

VILNIAUS UNIVERSITETAS

JŪRATĖ URBONIENĖ

ADAPTYVIŲJŲ PROGRAMAVIMO MOKYMO PRIEMONIŲ
PROJEKTAVIMAS

Daktaro disertacija

Technologijos mokslai, informatikos inžinerija (07 T)

Vilnius, 2014 metai

Disertacija rengta 2009 – 2013 metais Vilniaus universiteto Matematikos ir informatikos institute

Mokslinė vadovė:

prof. dr. Valentina Dagiienė (Vilniaus universitetas, technologijos mokslai, informatikos inžinerija – 07 T)

PADEKA

Esu be galo dėkinga visiems, kurie doktorantūros studijų metu buvo šalia ir yra tiesiogiai ar ne tiesiogiai prisidėję prie mano disertacijos rengimo.

Nuoširdžiai dėkoju:

Mokslinei vadovei prof. dr. Valentinai Dagienei už vadovavimą, pagalbą, begalinę kantrybę, šilumą ir nuolatinę paskatą studijuojant doktorantūroje bei rengiant disertaciją.

Disertacijos recenzentams doc. dr. Eugenijui Kurilovui ir dr. Gintautui Grigui už vertingas ir konstruktyvias pastabas, patarimus ir pagalbą tobulinant darbą.

Vilniaus universiteto Matematikos ir informatikos instituto direktoriui prof. habil. dr. Gintautui Dzemydai už supratimą ir paramą vykstant į tarptautinius doktorantų mokymus. Utenos kolegijos direktoriui prof. dr. Gintautui Bužinskui už sudarytas sąlygas studijuoti doktorantūroje ir skirtą finansinę paramą vykstant į tarptautinius doktorantų mokymus ir konferencijas.

Vilniaus universiteto Matematikos ir informatikos instituto Informatikos metodologijos skyriaus kolegoms ir kolegėms už konstruktyvią kritiką ir patarimus, bendramokslei Anitai Juškevičienei už nuolatinę paramą, patarimus, konstruktyvias diskusijas ir supratimą.

Utenos kolegijos kolegėms dr. Vaidai Bartkutei-Norkūnienei ir Aureolei Maračinskienei už supratimą ir palaikymą sunkiomis akimirkomis.

Vilniaus universiteto Matematikos ir informatikos instituto direktoriaus padėjėjai Danutei Rimeisienei už pagalbą ir kantrybę tvarkant disertacijos administracinius reikalus ir nuolatinį palaikymą.

Šeimos nariams ir artimiems žmonėms už supratingumą ir sudarytas sąlygas studijuoti doktorantūroje ir dirbti mokslinį darbą.

TURINYS

PAVEIKSLŲ SĄRAŠAS	6
LENTELIŲ SĄRAŠAS	8
TERMINŲ IR SANTRUMPŲ ŽODYNAS	10
1. ĮVADAS.....	14
1.1. DARBO AKTUALUMAS.....	14
1.2. TYRIMO OBJEKTAS	17
1.3. DARBO TIKSLAS	17
1.4. DARBO UŽDAVINIAI.....	17
1.5. NAUDOTI TYRIMO IR DUOMENŲ ANALIZĖS METODAI	17
1.6. MOKSLINIS NAUJUMAS	18
1.7. GINAMI TEIGINIAI.....	18
1.8. DARBO REZULTATŲ PRAKTINĖ SVARBA	19
1.9. DARBO APROBAVIMAS.....	19
1.10. DARBO APIMTIS IR STRUKTŪRA	22
2. PROGRAMAVIMO MOKYMO IR ADAPTYVIŲJŲ SISTEMŲ LITERATŪROS ANALIZĖ.....	25
2.1. Programavimo mokymo ypatumai	25
2.1.1. Programavimo mokymosi situacija.....	25
2.1.2. Bendrieji programavimo mokymo bruožai	27
2.1.3. Kodėl sunku išmokti programuoti.....	30
2.1.4. Mokomosios programavimo kalbos kriterijai	32
2.1.5. <i>SOLO</i> ir <i>Bloom</i> ‘o taksonomijų taikymas programavimui mokytis	40
2.1.6. Programavimo mokymo turinys	46
2.1.7. Programavimo mokymo strategijos.....	47
2.2. Egzistuojančių programavimo mokymo įrankių ir aplinkų analizė	53
2.3. Adaptyviosios technologijos ir mokymosi sistemos	58
2.4. Programiniai agentai ir agentinės sistemos	70
2.5. Mokymosi stiliai.....	76
2.5.1. Mokymosi stilių įvairovė.....	77

2.5.2.	Mokymosi stilių klasifikavimas naudojant Herrmann'o smegenų dominavimo instrumentą.....	78
2.6.	Mokymosi medžiagos elementai ir jų saugyklos	80
2.6.1.	Mokomieji objektai	80
2.6.2.	Metaduomenų standartai ir specifikacijos	81
2.6.3.	Mokomųjų objektų metaduomenų saugyklos	83
2.6.4.	Harvesting modelis reikiamų duomenų paieškai	84
2.6.5.	Mokomojo dalyko kūrimo automatizavimas.....	87
2.7.	Mokymo sistemos kūrimo modeliai.....	88
2.8.	Analizės išvados.....	93
3.	ADAPTYVIOSIOS PROGRAMAVIMO MOKYMOSI SISTEMOS PROJEKTAVIMAS.....	96
3.1.	Adaptyviosios programavimo mokymo sistemos kūrimo prielaidos..	96
3.2.	Mokomųjų objektų išrinkimas pagal mokymosi stilių.....	101
3.2.1.	Programavimo mokomieji objektai	101
3.2.2.	Mokomųjų objektų tipų reitingavimo metodas	108
3.3.	Mokomųjų objektų išrinkimas ir kokybės vertinimas vartotojų atžvilgiu	112
3.4.	Programavimo mokymosi kurso kūrimas (atvaizdavimas).....	116
3.5.	Adaptyvaus PMK teikimas besimokančiajam	121
3.6.	Mokymosi rezultatų vertinimas.....	124
3.7.	Adaptyviosios programavimo mokymosi sistemos modelis.....	126
3.8.	Skyriaus išvados.....	133
4.	MODELIO VERTINIMAS	134
4.1.	Sistemos kokybės ekspertinio vertinimo apibrėžimas	135
4.2.	Adaptyviosios programavimo mokymo priemonės modelio kokybės ekspertinis vertinimas.....	136
4.3.	Skyriaus išvados.....	141
	BENDROSIOS IŠVADOS IR REZULTATAI.....	143
	LITERATŪRA	146

PRIEDAI

1 priedas. Pasaulinių informatikos mokymo program	155
2 priedas. LOM metaduomenų schema	157
3 priedas. Mokymosi objekto metaduomenims aprašyti klasifikatoriai	158
4 priedas. HBDI klausimynas	163
5 priedas. Mokymosi objektų tipų tinkamumo ekspertinio vertinimo anketa	168
6 priedas. MO atitikimo mokymosi stiliui ekspertinio vertinimo rezultatai	170

PAVEIKSLŲ SĄRAŠAS

1 pav. Disertacijos struktūra	24
2 pav. Galimi programavimo pradmenų komponentai	28
3 pav. Programavimo sunkumus nulemiančios priežastys	31
4 pav. SOLO taksonomijos lygmenys (Biggs, 2003)	41
5 pav. Kognityvinė, emocinė ir psichomotorinė Bloom'o taksonomijos sritys ir lygmenys	42
6 pav. Bloom'o taksonomijos taikymas programavimui mokytis	44
7 pav. SOLO ir Bloom'o taksonomijų sąryšis (Targamadžė, Nauckūnaitė, 2009)	46
8 pav. Didaktinis trikampis (Berghlund, Lister, 2010)	51
9 pav. 3P mokymosi modelis (Biggs, 2003)	53
10 pav. Keturių procesų adaptyviųjų technologijų taikymo modelis (Schute, Zapata-Rivera, 2008)	60
11 pav. Keturių procesų adaptyviųjų technologijų taikymo modelis (Schute, Zapata-Rivera, 2008)	61
12 pav. Komunikavimo tarp agentų ir besimokančiųjų schema (Schute, Zapata-Rivera, 2008)	61
13 pav. Adaptyvios mokymosi sistemos modelis (Brusilovsky, Maybury, 2002)	64
14 pav. Adaptyviosios mokymosi sistemos pagrindiniai elementai (pagal Harris Liebergot)	64
15 pav. Prižiūravimo adaptavimo schema	68
16 pav. Sustiprinto adaptavimo schema	69
17 pav. Neprižiūravimo adaptavimo schema	70
18 pav. Nwana agentų klasifikacija (pagal Nwana)	74

19 pav. Davis agentų klasifikacija (pagal Davis)	74
20 pav. Reaguojančio agento architektūra	75
21 pav. Reaguojančio agento su būseną architektūra	75
22 pav. Agento sąveikavimas	76
23 pav. Kelių agentų sąveikavimas	76
24 pav. Agentų bendravimo schema (pagal Lupeikienę)	76
25 pav. Smegenų mokymo ir mokymosi modelis (pagal Herrmann)	78
26 pav. MO struktūra	81
27 pav. Lietuvos MO metaduomenų saugyklos schema (Kubilinskienė, 2012)	84
28 pav. Metaduomenų išrinkimo modelis (Fegen, 2007)	85
29 pav. Metaduomenų analizė, paieškos profilio kūrimas ir metaduomenų transformacijos procesas (Tennant, 2004)	86
30 pav. Pagalbinio profilio ir procedūrų kūrimas (Tennant, 2004)	86
31 pav. Paieškos rezultatų radimo procesas (Tennant, 2004)	87
32 pav. ADDIE modelio schema (pagal Davis, 2013)	89
33 pav. ADDIE modelio fazių aprašymas (pagal Davis, 2013)	89
34 pav. Veiklos analizės kvadrantas (pagal Clark, 2013)	90
35 pav. Seels and Glasgow modelio schema (pagal Seels, Glasgow, 1998)	91
36 pav. Dick ir Carey modelio schema (pagal Dick, Carey, 1990)	92
37 pav. APMS konteksto modelis	98
38 pav. MO duomenų struktūra	99
39 pav. APMS naudotojai	99
40 pav. APMS konteksto diagrama	100
41 pav. Mokymosi turinio ekosistema (adaptuota pagal Hodgins, 2002)	100
42 pav. MO tipų priskytrimo rūšiai atvaizdavimas	106
43 pav. Mokymosi stiliaus sąsajos su MO tipu atvaizdavimas	107
44 pav. Turinio MO išdėstymas	109
45 pav. Užduoties MO išdėstymas	110
46 pav. Vertinimo MO išdėstymas	110
47 pav. MO tipų reitingavimo modelis	111
48 pav. MO tipų reitingavimo programinio agento veikimo algoritmo schema	112
49 pav. MO išrinkimo modelis	113
50 pav. MO išrinkimo programinio agento veikimo algoritmo schema	114

51 pav. Harvesting programinio agento veikimo algoritmo schema.....	115
52 pav. MO tipų adaptavimo besimokančiajam modelis.....	115
53 pav. Mokymosi stiliaus stebėjimo programinio agento veikimo algoritmo schema	116
54 pav. Mokymosi kurso elementų išdėstymo struktūrinė schema	117
55 pav. Temos „Paprasti skaitiniai algoritmai“ dekomponavimas (adaptuota pagal Kurilovą, 2008).....	118
56 pav. PMK kūrimo modelis.....	119
57 pav. PMK kūrimo agento veikimo algoritmo schema	120
58 pav. Kompetencijų ir vertinamųjų užduočių sąryšis.....	120
59 pav. Adaptyvaus PMK teikimo modelio schema	121
60 pav. Adaptyvaus PMK teikimo agento veikimo algoritmo schema	123
61 pav. Mokymosi rezultatų vertinimo modelis	124
62 pav. Dalykinių kompetencijų vertinimo modelio schema	125
63 pav. Dalykinių kompetencijų vertinimo algoritmo schema.....	126
64 pav. APMS esybių ryšių diagrama	127
65 pav. APMS panaudos atvejų diagrama	127
66 pav. PMK kūrimo sekos diagrama.....	128
67 pav. Besimokančiojo modelio schema	128
68 pav. PMK naudojimo sekos diagrama	129
69 pav. APMS naudojimo schema	130
70 pav. APMS modelio schema.....	131
71 pav. APMS agentų veikimo sekos diagrama	133
72 pav. Modelio verifikavimo schema	134
73 pav. Ekspertų vertinimų standartinio nuokrypio priklausomybė nuo ekspertų skaičiaus (pagal Libby, 1978).....	136
74 pav. APMS ekspertinio vertinimo kriterijų ir alternatyvų sąsajos.....	138
75 pav. Fuzzy lingvistinių kintamųjų reikšmių sąsajos	139
76 pav. LOM metaduomenų schema	157

LENTELIŲ SĄRAŠAS

1 lentelė. Mokslininkų siūlomi programavimo kalbos, kuri būtų tinkama mokyti, kriterijai.....	30
--	----

2 lentelė. Mokslininkų siūlomi programavimo kalbos, kuri būtų tinkama mokyti, kriterijai.....	35
3 lentelė. Programavimo kalbų palyginimas pagal 17 kriterijų.....	38
4 lentelė. Programavimo mokymosi priemonių analizė	54
5 lentelė. Keturių procesų adaptyvaus ciklo scenarijai (Schute, Zapata-Rivera, 2008)	60
6 lentelė. Adaptojami kintamieji (Schute, Zapata-Rivera, 2008)	62
7 lentelė. Mokymosi stilių nustatymo instrumentai.....	77
8 lentelė. Herrmann'o mokymosi stilių aprašymas	79
9 lentelė. Mokymo sistemos konstravimo etapai (pagal Gagne, Briggs ir Wager).....	88
10 lentelė. Veiklos analizės kvadrantų aprašymas	90
11 lentelė. Dick ir Carey modelio etapų aprašymas	92
12 lentelė. MO tipai, tinkami programavimo mokymui (pagal LOM).....	102
13 lentelė. Atmesti MO tipai (pagal LOM)	103
14 lentelė. MO tipų reitingai pagal kvadrantus	110
15 lentelė. MO tipų reitingavimo agento aprašas	112
16 lentelė. MO išrinkimo programinio agento aprašas.....	113
17 lentelė. Mokymosi stiliaus stebėjimo agento aprašas	116
18 lentelė. PMK kūrimo agento aprašas	119
19 lentelė. Adaptyvaus PMK teikimo agento aprašas	122
20 lentelė. Dalykinių kompetencijų vertinimo agento aprašas.....	125
21 lentelė. MO paieškos saugykloje rezultatai	134
22 lentelė. APMS modelio vertinimo kriterijai (pagal ISO/IEC FCD 25010)	137
23 lentelė. APMS modelio vertinimo kriterijų reikšmės.....	138
24 lentelė. APMS ekspertų vertinimo rezultatai.....	139
25 lentelė. Lingvistinės skalės konvertavimo į neraiškiuosius skaičius reikšmės.....	140
26 lentelė. Konvertuoti į trikampus neraiškiuosius skaičius ekspertų vertinimo rezultatai	140
27 lentelė. Trikampių neraiškiųjų skaičių konvertavimo į lingvistinės skalės kintamuosius reikšmės.....	141
28 lentelė. MO atitikimo mokymosi stiliui ekspertinio vertinimo rezultatai	170

TERMINŲ IR SANTRUMPŲ ŽODYNAS

Terminų žodynas

Adaptyvioji sistema	sistema, galinti automatiškai keisti savo funkcijas, remdamasi sukaupta informacija arba pakitus aplinkos sąlygoms
Adaptyvus	pritaikantis prie besikeičiančių sąlygų. Adaptyvumu pasižyminti įranga įsimeina situacijas, įvykius, kaupia informaciją, jai pateikiamus duomenis ir visa tai analizuodama pati nustato tinkamesnius būsimus veiksmus
Agentas	esybė, reaguojanti į aplinką naudodamasi jutikliais ir daranti poveikį aplinkai savo manipulatoriais
Besimokančiojo modelis	besimokančiojo, kuris yra remiamas (palaikomas) adaptyviosios sistemos, pavaizdavimas (Shute, Zapata-Rivera, 2008)
Besimokantysis	bet kuris individas, kuris mokosi, įgyja žinias, įsitikinimus ar įgūdžius
Dalykinių kompetencijų vertinimas	procesas, kurio metu vertinami besimokančiojo mokymosi pasiekimai ir juos pagrindžiantys įrodymai
Gebėjimas	individo išmoktas veiksmo atlikimo būdas, pagrįstas įgytomis žiniomis ir įgūdžiais. Gebėjimai susidaro ir yra lavinami praktikuojantis. Išmokęs kurį veiksma, individas gali jį pakartoti ne tik įprastinėmis, bet ir pakitusiomis sąlygomis.
Herrmann'o mokymosi stilius	besimokančiojo mokymosi stilius pagal <i>Herrmann'o</i> mokymosi stilių klasifikaciją (sukurta <i>Ned Herrmann</i>). Pagal šią klasifikaciją išskiriami 4 mokymosi stiliai, atitinkantys žmogaus smegenų fizinio funkcionavimo atliekant užduotis principus
Interaktyvus	sąveikaujantis su vartotoju-žmogumi pagal gautus duomenis bei komandas ir pateikiantis rezultatus ar keičiantis duomenis
Įgūdis	vienas iš atlikimo lygmenų, pasireiškiantis kaip iki automatizmo išlavintas asmens mokėjimas; mokėjimo kokybiškumo rodiklis (kuo daugiau asmens mokėjimų dėl profesinės patirties tampa įgūdžiais, tuo asmens veikla yra profesionalesnė, nes dalį savo dėmesio asmuo gali skirti kitoms profesinės veiklos problemoms spręsti)
Kompetencija	vientisas asmens įsisąmonintų žinių, susiformavusių gebėjimų ir nuostatų junginys, leidžiantis jam sėkmingai ir atsakingai veikti konkrečiame socialiniame kontekste. Ugdymo programose įvairios asmens kompetencijos pateikiamos kaip ugdymo tikslai, numatomi mokymosi rezultatai

Konceptas	abstrakti arba bendra idėja, gaunama ar numanoma iš konkrečių atvejų ar įvykių
Konceptualus modelis	vaizdinis metodas, leidžiantis diagrama pavaizduoti priežastinius ryšius tarp faktorių kurie, manoma, yra reikšmingi nagrinėjamai problemai ar objektui
MO metaduomenų saugykla	tikslingai parengta sistema, kurią sudaro mokymosi objektų metaduomenys ir valdymo priemonės
Modelis	realaus reiškinių, proceso, struktūros, sistemos abstrakcija. Modelyje paliekamos tik tos savybės, kurios svarbios sprendžiamai problemai
Mokėjimas	kompetentingumo pagrindas, rodantis gebėjimą tinkamai atlikti profesinį veiksmą
Mokymosi aplinka	edukacinę vertę turinčios žmonių gyvenimo bei veiklos erdvės, įgalinančios individo asmeninį tobulėjimą, realizuojamą per mokymosi pastangas (Jucevičienė, 2001)
Mokymosi kursas	mokomojo dalyko programoje aprašytų turinio elementų realizavimas elektroninėje erdvėje (pvz. virtualioje mokymosi aplinkoje)
Mokymosi pasiekimai	subjektyviai įvertinti besimokančiojo studijavimo rezultatai (žinios, mokėjimai, vertybės ir požiūriai), pademonstruoti vertinimo metu
Mokymosi rezultatai	mokymosi metu įgytos žinios, gebėjimai, vertybinės nuostatos, kurių pasiekimo lygis įvertinamas skaitine reikšme/ verte
Mokymosi stilius	tai pamėgtas galvojimo, informacijos apdorojimo ir supratimo būdas; veiksmų, elgesio ir požiūrio rinkinys, kuris duotoje situacijoje nulemia mokymąsi
Mokomasis objektas	bet koks skaitmeninis išteklius (elektroninio mokymosi turinio elementas), kurį galima naudoti mokymui (-si) ir taikyti iš naujo kituose mokymo (-si) kontekstuose
Mokomojo dalyko programa	kurioms nors mokslinės disciplinos, mokomojo dalyko išdėstymo (mokymo) programa – didaktiškai pagrįstas dokumentas, kuriuo nurodoma mokomojo dalyko dėstymo apimtis, turinys ir svarbiausi mokymo reikalavimai
Mokomųjų objektų saugykla	tikslingai parengta sistema, kurią sudaro mokomieji objektai (MO), jų metaduomenys ir valdymo priemonės
Programavimas	kompiuterių programų kūrimo proceso dalis, kai algoritmas užrašomas kuria nors programavimo kalba

Programinis agentas	autonominė programinės įrangos dalis, vykdanči vartotojo nurodymus arba užprogramuotas situacijos sprendimus
Sistema	rinkinys tarpusavyje bendraujančių arba nepriklausomų objektų, realių arba abstrakčių, formuojančių integruotą visumą
Supratimas	mokėjimas paaiškinti bei argumentuotai pagrįsti, kodėl būtent taip: a) sprendžiama problema; b) atliekamas veiksmas ar veiksmų seka (veikla). Supratimas yra būtina, bet nepakankama sąlyga vienam iš atlikimo lygmenų – mokėjimui pademonstruoti. Supratimas leidžia asmeniui pačiam kontroliuoti savo veiklą: savarankiškai ir teisingai atlikti būtinus mokėjimo veiksmus, sujungti juos į vieningą sistemą ir spręsti profesines problemas bei profesiskai tobulėti. Supratimas yra svarbus kompetentingumo požymis, nes juo grindžiamas mokėjimas mokytis
Vertinimo kriterijus	rodiklis, nurodantis tikslo ir uždavinio įgyvendinimo lygį
Žinios	kompetentingumo elementas, parodantis asmens išmokus faktus, principus, teorijas ir patirtis, kurios yra susijusios su profesine bei kitomis socialinėmis veiklomis ir kurias asmuo gali pademonstruoti (skiriamos teorinės ir praktinės žinios)
Žinojimas	tikslingas ir efektyvus žinių bei kitų kompetentingumo elementų taikymas abstrakčiu lygmeniu. Žinojimas yra būtina, bet nepakankama gebėjimo sąlyga – galima žinoti ir pademonstruoti reikiamą žinių bei kitų kompetentingumo elementų taikymo seką, bet negalėti paaiškinti, kodėl veikla atliekama būtent tokia seka, nes nėra supratimo. Žinojimas yra supratimo prielaida. Supratimas yra žinojimo tvirtumo sąlyga
Žymė	nehierarchinis terminas arba reikšminis žodis, priskiriamas tam tikram informacijos vienetui (pvz., interneto žymė). Ši metaduomenų rūšis padeda apibūdinti elementą ir leidžia jį vėliau rasti atliekant paiešką.

Santrumpų žodynas

ACM	kompiuterijos technikos asociacija, angl. <i>Association for Computing Machinery</i>
ADDIE	sisteminis mokymosi konstravimo modelis 1975 metais sukurtas Floridos valstijos universiteto Edukacinių technologijų centre
ADL	iniciatyva, atsiradusi JAV gynybos departamento iniciuoto projekto metu, skatinusio esamas mokymosi ir informacines technologijas panaudoti mokymui ir kvalifikacijos tobulinimui, angl. <i>Advanced Distributed Learning</i>
APMS	adaptyvioji programavimo mokymosi sistema
LOM	duomenų modelis, dažniausiai užkoduotas XML, ir naudojamas aprašyti mokomiesiems objektams ir panašioms skaitmeniniams ištekliams, skirtiems remti mokymąsi, angl. <i>Learning Object Metadata</i>
MO	mokomasis objektas
MO_T	turinio mokomasis objektas, skirtas įsisavinti dalyko srities žinias ir supratimą
MO_U	užduoties mokomasis objektas, skirtas lavinti praktinius dalyko srities įgūdžius
MO_V	vertinimo mokomasis objektas, skirtas patikrinti dalyko srities supratimą ir įgytus praktinius įgūdžius
MOR	mokomųjų objektų tipų rūšis – turinio, užduoties arba vertinimo
OAI-PMH	metaduomenų išrinkimui iš saugyklų naudojamas protokolas, angl. <i>Open Archives Initiative Protocol for Metadata Harvesting</i>
PMK	programavimo mokymosi kursas, vartojamas adaptyviojoje programavimo mokymosi sistemoje
SMOS	specializuota mokomųjų objektų metaduomenų saugykla

1. ĮVADAS

1.1. DARBO AKTUALUMAS

Programavimo mokymas yra pakankamai opi problema ne tik Lietuvoje, bet ir daugelyje užsienio šalių. Šia tema yra parašyta nemažai mokslinių straipsnių, parengta ir apginta ne viena disertacija (Bennedsen, 2008; Caspersen, 2007; Kinnunen 2009, Sorva, 2012 ir kt.), yra kelios stiprios mokslininkų grupės, kurios užsiima programavimo, algoritmavimo mokymo klausimais (pvz., Aalto, Joensuu, Miuncheno technikos universitetuose). Šiuo metu pasaulyje aktyviai tyrinėjamas suasmenintas mokymas (-is) (toliau naudojama mokymasis), o jis gali būti siejamas su adaptyviosiomis mokymo priemonėmis-sistemomis.

Išmokti programuoti nėra lengva. Besimokantysis pirmiausia turi išsiugdyti algoritminį mąstymą. Pagrindinis šio mąstymo požymis – gebėjimas užduotį išreikšti algoritmu taip, kad jį galėtų išspręsti kompiuteris (t. y. mokėti suprogramuoti). Kokiomis priemonėmis ar metodais yra mokoma algoritminio mąstymo? Dažniausiai dėstytojas ar mokytojas supažindina su programavimo kalbos sintakse, algoritmus bei programų pavyzdžius užrašo lentoje ar parodo projektoriumi. Tai užima daug laiko ir neveiksminga. Retai naudojamos interaktyvios lentos ar projektoriai, kuriais mokytojo parašytą programą aiškiai mato visi besimokantieji ir visi kartu gali ją nagrinėti. Teigiamą poveikį mokymuisi turi informacijos vizualizavimas (Ebel, Ben-Ari, 2006), tačiau jis dažniausiai yra naudojamas kaip mokytojo pagalbinė priemonė – tiesiog mokytojas medžiagą pateikia kitokia forma, panaudodamas vaizdo medžiagą.

Akcentuojama, kad veikiant (atliekant užduotis praktiškai) žmogaus atmintis optimizuoja mokymąsi (Caspersen, 2007). Taip pat svarbu yra tai, jog dėmesio kokybė koreliuoja su mokymosi veiksmingumu – kuo daugiau laiko išlaikomas besimokančiojo dėmesys, tuo geresni jo pasiekimai (Ebel, Ben-Ari, 2006). Šiuolaikinės technologijos, kompiuteriai ir internetas suteikia galimybes mokymąsi padaryti įdomesnę ir efektyvesnę, palengvinti mokančiojo darbą, motyvuoti besimokantįjį.

Mokant ir mokantis programavimo (algoritmavimo) svarbu ne tik klaidų ar teisingų sprendimų skaičius, bet ir besimokančiojo bei jį mokančiojo asmens sąveika, komunikavimas. Dažnai ta sąveika vyksta neakivaizdžiai, per tarpininką – kompiuterį. Mokantis daug laiko sugaištama ieškant, kaip ištaisyti padarytas klaidas, kaip iš jų pasimokyti, kaip laiku gauti atsakymus į rūpimus klausimus. Tam kuriamos įvairios automatinės programavimo ir algoritmavimo mokymo sistemos (pvz., *TRAKLA2*, *VILLE* ir kt.). Šios sistemos moko programavimo ir algoritmavimo principų, tačiau jos nėra pakankamai draugiškos besimokančiajam, o ypač joms trūksta adaptyvumo.

Jau kelis dešimtmečius ieškoma efektyvesnių mokymo bei mokymosi priemonių, metodų bei būdų. Vis labiau akcentuojama individualizuoto mokymosi svarba – kai atsižvelgiant į individualias besimokančiojo savybes parenkami mokomieji objektai (MO), mokymosi veiklos, mokymosi scenarijai ir pan. Užtikrinti individualizavimą padeda ir adaptyviosios mokymosi sistemos, kurios leidžia mokymosi metu priderinti besimokančiajam tinkamiausią mokymosi turinį. Tuo pačiu bandoma pritaikyti ir pasinaudoti anksčiau sukurtais kokybiškais ir saugyklose saugomais mokomaisiais elementais – MO.

Mokymosi adaptavimas individualiems besimokančiojo poreikiams ir Lietuvoje, ir užsienyje analizuojamas jau senokai. Yra atlikta nemažai tyrimų, kurių metu įvairiais metodais – atsižvelgiant į besimokančiojo savybes ir poreikius (Weber, Brusilovsky, 2001; Baziukaitė ir kt., 2008), analizuojant besimokančiojo veiklas elektroninėje mokymosi aplinkoje (Preidys, Žilinskienė, 2012), nustatant besimokančiojo mokymosi stilių (Franzoni, Assar, 2009) – parenkamas arba leidžiama pačiam besimokančiajam pasirinkti tinkamiausią mokymosi turinį arba mokymosi strategiją (Preidys, Žilinskienė, 2012), mokantis aplinkoje palaikoma adaptyvi navigacija (Weber, Brusilovsky, 2001; Paramythis, Loidl-Reisinger, 2004; Komlenov, Budimac, Ivanovic, 2009). Mokymosi sėkmė priklauso nuo to, kaip maksimaliai pasiekiami mokymosi tikslai – įgyjamos reikiamos dalykinės kompetencijos (žinios bei įgūdžiai, kurie yra aprašomi mokymosi rezultatais), bei kokias emocijas

mokymosi patiria besimokantieji. Mokymosi sėkmę didžia dalimi lemia mokymosi efektyvumas, kuris priklauso nuo: noro mokytis ir mokėjimo mokytis. Pastaruosius lemia ir individualios besimokančiojo savybės, kurios skiriasi priklausomai nuo mokymosi stiliaus. Moksliniuose darbuose dažniausiai nagrinėjamas mokymosi adaptavimas atsižvelgiant į mokymosi stilius pagal *Kolb, Honey - Mumford, Felder - Silverman* klasifikacijas (Hawk, Shah, 2007). Tuo tarpu yra vos keli darbai, kuriuose *Herrmann*‘o stilių klasifikacija nagrinėta kompiuterių mokslų mokymosi kontekste (Kannangara, 2013), tačiau taikymų mokymosi sistemose neaptikta.

Disertaciniame darbe nagrinėjama programavimo mokomojo dalyko medžiagos adaptavimas prie besimokančiojo individualių savybių – jo mokymosi stiliaus pagal *Herrmann*‘o mokymosi stilių klasifikaciją (toliau *Herrmann*‘o mokymosi stilius). Tuo tikslu išnagrinėti programavimo mokymosi ypatumai, egzistuojančios programavimo mokymo aplinkos ir sistemos, adaptyviosios mokymosi sistemos, programiniai agentai ir agentinės sistemos, besimokančiųjų mokymosi stiliai ir mokymosi medžiagos elementai ir jų saugyklos bei mokymosi sistemos kūrimo modeliai. Darbe pateikiami šio darbo autorės sudaryti metodai: MO tipų reitingavimo pagal tinkamumą *Herrmann*‘o mokymosi stiliui metodas; patobulintas mokomojo dalyko programos susiejimo su dalykinės srities MO ir *Bloom*‘o taksonomijos lygmenimis metodas; programavimo mokymosi medžiagos elementų (MO) adaptavimo pagal besimokančiojo *Herrmann*‘o mokymosi stilių. Visi šie metodai kaip atskiri daugiaagentinės sistemos programiniai agentai integruoti į adaptyviosios programavimo mokymo priemonės (toliau adaptyvioji programavimo mokymosi sistema – APMS) modelį. Sukurtą konceptualų modelį įvertino parinkti ekspertai – buvo įsitikinta modelio vartojimo kokybe, t. y. efektyvumu, našumu, produktyvumu ir naudojamumu adaptuojant dalyko medžiagą besimokančiajam.

1.2. TYRIMO OBJEKTAS

Disertacinio darbo tyrimo objektas yra adaptyvioji programavimo mokymo priemonė (APMS).

1.3. DARBO TIKSLAS

Disertacinio darbo tikslas – ištyrus programavimo mokymo problematiką ir besimokančiųjų mokymosi galimybes bei poreikius sukurti adaptyviosios programavimo mokymo sistemos konceptualų modelį.

1.4. DARBO UŽDAVINIAI

Siekiant įgyvendinti pagrindinį tikslą iškelti tokie uždaviniai:

1. Atlikti programavimo mokymui naudojamų programinių sprendimų palyginamąją analizę adaptyvumo atžvilgiu.
2. Išskirti programavimo mokymui tinkamus mokomųjų objektų tipus ir juos suklasifikuoti pagal besimokančiojo mokymosi stilius.
3. Sukurti metodą, kuris leistų išrinkti programavimo mokomuosius objektus iš atitinkamų saugyklų atsižvelgiant į mokomojo dalyko programos reikalavimus ir besimokančiojo mokymosi stilių.
4. Parengti adaptyviosios programavimo mokymosi sistemos, adaptuojančios mokymosi medžiagą prie besimokančiojo atsižvelgiant į jo mokymosi stilių, konceptualų modelį.
5. Atlikti adaptyviosios programavimo mokymosi sistemos konceptualaus modelio vartojimo kokybinį vertinimą.

1.5. NAUDOTI TYRIMO IR DUOMENŲ ANALIZĖS METODAI

Rengiant disertacinį darbą buvo vartojami šie tyrimo ir duomenų analizės metodai:

- mokslinės literatūros ir tyrimų sisteminė analizė – ji naudota analizuojant programavimo mokymo ypatumus, metodus, problematiką, adaptyviųjų ir agentinių sistemų savybes;

- programavimo mokymo priemonių tyrimas – naudotas bandant išsiaiškinti programavimo mokymui naudojamų priemonių savybes adaptyvumo požiūriu;
- anketinė apklausa – naudota aiškintis besimokančiųjų mokymosi poreikį;
- ekspertinis vertinimas – naudotas vertinant mokomųjų objektų tipų tinkamumą mokymosi stiliui ir sumodeliuotos sistemos vartojimo kokybę;
- modeliavimas – naudotas kuriant ir aprašant APMS modelį.

1.6. MOKSLINIS NAUJUMAS

- Parengtas programavimo mokomųjų objektų tipų ekspertinio vertinimo rezultatų naudojimo šiems tipams reitinguoti pagal *Herrmann*‘o mokymosi stilių metodas, leidžiantis individualizuoti mokymąsi;
- Pasiūlytas mokomojo dalyko programos susiejimo su dalykinės srities mokomaisiais objektais ir *Bloom*‘o taksonomijos lygmenimis metodas;
- Parengtas metodas, kuriuo programavimo mokomojo dalyko mokymosi medžiaga (mokomieji objektai) adaptuojama atsižvelgiant į besimokančiojo mokymosi stilių, remiantis *Herrmann*‘o mokymosi stilių klasifikacija.

1.7. GINAMI TEIGINIAI

- Mokomųjų objektų tipų tinkamumo besimokančiojo *Herrmann*‘o mokymosi stiliui ekspertinis vertinimas padeda šiuos objektus pritaikyti mokomojo dalyko mokymosi rezultatams pasiekti. Šis metodas leidžia susieti mokomųjų objektų tipus su besimokančiojo mokymosi stiliumi ir taip individualizuoti mokymąsi;
- Mokomųjų objektų susiejimo su mokomojo dalyko programa metodas leidžia sukurti technologiją, kuri pažymėtus mokomojo dalyko medžiagos elementus (mokomuosius objektus) susieja su *Bloom*‘o taksonomijos lygmenimis ir tuo pačiu formuoja formalaus vertinimo užduočių svorius dalykinėms kompetencijoms įvertinti;

- Programavimo mokymosi sistema, adaptuojanti mokomuosius objektus prie besimokančiojo pagal jo *Herrmann*'o mokymosi stilių, leidžia nukreipti mokymąsi siekti mokomojo dalyko programoje numatytų dalykinių kompetencijų bei įvertinti jų pasiekimo lygį;
- Konceptuali modeliu aprašyta adaptyvioji programavimo mokymosi sistema yra efektyvi, naši, produktyvi ir naudotina (lanksti ir suderinama su kontekstu).

1.8. DARBO REZULTATŲ PRAKTINĖ SVARBA

- Naudojant pasiūlytą adaptyviosios programavimo mokymosi sistemos modelį galima sukurti sistemą, kuri pasižymėtų tokiais parametrais: efektyvumu, našumu, produktyvumu ir naudojamumu (lankstumu ir konteksto suderinamumu). Ši sistema galėtų būti integruojama kaip atskiras modulis į egzistuojančias ir plačiai naudojamas virtualias mokymosi aplinkas, pvz., *Moodle*.
- Pasiūlytas modelis gali būti taikomas kitose mokymosi srityse atitinkamai išskiriant tos srities mokymuisi labiausiai tinkančius mokomųjų objektų tipus ir mokomojo dalyko programos apraše įvardinant temas, dalykines kompetencijas ir *Bloom*'o taksonomijos lygmenis.

1.9. DARBO APROBAVIMAS

1.9.1. Straipsniai recenzuojamuose mokslo leidiniuose

Disertacijos tematika buvo parengti straipsniai recenzuojamuose mokslo leidiniuose:

- Urbonienė J. Adaptyvios technologijos mokymosi procese. *Mokslinės-praktinės konferencijos „Informacinės technologijos 2010: teorija, praktika, inovacijos“ pranešimų medžiaga*. ISBN 978-609-8020-10-6. Alytaus kolegijos leidykla, 2010, p. 86-91

- Dagienė V., Urbonienė J. Programavimo mokymasis: lyginamoji kalbos ir aplinkos analizė. *Informacijos mokslai*. ISSN 1392-0561, VšĮ Vilniaus universiteto leidykla, 2010 54, p. 44-62
- Urbonienė J., Juškevičienė A. Web 2.0 technologijų adaptuojamumo programavimo mokymui galimybių tyrimas. *Studijos šiuolaikinėje visuomenėje. Recenzuojamų straipsnių rinkinys* (įtrauktas į EBSCO). ISSN 2029-431X, Šiaurės Lietuvos kolegijos leidykla, 2011, p. 161-170
- Urbonienė J. Programavimo mokymo aplinkų efektyvumo mokymo(si) atžvilgiu analizė. *Studijos šiuolaikinėje visuomenėje. Mokslo darbai*. ISSN 2029-431X, 2012 Nr. 3 (1) p. 243-254
- Urbonienė J. Adaptyvios programavimo mokymosi sistemos modelis. *Informacijos mokslai*. ISSN 1392-0561, VšĮ Vilniaus universiteto leidykla, 2013 66, p. 108-122
- Jadzgevičienė V., Urbonienė J. The Possibilities of Virtual Learning Environment Tool Usability for Programming Training. *Proceedings of the 6th International Conference Innovative Information Technologies for Science, Business and Education, IIT-2013* November 14-16, 2013
- Jadzgevičienė V., Urbonienė J. Cooperation in Programming Learning. *Proceedings of the 12th Koli Calling International Conference on Computing Education Research*, ACM ISBN: 978-1-4503-1795-5, 2012, p. 143-144

1.9.2. Pranešimai mokslinėse konferencijose ir seminaruose

Pranešimai skaityti Lietuvos ir užsienio mokslinėse konferencijose:

- Pranešimas „Adaptyvios technologijos mokymosi procese“. VIII mokslinė-praktinė konferencija „Informacinės technologijos 2010: Teorija, praktika, inovacijos“, Alytaus kolegija, Alytus, Lietuva. 2010-05-21
- Pranešimas „Web 2.0 technologijų adaptuojamumo programavimo mokymui galimybių tyrimas“. Mokslinė praktinė konferencija „Studijos

šiuolaikinėje visuomenėje“. Šiaurės Lietuvos kolegija, Šiauliai, Lietuva. 2011-03-01

- Stendinis pranešimas „Interneto technologijos ir modernus programavimo mokymas“. Tarptautinė mokslinė-praktinė konferencija „Darnaus vystymosi aspektai: teorija ir praktika“. Utenos kolegija, Utena, Lietuva. 2011 m. balandžio 14-15 d.
- Pranešimas „Mokymasis panaudojant internetines technologijas“. Respublikinė mokslinė praktinė konferencija „Mokslas ir studijos 2011: teorija ir praktika“. Šiaurės Lietuvos kolegija, Šiauliai, Lietuva. 2011-05-12
- Pranešimas „Programavimo mokymo aplinkų efektyvumo mokymo(si) atžvilgiu analizė“. Studijos šiuolaikinėje visuomenėje. Mokslinė praktinė konferencija „Studijos šiuolaikinėje visuomenėje“. Šiaurės Lietuvos kolegija, Šiauliai, Lietuva. 2012-02-23
- Pranešimas „Cooperation in Programming Learning“. 12th Koli Calling International Conference on Computing Education Research. Tahko, Suomija, 2012-11-15 – 2012-11-18
- Pranešimas „The Possibilities of Virtual Learning Environment Tool Usability for Programming Training“. Tarptautinė mokslinė-praktinė konferencija „Innovative Information Technologies for Science, Business and Education, IIT-2013“. Vilniaus verslo kolegija, Vilnius, 2013-11-14 – 2013-11-16

Pranešimai disertacijos tema skaityti tarptautiniuose doktorantų mokymuose:

- Tarptautiniai doktorantų mokymai „JTEL Summer School 2010 on Technology Enhanced Learning“. Ohrid, FYR Macedonia. 2010-06-10
- Tarptautiniai doktorantų mokymai „JTEL Summer School 2011 on Technology Enhanced Learning“. Chania, Crete, Greece. 2011-06-01

- Tarptautiniai doktorantų mokymai: „Informatics and Informatics Engineering Education Research: Methodologies, Strategies and Implementation“. Druskininkai, Lietuva. 2011-12-02
- Tarptautiniai doktorantų mokymai „JTEL Summer School 2012 on Technology Enhanced Learning“ (Estoril, Portugal). 2012-05-22
- Tarptautiniai doktorantų mokymai „Informatics Engineering Education Research: Methodologies, Methods and Practice“. Druskininkai, Lietuva, 2012-12-03 – 2012-12-07
- 10th World Conference on Computers in Education „Learning while we are connected“. Tarptautinis doktorantų konsorciumas. Torun, Lenkija, 2013-05-01 – 2013-05-05
- Tarptautiniai doktorantų mokymai „Informatics Engineering Education Research: Methodologies, Methods and Practice“. Druskininkai, Lietuva, 2013-12-03 – 2013-12-07

1.10. DARBO APIMTIS IR STRUKTŪRA

Darbą sudaro: terminų ir santrumpų žodynas, 3 pagrindinės dalys – skyriai, išvados ir rezultatai, naudotos literatūros sąrašas ir priedai. Darbo apimtis yra 170 puslapių. Tekste panaudoti 76 paveikslai, 28 lentelės ir 6 priedai. Rašant disertaciją buvo naudotasi 116 literatūros šaltinių.

Pirmajame skyriuje pateikiamas darbo įvadas, pristatomas darbo aktualumas, darbo tikslai ir uždaviniai, mokslinis naujumas, praktinė darbo reikšmė ir darbo aprobavimas.

