601
B3	1,242	<b>0,93798</b>	0,9
Penktas mokinys			
B1	<b>0,747389</b>	<b>-0,693586</b>	-0,695124
B2	0,760825	<b>-0,726335</b>	<b>0,895314</b>
B3	1,242	0,93798	0,941958

### 2.4.3 Parametro $r$ tyrimo rezultatų $Q(s, a)$ lentelių palyginimas

Remdamiesi apmokymo metu gautais rezultatais (po penkių mokymo žingsnių visais trimis atvejais) galime pastebime, kad lygiagrečiuose mokymo etapuose apmokoma beveik vienodai (lentelėse esminiai pasikeitimai vyksta tose pačiose vietose). Visų trijų atvejų patarėjo veiksmų lyginamosiose lentelėse galime išvelgti beveik idealų kiekvieno mokymo etapo sutapimą (nesutampa reikšmės, kai apmokyme dalyvavo trys ir penki mokiniai). Nesutampančios reikšmės 26, 27 lentelėse paryškintos.

26 lentelė Nesutapimai apmokyme dalyvaujant trimis mokiniams

Modifikacija \ sunkumo lygmuo	B1	B2	B3
kai $r_1 = 0,8$ ir $r_0 = 0,2$	A2	A2	A1
kai $r_1 = 100$ ir $r_0 = 0,01$	A2	A2	A1
kai $r_1 = 1$ ir $r_0 = -1$	<b>A1</b>	<b>A1</b>	A1

27 lentelė Nesutapimai apmokyme dalyvaujant penkiems mokiniams

Modifikacija \ sunkumo lygmuo	B1	B2	B3
kai $r_1 = 0,8$ ir $r_0 = 0,2$	A1	<b>A1</b>	A1
kai $r_1 = 100$ ir $r_0 = 0,01$	A1	A3	A1
kai $r_1 = 1$ ir $r_0 = -1$	A1	A3	A1

Pirmasis nesutapimas (kai apmokyme dalyvavo trys mokiniai) pasireiškia tik tame modelyje, kuriame skaičiavimams naudojama neigiama užduoties neįvykdymo premija. Šis nesutapimas gali atsirasti dėl ženklų kitimo skaičiavimo operacijų metu, kai operandai nuolat kaitaliojasi ženklais ir jų reikšmės būna artimesnės nuliui palyginus su kitais.



Antras nesutapimas (kai apmokyje dalyvavo penki mokiniai) pasireiškia pirmame modelyje, kai abi premijos yra iš intervalo  $r \in (0,1)$ ,  $r \in R$ . Šis nesutapimas gali kilti dėl sumavimo skaičiavimo operacijų.

Taip pat, analizuodami bet kurio iš trijų modelių  $Q(s, a)$  lentelių reikšmes, mes galime pamatyti, kurie stiliai yra naudojami tam tikrame uždavinių sunkumo lygmenyje (nenulinės reikšmės), kokią tendenciją turi tam tikras stilius. Jei stiliaus įvertis  $Q(s, a)$  lentelėje yra pakankamai mažas, lyginant su pirmo galimo apmokymo rezultatu, tai galime teigti, kad tas stilius yra neefektyvus (mokiniai neišsprendžia uždavinių ir, galbūt, reikėtų stilių modifikuoti). O jei didelis – tai galime daryti išvadą, kad stilius yra pakankamai aiškus, juo pateikiama informacija yra suprantama.

Remiantis lentelių duomenimis galime pamatyti, kurie stiliai nėra pateikiami (kai stilių reikšmės yra nulinės). Tai gali būti atsitiktinio uždavinių stiliaus „parinkėjo“ problema, kurią reikėtų spręsti realizuojant tokią mokymosi sistemą.

Iš dalies galima laikyti, kad pirmieji du modeliai (kai  $r_1 = 0,8$  ir  $r_0 = 0,2$  ir  $r_1 = 100$  ir  $r_0 = 0,01$ ) yra panašūs pagal savo veikimo principą ir jų duodami rezultatai prasminiu požiūriu yra beveik identiški.

Žemiau matome palyginimų lentelę (28 lentelė), kurioje pateikta e. patarėjo (po pilno nagrinėto apmokymo) siūlomas stilius kiekvieno sudėtingumo uždaviniams spręsti, uždavinius sprendusių mokinių statistika (kuris stilius palankiausias ir nepalankiausias tam tikro sunkumo uždaviniams spręsti) ir, remiantis atsitiktinai sugeneruota strategija (28 lentelė), palankiausias stilius uždaviniams spręsti kiekviename sunkumo lygmenyje.

28 lentelė. Stilių pasiūlos ir atitikmens palyginimų lentelė

	<b>E. patarėjo siūlomas stilius</b>	<b>Palankiausias stilius</b>	<b>Nepatogiausias stilius</b>	<b>Palankiausias stilius nagrinėtai strategijai</b>
B1	A1	A2, A3	A1	A2
B2	A1, A3	A2	A1, A2, A3	A2
B3	A1	A1, A3	A1, A3	A2

Stilių pasiūlos ir atitikmens palyginimų lentelėje (28 lentelė) matome, kad nei vienas e. patarėjo siūlomas stilius nėra idealus nagrinėtomis aplinkybėmis, nes pasitaiko, kad siūlomas stilius nėra palankiausias pagal sprendimų rezultatus arba patenka ir į siūlomų, ir į nepalankių stilių aibę. Tokie nesutapimai ir loginės klaidos gali būti dėl mažos tyrimo imties (mokymo procese dalyvavo tik penki mokiniai).

## 2.5 Apibendrinimas

Antrame skyriuje buvo apžvelgta e. mokymosi sistemų intelektualizavimo reikmė ir su ja susijusios problemos. Išanalizuotas Q-learning algoritmas bei apžvelgtos jo alternatyvos, naudojamos dirbtinio intelekto sistemose. Pagilintos žinios agentinių technologijų srityje ir agentų funkcionavimo srityje.

Atliekant Q-learning algoritmo parametrų analizę, skaičiavimai buvo atlikti neautomatizuotu būdu – skaičiuojant parankinėmis priemonėmis (popierius, pieštukas, kalkuliatorius). Todėl yra padidėjusi klaidos atsiradimo rizika dėl žmogiškojo faktoriaus. Siekiant kokybiškai iširti ir patikrinti matematinių skaičiavimų teisingumą yra rekomenduojama skaičiavimus atlikti skaitmeninėje terpėje, automatizuojant skaičiavimo procesus.

Gautus tyrimo rezultatus galima naudoti tolimesniuose tyrimuose, optimizuojant ir automatizuojant e. mokymosi sistemas.

### 3. PROBLEMINIŲ UŽDAVINIŲ SPRENDIMAS Q-LEARNING ALGORITMU

Tyrimo tikslas yra sukurti taikomąsias programas, kurios realizuotų antrame skyriuje nagrinėtą Q-learning algoritmą statinėje ir kintamosiose aplinkose. Šiam darbui atlikti buvo pasirinktas IBM genties nešiojamas kompiuteris ASUS, turintis Intel® Celeron® M procesorių 1.60 GHz ir 896 MB RAM. Operacinė sistema naudojama Microsoft Windows XP®. Programavimo kalba pasirinkta JAVA, kadangi ši kalba geba pati tvarkytis su nenaudojama atmintimi ir ją efektyviai paskirstyti vykdomajai užduočiai. Programavimo įrankiu pasirinkta nemokamo naudojimo JAVA *NetBeans* IDE v.6.5.1 aplinka, kadangi šioje aplinkoje yra patogiau kurti vartotojui priimtinas programas dizaino ir pateikimo atžvilgiu bei sutaupoma laiko, kuriant vizualius objektus. Objektų šabloninis kodas yra automatizuotai generuojamas darbinės aplinkos priemonėmis (Ryškus, 2004).

#### 3.1. Statinės aplinkos modelio analizė ir taikymas

Naudojantis JAVA programavimo kalbos teikiamomis galimybėmis galima sukurti daugkartinio naudojimo taikomąją programą su vartotojui patogia sąsaja. Tuo tikslu buvo sukurtas būsimos programos procesų šabloninis modelis (bendras veikimo planas), kuriuo bus vadovaujamasi tolimesniuose tyrimuose (23 paveikslas). Sąlyginiai perėjimai tarp programos procesų leidžia efektyviai kontroliuoti programos darbą ir, esant būtinybei, greitai bei nesunkiai keisti parametrų reikšmes.

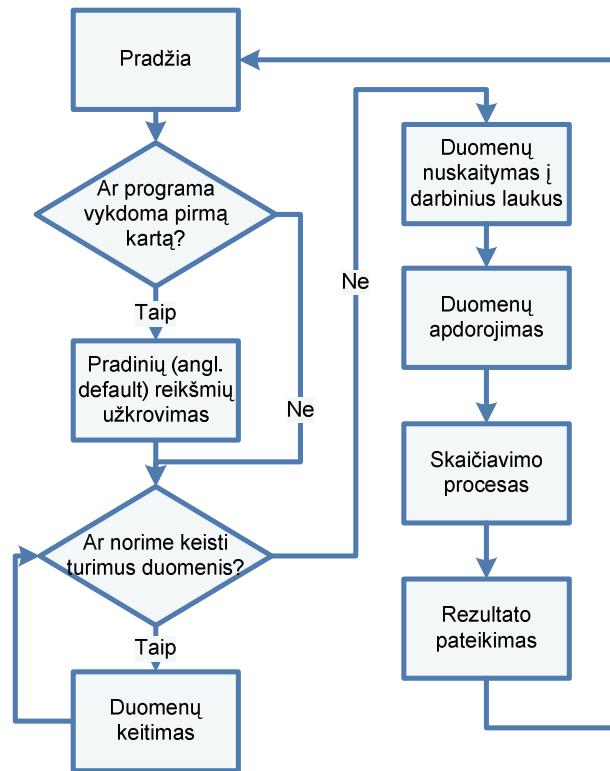
##### 3.1.1 Algoritmo inicializacija

Tyrimo objektu buvo pasirinktas statinės mokymosi aplinkos modelis. Modelis vadinamas statiniu, nes visos pradinės reikšmės yra žinomos dar prieš uždavinio sprendimą. Inicializacijos metu, pagal L. Ramašausko mokslo tiriamajame darbe (pirma dalis) nagrinėtus parametrus, sistema apibrėžia šias parametrų vertes (29 lentelė).

29 lentelė. Statinės aplinkos inicializuojamų parametrų reikšmės

Parametras	Vertė	Paaškinimas
$\alpha$	0,9	Mokymosi greičio koeficientas
$\lambda$	0,1	Pasitikėjimo būsenos vertėmis koeficientas
$r_0$	0,2	Apdovanojimas už neigiamą rezultatą
$r_1$	0,8	Apdovanojimas už teigiamą rezultatą
$Q(s, a)$	Visi matricos laukai lygus 0	Būsenų darbinė lentelė

23 paveikslas. Bendrinis programos procesų modelis



Strategijų užrašymui buvo pasirinktas masyvo duomenų tipas, kadangi dvimačiame masyve galima patogiai ir aiškiai keisti laukų reikšmes. Šiuo konkrečiu atveju buvo apibrėžtas dvimatis  $S(n, m)$  masyvas su trimis stulpeliais ir penkiomis eilutėmis, kur stulpeliai žymi konkrečias būsenas, o eilutės atitinka mokinius. L. Ramašausko pasiūlytą strategijų žymėjimą pakeičiame į skaičiavimams ir teksto apdorojimo operacijoms patogesnę atlikti stilių.

Jeigu mokinio B1 uždavinių sprendimo seka buvo A1, A3, tai masyve šie duomenys atvaizduojami tekstiniu formatu 1 ir 3. Paskutinis skaitmuo atitinka sėkmingo išsprendimo būseną. Tuomet turėti strategijų duomenys  $S(n, m)$  masyve atrodys taip (30 lentelė).

30 lentelė. Inicializuotas  $S(n, m)$  mokinių sprendimo strategijų masyvas

$S(n, m)$	<b>B1</b>	<b>B2</b>	<b>B3</b>
1-as mokinys	13	222	13
2-as mokinys	2	12	1
3-ias mokinys	113	131	21
4-as mokinys	32	2	2
5-as mokinys	121	23	33

Algoritmo fragmentas, realizuojantis pagrindinės  $Q(s, a)$  matricos inicializacijos procesą, pateiktas 31 lentelėje.

31 lentelė.  $Q(s, a)$  lentelės inicializacija nulinėmis vertėmis

```
String NullValue = "0";
    q11.setText(NullValue);
    q21.setText(NullValue);
    q31.setText(NullValue);
    q12.setText(NullValue);
    q22.setText(NullValue);
    q32.setText(NullValue);
    q13.setText(NullValue);
    q23.setText(NullValue);
    q33.setText(NullValue);
```

Šis etapas apsaugo programą nuo nepageidaujamų duomenų, kurie gali būti pasilikę nuo ankstesnių skaičiavimų, įvedimo klaidos ar vartotojo gebėjimo keisti laukų reikšmes. Inicializacija vykdoma iš karto, vykdant programą bei paspaudus mygtuką „**Inicializuoti Q(s, a)**“ programos vykdymo metu. Pastarasis veiksmas atitinka  $Q(s, a)$  lentelės pradinių reikšmių atkūrimo (angl. *Reset*) funkciją.

Vartotojui yra suteikta laisvė bet kada programos vykdymo metu keisti parametrų reikšmes, tačiau pirminis uždavinio tikslas yra išspręsti Q-learning uždavinį su inicializuotais parametrais.

Sekantis etapas yra pradinių strategijų inicializacija. Uždavinio sąlygos suformuotos strategijos yra vartotojui pateikiamos kaip pradinės vertės (angl. *Default Values*) (32 lentelė). Pastarąsias reikšmes vartotojas taipogi gali keisti savo nuožiūra.

32 lentelė.  $S(n, m)$  pradinių strategijų inicializacija

```
s11.setText("13"); s12.setText("222"); s13.setText("13");
s21.setText("2"); s22.setText("12"); s23.setText("1");
s31.setText("113"); s32.setText("131"); s33.setText("21");
s41.setText("32"); s42.setText("2"); s43.setText("2");
s51.setText("121"); s52.setText("23"); s53.setText("33");
```

Programa taip pat leidžia vartotojui gražinti pradines  $S(n, m)$  vertes ir išvalyti strategijas, atitinkamai paspaudus „**Pradinės strategijos**“ ir „**Valyti strategijas**“ pavadintus mygtukus.

### 3.1.2 Duomenų apdorojimo metodas

Inicializacijos ir pradinių duomenų keitimo fazės laikomos užbaigtomis, kai vartotojas nuspaudžia mygtuką „**Skaičiuoti**“. Tuomet pradami vidiniai skaičiavimo procesai, tiesiogiai siejantys Q-learning algoritmą ir rezultatus.