Antrajame skyriuje nagrinėjama programavimo mokymo ypatumai, egzistuojančios programavimo mokymo aplinkos ir sistemos, adaptyvios technologijos ir mokymosi sistemos, programiniai agentai ir agentinės sistemos, besimokančiųjų mokymosi stiliai ir mokymosi medžiagos elementai ir jų saugyklos, mokymosi sistemos kūrimo modeliai.

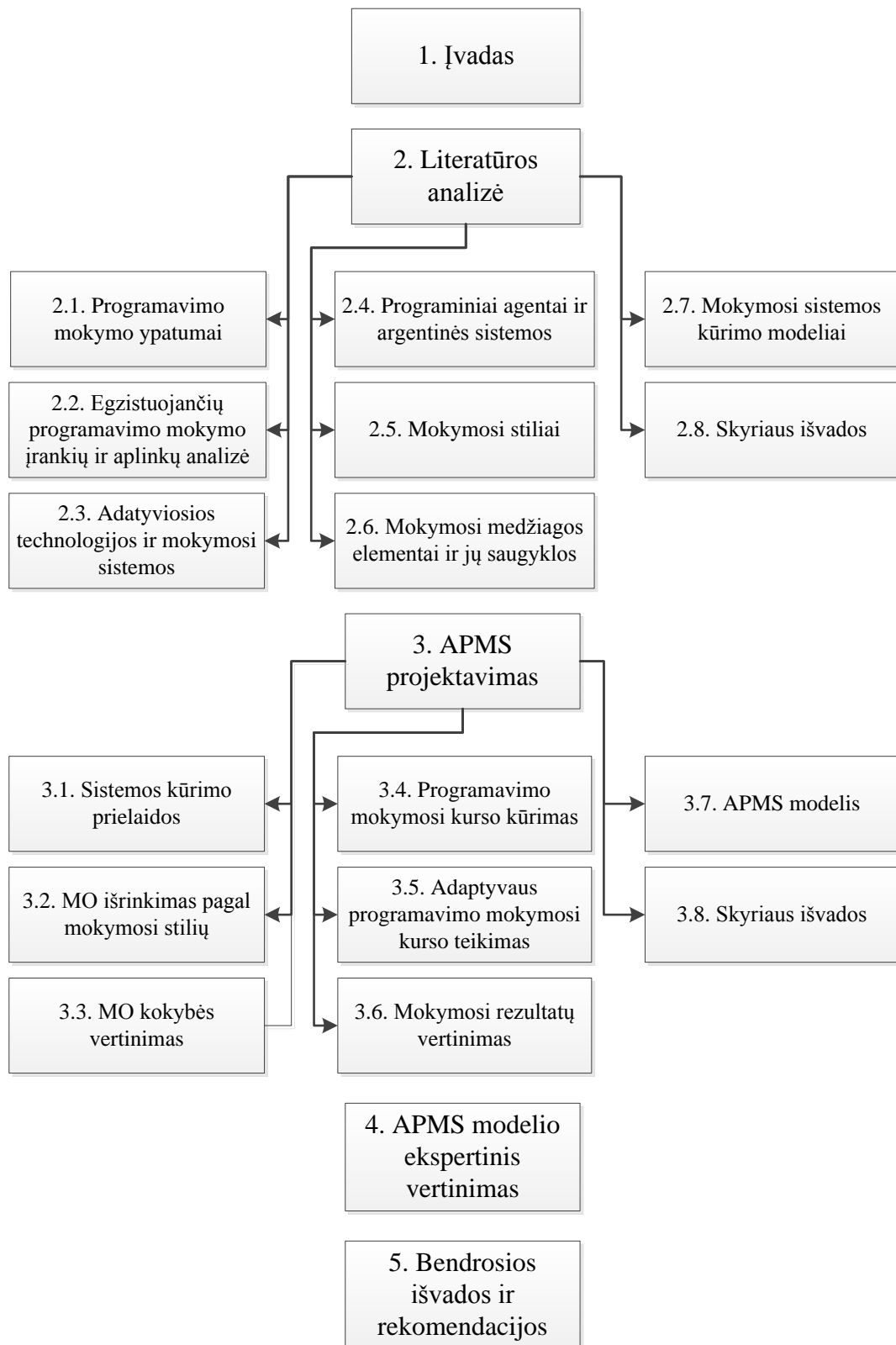
Trečiajame skyriuje aprašomas adaptyviosios programavimo mokymosi sistemos konceptualaus modelio projektavimas. Sistema projektuojama kaip daugiaagentinė sistema, naudojanti specializuotoje mokomųjų objektų

saugykloje saugomus kokybiškus programavimo mokomojo dalyko programą atitinkančius (pažymėtus) mokomuosius objektus. Mokomieji objektai yra besimokančiajam parenkami ir reitinguotai pateikiami atsižvelgiant į jo *Herrmann*‘o mokymosi stilių, o mokomųjų objektų tipų tinkamumas mokymosi stiliui nustatomas panaudojant ekspertinio vertinimo metodą.

Ketvirtajame skyriuje pateikiamas sukurto adaptyviosios programavimo mokymosi sistemos modelio vartojimo kokybės vertinimas. Naudojamas ekspertinio vertinimo metodas, kuris leidžia įvertinti sukurto sistemos modelio naudojimo kokybę tokiais požiūriais: efektyvumo, našumo, produktyvumo ir naudojamumo (lankstumo ir konteksto suderinamumo).

Darbo pabaigoje pateikiamas rezultatų apibendrinimas ir išvados.

Disertacinio darbo struktūros schema:



1 pav. Disertacijos struktūra

2. PROGRAMAVIMO MOKYMO IR ADAPTYVIŲJŲ SISTEMŲ LITERATŪROS ANALIZĖ

2.1. Programavimo mokymo ypatumai

Programavimo mokytis pradedama įvairaus amžiaus: nuo pradinių klasių iki aukštosios mokyklos ir dar vėliau. Tačiau naujokui programavimo mokymasis yra pakankamai sudėtingas bet kokiame amžiaus tarpsnyje. Pirmiausiai reikia suprasti programos vykdymo eigą, gebėti spęsti problemas, t. y. reikia pakeisti savo mąstymą – gebėti problemos sprendimą išreikšti algoritmu ir pagal jį išspęsti (suprogramuoti). Taigi, tuo pačiu reikia mokytis programavimo kalbos – sintaksės ir semantikos bei tai taikyti rašant programą, o taip pat suprasti naudojamos programavimo paradigmos esmę ir principus. Siekiant išsiaiškinti programavimo mokymo ypatumus toliau šiame skyrelyje nagrinėjama programavimo mokymosi situacija Lietuvoje ir užsienio šalyse, bendrieji programavimo mokymo bruožai, mokymosi sunkumo požymiai, apibrėžiami mokomosios programavimo kalbos kriterijai. Taip pat nagrinėjamas *SOLO* ir *Bloom*'o taksonomijų taikymas programavimui mokytis, programavimo mokymo metodai ir mokymosi turinys.

2.1.1. Programavimo mokymosi situacija

Mokyklose mokoma programuoti *Basic*, *Pascal*, *Delphi*, *C*, *C++* ir *Java* programavimo kalbomis.

Užsienio universitetuose įvairiose informatikos ir kompiuterijos studijose yra dėstomas programavimo įvadas, kurio metu mokytis programuoti naudojama *Pascal*, *C*, *C++* bei *Java* kalbos. Kai kur (pvz., Suomijoje, Austrijoje) mokoma programuoti *Python* ar kitomis programavimo kalbomis. Taip pat programavimo kursas yra integruojamas į kitus kursus.

UNESCO pateiktose informatikos (informacinių technologijų) mokymo rekomendacijose (Anderson, 2002) programavimo siūlo mokytis pasirinktiniu, sustiprintu lygiu. Daugelyje kitų dokumentų, pavyzdžiui, informacijos apdorojimo federacijos IFIP darbo grupės WG 3.1 parengtose rekomendacijose

(Taylor ir kt., 1993; Van Weert, 1993) kreipiamas dėmesys algoritminiam mąstymui ugdyti. Amerikos kompiuterijos mokslų mokytojų asociacijos ir kompiuterijos technikos asociacijos ACM parengtame standarte (*K-12*, 2011) taip pat akcentuojamas algoritminis mąstymas bei apibrėžiami kompiuterijos praktikos ir programavimo įvairių lygių gebėjimai.

Pirmoji pažintis su programavimu Lietuvoje vyksta bendrojo lavinimo mokykloje penktojoje klasėje, kai mokytojai supažindina su programų konstravimu (pasitelkdami *Logo* arba *Scratch* aplinkas). Vėliau dalis jaunuolių programuoti pradeda dar besimokydami mokykloje – 10–12 klasėje pasirinkdami programavimo modulį (čia dėmesys skiriamas algoritmavimui), kita dalis programuoti mokosi įstoję į aukštąją mokyklą (universitetą ar kolegiją), dar kiti mokytis pradeda vyresniame amžiuje – pradėję karjerą darbe. Jaunesnio amžiaus vaikai individualiai pradėti mokytis programuoti gali Vilniaus universiteto Matematikos ir informatikos instituto Jaunųjų programuotojų neakivaizdinėje mokykloje (JPM), KTU Jaunųjų kompiuterininkų mokykloje, Vilniaus kompiuterininkų akademijoje, Šiaulių jaunųjų kompiuterininkų mokykloje. Šiuo atveju mokymasis vyksta savaitgalį atvykstant į mokymo vietas arba nuotoliniu būdu.

Lietuvoje nuo 1986 m. iki pat pertvarkos 1999 m. privalomas informatikos kursas buvo dėstomas 10–12 klasėse, didžiausia dalis kurio buvo skirta algoritmavimui, dėl kompiuterių stygiaus dažnai jo buvo mokoma tik teoriniu lygiu. Taip darė ir kitos šalys, ypač kurių mokyklų programos orientuojasi į fundamentalių žinių perteikimą. Algoritmavimu siekiama išmokyti suprasti formalių žymenų reikšmę, gebėti juos vartoti atitinkamiems veiksams aprašyti. Šio dalyko dalis glaudžiai siejasi su matematika.

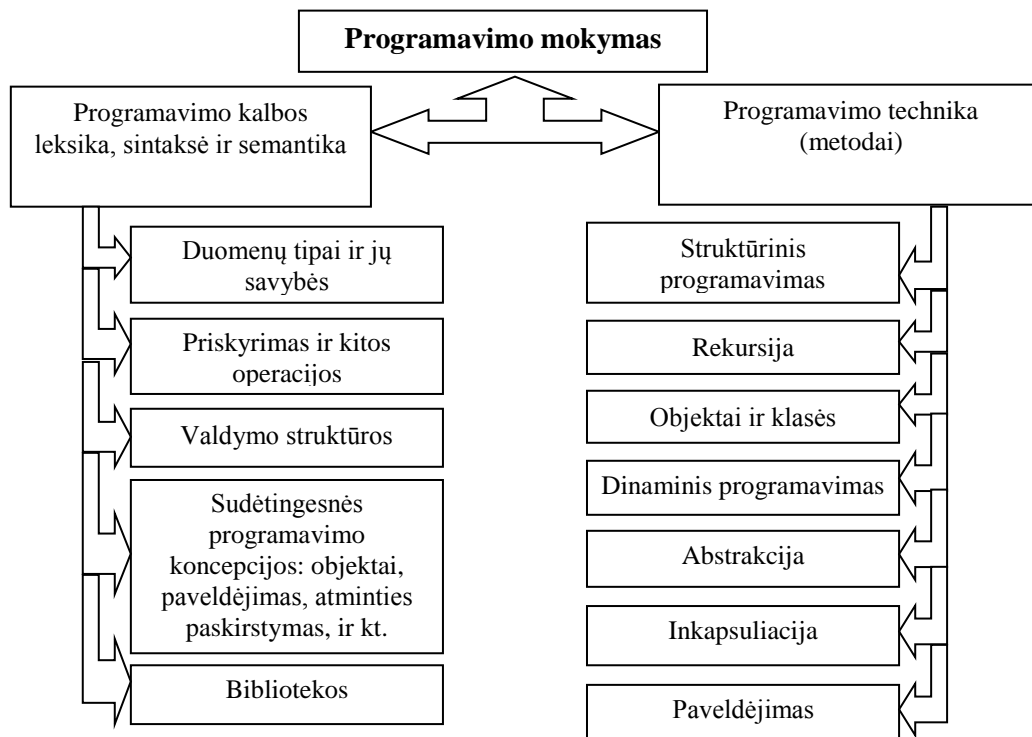
Lietuva turi ilgametę algoritmavimo mokymo patirtį, gajas sąsajas su taikomosios matematikos uždaviniais, nemažą pedagogų bei mokslininkų įdirbį šioje srityje. Sukurtus algoritmus reikia vykdyti kompiuteriu. Aktualus programavimo kalbos pasirinkimo klausimas. Lietuvos mokyklose nuo pat informatikos atsiradimo algoritmus buvo siūloma mokyti Paskaliu, tuo rėmėsi daugelį metų mokyklose vartojami informatikos, vėliau – informacinių

technologijų vadovėliai (Dagienė, 2003). Reformuotos Lietuvos mokyklos informacinių technologijų bendrojoje programoje algoritmams mokytis nesiūloma jokių konkrečių priemonių. Tačiau dauguma vadovėlių remiasi Paskalio kalbos minimalių konstrukcijų vartojimu. Čia dėstomi sveikieji ir realieji skaičiai, loginis tipas, priskyrimo bei sąlyginis sakiniai, ciklai.

Sparčiai vystantis technologijoms, jos vis dažniau naudojamos ir mokymosi procese. Yra kuriamos įvairios mokomosios programos bei svetainės, automatiniai mokymo ir žinių tikrinimo įrankiai, virtualios mokymosi aplinkos. Įvairiose mokymosi priemonėse gausu naudingos tekstinės, vaizdinės, garsinės medžiagos. Taip pat vis dažniau didelis dėmesys mokymosi procese skiriamas interaktyvumui. Pastaruoju metu tiek Lietuvoje, tiek ir užsienio šalyse vis dažniau imama kalbėti apie adaptyvumą.

2.1.2. Bendrieji programavimo mokymo bruožai

Programavimo mokymą galima suskirstyti į dvi stambias grupes: 1) programavimo kalbos sintaksės ir semantikos įvaldymas, 2) programavimo technikos (metodų) išmokimas (2 pav.). Programavimo kalbos sintaksė ir semantika pradedančiajam reikalinga tam, kad galėtų rašyti programas. Svarbiausia yra ne išmokti sintaksės konstrukcijas, bet gebėti jas taikyti. Svarbiausios programavimo kalbų konstrukcijos pavaizduotos 2 paveiksle. Programavimo technika (metodai) apima pagrindinių koncepcijų ir technikos mokymą. Taip pat reikia atkreipti dėmesį į programos projekto šablono (Modelis, peržiūra ir valdymas) struktūrą, algoritmo sudarymo modelius („Skaldyk ir valdyk“, dinaminis programavimas).



2 pav. Galimi programavimo pradmenų komponentai

Mokant programavimo technikos, kalba yra tik priemonė bendroms programavimo sąvokoms išreikšti ir iliustruoti. Programavimo technikos koncepcijos turi būti įvedamos taip, kad jos nepriklausytų nuo kalbos. Tuomet, kai yra mokoma programavimo technikos vis tik reikia pasirinkti kalbą, kuri nebūtų per daug reta ar abstrakti. Kai mokoma nenaudojamos kalbos, besimokantysis gali pagalvoti, jog tam tikros programavimo paradigmos jie gali išmokti naudodami būtent šią kalbą ir kitose dažniau naudojamose kalbose to nėra.

Pasitaiko klaidinga programavimo mokymosi samprata, manoma, jog tai yra mokymasis užrašyti uždavinio sprendimą programos tekstu panaudojant programavimo kalbos konstrukcijas. Programos rašymas yra tik vienas iš programavimo įgūdžių. Vienodai svarbu yra gebėti skaityti ir suprasti programų tekstus. Juk programuotojas nemažai laiko praleidžia nagrinėdamas pavyzdžius – kitų parašytas programas (Mannila, 2007; Blonskis, Dagienė, 2003). Galima būtų manyti, jog mokantis rašyti programas savaime išmokstama jas skaityti, sekti programos vykdymą. Tačiau tyrimai rodo visai ką kitą. Winslow'o tyrimas parodė, kad gebėjimas parašyti programą ir

gebėjimas ją skaityti mažai koreliuoja. Abiejų turi būti mokoma (Winslow, 1996). Spinellis (2003) taip pat pabrėžia, kad mokant programavimo reikia nepamiršti, kad gebėjimas skaityti ir suprasti jau parašytas programas yra labai svarbus, nes analizuojant kitų parašytas programas formuojasi savitas programavimo stilius. Taigi, programų skaitymo gebėjimą irgi reikia ugdyti.

Šiuolaikinės programavimo kalbos leidžia kurti aukštos kokybės produktus. Tačiau programuojant, o tuo labiau mokantis programuoti labai svarbu galvoti apie tai, kaip programoje sukurti ir naudojami elementai (kintamieji, duomenų struktūros, klasės ir pan.) sąveikauja su anksčiau sukurtaisiais bei kaip jie ateityje galės sąveikauti su kitais. Taigi, svarbu ugdyti loginį ir algoritminį mąstymą – gebėjimą analizuoti realaus gyvenimo problemas, jas išskaidyti į smulkesnius uždavinius, argumentuoti sprendimo būdų pasirinkimus bei sintezuoti rezultatus norint gauti bendrą problemos sprendimo rezultatą. Išugdyti mąstymo įgūdžiai padeda lengviau suprasti programavimo esmę bei leidžia produktyviau mokytis programuoti (Mayer ir kt., 1986).

Šiuo metu pasaulyje egzistuoja šios pagrindinės programavimo paradigmos:

- Vizualusis programavimas;
- Objektinis programavimas;
- Struktūrinis programavimas;
- Kliento-serverio ir interneto;
- Modelių (šablonų) kūrimo;
- Lygiagretumo ir vienalaikiškumo;
- Loginis programavimas;
- Funkcinis programavimas.

Bruce (2004) siūlė tris pradedančiųjų programuoti mokymo požiūrius:

- 1) mokyti objektų po to, kai besimokantieji išmoksta būtinas konstrukcijas;
- 2) pirmiausiai mokyti objektų, tam panaudojant įvairius pedagoginius įrankius;

3) pradedančiųjų kurse mokyti struktūrinio (procedūrinio) ar funkcinio programavimo kalbas, o objektinį požiūrį palikti papildomam kursui.

Gries (2008) išreiškė griežtą nuomonę, jog pradedančiuosius mokyti programuoti privaloma pradėdant nuo objekcinio programavimo. Kitoks požiūris sutinkamas Moritz ir Blank (2005) bei Bennedsen ir Caspersen (2004) darbuose. Abiejų darbų autoriai siūlo pirmiausiai mokyti objekcinio programavimo konceptualiam modeliavimui ir kūrimui panaudojant modeliavimo kalbos UML elementus.

Sebesta (2012) apibrėžia 3 kriterijus ir 9 charakteristikas, padedančias pasirinkti programavimo paradigmą (1 lentelė):

1 lentelė. Mokslininkų siūlomi programavimo kalbos, kuri būtų tinkama mokyti, kriterijai

	Skaitomumas	Rašomumas	Patikimumas
Sintaksės konstrukcijos	✓	✓	✓
Valdymo struktūros	✓	✓	✓
Duomenų tipai	✓	✓	✓
Paprastumas/ortogonalumas	✓	✓	✓
Abstrakcija		✓	✓
Išraiškingumas		✓	✓
Tipo tikrinimas			✓
Išimčių tvarkymas			✓
Ribota vaizdo iškreiptis			✓

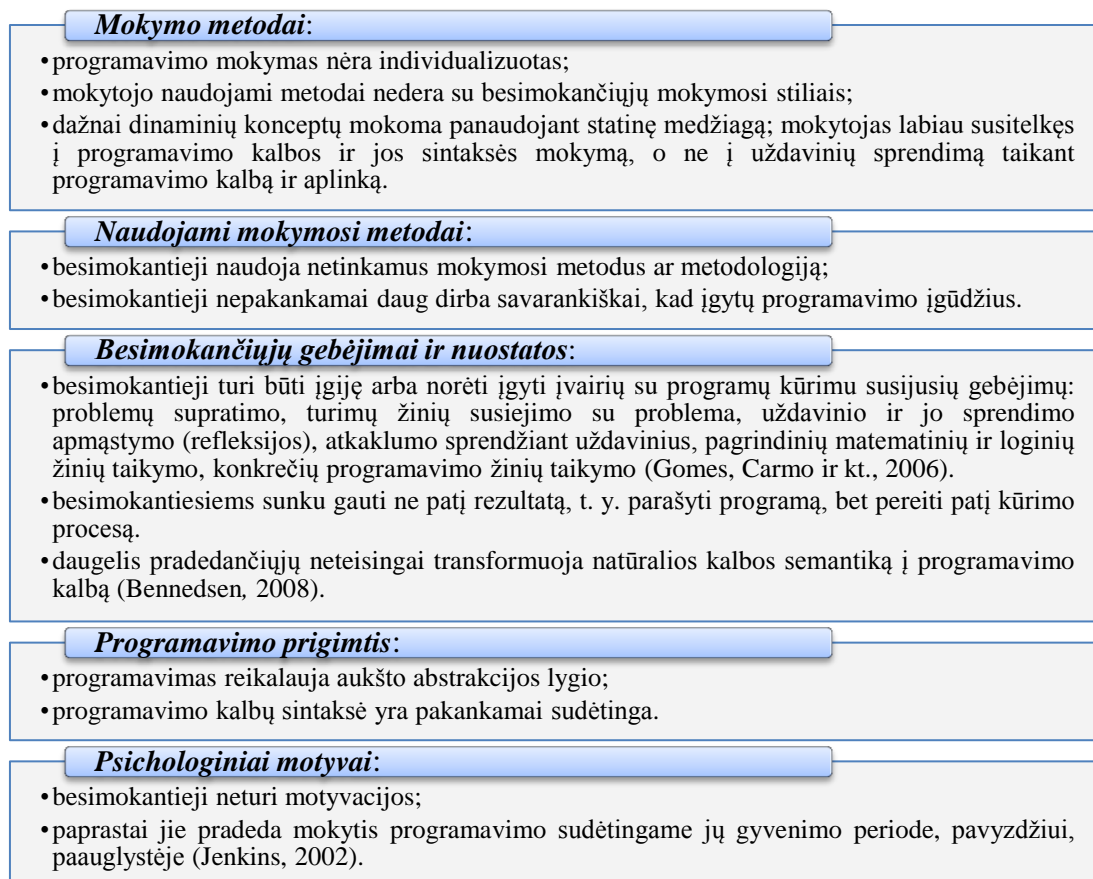
2.1.3. Kodėl sunku išmokyti programuoti

Programavimas – sudėtingas intelektinis procesas, tad norint juo sudominti besimokančiuosius, reikia jį pateikti paprastai ir patraukliai. Tačiau kad ir kaip patraukliai būtų mokoma, vien įgyjamų žinių ir pateikiamų pavyzdžių neužtenka. Reikia ir pačiam įsijungti į procesą, lavinti programavimo įgūdžius bei logiškai mąstyti. Pažinimo teorijų šalininkai bandė atsakyti į klausimą, kodėl daugelis nesugeba išmokyti programuoti? Jie pastebėjo, kad:

- sunku suprasti programų tikslus bei jų ryšį su kompiuteriu;
- sunku suprasti programavimo kalbos sintaksę ir semantiką (Robins, Rountree J., Rountree N., 2003);
- įgytas neteisingas programavimo konstrukcijų suvokimas;
- nesugebama spęsti problemas (Jenkins, 2002; Gomes, Mendes, 2007);
- nesugebama skaityti ir suprasti programos tekstą (Lister ir kt., 2004);

Mannila, 2007).

Giliau patyrinėjus programavimo mokymo problemas išsiskiria penki komponentai, kurie lemia programavimo mokymo sunkumus: 1) mokymo metodai, 2) mokymosi metodų naudojimas, 3) besimokančiųjų gebėjimai ir nuostatos, 4) programavimo prigimtis, 5) psichologiniai motyvai (3 pav.).



3 pav. Programavimo sunkumus nulemiančios priežastys

Siekiant išsiaiškinti besimokančiųjų požiūrį į programavimo mokymą buvo atliktas žvalgomasis tyrimas (naudotas anketinės apklausos metodas), kuriame dalyvavo skirtingos programavimo kursą išklausiusių respondentų grupės. Apklausos dalyviais buvo 29 respondentai studijuojantys Informacinių sistemų technologijų studijų programą, 13 – studijuojančių Programavimo inžinieriaus specialybę, 13 – studijuojančių Informatikos specialybę, 12 studijuojančių Matematikos ir informatikos specialybę ir 9 – studijuojantys Programavimą ir internetines technologijas. Iš viso buvo apklausta 76 studentai. Beveik pusė respondentų, t. y. 48,68 % visų apklaustųjų, teigė, jog išmokti programuoti yra sunku, 39,47 % apklaustųjų nurodė, kad išmokti

programuoti nėra sunku, o likusieji 11,84 % negalėjo apsispręsti. Tačiau nuomonės skyrėsi priklausomai nuo būsimos darbinės srities. Studentai, kuriems dirbant pagal įgytą specialybę ateityje teks programuoti, daugiau rinkosi atsakymą, kad išmokti programuoti nėra sunku (88,46 %), tuo tarpu tie, kuriems ateityje neteks programuoti, daugiau teigė, jog išmokti programuoti yra sunku (56 %). Paprašius įvardinti priežastis, dėl kurių, jų manymu, yra sunku išmokti programuoti, respondentai buvo pakankamai savikritiški ir nurodė, jog sunkumus nulemia tai, jog besimokantieji nepakankamai daug dirba savarankiškai (taip teigė 44,74 % respondentų) ir tai, jog jiems trūksta motyvacijos (47,37 %). Taip pat buvo įvardinamos tokios priežastys, kaip programavimo kalbų sintaksės sudėtingumas (30,26 %), kad besimokantiejiems sunku gauti ne patį rezultatą (parašyti programą), bet pereiti kūrimo procesą (36,84 %). Apklauso metu nustatyti ir besimokančiųjų mokymosi poreikiai: mokymasis kai mokantysis konsultuoja individualiai (70,9 %); mokymasis naudojant technologijas (49,1 %) (Jadzgevičienė, Urbonienė, 2013).

Kad ir koks sudėtingas būtų programavimo mokymasis, tačiau jis yra naudingas besimokančiajam. Mokymasis programuoti ugdo loginį mąstymą ir dėl jo keičiasi besimokančiųjų mokymosi stilius. Mokslininkų tyrimai rodo, kad programavimas formuoja gebėjimą išskirti svarbiausius užduoties, problemos sprendimo etapus: analizę – užduoties išskaidymą į atskiras dalis; plano sudarymą – kiekvienos dalies sprendimo atskyrimą; sintezę – atskirų dalių apjungimą.

Norint, kad programavimo mokymasis taptų patrauklus ir paprastas reikia tiek mokančiojo, tiek besimokančiojo pastangų, nuostatų, palankaus nusiteikimo, glaudaus tarpusavio bendradarbiavimo. Taip pat labai svarbu yra pa(si)rinkti tinkamas – mokymosi atžvilgiu efektyvias – programavimo priemones: kalbą bei aplinką.

2.1.4. Mokomosios programavimo kalbos kriterijai

Programavimo mokymas yra neatsiejamas nuo programavimo kalbos.

Tačiau mokymo sudėtingumas arba paprastumas priklauso nuo to, kokia programavimo kalba pasirenkama mokymuisi. Daug debatų vyksta dėl pačios programavimo kalbos (Renaud, Barrow, le Roux, 2001):

- kaip pasirinkti pirmąją programavimo kalbą;
- kurios programavimo kalbos mokyti, norint išmokyti kelias paradigmas. Mancordis ir kt. (Mancordis, Holt, Penny, 1993) išskiria pagrindines mokymo problemas:
 - įrankių integracijos stoka;
 - įrankių, skirtų įvairioms programavimo veikloms, ne tik programų teksto rašymui, stoka;
 - programavimo kalbų ir įrankių gausa.
- kiek programavimo kalbų gali besimokantysis mokytis.

Per pakankamai trumpą laikotarpį (apie 50 metų) buvo sukurta apie 1000 giminingų programavimo kalbų. Apie tai, jog pakankamai svarbus yra programavimo istorijos mokymasis, jau buvo daug diskutuojama paskutiniajame dvidešimtojo amžiaus dešimtmetyje (Pérez-Quiñones, 1998). 1978 metais vien tik JAV buvo naudojama apie 170 programavimo kalbų. 1993 m. Jean Sammet suskaičiavo, jog buvo sukurta apie 1000 programavimo kalbų, iš kurių 400-500 realizuotos ir tik ne daugiau kaip 300 buvo naudojamos (Sammet, 1996). Tačiau mokymui yra naudojamos tik kelios programavimo kalbos.

Norint pasirinkti programavimo kalbą, pirmiausiai reikia nustatyti tam tikrus kriterijus, kurie kalbą daro gera. Renkantis gerą kalbą reikėtų atsižvelgti į ateities perspektyvas, todėl programavimo kalba turi būti tokia, jog (Lawlis, 1997):

- jos sintaksė nepriklausytų nuo ypatingų techninių priemonių ar nuo konkrečios operacinės sistemos;
- sintaksė būtų standartizuota ir kompiliatoriai derėtų su šiais standartais;
- ji palaikytų programinės įrangos kūrimo technologijas, apribodama ar net uždrausdama prastus metodus, ir skatintų programų tobulinimo ir priežiūros darbus;

- ji veiksmingai palaikytų tam tikrą taikymo sritį (-is);
- ji palaikytų tam tikrą sistemos patikimumo ir saugumo lygį;
- jos kompiliatorių realizacija derėtų su esama technologijų būsena;
- būtų prieinami tinkami programinės įrangos konstravimu paremti įrankiai ir aplinkos.

Laikytis nurodytų kriterijų yra pakankamai sunku, kadangi skirtingose situacijose gali tekti naudoti skirtingas kalbas. Tačiau pažeidus šiuos dalykus, atsiranda papildomi rizikos faktoriai, kurie susiję su artimiausio laikotarpio plėtra bei ateities technologijų pokyčiais. Pagrindinės rizikos, su kuriomis susiduriama pažeidus kriterijus, yra:

- jeigu kalba yra priklausoma nuo konkrečios platformos, atsiranda mobilumo (perkeliamumo) problemų pavojus. Techninės ir programinės įrangos galimybės taip pat tampa ribotos, tiek diegiant originalią sistemą, tiek ateities atnaujinimus;
- jeigu kompiliatorių realizacija neatitinka standartinės kalbos sintaksės, sukuriama unikalūs programiniai sprendimai. Tai taip pat kelia pavojų mobilumui (perkeliamumui) bei apsunkina ateities atnaujinimų galimybes;
- jeigu programinės įrangos kūrimui naudojami netinkami metodai, ir juos bus leidžiama naudoti programos kūrimo ir derinimo metu, tada dėl blogų programos teksto savybių jos testavimas ir priežiūra taps košmariškais;
- prastas taikymo srities atitikimas apsunkins kūrimą, taip pat programos teksto aiškumą ar skaitomumą;
- jeigu sistemos patikimumas atsidurs pavojuje, ji ne tik atliks mažiau nei buvo tikėtasi, tačiau ji taps žymiai brangesnė per visą savo gyvavimo laikotarpį. Jeigu iškyla grėsmė sistemos saugumui, tai jos gyvavimas ir savybės taip pat atsiduria pavojuje;
- pasenusį kompiliatorių yra „žema“ ir sunku naudoti, kuriant nestandartines programas. Jas sudėtinga kurti ir palaikyti. Pasenęs kompiliatorius taip pat gali trukdyti naudoti pagrindines kalbos funkcijas;

- atitinkamų automatizuotų kūrimo priemonių trūkumas komplikuoja kūrėjo produktyvumą ir sistemos kokybę.

Aukščiau išvardintus kriterijus turėtų tenkinti programavimo kalbos, kurios nori neatsilikti nuo šios dienos technologinių sprendimų ir ateityje išlikti naudojamos bei paklausios. Šiuos kriterijus turėtų tenkinti ir mokymuisi skirtos programavimo kalbos. Tačiau mokymuisi skirtos programavimo kalbos turi ir savitų, su besimokančiuoju ir mokymo tikslais susijusių, kriterijų, kurie toliau ir aptariami.

Mokomosios programavimo kalbos kriterijus nagrinėjo daugelis mokslininkų (Kölling, Koch, Rosenberg, 1995; Milbrandt, 1993; Parker, Chao, Ottaway, Chang, 2006; Mannila, de Raadt, 2006). Reikia pasakyti, kad autorių pasirinkti mokomosios programavimo kalbos kriterijai yra gana skirtingi, tik nedaugelis visiškai sutampa (2 lentelė).

2 lentelė. Mokslininkų siūlomi programavimo kalbos, kuri būtų tinkama mokyti, kriterijai

Eil. Nr.	Mokymui tinkamos programavimo kalbos kriterijai	Kölling, Koch, Rosenberg (1995)	Milbrandt (1993)	Parker, Chao, Ottaway, Chang (2006)	Mannila, De Raadt (2006)
1	Naudojama ne tik mokymui			+	+
2	Tinka mokyti įvairiais lygiais			+	+
3	Tvarkingos, paprastos konstrukcijos	+			
4	Prasmingi baziniai žodžiai, leidžia vartoti ilgus prasmingus vardus		+		
5	Palaiko pagrindines objektinio programavimo koncepcijas	+		+	
6	Programavimo konstrukcijos leidžia išvengti klaidų programose	+			
7	Reikalauja aprašyti tipus ir kintamuosius	+	+		
8	Prasmingai sudarytos programavimo konstrukcijos	+			
9	Nepriklauso nuo operacinės sistemos			+	
10	Palengvina programų vykdymą	+			
11	Paprasta, aiški, nuosekli sintaksė	+	+	+	
12	Nedaug programavimo konstrukcijų, jos nedubliuojamos	+	+		
13	Leidžia atlikti sudėtingus skaičiavimus		+		
14	Leidžia programas skaidyti į gabalus			+	+
15	Siūlo vientisą programų kūrimo sistemą			+	+
16	Patikima, saugi, efektyvi			+	+
17	Lengvai išplečiama			+	+
18	Lengva pereiti prie kitų kalbų	+			

Eil. Nr.	Mokymui tinkamos programavimo kalbos kriterijai	Kölling, Koch, Rosenberg (1995)	Milbrandt (1993)	Parker, Chao, Ottaway, Chang (2006)	Mannila, De Raadt (2006)
19	Numato pagalbą kuriant programą	+			+
20	Skatina rašyti iš karto teisingas programas				+
21	Greitas grįžtamasis ryšys		+		
22	Turi patogią ir paprastą programavimo aplinką	+		+	
23	Integruota programų kūrimo aplinka				+
24	Programos kuriamos ne tik rašant tekstą				+
25	Sudaro galimybę programoms tobulinti				+
26	Skatina mokytis programinės įrangos galimybių taikymą				+
27	Lengvai išmokstama	+	+		
28	Paprastas įvedimas ir išvedimas		+		
29	Turi patogius programų derinimo įrankius		+	+	
30	Gali tikti įvairiems įtaisams valdyti				+
31	Turi priemonių programavimui tinkluose			+	
32	Yra nemaža naudotojų bendruomenė				+
33	Patinka mokytojams			+	
34	Integruota su aplinka, visiems prieinama			+	
35	Integruota aplinkoje, yra atvira, kiekvienas gali prisidėti prie jos plėtojimo			+	
36	Yra parengta geros mokymosi medžiagos			+	+
37	Nereikalauja finansinių išlaidų			+	

Mokslininkams Mannila, de Raadt ir Parker buvo svarbu, kad kalba būtų tinkama mokymui įvairiais lygiais. Tačiau ji tuo pačiu turėtų būti naudojama ne vien tik mokymo tikslais (1, 2 kriterijai)¹. Pastaraisiais metais programuotojų tarpe vyrauja objektinio programavimo paradigma, todėl Kölling, Koch ir Rosenberg savo straipsnyje kalba apie reikalavimus pirmajai mokymuisi skirtai objektinei programavimo kalbai. Autoriai nurodo, kad kalba turi palaikyti pagrindines objektinio programavimo koncepcijas (5), nes objektinės konstrukcijos yra ne papildomos kitų galimų struktūrų dalys, bet pagrindinės programavime naudojamos abstrakcijos. Taip pat mokantis labai

¹ Čia ir toliau pateikiami lentelėje „Mokslininkų siūlomi programavimo kalbos, kuri būtų tinkama mokytis, kriterijai“ nurodytų kriterijų eilės numeriai

svarbios yra naudojamos programavimo kalbos koncepcijos bei sintaksė. Jos turi būti kuo paprastesnės, kuo aiškesnės. Tai akcentuoja ir Milbrandt (3, 4, 5, 6, 7, 8, 10, 11, 12). Sintaksės minimalumą, paprastumą ir nuoseklumą pabrėžia Parker bei kt. Nuoseklumas ir suprantamumas užtikrina, kad ta pati sintaksė bus naudojama panašioms semantinėms konstrukcijoms ir skirtinga sintaksė bus naudojama skirtingoms konstrukcijoms. Kalboje neturėtų būti konstrukcijų, kurios susijusios su naudojamos įrangos techninėmis bei programinėmis priemonėmis ir neturi jokios semantinės vertės (11, 12).

Autoriai taip pat mini programavimo kalbos efektyvumą ir lankstumą. Milbrandt svarbu, kad kalba leistų atlikti sudėtingus skaičiavimus (13), tačiau Parker ir kt. bei Mannila su de Raadt pabrėžia, jog svarbi galimybė programą skaidyti gabalais (14), vieningos sistemos užtikrinimas (15) ir pačios kalbos patikimumas ir saugumas (16), išplečiamumas (17). Kölling ir kt. pažymi galimybę pareiti prie kitų plačiai naudojamų programavimo kalbų (18).

Straipsnių autoriai nurodo, jog kalba turi turėti patogią darbui ir paprastą programavimo aplinką, kad besimokantieji galėtų susikaupti programavimo koncepcijų mokymuisi, o ne pačios aplinkos pažinimui (24, 25, 26, 27, 28). Ji turi skatinti rašyti teisingas programas (20), teikti greitą grįžtamąjį ryšį (21) bei pagalbą rašant programas ir užtikrinant jos teisingumą (19). Taip pat aplinkoje turi būti išvystyti kūrimo, derinimo ir programos tikrinimo įrankiai (29, 30, 31).

Kadangi dauguma darbų atliekama žymiai geriau, jei juos daro ne po vieną, o grupės žmonių, todėl Parker ir kt. bei Mannila su de Raadt pabrėžia programavimo kalbos vartotojų bei mokytojų bendruomenių egzistavimo svarbą (32, 33, 37, 38), taip pat ir pačios kalbos prieinamumą (34, 35) bei kainą (37).

Ilgą laiką programavimo mokymui daugelyje pasaulio mokymo institucijų buvo naudojama *Pascal* programavimo kalba. Paskutiniaisiais metais pradėta naudoti daug įvairių kitų kalbų. Tuo pačiu vyksta daugybė diskusijų, kuri programavimo kalba labiausiai tinka mokyti pradedančius programuoti (Duke ir kt., 2000; Gee ir kt., 2005; Gupta, 2004; Bishop, 1997). Mannila ir

de Raadt pateikia 17 kriterijų sąrašą, kuris buvo pasiūlytas keturių vadinamųjų „mokymo programavimo kalbų“ kūrėjams Seymour Papert (*LOGO* kūrėjas), Niklaus Wirth (*Pascal* kūrėjas), Guido van Rossum (*Python* kūrėjas) ir Bertrand Meyer (*Eiffel* kūrėjas) prašant įvertinti jų sukurtąsias kalbas (Mannila, de Raadt, 2006). Pagal šį sąrašą vėliau buvo lyginamos mokymui skirtos programavimo kalbos (3 lentelė).

3 lentelė. Programavimo kalbų palyginimas pagal 17 kriterijų

Programavimo kalba	C	C++	Eiffel	Java	JavaScript	Logo	Pascal	Python	Visual Basic
Kriterijai									
Mokymasis									
1. Tinka mokyti įvairiais lygiais	-	-	+	-	-	+	+	+	-
2. Gali tikti įvairiems įtaisams valdyti	-	-	+	+	+	+	-	+	+
3. Siūlo vientisą programų kūrimo sistemą	+	+	+	+	+	-	+	+	+
4. Skatina programinės įrangos mokymąsi	-	-	+	±	-	+	-	-	-
Programos rašymas ir aplinka									
5. Integruota ir interaktyvi	-	-	-	-	-	+	-	+	-
6. Skatina rašyti iš karto teisingas programas	-	±	+	±	-	-	-	±	-
7. Leidžia programas skaidyti į gabalus	+	+	+	+	+	+	+	+	+
8. Siūlo vientisą programų kūrimo sistemą	-	-	+	±	-	-	-	-	-
Pagalba ir prieinamumas									
9. Yra nemaža naudotojų bendruomenė	+	+	+	+	+	-	-	+	+
10. Integruota su aplinka, yra atvira, kiekvienas gali prisidėti prie jos plėtojimo	-	-	-	-	-	-	-	+	-
11. Yra nuolat palaikoma įvairiose aplinkose	+	+	+	+	+	+	+	+	-
12. Integruota aplinkoje, visiems prieinama	+	+	+	+	+	+	+	+	-
13. Yra parengta geros mokymosi medžiagos	-	+	+	+	-	+	+	+	+
Tinkamumas programuotojams specialistams									
14. Naudojama ne tik mokymui	+	+	+	+	+	-	-	+	+
15. Lengvai išplečiama	+	+	+	+	-	-	-	+	+
16. Patikima, saugi, efektyvi	+	+	+	+	+	-	+	+	+
17. Programos kuriamos ne tik rašant tekstą	-	+	+	+	+	+	-	+	+
Iš viso:	8	11	15	14	9	9	7	15	9

Visi 17 kriterijų suskirstyti į 4 stambias dalis pagal sritis: 1) mokymąsis (1-4 kriterijai)², 2) programos rašymas ir aplinka (5-8), 3) pagalba ir prieinamumas (9-13) bei 4) taikymas (14-17). Mokymosi atžvilgiu iš analizuojamų kalbų tinkamiausia yra *Eiffel* kalba. Ji tenkina visus keturis

² Čia ir toliau pateikiami lentelėje „Programavimo kalbų palyginimas pagal 17 kriterijų“ nurodytų kriterijų eilės numeriai

kriterijus (1, 2, 3 ir 4), t. y. tinkama mokymui, gali būti naudojama pritaikant fizinius analogus, siūlantį bendrą sistemą bei skatinanti programinės įrangos mokymąsi kuriant programas. Tuo tarpu beveik visos likusios programavimo kalbos netenkina arba tinkamumo mokymui (1), arba skatinimo mokytis programinę įrangą kuriant (4) kriterijų.