Kadangi programoje yra numatyta parametrų realaus laiko keitimo vykdymo metu galimybė, pradėjus skaičiavimų procesus pirmiausia sistema nusiskaito reikšmes į savo darbinės duomenų struktūras ir atlieka esminius duomenų tipų keitimo procesus (33 lentelė)

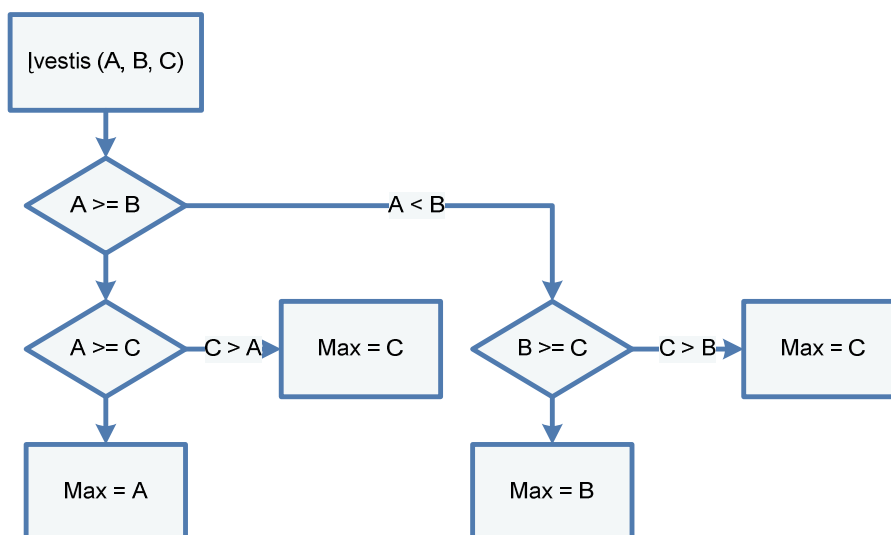
33 lentelė. Duomenų nuskaitymas į darbinės duomenų struktūras ir duomenų tipų keitimas

```
double q[][] = new double[3][3];  
q[0][0]=Double.parseDouble(q11.getText());  
...  
double lambda = Double.parseDouble(lambdaINPUT.getText());  
double r0 = Double.parseDouble(r0input.getText());
```

Vartotojo sąsajos aplinkoje visi duomenys yra laikomi tekstinio formato pavidalu arba simbolių eilutės tipo. Skaičiavimo operacijoms atlikti visus tekstinius duomenis reikia pakeisti į skaičiavimo operatoriams priimtinius duomenų tipus. Tai padaryti leidžia duomenų tipų suderinamumo atitinkami metodai (Ryškus, 2004).

Vienas iš pagrindinių ir pirminių Q-learning vykdymo dalių yra maksimalios reikšmės iš atitinkamo  $Q(s, a)$  lentelės regiono išskyrimo vidinis procesas (vykdomas duomenų apdorojimo proceso metu). Buvo panaudotas trijų sąlyginių perėjimų algoritmas maksimalios reikšmės išrinkimui (24 paveikslas) iš trijų pradinių įvesties  $A$ ,  $B$  ir  $C$  narių. Lyginimo operacijų metu yra išrenkama didžiausia reali reikšmė, kuri vėliau naudojama tolimesniuose skaičiavimo procesuose.

24 paveikslas. Maksimalios reikšmės išskyrimo algoritmas



Šis algoritmas uždavinio sprendime yra realizuotas (34 lentelė) būdu, kur skaičius  $A$  atitinka  $q[x][0]$ ,  $B - q[x][1]$  ir  $C - q[x][2]$  lentelės narius iš einamosios  $x$ -osios eilutės.

34 lentelė. Maksimalios realios  $Q(s, a)$  lentelės  $x$ -ojo regiono išrinkimo algoritmo realizacija

```
if (q[x][0] >= q[x][1])
{
    if (q[x][0] >= q[x][2])
    {
        max = q[x][0];
    }
    else
    {
        max = q[x][1];
    }
}
else
{
    if (q[x][1] >= q[x][2])
    {
        max = q[x][1];
    }
    else
    {
        max = q[x][2];
    }
}
```

Q-learning algoritmas yra iteratyvus. Pagrindiniai skaičiavimai atliekami tiek kartų, kiek buvo spęsta uždavinių. Skaičiavimams taikomas anksčiau apžvelgtas adaptuotas AIES Q-learning algoritmas. Programos kode  $Q[x][a]$  atitinka  $x+1$ -ojo mokinio sprendžiamo  $a+1$  stiliaus uždavinius  $Q$  lentelės reikšmė. Naudojamas  $x+1$  ir  $a+1$  indeksavimas, nes JAVA programavimo kalboje visi indeksai pradedami numeruoti nuo nulinės pozicijos. Tai reiškia, kad pirmasis mokinytis turės nulini indeksą, antrasis – pirmą, trečiasis – antrą,  $n$ -asis –  $n+1$ -ą (Ryškus, 2004). Paskutinio algoritmo kodas matomas programos fragmente (35 lentelė).

35 lentelė. Q-learning algoritmo kodo sąlygos sakinio pavyzdys

```
double suma=0;
if (ilgis == ii+1)
{
    suma = (1-alpha)*q[x][a]+alpha*r1+lambda*max;
    q[x][a]=suma;
}
else
{
    suma = (1-alpha)*q[x][a]+alpha*r0+lambda*max;
    q[x][a]=suma;
}
```

### 3.2 Mišrios aplinkos modelio realizavimo programa

Šios kuriamos programos veikimo modelis nesiskiria nuo 3.1 skyriuje sudarytos programos, nes naudoja tuos pačius pagrindinius procesus, kurie yra pavaizduoti 23 paveiksle. Programos modelis vadinamas „mišrios aplinkos modeliu“, nes jis yra statinės ir dinaminės aplinkų sintezės rezultatas. Dinamiškumą pažymi kaskart skirtingų atsitiktinių strategijų generavimas, vartotojo galimybė keisti mokinių skaičius ir strategijų maksimalaus ilgio vertes. Statiškumą pažymi dinamiškai sugeneruotų ar įvestų reikšmių naudojimas atliekant skaičiavimus.

Esminiai skirtumai nuo pirmutinės programos yra šie:

- kintami mokinių ir strategijų ilgio skaičiai;
- atsitiktinis strategijų generavimas;
- atliekama minimali įvedamų parametrų kontrolė;
- informatyvesnė vartotojo sąsaja.

Toliau pateikiami šių skirtumų realizavimo ypatumai.

#### 3.2.1 Skirtumų apžvalga ir realizavimas

Statiniam modeliui sukurtas papildomas laukas, leidžiantis vartotojui keisti dalyvaujančių mokymo procese mokinių skaičių ir didžiausią išsprendimo strategijos ilgį (25 paveikslas).

25 paveikslas. Papildomi programos įvesties laukai

Dalyvaujančių mokinių mokyme skaičius	<input type="text" value="5"/>
Išsprendimo strategijos maksimalus kelias	<input type="text" value="5"/>



Šiems laukams yra suteikiamos pradinės reikšmės, lygios 5 (36 lentelė). Siekiant išvengti didelių skaičiavimų, buvo įvesti apribojimai, kad maksimalus mokinių skaičius negali viršyti 1000 ir laikoma, kad mokinys iš dešimto karto tikrai išsprendžia jam pateiktą užduotį (gali ją išspręsti ir greičiau).