Patogiausia kūrimui bei turinti patogiausią aplinką taip pat yra *Eiffel* programavimo kalba. Ši kalba yra vienintelė, kuri turi vientisą programų kūrimo aplinką (8). Be to, *Eiffel* programavimo kalba yra vienintelė kalba, skatinanti rašyti teisingą programą (6), nors klaidų tikrinimas su atskirų programos dalių testavimu įmanomas ir *C++*, *Java* ir *Python* programavimo kalbose. Visos nagrinėjamos kalbos leidžia skaidyti rašomą programą ir derinti bei vykdyti atskiras programos dalis (7), o tai irgi yra mokymuisi pakankamai svarbus kalbos kriterijus.

Kalbant apie programavimo kalbos draugiškumą ir prieinamumą (3-joji kriterijų dalis), čia pirmauja *Python* kalba. Ji tenkina visus 5 šios dalies kriterijus. Iš lyginamųjų kalbų su integruotomis aplinkomis tik *Python* yra atviroji (10). Na, o kaip pačią „nedraugiškiausią“ kalbą galima būtų įvardinti *VisualBasic*. Nors ši kalba turi nemažą vartotojų bendruomenę (9) bei yra parengta daug geros mokymosi medžiagos (13), tačiau ji yra mokama ir programos kūrimo metu nėra vykdomas besimokančiojo palaikymas.

Žiūrint į kalbas iš taikomumo ne vien mokymosi tikslams taško, tai tik *Logo* ir *Pascal* kalbos nėra tinkamos kitose srityse. Jos tenkina tik po vieną iš keturių šios dalies kriterijų. *Logo* kalba programą galima kurti ne tik tekstą įvedant klaviatūra (17), o *Pascal* – įvardinama kaip patikima ir efektyvi (16). Likusios 7 kalbos puikiai tinka taikymui ne ugdymo srityje.

Iš pateiktos palyginimo lentelės galima pamatyti, kad daugiausiai kriterijų tenkina *C++*, *Eiffel*, *Java* ir *Python* programavimo kalbos. Jos turėtų labiausiai tikti mokymui. Tačiau *Eiffel* programavimo kalba nėra populiari įvairiose taikymo srityse. Be to, *Pascal* kalba, nors ir tenkinanti pakankamai nedaug kriterijų, vis dar yra populiari įvairiose šalyse (Lietuvoje, Vokietijoje ir kt.) mokant programavimo. Ši kalba plačiai naudojama mokymui, tačiau

bėgant laikui, nebuvo taip plėtojama kaip kitos programavimo kalbos. C++ kalba ilgą laiką yra tarp pačių populiariausių, tačiau jos sintaksė yra pakankamai sudėtinga, todėl ji labiau tinkama programavimo mokymui aukštojoje mokykloje.

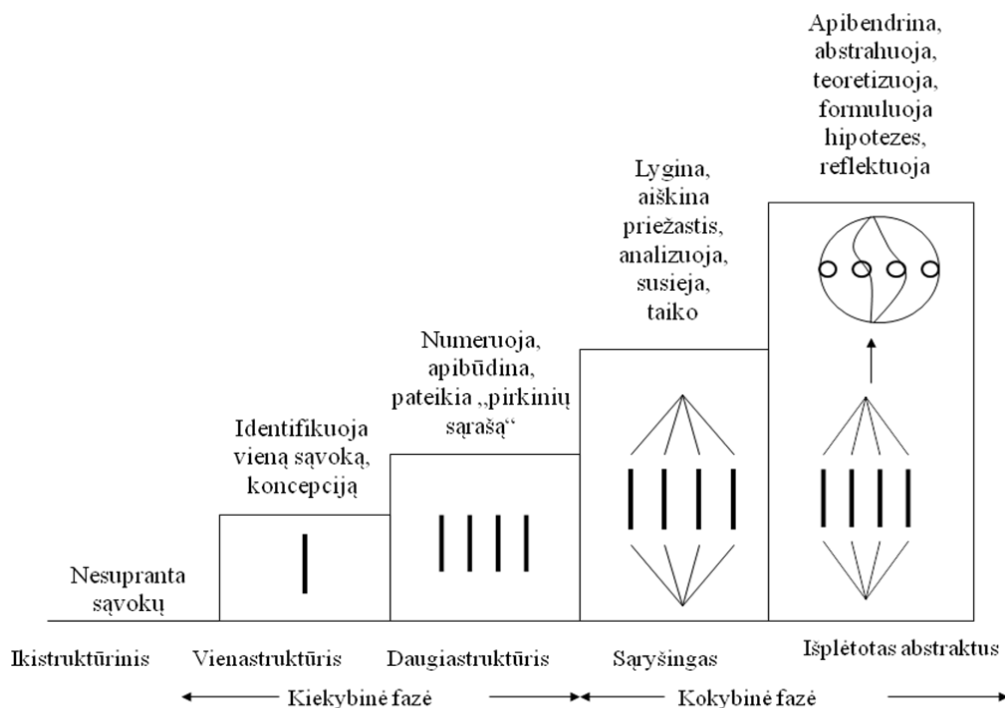
Norint mokytis programavimo, pirmiausiai reikia apsibrėžti kodėl jo norima mokytis. Ar mokomasi norint įgyti programavimo įgūdžių bei ugdytis algoritminį mąstymą, ar norima savo ateitį sieti su programavimu – kurti įvairias pavienes tam tikros srities taikomąsias aplikacijas, ar norima išmokti programuoti plačiai naudojama programavimo kalba ir įsilieti į programuotojų profesionalų gretas bei kurti sisteminę programinę įrangą, ar norima programuoti svetaines bei vis populiarėjančias mobiliąsias aplikacijas. Nuo viso to labai priklauso programavimo kalbos pasirinkimas. Galima mokytis programuoti aukštesnio lygio algoritmine programavimo kalba (pvz., *Java*, *C++*), labai aukšto lygio programavimo kalba (pvz., *APL*, *Mathematica*), specifinės srities programavimo kalba (pvz., *HTML*), assemblerio kalba, makro assemblerio kalba, skriptavimo kalba (pvz., *PHP*, *Python*).

2.1.5. SOLO ir Bloom'o taksonomijų taikymas programavimui mokytis

Norint įvertinti besimokančiojo žinių ir gebėjimų lygį yra pasitelkiamos įvairios taksonomijos. Dažniausiai yra naudojamos pasiekimų lygio *SOLO*³ ir pažinimo srities *Bloom*'o (1956) taksonomijos.

SOLO taksonomijos hierarchinė struktūra leidžia nustatyti besimokančiųjų supratimo sudėtingumo lygį ir įvertinti mokymosi kokybę (4 pav.). Tačiau *SOLO* taksonomija gali būti naudojama ne tik mokymosi pasiekimų vertinimui, ji taip pat naudinga konstruojant mokymosi kursą, kuris yra paremtas mokymosi pasiekimais (angl. *Learning Outcomes*), t. y. dalykinių kompetencijų ugdymu.

³ angl. *Structure of the Observed Learning Outcomes*



4 pav. SOLO taksonomijos lygmenys (Biggs, 2003)

Lister ir kt. (2006) panaudojo *SOLO* taksonomiją aprašydami pradedančiųjų programuotojų programavimo srities supratimą. Autoriai tai nusako penkiais *SOLO* lygmenimis:

1. *Ikistruktūrinis*. Kalbama apie nedidelės dalies programos teksto skaitymą ir suvokimą. Besimokantysis dar nelabai skiria esminių programavimo konceptų, klaidingai supranta esmines programavimo konstrukcijas, atkartoja kitų nuomonę. Pavyzdžiui, besimokantysis painioja masyvo indeksą su pačiu masyvo elementu (t. y. turi neteisingą indekso supratimą).

2. *Vienastruktūrinis*. Besimokantysis parodo tam tikrą, bet ne visų, problemos aspektų teisingą supratimą. Kai besimokantysis turi dalinį supratimą, jo pateikiamos prielaidos yra tarsi išsilavinusio toje srityje.

3. *Daugiastruktūrinis*. Besimokantysis demonstruoja visų problemos dalių supratimą, tačiau neturi ryšio tarp šių dalių supratimo – besimokantysis nesugeba matyti, kad iš medžių gaunamas miškas. Pavyzdžiui, besimokantysis gali rankiniu būdu užrašyti programos kodą ir gauti reikiamą galutinę duoto kintamojo reikšmę, bet nesuprasti, ką ta programa atlieka;

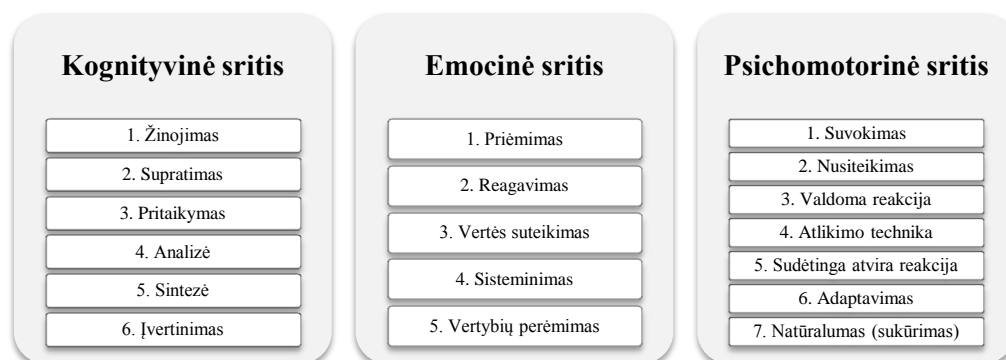
4. *Sąryšingas*. Besimokantysis sujungia problemos dalis į vientisą struktūrą ir

panaudoją šią struktūrą uždavinio sprendimui – besimokantysis mato mišką. Pavyzdžiui, išanalizavęs visą programos kodą besimokantysis gali pasakyti, ką ta programa daro. Jam nebereikia rankiniu būdu analizuoti programos kodo. Lister ir kt. taip pat pabrėžia, kad dauguma reliacinių atsakymų prasideda kaip daugiastuktūriniai. Besimokantysis iki tol, kol supranta programos kodą, jį rašo rankiniu būdu, ir pateikia atsakymą neužbaigdamas iki galo rankinio programos kodo;

5. *Išplėtotas abstraktus*. Tai aukščiausias *SOLO* lygmuo. Besimokančiojo atsakas peržengia spręstiną problemą ir susieja problemą su platesniu kontekstu. Pavyzdžiui, besimokantysis gali pakomentuoti galimus apribojimus ar būtinas sąlygas, kurios turi būti įvykdytos, kad programa veiktų teisingai.

Pažinimo srities *Bloom*‘o taksonomija leidžia pajauti skirtumus tarp būdų, kuriuos taiko pradedantieji ir patyrę spęsdami problemas (uždavinius). *Bloom*‘as apibrėžia mokymosi lygmenis, kuriuos reikia pereiti palaipsniui (Bloom, 1956). *Bloom*‘o taksonomijoje mokymosi tikslai ir juos nusakantys mokymosi rezultatai klasifikuojami pagal tris sritis, turinčias savus lygmenis (5 pav.):

1. Kognityvinę (pažintinę), apibūdinančią žinojimą ir intelektualinių įgūdžių vystymą.
2. Emocinę, apibūdinančią vertybes ir požiūrius.
3. Psichomotorinę, apibūdinančią fizinius judesius, koordinaciją, motorinius mokėjimus ir įgūdžius, padedančius suformuoti konkrečiai profesijai reikalingus praktinius mokėjimus.



5 pav. Kognityvinė, emocinė ir psichomotorinė *Bloom*‘o taksonomijos sritys ir lygmenys

Pažinimo (kognityviosios) srities tikslai yra susiję su intelektualiais procesais – žinojimu, supratimu, atpažinimu, mąstymu analizuojant, sintetinant ar vertinant žinias. Klasikinėje Bloom'o sudarytoje klasifikacijoje išskirti 6 galimi kognityvinių tikslų lygiai: žinojimo, supratimo, taikymo, analizės, sintezės, vertinimo (5 pav.). Nemažai mokslininkų nagrinėjo Bloom'o taksonomijos pritaikomumą kompiuterių mokslų sritims. Vieni analizavo taksonomijos taikymą programavimui mokytis (Lister, Leaney, 2003), kiti tyrė ar ši taksonomija iš viso gali būti taikoma kompiuterių moksluose (Johnson, Fuller, 2007), dar kiti didžiausią dėmesį skyrė taksonomijos taikymui vertinant žinias (Thompson ir kt., 2008; Starr, Manaris, Stalvey, 2008). Galima teigti, kognityvinių tikslų lygmenys yra glaudžiai siejami su programavimo srities pažinimo psichologija. Kognityvinės srities kiekvieno lygmens programavimo mokymosi veiklas galima matyti 6 paveiksle.

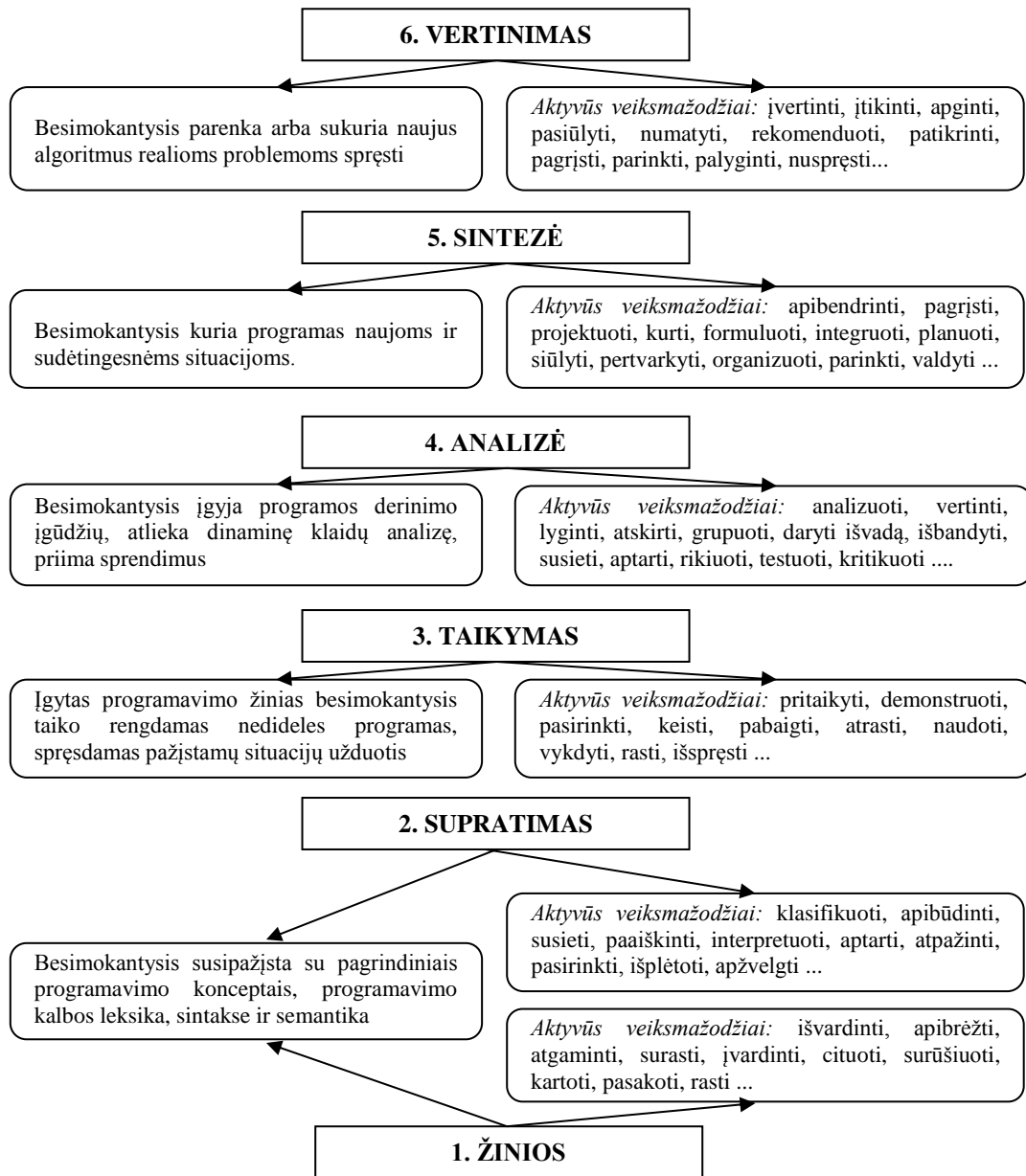
Žinojimas pasiekiamas atmintimi (žinoti – tai prisiminti faktus, sąvokas, metodus ir kt.). Formuluoiant žinojimo lygmens tikslus iš besimokančiojo tikimasi, kad mokymosi pabaigoje jis žinotų: faktus; tam tikros srities terminologiją; tam tikrus metodus; principus ir apibendrinimus it kt.

Supratimo lygmuo pasireiškia, kai besimokantysis sugeba interpretuoti žiniomis ir mokėjimais, perpasakoti savais žodžiais, gautai informacijai suteikdamas kitokią formą arba atpažindamas ją kita forma. Formuluoiant supratimo lygmens tikslus, siekiama parodyti, kad besimokantieji mokymosi pabaigoje turi: paaiškinti tai, ką žino; interpretuoti ką žino; apibūdinti, kaip jie sužinojo ir kt.

Taikymas reiškia sugebėjimą panaudoti sąvokas, taisykles, dėsnius vienokiomis ar kitokiomis konkrečiomis aplinkybėmis. Formuluoiant taikymo lygmens tikslus, siekiama išreikšti gebėjimą panaudoti tam tikrą teoriją ar informaciją naujoje situacijoje.

Analizavimo lygmenyje besimokantieji demonstruoja gebėjimus atskirti, klasifikuoti, palyginti, sugretinti elementus, ryšius, priklausomybes ir pan. Šio lygio tikslais siekiama nustatyti, ar besimokantieji mokymosi pabaigoje gali analizuoti: konkrečios srities reiškinius; tam tikros srities ryšius ir santykius;

skirstymo, skaidymo ir klasifikavimo loginius veiksmus; organizacinius principus.



6 pav. Bloom'o taksonomijos taikymas programavimui mokytis

Žinių *sintezės* atveju besimokantysis sugeba sudaryti, papasakoti, perduoti, sukurti, suformuluoti planus, hipotezes, jas derinti tarpusavyje. Šiame lygmenyje siekiama nustatyti besimokančiųjų gebėjimą dirbti su atskiromis dalimis, sujungiant jas į visumą, sudarančią iki šiol nežinomą struktūrą ar modelį.

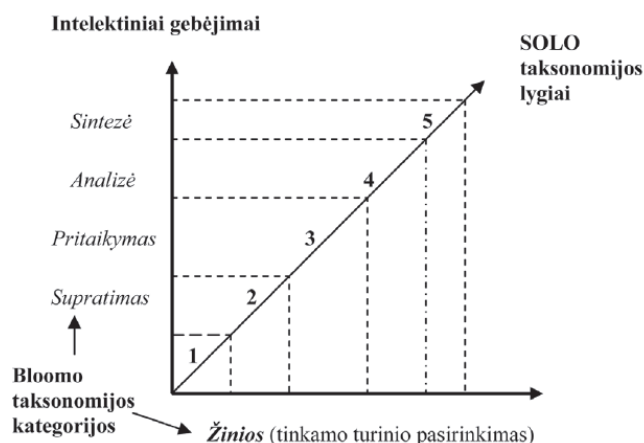
Vertinimo lygmenyje (arba apibendrinimo lygmenyje) besimokantysis geba kokybės ir kiekybės požiūriu spręsti apie tai, kiek koks nors turinys

atitinka kokius nors kriterijus. Šio lygmens tikslais siekiama ugdyti gebėjimą argumentuoti, palyginti priešingus požiūrius ir argumentus, daryti sprendimus ir kt. Be to, norima nustatyti, kad besimokantieji mokymosi pabaigoje sugebėtų priimti sprendimus remdamiesi vidiniais samprotavimais, demonstruodami vertybines nuostatas ar mąstydami abstrakčiai.

Besimokantysis programavimo turi pradėti nuo pirmojo ir antrojo lygmenų – žinių ir supratimo – ir mokytis pagrindinių kompiuterio programų konceptų (sąvokų, principų), taip pat programavimo kalbos sintaksės ir semantikos. Vėliau jis dažniausiai susiduria su sunkumais pereidamas į kitą lygmenį – taikymą. Šiame lygmenyje naujokai turi taikyti savo faktines žinias kurdami programas įvairių pažįstamų situacijų kontekste. Ketvirtasis lygmuo – analizės – taip pat sudėtingas besimokantiems. Šiame lygmenyje jie turi pasiekti kodui derinti reikalingą supratimo lygį, t. y. dinamiškai (programos vykdymo metu) analizuoti klaidas ir pradėti kurti sprendimus mažiau žinomoms situacijoms. Penktasis lygmuo – sintezės – gerokai viršija naujoko programuotojo sugebėjimus. Šiame lygmenyje programuotojas turi sukurti sprendimus naujoms ir sudėtingoms situacijoms. Šeštasis lygmuo – vertinimas – yra skirtas ekspertams. Šiame lygmenyje ekspertai gali sukurti naujus algoritmus arba gali pasirinkti tarp alternatyvių platformų, algoritmų ir kt., atsižvelgiant į jų tinkamumą problemos sprendimui.

Taigi galima daryti išvadą, kad pradedančiajam programuotojui sudėtinga tapti ekspertu nepereinant kiekvieno iš šešių lygmenų.

Bloom‘o ir *SOLO* taksonomijos yra tarpusavyje susijusios (7 pav.): *SOLO* taksonomijos nurodomi intelektiniai gebėjimai iš esmės yra tie patys *Bloom*‘o taksonomijos pažinimo procesai. Tik *SOLO* taksonomijoje atskirai nenurodomas aukščiausias lygis – vertinimas. Pirmieji trys lygiai atitinka kiekybinį etapą, likusieji – kokybinį. Mokymosi proceso organizavimui reikšmingesnė yra *Bloom*‘o taksonomija.



7 pav. SOLO ir Bloom'o taksonomijų sąryšis (Targamadžė, Nauckūnaitė, 2009)

Mokant programavimo būtina remtis bendraisiais pedagogikos ir psichologijos dėsniais, atsižvelgti į specifinius reikalavimus, numatyti problemas. Todėl programavimui mokytis vertinga taikyti Bloom'o, SOLO ir kitas pažinimo mokslų taksonomijas bei pagal jas derinti mokymo turinį.

Kuriant APMS naudojamą programavimo mokymosi kursą bus remiamasi Bloom'o taksonomija, o SOLO taksonomija tiesiogiai nenaudojama. Rengiant mokomojo dalyko aprašą būtina atsižvelgti, jog dalykinėms kompetencijoms formuluoti turi būti naudojami 6 paveiksle įvardinti Bloom'o taksonomijos lygmenis atitinkantys aktyvūs veiksmažodžiai.

2.1.6. Programavimo mokymo turinys

Vienas svarbiausių programavimo mokymosi tikslų – kurti programinę įrangą, taikomąsias programas, tai būtina turėti omeny formuojant mokomųjų programavimo dalykų turinį. Daugybė mokslininkų diskutuoja apie tai, koks turėtų būti pradedančiųjų programuoti mokymosi kursas, kokių dalykų mokytis. Inicijuojami įvairūs projektai BABELnot (Lister ir kt., 2012), BRACElet (Whalley, Clear, Lister, 2007), kurių metu bandoma nustatyti sąryšius tarp įvairiuose mokymo institucijose (universitetuose) naudojamų programavimo kursų, besimokančiųjų įgytų kompetencijų. Tai tampa aktualu tiek konstruojant naują mokymosi kursą, tiek siekiant įskaityti jau išklaustytus kursus, kai besimokantysis migruoja iš vienos mokymo institucijos į kitą.

Programavimo pagrindiniai konceptai, kuriuos reikia išmokti įvardijami:

- Pasaulinėse informatikos mokymo programose (*Computer Science Curriculum 2008*, CS2008 ir *Computer Science Curriculum 2013*, CS2013), parengtose ACM/IEEE-CS (*Association for Computing Machinery, IEEE Computer Society*, 2008; 2012) (ištrauka pateikta 1 priede);
- Informatikos standarte (*CSTA K–12 Computer Science Standards 2011*), parengtame kompiuterijos mokytojų asociacijos (*Computer Science Teachers Association, Association for Computing Machinery*, ištrauka pateikta 1 priede).

Pasaulinėse informatikos programose įvardijamos pagrindinės temos, kurių turėtų būti mokoma bei šias temas atitinkančios dalykinės kompetencijos (mokymosi rezultatai). Siūloma dalyko *Programavimo pagrindai* sandara: Pagrindiniai konstruktai; Algoritminis problemų sprendimas; Duomenų struktūros; Rekursija; Įvykiais grįstas programavimas; Objektinis programavimas; Informacijos saugumo pagrindai; Saugus programavimas. Kurse *Algoritmai ir sudėtingumas* siūloma mokyti: Pradinė analizė; Algoritmavimo strategija; Pagrindiniai algoritmai; Paskirstyti algoritmai; Pagrindiniai skaičiavimai; P prieš NP; Automatų teorija; Sudėtinga analizė; Kriptografiniai algoritmai; Geometriniai algoritmai; Lygiagretūs algoritmai.

Bet kuriam mokymosi kursui aprašyti būtina žinoti kursą sudarančias temas bei tas temas atitinkančias siektinas dalykines kompetencijas. Toliau būtina apibrėžti mokomojo dalyko elementų tarpusavio sąsajas.

2.1.7. Programavimo mokymo strategijos

Pasak Teresevičienės, Oldroyd ir Gedvilienės (2004), *efektyvus mokymasis* yra veikla, kurioje dera proto, kūno, aplinkos ir konteksto veikla. Mokymosi sėkmė priklauso nuo to, kaip maksimaliai pasiekiami mokymosi tikslai – įgyjamos reikiamos žinios bei įgūdžiai, bei kokias emocijas mokydami patiria besimokantieji. Mokymosi sėkmę didžia dalimi lemia mokymosi efektyvumas, kuris priklauso nuo: noro mokytis, kurį gali nulemti

mokymuisi naudojama sistema, ir mokėjimo mokytis, kurį gali įtakoti individualizuojamos mokymosi veiklos.

Išanalizavus įvairių edukologijos srities mokslinę literatūrą galima sakyti, jog mokymasis gali būti efektyvesnis kai yra:

- motyvuojantis mokymosi kontekstas;
- gerai struktūruota žinių bazė;
- mokomasi vis naujų dalykų;
- mokomasi skirtingais būdais ar metodais;
- asmenybės tipą atitinkanti besimokančiojo veikla;
- besimokančiojo supratimas apie tai, kaip yra geriau mokytis (t. y. kokį mokymosi stilių naudoti);
- mokantis naujų dalykų pasinaudojama jau išmoktais anksčiau;
- ieškoma galimybės naują informaciją susieti su ankstesne patirtimi – diskutuojama, užduodami klausimai;
- besimokantysis kūrybiškai aktyvus;
- skatinamas besimokančiojo savarankiškumas bei atsakomybės jausmas;
- įgyjama praktinės patirties;
- kur įmanoma naudojamos vaizdinės priemonės;
- praktika yra refleksyvi;
- kitas asmuo mokomas to, ką išmoko pats besimokantysis;
- surandamas atsakymas dar prieš tai, kai pasidaro sunku įsiminti (t. y. atsakymas surandamas greitai);
- formuojantis grįžtamasis ryšys;
- konstruktyviai reflektuojama, stebima sava patirtis;
- bendraminčių būrys, su kuriais aptariama, diskutuojama;
- mokymosi aplinka yra vieninga ir vientisa.

Yra 2 vienas kitą papildantys programavimo mokymo požiūriai (Haridi, Van Roy, 2004):

1. Skaičiavimais paremtas mokymo požiūris. Šiuo atveju programavimas pristatomas kaip būdas atrasti problemos mašininį realizavimą;

2. Logika parentas mokymo požiūris. Programavimas pristatomas kaip matematinės logikos šaka.

Besimokančiuosius reikia mokyti (Renaud, Barrow, le Roux, 2001):

- įvaldyti pagrindinius programavimo įgūdžius;
- gebėjimo logiškai mąstyti ir spręsti problemas;
- realizuoti tų problemų spendimus;
- gebėjimų sintezuoti idėjas;
- būti profesionaliu bei etišku;
- gebėjimų aptartis savo idėjas su kitais;
- noro ir gebėjimų kurti bei mokytis ateityje.

Yra išskiriama nemažai programavimo kalbų mokymo strategijų. Galima rasti skirtingus įvairių autorių pateikiamus skirstymus.

Papp-Varga išskiria 7 programavimo kalbų mokymo strategijas (Papp-Varga, 2008):

1. Sakiniais grindžiamas. Į kalbą žiūrima kaip į sakinių (teiginių) rinkinį ir kiekvieno elemento mokoma tam tikra tvarka;
2. Kalbos kaip įrankio naudojimas. Pvz. mokant programavimo ir duomenų bazių valdymo, didžiausias dėmesys teikiamas duomenų bazių mokymui, o vėliau reikiamu lygiu pasitelkiami programavimo kalbos įrankiai;
3. Programinės įrangos technologijomis grindžiamas. Programavimo kalbos mokymo metodai pritaikomi prie programinės įrangos kūrimo metodų bei technologijų, kurie ir nulemia kalbos ar kelių kalbų pasirinkimą;
4. Užduotimis grindžiamas. Naujų programavimo kalbos žinios inicijuojamos kai sprendžiant užduotis susiduriama su įvairiomis problemomis;
5. Kalba grindžiamas mokymas. Kalba yra kaip struktūruotas vienetas. Perteikiama programavimo kalbos logika ir būtinu lygiu bei tvarka pateikiami konkretūs kalbos elementai;
6. Veiksmu grindžiamas mokymas. Kalbos teiginių mokoma stengiantis perprasti kaip sakiniai operuojama, t. y. stengiamasi išsiaiškinti, kas nutiks, kai sakiniai bus vykdomi – paprastesniais atvejais veiksmi gali būti formuojami kita žinoma programavimo kalba;

7. Pavyzdžiais grindžiamas mokymas. Programavimo kalbos mokoma analizuojant tam tikrų užduočių pavyzdžius.

Haridi ir Van Roy išskiria įžanginio programavimo mokomojo dalyko modelius (Haridi, Van Roy, 2004):

- pirmiausiai procedūrinis programavimas;
- pirmiausiai objektinis programavimas;
- pirmiausiai funkcinis programavimas;
- pirmiausiai loginis programavimas;
- pirmiausiai aparatūra;
- pirmiausiai algoritmai;
- pirmiausiai sąvokos;
- pirmiausiai platus užmojis.

Aalto universitete programavimo mokoma taikant 7 žingsnių probleminį mokymą (Kinnunen, Malmi, 2005; Nuutila ir kt., 2008):

- *Atidarymo sesija:*
 1. Atvejo nagrinėjimas. Grupė susipažįsta su pateikto atvejo medžiaga.
 2. Problemos identifikavimas. Nustatoma pradinė atvejo problema.
 3. Idėjų svarstymas. Besimokantieji pateikia savas asociacijas bei idėjas apie problemą ir taip stengiasi nustatyti, kas jau yra žinoma ir kaip tai yra susiję su jau turimomis žiniomis. Idėjos išsakomos garsiai ir užrašomos ant baltos lentos užklijuojant užrašus.
 4. Nupiešiamas aiškinamasis modelis. Yra apibrėžiama pradinė problemos paaiškinimo versija ir identifikuojami svarbiausi konceptai bei jų tarpusavio ryšiai.
 5. Iškeliami mokymosi tikslai. Identifikuojamos tos aiškinamojo modelio dalys, kurios yra paslaptingos, neaiškios arba tiesiog mažiau žinomos, ir svarbiausios iš jų pasirenkamos kaip grupės mokymosi tikslai.
- *Mokymosi periodas:*

6. Savarankiškas mokymasis. Kiekvienas besimokantysis, siekdamas įgyvendinti visus mokymosi tikslus, mokosi savarankiškai. Ši fazė apima informacijos rinkimą ir dažniausiai nemažas skaitymų apimtis.

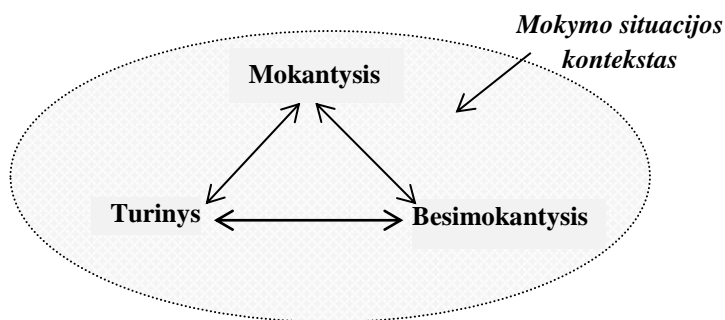
- *Uždarymo sesija:*

7. Diskusija apie išminktą medžiagą. Apsirūpinusi naujomis žiniomis, grupė susirenka iš naujo aprašti atvejus. Diskusijos metu paaiškinami pagrindiniai konceptai ir mechanizmai, analizuojama medžiaga ir įvertinamas jos pagrįstumas ir svarba.

Nagrinėjant programavimo mokymo klausimus vis aktualūs lieka klausimai kaip galima mokyti programavimo neprisirišant prie dabartinių priemonių ir kalbų? Kaip galima mokyti programavimo kaip vieningo mokomojo dalyko, turinčio mokslinį pagrindą?

Kiekvienoje mokymo situacijoje egzistuoja 3 pagrindiniai komponentai:

1) besimokantysis, 2) mokantysis ir 3) turinys (8 pav.). Visi šie trys komponentai sudaro taip vadinamąjį didaktinį trikampį (jį pirmasis apibrėžė Johann Friedrich Herbart, vokiečių filosofas, psichologas, o vėliau savo darbuose analizavo Pertti Kansanen ir Matti Meri (1999) bei kt.). Todėl modeliuojant mokymo ar mokymosi aplinką labai svarbu apibrėžti mokymo kontekstą, nes jis leidžia vienareikšmiškai susieti mokomojo dalyko elementus su konkrečia pedagogine situacija, palengvina žmogaus intelekto modeliavimą kompiuteryje ir pagerina mokomojo dalyko pritaikymą besimokantiejiems (Brezillon, 2011), taip pat leidžia analizuoti kiekvieną iš didaktinio trikampio komponentų bei šių komponentų tarpusavio ryšius (Berglund, Lister, 2010).



8 pav. **Didaktinis trikampis** (Berglund, Lister, 2010)

Programavimo mokymo kontekste didaktinio trikampio kampas *Turinys* apima programavimo kalbas, metodus bei priemones. Kampas *Besimokantysis* apima besimokančiojo mokymosi stilių (pvz. kokiems dalykams teikiama pirmenybė – vizualiems, tekstui ar pan.) bei motyvaciją (vidinę arba išorinę). Kampas *Mokantysis* nagrinėja mokančiojo vaidmenį mokymo procese (ar mokantysis yra pagrindinis mokymo procese – turinio perteikėjas, ar mokymas nukreiptas į besimokantįjį, o mokantysis yra tik patarėjas). Šiuolaikinės didaktikos tyrėjai pabrėžia, jog šiuolaikinis mokantysis turėtų ne vadovauti, o padėti mokytis, skatinti besimokančiuosius veikti aktyviai ir tokiu būdu įgyti naujų žinių. Vis dažniau mokančiuoju tampa kompiuterinės technologijos. Nagrinėjant pagrindinių komponentų santykius *Turinio ir mokančiojo* svarbu yra ko mokoma (mokantysis pats turi suprasti tai ko moko), kokiais metodais mokoma – reikėtų programavimo procesą atskleisti taip, kad jis palengvintų ir skatintų programavimo mokymąsi; mokyti įgūdžių, o ne vien perteikti žinias; sąvokos turi būti tam tikro abstrakcijos lygio; naudojama aiški terminologija; kruopščiai apmąstoma ir atrenkama medžiaga, kad nebūtų naudojamos naujos sąvokos jų nepaaiškinus; daugiau susitelkti į visumą, o ne tik į atskiras detales. *Besimokančiojo ir turinio* santykiuje yra svarbu, kad turinys skatintų besimokančiojo motyvaciją. Išmokimo rezultatai taip pat priklauso ir nuo besimokančiojo mokymosi stiliaus, todėl turinys turi leisti prisitaikyti besimokančiajam – turinys turi būti pritaikomas ne pagal mokytojo poreikius, bet pagal besimokančiojo. *Besimokančiojo ir mokančiojo* santykis apima mokančiojo ir besimokančiojo tarpusavio ryšius mokymo procese. Tas ryšys glaudžiai siejasi su mokančiojo ir besimokančiojo kultūromis. Svarbu išsiaiškinti ir besimokančiojo motyvaciją.

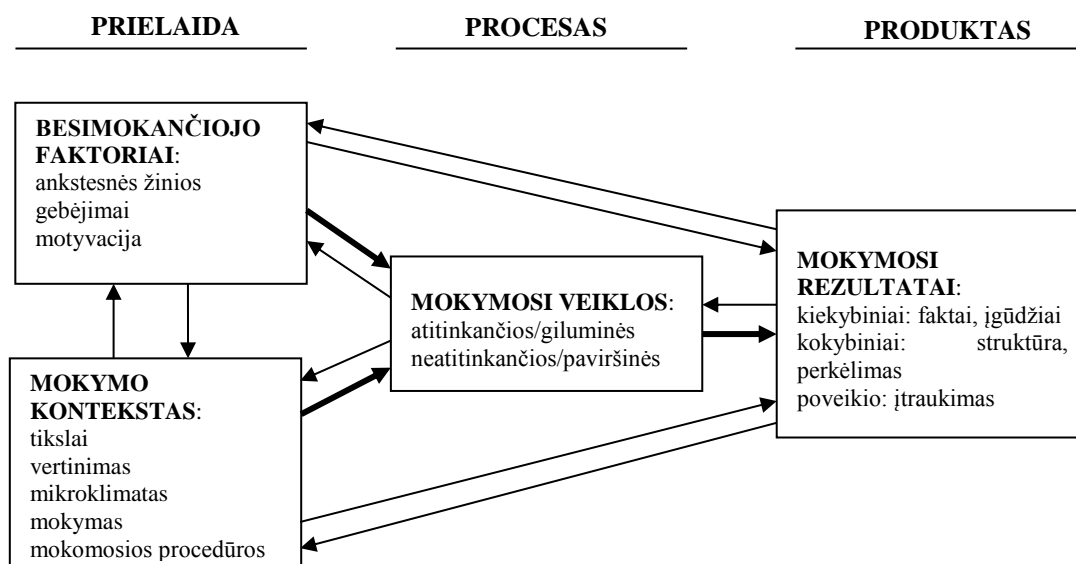
Labai dažnai besimokantiesiems programavimo mokymasis nėra patrauklus, atrodo sunkus, jie neturi motyvacijos. Norint motyvuoti besimokančiuosius reikia pastangų bei tinkamų pagalbinių ir programavimo priemonių pasirinkimo. Taip pat labai svarbus yra glaudus ryšys tarp mokytojo ir besimokančiojo, patrauklus turinio perteikimas.

Biggs (2003) apibrėžia 3P mokymosi modelį (9 pav.), kur:

Prielaida (angl. *Presage*) – situacija esanti prieš mokymąsi;

Procesas (angl. *Process*) – mokymosi eiga;

Produktas (angl. *Product*) – mokymosi rezultatai.



9 pav. 3P mokymosi modelis (Biggs, 2003)

Prielaidos yra dviejų tipų – susijusios su besimokančiuoju (ankstesnės su mokymosi kursu susijusios žinios, besimokančiojo gebėjimai, asmeninės savybės bei susidomėjimas mokomuoju dalyku, dalyko universalumas besimokančiojo atžvilgiu) ir susijusios su mokymosi kontekstu (ko ir kaip turi būti mokoma, kaip turi būti vertinama, mokytojų kvalifikacija, aplinkos draugiškumas ir pan.). Procesas apima mokymuisi naudojamų veiklų identifikavimą bei naudojimą, produkto sritis leidžia numatyti mokymosi rezultatų vertinimo metodus ir priemones. Visi Biggs modelyje apibrėžti faktoriai yra svarbūs parenkant ir programavimo mokymosi priemones, įrankius ir aplinką.

2.2. Egzistuojančių programavimo mokymo įrankių ir aplinkų analizė

Lietuvos mokymo institucijose sutinkamos įvairios programavimo mokymui naudojamos aplinkos. Mokyklose dar vis mokoma programuoti *Pascal* programavimo kalba ir naudojama šią kalbą realizuojanti *FreePascal* aplinka. Aukštosios mokyklos moko programuoti panaudojant *C++*, *Java*,

VisualBasic programavimo kalbas, kai kur pradeda mokyti *Python*. Šioms kalboms naudojamos įvairios aplinkos: C++ – *Borland C++*, *Dev C++*, *VisualStudio*, *CodeBlocks*; Java – *JBuilder*, *Eclipse*, *NetBeans*, *BlueJ*. Taip pat analizei pasirinktos programavimo aplinkos *Alice* bei *Scratch*, kurios leidžia tiesiogiai kurti interaktyvias istorijas, animacinius filmus ir paprastus vaizdo žaidimus. Tai laisvai prieinamos mokymo priemonės, skirtos mokytis objektinio programavimo pradmenų. Jos leidžia besimokantiejiems mokytis pagrindinių programavimo sąvokų kuriant objektus (pvz., žmones, gyvūnus ir transporto priemones), įkuriant juos virtualiame pasaulyje ir animuojant. Programavimo aplinkos buvo analizuojamos jas įdiegus į kompiuterį bei atsižvelgiant į anksčiau 2.1.7. skyrelyje aprašytus mokymosi efektyvumo kriterijus. Visi kriterijai suskirstyti į 5-ias kriterijų grupes: mokymasis, patogumas, pritaikomumas, pagalba ir pasiekiamumas ir tinkamumas programuotojams specialistams (5 lentelė). Analizė buvo atliekama atsižvelgiant į tai, jog programuoti mokosi pradedantieji.