36 lentelė. Pradinių reikšmių nustatymas naujiems programos laukams

```
jLabel13.setText("Dalyvaujančių mokinių mokyme skaičius");
    mokSK.setText("5");
...
jLabel15.setText("Išsprendimo strategijos maksimalus kelias");
    maxSTRkelias.setText("5");
```

### 3.2.2 Atsitiktinis strategijų generatorius

Siekiant įvairovės ir skirtingų pradinių duomenų bei jų generavimo automatizavimo, buvo sukurtas elementarus skaičių generatorius (37 lentelė), kuris generuoja atsitiktinius sveikus skaičius nuo 1 iki maksimalaus galimo strategijos išsprendimo ilgio. Generavimas vykdomas tiek kartų, kiek reikia sukurti strategijų.

37 lentelė. Strategijos kelio ilgio generatorius

```
int maxkelias = Integer.parseInt(maxSTRkelias.getText());
...
int result = 1 + (int)(Math.random()*(maxkelias-1)+0.5);
```

Žinant strategijos ilgį yra panaudojamas antras atsitiktinis skaičių generatorius, kuris sugeneruoja išsprendimo strategijos kelią (38 lentelė). Šis generatorius generuoja strategijos narius, kurie yra iš intervalo nuo 1 iki 3. Šie režiai yra nustatyti sprendžiamo uždavinio sąlygoje. Šį uždavinį būtų galima tobulinti, įvedant kintamo ilgio išsprendimo laukus (pavyzdžiui, kai mokymo procese galima pasirinkti galimų sunkumo lygių skaičių), tačiau šiuo atveju, paprastumo ir aiškumo dėlei, buvo pasirinktas statinis laukas su nekintančiu laukų skaičiumi.

38 lentelė. Išsprendimo strategijos kelio atsitiktinis generatorius

```
String seka="";
for (int lstr=1;lstr<=result;lstr++)
{
    String fin = "";
    int finINT = 1 + (int)(Math.random()*(2.5));
    fin = Integer.toString(finINT);
```

```
seka = seka + fin;  
}
```

### 3.2.3 Programos kontrolė ir informatyvesnė vartotojo sąsaja

Programa buvo papildyta minimalia įvedamų parametrų kontrole. Kadangi programa yra „jautri“ įvedamų laukų reikšmėms (t. y. į laukus galima įvesti visus galimus simbolius), įvedus nekorektiškus parametrus programa neveiks. Pastaroji, kaip ir pirmoji programa, yra skirta atidžiam vartotojui, kuris įveda tik korektiškas reikšmes. Siekiant sumažinti galimos įvedimo klaidos tikimybę, programa buvo papildyta nauja (mokinių skaičiaus ir maksimalios išsprendimo strategijos ilgio) įvesties parametrų tikrinimo funkcija. Paspaudus mygtuką „**Tikrinti parametrus**“, programa patikrina, ar įvesti parametrai yra programos numatytuose režiuose. Jei programa aptinka rėžių peržengimą, ji informuoja vartotoją apie neatitikimus informacijos pateikimo lange (26 paveikslas).

26 paveikslas. Tikrinamų parametrų ataskaitos pateikimas vartotojui

Dalyvaujančių mokinių mokyme skaičius  Sugeneruotos atsitiktinės strategijos pagal pateiktus parametrus

Išsprendimo strategijos maksimalus kelias

**Tikrinti parametrus**

Inicializuoti Q(s,a)

Generuoti strategijas

Parodyti strategijas

Valyti strategijas

**SKAIČIUOTI**

Informacija: Nekorektiskas mokiniu skaicius (maziau uz viena). Patikrinkite.  
Nekorektiskas maksimalus kelio skaicius (daugiau nei 10). Patikrinkite.

## 3.3 Sukurtų programų duomenų ir realizavimo analizė

### 3.3.1 Programos v100 rezultatų palyginimas

Remiantis sudarytu bendriniumi programos procesų modeliu ir sukurtaisiais algoritmais, parašyta programa v100, kuri pagal duotus parametrus apskaičiuoja  $Q$  lentelės reikšmes. Šios programos tikslas buvo patikrinti 2 skyriuje gautą skaičiavimų tikslumą, palyginti rezultatus ir sukurti automatizuotą aplinką, kuri būtų patogi vartotojui ją valdyti ir sekti rezultatus. Pradinis programos darbo langas matomas 27 paveiksle.

27 paveikslas. Pirminės programos v100 pradinis vaizdas

Atlikus skaičiavimus su statiniais pradiniais duomenimis, gaunamas rezultatas yra atvaizduojamas atitinkamuose programos laukuose ir matomas 28 paveiksle. Gautos  $Q$  lentelės reikšmės mažai skiriasi nuo antrajame skyriuje apskaičiuotų rezultatų (39 lentelė). Skirtumai tarp rezultatų atsiranda dėl to, kad skaičiuojant  $Q$  lentelės reikšmes primityviai, nenaudojant kompiuterinių įrankių, buvo apvalinama penkių skaičių po kablelio tikslumu, o panaudojus kompiuterines technologijas – imamos pakankamai tikslios reikšmės. Didesni skirtumai galėjo atsirasti dėl žmogiškojo faktoriaus klaidos, skaičiuojant primityviu (popierius + pieštukas) būdu. Šiuo atveju programos pateikiami rezultatai yra tikslesni ir tikėtina, kad korektiški. Palyginimui pateikiami programos ir antro skyriaus skaičiavimų,  $Q$  lentelių galutiniai rezultatai, gauti naudojant statinius duomenis.

39 lentelė. Programos ir primityvaus skaičiavimų rezultatų palyginimas

Programos rezultatai				Primityvaus skaičiavimo rezultatai			
$Q(s, a)$	A1	A2	A3	$Q(s, a)$	A1	A2	A3
B1	0,7848018	0,3528	0,35352	B1	0,779668	0,345184	0,344232
B2	0,8365536	0,3589056	0,83038176	B2	0,826076	0,348011	0,745781
B3	0,8676	0,83214	0,840816	B3	0,87281	0,767952	0,831765

## 28 paveikslas. Q-learning algoritmo veikimo rezultatas su antrojo skyriaus duomenimis

Programa V1.0.0 - Statinio modelio realizacija

Programa realizuoja pritaikytą Q-learning algoritmą statinėje aplinkoje, kai įvesties parametrai yra žinomi iš anksto.

Norėdami atlikti skaičiavimus, paspauskite mygtuką "SKAIČIUOTI".

Vartotojas bet kuriuo metu gali keisti ar grąžinti senas įvesties parametrų reikšmes.

Programą sukūrė: LUKAS RAMAŠAUSKAS, VU KF, Informacijos sistemų vadyba, 2010 Vilnius.

Alpha=	<input type="text" value="0.9"/>	Veiksmas a1	Veiksmas a2	Veiksmas a3	
Lambda=	<input type="text" value="0.1"/>	Būsena B1	<input type="text" value="0.78480180000000001"/>	<input type="text" value="0.35280000000000006"/>	<input type="text" value="0.35352"/>
r0=	<input type="text" value="0.2"/>	Būsena B2	<input type="text" value="0.83655360000000001"/>	<input type="text" value="0.3589056"/>	<input type="text" value="0.83038176000000001"/>
r1=	<input type="text" value="0.8"/>	Būsena B3	<input type="text" value="0.8676"/>	<input type="text" value="0.83214000000000001"/>	<input type="text" value="0.840816"/>

	B1 stilius	B2 stilius	B3 stilius
1-as mok.	<input type="text" value="13"/>	<input type="text" value="222"/>	<input type="text" value="13"/>
2-as mok.	<input type="text" value="2"/>	<input type="text" value="12"/>	<input type="text" value="1"/>
3-ias mok.	<input type="text" value="113"/>	<input type="text" value="131"/>	<input type="text" value="21"/>
4-as mok.	<input type="text" value="32"/>	<input type="text" value="2"/>	<input type="text" value="2"/>
5-as mok.	<input type="text" value="121"/>	<input type="text" value="23"/>	<input type="text" value="33"/>

### 3.3.2 Eksperimento paklaidų įvertinimas

Algoritmo kokybės įvertinimui atliktas eksperimentas (Eidukas, 2002). Programos v100 bandymų metu gautųjų rezultatų kiekybiniai įverčiai (bendrieji skirtumai, nuokrypų absoliučios reikšmės) pateikti 40 lentelėje.

40 lentelė. Absoliučių skirtumų tarp rezultatų šaltinių lentelė

$Q(s, a)$	A1	A2	A3
B1	0,0051338	0,007616	0,009288
B2	0,0104776	0,010945	<b>0,08460076</b>
B3	0,00521	<b>0,064188</b>	0,009051

Iš skirtumų (40) lentelės duomenų matyti, kad tik du laukai viršija daugiau nei dviejų procentų netikslumą (lentelėje pažymėti ryškesniu šriftu) ir tik du laukai nežymiai peržengia vieno procento skirtumą (lentelėje pažymėti pasvirusiu šriftu). Tai leidžia pasitikėti primityvaus (antrajame skyriuje) skaičiavimo gautais apytiksliais rezultatais ir vadovautis programos pateikiamomis  $Q$  lentelės tiksliais reikšmėmis.

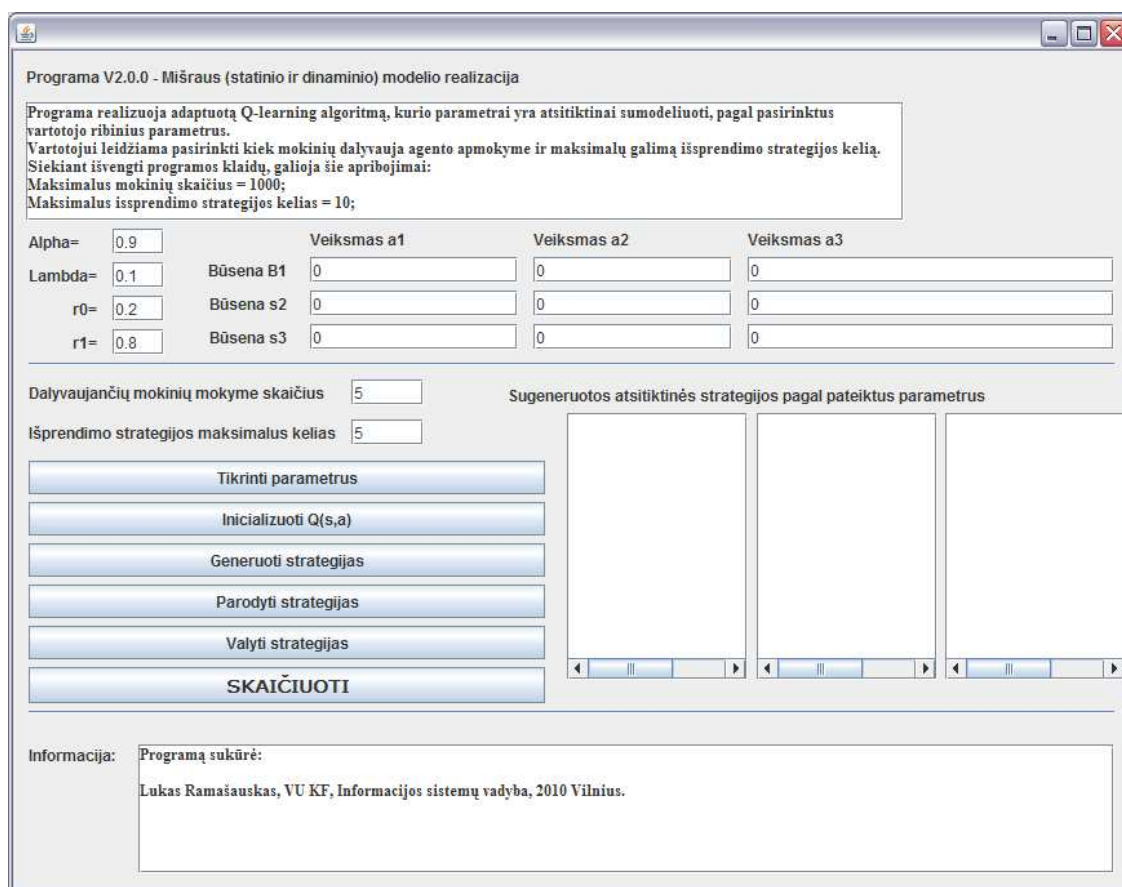
### 3.3.3 Programos v200 aprašymas

Sukurtoji programa yra dinamiškesnė ir profesionalesnė, nei anksčiau apžvelgtoji v100, kadangi gali atlikti sudėtingesnius skaičiavimus, gali modeliuoti vis skirtingas strategijas bei apmokymo aplinkas. Taip pat programa atlieka ir minimalų parametrų tikrinimą, siekiant išvengti neko-  
rektiškų įvesties parametrų. Siekiant minimalios kontrolės ir apsaugos nuo neko-  
rektiškų įvesties duomenų, programa buvo papildyta naujomis funkcijomis, kurios tikrina pradinis duomenis ir, įvykus incidentui, informuoja vartotoją.