4 lentelė. Programavimo mokymosi priemonių analizė

Programavimo aplinka Kriterijai	JBuilder	Eclipse	NetBeans	BlueJ	Visual Studio	Borland C++	Dev C++	Codeblocks (GCC)	Free Pascal	Python IDLE	Alice	Scratch
Mokymasis												
1. Tinkamumas mokytis įvairiais lygiais	±	±	±	+	-	-	-	±	+	+	+	±
2. Siūloma vientisa programų kūrimo sistema	+	+	+	+	+	+	+	+	+	+	+	-
3. Skatinamas programinės įrangos mokymasis	-	-	-	+	-	-	-	+	-	-	+	+
4. Skatinamas domėjimasis programavimu	-	-	+	+	-	-	-	+	-	+	+	+
5. Skatinamas kūrybinis aktyvumas	-	-	-	-	-	-	-	-	-	-	+	+
6. Mokymosi stiliaus palaikymas	-	+	+	-	-	-	-	-	-	-	+	+
7. Galimas bendras programų kūrimas	-	+	+	-	-	-	+	-	-	-	-	+
8. Yra bendravimo galimybės	-	-	-	-	-	-	-	+	-	-	-	-
9. Greitas, funkcionalus ir formuojantis grįžtamasis ryšys	±	±	±	±	±	±	±	±	±	±	+	+
10. Skatina rašyti iš karto teisingas programas	±	±	±	±	±	±	±	±	±	±	+	+
11. Integruota mokomoji medžiaga	-	-	-	-	-	-	-	-	-	-	-	±
12. Galima rasti mokymosi pagalbą	+	+	+	+	+	+	+	-	+	+	±	+
Patogumas												
13. Paprastas įdiegimas	+	+	+	+	±	+	+	+	+	+	+	+

Kriterijai	Programavimo aplinka											
	JBuider	Eclipse	NetBeans	BlueJ	Visual Studio	Borland C++	Dev C++	Codeblocks (GCC)	Free Pascal	Python IDLE	Alice	Scratch
14. Interaktyvi	-	-	-	-	±	-	-	-	-	-	+	+
15. Leidžia programas skaityti į gabalus	+	+	+	+	+	+	+	+	+	+	+	+
16. Nepriklausomumas nuo platformos	+	-	+	+	-	-	-	+	+	+	+	+
17. Portatyvumas	+	+	+	+	+	+	+	+	+	+	-	+
Pritaikomumas												
18. Naudotojo sąsaja lietuvių kalba	-	-	-	-	-	-	-	-	+	-	-	+
19. Naudotojo sąsajos lokalizavimo galimybė	+	+	+	+	-	-	+	+	+	-	+	+
20. Laiko zonų ir datų lokalizavimas	-	-	-	-	-	-	-	-	+	-	-	-
21. Alternatyvių kalbų pritaikymas	-	-	-	-	-	-	+	-	+	-	-	+
22. Kelių programavimo kalbų palaikymas	-	+	+	-	+	-	+	+	-	-	+	-
23. Adaptavimo galimybės	-	-	-	-	-	-	-	-	-	-	-	-
24. Kalbos ir dizaino pritaikomumas	-	-	-	-	-	-	-	-	-	-	-	-
25. Individualizavimo galimybės	±	±	±	±	±	±	±	±	±	±	±	±
Pagalba ir pasiekiamumas												
26. Yra nemaža naudotojų bendruomenė	-	+	+	+	±	±	±	+	±	+	+	+
27. Yra atvira, kiekvienas gali prisidėti prie jos plėtojimo	-	+	+	+	-	-	+	+	+	+	+	+
28. Visiems prieinama	-	+	+	+	-	-	-	+	+	+	+	+
29. Yra parengta geros mokymosi medžiagos	+	+	+	+	-	+	-	-	+	+	+	+
Tinkamumas programuotojams specialistams												
30. Naudojama ne tik mokymui	+	+	+	+	+	+	-	-	-	+	-	-
31. Lengvai išplečiama	-	+	+	-	+	+	-	+	-	-	-	-
32. Patikima, saugi, efektyvi	±	+	+	+	+	+	±	±	-	-	+	-
33. Programos kuriamos ne tik rašant tekstą	+	-	-	-	+	-	-	-	-	-	+	+
Iš viso +/-:	9/5	10/4	18/4	16/3	9/6	9/4	10/5	14/5	14/4	13/3	20/2	21/3

Iš analizuotų programavimo mokymo aplinkų tinkamiausios *mokymo* atžvilgiu ir tenkinančios daugiausiai šios grupės kriterijų yra *NetBeans*, *Codeblocks*, *Allice* ir *Scratch* aplinkos. Šios aplinkos turi priemones, leidžiančias mokytis nuosekliai pereinant įvairius *SOLO* ar *Bloom*’o taksonomijų lygmenis (1)**, siūlo vientisą programų kūrimo sistemą (2), kas mažiau blaško besimokančiuosius, skatina domėjimąsi tiek programine įranga, tiek pačiu programavimu (3, 4), bei skatina teisingų programų rašymą (10). Visos aplinkos programuojant praneša apie klaidas, tačiau pateikiamas

** Čia ir toliau skaičiai skliausteliuose nurodo kriterijų numerius pagal lentelę „Programavimo mokymosi aplinkų analizė“

grįžtamasis ryšys ne visuomet akivaizdžiai suprantamas bei tinkamai interpretuojamas (9). Norint ištaisyti klaidas, dažnai reikia būti įgijus daugiau programavimo įgūdžių, kad galima būtų greitai nustatyti tas klaidas bei jas ištaisyti. Tiesa, visos nagrinėtosios aplinkos turi pagalbos sistemą – žinynus (12), tačiau tie žinynai yra anglų kalba. Iš 4 aplinkų lyderių mokymosi atžvilgiu tik *Codeblocks* nėra tinkama įvairių mokymosi stilių atstovams, kadangi aplinkoje nėra pagalbos sistemos, kur galima būtų rasti pavyzdžių bei juos analizuoti (šios veiklos tinkamos analitikams bei sistemintojams). Tuo tarpu šioje aplinkoje yra nuoroda į diskusijų forumus, o tai sudaro galimybes dirbti bendraujant (8). *NetBeans* yra galimybė kurti bendras programas, pasinaudojant tinklu, o *Scratch* sukurtas programas tiesiai iš aplinkos galima publikuoti interneto svetainėje ir taip dalintis jomis su kitais (7).

Analizuojant aplinkas *patogumo* atžvilgiu pastebėta, jog visos aplinkos yra labai lengvai įdiegiamos (13), leidžia programas skaidyti į gabalus (15) bei yra portatyvios aplinkų versijos (17), kas leidžia greitai ir paprastai sukurtas programas persikelti į keičiamas laikmenas bei esant reikalui paleisti jas neįdiegiant į kompiuterį atitinkamos programavimo aplinkos.

Negalima būtų išskirti nei vienos aplinkos, kuri tinkama aplinkos *pritaikomumo* atžvilgiu. Nors beveik visos aplinkos gali būti lokalizuojamos (19), o *Dev C++*, *Free Pascal* ir *Scratch* jau yra lokalizuotos keliomis kalbomis (18, 21), tačiau nei viena aplinka neturi adaptavimo (23, 24) galimybių, kurios didžia dalimi įtakoja besimokančiojo mokymosi motyvaciją.

Beveik visos mokymosi aplinkos turi nemažas naudotojų bendruomenes (26), yra visiems prieinamos (28), atviros (27), todėl bet kas gali prisidėti prie jos plėtojimo. Daugumai aplinkų yra parengta geros mokymosi medžiagos (tiesa, ne visoms lietuvių kalba). Iš nagrinėtųjų aplinkų tik *Eclipse*, *NetBeans*, *VisualStudio* bei *Borland C++* gali būti tinkamiausios programuotojams profesionalams (30, 31, 32, 33).

Apibendrinant galima matyti, jog iš viso daugiausiai kriterijų tenkina *NetBeans*, *Alice* ir *Scratch* aplinkos. *Scratch* mokymosi aplinka yra skirta jaunesnio mokyklinio amžiaus vaikų mokymui ir visiškai nenaudojama

programuotojų profesionalų, todėl ji labai tinkama supažindinant besimokančiuosius su programavimo principais. Vėlesniam programavimo mokymuisi tiktų *Alice* (tinka programuoti *C++*, *Java* ir *C#* programavimo kalbomis) arba *NetBeans* (tinka *Java*, *C++*, *C*, *PHP*). Tačiau nei viena aplinka neleidžia užtikrinti visiško mokymosi efektyvumo.

Taigi šiuo metu naudojamos programavimo mokymosi aplinkos nėra tinkamos pradedant mokytis programuoti, nes jose nėra galimybės tinkamai (gimtąja besimokančiojo kalba) pateikti mokymosi medžiagą bei gauti greitą, funkcionalų ir formuojantį grįžtamąjį ryšį. Tiesa, už aplinkos ribų (internetinėse svetainėse bei knygose) galima rasti įvairios mokymosi medžiagos (angl. *tutorials*). Aplinkose nėra adaptavimo bei pakankamai ribotos individualizavimo galimybės. Taip pat šios priemonės neleidžia įgyvendinti naujos kartos mokymosi koncepcijos (poreikių), t. y. nėra galimybės mokytis bendraujant ir bendradarbiaujant.

Įvairias programų vizualizavimo priemones (*Basic Programming*, *LOPLE*, *DYNAMOD*, *DynaLab*, *Amethyst*, *Bradman*, *EROSI*, *VisMod*, *Fernández ir kt. įrankis*, *DISCOVER*, *VINCE*, *ORGE*, *JAVAVIS*, *Seppälä įrankis*, *OOP-Anim*, *JavaMod*, *JIVE*, *Memview*, *JavaTool*, *PlanAni*, *Eliot*, *Jeliot*, *GRASP/jGRASP*, *VIP*, *The Teaching Machine*, *ViLLE*, *Jype*, *the Online Python Tutor*, *WinHIPE*, *CSmart*, *Online Tutoring System*) bei jų savybes savo disertaciniame darbe nagrinėjo Sorva (2012). Nagrinėtosios priemonės leidžia vizualizuoti įvairias mokymosi veiklas, tačiau nei viena neturi galimybės individualizuoti mokymąsi atsižvelgiant į besimokančiojo mokymosi stilių.

Atliekant sisteminę mokslinių straipsnių analizę *ACM* skaitmeninėje bibliotekoje, paieškos užklausoje buvo įvesti raktiniai žodžiai programavimas (*programming*) ir adaptyviosios priemonės (*adaptive tools*). Pagal šios užklauskos rezultatus pateikta 29140 mokslo darbų, kuriuose yra nagrinėjami įvairūs programavimo mokymo klausimai (kokią programavimo paradigmą ir kaip mokytis, adaptyvus programavimo konceptų mokymas), pristatomos adaptyviosios sistemos, leidžiančios adaptuoti programavimo mokymosi turinį.

Užklausoje panaudojus raktinius žodius programavimas (*programming*), adaptavimas (*adaptation*) ir mokymosi stiliai (*learning styles*) gauta 3579 mokslo darbai.

2.3. Adaptyviosios technologijos ir mokymosi sistemos

Mokymosi procese naudojamos adaptyviosios technologijos ir adaptyviosios sistemos leidžia pagerinti mokymosi procesą, kadangi mokymosi turinys ir mokymosi strategijos yra pritaikomos prie besimokančiojo galimybių bei poreikių. Siekiant nustatyti mokymosi procese naudojamų adaptyviųjų sistemų tinkamumą toliau yra analizuojamos adaptyviosios technologijos, keturių procesų adaptyvus ciklas bei mokymosi metu sutinkami adaptavimo scenarijai, taip pat aptariamos įvairios adaptyviosios mokymosi sistemos ir adaptyviosios mokymosi sistemos modelis.

Terminas adaptyvus turi įvairių reikšmių. Mokymosi kontekste adaptyvus gali būti suprantamas kaip *padedantis arba galintis prisitaikyti, lengvai pritaikomas arba galintis būti pritaikytas, galintis adaptuotis, pasikeičiantis, siekiant prisitaikyti prie naujos situacijos*.

Adaptyvioji sistema pati prisitaiko prie individualių besimokančiojo charakteristikų ir poreikių. Ši tikslą padeda pasiekti adaptyviosios technologijos, kurias valdo skaičiavimo įrenginiai bei adaptuoja turinį pagal skirtingus besimokančiųjų poreikius ir savybes. Adaptyviųjų technologijų panaudojimo efektyvumas priklauso nuo tikslaus ir informatyvaus besimokančiojo modelio. Adaptyviųjų technologijų galimybės skirtingos:

- vienos technologijos leidžia realiu laiku pateikti mokymo ir vertinimo turinį, kuris prisitaiko prie besimokančiųjų poreikių ir pageidavimų;
- kitos technologijos leidžia simuliuoti dinامينius įvykius, įgūdžius formuojančias papildomas praktines galimybes ar naudoti alternatyvias daugialypės terpės galimybes (dažniausiai besimokantiesiems su negalia).

Mokymosi turinį adaptuoti naudinga, nes yra:

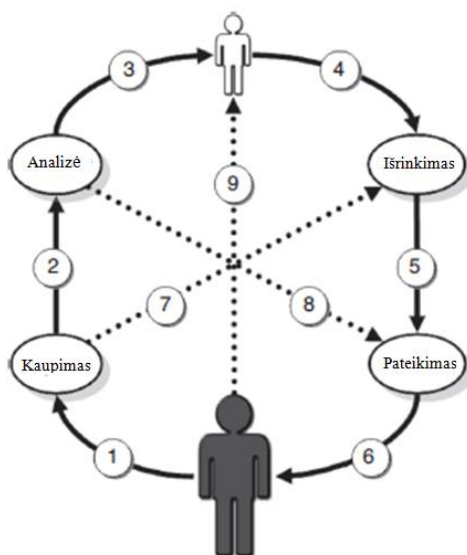
- pradinių žinių, įgūdžių ir gebėjimų skirtumai. Šie skirtumai gali būti labai dideli. Kiekvienas besimokantysis ateina su savais ankstesniais įgūdžiais

bei žiniomis, todėl mokymasis bus efektyvesnis, jei bus atsižvelgiama į jau sukauptą patirtį. Taip pat besimokantieji gali būti su skirtingais gebėjimais ir įvairiomis negaliomis;

- demografiniai ir sociokultūriniai skirtumai. Tai tampa aktualu, kai kartu mokosi įvairių kultūrų besimokantieji;
- poveikio (emociniai) skirtumai. Besimokantieji skiriasi savo emocinėmis būsenomis (nusivylimas, nuobodulys, motyvacija ir pasitikėjimas), kurios įtakoja mokymąsi.

Norint, kad adaptyviosios technologijos leistų pagerinti mokymąsi, reikia labai gerai nustatyti besimokančiojo savybes (pvz., žinias, įgūdžius, motyvaciją, atkaklumą, galimybes). Besimokančiojo informacija vėliau naudojama kaip pagrindas nustatyti optimalų turinį (t. y. pateikti aprašymus, mokomąsias užduotis, patarimus ir pan.). Shute ir Zapata-Rivera (2008) apibrėžia keturių procesų ciklą (10 pav.), leidžiantį besimokančiajam pasiekti atitinkamą mokymosi turinį ir išteklius. Šį ciklą sudaro:

- *Kaupimas* – apima asmeninės informacijos apie besimokantįjį jam sąveikaujant su aplinka rinkimą;
- *Analizė* – šis procesas sukuria ir palaiko tam tikros srities besimokančiojo modelį, vėliau pateikiant informaciją atsižvelgiama į esamos būsenos (gautos kaupimo etape) išvadas;
- *Išrinkimas* – mokymosi medžiaga išrenkama pagal sistemos palaikomą besimokančiojo modelį ir pagal sistemos tikslus (pvz., kitas MO);
- *Pateikimas* – atsižvelgiant į išrinkimo proceso rezultatus besimokančiajam pateikiamas specifinis jo poreikius ir savybes atitinkantis mokymosi turinys. Tinkamai naudojant įvairias terpes, prietaisus ir technologijas informacija efektyviai perduodama besimokančiajam.



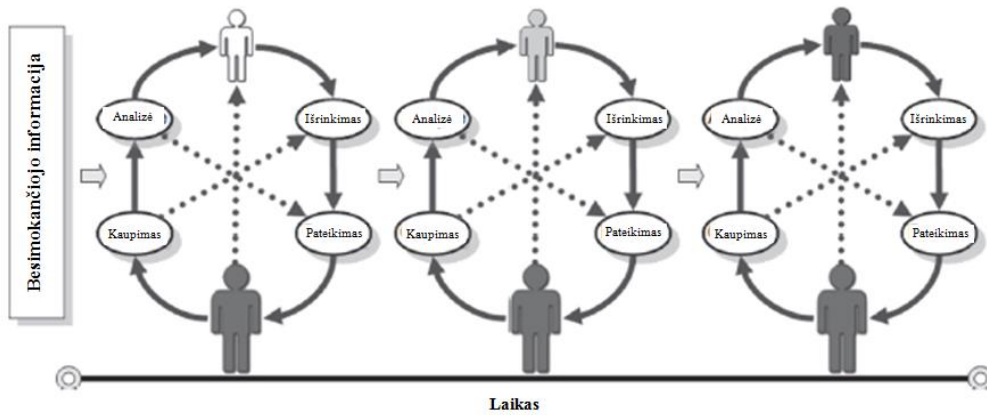
10 pav. Keturių procesų adaptyviųjų technologijų taikymo modelis (Schute, Zapata-Rivera, 2008)

Šis modelis leidžia naudoti įvairius alternatyvius scenarijus. 5 lentelėje pateikta keletas scenarijų, kuriais galima skirtingų tipų adaptacija.

5 lentelė. Keturių procesų adaptyvaus ciklo scenarijai (Schute, Zapata-Rivera, 2008)

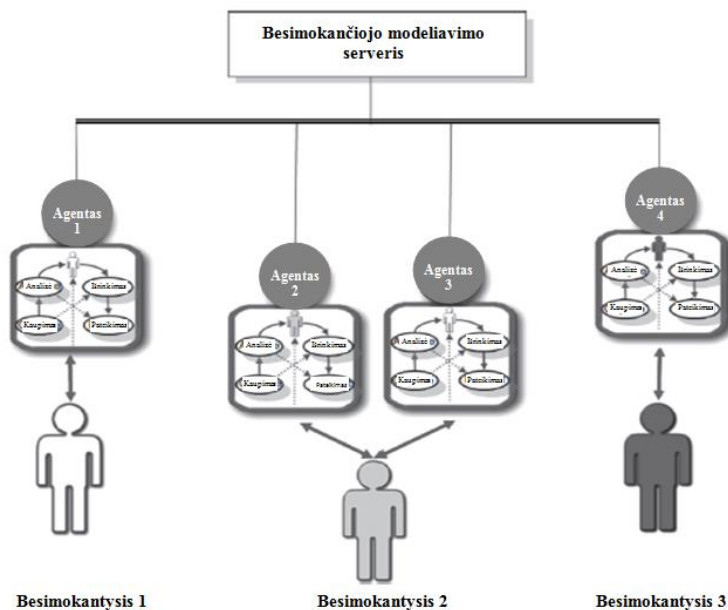
Scenarijus	Aprašymas
Visas ciklas (1, 2, 3, 4, 5 ir 6) (angl. <i>A complete cycle</i>)	Vykdomi visi ciklo procesai: kaupiama aktuali informacija, ji analizuojama, atnaujinami kintamieji, išrenkami tinkami dabartinius besimokančiojo poreikius atitinkantys išteklių ir strategijos, ir atitinkamu būdu jie padaromi pasiekiamais besimokančiajam. Šis ciklas tęsiamas tol, kol pasiekiami visi tikslai.
Modifikuojantis adaptyvus ciklas (1, 2, 3, 4, 5, 6 ir 9) (angl. <i>Modifying the adaptive cycle</i>)	Besimokančiajam leidžiama sąveikauti su besimokančiojo modeliu. Šis sąveikos pobūdis ir poveikis besimokančiojo modeliui gali skirtis. Leidžiant žmogui sąveikauti su modeliu galima sumažinti analizės ir išrinkimo procesų sudėtingumą, sumažinant neapibrėžtumą. Taip pat gali būti naudinga didinant besimokančiojo supratimą ir savirefleksiją.
Stebėjimo kelias (1, 2 ir 3) (angl. <i>Monitoring path</i>)	Besimokantysis yra nuolat stebimas; surinkta informacija analizuojama ir naudojama profilio atnaujinimui (pvz. vidaus saugumo stebėjimo sistema, asmenų analizė profilių rizikos analizės tikslais). Į šį kelią gali būti žiūrima kaip į ciklą, kuris pasibaigia trečioje dalyje, o ne grįžta prie besimokančiojo.
Trumpos (ar laikinos) atminties ciklas (1, 7, 5 ir 6) (angl. <i>Short (or temporary) memory cycle</i>)	Mokymosi turinio išteklių išrinkimas atliekamas panaudojant naujausią informaciją (pvz. einamojo testo rezultatai ar navigacijos komandos). Palaikomas nepastovaus besimokančiojo modelis. Adaptacija atliekama pasinaudojant vėliausios besimokančiojo ir kompiuterio sąveikos informacija.
Trumpos (ar laikinos) atminties be išrinkimo ciklas (1, 2, 8 ir 6) (angl. <i>Short (or temporary) memory, no selection cycle</i>)	Laikomasi iš anksto apibrėžtos mokymosi struktūros. Yra palaikomas ne besimokančiojo (<i>no learner</i>) modelis. Šis iš anksto apibrėžtas kelias diktuoja, kurie mokymosi išteklių ir testavimo medžiaga yra pritaikoma besimokančiajam.

Bėgant laikui ir vis sukantis adaptyviajam ciklui besimokančiojo modelis evoliucionuoja – jis tampa labiau ištobulintas ir tikslesnis (11 pav.).



11 pav. Keturių procesų adaptyviųjų technologijų taikymo modelis (Schute, Zapata-Rivera, 2008)

Dar labiau išplėtotame keturių procesų adaptyviajame modelyje naudojami programiniai agentai (12 pav.). Kiekvienas agentas padeda palaikyti atskiro besimokančiojo nuostatas. Besimokančiojo modelio informacija ir mokymosi ištekliai gali būti paskirstyti skirtingose vietose. Agentai norėdami gauti prieigą prie informacijos, padedančios besimokančiajam pasiekti mokymosi tikslus, komunikuoja tarpusavyje tiesiogiai arba per besimokančiojo modeliavimo serverį.



12 pav. Komunikavimo tarp agentų ir besimokančiųjų schema (Schute, Zapata-Rivera, 2008)

Adaptavimo procese yra naudojama aparatinė įranga bei programinės technologijos. Aparatinė įranga gali būti: biologiniai prietaisai, kalbos įvedimo prietaisai, galvos gestų įvedimo prietaisai ir pagalbinės technologijos. Programinės technologijos – tai programos ar metodai, kurie kaupia, analizuoja, išrenka ir pateikia informaciją. Pagrindiniai jų tikslai yra sukurti besimokančiojo modelį (diagnozės funkcija) ir vėliau panaudoti besimokančiojo modelio informaciją (norminamoji funkcija). Tam tikslui gali būti naudojami įvairūs modeliavimo būdai: kiekybinis, kokybinis arba pažintinis modeliavimas.

Į vieną sistemą integruojant kelias technologijas yra sukuriamos adaptyviosios sistemos. Kuriant adaptyviasias sistemas gali būti adaptuojami skirtingi besimokančiojo ir mokymo kintamieji (6 lentelė). Tai priklauso nuo adaptyviosios sistemos paskirties.

6 lentelė. **Adaptuojami kintamieji** (Schute, Zapata-Rivera, 2008)

Besimokančiojo kintamieji	Mokymo kintamieji
<p>Pažintiniai gebėjimai (pvz. matematikos įgūdžiai, skaitymo įgūdžiai, pažinimo pažangos etapas, problemų sprendimas, analogiškas samprotavimas)</p>	<p>Grįžamojo ryšio tipas (pvz., patarimai, paaiškinimai) ir laikas (pvz., greitas, uždelstas)</p>
<p>Meta pažinimo įgūdžiai (pvz., savęs paaiškinimas, savęs vertinimas, refleksija, planavimas)</p>	<p>Turinio nuoseklumas (pvz., sąvokos, mokomieji objektai, uždaviniai, elementai ir/ar problemos sprendimui)</p>
<p>Emocinės būsenos (pvz., motyvacija, dėmesys, įtraukimas)</p>	<p>Parama (pvz., palaikymas ir išnaikinimas kaip nurodoma, atlygiai)</p>
<p>Papildomi kintamieji (pvz., asmenybė, mokymosi stilius, socialiniai įgūdžiai, tokie kaip bendradarbiavimas, ir suvokimo įgūdžiai)</p>	<p>Medžiagos peržiūra (pvz., apžvalga, išankstinė peržiūra, peržiūra po vizualizavimo tikslų ir / arba sprendimų struktūros)</p>

Adaptyvus mokymasis – tai organizuoto mokymosi, kuris susitelkia ties ankstesniais pasiekimais ir naudoja juos kaip pagrindą konstruojant būsimas veiklas ir pasiekimus, stilius. Pagrindinis adaptyviųjų sistemų taikymo tikslas yra sukurti mokymosi atžvilgiu patikimą ir lanksčią aplinką, kuri remtų besimokančiųjų su įvairiais gebėjimais, negaliomis, interesais, biografijos faktais ir kitomis charakteristikomis mokymąsi. Šio tikslo įgyvendinimas labai

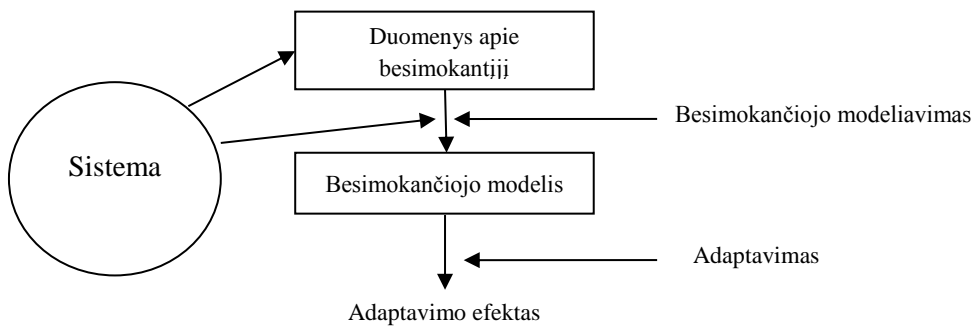
priklauso nuo tikslių konkrečiau besimokančiojo arba besimokančiųjų grupės savybių nustatymo (žinių tipo ir žinių lygio, įgūdžių, asmenybės bruožų, emocinių būsenų) ir sprendimo, kaip sustiprinti informaciją, siekiant pagerinti mokymąsi.

Mokymosi adaptavimas atsižvelgiant į besimokančiojo poreikius ir galimybes yra žinomas jau nuo senų laikų. Tai yra minima Corno ir Show (1986), Federico ir kt. (1980), Glaser (1977), Reiser (1987), Tobias (1989), Wang ir Lindvall (1984) darbuose. Jau 1902 metais Dewey savo rašinyje „Vaikas ir mokymo programa“ smerkia vieno tipo mokymosi programos kūrimo nuostatą, kai yra pateikiama vienoda, nelanksti instrukcijų seka.

Adaptyviųjų mokymosi sistemų kūrimas ir įgyvendinimas galimas penkiais skirtingais požiūriais:

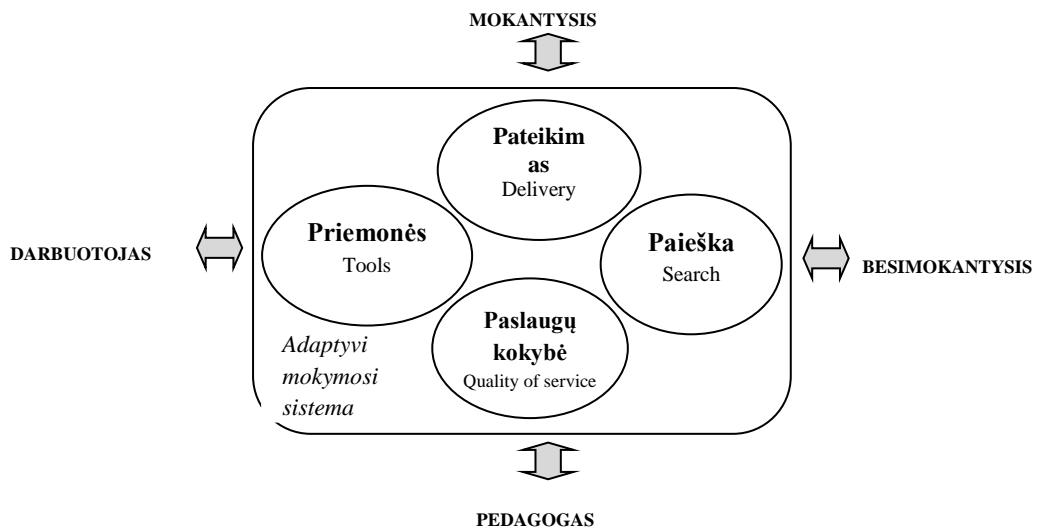
- adaptuoti mokymąsi makro lygiu, leidžiant pasirinkti alternatyvius tikslus, mokymosi programos turinio gilumą, dėstymo sistemas ir pan.;
- specifinėms besimokančiojo savybėms adaptuoti specifines mokymosi procedūras ir strategijas. Kadangi šis požiūris reikalauja, kad būtų identifikuotos besimokančiojo mokymuisi būdingiausios savybės ir pagal jas parinktos geriausios mokymosi strategijos, tai yra grindžiama su gabumų nustatymo sąveika;
- adaptuoti mokymąsi mikro lygiu, nustatant besimokančiojo specifinius mokymosi poreikius ir pateikiant šiuos poreikius atitinkančius nurodymus (instrukcijas);
- naudoti adaptyviasias hypermedines ir saityno (pasiekiamas internetu) sistemas;
- naudoti sistemų grupę, kuri sukurta remiantis specifiniais pedagoginiais požiūriais. Šiose sistemose taikomi konstruktyvizmo, motyvacijos teorijos, socialinio mokymosi teorijos ir metapažinimo pedagoginiai požiūriai.

Brusilovsky ir Maybury (Brusilovsky, Maybury, 2002) apibrėžia 13 pav. pavaizduotos adaptyvios sistemos modelį:



13 pav. **Adaptyvios mokymosi sistemos modelis** (Brusilovsky, Maybury, 2002)

Saityno mokymosi sistemas (pvz., virtualias mokymosi aplinkas), atsižvelgiant į technines priemones, kuriomis jos realizuojamos, galima suskirstyti į keturias kategorijas: turinio, dėstymo, paieškos ir paslaugų kokybės (naudojant teleatvaizdavimo, virtualios realybės, multisensorines technologijas). Tuo remiantis adaptyviają mokymosi sistemą galima pavaizduoti kaip šių keturių kategorijų rinkinį (14 pav., <http://www.atp.nist.gov/atp/97wp-lt.htm>).



14 pav. **Adaptyviosios mokymosi sistemos pagrindiniai elementai** (pagal Harris Liebergot)

Adaptavimas elektriniame mokymėsi remiasi tiek adaptavimo procesu, tiek būsenomis, kurios yra adaptuojamos. Terminas adaptyvus čia yra siejamas su plačiu sistemos savybių ir galimybių asortimentu. Yra išskiriami skirtingi adaptavimo tipai: išankstinis arba reaguojantis, savarankiškas (arba planuotas, bendras arba rinktinis (Moura, 2006).

Mokymosi aplinka laikoma adaptyvia, jeigu ji sugeba (Paramythis, Loidl-Reisinger, 2004):

- stebėti jos vartotojų veiksmus;
- interpretuoti šiuos veiksmus specifinės srities pagrindu;
- pagal interpretuojamus veiksmus nustatyti vartotojo reikalavimus ir nuostatas taip, kad galima būtų juos pavaizduoti asocijuotame modelyje;
- paveikti leistinas žinias taip, kad atsižvelgiant į jos vartotoją ir temos sritį dinamiškai palengvintų mokymosi procesą.

Adaptyviosios elektroninio mokymosi sistemos palaiko prisitaikymą atsižvelgiant į besimokančiajam būdingus bruožus. Brusilovsky adaptyviosiose hipermedia sistemose išskiria tris adaptavimo rūšis: turinio, maketo ir navigacijos (Paramythis, Loidl-Reisinger, 2004). Turinio adaptavimas mokymosi aplinkose priklauso nuo besimokančiojo nuostatų ir asmeninių įgūdžių, taigi pateikiama optimaliai apibrėžta seka. Maketas turi atkreipti dėmesį į pažintinę perkrovą – mokymosi stilių. Navigacijos adaptavimą mokymosi sistemose lemia mokymosi sekos, kurios aprašo mokymosi veiklų tvarką ir organizavimą. Mokymosi stilius čia dar kartą nulemia individualius poreikius nuoseklumo tipui. Pavyzdžiui, besimokantieji-analitikai pageidauja mokytis nuosekliai, tuo tarpu atsitiktiniams besimokantiesiems patinka pasirinkti nuosavą sąveikos kursą.

Dublino Trinity koledžo žinių ir duomenų inžinerijos grupės (*Knowledge and Data Engineering Group*) atstovai išskiria galimus adaptavimo metodus (Conlan ir kt., 2003):

- adaptyvus pristatymas – pateikiamo turinio individualizavimas;
- adaptyvi navigacija – dinamiškai generuojama navigacija ir mokymosi sekos;
- istorinis pritaikymas – pritaikymas laiko kontekste;
- struktūrinis pritaikymas – pritaikymas ir pateikimas erdvėje.

Savo ruožtu Paramythis ir Loidl-Reisinger (2004) išskiria keturias adaptavimo kategorijas:

- adaptyvi sąveika;
- adaptyvus mokomojo dalyko dėstymas;
- turinio atskleidimas ir komponavimas;
- adaptyvus bendradarbiavimo palaikymas.

Esmahi (2009) pabrėžia, jog sistema gali adaptuoti mokymo procesą kelias lygiais:

- mokymosi kurso turinio adaptacija: adaptyvus mokomojo dalyko medžiagos pateikimas įterpiant, pašalinant, rūšiuojant ir užtemdant fragmentus;
- mokymosi kurso navigacijos adaptacija: adaptacija palaikoma nuorodomis – nuorodos slepiamos, rūšiuojamos, uždraudžiamos ar pašalinamos, taip pat mokymosi metu generuojamos naujos nuorodos;
- mokymosi strategijos adaptacija: paskatomis grįstas mokymasis, atvejų analize grįstas mokymasis, probleminis mokymasis;
- sąsajos adaptacija: palaikyti besimokantį per sąsają, kuri jam yra labiausiai patinkanti;
- sąveikos adaptacija: turi būti intuityvi, grįsta besimokančiojo profilio informacija.

Visos adaptavimo kategorijos mokymosi sistemose grindžiamos gerai nusistovėjusių modelių ir procesų seka. Adaptyviosiose mokymosi sistemose gali būti sutinkami srities, besimokančiojo, grupės ir adaptavimo modeliai (Lee, Park, 2008).

Adaptyviosios hypermedinės sistemos susideda iš srities, vartotojo, mokymo modelių ir adaptyvaus variklio (Avgeriou, 2000).

Brusilovsky ir Henze (2007) išskiria pagrindines adaptavimo modelius pagal naudojamą techniką:

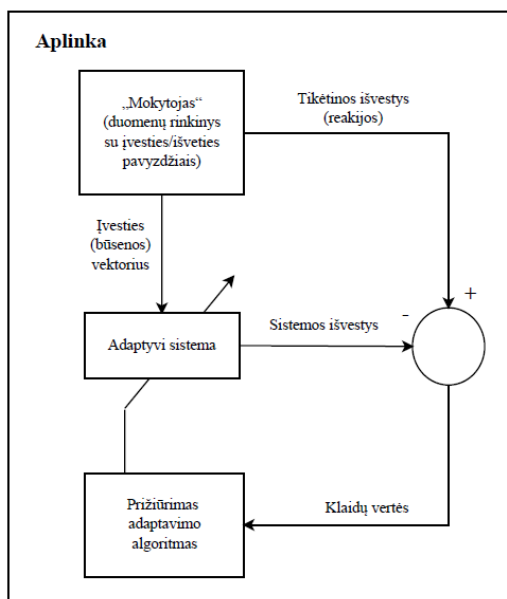
- *Rankinis mokymosi išteklių indeksavimas.* Adaptavimui reikalingus elementus išrenka ir atitinkamuose dokumentuose išsaugo vartotojas, vėliau šie elementai naudojami optimaliai atitiktai rasti ir įvertinti;

- *Automatinis raktiniais žodžiais paremtas indeksavimas.* Adaptavimui reikalingi elementai apibrėžiami raktiniais žodžiais, kurie naudojami optimaliai atitikčiai surasti ir įvertinti;
- *Metaduomenimis paremtas adaptavimas.* Adaptavimui reikalingi elementai apibrėžiami metaduomenimis, kurie naudojami optimaliai atitikčiai surasti ir įvertinti;
- *Bendruomene paremtas adaptavimas.* Kai išvada apie adaptavimui tinkamų elementų optimalią atitiktį remiasi kitų tas pačias charakteristikas turinčių naudotojų patirtimi (Brusilovsky, Henze, 2007)

Adaptyviųjų technologijų bei su jomis susijusių adaptavimo sistemų panaudojimas leidžia mokymosi procese pasiekti įvairių tikslų: tiek padėti įsisavinti mokymosi turinį, tiek padėti atlikti įvairias specifines užduotis, tiek padėti kaupti informaciją apie besimokančiojo mokymosi įpročius, mokymosi seką, pasiekimus, atsirandančias problemas bei sunkumus. Hauger ir Köck (2007) atliko adaptyviųjų mokymosi sistemų *AHA*, *ALFANET*, *ELM-ART*, *ITEM/IP*, *InterBook*, *KnowledgeSea II*, *METHOD*, *NetCoach* ir kt. analizę. Nagrinėtose sistemose adaptyviai pateikiama pagalba ir palaikymas, rodomos arba slepiamos nuorodos. Kai kuriose sistemose, pvz. *KnowledgeSea II*, naudojamas personalizavimas leidžiant besimokantiesiems naršyti ir gauti prieigą prie rekomenduojamos informacijos, kitur pvz., *KnowledgeTree* adaptyvumas palaikomas atsižvelgiant į mokymosi veiklas. Nagrinėtųjų adaptyviųjų sistemų tarpe yra kelios, skirtos mokyti programuoti, pvz. *iWeaver*, skirta mokyti *Java* programavimo kalbos, kai nustačius besimokančiojo asmenines savybes yra pateikiamos personalizuotos rekomendacijos ir atitinkamai vartotojo sąsajoje atvaizduojami prieinami mokymosi įrankiai. Taigi nors jau yra sukurta adaptyviųjų mokymosi sistemų, tačiau nei vienoje iš jų mokymosi medžiaga nėra kuriama automatizuotai arba teikiant mokymą nėra atsižvelgiama į besimokančiojo mokymosi stilių.

Eberhart, Shi (2007) išskiria tris sistemos adaptavimo rūšis:

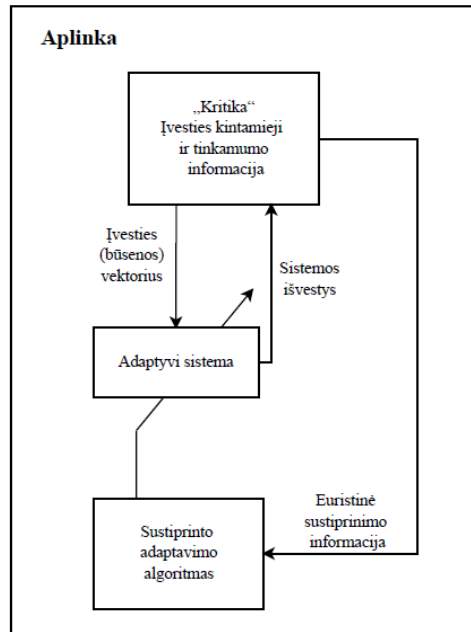
- Prižiūrimas adaptavimas (15 pav.). Šiuo atveju dalyvauja „mokantysis“, kuris pateikia tam tikrus konkrečius pavyzdžius (egzempliorius). Sistemos adaptyvaus algoritmo charakteristikos:
 - adaptuojant vienu metu atliekamas vienas žingsnis (iteracija). Sistema prisitaiko kopijuodama ankstesnius mokymo įvesties/išvesties rezultatus juos apibendrindama;
 - sistemos charakteristikų metrika dažnai yra atvirkščiai proporcinga įvesties/išvesties egzempliorių klaidų sumos funkcijai. Kiekvienam atskiram egzemplioriui saugoma atskiro atvejo kvadratinų nuokrypių nuo grupės vidurkio suma, kvadratinis nuokrypis (skirtumas tarp tikėtinų rezultatų ir tikrųjų rezultatų) ir absoliučių klaidų suma. Prižiūrimo adaptavimo algoritmas naudoja visų egzempliorių vidutinę paklaidą siekiant pritaikyti esamam konkrečiam atvejui.



15 pav. Prižiūrimo adaptavimo schema

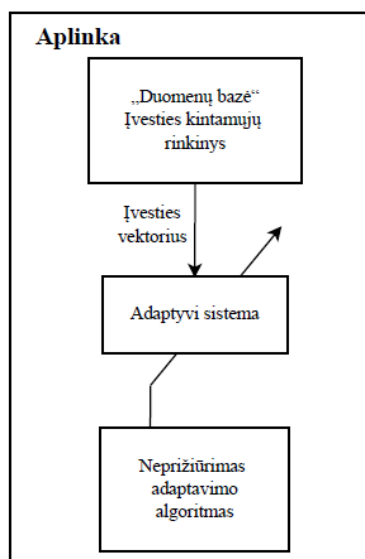
- Sustiprintas adaptavimas (16 pav.). Šiuo atveju adaptavimas vyksta sistemai sąveikaujant su euristicine informacija (tam tikra kritika), kuri sustiprina adaptavimo efektą. Sistemos adaptyvaus algoritmo charakteristikos:

- adaptyvioji sistema dažniausiai komunikuoja su įvesties (būsenos) vektoriais, laukdama kol įvykdoma seka nustatanti sistemos tinkamumą;
- kritika yra susijusi tik su išvesties rezultatais, ir nesiejama su kiekvienos įvesties paklaidos matavimais.



16 pav. Sustiprinto adaptavimo schema

- Neprižiūrimas adaptavimas (17 pav.). Šiuo adaptavimo atveju nėra nei mokačio, nei kritikos. Sistemos adaptyvaus algoritmo charakteristikos:
 - sistema tiesiog vykdo savo darbą, kuris gali būti siejamas su veiklos klasterizavimu arba „kompetentingu“ mokymusi;
 - išvados apie tai, kaip neprižiūrimas adaptyvus algoritmas veikė, kaip tinkamai jis veikė, ar jis yra tinkamas ir naudingas, daromos tik po to, kai algoritmas baigia savo darbą. Šį vertinimą dažniausiai atlieka žmonės arba kitos intelektinės sistemos.



17 pav. Neprižiūravimo adaptavimo schema

Adaptavimui sistemose palaikyti yra kuriami programiniai agentai. Kitame skyrelyje analizuojama programinių agentų samprata, savybės bei tarpusavio ryšiai.