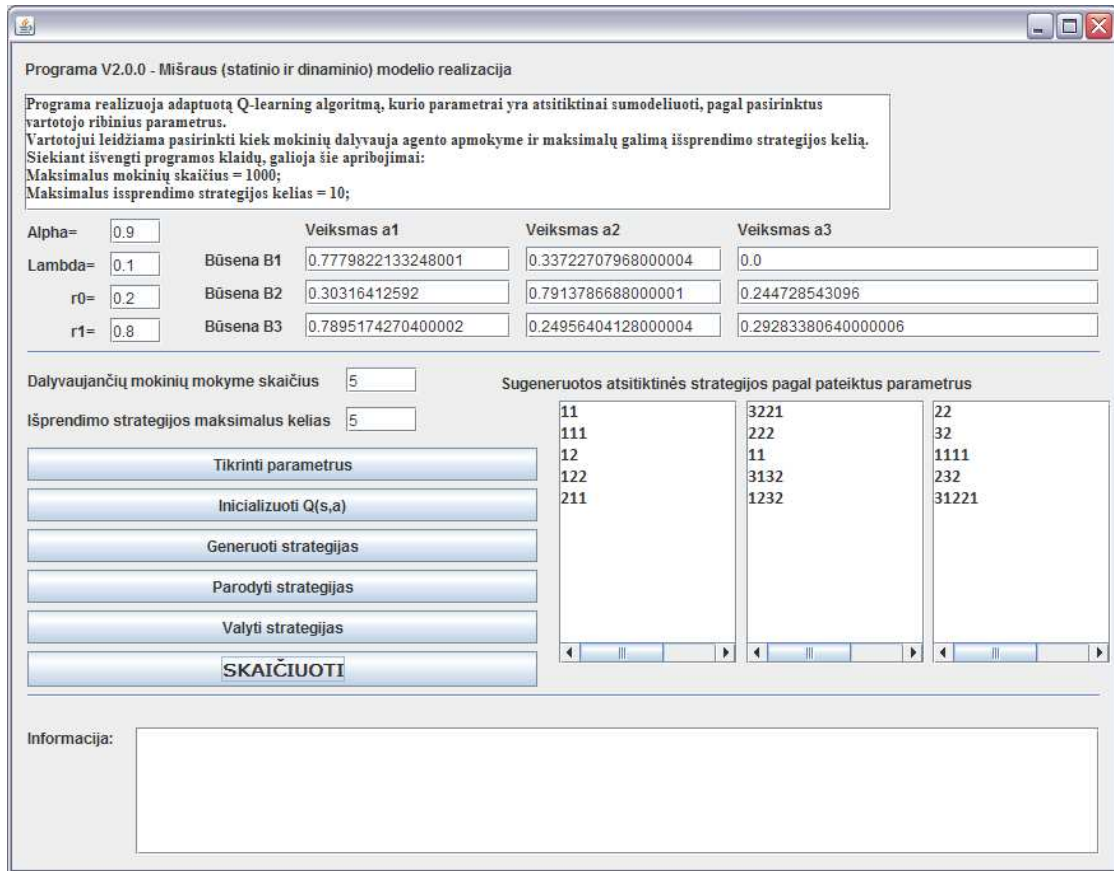
Vartotojui paliktos teisės keisti parametrų reikšmes, jas pakeičiant kitomis. Vienas iš esminių privalomųjų vartotojo žingsnių yra tai, kad vartotojas privalo pasinaudoti strategijų generavimo funkcija, norėdamas toliau dirbti su programa. Jei jį domina gautosios strategijos, jas galima pamatyti, paspaudus mygtuką „**Parodyti strategijas**“. Jei vartotojas yra nepatenkintas strategijomis, jis gali jas išvalyti, modifikuoti ar tiesiog sugeneruoti naujas.

Derinant programą, lygiagrečiai buvo atliekamas skaitmeninis eksperimentas. Pradinis sukurtosios programos darbinio lango vaizdas matomas 29 paveiksle, o atsitiktinai parinkto programos pirmosios iteracijos įvykdymo rezultatai – tame pačiame darbo lange – parodytas 30 paveiksle.

29 paveikslas. Programos v200 darbo langas eksperimento pradžioje



### 30 paveikslas. Programos pirmosios iteracijos įvykdymo rezultatai



### 3.4 Apibendrinimas

Šiame skyriuje buvo atliktas skaitmeninis eksperimentas, kurio rezultatai akivaizdžiai parodė sukurtųjų ir analitiškai įvertintų VMA algoritmų ir komunikacinių modelių gyvybingumą, remiantis teoriniais mokslinių tyrimų analogijos ir ekstrapoliacijos metodais. Viso darbo rezultatų apibendrinimui pateikiama 41 lentelė, patvirtinanti tyrimo hipotezę ir leidžianti suformuluoti darbo išvadas, kuriose matyti pasiektas darbo tikslas ir išspręsti uždaviniai.

Tyrimo metu buvo detaliam susipažinta su Q-learning algoritmo techniniais veikimo žingsniais ir praktine jų realizacija JAVA programavimo kalba. Sukurtos dvi programos, kurios realizuoja Q-learning algoritimą naudojančius agentus skirtingose mokymosi aplinkose. Programa v100 (bylos pavadinimas *versija100.java*) imituoja statinės mokymosi aplinkos modelį, o programa v200 (bylos pavadinimas *versija200.java*) imituoja agento veikimą mišrioje mokymosi aplinkoje. Pastaroji programa turi daugiau galimybių imituoti skirtingus veikimo atvejus be papildomo vartotojo įsikišimo. Buvo palyginti antrojo skyriaus darbo rezultatai su programos v100 rezultatais ir esminių skirtumų nebuvo pastebėta, o tai leidžia manyti, kad programa dirba korektiškai. Kadangi programa v200 yra

sukurta programos v100 pagrindu, tranzityviai sprendžiame apie programos v200 teikiamų rezultatų korektiškumą.

41 lentelė. Modelių su dirbtinio intelekto įskiepiams taikymo VMA tyrimuose rezultatai

Modelio tipas	Panaudota darbe
Verbalinis (žodinis, vaizdinis) modelis yra aprašomasis modelis, leidžiantis neformaliai išreikšti žmogaus idėją arba požiūrį. Toks modelis būna pradinis informacijos šaltinis, tiriant žmogiškojo pažinimo procesus.	Verbalinis modelio aprašymas panaudotas formuluojant naujus VMA algoritmus su dirbtinio intelekto (agentiniais, multiagentiniais programiniais) įskiepiams.
Natūrinis (fizinis, materialus) modelis realiai demonstruoja tam tikras sistemos savybes. Tokio modelio kūrimas ir taikymas reikalauja natūrinio eksperimento ir modeliavimo metodikų derinimo, todėl patys modeliai užima tarpinę vietą tarp teorinių ir eksperimentinių metodų.	Kuriami VMA objektai, kuriuos pagal jų tikslų aprašymą generuoja ir pavaizduoja darbinis arba paslaugų kompiuteris. Nustatyta, kad tokį natūrinį modelį panaudoti neefektyvu, nes daug skaičiavimų reikia atlikti "rankomis".
Simbolinis modelis išreiškiamas tam tikros formaliosios kalbos priemonėmis. Tai aukščiausias žinių apibendrinimo lygmuo, nes juose žinių pateikimo forma leidžia abstrahuotis nuo konkretaus turinio. Vienas abstraktus ženklinis aprašas apibendrina didelę galimų sistemos sudėties ir elgsenos variantų įvairovę.	VMA tyrimams ir agentiniam intelektualizavimui sukurtieji algoritmai įgyvendinti Java programavimo kalba (JDK).

Darbe apžvelgtus matematinius modelius ir jų algoritmų realizacijų tyrimus galima tęsti, siekiant išsiaiškinti sistemoje vykstančių Markovo procesų subtilybes ir optimaliai panaudoti jų matematinės savybes. Taip pat būtų naudinga išsiaiškinti daugiaagentinių mokymosi aplinkų architektūros problemas ir patobulinti pačių aplinkų programinio realizavimo būdus, kadangi jos tampa svarbiu šiuolaikinių elektroninio mokymosi technologijų komponentu ir turi neabejotinas pritaikymo perspektyvas Lietuvos švietimo ir mokymo įstaigose.

## IŠVADOS

1. Atlikus nuotolinio mokymosi sistemų ir jų aplinkų mokslinę analizę ir išnagrinėjus informacijos bei žinių vadybinius aspektus sudarytas tose sistemose vykstančių informacijos ir žinių kaitos procesų modelis. Sukurti algoritmų pavyzdžiai, leidžiantys algoritmuoti tiek paprastas (vieno agento), tiek sudėtingas (daugelio agentų, multiagentines) e. mokymosi sistemas, naudojančias sukurtojo informacijos ir žinių kaitos modelio elementus. Nustatyti e. mokymosi sistemų naudojimo ir realizavimo privalumai ir trūkumai.

2. Ištirtos naujos algoritmų su dirbtinio intelekto įskiepiams panaudojimo galimybės, taikant sukurtąjį konceptualų informacijos ir žinių kaitos procesų modelį e. mokymuisi, taip pat agentinėse e. mokymosi sistemose. Išanalizuotas Q-learning algoritmas bei apžvelgtos jo alternatyvos, naudojamos dirbtinio intelekto sistemose bei algoritmo emuliacijos metu gauti rezultatai leido triviale būdu pasiekti realaus e. mokymosi sistemos elemento – e. patarėjo atliekamo darbo gerą rezultatą.

3. Atliktas mokslinis – tiriamasis darbas leido sukurti naujas vartotojui patogias taikomas programas, kurios emuliuotų agentų dirbtinio intelekto vykdomus skaičiavimus e. mokymosi sistemose ir pagerintų mokinio ir sistemos komunikaciją. Gauti tiriamojo darbo rezultatai leido nustatyti potencialias skaičiavimo netikslumų galimybes sukurtojo ir įgyvendinto modelio realizacijoje bei leido išvelgti sukurtųjų modelių pritaikomumą realiose e. mokymosi aplinkose.

4. Apibendrinus rezultatus ir įvertinus atlikto darbo mokslinių tyrimų apimtį matyti, kad mokslinis – tiriamasis darbas gali būti tęsiamas ir gilinamas tiek informacijos ir žinių vadybos analizės aspektais e. mokymosi aplinkų valdymo srityje, tiek dirbtinio intelekto įskiepių optimizavimo ir tobulinimo kryptimis e. mokymosi sistemose.



## SANTRAUKA ANGLŲ KALBA

One of the branches of E-Learning is Reinforcement Learning, whose origins go back to cybernetics, statistics, psychology, neuroscience and computer science intersections, based on Intelligent Multiagent Software action in a dynamic environment. Modern E-Learning systems must react to the challenges and meet up difficulties associated with human-computer interaction and communication problems. Therefore is important to explore application of Artificial Intelligence in the new E-Learning systems.

The aim of this work is to explore the use of artificial intelligence implementations for E-Learning systems and increase its opportunities hereby introduce them to a multiagent E-Learning environment and to create user-friendly applications emulating the calculations and summarize the results. Presented work describes the new research of E-Learning and AI algorithms provided with a clear scientific value and the actual practical application, improving the training and education problems, using created system modules saves student time by selecting the optimal presentation materials.

The results of research topics of this work are relevant now and possibly will be even more important in the future.

## LITERATŪROS SĄRAŠAS

- ABARIUS, P.** *Dirbtinio intelekto metodų naudojimas adaptyviose mokymosi sistemose*. E. Švietimas: mokslas, studijos ir verslas: konferencijos pranešimų medžiaga. Kaunas, 2010, p. 40 – 44 .
- AMANDI, A. and others.** *Intelligent Agents for Distance Learning*. Informatics in Education, Institute of Mathematics and Informatics. Vilnius, 2003, vol. 2, no. 2, p. 161 – 180.
- ATKOČIŪNIENĖ, Z.** *Informacijos ir žinių vadyba informacijos ir komunikacijos mokslų sistemoje*. Informacijos mokslai (37), Vilnius, 2006, 22 - 29 p., ISSN 1392 – 0561.
- BALBIERIS, G., et all.** *Virtualioji mokymosi aplinka mokyklai [interaktyvus]*. [Vilnius]: 2005 [žiūrėta 2011 m. balandžio 22 d.]. Prieiga per internetą:  
<<http://mtp.smm.lt/dokumentai/informacijasvietimui/metodinesrekomendacijos/200506VMA.doc>>.
- BAZIUKAITĖ, D.** *Concepts of adaptive based virtual learning environment*. Proceedings of the International Conference held at Kaunas University of Technology, Lithuania, 2003, 8 – 9 May, p. 63 – 66.
- BAZIUKAITĖ, D.** *Learner Oriented Methods to Enhance Capabilities of Virtual Learning Environment*. Doctoral dissertation, Physical Sciences, Informatics (09P). Klaipėda, 2007.
- BERRY, M.** *Knowledge management in education and the contribution of Virtual Learning Environments*. December 2005, published in Computer Education 112, Spring 2006 NAACE, 2005.
- BIELSKIS, A. A. ir VAITKUS, M. E.** *patarėjo mokymuisi intelektualizacija*. Vadyba: mokslo tiriamieji darbai, Nr.1(4). Klaipėda: Vakarų Lietuvos verslo kolegija, 2004, 15 – 20 p., ISSN 1648 – 7974.
- BOTA, F. and others.** *A Framework for Student Knowledge Evaluation in Internet Environments*. Studies in Informatics and Control [interaktyvus]. March 2005 [žiūrėta 2011 m. balandžio 10 d.]. vol. 14, no. 1. Prieiga per internetą: < [http://sic.ici.ro/sic2005\\_1/art1.pdf](http://sic.ici.ro/sic2005_1/art1.pdf)>.
- BRUSILOVSKY, P.** *Adaptive and Intelligent Technologies for Web-Based Education*. Künstliche Intelligenz, 1999, Nr. 4, p. 19 – 25.
- BRUSILOVSKY, P.** *Adaptive hypermedia: User Modeling and User Adapted Interaction*. Ten Year Anniversary Issue (Alfred Kobsa ed.), 2004, 11 (1/2), p. 87 - 110.
- DENISOVAS, V.** *Modeliavimas dalykų dėstyje*. Informatika, Nr.2 (36). Vilnius: Matematikos ir informatikos institutas, 2000, p. 5 – 35.

- DILLENBOURG, P.** *Virtual learning environments*. Eun conference 2000: learning in the new millennium: Building new education strategies for schools [interaktyvus]. 2000 [žiūrėta 2001 m. kovo 28d.]. Prieiga per internetą: <<http://tecfa.unige.ch/tecfa/publicat/dil-papers-2/Dil.7.5.18.pdf>.>
- EIDUKAS, D.** Eksperimento planavimo teorija. Kaunas: Technologija, 2002, p. 138.
- ELSKYTĖ, V.** *Žinių vadyba: paskaitos skaidrės* [interaktyvus]. Vilniaus Gedimino technikos universitetas, Verslo technologijų katedra, 2010 [žiūrėta 2011 m. sausio 20 d.]. Prieiga per internetą: <[www.vgtu.lt/upload/vyf\\_vtk/ziniu%20vadyba.ppt](http://www.vgtu.lt/upload/vyf_vtk/ziniu%20vadyba.ppt)>.
- FELDER, R. M.** *Reaching the second tier: learning and teaching styles in college science education*. Journal of College Science Teaching, 1993, 23 (5), p. 286 – 290.
- GALEEV, I. and others.** *Problems of building adaptive integrated learning environments Advanced Learning Technologies*. Proceedings. The 3rd IEEE International Conference, July 2003.
- GOLOUCHOVA, J.** *Neprižiūrimų mokymosi algoritmų efektyvumo ir jais gaunamų klasterių pastovumo tyrimas*. Fundamentiniai tyrimai ir inovacijos mokslų sandūroje: respublikinės jaunujų mokslininkų konferencijos darbai. Klaipėda: Klaipėdos universiteto leidykla, 2008, 16 – 20 p., ISBN 978-9955-18-329-7.
- GUBAIDULIN, I. ir BAZIUKAITĖ, D.** *Besimokančiųjų rezultatų diskretizavimo virtualioje mokyimo(si) aplinkoje posistemė: bakalauro baigiamasis darbas*. Klaipėdos universitetas, Klaipėda, 2007.
- HAAPALA, A.** *Promoting Different kinds of Learners towards Active Learning in the Web – Based Environment*. Informatics in Education, vol. 5, no. 2, Vilnius, 2006, p. 207 – 218.
- IGLESIAS, A. and others.** *An Experience Applying Reinforcement Learning in a Web-Based Adaptive and Intelligent Educational System*. Informatics in Education, vol. 2, no. 2, Vilnius, Institute of Mathematics and Informatics, 2003, p. 223 – 240.
- IGLESIAS, A. and others.** *Learning to teach database design by trial and error*. In 4th International Conference on Enterprise Information Systems. Ciudad Real, Spain, 2002. p. 500 – 505.
- KAEBLING, L. and others.** *Reinforcement learning: a survey*. International Journal of Artificial Intelligence Research, 1996, p. 237 – 285.
- LAKAMI, C. Jain ir MARTIN, N. M.** *Fusion of Neural Networks, Fuzzy Systems and Genetic Algorithms: Industrial Applications*. CRC Press, 1998, 297 p. ISBN 0849398045.
- LIPEIKIENĖ, J.** *Virtual Learning Environments as a Supplement to Traditional Teaching*. Informatics in Education, vol. 2, no. 1. Vilnius, 2003, p. 53 – 64.
- MARKEVIČIŪTĖ, L.** *Informacijos vadybos aprėptys ir sąsajos*. Informacijos mokslai (44). Vilnius, 2008, p. 58 – 77, ISSN 1392 – 0561.