2.4. Programiniai agentai ir agentinės sistemos

Adaptyvioje sistemoje būna įvairių agentų. Agentas – tai esybė, suvokianti aplinką savo jutikliais ir daranti poveikį tai aplinkai savo manipulatoriais. Mokslo darbuose galima rasti daug tyrimų grindžiamų programiniais agentais, tačiau vieningos programinio agento apibrėžties nėra. Dažniausiai apibrėžimai kyla iš agentų panaudojimo pavydžių, realizavimo galimybių, pavyzdžiui:

- *Programinis agentas* – tai programinės įrangos objektas, kuris nepertraukiamai ir autonomiškai funkcionuoja tam tikroje aplinkoje ir gali bendrauti su kitais agentais ar procesais (Shoham, 1997);
- *Programinis agentas* – autonominė programinės įrangos dalis, vykdanči vartotojo nurodymus arba užprogramuotus situacijos sprendimus. Pagrindinė agentų idėja yra ta, kad jie nėra griežtai iškviečiami vykdyti tam tikrą užduotį, tačiau yra patys aktyvūs. Priklausomai nuo tipų jie gali mokytis, daryti išvadas, tarpusavyje komunikuoti siekdami bendrų tikslų, keisti vietą (Lupeikienė, 2007);

- *Intelektualus agentas* nuolat atlieka tris funkcijas: pastoviai stebi aplinką ir fiksuoja (suvokia) jos būsenas; veikia taip, kad paveiktų aplinkos būseną; pastoviai interpretuoja fiksuotas aplinkos būsenas ir daro išvadas (inicijuoja veiksmus reaguodamas į aplinkos pokyčius) (Hayes – Roth, 1995);
- *Agentas* yra autonominė, problemoms spręsti skirta kompiuterinė sistema, sugebanti vienodai efektyviai veikti dinamiškoje ir atviroje aplinkoje (Agent Technology Roadmap, 2004).

Aukščiausiu abstrakcijos lygmeniu agentai skirstomi į (Lupeikienė, 2007):

- agentas-žmogus;
- agentas-aparatūra;
- programinis agentas.

Išskiriamos agentų rūšys:

- protingi (intelektualūs) agentai, sugebantys mokytis ir daryti išvadas;
- autonominiai agentai, sugebantys modifikuoti tikslo pasiekimo būdus;
- išskirstyti agentai, fiziškai esantys skirtinguose kompiuteriuose, jie privalo tarpusavyje komunikuoti ir neveikia pavieniui;
- mobilūs agentai, sugebantys keisti vietą (procesorių) siekdami savo tikslo.

Tam, kad programinė įranga galėtų būti laikoma agentu, ji turi pasižymėti tokiomis savybėmis – agentų savybėmis (Lupeikienė, 2007):

- *Reagavimas* (fundamentali savybė). Agentai funkcionuoja aplinkoje, todėl jie turi tinkamai reaguoti į aplinkos pokyčius ar gaunamą informaciją. Taip pat agentai turi tinkamus jutiklius ir savo (vidinį) aplinkos modelį (histeretiniai agentai);
- *Autonomiškumas* (fundamentali savybė). Agentai funkcionuoja be komandų ir nurodymų iš aplinkos. Agentas turi kontroliuoti išteklius ir turėti įgaliojimus veiksams vykdyti. Taip pat agentai gali turėti skirtingą autonomiškumo laipsnį ir įgaliojimus;
- *Proaktyvumas*. Agentas imasi iniciatyvos ir pasirenka sprendinį. Agentas turi tikslą. Tikslų sistema yra pakankamai sudėtinga;

- *Samprotavimas/mokymasis.* Agentas turi šias dalis: žinių bazę, samprotavimo bloką (mechanizmą) ir gebėjimą prisitaikyti prie pokyčių. Taip pat jis pasižymi tam tikru racionalumu;
- *Mobilumas.* Agentas turi gebėjimą keisti savo buvimo vietą. Taip yra sumažinama tinklo apkrova bei sumažinamos ryšio (sujungimo) išlaidos. Yra numatomos susitikimo vietos, agentūros, teikiamos sutartos paslaugos;
- *Komunikavimas.* Agentai komunikuoja tam, kad užmegztų kontaktą su aplinka. Yra naudojama tam tikra agentui suprantama komunikavimo kalba. Pokalbiai (komunikavimas) vyksta pagal protokolus, kurie apibrėžia leistinas kalbos aktų sekas (kalbos aktais vadinami pranešimai, nurodantys visus veiksmus, kurie turi būti atlikti);
- *Kooperavimasis.* Agentų kooperavimasis leidžia realizuoti situacijas, kurios viršija vieno agento galimybes;
- *Charakteris.* Agentų išorinė elgsena paprastai būna su kiek galima daugiau žmogaus turimų savybių. Taip sutinkamas dorumas, patikimumas, pan. Galimos emocinės būsenos (pyktis, liūdesys, nusiminimas, pan.).

Agentas yra intelektualus, jei:

- gyvybingas – geba saugotis nuo funkcionalumo pažeidimų ir, jiems įvykus, automatiškai tą funkcionalumą atkurti;
- adaptyvus – geba automatiškai perkonfigūruoti savo funkcines galimybes ir keisti savo sąsajas;
- generatyvus – geba generuoti artefaktus;
- aktyviai asistuojantis – numato veiksmus ir duoda patarimus, padeda priimti projektinius ir kitus reikalingus sprendinius;
- aktyvus – gali inicijuoti tinkamus veiksmus;
- geba mokytis – panaudodamas indukcijos ar kitus metodus, sugeba generuoti naujas žinias ir jas kaupti.

Taigi, programiniu agentu, o kartu ir autonominiu agentu, galima vadinti sistemą, kuri egzistuoja aplinkoje arba būdama jos dalis, vykdo paruoštą planą (užduotį) stebėdama aplinką ir adekvačiai (atitinkamai) imdamasi

priemonių (veiksmų) aplinkai keisti realizuoja (įgyvendina) planą (užduotį) (Lupeikienė, 2007). Autonominiai agentai egzistuoja tam tikroje aplinkoje. Pasikeitus aplinkai (patekus į kitą aplinką) agentas virsta (tampa) nebe agentu. Vadinasi egzistuoja reliatyvumas aplinkos atžvilgiu, t. y. vienos aplinkos atžvilgiu sistema bus agentas, kitos ne.

Dažnai terminas *programinis agentas* tapatinamas su kompiuterinės programos sąvoka (Franklin, Graesser, 1996). Paprasčiausia kompiuterinė programa veikia realaus pasaulio aplinkoje, t. y. programai pateikiami duomenys, o algoritmą įvykdžiusi programa grąžina atsakymus, tačiau grąžinti atsakymai neturi jokio poveikio programos veikimui ateityje. Paprasta programa neatitinka ir nuolatinio egzistavimo (veikimo) sąlygos. Programą paleidus, atlikusi užduotį ji patenka į komą, t. y. iš naujo ji gali pradėti veikti tik naujai iškviesta (paleista). Todėl, programinis agentas yra programa, tačiau ne kiekviena programa yra programinis agentas.

Kadangi iki šiol nėra tikslaus programinio agento apibrėžimo, todėl šie agentai gali būti klasifikuojami įvairiais būdais, pavyzdžiui, pagal savybes, kuriomis pasižymi, pagal jų funkcionalumo bruožus, pagal tai kokias užduotis gali atlikti, pagal jų technologines schemas.

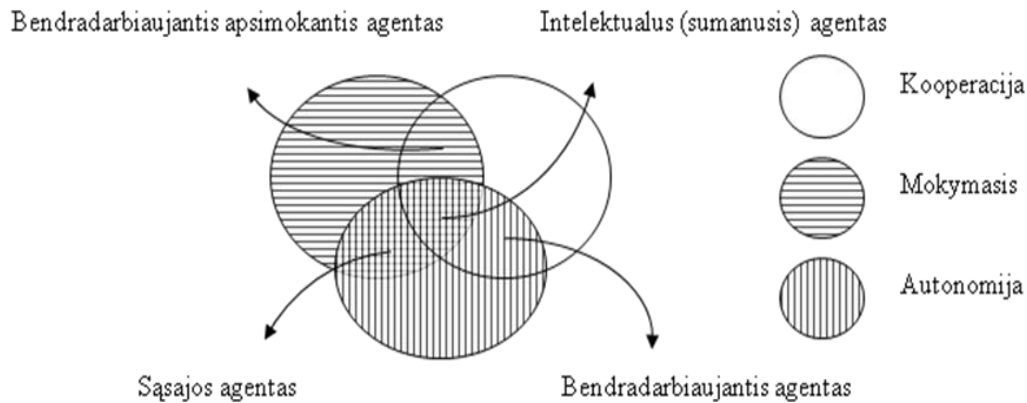
Brustoloni (1991) išskyrė tris agentų klases:

- savireguliuojantys agentai;
- planuojantys agentai;
- adaptyvūs agentai.

Nwana (1996) pasiūlė agentus klasifikuoti remiantis elgesio bruožais (18 pav.), t. y. pagal tai, kaip stipriai agente pasireiškia kooperacijos, apsimokymo ir autonomiškumo savybės. Jo išskiriami pagrindiniai agentų tipai:

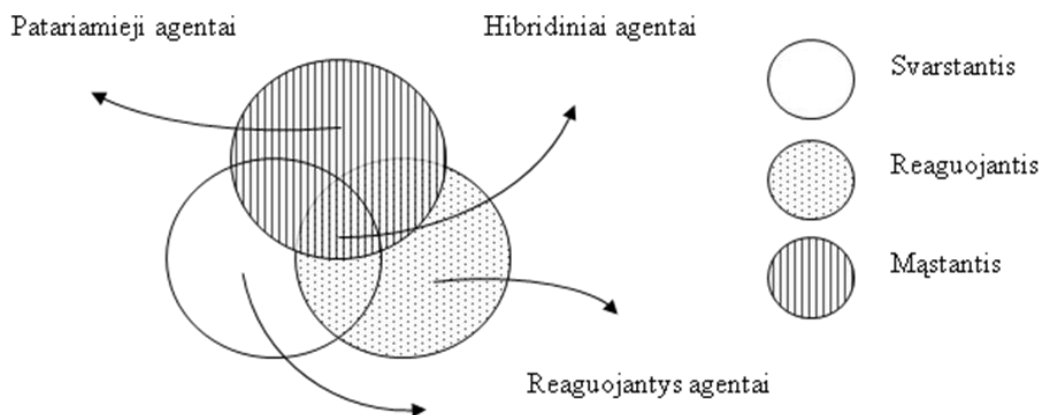
- bendradarbiaujantys agentai;
- sąsajos agentai;
- mobilieji agentai;
- informaciniai/internetiniai agentai;

- reaaguojantys agentai;
- hibridiniai agentai;
- sumanieji agentai.



18 pav. Nwana agentų klasifikacija (pagal Nwana)

Davis (1997) pasiūlyta klasifikacija remiasi trimis intelektinėmis agento savybėmis (gebėjimas svarstyti, gebėjimas reaguoti ir gebėjimas mąstyti) ir kiek stipriai jos išreikštos agente (19 pav.). Šios intelektinės savybės:

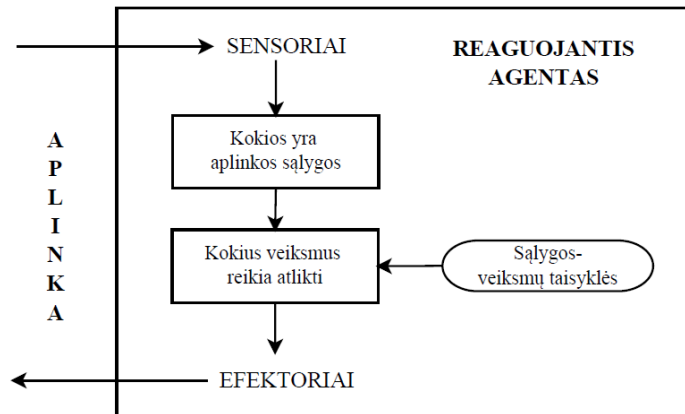


19 pav. Davis agentų klasifikacija (pagal Davis)

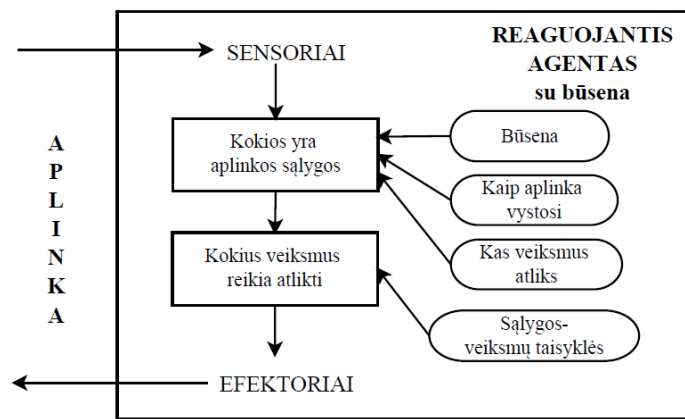
Pagal architektūrą agentai skiriami į:

- reaguojančius agentus;
- reaguojančius agentus su būseną.

Reaguojančio agento architektūra pateikiama 20 paveiksle, o reaguojančio agento su būseną – 21 paveiksle:



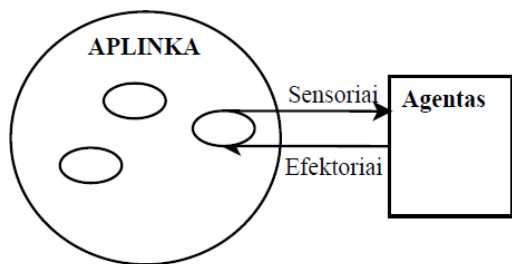
20 pav. Reaguojančio agento architektūra



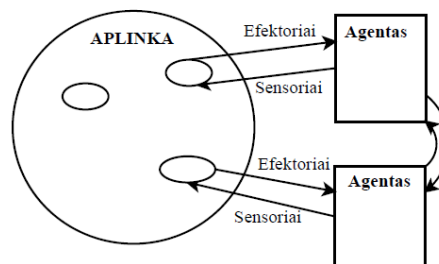
21 pav. Reaguojančio agento su būseną architektūra

Agentai laikosi tikėjimų norų ir ketinimų principo (*BDI*, angl. *Belief, Desire, Intention*). Pagal šį principą, įvykus kokiam nors įvykiui, agentas į jį reaguoja remdamasis savo žiniomis (*tikėjimais*) ir tikslais, kuriuos nori pasiekti šiuo įvykiu (*troškimu*). Tikslai įgyvendinami pagal iš anksto numatytus planus (*ketinimus*).

Vieno agento sistemoje agentas modeliuoja save patį, aplinką ir tarpusavio sąveiką. Agentas kartu yra aplinkos dalis ir tuo pačiu aplinkoje jis dar turi papildomus komponentus. Agentinė sistema veikia centralizuotai. Jei egzistuoja kiti agentai, jie yra traktuojami kaip aplinkos dalis. Agentai modeliuojami kaip turintys savo tikslus, veiksmus ir dalykinės srities žinias. Padedamas sensorių ir efektorių agentas sąveikauja su aplinka (22 pav.) ir kitais agentais (23 pav.).

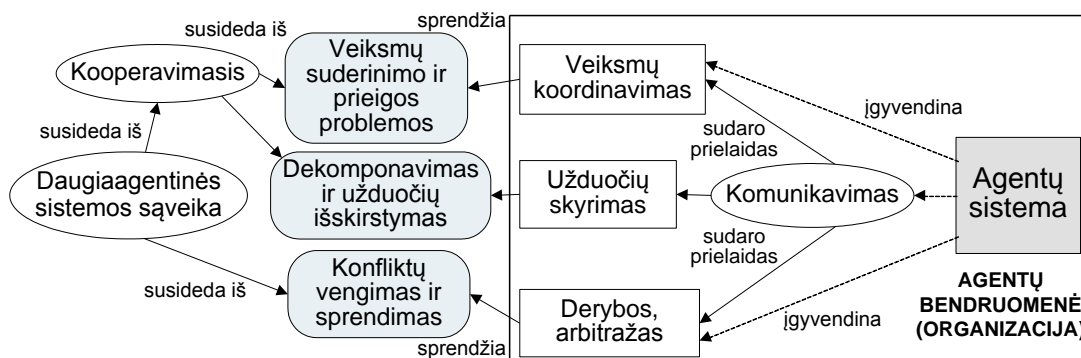


22 pav. **Agento sąveikavimas**



23 pav. **Kelių agentų sąveikavimas**

Paprastai sistemose yra naudojami keli agentai. Jie veikdami bendrauja tarpusavyje taip sudarydami daugiaagentinę sistemą. Agentų bendravimas vyksta pagal 24 paveiksle pateiktą schemą.



24 pav. **Agentų bendravimo schema** (pagal Lupeikienę)

APMS turėtų atlikti skirtingo pobūdžio veiksmus, todėl ji bus modeliuojama kaip daugiaagentinė sistema, kur kiekvienam specifiniam veiksmui atlikti veiks atskiras agentas (žr. 3 skyrių). Sistemoje veiks reaguojantys agentai.

2.5. Mokymosi stiliai

Dažniausiai mokant programavimo yra gilinamasi į metodologinius dalykus (įvairių mokymo metodų taikymą, mokymosi turinio nuoseklumą), tačiau mažai dėmesio kreipiama į psichologines ar fiziologines besimokančiojo savybes. Tuo tarpu, jos yra pakankamai svarbios užsiimant bet kokia veikla. Programavimas bei programavimo mokymas irgi ne išimtis – mokymas turi būti individualizuojamas.

2.5.1. Mokymosi stilių įvairovė

Paskutiniu metu vis dažniau yra kalbama apie mokymosi individualizavimą, personalizavimą ir adaptavimą besimokančiojo asmeniniams poreikiams, jo pažinimo ar mokymosi stiliui. Yra daug įvairių mokymosi stilių klasifikacijų (Hawk, Shah, 2007), kurios, atsižvelgiant į skirtingus kriterijus ir panaudojant skirtingus instrumentus (7 lentelė), leidžia įvairiai klasifikuoti besimokančiuosius bei parinkti geresnius mokymosi rezultatus sąlygojančias mokymosi veiklas bei mokymosi turinį. Dauguma mokymosi stilių yra nustatoma naudojant įvairius savianalizės klausimynus.

7 lentelė. Mokymosi stilių nustatymo instrumentai

Mokymosi stilių nustatymo instrumentas	Besimokančiųjų mokymosi stiliai	Validumas/poveikis
<i>Cognitive Style Index</i> (CSI) (Allison, Hayes, 2012)	Remiamasi intuicija – analize	nenustatytas
<i>Gregoric Style Delineator</i> (GSD) (Herod, 2005)	Abstraktus nuoseklusis, konkretus nuoseklusis, abstraktus atsitiktinis, konkretus atsitiktinis	abejotinas
<i>Herrmann Brain Dominance Instrument</i> (HBDI) (Russel, 2004; Bunderson)	Kairysis – dešinysis smegenų pusrutuliai (pilnas smegenų modelis)	nenustatytas
<i>Learning Styles Inventory</i> (LSI) (Beres ir kt., 2012)	Kolb patirtinis mokymosi modelis	abejotinas
<i>Cognitive Style Analysis</i> (CSA) (Rezaei, Katz, 2004)	Holistinis – analitinis Verbalinis – vizualus	abejotinas
<i>Inventory of Learning Styles</i> (ILS)	Gylis proceso eigos arba produkto prasme	abejotinas
<i>The Myers-Briggs Type Indicator</i> (MBTI) (Beres ir kt., 2012)	16 asmenybės tipų	mažas

Mokantis programuoti svarbus besimokančiojo mąstymas ir jo veikla. Kad besimokantysis būtų aktyvus, jam reikia keturių skirtingų, bet tarpusavyje glaudžiai susijusių gebėjimų:

- aktyviai veikti (patirtis);
- stebėti ir mąstyti (mąstymas);
- formuoti sąvokas ir apibendrinimus (apibendrinimas);
- išbandyti sąvokas ir konceptus naujose situacijose (išbandymas).

Kiekvienas besimokantysis turi skirtingą šių gebėjimų lygį, nes kiekvienu atveju besimokančiajam būdingos tam tikros savybės, jie informaciją įsisavina bei naudoja individualiu būdu. Mokymosi procesą galima

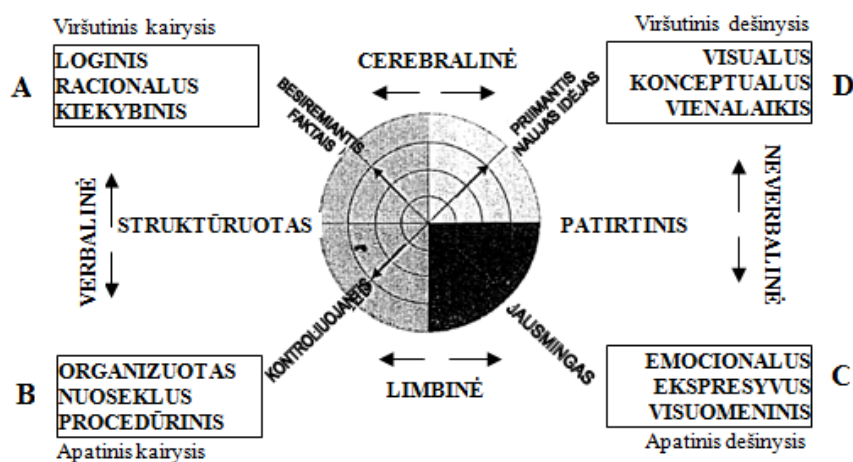
suskirstyti į 4 kategorijas: kontekstas, informacija, informacijos apdorojimas ir atsakas. Atsižvelgiant į kategoriją atitinkamai gali būti naudojama mokymosi stilių klasifikacija:

- Kontekstas – *Dunn ir Dunn* mokymosi stilių klasifikacija;
- Informacija – *Bandlerio ir Grindlerio* mokymosi stilių klasifikacija;
- Informacijos apdorojimas – *Herrmann*‘o mokymosi stilių klasifikacija;
- Atsakas – keturių klausimų sistema.

Nuo besimokančiojo stiliaus priklauso jų mokymosi veikla, naudojami mokymosi metodai, keliami klausimai, bendravimas, apibendrinimas. Čia taip pat labai svarbi protinė veikla, kuri sąlygoja algoritminę bei loginę mąstymą. *Herrmann*‘o mokymosi stilių klasifikacija remiasi žmogaus smegenų pusrutulių darbu. Ši klasifikacija aptariama toliau.

2.5.2. Mokymosi stilių klasifikavimas naudojant Herrmann‘o smegenų dominavimo instrumentą

Nedas Hermanas (*Ned Herrmann*) 1970 metais sukūrė smegenų funkcionavimo mokymo ir mokymosi modelį. Remiantis šiuo modeliu galima nustatyti asmens kiekvieno iš keturių kvadrantų (A, B, C ir D) išreikštumo laipsnį (25 pav.). Tai yra nustatoma naudojant specialų 120 klausimų klausimyną (*HBDI* – angl. *Herrman Brain Dominance Instrument*, 4 priedas) bei analizuojant jo rezultatus.



25 pav. Smegenų mokymo ir mokymosi modelis (pagal Herrmann)

Šis metodas leidžia besimokančiuosius suskirstyti pagal jų mąstymo – informacijos apdorojimo – pobūdį. Pagal smegenų fizinio funkcionavimo atliekant užduotis principus yra skiriamos 4 būsenos (kvadrantai) (8 lentelė). Tačiau gali būti taip, jog pakankamai išreikštos būna du kvadrantus atitinkančios savybės (tai būdinga 58 % žmonių) arba žmogui gali būti būdingas vienodas visų keturių kvadrantų išreikštumo lygis (tokių žmonių yra apie 3 %) (Lumsdaine, Lumsdaine, 1995).

8 lentelė. **Herrmann'o mokymosi stilių aprašymas**

Būsena (kvadrantas)	Smegenų pusrutulis / nervų sistema	Mąstymo tipas	Raktiniai žodžiai	Tinkamos veiklos
A	Kairysis/ cerebralinė	Analitinis	Loginis, analitinis, kiekybinis, faktinis, kritinis	Duomenų rinkimas, informacinių paskaitų klausymas, tekstinės informacijos skaitymas. Sprendžiant idėjas remiamasi faktais, kriterijais ir loginiu mąstymu
B	Kairysis/ limbinė	Nuoseklus	Nuoseklus, organizuotas, planuotas, detalizuotas, struktūruotas	Sekimas pagal nurodytą kryptį, pasikartojančios išsamiai aprašytos problemos, laiko valdymas ir tvarkaraščiai
C	Dešinysis/ limbinė	Tarpasmeninis	Emocionalus, visuomeninis, sensorinis, kinestetinis, simbolinis	Klausymasis ir dalinimasis idėjomis, prasmės ieškojimas, sensorinė įvestis, grupinis tyrimas
D	Dešinysis/ cerebralinė	Vaizdingas	Vizualus, holistinis, inovatyvus	Žiūrėjimas į paveikslėlius, iniciatyvos rodymas, simuliacija (nuolatinis klausinėjimas „kas jeigu...“), vaizdinės priemonės, problemos įvertinimas, smegenų lietus

Žinant besimokančiojo individualias mąstymo ir informacijos apdorojimo galimybes galima ieškoti individualių mokymo veiklų, priemonių bei mokymosi turinio elementų (MO) arba suteikti galimybę pačiam pasirinkti labiausiai jam tinkančius.

2.6. Mokymosi medžiagos elementai ir jų saugyklos

Iki šiol nėra suformuluota visuotinai priimto mokomųjų veiklų ir mokomųjų metodų žodyno, o MO aprašyti yra visuotinai priimtas standartas, todėl APMS mokymosi medžiaga bus siejama su MO.

2.6.1. Mokieji objektai

Mokomasis objektas (MO) – tai bet koks skaitmeninis išteklius, kurį galima naudoti mokymuisi ir taikyti iš naujo kituose mokymosi kontekstuose (Dagienė, Kurilovas, 2008; Wiley, 2000)

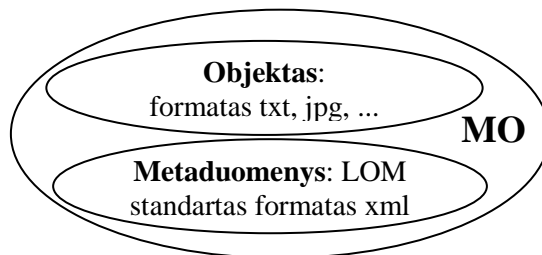
El. mokymo tyrėjai (Wiley, Merrill, Dolphin ir kt., Yang ir Yang) siūlo skirtingas mokomųjų objektų klasifikacijas:

- Wiley (2000) mokomuosius objektus skirsto į penkis tipus: fundamentalus, uždaras sudėtinis, atviras sudėtinis, generuojamasis-pristatymas ir generuojamasis-mokomasis;
- Merrill (2000) mokomuosius objektus klasifikuoja atsižvelgdamas į jų progresyvinę būklę: daugialypiai objektai, informatyvūs objektai, mokieji objektai, kursai ir rinkiniai;
- Dolphin ir Miller (2002) skiria 3 tipus: generuojamasis, jungtinis ir adaptyvus;
- Kiti autoriai tokie kaip Yang ir Yang dalo juos į 3 tipus: pagrindinis objektas, atvaizduojantis mokymo medžiagą; mazgo objektas, atitinkantis mokymo medžiagos skiltį, skyrių ir/ar poskyrį; lapo objektas, atitinkantis einamąjį žinių apibūdinimą (Yang, Yang, 2005);
- Churchill išskiria 6 mokomųjų objektų grupes: informacijos objektas, konceptualus modelis, kontekstuali reprezentacija, praktinis objektas, simuliacinis objektas ir atvaizdavimo objektas (Churchill, 2007).

Kad ir koks būtų objektų skirstymas, MO yra naudojami mokymuisi ir tuo pačiu jie padeda siekti numatytų mokymosi tikslų, kurie aprašomi per dalykines kompetencijas (mokymosi rezultatus).

Kiekvienas MO susideda iš dviejų dalių (26 pav.): MO ir jį aprašančių duomenų – metaduomenų (Kubilinskienė, 2012). Pagal MO technologinį sprendimą, jie gali būti:

- struktūrizuoti, sukurti pagal tam tikrą standartą;
- ne struktūrizuoti.



26 pav. MO struktūra

2.6.2. Metaduomenų standartai ir specifikacijos

Šiame skyrelyje aprašomi ir analizuojami egzistuojantys ir naujai atsirandantys standartai susiję su mokymosi ištekliais, turinio srities paketavimu ir valdymu.

MO yra ne tik pats objektas (paveikslas, tekstas ir pan.), bet ir su juo susiję mokymosi medžiagą aprašantys metaduomenys, kuriuose nurodomas objekto autorius, pavadinimas, paskirtis, kalba, tema, tipas ir kt. Pagal šiuos duomenis objektą galima rasti MO saugyklose (žr. 2.6.3. skyrelį).

MO saugyklose naudojami metaduomenų standartai (*IEEE LOM, Dublin Core, ADL SCORM, ISO Metadata Standard MLR* ir pan.) ir specifikacijos (*IMS Learning Resource Meta-data Specification* ir kt.) leidžia vieningai aprašyti adaptavimo elementus – mokymosi išteklius, besimokančiuosius, veiklos architektūrą ir turinio paketus. Yra keletas standartizavimo ir specifikavimo institucijų (*IEEE, ADL, ISO*), kurios kuria metaduomenų standartus šių elementų aprašymui. Atitinkami standartai:

- padaro skaitmeninį turinį mažiau priklausomą nuo jam pateikti naudojamų paslaugų bei techninės įrangos;
- užtikrina vienodą ir aiškią prieigą prie tinklinių mokymosi išteklių ir paslaugų;

- padeda pratęsti kapitalo įsigijimo ir organizacinių pokyčių gyvavimo laiką;
- palengvina sistemos komponentų ir duomenų išteklių integraciją ir priežiūrą.

ADL tiesiogiai rišasi su standartu *IEEE 1484.12.1-2002 Learning Object Metadata – LOM*, skirtu mokomiesiems objektams aprašyti, ir *SCORM* (angl. *Sharable Content Object Reference Model*) – visuma techninių standartų, kurie leidžia MO rasti internete, juos konvertuoti, paskirstyti, pakartotinai naudoti ir eksportuoti mokymosi turinį standartizuotu būdu. *SCORM* yra skirtas mokymosi įrankių vartotojams, kurie kuria MO sistemas ir naujus elektroninio mokymosi turinio įrankius (Abarius, Liubinas, 2014).

Europoje plačiai naudojamas *LOM* tarptautinis standartas ir jo taikymo modeliai. *LOM* (angl. *Learning Object Metadata*) – tai duomenų modelis, dažniausiai užkodotas *XML*, ir naudojamas aprašyti mokomiesiems objektams ir panašioms skaitmeniniams ištekliams, skirtiems remti mokymąsi. Mokomųjų objektų metaduomenų tikslas suteikti galimybę mokomuosius objektus pakartotinai panaudoti bei palengvinti jų tarpusavio sąveiką, dažniausiai virtualių mokymosi aplinkų (VMA) kontekste. *LOM* duomenų modelio savybių laukai pateikti 2 priedo 1 paveiksle.

SCORM – tai standartų ir specifikacijų rinkinys, skirtas žiniatinklio pagrindu grįstam elektroniniam mokymuisi. Jis apibrėžia ryšius tarp kliento turinio ir priimančiosios sistemos, vadinamosios realiu laiku generuojamos aplinkos, kuri dažniausiai yra palaikoma VMA. *SCORM* taip pat apibrėžia, kaip turinys gali būti pakuojamas į perkeliamą ZIP failą, vadinamąjį paketų apsikeitimo formatą (angl. *Package Interchange Format*). *SCORM* apibrėžia, kad turinys turi būti⁴:

- pakuojamas į ZIP failą;
- aprašytas XML faile;
- komunikuojantis naudojant *JavaScript*;
- nuoseklus naudojant XML taisykles.

⁴ Prieiga per internetą: <http://scorm.com/scorm-explained/technical-scorm/>

Naujausia *SCORM* versija *Tin Can API* versija 1.0 sukurta 2013 balandį ir pervadinta *Experience API (xAPI)* ir *Next Generation SCORM*. Ji leidžia spręsti daugelį problemų, kurios buvo sietinos su ankstesnėmis *SCORM* versijomis. Mobilus mokymasis, komandinis mokymasis, kryžminių sričių funkcionalumas, nuoseklumo nustatymas, simuliacija/ rimti žaidimai dabar lengvai įgyvendinami. *Tin Can API* pašalina turinį iš virtualios mokymosi aplinkos ir leidžia turiniui išsiųsti pranešimus, kurie atrodo taip – [aktorius] [veiksmažodis] [objektas] arba „aš padariau tai“, į mokymosi įrašų saugyklą. Mokymosi įrašų saugyklos gali būti savarankiškos arba gali būti virtualios mokymosi aplinkos dalis.

IMS Learning design yra formalus pedagoginis standartas, skirtas aprašyti technologijų palaikomus mokymosi scenarijus, kurie remiasi mokymo konstravimo modeliais. Tai formali kalba, leidžianti modeliuoti mokymosi procesą.

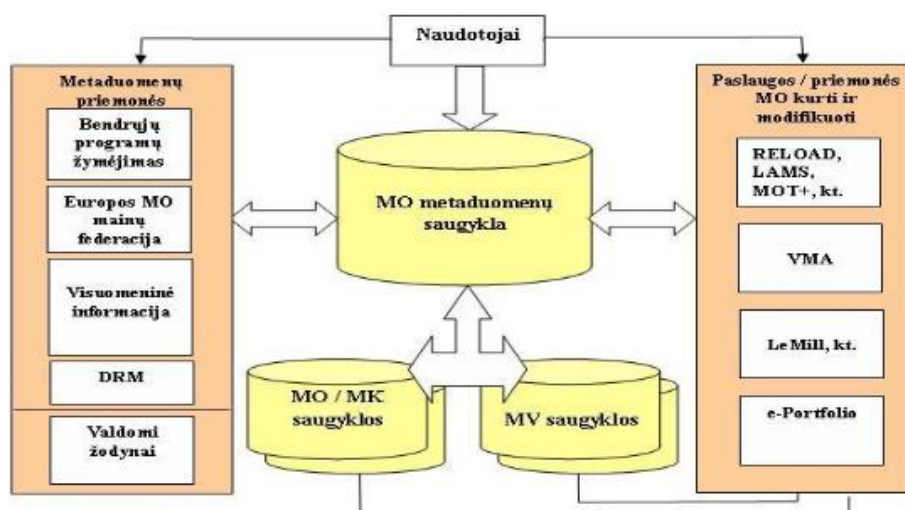
2.6.3. Mokomųjų objektų metaduomenų saugyklos

MO metaduomenų saugykla – tikslingai parengta sistema, kurią sudaro MO metaduomenys ir jų valdymo priemonės (Kubilinskienė, 2012). Saugykloje galima aprašyti skirtingų tipų ir skirtingų technologinių sprendimu parengtus MO. Aprašai sistemoje yra kuriami struktūrizuoti, naudojant MO metaduomenų standartą.

Pagrindinė saugyklų paskirtis (Kubilinskienė, 2012):

- Surasti MO metaduomenų saugykloje, naudojant įvairius paieškos būdus;
- Standartizuotai kurti MO aprašus, kurie leistų aprėpti kiekvieno MO savybes: edukacines, technines, taip pat skaitmeninio naudojimo teises;
- Atlikti MO aprašų tvarkymo funkcijas: duomenų tvarkymą, techninį ir funkcinį suderinamumą;
- Atlikti MO aprašų mainus;
- Kaupti vertinimus ir komentarus, dalintis savo patirtimi apie parsisiųstą MO.

Laikui bėgant saugyklų internete vis daugėja. ADL (angl. *Advanced Distributed Learning*) iniciatyva, kuri atsirado JAV gynybos departamento inicijuoto projekto metu, skatinusio esamas mokymosi ir informacines technologijas panaudoti mokymui ir kvalifikacijos tobulinimui, ir kuriai jau daugiau kaip 15 metų, leido įvairiose šalyse (ypatingai JAV) sukurti dideles saugyklas (laisvos prieigos arba mokamas). Laisvos prieigos saugyklų registre (angl. *Registry of Open Access Repositories*, prieiga internetu <http://roar.eprints.org/>) galima rasti 51 MO saugyklą. Europos mokymosi išteklių mainų portale (<http://lreforschools.eun.org/>) esančioje MO saugykloje galima rasti mokymosi išteklių aprašų iš skirtingų šalių. Lietuvoje taip pat yra sukurta MO saugykla, kurioje naudojamas Europos mokyklų tinklo parengtas LRE LOM 3.0 taikymo modelis. Saugyklos architektūros schema pateikta 27 paveiksle.



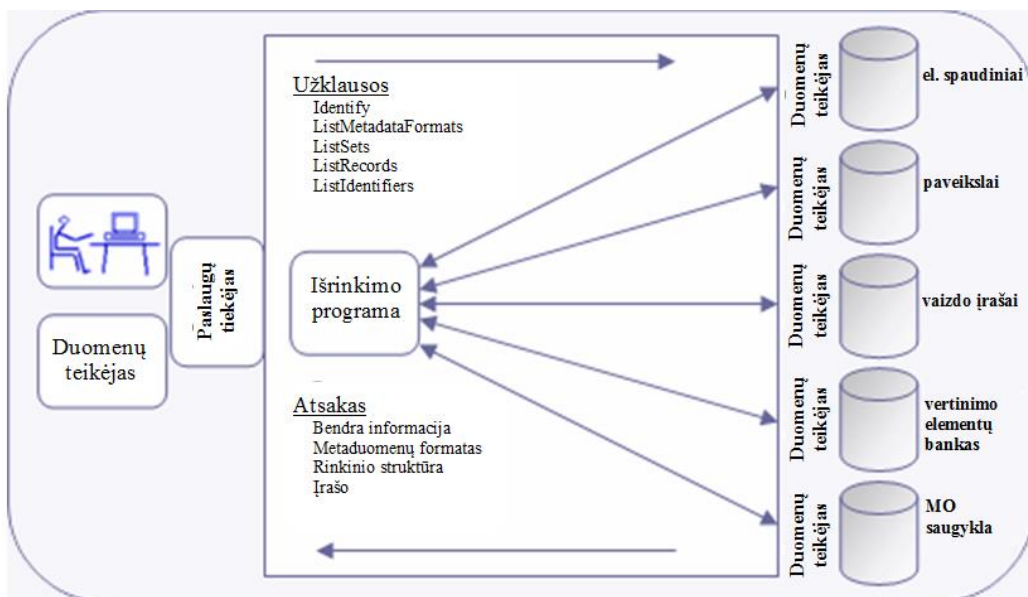
27 pav. Lietuvos MO metaduomenų saugyklos schema (Kubilinskienė, 2012)

2.6.4. Harvesting modelis reikiamų duomenų paieškai

Norint išrinkti ir naudoti reikiamus MO, saugomus įvairiose saugyklose, yra naudojami MO išrinkimo metodai. Vienas tokių yra *Harvesting* metodas. Duomenų išrinkimui naudojama nepriklausoma metaduomenimis paremta klientinė duomenų išrinkimo programa (angl. *Harvester*), kuri vykdo OAI-PMH (ang. *Open Archives Initiative Protocol for Metadata Harvesting*) užklausas. Programoje naudojamos dvi klasės: duomenų teikėjai,

administruojantys sistemas, kurios metaduomenims pateikti palaiko OAI-PMH, ir paslaugų tiekėjai, kurie formuojant reikšmėmis apibrėžiamas paslaugas naudoja išrinktus OAI-PMH protokolą palaikančius metaduomenis. Duomenų išrinkimo programa valdo paslaugų tiekėją nurodydama kaip išrinkti metaduomenis iš saugyklų (The Open Initiative..., 2008). OAI-PMH palaiko šešių tipų užklausas (apibrėžiamas veiksmazodžiais, 28 pav.), kurios perduodamos naudojant HTTP GET ir POST metodus:

- `Identify` – archyvo aprašymas;
- `ListMetadataFormats` – duomenų teikėjo palaikomų metaduomenų formatų sąrašas;
- `ListSets` – numatomų rinkinių sąrašas;
- `ListRecords` – išrenkami įrašai iš saugyklų;
- `ListIdentifiers` – išteklių identifikatorių sąrašas, sutrumpinta `ListRecord` įrašo forma;
- `GetRecord` – individualus įrašas.



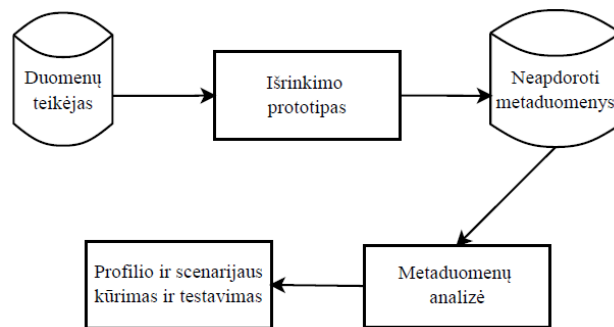
28 pav. **Metaduomenų išrinkimo modelis** (Fegen, 2007)

Norint naudoti šį metodą, paslaugų tiekėjai beveik visais atvejais turi mažiausiai normalizuoti išrenkamus metaduomenis. Tačiau dažnai tenka atlikti daugiau sudėtingų funkcijų, pvz., pridėti duomenų šaltinio laukus, didinti

metaduomenų granuliaciją arba klasifikuoti laukus nurodant, kad taikant tam tikrą sąlygą išrenkami duomenys įtraukiami į tam tikrą specialų žodyną.

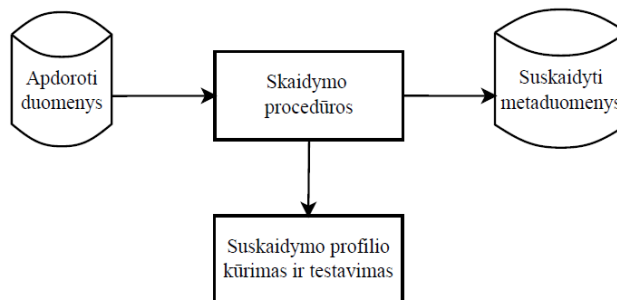
Kai kurios transformacijos gali būti atliktos be dialogo su duomenų teikėju (pvz., duomenų normalizavimas), tačiau daugeliu atvejų paslaugų tiekėjai gali produktyviai komunikuoti su duomenų teikėju. Jei didelio metaduomenų rinkinio negali paveikti duomenų teikėjas, tada bent paslaugų tiekėjas galėtų atrasti metaduomenų transformacijos procesui naudingą informaciją. Todėl kuriant metaduomenų saugyklą išrenkamiems duomenims siūloma naudoti iteracinį procesą (Tennant, 2004)).

Procesas pradedamas naudojant pagrindinį išrinkimo prototipą, kuris analizuoja metaduomenis, išrinkimo profilio specifikaciją ir reikiamas atlikti metaduomenų transformacijas (29 pav.). Išrenkamiems duomenims vizualizuoti gali būti naudojamos įvairios programinės priemonės, pvz., *Microsoft Excel*, leidžiančios nustatyti išrenkamų duomenų savybes ir anomalijas (Dushay, Hillman, 2003).