- MEERTS, J.** *Course Management Systems (CMS) EDUCAUSE: Evolving Technologies Committee*. Institution: Wesleyan University [interaktyvus]. [žiūrėta 2011 m. balandžio 19 d.]. Prieiga per internetą: < <http://net.educause.edu/ir/library/pdf/DEC0302.pdf> >.
- MURRAY, T.** *Authoring Intelligent Tutoring Systems: an analysis of the state of the art*. International Journal of Artificial Intelligence in Education, Nr. 10, 1999, p. 98 – 129.
- NIKJE, H. and others.** *Automatic learning of domain model for personalized hypermedia applications*. IJCAI-05, 2005 [interaktyvus]. [žiūrėta 2011 m. balandžio 2 d.]. Prieiga per internetą: < <http://ijcai.org/papers/post-0282.pdf> >.
- PARAMYTHIS, A. and others** *Adaptive Learning Environments and e-Learning Standards*. Proceedings of the 2nd European Conference on e-Learning (ECEL2003), Glasgow, Scotland, 6 – 7 November, Academic Conferences International Reading, 2003.
- RAMAŠAUSKAS, L.** *Dirbtinio intelekto sistemų naudojimas e. mokymosi sistemose: mokslo tiriamasis darbas* (Vadovas: Doc. dr. Povilas Abarius). Vilniaus universitetas. Vilnius, 2010.
- RAMAŠAUSKAS, L.** *Paslėptųjų Markovo grandinių modeliai bioinformatikoje ir jų tyrimas: bakalauro baigiamasis darbas*. Vilniaus universitetas, Matematikos ir informatikos fakultetas. Vilnius, 2009, p. 35.
- RAUDELĪUNIENĖ, J.** *Žinių vadyba (ŽV): samprata: paskaitos skaidrės*. Vilniaus Gedimino technikos universitetas [interaktyvus]. 2010 [žiūrėta 2011 m. sausio 24 d.]. Prieiga per internetą: < [http://e-stud.vgtu.lt/users/files/dest/8002/1\\_zv.pdf](http://e-stud.vgtu.lt/users/files/dest/8002/1_zv.pdf) >.
- RAUDYS, Š.** *Žinių išgavimas iš duomenų: vadovėlis*. Klaipėda: Klaipėdos universiteto leidykla, 2008, 170 p. ISBN 978-9955-18-345-7.
- RIČKUTĖ, L.** *Nuotolinio mokymosi intelektualizavimo galimybė naudojant testus atviro kodo sistemoje*. Vadyba: mokslo tiriamieji darbai, Nr. 1(16). Klaipėda: Vakarų Lietuvos verslo kolegija, 2005, p. 178 – 186, ISSN 1648-7974.
- RYŠKUS, A.** *Programavimas JAVA: pirmoji pažintis*. 2 – asis leid. Kaunas: Technologija, 2004, ISBN 9955-09-274-5.
- RUTKAUSKIENĖ, D., ABARIUS P., et all.** *Nuotolinis mokymasis: mokomoji knyga*. Kaunas: Technologija, 2003, 255 p., ISBN 9955-09-321-8.
- RUTKAUSKIENĖ, D., et all.** *Nuotolinio mokymosi dėstytojo vadovas: mokomoji knyga*. Kaunas: Technologija, 2007, 188 p., ISBN 9955-25-190-5.
- RUŽEVIČIUS, J.** *Verslo informacijos kokybės vertinimas*. Informacijos mokslai (40), Vilnius, 2007, 47 - 56 p., ISSN 1392-0561.
- SUTTON, R. S. ir BARTO, A. G.** *Reinforcement learning: An introduction*. MIT Press, Cambridge, MA, A Bradford Book, 1998, p. 334.

**TANKELEVIČIENĖ, L.** *Characteristics of Domain Ontologies for Web based Learning and their Applications for Quality Evaluation*. Informatics and Education, vol. 8, no.1. Vilnius, 2009, p. 131 – 152.

**UBON, A. N., and others.** *Knowledge Management in Online Distance Education*. Proceedings of the 2002 International Conference: A Research Based Conference on E-Learning in Higher Education and Lifelong Learning. Sheffield: University of Lancaster and Sheffield

**VASILECAS, O.** *Informacinių sistemų kūrimo metodai ir technikos*. Klaipėda: Klaipėdos universiteto leidykla, 2008, 252 p., ISBN – 978-9955-18-344-0.

**VASILYEVA, E., and others.** *Knowledge Management Challenges in Web-Based Adaptive E-Learning Systems*. Proceedings of I-KNOW '05 Graz, Austria, 2005, June 29 – July 1.

**VIRVOU, M.** *Adaptivity and User Modeling Research at the Department of Informatics*. University of Piraeus, SMC Electronic Newsletter [interaktyvus]. 2011 [žiūrėta 2011 m. kovo 20 d.]. Prieiga per internetą:

[http://ieeesmc.org/announcements/newsletter/Sep2004/SMC\\_UnivOfPiraeus.pdf](http://ieeesmc.org/announcements/newsletter/Sep2004/SMC_UnivOfPiraeus.pdf).

**WATKINS, Ch. ir DAYAN, P.** *Technical Note: Q – Learning*. Machine Learning, 8, Boston, Netherlands, 1992, p. 279 – 292.

**WEISS, G.** *Multiagent Systems: A Modern Approach to Distributed Modern Approach to Artificial Intelligence*. MIT Press, London, 1999, 619 p., ISBN 0-262-23203-0.

## PRIEDAI

Elektroninė darbo dokumentų ir programų laikmena (CD) – 1 vnt.

Laikmenos turinys:

1. md\_lr2011.pdf 1,23 Mb.
2. versija100.java JAVA programa (pirmoji).
3. versija200.java JAVA programa (antroji).