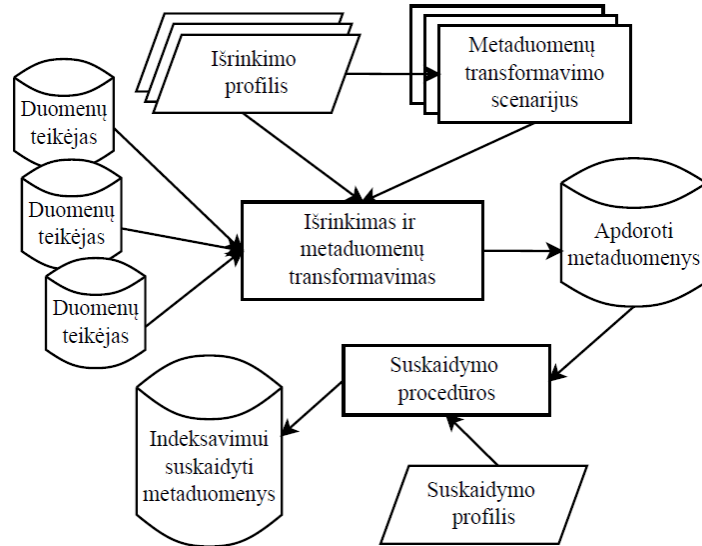
29 pav. Metaduomenų analizė, paieškos profilio kūrimas ir metaduomenų transformacijos procesas (Tennant, 2004)

Kitame žingsnyje vyksta tinkamų įrašų išrinkimui reikalingų veiksmų-operacijų skaidymas ir nurodymas paieškos tarnybai (30 pav.).



30 pav. Pagalbinio profilio ir procedūrų kūrimas (Tennant, 2004)

Paskutiniame žingsnyje nustatomos išrinkimo paslaugos, kurios kuriant atitinkamą išrinktų ir transformuotų duomenų rinkinį naudoja sukurtą profilį ir apibrėžtus skaidymo bei transformacijos procesus (31 pav.).



31 pav. Paieškos rezultatų radimo procesas (Tennant, 2004)

Tinkamai realizuoti išrinkimo proceso kūrimo aspektai gali būti naudojami atliekant metaduomenų transformacijas kitose metaduomenis naudojančiose veiklose.

2.6.5. Mokomojo dalyko kūrimo automatizavimas

Išrinktus MO saugyklose saugomus MO tenka pritaikyti savo kuriamam mokomojo dalyko kursui. Kaip minėta 2.1.6. skyrelyje, APMS mokymosi kurso elementai (MO) bus siejami su dalyko temomis, siektinomis dalykinėmis kompetencijomis ir *Bloom*‘o taksonomijos lygmenimis. Šį susiejimą gali atlikti žmogus – mokymosi kurso kūrėjas. Tačiau kurso kūrimo procesas gali būti automatizuojamas. Vieną tokių automatizavimo metodų siūlo Kurilovas (2008). Jo siūlomame metode vietoje didelės dalykinės kompetencijos sąryšiui yra naudojamos mažesnių dalykinių kompetencijų, nusakomų LOM dalyku ar tema ir subkompetencijomis (pažodžiui „veiksmažodžio išreiškimu“, angl. *Action verb expression Taxonomy*), taksonomijos (Kurilovas, 2008; 2012). MO yra siejami smulkiomis dalykinėmis kompetencijomis. Kadangi APMS kursui bus naudojamos iš anksto parengtos smulkesnės dalykinės kompetencijos,

tačiau šias kompetencijas būtina susieti su *Bloom*‘o taksonomijos lygmenimis, todėl Kurilovo pasiūlytasis programos žymėjimo metodas sistemos projektavimo etape praplečiamas (žr. 3.4. skyrelį).

2.7. Mokymo sistemos kūrimo modeliai

Mokymo konstravimas (ID – angl. *Instructional Design*) – tai sisteminis metodas kurti atitinkamas instrukcijas. Pagrindinis jo tikslas yra patobulinti instrukcijas siekiant pagerinti patį mokymąsi. Kadangi projektuojama sistema yra skirta mokymuisi, todėl ji yra grindžiama mokymo sistemos kūrimo modeliais (ISD – angl. *Instructional Systems Design Model*).

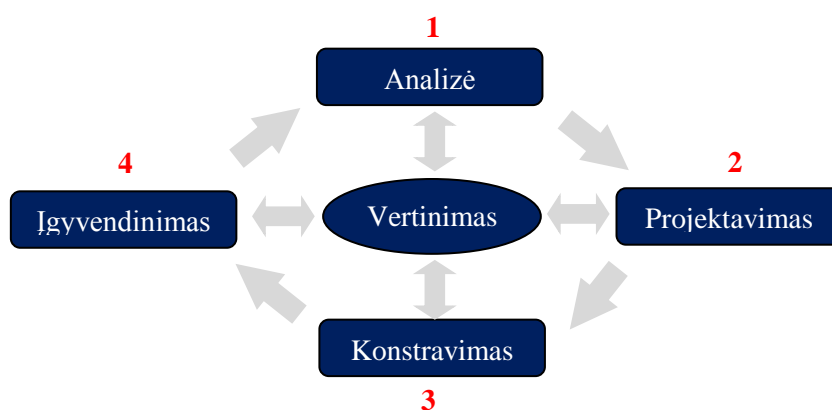
Gagne, Briggs ir Wager išskiria 14 mokymo sistemos kūrimo etapų, kurie apima skirtingus komponentų lygmenis (Seels, Glasgow, 1998, 9 lentelė). Čia yra remiamasi Gagne‘s mokymo teorijos devyniais mokymo žingsniais: 1) Patraukti dėmesį; 2) Informuoti besimokančiuosius apie tikslus; 3) Ankstesnės mokymosi patirties sužadėjimas; 4) Pristatyti paskatinimus. 5) Pateikti mokymosi gaires; 6) Suaktyvinti veiklą; 7) Pateikti grįžtamąjį ryšį; 8) Įvertinti veiklą; 9) Padidinti išlaikymą atmintyje ir perdavimą.

9 lentelė. **Mokymo sistemos konstravimo etapai** (pagal Gagne, Briggs ir Wager)

Komponento lygmuo	Atliekama veikla
Sistemos lygmuo	1. Poreikių, tikslų ir prioritetų analizė. 2. Resursų, apribojimų ir alternatyvių pateikimo sistemų analizė. 3. Kurso ir kurso turinio apimties bei nuoseklumo apibrėžimas; Pateikimo sistemos projektavimas.
Kurso lygmuo	4. Kurso struktūros ir nuoseklumo apibrėžimas. 5. Kurso tikslų analizė.
Paskaitos lygmuo	6. Veiklos tikslų apibrėžimas. 7. Paskaitos (ar modulio) plano parengimas. 8. Medžiagos ir terpės rengimas, parinkimas 9. Besimokančiųjų veiklos įvertinimas (veiklos pamatavimas).
Sistemos lygmuo	10. Mokytojo parengimas. 11. Formuojantis vertinimas. 12. Testavimas ir pataisymai. 13. Apibendrinamasis vertinimas. 14. Įdiegimas ir sklaida.

Vienas iš seniausių sisteminių mokymosi kūrimo modelių yra ADDIE modelis, 1975 metais sukurtas Floridos valstijos universiteto Edukacinių technologijų centre (angl. *the Center for Educational Technology at Florida*

State University). Šis modelis apibrėžia pagrindinius mokymosi sistemos konstravimo procesus ir skirtas įvertinti konkrečios veiklos charakteristikas, kurioms besimokantysis turėtų įgyti reikiamus įgūdžius ir kurios yra būtinos norint atlikti tam tikrą darbą ar užduotį. Efektyvios mokymosi ir veiklos palaikymo sistemos kūrimas remiantis ADDIE modeliu susideda iš penkių cikliškai pasikartojančių fazių – analizės, projektavimo, konstravimo, įgyvendinimo bei vertinimo (32 pav. ir 33 pav.). Kiti dažniausiai naudojami mokymosi sistemos kūrimo modeliai yra ADDIE modelio variacijos.



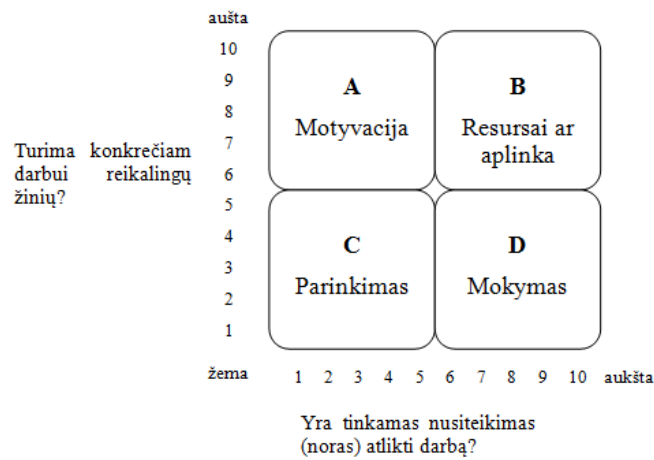
32 pav. **ADDIE modelio schema** (pagal Davis, 2013)

Analizė	•reikalavimų, poreikių, užduočių ir auditorijos analizė
Projektavimas	•mokymosi tikslų, struktūros, pateikimo terpės nustatymas
Konstravimas	•mokymosi medžiagos rengimas
Įgyvendinimas	•parengtos medžiagos pateikimas tikslinei auditorijai
Vertinimas	•įvertinamas mokymosi efektyvumas, atsižvelgiant į besimokančiųjų grįžtamąjį ryšį

33 pav. **ADDIE modelio fazių aprašymas** (pagal Davis, 2013)

ADDIE modelis naudojamas kaip mokymosi kurso kūrimo įrankis, tačiau juo neįvertinama ar mokymosi procesas buvo tinkamas. Todėl norint įvertinti veiklos savybes būtina naudoti veiklos analizės įrankius, pvz. veiklos analizės kvadrantą (*PAQ* – angl. *Performance Analysis Quadrant*, 34 pav. ir 10 lentelė). Juo yra nustatoma „ar turima konkrečiam darbui reikalingų žinių“ ir „ar yra tinkamas nusiteikimas (noras) atlikti tą darbą“. Priskiriant

skaitmenines reikšmes nuo 1 iki 10 gali būti nustatoma, kuriam kvadrantui priklauso besimokantysis.



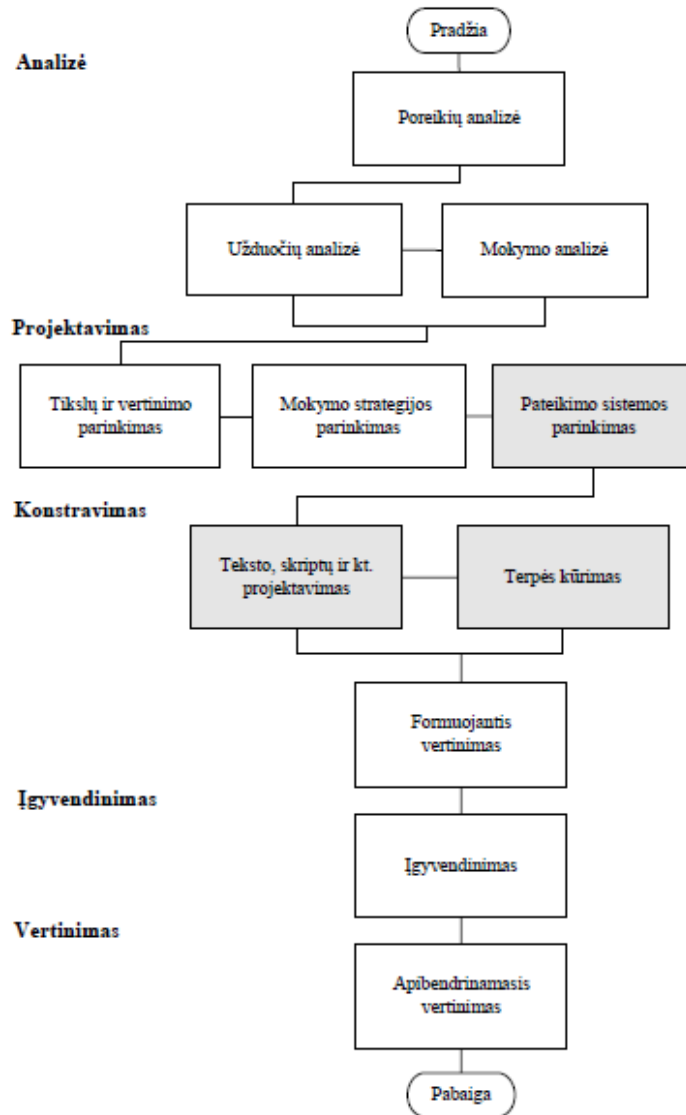
34 pav. Veiklos analizės kvadrantas (pagal Clark, 2013)

10 lentelė. Veiklos analizės kvadrantų aprašymas

Kvadrantas	Apibūdinimas ir veiksmai
A kvadrantas (<i>Motyvacija</i>)	Jei besimokantysis turi pakankamas darbui atlikti žinias, bet turi nepakankamą požiūrį, tai gali būti vertinama kaip motyvacijos problema. Tokiu atveju besimokančiojo veiksmų seka turi būti patikslinta.
B kvadrantas (<i>Resursai/Procesas/Aplinka</i>)	Jei besimokantysis turi tiek darbui reikiamas žinias, tiek tinkamą požiūrį, bet veikla yra nepatenkinama, tuomet gali būti besimokančiojo kontrolės problema, pvz., trūksta resursų ar laiko, užduočiai atlikti reikia patobulinti veiklos procesą ir pan.
C kvadrantas (<i>Parinkimas</i>)	Jeigu besimokančiajam trūksta tiek žinių, tiek tinkamo požiūrio, tai gali reikšti, jog asmuo buvo nukreiptas netinkama linkme. Tokiu atveju turėtų būti tinkamai parenkama veikla ir teikiama tinkama parama.
D kvadrantas (<i>Mokymas ir/ar Treniravimas</i>)	Jei besimokantysis nori veikti, tačiau neturi tinkamų veiklai žinių ar įgūdžių, tokiu atveju būtina tam tikro tipo mokymo strategija – besimokančiojo mokymas ar treniravimas.

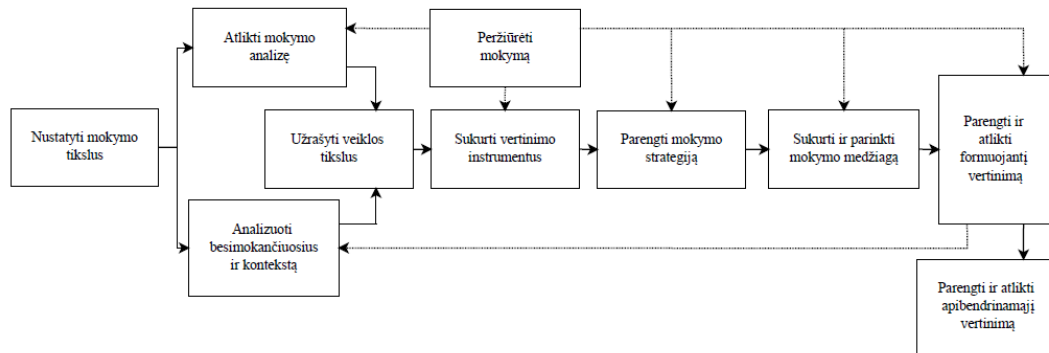
Dažnai ADDIE modelis dar yra vadinamas proceso modeliu, nes jis tik tuomet yra tinkamas, jei aklai yra sekama jo žingsniais.

Seels and Glasgow (1998) išplėtė ADDIE modelio žingsnius orientuodamiesi į pradedančiuosius mokymosi kūrėjus (35 pav.). Naudojantis šiuo modeliu mokymas projektuojamas skaidant procesus į konkretesnius veiksmus. Modelis leidžia konstruoti į produktą orientuotą mokymą ir paprastai naudojamas rengiant mokymosi paketą.



35 pav. Seels and Glasgow modelio schema (pagal Seels, Glasgow, 1998)

Dick ir Carey mokymo konstravimo modelis (sukurtas 1978 m. ir aprašytas knygoje *The systematic design of instruction*) taip pat remiasi procesų skaldymo į mažesnius komponentus metodu. Pagal šį modelį parengtos mokymo instrukcijos sukuria tinkamas sąlygas siekiant numatytų mokymosi rezultatų – reikiamų žinių ir įgūdžių. Dick ir Carey modelį sudaro 10 etapų (36 pav. ir 11 lentelė).



36 pav. **Dick ir Carey modelio schema** (pagal Dick, Carey, 1990)

11 lentelė. **Dick ir Carey modelio etapų aprašymas**

Dick ir Carey modelio etapas	Atliekamos veiklos
Nustatyti mokymo tikslus	Nustatyti ką besimokantieji turi gebėti atlikti pabaigę mokymo kursą
Atlikti mokymo analizę	Nustatyti ką besimokantysis žingsnis po žingsnio turi daryti siekiant tikslo; apibrėžti kokių įgūdžių bei žinių jam reikės
Analizuoti besimokančiuosius ir kontekstą	Nustatyti besimokančiųjų ankstesnius įgūdžius, nuostatas bei požiūrį, mokymosi aplinkos charakteristikas; naudinga informacija yra apie ankstesnę veiklą, žinias, susijusias su nagrinėjama tema, požiūrį į turinį ir numatytą pateikimo sistemą, akademinę motyvaciją, Kokių įgūdžių reikės besimokantiesiems atliekant mokymosi užduotis?
Užrašyti veiklos tikslus	Nustatyti ką besimokantieji galės daryti turėdami tam tikrus įgūdžius, sąlygas ir kriterijus
Sukurti vertinimo instrumentus	Apibrėžti kriterijais paremtą vertinimo instrumentą atitinkantį veiklos tikslus
Parengti mokymo strategiją	Parengti strategiją veiklos prieš mokymąsi (motyvacija, tikslai ir elgesys), informacijos pateikimo (mokymo seka, informacija, pavyzdžiai), besimokančiųjų dalyvavimo (praktinė veikla ir grįžtamasis ryšys), testavimo (tikrinimą prieš ir tikrinimą po), veiklos po mokymosi (pasitaisymas, rezultatų pagerinimas, įsiminimas ir perkėlimas)
Sukurti ir parinkti mokymo medžiagą	Naudoti mokymo strategijas rengiant mokymą; numatyti kokio tipo mokomąją medžiagą naudosime (spausdintą, pateiktą tam tikroje terpėje, abiejų rūšių)
Parengti ir atlikti formuojantį vertinimą	Kaupti duomenis, kurie gali būti naudojami gerinant mokymą – ką turime peržiūrėti siekiant pagerinti mokomąją medžiagą, kaip mokymą padaryti kuo efektyvesniu didesniai skaičiui besimokančiųjų
Peržiūrėti mokymą	Naudoti formuojančio vertinimo metu gautus duomenis siekiant patikrinti mokymo analizės, besimokančiųjų ir konteksto analizės, veiklos tikslų, vertinimo instrumentų, mokymo strategijos ir mokymo pagrįstumą
Naudoti apibendrinamąjį vertinimą	Įvertinama: Ar sistema kaip visuma buvo efektyvi? Ar mokymas vyko? Ar pasiekėme norimus rezultatus?

Mokantis programuoti svarbiausias veiksnys yra besimokančiojo veikla, žinome, kad ADDIE modelio etapai leidžia apibrėžti veiklos charakteristikas ir pagal tai konstruoti mokymosi sistemą, todėl šio modelio elementai – analizės,

projektavimo, konstravimo ir įgyvendinimo etapai – bus naudojami projektuojamoje sistemoje.

2.8. Analizės išvados

1. Analizuojant programavimo mokymosi ypatumus pastebėta, kad programavimo mokymasis yra sudėtingas intelektinis procesas, kurio metu būtinas programavimo srities konceptų supratimas, jų panaudojimo išmanymas, įvairių pavyzdžių studijavimas, taip pat svarbus besimokančiojo aktyvus dalyvavimas sprendžiant panašias užduotis. Atsižvelgiant į šiuos bruožus turėtų būti naudojamos mokymosi veiklos ir prie jų deriniami atitinkamų tipų MO. Taip pat svarbus mokymosi individualizavimas, kadangi besimokančiajam yra būdingas savitas informacijos apdorojimo stilius, į kurį atsižvelgiant būtų padidinamas mokymosi efektyvumas. Kaip parodė žvalgomas tyrimas, mokymosi individualizavimo poreikį akcentuoja ir patys besimokantieji.
2. Literatūros šaltinių ir mokslinių straipsnių analizė parodė, kad nėra adaptyviųjų programavimo mokymosi sistemų, kurios automatiškai kurtų mokymosi kursą ir jį adaptuotų prie besimokančiojo individualių savybių atsižvelgiant į jo mokymosi stilių. Kadangi mokant programavimo ir rengiant programavimo mokymosi kursą bei priemones susiduriama su daugybe problemų – reikia parinkti programavimo kalbą ir paradigmą, programavimo aplinką, apibrėžti ugdomas dalykines kompetencijas ir nustatyti jas pasiekti leidžiančias mokymosi temas ir mokymosi veiklas, numatyti ir parengti mokymosi medžiagą, mokomąsias bei vertinamąsias (leidžiančias vertinti dalykines kompetencijas) užduotis. Kiekvienas skirtingas parinkimas iš naujo reikalauja didelių kurso kūrėjo pastangų. Todėl tikslinga sukurti sistemą, kuri būtų universali ir automatizuotų didžiąją dalį kurso kūrėjo rutininių darbų, atsižvelgdama į pateikiamą mokomojo dalyko apraše pateikiamą informaciją. Kaip mokomosios medžiagos elementus naudojant MO ir atliekant automatinę programos žymėjimą gali būti kuriamas mokomojo dalyko apraše (sudaromame

mokančiojo) nurodytą turinį atitinkantis mokymosi kursas. Veiksmų ir reakcijų į aplinkos faktorius automatizavimui sistemoje tikslinga naudoti programinius agentus.

3. Adaptyviosios programavimo mokymo sistemos modeliui kurti taikytinas ADDIE mokymo kūrimo modelis, kurio etapai detalizuojami taip: analizė – besimokančiųjų savybių analizė ir mokomųjų objektų paruošimas, projektavimas – programavimo mokomojo dalyko aprašo pritaikymas, konstravimas – programavimo mokymosi medžiagos pritaikymas, t. y. mokomųjų objektų žymėjimas pagal mokomojo dalyko programos aprašą, įgyvendinimas – mokomojo dalyko kurso adaptyvus pateikimas besimokančiajam.
4. Sistemai modeliuoti pasirinktas konceptualus modelis – vaizdinis metodas, leidžiantis diagrama pavaizduoti priežastinius ryšius tarp faktorių kurie, manoma, yra reikšmingi nagrinėjamai problemai ar objektui. Geras modelis turėtų aiškiai susieti objektą su jį tiesiogiai veikiančiais faktoriais (tiesioginėmis grėsmėmis ar galimybėmis), netiesioginiais faktoriais ir strateginėmis veiklomis, kurios įtakoja šiuos faktorius. Konceptualus modelis yra lydimas tekstinio aprašymo, kuris sudarytą diagramą paaiškina žodžiais. Sudarant konceptualų modelį aprašomi visi į modelį įtraukiami elementai ir šių elementų elgesį apibrėžiančios taisyklės.
5. Siekiant sistemos elementų daugkartinio panaudojamumo, mokomojo dalyko medžiagą galima sudaryti iš MO saugyklose saugomų MO. Mokymosi kokybė turi būti užtikrinama įvertinant MO kokybę prieš tai, kai jie pradedami naudoti mokymuisi.
6. Analizės metu nagrinėti šaltiniai parodė, kad nebuvo išskirti ir aprašyti programavimo mokymuisi tinkami mokomųjų objektų tipai ir jie susieti su mokymosi stiliais pagal *Herrmann*'o mokymosi stilių klasifikaciją. Moksliniuose darbuose pateikiama informacija bei analizė leidžia išskirti pagrindines programavimo mokymosi veiklas: informacijos šaltinių (sąvokų, terminų, konceptų) studijavimas; pavyzdžių studijavimas; grafinės informacijos studijavimas; vaizdo medžiagos studijavimas; modeliavimas

bei imitavimas; užduočių sprendimas; problemų sprendimas; projektų rengimas. Šios veiklos įtakoja mokymosi kurse naudojamų MO tipų parinkimą.

6. Analizės metu nepavyko aptikti, kad būtų minimas mokomojo dalyko programos (temų ir dalykinių kompetencijų) tikslus susiejimas su MO ir *Bloom*'o taksonomijos lygmenimis. Kadangi projektuojamoje adaptyviojoje programavimo mokymosi sistemoje numatomas dalykinių kompetencijų vertinimas atsižvelgiant į *Bloom*'o taksonomijos lygmenis, anksčiau minėtinas susiejimas turėtų būti įgyvendinamas.
7. Mokslinių darbų analizė parodė, kad niekur nebuvo minimas mokomojo dalyko elementų (MO) adaptavimas atsižvelgiant į besimokančiojo mokymosi stilių pagal *Herrmann*'o klasifikaciją. Kadangi projektuojamąją mokymosi sistema siekiama padidinti mokymosi efektyvumą, kurį nulemia mokymuisi naudojamos informacijos priėmimas ir apdorojimas individualiomis žmogaus galimybėmis bei greitas reikiamos informacijos radimas, adaptyviojoje programavimo mokymosi sistemoje būtina atsižvelgti į individualias besimokančiojo savybes ir pagal tai mokymosi metu adaptuoti mokymosi medžiagą.

3. ADAPTYVIOSIOS PROGRAMAVIMO MOKYMO SI SISTEMOS PROJEKTAVIMAS

Šiame skyriuje išsamiai aprašoma adaptyviosios programavimo mokymosi sistemos (APMS) modelio projektavimo eiga. APMS modeliui kurti naudojamas ADDIE mokymosi kūrimo metodas (žr. 2.6. skyrelį), kurio etapai apima:

- analizė – besimokančiųjų savybių analizės ir mokymosi objektų paruošimas (3.2. skyrelis ir 3.3. skyrelis),
- projektavimas – programavimo mokomojo dalyko aprašo pritaikymas (3.4. skyrelis),
- konstravimas – programavimo mokymosi medžiagos pritaikymas, t. y. mokomųjų objektų žymėjimas pagal mokomojo dalyko programos aprašą (3.4. skyrelis),
- įgyvendinimas – mokomojo dalyko kurso adaptyvus pateikimas besimokančiajam (3.5. skyrelis ir 3.6. skyrelis);
- vertinimas – APMS konceptualus modelio vartojimo kokybės vertinimas nagrinėjams 4 skyriuje (4.2. skyrelis).

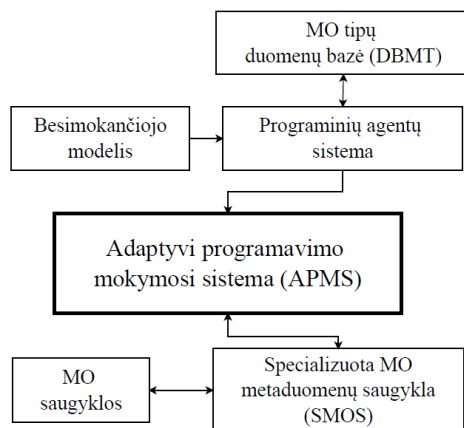
3.1. Adaptyviosios programavimo mokymo sistemos kūrimo prielaidos

Mokslo darbų analizė parodė, jog nėra tokių sistemų, kurios leistų automatizuotai kurti ir adaptuoti programavimo mokymosi kursą (PMK, t. y. mokomojo dalyko programą ir jos elementus (MO)) prie besimokančiojo individualių savybių pagal jo mokymosi stilių. Remiantis 2 skyriaus išvadomis galima teigti, jog tikslinga sukurti adaptyviąją sistemą grįstą MO susiejimu su *Herrmann*‘o mokymosi stiliais ir MO susiejimu su mokomojo dalyko programa (temomis, dalykinėmis kompetencijomis) ir *Bloom*‘o taksonomijos lygmenimis. Sistemos pagrindu gali būti pasirinkta specializuota programavimo MO metaduomenų ir žymių (vartotojų sukurtų metaduomenų) saugykla (SMOS), praturtinta intelektualiomis technologijomis – daugiaagentine sistema. Šios daugiaagentinės sistemos pagrindą turi sudaryti

programiniai agentai, įgyvendinantys anksčiau minėtasias sistemos savybes – *Herrmann*'o mokymosi stilių susiejimo su programavimo mokymuisi tinkamais MO tipais bei mokomojo dalyko programos susiejimo su programavimo MO ir *Bloom*'o taksonomijos lygmenimis, leidžiančiais apibrėžti dalykinių kompetencijų pasiekimo lygį. Projektuojamoje sistemoje naudojamas susitiprinto adaptavimo būdas (žr. 2.4. skyrelį), kai adaptavimas vykdomas remiantis ekspertinio vertinimo reitinguojant MO tipus rezultatais. Sistemos adaptyvumui ir kokybei užtikrinti taip pat naudojami papildomi agentai:

- programavimo dalykinės srities kokybiškų MO metaduomenų išrinkimo iš nurodytų MO saugyklų į SMOS,
- MO tipų reitingavimo programinis agentas,
- adaptyvaus PMK teikimo;
- PMK dalykinių kompetencijų vertinimo.

Taigi APMS sudaro SMOS kartu su adaptavimo funkcijas padedančiais realizuoti programiniais agentais (37 pav.). SMOS kuriama MySQL duomenų bazių pagrindu. Projektuojama APMS naudos MO, kurie saugomi įvairiose jau egzistuojančiose anksčiau sukurtose MO saugyklose. Adaptavimas bus paremtas metaduomenimis (žr. 2.4. skyrelį). Tam, kad būtų naudojami tik kokybiški ir mokomojo dalyko kontekstą atitinkantys MO, jų metaduomenų informacija pirmiausiai išsaugoma specializuotoje SMOS. Šiuos MO pagal apibrėžtas adaptavimo taisykles naudos individualias savybes turintys besimokantieji. Adaptyviosiose sistemose besimokančiojo modelis pateikia besimokančiojo vaizdą ir taiko mokymosi medžiagą bei veiklas pagal individualias besimokančiojo savybes bei ankstesnę mokymosi patirtį (2.4. skyrelis), tačiau APMS bus naudojama besimokančiojo profilyje saugoma informacija apie jo savitą mokymosi stilių (pagal *Herrmann*'o klasifikaciją).

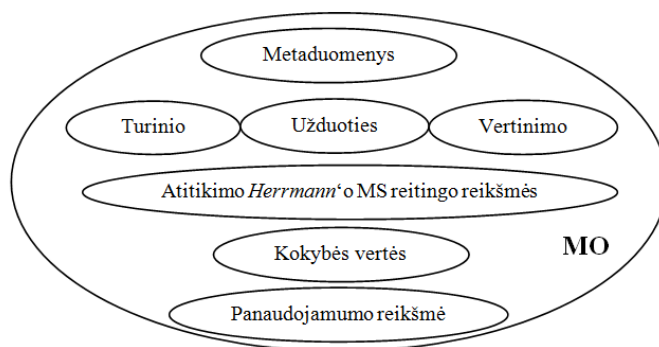


37 pav. APMS konteksto modelis

Bet kuris MO aprašomas metaduomenimis bei padeda ugdyti atitinkamą dalykinę kompetenciją (numatytą mokymosi rezultatą). Atsižvelgiant į MO tipą ir panaudojimo projektuojamoje APMS paskirtį, MO suskirstysime į 3 rūšis: turinio (MO_T) – MO padedantys mokytis mokomojo dalyko programoje numatytas temas; užduoties (MO_U) – MO skirti mokomojo dalyko praktiniams įgūdžiams ugdyti; vertinimo (MO_V) – MO leidžiantys įvertinti mokomojo dalyko programoje numatytų dalykinių kompetencijų pasiekimo lygį. MO priskyrimo rūšiai informacija bus saugoma kaip atskira MO žymė SMOS. Kadangi APMS veikimui reikalinga ne tik MO metaduomenų informacija, kuria aprašomi MO prieš juos patalpinant į saugyklas, bus kuriama nauja SMOS su papildomomis žymėmis. Taigi, SMOS yra saugomi MO:

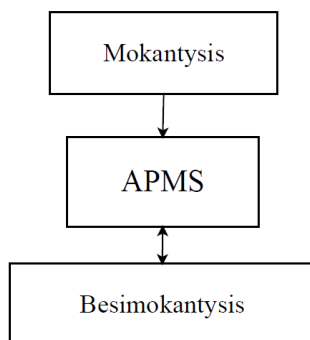
- MO metaduomenys;
- MO priklausymo tam tikrai rūšiai (turinio, užduoties, vertinimo) žymė;
- MO tipo atitikimo *Herrmann*'o mokymosi stiliui reikšmių žymės (tai ekspertinio vertinimo metu gautos ir MO tipų duomenų bazėje (DB_{MT}) išsaugotos procentinės vertės, nurodančios kiek tam tikro tipo MO yra tinkamas *Herrmann*'o mokymosi stiliui, 3.2.2. skyrelis);
- MO kokybės vertinimo žymės (MO naudotojų reitingų reikšmės ir parsisiuntimų skaičiaus vertė, 3.2.3 skyrelis),
- MO panaudojamumo žymė (skaitinė reikšmė, nurodanti kiek kartų MO buvo panaudotas).

SMOS saugomo MO struktūra matoma 38 paveiksle.



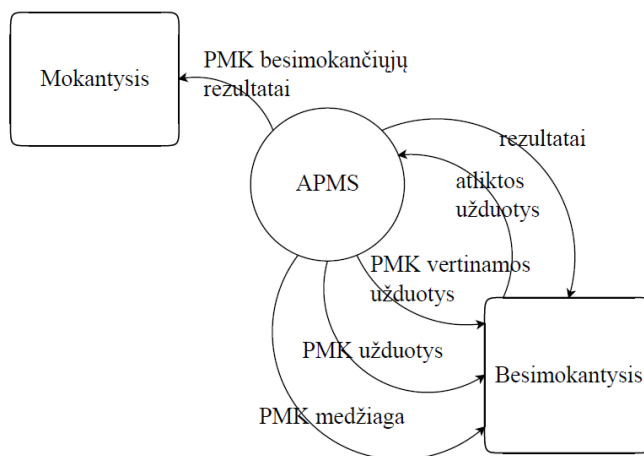
38 pav. MO duomenų struktūra

APMS naudosis (39 pav.) tiek mokantysis (PMK prižiūrintis) asmuo, kuris pagal sugeneruotą PMK (3.4. skyrelis) gali stebėti besimokančiųjų mokymosi rezultatus ir dalykinių kompetencijų pasiekimo lygį, tiek besimokantysis, kuris naudosis automatikškai sugeneruotu PMK bei sieks atitinkamo numatytųjų dalykinių kompetencijų lygio: aukščiausias kompetencijų lygis bus tada, kai mokymosi rezultatai (MR) sieks 85-100 % (arba atitinkamai 9-10 dešimtbalėje vertinimo sistemoje), vidutinis kompetencijų lygis – kai MR sieks 65-84 % (arba 7-8), minimalus būtinas kompetencijų lygis – kai MR sieks 45-64 % (arba 5-6).



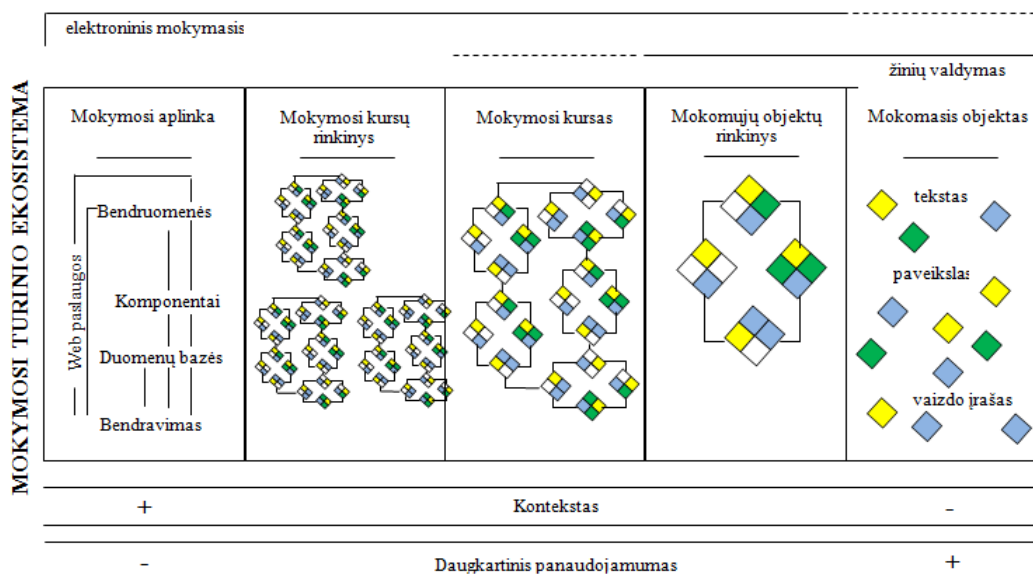
39 pav. APMS naudotojai

Kuriant prasmingą mokymosi sistemą svarbus yra sistemos kontekstas (žr. 2.1.7. skyrelį) ir santykis su vartotojais. APMS vartotojų santykis su sistema pavaizduotas 40 paveiksle.



40 pav. APMS konteksto diagrama

Net žinant mokymosi sistemos taikymo kontekstą labiau patyrusiems kūrėjams dažniausiai reikia tam tikros struktūros, pagal kurią kuriama mokymosi patirtis ir galima panaudoti neapdorotose duomenų saugyklose saugomus tarpusavyje nesusijusius mokomojo dalyko programos elementus – MO. Tinkami elementai turi būti grupuojami ir struktūruojami pagal kontekstą ir mokymosi požiūrį, kuris susijęs su specifine užduotimi. Remiantis MO turinio apimties lygmenų suskirstymu (3 priede pateiktu klasifikatoriumi MO metaduomenims aprašyti pagal LOM standartą) mokymosi turinio ekosistemą galima pavaizduoti konteksto ir daugkartinio panaudojamumo sąryšiu (41 pav.).



41 pav. Mokymosi turinio ekosistema (adaptuota pagal Hodgins, 2002)

Matome, jog kuo aukštesnis mokymosi turinio lygmuo (mokymosi kursas atitinka trečiąjį turinio apimties (agregavimo) lygį pagal LOM), tuo labiau svarbus kontekstas, tačiau mažesnis daugkartinio panaudojamumo lygis. Tuo tarpu žemesnio lygio elementai (MO yra pirmojo turinio apimties agregavimo lygio pagal LOM) turi didesnę daugkartinio panaudojamumo lygį. Taigi kuriant mokymosi kursą gali būti naudojami anksčiau sukurti įvairiose MO saugyklose saugomi kokybiški MO, kurie yra lengvai modifikuojami, lokalizuojami ir gali būti naudojami įvairiuose mokymosi kontekstuose (Kurilovas, Šerikovienė, Vuorikari, 2014).

3.2. Mokomųjų objektų išrinkimas pagal mokymosi stilių

3.2.1. Programavimo mokomieji objektai

Kaip buvo išsiaiškinta mokslinės literatūros analizės bei besimokančiųjų apklausos metu labai svarbus yra mokymosi individualizavimas. Mokymosi individualizavimas svarbus, nes (Motiejūnienė, 2008):

- individualizuotas mokymasis didina lygiateisiškumą ir efektyvumą;
- mažinamas pasiekimų skirtumas;
- atsižvelgiama į įvairių besimokančiųjų ir grupių poreikius bei savybes;
- kiekvienas besimokantysis daro maksimalią pažangą.

Mokymosi sėkmę paprastai lemia pasirinktos mokymosi veiklos. Literatūros šaltinių analizė parodė, kad mokantis programuoti būtinas programavimo srities konceptų supratimas, jų panaudojimo išmanymas, įvairių pavyzdžių studijavimas, taip pat svarbus besimokančiojo aktyvus dalyvavimas sprendžiant panašias užduotis, todėl programavimo mokymuisi yra tinkamos mokymosi veiklos:

- informacijos šaltinių (sąvokų, terminų, konceptų) studijavimas;
- pavyzdžių studijavimas;
- grafinės informacijos studijavimas;
- vaizdo medžiagos studijavimas;
- modeliavimas bei imitavimas;

- užduočių sprendimas;
- problemų sprendimas;
- projektų rengimas.

Mokymosi veiklos glaudžiai susijusios su mokymuisi naudojamais MO. Remiantis 3 priede pateiktu klasifikatoriumi MO metaduomenims aprašyti pagal LOM standartą ir atsižvelgiant į aukščiau nurodytas mokymosi veiklas, programavimo mokymuisi yra tinkami 12 lentelėje pateikti MO tipai.

12 lentelė. MO tipai, tinkami programavimo mokymui (pagal LOM)

Kodas	Pavadinimas lietuvių kalba	Aprašymas (angliškas pavadinimas)*
5.2.01.0	testas	Testas, anketa, apklausa ir pan. žinių patikrinimui (<i>assessment</i>)
5.2.04.0	demonstracija	Mokomojo objekto vizuali priemonė (<i>demonstration</i>)
5.2.05.0	pratybos, praktinė užduotis	Praktinius įgūdžius formuojanti priemonė (<i>drill and practice</i>)
5.2.06.0	tyrinėjimas	Mokomojo objekto pažinimas, jį stebint, tiriant, bandant, sprendžiant iškeltas problemas ir formuluojant išvadas (pažinimas eksperimentuojant) (<i>exploration</i>)
5.2.07.0	modeliavimas (imitavimas)	Modelis, iliustracija, animacija, imituojantis pavyzdys ir pan. (<i>simulation</i>)
5.2.10.1	garso įrašas (komponentė)	Garso įrašas, skirtas mokymuisi arba mokomojo objekto kūrimui (<i>audio</i>)
5.2.10.2	paveikslas (komponentė)	Paveikslas, skirtas mokymuisi arba mokomojo objekto kūrimui (<i>image</i>)
5.2.10.3	vaizdo įrašas (komponentė)	Videofilmas, skirtas mokymuisi arba mokomojo objekto kūrimui (<i>video</i>)
5.2.21.0	projektas	Projekto aprašymas, turintis kokybinius ir kiekybinius rodiklius, unikalų tikslo pasiekimų kelią. Pvz., tarpdalykinių ryšių, bendradarbiavimo, klasės veiklos, konkurso, egzaminų, nacionalinis arba Europos projektas mokymosi tikslais ir kt. (<i>project</i>)
5.2.26.0	glosarijus (aiškinamųjų terminų žodynas)	Vienakalbis, dvikalbis arba daugiakalbis paaiškinantis terminus, žodžius, ženklus žodynas (<i>glossary</i>)
5.2.27.0	žodynas	Žodžių arba dalykinių sričių terminų sąrašas su vienos ar kelių kalbų atitikmenimis (<i>reference</i>)
5.2.28.0	enciklopedinis šaltinis	Enciklopedija, duomenų bazė, informacinė sistema ir kt. šaltinis temos informacijai gauti (<i>reference</i>)
5.2.29.0	pateiktis	Skaidrėmis teikiama informacija, apipavidalinant ją programine įranga (<i>presentation</i>)
5.2.30.1	interneto tinklaraštis	Autoriaus arba grupės kūrėjų leidinys skelbiamas kompiuterių tinkle (žiniatinklyje) (<i>weblog</i>)
5.2.30.2	interneto vikis	Žiniatinklio rūšis, kurios turinį naudotojas gali papildyti, patikslinti, kurti (Vikiu taip pat vadinama ir programinė įranga tokių svetainių kūrimui) (<i>wiki</i>)

* Pagal LRE Learning resours types žodyno reikšmę

Kiti klasifikatoriuje esantys MO tipai, kurie PMK konteksto prasme nėra tinkami naudoti, buvo atmeti. Atmetama buvo analizuojant MO tipo aprašymą ir taikymo sritis. Atmetimo priežastys išvardytos 13 lentelėje.

13 lentelė. **Atmeti MO tipai** (pagal LOM)

Kodas	Pavadinimas lietuvių kalba (pavadinimas anglų kalba)	Aprašymas	Atmetimo priežastis
5.2.02.0	klausimų rinkinys (<i>question bank</i>)	Klausimų aibė testui sudaryti	APMS bus nustatyti MO svoriai, todėl testo smulkesnės dalys nenaudotinos
5.2.03.0	testo elementas (<i>assessment item</i>)	Testo dalis: klausimas ir jo teisingas atsakymas	APMS bus nustatyti MO svoriai, todėl testo smulkesnės dalys nenaudotinos
5.2.08.0	edukacinis žaidimas (<i>educational game</i>)	Edukacinis žaidimas, suteikiantis dalykinių žinių ir lavinantis gebėjimus	Tokios veiklos APMS nenagrinėjamos, todėl MO tipas nenaudotinas
5.2.09.0	išteklų paketas (<i>resource pack</i>)	Kursui, programai, moduliui, pamokos temai skirtas rinkinys priemonių su instrukcijomis mokytojui ir mokiniui	APMS bus nustatyti MO svoriai, todėl naudoti šio tipo MO netikslinga
5.2.11.0	taikomoji programa (<i>application</i>)	Kompiuterinė programa mokymui ir praktinėms užduotims, pvz: tekstų rengyklė, skaičiuoklė, grafikos programa, interneto naršyklė ir pan.	Tipas neleidžia nustatyti konkrečios paskirties, todėl jis nenaudotinas
5.2.12.0	įrankis (<i>tool</i>)	Redagavimo, piešimo ir kitų funkcijų kompiuterinės programos naudotojo kūrybai	Tokios veiklos APMS nenagrinėjamos, todėl MO tipas nenaudotinas
5.2.13.0	programos, modulio, skyriaus planas (<i>course/module programme</i>)	Programos, jos modulio, skyriaus, temos nustatytam laikotarpiui mokymo planas, nurodant tikslus, uždavinius, mokytojo ir mokinių veiklą, priemones	APMS bus nustatyti MO svoriai, todėl naudoti šio tipo MO netikslinga
5.2.14.0	pamokos ar pamokų ciklo planas (<i>lesson plan</i>)	Dalyko pamokos ar pamokų ciklo planas, nurodant tikslus, uždavinius, mokytojo ir mokinių veiklą, priemones	APMS bus nustatyti MO svoriai, todėl naudoti šio tipo MO netikslinga
5.2.15.0	pedagogo vadovas (<i>educator guide</i>)	Rekomendacinė paremta gerąja patirtimi pagalbos knyga pedagogui	APMS bus nustatyti MO svoriai, todėl naudoti šio tipo MO netikslinga
5.2.16.0	vertinimas (atestavimas) (<i>evaluation form</i>)	Vertinimo metodai ir jų taikymo priemonė	APMS automatiškai vertinamos besimokančiojo dalykinės kompetencijos, todėl naudoti šio tipo MO netikslinga
5.2.17.0	naudotojo vadovas (<i>user guide</i>)	Instrukcija naudotojui kaip dirbti su mokomojo objekto priemone	APMS tikslas pateikti programavimo mokymosi medžiagą, o ne naudojimosi MO priemone

Kodas	Pavadinimas lietuvių kalba (pavadinimas anglų kalba)	Aprašymas	Atmetimo priežastis
5.2.18.0	atvejo tyrimas (<i>case study</i>)	Atvejo, proceso, temos tyrimo aprašymas, pritaikytas diskusijai, analizei, pristatymui	Tokios veiklos APMS nenagrinėjamos, todėl MO tipas nenaudotinas
5.2.19.0	paieška (<i>enquiry-oriented activity</i>)	Informacijos paieškai, apdorojimui, tvarkymui, arba pritaikymui skirtas aprašymas	APMS pateikia atrinktus MO, todėl paieškos naudojimas netikslingas
5.2.20.0	eksperimentas (<i>experiment</i>)	Eksperimento, hipotezės, tiesos patikrinimo arba naujo atradimui skirtas aprašymas	Tokios veiklos APMS nenagrinėjamos, todėl MO tipas nenaudotinas
5.2.22.0	vaidmenų žaidimas (<i>role play</i>)	Mokymosi iš žaidimo vaidmenų aprašymas	Tokios veiklos APMS nenagrinėjamos, todėl MO tipas nenaudotinas
5.2.23.0	popamokinė veikla (neformalusis ugdymas) (<i>after-school activities</i>)	Organizuotos popamokinės veiklos (neformaliojo vaikų ugdymo) aprašymas, skirtas asmeninėms, socialinėms, edukacinėms, profesinėms kompetencijoms ugdyti	Tokios veiklos APMS nenagrinėjamos, todėl MO tipas nenaudotinas
5.2.24.0	straipsnis (<i>article</i>)	Metodinis leidinio straipsnis, jo aprašymas	Šis MO tipas galėtų būti taikomas aukštesnio lygio kompetencijoms ugdyti siekiant ekspertų lygio, APMS tai nenagrinėjama
5.2.25.0	skaitmeninis leidinys (<i>digitize publication</i>)	Knyga, vadovėlis, straipsnis skaitmenine (elektronine) forma	Per stambus MO, nagrinėjant visą pateikiamą informaciją būtų nuklystama nuo temos kompetencijų ugdymo
5.2.30.3	mokymosi svetainė (<i>web resource / learning web site</i>)	Tarpusavyje susieti keli arba daugiau tinklapių, turintys apibendrinantį pradinį puslapį (home page) mokymosi tikslais	Per stambus MO, nagrinėjant visą pateikiamą informaciją būtų nuklystama nuo temos kompetencijų ugdymo
5.2.30.4	teminė mokymosi svetainė (<i>web resource / thematic website</i>)	Visuotinė vienos srities, temos svetainė skirta visuomenės ugdymui ir švietimui	Per stambus MO, nagrinėjant visą pateikiamą informaciją būtų nuklystama nuo temas kompetencijų ugdymo
5.2.30.5	konkursų, olimpiadų svetainė (<i>web resource / competition website</i>)	Informacinė interneto svetainė apie nuolat vykstančius konkursus, olimpiadas, varžybas, egzaminus	Šis MO tipas galėtų būti taikomas aukštesnio lygio kompetencijoms ugdyti siekiant ekspertų lygio, APMS tai nenagrinėjama
5.2.30.6	kitas interneto išteklius (<i>other web resource</i>)	Kita klasifikatoriuje nenurodytas žiniatinklis skirtas mokymui	Tipas neleidžia nustatyti konkrečios paskirties, nagrinėjant visą pateikiamą informaciją būtų nuklystama nuo temas kompetencijų ugdymo
5.2.31.0	kitas išteklius (<i>other</i>)	Kitas klasifikatoriuje nenurodytas mokomojo objekto tipas	Tipas neleidžia nustatyti konkrečios paskirties

Taigi, įvertinus tinkamų programavimo mokymuisi MO tipų išrinkimo rezultatus, MO tipų aibę MOT (mokomojo objekto tipas) sudaro:

MOT = {T, DE, PU, TYR, M, GAR, PAV, VAI, PR, G, Z, ES, PAT, TIN, VIK}

Čia:

T	– testas
DE	– demonstracija
PU	– pratybos, praktinė užduotis
TYR	– tyrinėjimas
M	– modeliavimas
GAR	– garso įrašas
PAV	– paveikslas
VAI	– vaizdo įrašas
PR	– projektas
G	– glosarijus (aiškinamųjų terminų žodynas)
Z	– žodynas
ES	– enciklopedinis šaltinis
PAT	– pateiktis
TIN	– interneto tinklaraštis
VIK	– interneto vikis

Kadangi pagal paskirtį mokymosi kurse MO tipas gali būti priskiriamas atitinkamai rūšiai (turinio, užduoties arba vertinimo), tai aibę MOR (mokomųjų objektų rūšies poaibiai) sudaro atitinkami poaibiai:

MOR = {MO_T; MO_U; MO_V}

Čia:

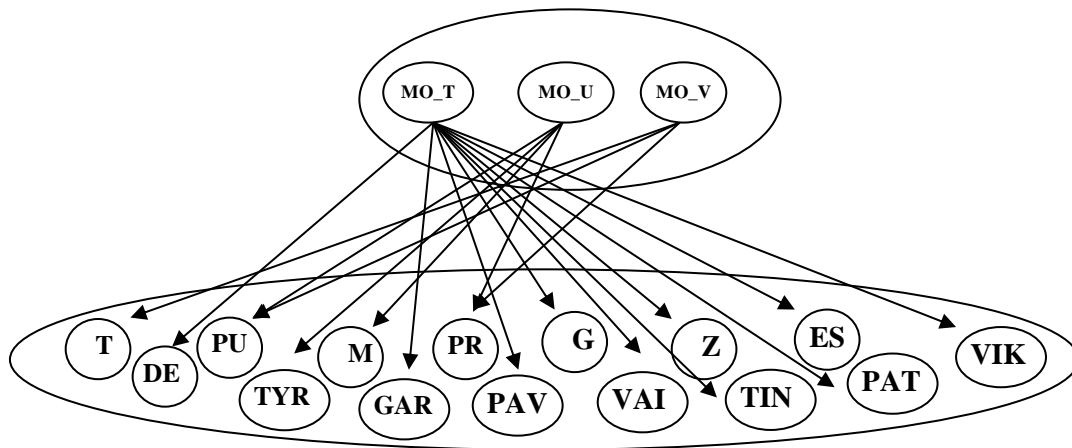
MO_T	– turinio mokomasis objektas
MO_U	– užduoties mokomasis objektas
MO_V	– vertinimo mokomasis objektas

Poaibiai bei jiems priskiriami MO tipai pasiskirstytų taip:

MO_T	– turinio mokomasis objektas
	DE – demonstracija
	GAR – garso įrašas
	PAV – paveikslas
	VAI – vaizdo įrašas
	G – glosarijus (aiškinamųjų terminų žodynas)
	Z – žodynas
	ES – enciklopedinis šaltinis
	PAT – pateiktis
	TIN – interneto tinklaraštis
	VIK – interneto vikis

MO_U	– užduoties mokomasis objektas
	PU – pratybos, praktinė užduotis
	TYR – tyrinėjimas
	M – modeliavimas (imitavimas)
	PR – projektas
MO_V	– vertinimo mokomasis objektas
	T – testas
	PU – pratybos, praktinė užduotis
	PR – projektas

Pritaikius aibių atvaizdavimo metodą, aibę MOT galima suskirti į 3 poaibius MOR (42 pav.).



42 pav. MO tipų priskyrimo rūšiai atvaizdavimas

Kaip buvo minėta anksčiau (2.5. skyrelis), taikant *Herrmann*‘o mokymosi stilių klasifikaciją išskiriami 4 skirtingi besimokančiųjų tipai, kurie atitinka kvadrantus (A, B, C ir D). Pagal kiekvieno stiliaus atstovo savybes bei mokymuisi tinkamas mokymosi veiklas (8 lentelė) besimokantiejiems galima priskirti labiausiai jiems tinkančius MO tipus. Šis priskyrimas taip pat priklauso nuo kvadranto išreikštumo lygio. Jei pas besimokantįjį vienodai ryškios kelių kvadrantų savybės, jam gali būti tinkami tuos kelis kvadrantus atitinkantys MO tipai.

Taigi, bet kurio X besimokančiojo mokymosi stilių MS_X (pagal HBDI) galima apibrėžti kaip 4 skaitines reikšmes turintį aibės elementą:

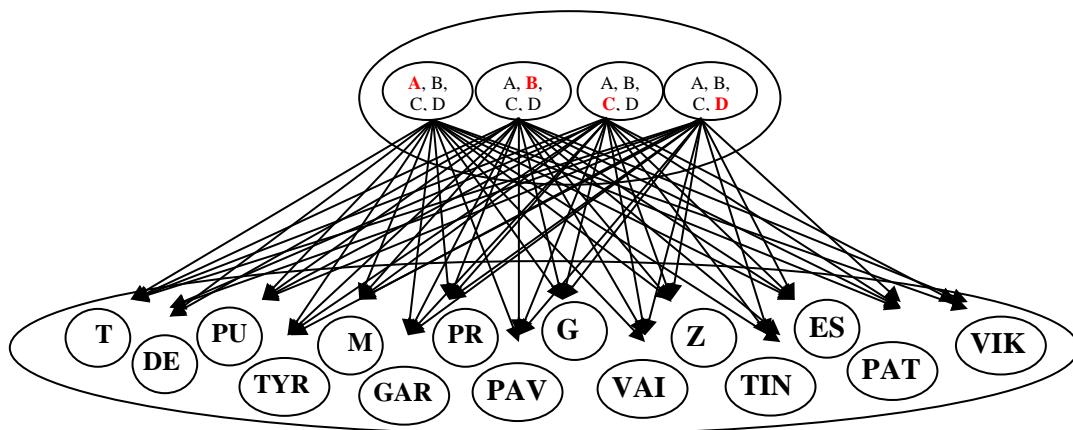
$$MS_X = \{KV_A; KV_B; KV_C; KV_D\},$$

čia – KV_A, KV_B, KV_C, KV_D yra atitinkamo kvadranto išreikštumo lygis, t. y. klausimynu nustatyta besimokančiojo kvadranto (A, B, C arba D) procentinė reikšmė, kuri yra saugoma besimokančiojo profilyje.

Besimokančiajam būdingi kvadrantą (-us) atitinkantys bruožai, kurio (-ių) skaitinė reikšmė atitinka vieną iš verčių: 1 = savybės stipriai pasireiškia (HBDI testo rezultatas > 66), 2 = savybės mažiau būdingos, tačiau pasireiškiančios, kai reikalauja situacija (rezultatas nuo 34 iki 66), 3 = galimas savybių vengimas (rezultatas < 34).

Tyrimais įrodyta, kad kompiuterių mokslų atstovams yra būdingas B kvadranto dominavimas (Lumsdaine, Lumsdaine, 1995). Tikslingai nukreipiant atlikti reikiamas veiklas, kvadranto dominavimas gali būti pakeičiamas. Taip pat įrodyta, kad į inžinerinių mokslų mokymosi turinį būtina įtraukti C ir D kvadrantams būdingas veiklas (Lumsdaine, Lumsdaine, 1995), kad būtų ugdomos savybės, reikalingos kitoms sritims. Taigi, nepriklausomai nuo to, kurio kvadranto savybės yra labiausiai išreikštos, besimokančiajam turi būti siūlomi visų tipų MO. Besimokančiajam automatiškai (pateikiant MO sąrašą) pirmiausiai turi būti siūlomi labiausiai jo mokymosi stilių atitinkantys MO tipai ir konkretūs MO.

Panaudojant aibių atvaizdavimo metodą, kiekvienam mokymosi stiliaus aibės {KV_A, KV_B, KV_C, KV_D} elementui galima priskirti MO tipų aibės {T, DE, PU, TYR, M, GAR, PAV, VAI, PR, G, Z, ES, PAT, TIN, VIK} elementą (43 pav.).



43 pav. Mokymosi stiliaus sąsajos su MO tipu atvaizdavimas

Tuo pasireiškia ryšys tarp besimokančiojo profilio informacijos ir programinių agentų sistemos (37 pav.).

3.2.2. Mokomųjų objektų tipų reitingavimo metodas

Siekiant nustatyti MO tipų tinkamumą tam tikram mokymosi stiliui pagal *Herrmann*'o kvadrantus, buvo sudaryta ekspertų grupė, įvertinusi anksčiau pateiktus programavimo mokymuisi tinkamus MO tipus naudojant lingvistinę 4 reikšmių vertinimo skalę:

4 – MO tipas yra labai tinkamas kvadrantui;

3 – MO tipas yra tinkamas kvadrantui;

2 – MO tipas yra vidutiniškai tinkamas kvadrantui;

1 – MO tipas gali būti naudojamas.

MO vertinimo ekspertais buvo parinkti programavimo mokymo patirtį bei informatikos inžinerijos krypties daktaro mokslo laipsnį turintys asmenys, informatikos inžinerijos krypties doktorantai ir edukologijos doktorantai. MO tipus vertino 18 ekspertų iš Lietuvos ir užsienio šalių (Anglijos, Slovėnijos, Suomijos, Austrijos). Ekspertams buvo pateiktas *Herrmann*'o kvadrantus atitinkančių savybių aprašas bei prašoma nurodyti (parenkant vieną iš 4) vertinimo reikšmę (5 priedas). Ekspertų parinkimų skaičius matomas 6 priede pateiktoje 28 lentelėje.

Kvadrantą atitinkančių kiekvienos vertinimo skalės reikšmės ekspertų parinkimų skaičių galima pavaizduoti matrica, čia k_r – vertinimo skalės reikšmė, $r = 1, 2, \dots, 4$, MOT_t – mokomojo objekto tipas, kur $t = 1, 2, \dots, n$), x_{tr} – ekspertų parinkimų skaičius:

	k_1	k_2	k_3	k_4
MOT_1	x_{11}	x_{12}	x_{13}	x_{14}
MOT_2	x_{21}	x_{22}	x_{23}	x_{24}
...
MOT_n	x_{n1}	x_{n2}	...	x_{n4}

Vėliau pagal 1 formulę buvo apskaičiuojamas MO tipo (MOT_t , kur $t = 1, 2, \dots, n$) svertinis kvadrantą (MS_i , kur $i = 1, 2, \dots, 4$) atitinkantis ekspertų vertinimų vidurkis vid_{ki} .

$$vid_{ti} = \frac{\sum_{j=1}^4 k_j x_{tj}}{\sum_{j=1}^4 k_j} \quad (1)$$

Gauti rezultatai gali būti vaizduojami matrica:

	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>
<i>MOT</i> ₁	<i>vid</i> ₁₁	<i>vid</i> ₁₂	<i>vid</i> ₁₃	<i>vid</i> ₁₄
<i>MOT</i> ₂	<i>vid</i> ₂₁	<i>vid</i> ₂₂	<i>vid</i> ₂₃	<i>vid</i> ₂₄
...
<i>MOT</i> _{<i>n</i>}	<i>vid</i> _{<i>n</i>1}	<i>vid</i> _{<i>n</i>2}	...	<i>vid</i> _{<i>n</i>4}

Toliau buvo apskaičiuota procentinė MO tipo atitikimo *Herrmann*'o kvadrantui vertė (%). Kadangi MO tipas labiausiai tinka besimokančiojo mokymosi stiliui tada, kai jo skaitinė vertė lygi 4, tai procentinė vertė gaunama pritaikius proporcijos taisyklę (2 formulė):

$$\% = \frac{vid_{ti} \cdot 100}{4} \quad (2)$$

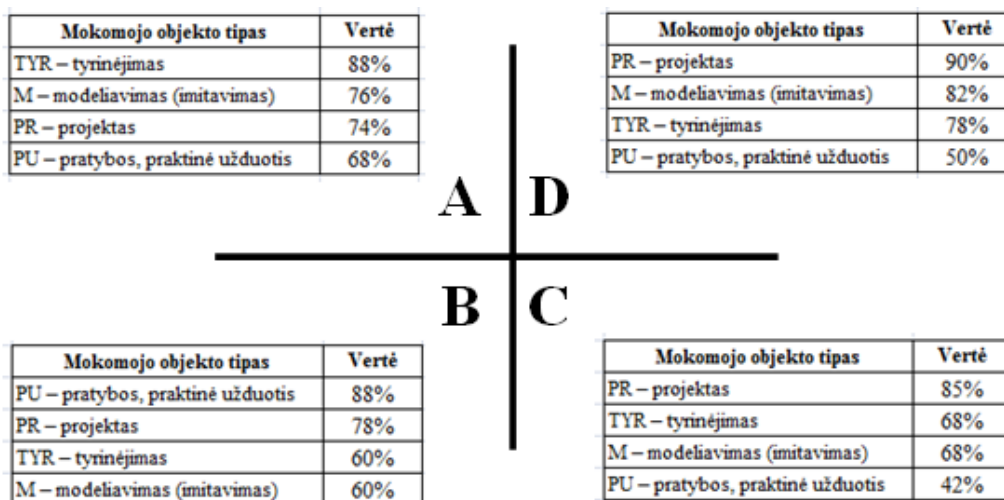
Atsižvelgiant į gautus rezultatus galima sudaryti MO tipų rūšių poaibus MOR (MO_T, MO_U ir MO_V) sudarančių MO tipų reitinguotą išsidėstymą.

Turinio MO poaibiui MO_T priklausančių MO tipų reitinguotas išdėstymas pateiktas 44 paveiksle.

<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr><th style="text-align: left;">Mokomojo objekto tipas</th><th style="text-align: left;">Vertė</th></tr> </thead> <tbody> <tr><td>G – glosarijus</td><td>85%</td></tr> <tr><td>ES – enciklopedinis šaltinis</td><td>85%</td></tr> <tr><td>Z – žodynas</td><td>79%</td></tr> <tr><td>D – demonstracija</td><td>68%</td></tr> <tr><td>PAT – pateiktis</td><td>68%</td></tr> <tr><td>TIN – interneto tinklaraštis</td><td>61%</td></tr> <tr><td>PAV – paveikslas</td><td>60%</td></tr> <tr><td>GAR – garso įrašas</td><td>58%</td></tr> <tr><td>VAI – vaizdo įrašas</td><td>58%</td></tr> <tr><td>VIK – interneto vikis</td><td>58%</td></tr> </tbody> </table>	Mokomojo objekto tipas	Vertė	G – glosarijus	85%	ES – enciklopedinis šaltinis	85%	Z – žodynas	79%	D – demonstracija	68%	PAT – pateiktis	68%	TIN – interneto tinklaraštis	61%	PAV – paveikslas	60%	GAR – garso įrašas	58%	VAI – vaizdo įrašas	58%	VIK – interneto vikis	58%	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr><th style="text-align: left;">Mokomojo objekto tipas</th><th style="text-align: left;">Vertė</th></tr> </thead> <tbody> <tr><td>PAV – paveikslas</td><td>93%</td></tr> <tr><td>VAI – vaizdo įrašas</td><td>83%</td></tr> <tr><td>D – demonstracija</td><td>75%</td></tr> <tr><td>VIK – interneto vikis</td><td>74%</td></tr> <tr><td>TIN – interneto tinklaraštis</td><td>71%</td></tr> <tr><td>PAT – pateiktis</td><td>65%</td></tr> <tr><td>GAR – garso įrašas</td><td>64%</td></tr> <tr><td>ES – enciklopedinis šaltinis</td><td>40%</td></tr> <tr><td>G – glosarijus</td><td>33%</td></tr> <tr><td>Z – žodynas</td><td>33%</td></tr> </tbody> </table>	Mokomojo objekto tipas	Vertė	PAV – paveikslas	93%	VAI – vaizdo įrašas	83%	D – demonstracija	75%	VIK – interneto vikis	74%	TIN – interneto tinklaraštis	71%	PAT – pateiktis	65%	GAR – garso įrašas	64%	ES – enciklopedinis šaltinis	40%	G – glosarijus	33%	Z – žodynas	33%		
Mokomojo objekto tipas	Vertė																																														
G – glosarijus	85%																																														
ES – enciklopedinis šaltinis	85%																																														
Z – žodynas	79%																																														
D – demonstracija	68%																																														
PAT – pateiktis	68%																																														
TIN – interneto tinklaraštis	61%																																														
PAV – paveikslas	60%																																														
GAR – garso įrašas	58%																																														
VAI – vaizdo įrašas	58%																																														
VIK – interneto vikis	58%																																														
Mokomojo objekto tipas	Vertė																																														
PAV – paveikslas	93%																																														
VAI – vaizdo įrašas	83%																																														
D – demonstracija	75%																																														
VIK – interneto vikis	74%																																														
TIN – interneto tinklaraštis	71%																																														
PAT – pateiktis	65%																																														
GAR – garso įrašas	64%																																														
ES – enciklopedinis šaltinis	40%																																														
G – glosarijus	33%																																														
Z – žodynas	33%																																														
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr><th style="text-align: left;">Mokomojo objekto tipas</th><th style="text-align: left;">Vertė</th></tr> </thead> <tbody> <tr><td>G – glosarijus</td><td>82%</td></tr> <tr><td>Z – žodynas</td><td>82%</td></tr> <tr><td>ES – enciklopedinis šaltinis</td><td>79%</td></tr> <tr><td>PAT – pateiktis</td><td>71%</td></tr> <tr><td>D – demonstracija</td><td>69%</td></tr> <tr><td>VIK – interneto vikis</td><td>63%</td></tr> <tr><td>GAR – garso įrašas</td><td>61%</td></tr> <tr><td>VAI – vaizdo įrašas</td><td>54%</td></tr> <tr><td>PAV – paveikslas</td><td>53%</td></tr> <tr><td>TIN – interneto tinklaraštis</td><td>53%</td></tr> </tbody> </table>	Mokomojo objekto tipas	Vertė	G – glosarijus	82%	Z – žodynas	82%	ES – enciklopedinis šaltinis	79%	PAT – pateiktis	71%	D – demonstracija	69%	VIK – interneto vikis	63%	GAR – garso įrašas	61%	VAI – vaizdo įrašas	54%	PAV – paveikslas	53%	TIN – interneto tinklaraštis	53%	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr><th style="text-align: left;">Mokomojo objekto tipas</th><th style="text-align: left;">Vertė</th></tr> </thead> <tbody> <tr><td>VAI – vaizdo įrašas</td><td>78%</td></tr> <tr><td>VIK – interneto vikis</td><td>75%</td></tr> <tr><td>TIN – interneto tinklaraštis</td><td>72%</td></tr> <tr><td>GAR – garso įrašas</td><td>71%</td></tr> <tr><td>PAV – paveikslas</td><td>64%</td></tr> <tr><td>PAT – pateiktis</td><td>60%</td></tr> <tr><td>D – demonstracija</td><td>58%</td></tr> <tr><td>G – glosarijus</td><td>33%</td></tr> <tr><td>ES – enciklopedinis šaltinis</td><td>33%</td></tr> <tr><td>Z – žodynas</td><td>32%</td></tr> </tbody> </table>	Mokomojo objekto tipas	Vertė	VAI – vaizdo įrašas	78%	VIK – interneto vikis	75%	TIN – interneto tinklaraštis	72%	GAR – garso įrašas	71%	PAV – paveikslas	64%	PAT – pateiktis	60%	D – demonstracija	58%	G – glosarijus	33%	ES – enciklopedinis šaltinis	33%	Z – žodynas	32%		
Mokomojo objekto tipas	Vertė																																														
G – glosarijus	82%																																														
Z – žodynas	82%																																														
ES – enciklopedinis šaltinis	79%																																														
PAT – pateiktis	71%																																														
D – demonstracija	69%																																														
VIK – interneto vikis	63%																																														
GAR – garso įrašas	61%																																														
VAI – vaizdo įrašas	54%																																														
PAV – paveikslas	53%																																														
TIN – interneto tinklaraštis	53%																																														
Mokomojo objekto tipas	Vertė																																														
VAI – vaizdo įrašas	78%																																														
VIK – interneto vikis	75%																																														
TIN – interneto tinklaraštis	72%																																														
GAR – garso įrašas	71%																																														
PAV – paveikslas	64%																																														
PAT – pateiktis	60%																																														
D – demonstracija	58%																																														
G – glosarijus	33%																																														
ES – enciklopedinis šaltinis	33%																																														
Z – žodynas	32%																																														

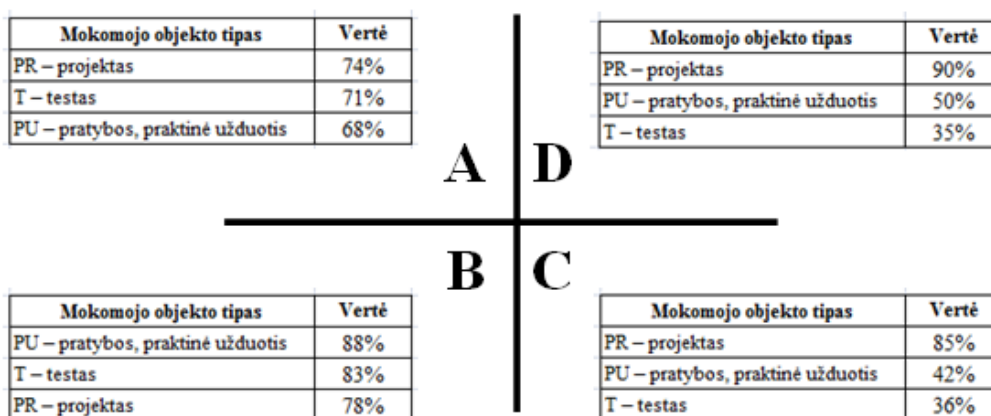
44 pav. Turinio MO išdėstymas

Užduoties MO poaibiui MO_U priklausančių MO tipų reitinguotas išdėstymas pateiktas 45 paveiksle.



45 pav. Užduoties MO išdėstymas

Vertinimo MO poaibiui MO_V priklausančių MO tipų reitinguotas išdėstymas pateiktas 46 paveiksle.



46 pav. Vertinimo MO išdėstymas

Remiantis gautais rezultatais galima sudaryti MO tipų reitingų matricą (14 lentelė).

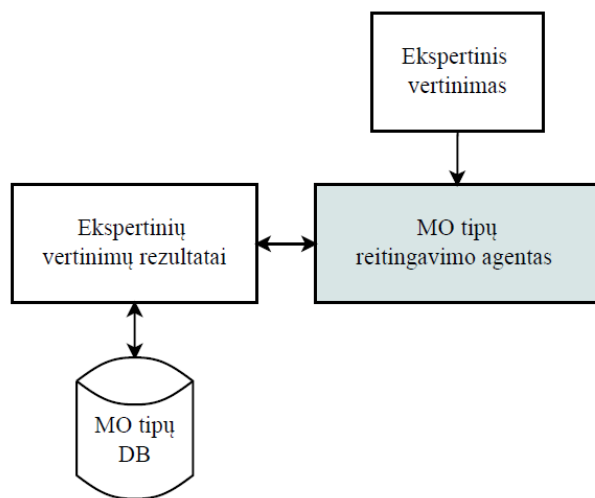
14 lentelė. MO tipų reitingai pagal kvadrantus

Poaibis	Mokomojo objekto tipas	Reitingo reikšmė pagal kvadrantą			
		A	B	C	D
MO_T	– turinio mokomasis objektas				
	D – demonstracija	3	4	7	3
	GAR – garso įrašas (komponentė)	6	6	4	7
	PAV – paveikslas (komponentė)	5	8	5	1
	VAI – vaizdo įrašas (komponentė)	6	7	1	2
	G – glosarijus (aiškinamųjų terminų žodynas)	1	1	8	9

Poaibis	Mokomojo objekto tipas	Reitingo reikšmė pagal kvadrantą			
		A	B	C	D
	Z – žodynas	2	1	9	9
	ES – enciklopedinis šaltinis	1	2	8	8
	PAT – pateiktis	3	3	6	6
	TIN – interneto tinklaraštis	4	8	3	5
	VIK – interneto vikis	6	5	2	4
MO_U	– užduoties mokomasis objektas				
	PU – pratybos, praktinė užduotis	3	1	4	4
	TYR – tyrinėjimas	1	3	2	2
	M – modeliavimas (imitavimas)	2	4	3	3
	PR – projektas	4	2	1	1
MO_V	– vertinimo mokomasis objektas				
	T – testas	2	2	3	3
	PU – pratybos, praktinė užduotis	3	1	2	2
	PR – projektas	1	3	1	1

MO tipų atitikimas pagal *Herrmann*'o mokymosi stilių yra išsaugomas MO tipų duomenų bazėje (DB_{MT}). Išsaugoma MO tipo vertinimo metu ir paskaičiuota atitikimo mokymosi stiliui ekspertinio vertinimo procentinė vertė. Šias reikšmes naudos MO išrinkimo programinis agentas, išrinktiems ir SMOS saugomiems MO priskirdamas atitinkamas žymes (pvz., A – 71 %, B – 83 %, C – 36 %, D – 35 %).

Ekspertinio vertinimo pasikeitimus (atsiradus naujiems ekspertų vertinimams) stebi ir duomenis atnaujina MO reitingavimo programinis agentas (47 pav. ir 15 lentelė).

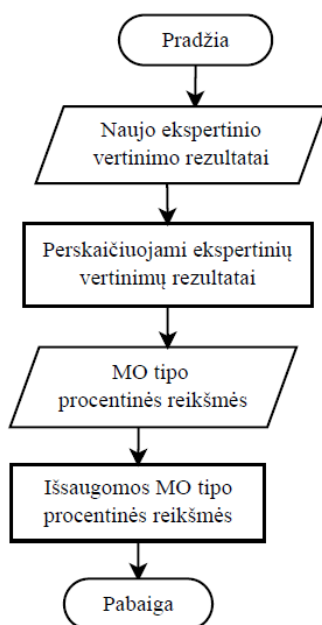


47 pav. MO tipų reitingavimo modelis

15 lentelė. MO tipų reitingavimo agento aprašas

Laukiamas aplinkos įvykis	Ekspertas užpildo ekspertinio vertinimo formą
Programinio agento veiksmai	<ol style="list-style-type: none"> 1. Iš DB_{MT} gaunami ekspertinių vertinimų apibendrinti rezultatai – atitikimo kvadrantui vertės (4, 3, 2, 1) pasirinkimų skaičius ir paskaičiuota procentinė vertė. 2. Perskaičiuojami ekspertinių vertinimų apibendrinti rezultatai atsižvelgiant į naujai gauto vertinimo rezultatus. 3. Atnaujinami ekspertinių vertinimų rezultatai įrašant į DB_{MT}.

MO tipų reitingavimo programinio agento algoritmo schema pateikta 48 paveiksle.



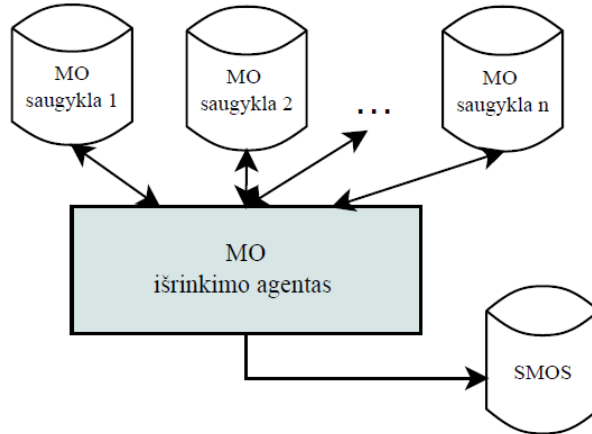
48 pav. MO tipų reitingavimo programinio agento veikimo algoritmo schema

MO tipų tinkamumo mokymosi stiliui procentines vertės vėliau bus naudojamos teikiant PMK konkrečiam besimokančiajam, kuriam nustatytas individualus mokymosi stilius (3.5. skyrelis).

3.3. Mokomųjų objektų išrinkimas ir kokybės vertinimas vartotojų atžvilgiu

Kuriant PMK yra pildoma (papildoma) SMOS. MO išrenkami iš MO saugyklų panaudojant *Harvesting* agentą. SMOS papildyti padeda MO išrinkimo programinis agentas, kuris prieš įtraukiant MO į SMOS įvertina jo kokybę. Remiantis Kurilovo, Sėrikovienės ir Vuorikari atliktų tyrimų rezultatais (2014), nustatyta, kad MO ekspertinis vertinimas parodo analogiškus rezultatus, kaip ir pačių vartotojų vertinimas MO reitinguojant

(vertinimas žvaigždutėmis) bei siunčiantis ir naudojant (parsisiuntimų skaičius). Todėl projektuojamoje sistemoje MO kokybė vertinama pagal saugyklose saugomų MO vartotojų nuomonę bei parsisiuntimų skaičių. MO išrinkimo modelis matomas 49 paveiksle. MO išrinkimo programinio agento aprašas pateiktas 16 lentelėje.

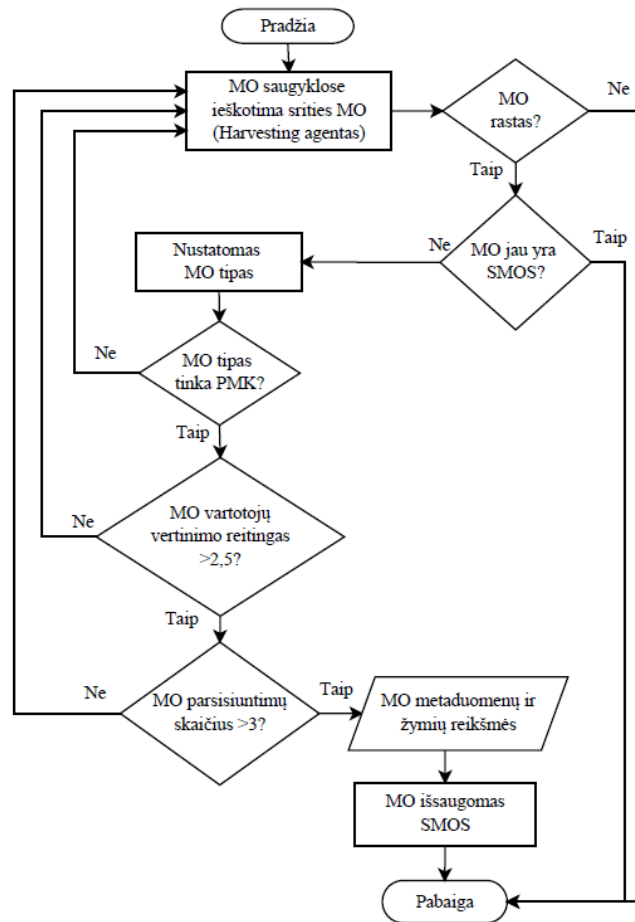


49 pav. MO išrinkimo modelis

16 lentelė. MO išrinkimo programinio agento aprašas

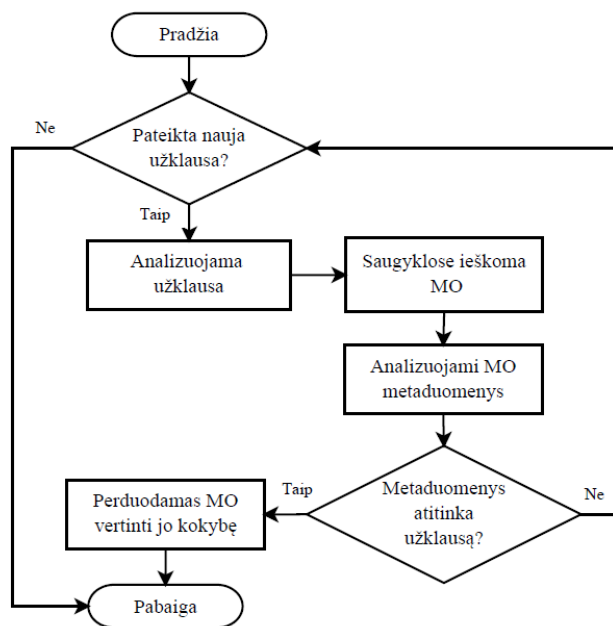
Laukiamas aplinkos įvykis	Pateikiama mokomųjų objektų paieškos užklausa
Programinio agento veiksmai	<ol style="list-style-type: none"> Atsižvelgiant į paieškos užklausa (mokomojo dalyko programos temą) pagal raktažodį ir MO metaduomenų Aprašo lauką (angl. <i>Description</i>) atliekama MO paieška MO saugyklose (paieškai atlikti naudojamas <i>Harvesting</i> metodas – programinis agentas). Surasti MO vertinami kokybės atžvilgiu: <ul style="list-style-type: none"> peržiūrima vartotojų vertinimo statistika – tinkami tik tie MO, kurių vartotojų vertinimas yra ne mažiau kaip 2,5 žvaigždutės iš 5 (kas atitinka 50 % pagal ekspertinį vertinimą); peržiūrimas MO parsisiuntimų skaičius – tinkami tik tie MO, kurie yra parsisiųsti ne mažiau kaip 3 kartus. Tinkamo MO metaduomenys išsaugomi SMOS ir įvedamos žymės apie jo vartotojo reitingą ir parsisiuntimų skaičių (pvz. vartotojų reitingas 3 žv., parsisiuntimų skaičius 5).

MO išrinkimo programinio agento veikimo algoritmo schema pateikta 50 paveiksle.



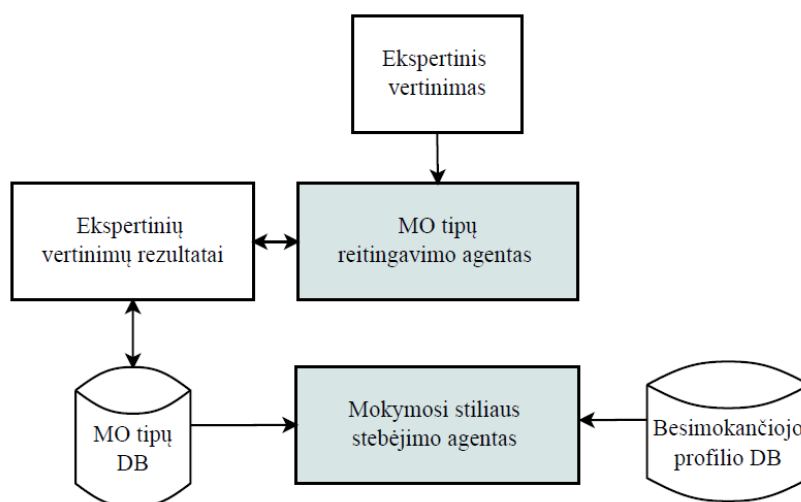
50 pav. MO išrinkimo programinio agento veikimo algoritmo schema

MO paieškai saugyklose naudojamas papildomas *Harvesting* programinis agentas. Agentui yra nurodoma, kuriose MO saugyklose turi būti atliekama paieška. APMS atveju paieška turi būti atliekama MO saugyklose, kur saugomos vartotojų vertinimo žvaigždutėmis ir parsisiuntimų skaičiaus žymės. Taip pat agentui nurodomas MO išrinkimo periodiškumas – kas kiek laiko turi būti atliekama naujų MO paieška (pvz., kartą per savaitę). *Harvesting* agentas, gavęs paieškos užklausą pagal raktinius žodžius (mokymosi programos temas) analizuodamas MO metaduomenų laukus (*Keyword* ir *Description*) ieško atitikimo. Suradus užklausą atitinkantį MO, jo metaduomenys yra parsienčiami į nurodomą vietą (APMS atveju į SMOS). *Harvesting* programinio agento veikimo algoritmo schema pateikta 51 paveiksle.



51 pav. Harvesting programinio agento veikimo algoritmo schema

Kadangi besimokančiajam gali būti būdingi kelių mokymosi stilių bruožai (pagal *Herrmann*'o kvadranto išeiškštumo lygi), tai išrenkant ir siūlant jam tinkamiausių tipų MO pirmiausiai turi būti perkaičiuota MO tipų procentinė vertė, atsižvelgiant į MO tipo atitinkamo *Herrmann*'o kvadranto procentinę vertę. Tą atlieka mokymosi stiliaus stebėjimo programinis agentas (52 pav. ir 17 lentelė).

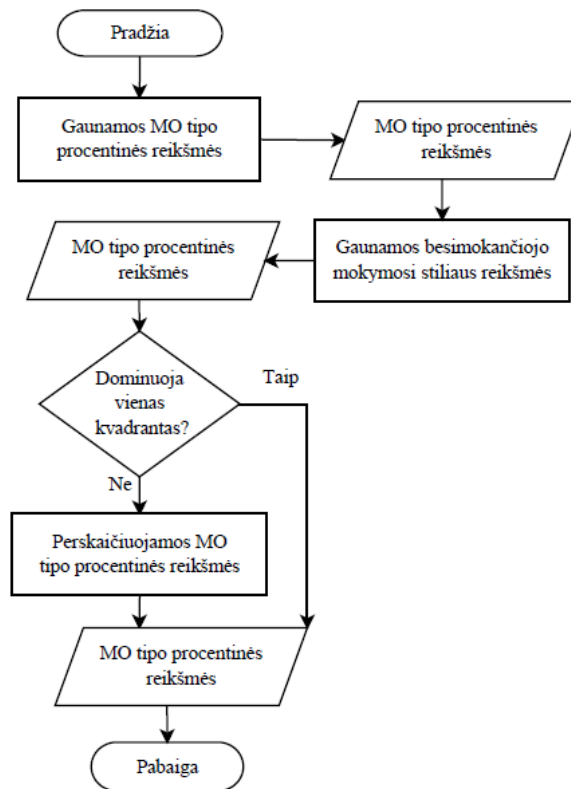


52 pav. MO tipų adaptavimo besimokančiajam modelis

17 lentelė. Mokymosi stiliaus stebėjimo agento aprašas

Laukiamas aplinkos įvykis	Gaunamos besimokančiojo mokymosi stiliaus reikšmės
Programinio agento veiksmai	<ol style="list-style-type: none"> 1. Gaunamos besimokančiojo mokymosi stiliaus reikšmės. 2. Pagal gautas reikšmes nustatomas <i>Herrmann</i>'o mokymosi stiliaus kvadranto dominavimas. 3. – jei dominuoja vienas <i>Herrmann</i>'o mokymosi stilių kvadrantas, iš DB_{MT} gaunamos MO tipų procentinės reikšmės. – jei būdingos kelių <i>Herrmann</i>'o mokymosi stilių kvadrantų savybės, perskaičiuojamos MO tipų procentinės reikšmės – skaičiuojamas atitikimo mokymosi stiliui procentinių verčių svertinis vidurkis.

Mokymosi stiliaus stebėjimo programinio agento algoritmo schema pateikta 53 paveiksle.



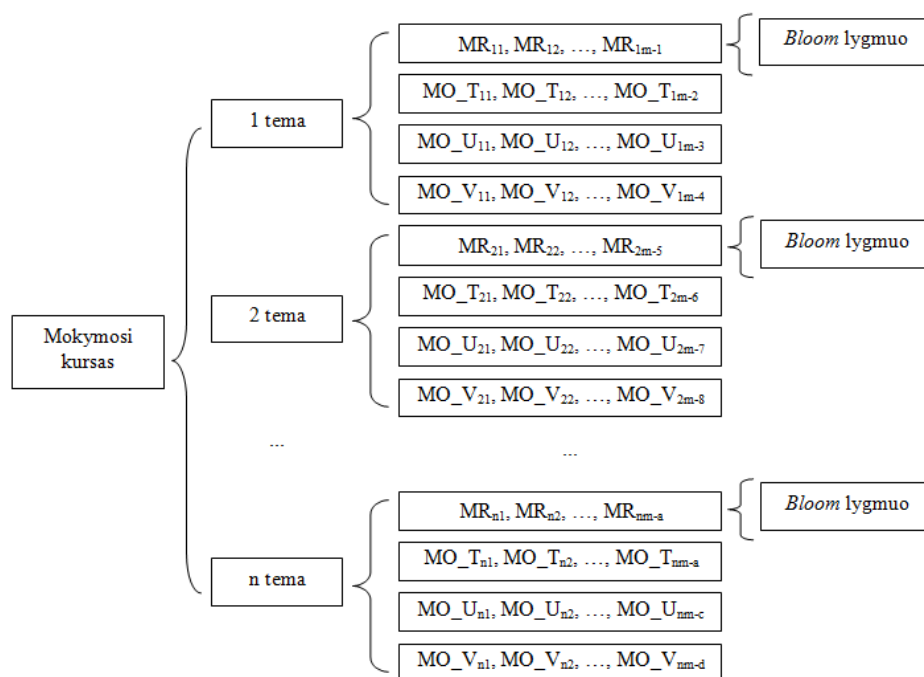
53 pav. Mokymosi stiliaus stebėjimo programinio agento veikimo algoritmo schema

MO tipų procentinės vertės perskaičiuojamos atsižvelgiant į besimokančiojo mokymosi stiliaus nustatymo metu gautas skaitines vertes (testo rezultatus).

3.4. Programavimo mokymosi kurso kūrimas (atvaizdavimas)

Siekiant išskirti pagrindinius reikiamus išmokti programavimo

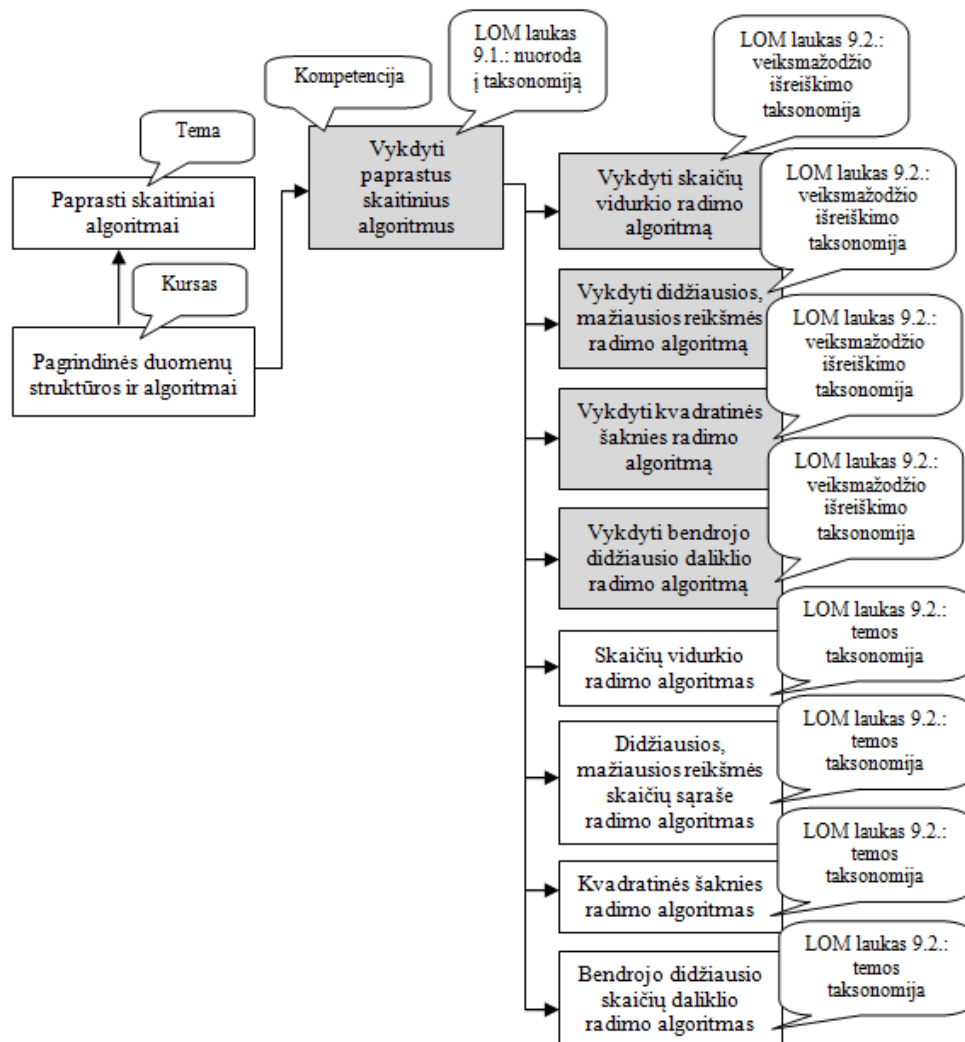
konceptus ir apibrėžti mokomojo dalyko programą (dalyko temas, žinias, supratimą, gebėjimus, kuriuos tikimasi, kad besimokantysis įgis) remtasi ACM parengtomis rekomendacijomis informatikos mokomųjų dalykų programoms. Čia yra įvardintos pagrindinės mokomojo dalyko programos temos, kurios turėtų būti mokomos bei šias temas atitinkančios dalykinės kompetencijos. Kiekvieną kompetenciją galima susieti su 6-iais *Bloom*‘o taksonomijos lygmenimis (Dagienė, Urbonienė, 2010). Taigi, PMK gali būti kuriamas atsižvelgiant į temas ir jų mokymosi metu siekiamas ugdyti kompetencijas, kurios atitinka vieną iš *Bloom*‘o taksonomijos lygmenų (žinios, supratimas, taikymas, analizė, sintezė ir vertinimas), bei joms priskiriant 3 rūšių MO – turinio, užduoties ir vertinimo (54 pav.).



54 pav. Mokymosi kurso elementų išdėstymo struktūrinė schema

Atsižvelgiant į tai, jog PMK kūrimas remiantis mokomojo dalyko programa yra varginantis ir pakankami daug laiko reikalaujantis darbas, yra skurta automatizuotų mokymosi kurso programos žymėjimo metodų (angl. *Curriculum mapping*), kurie leidžia išvengti kurso kūrimo rankiniu būdu. Kurilovo pasiūlytas mokymosi kurso žymėjimo metodas (Kurilovas, 2008) leidžia mokymosi kursą kurti susiejant jo temas, ugdomas dalykines kompetencijas ir atitinkamus MO bei tą informaciją įrašant MO metaduomenų

laukuose 9.1 Paskirtis (*Purpose*) ir 9.2 Kompetencijos ir temos (*Taxom path*). Pritaikius anksčiau minėtąjį metodą, mokomojo dalyko „Algoritmai ir sudėtingumas“ temos „Paprasti skaitiniai algoritmai“ (CS2013) struktūros vaizdas pateiktas 55 paveiksle.

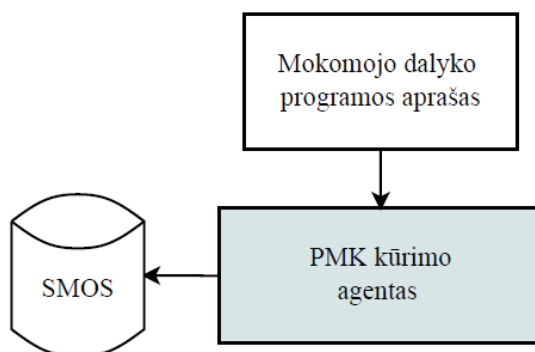


55 pav. Temos „Paprasti skaitiniai algoritmai“ dekomponavimas (adaptuota pagal Kurilovą, 2008)

Kadangi APMS kompetencijos siejamos su *Bloom*‘o taksonomijos lygmenimis, ši informacija taip pat turi būti saugoma kaip MO metaduomenų lauko įrašas (įrašoma lauke 9.2.).

Taigi turint mokomojo dalyko programos aprašą, kuriame nurodomos temos, aktyviais veiksmoždžiais aprašytos dalykinės kompetencijos (žr. 2.1.5. skyrelį) ir nurodytas siekiamas *Bloom* ‘o taksonomijos lygmuo, PMK

kūrimo programinis agentas nustato MO sąsajas su atitinkamomis temomis, dalykinėmis kompetencijomis ir *Bloom*’o taksonomijos lygmeniu. PMK kūrimo modelis pateiktas 56 paveiksle, o PMK kūrimo programinio agento aprašas pateiktas 18 lentelėje.

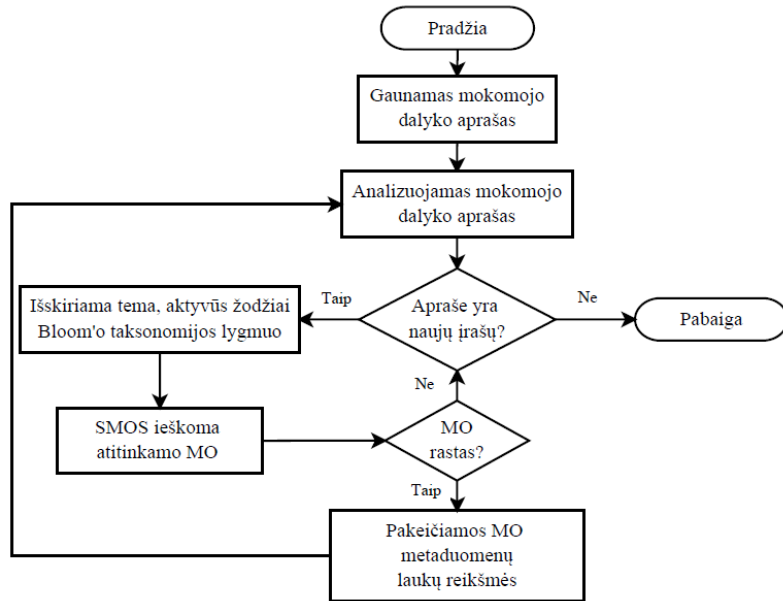


56 pav. PMK kūrimo modelis

18 lentelė. PMK kūrimo agento aprašas

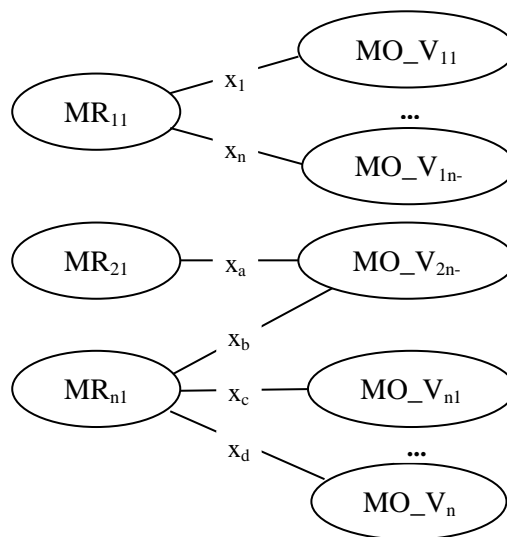
Laukiamas aplinkos įvykis	Gaunamas mokomojo dalyko programos aprašas, kur dalykinės kompetencijos aprašomos panaudojant aktyvius veiksmažodžius
Programinio agento veiksmai	<ol style="list-style-type: none"> 1. Gaunamas mokomojo dalyko programos aprašas. 2. Pagal gautą aprašą išskiriamos temos ir jas atitinkančios dalykinės kompetencijos. 3. Informacija apie atitikimą temai, dalykinei kompetencijai ir <i>Bloom</i>’o taksonomijos lygmeniui išsaugoma MO metaduomenų laukuose 9.1., 9.2. – 9.2.2.1 ir 9.2.2.2.

PMK kūrimo programinio agento algoritmo schema pateikta 57 paveiksle.



57 pav. PMK kūrimo agento veikimo algoritmo schema

Vertinamosios užduotys (vertinimo rūšies MO) siejamos su dalykinėmis kompetencijomis (mokymosi rezultatais) atitinkamais ryšiais su numatytais svoriais x (58 pav., čia $0 < x_x \leq 1$). Svoriai yra priskiriami atsižvelgiant į tai, kurį *Bloom*'o taksonomijos lygmenį atitinka vertinamoji užduotis – aukštesnio lygio užduotis turi didesnę svorį.

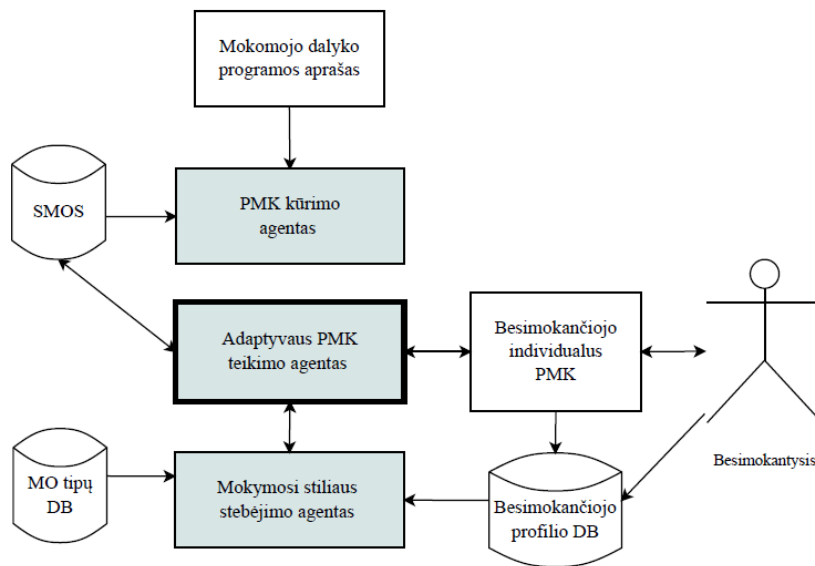


58 pav. Kompetencijų ir vertinamųjų užduočių sąryšis

Sukurtas PMK toliau gali būti teikiamas naudoti besimokančiajam. Kitame skyrelyje yra aprašomas mokomojo dalyko programos elementų (MO) teikimo besimokančiajam mechanizmas.

3.5. Adaptyvaus PMK teikimas besimokančiajam

Sukurtas PMK pateikiamas besimokančiajam atsižvelgiant į jo mokymosi stilių pagal *Herrmann*'o mokymosi stilių klasifikaciją. APMS projektuojama kaip daugiaagentinė sistema, kurioje be anksčiau minėtųjų veikia dar vienas agentas – adaptyvaus PMK teikimo programinis agentas, padedantis pateikti besimokančiajam jo individualias savybes (mokymosi stilių pagal *Herrmann*'o klasifikaciją) labiausiai atitinkančius PMK kūrimo metu susietus MO. MO išdėstomi juos rikiuojant pagal didžiausią tinkamumą mokymosi stiliui, naujumą (pirmiau pateikiami dar besimokančiojo nenagrinėti) ir panaudojamumo skaičių (pirmiau pateikiami tie, kurie buvo daugiau kartų naudojami kitų). Adaptyvaus PMK teikimo modelio schema matoma 59 paveiksle, programinio agento aprašymas pateikiamas 19 lentelėje.

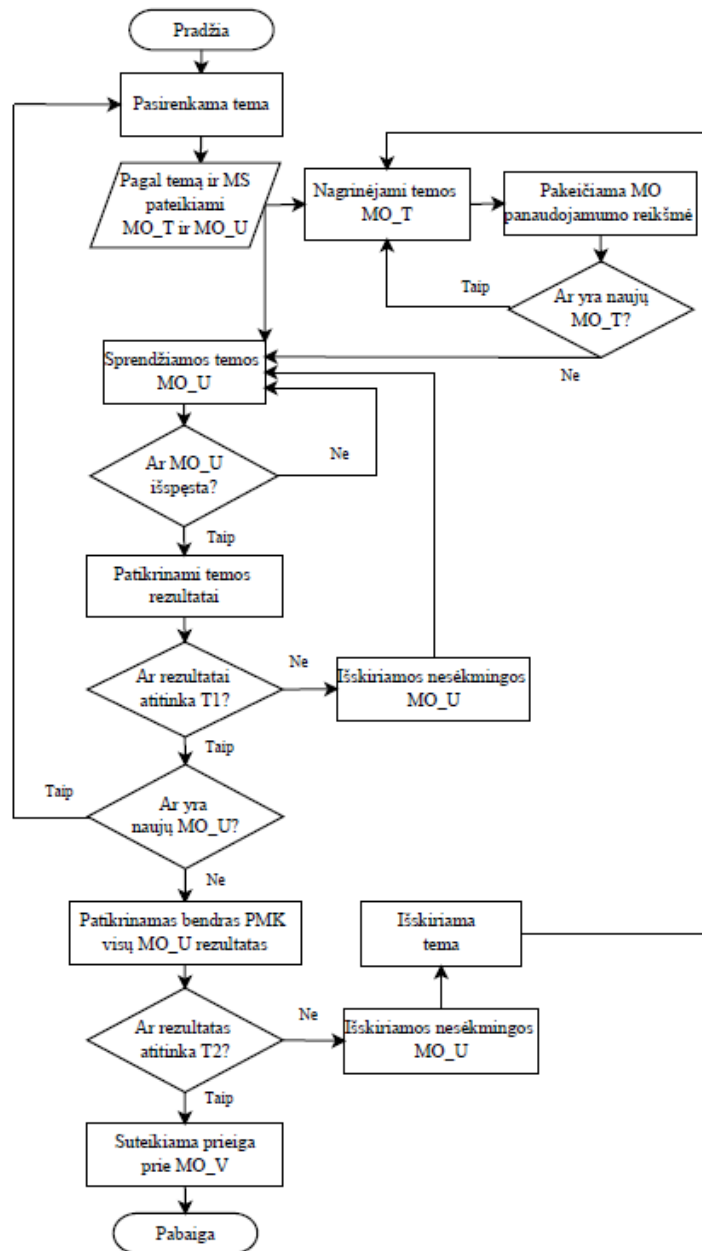


59 pav. Adaptyvaus PMK teikimo modelio schema

19 lentelė. **Adaptyvaus PMK teikimo agento aprašas**

Laukiamas aplinkos įvykis	Besimokantysis prisijungia prie PMK aplinkos
Programinio agento veiksmas	<ol style="list-style-type: none"> 1. Pateikiamas mokymosi turinys – mokomosios temos. 2. Pasirinkus temą, atsižvelgiant į besimokančiojo mokymosi stilų parenkami ir besimokančiajam pateikiami atitinkami MO – pirmiausiai pateikiami pagal ekspertinio vertinimo rezultatus labiausiai tinkančių tipų, nauji ir dažniausiai naudojami MO. Atskirai pateikiami turinio ir užduoties MO (MO_T ir MO_U). 3. Panaudojus atitinkamą MO (jį atvėrus ir išlaikius atvertą daugiau kaip 1 min.), tai užfiksuojama pakeičiant MO panaudojamumo žymės skaitinę reikšmę – prie ankstesnės reikšmės pridėdant 1. 4. Besimokančiojo atliekami veiksmai (naudoti MO) perduodami besimokančiojo stiliaus stebėjimo agentui, kad šis įrašytų informaciją apie naudotus MO į besimokančiojo profilio DB. 5. Atliekant numatytas mokymosi užduotis (nagrinėjant MO_U): <ul style="list-style-type: none"> - pasiekus atitinkamą taisyklėmis apibrėžtą minimalų temos užduočių atlikimo lygį, leidžiama atlikti kitos temos mokymosi užduotis; - nepasiekus atitinkamo taisyklėmis apibrėžto minimalaus temos užduočių atlikimo lygio, besimokantysis grąžinamas nagrinėti tos temos turinio MO (pateikiami tą temą atitinkantys MO su tema susieti PMK kūrimo metu ir suderinami su besimokančiojo mokymosi stiliu) ir iš naujo atlikti mokymosi užduotis (pirmiausiai pateikiamos dar nenagrinėtos mokymosi užduotys, jei tokių yra). 6. Sėkmingai atlikus visas numatytas mokymosi užduotis ir: <ul style="list-style-type: none"> - pasiekus minimalų lygį ($\geq 45\%$), besimokantysis gauna prieigą prie vertinimo MO; - nepasiekus taisyklėmis apibrėžto lygio, t. y. kai $> 45\%$, besimokantysis grąžinamas į tą temą, kur buvo pasiektas prasčiausias mokomųjų užduočių atlikimo lygis.

Besimokančiajam yra siūlomi PMK temą, ugdomą dalykinę kompetenciją atitinkantys MO. Pasirinkus temą MO pateikiami sąrašu, kur pirmiausiai pateikiami besimokančiojo mokymosi stiliui labiausiai tinkančių tipų MO. Jei SMOS yra keli tą temą atitinkantys MO, yra siūloma peržiūrėti juos visus. Adaptyvaus PMK teikimo agento veikimo algoritmo schema pateikiama 60 paveiksle.



60 pav. Adaptyvaus PMK teikimo agento veikimo algoritmo schema

Viena iš svarbių sistemos užduočių padėti įvertinti besimokančiojo pasiektų dalykinių kompetencijų lygį ir esant reikalui, jei mokymosi rezultatas yra nepatenkinamas (nepasiekiamas taisyklėmis apibrėžtas lygis), atsižvelgiant į ankstesnę mokymosi patirtį (peržiūrėtą/neperžiūrėtą mokymosi medžiagą ir sėkmingai/nesėkmingai įveiktą užduotį), pateikti besimokančiajam sąrašą jo mokymosi stilių atitinkančių mokomųjų objektų. Tuo pačiu sudaroma galimybė pačiam besimokančiajam rinktis sąrašo elementus, neatsižvelgiant į jų išdėstymo eiliškumą.

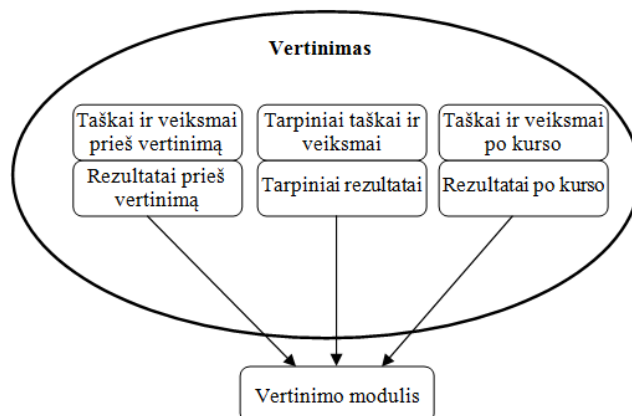
Mokymosi eigoje besimokantysis spendžia mokymosi užduotis (naudojami užduoties MO – MO_U), kurių rezultatai fiksuojami ir šie įrašai naudojami tolimesnei mokymosi eigai generuoti. Mokymosi užduočių rezultatai naudojami tik neformaliai PMK vertinimui.

Jei mokymosi taisyklės netenkinamos (mokymosi rezultatų lygis yra < 45 %), besimokantysis privalo grįžti ir pakartoti reikiamą medžiagą bei jam, pagal jo mokymosi stilių, yra pateikiami tinkamiausi MO, su tema susieti PMK kūrimo metu. Pateikiamas MO sąrašas, kur pirmiausiai pateikiami labiausiai besimokančiojo mokymosi stilių atitinkantys (atsižvelgiant į MO tipų ekspertinio vertinimo procentines vertes) dar nenagrinėti ir daugiausiai panaudoti (atsižvelgiant į panaudojamumo skaitinę vertę) MO.

Sėkmingai įveikus mokymosi užduotis, t. y. jei pasiekiami taisyklėmis aprašyti rezultatai, yra atliekamas mokymosi rezultatų įvertinimas. Tai aptariama kitame skyrelyje.

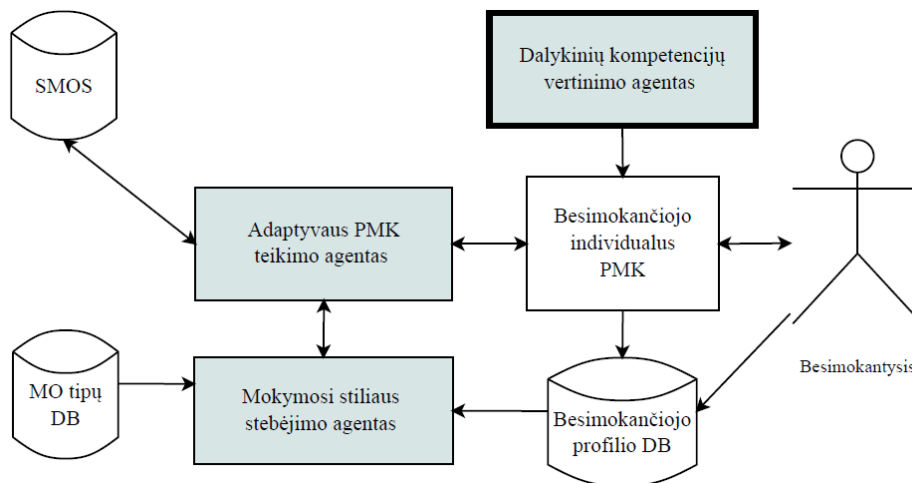
3.6. Mokymosi rezultatų vertinimas

Mokymosi rezultatams įvertinti sprendžiamos juos atitinkančios vertinamosios užduotys (naudojami vertinimo MO – MO_V). Užduočių išsprendžiamumo lygis taip pat aprašomas taisyklėmis. Vertinamųjų užduočių rezultatai naudojami tik formaliam PMK vertinimui ir pagal gautus rezultatus nurodomas dalykinių kompetencijų paiekimo lygis. Apibendrintas mokymosi rezultatų vertinimo modelis pateiktas 61 paveiksle.



61 pav. Mokymosi rezultatų vertinimo modelis

Daugiaagentinėje sistemoje rezultatams vertinti veikia dalykinių kompetencijų vertinimo programinis agentas, stebintis vertinimo užduočių rezultatus ir atitinkamai pateikia kompetencijų pasiekimo lygio informaciją. Dalykinių kompetencijų vertinimo modelio schema pateikta 62 paveiksle, o kompetencijų vertinimo agento aprašas pateiktas 20 lentelėje.

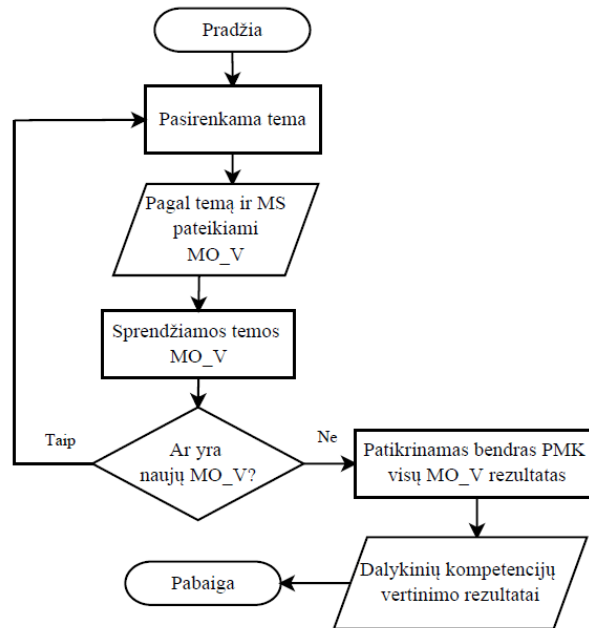


62 pav. **Dalykinių kompetencijų vertinimo modelio schema**

20 lentelė. **Dalykinių kompetencijų vertinimo agento aprašas**

Laukiamas aplinkos įvykis	Besimokantysis pradeda atlikti vertinimo užduotis
Programinio agento veiksmai	<ol style="list-style-type: none"> 1. Pateikiamas mokymosi turinys – mokomosios temos, mokymosi ir vertinimo užduotys. 2. Pasirinkus temą, atsižvelgiant į besimokančiojo mokymosi stilių parenkamos ir besimokančiajam pateikiamos vertinimo užduotys (atitinkami MO_V) – pirmiausiai pateikiami pagal ekspertinio vertinimo rezultatus labiausiai tinkančių tipų, nauji ir dažniausiai naudojami MO. 3. Panaudojus atitinkamą vertinimo užduoties MO (jį atvėrus ir išlaikius atvertą daugiau kaip 1 min.), tai užfiksuojama pakeičiant MO panaudojamumo žymės skaitinę reikšmę – prie ankstesnės reikšmės pridedant 1. 4. Besimokančiojo atliekami veiksmai (naudoti MO) perduodami besimokančiojo stiliaus stebėjimo agentui, kad šis įrašytų informaciją apie naudotus MO į besimokančiojo profilio DB. 5. Atlikus rezultatų vertinimo užduotis mokymasis gali būti baigiamas, tačiau besimokančiajam paliekama galimybė pakartoti norimas temas peržiūrint jas atitinkančius MO. Taip pat besimokančiajam pateikiamas mokomojo dalyko programos dalykinių kompetencijų pasiekimo lygis: aukščiausias kompetencijų lygis, kai MR siekia 85-100 % (arba 9-10 balai), vidutinis – kai MR siekia 65-84 % (arba 7-8), minimalus būtinas kompetencijų lygis – kai MR siekia 45-64 % (arba 5-6)

PMK dalykinių kompetencijų vertinimo algoritmas pateiktas 63 paveiksle.

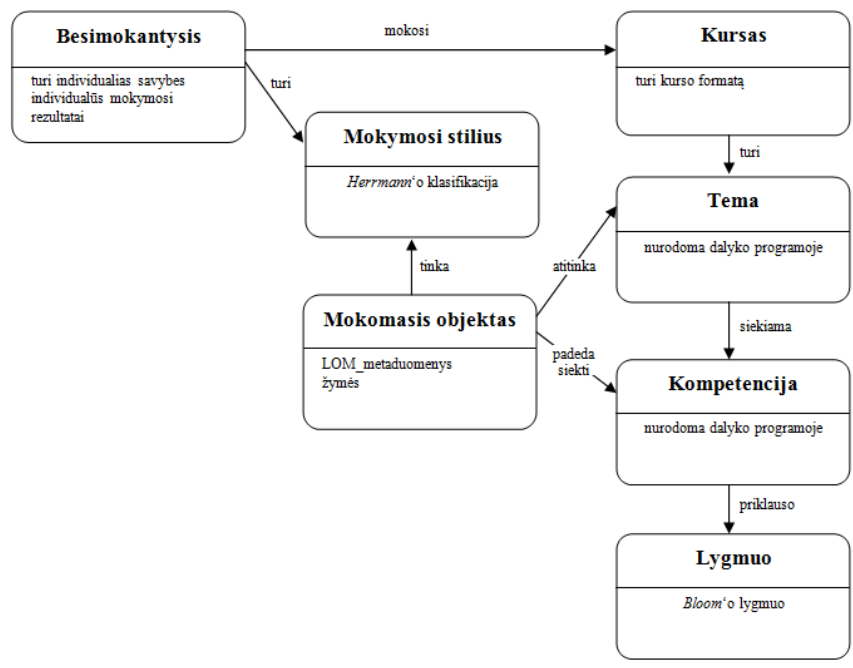


63 pav. Dalykinių kompetencijų vertinimo algoritmo schema

Mokymosi rezultatai yra vertinami atsižvelgiant į tai, kurį *Bloom*’o taksonomijos lygmenį atitinka vertinimo užduotis (t. y. su kuriuo lygmeniu susietas MO_V). Kuo aukštesnis lygmuo, tuo užduoties svoris galutiniam vertinimui yra didesnis.

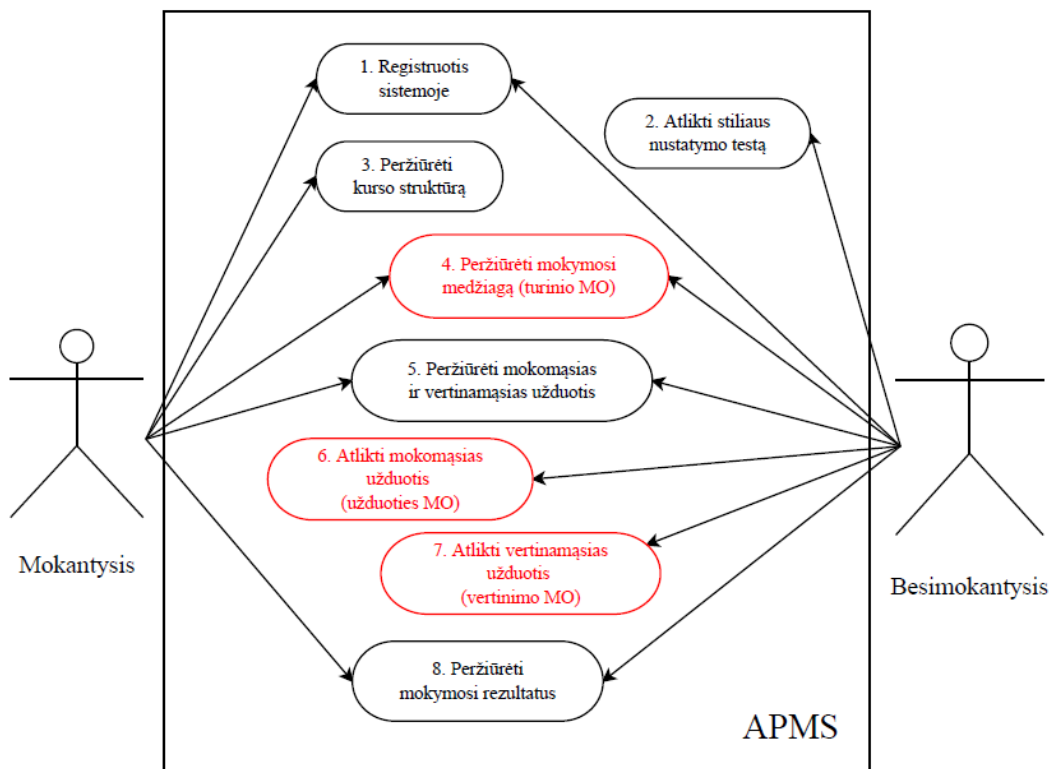
3.7. Adaptyviosios programavimo mokymosi sistemos modelis

APMS esybių ryšių diagrama (64 pav.) nurodo sistemoje egzistuojančias esybes ir jų tarpusavio ryšius. Taigi, besimokantysis turi individualias savybes bei jam būdingą mokymosi stilių, tačiau jis priklausydamas tam tikram lygiui (privalomam arba papildomam) mokosi atitinkamą mokymosi kursą. Kursas savo ruožtu turi temas, kurios leidžia siekti numatytų atitinkamų mokymosi rezultatų. Mokymuisi naudojami MO turi sietis su besimokančiojo mokymosi stiliumi, numatoma kompetencija bei mokomąja tema.



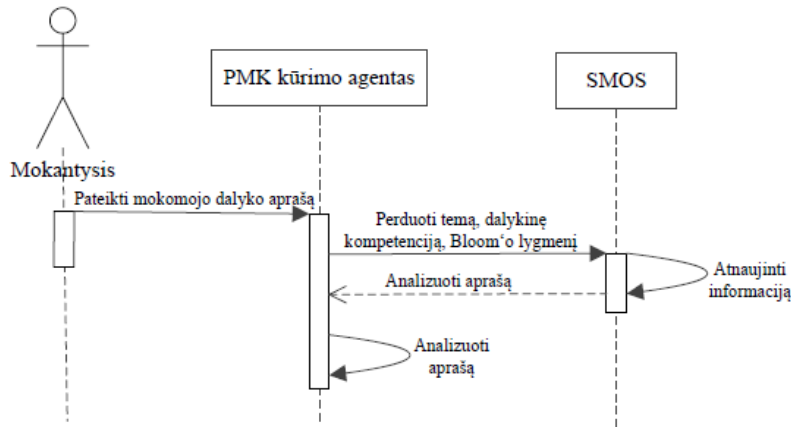
64 pav. APMS esybių ryšių diagrama

APMS yra dviejų tipų vartotojai – mokantysis ir besimokantysis. Šiems vartotojams suteikiamos skirtingos prieigos prie APMS paslaugų teisės. Sistemos panaudos atvejų diagrama pateikiama 65 paveiksle (raudonai žymimos sistemos adaptyvios paslaugos).



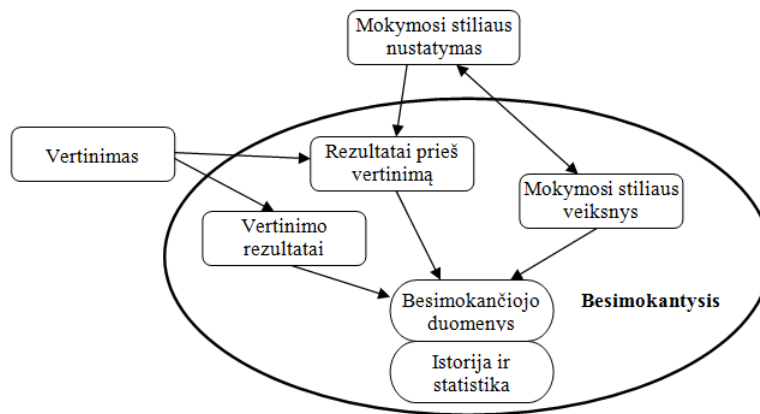
65 pav. APMS panaudos atvejų diagrama

APMS mokymosi kursas generuojamas automatiškai programinių agentų. APMS vartotojas-mokantysis turi galimybę stebėti kurso kūrimo rezultatus bei besimokančiųjų mokymosi rezultatus. Kurso kūrimo sekos diagrama pateikta 66 paveiksle.



66 pav. PMK kūrimo sekos diagrama

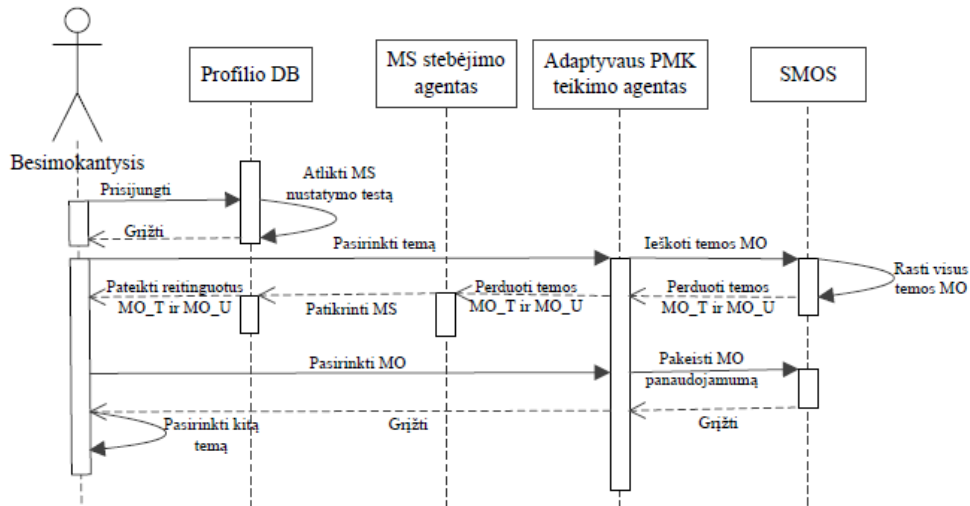
Sugeneruotą PMK APMS dalyko mokymuisi naudoja sistemos vartotojas-besimokantysis (67 pav.).



67 pav. Besimokančiojo modelio schema

Prisijungęs prie sistemos besimokantysis atlieka mokymosi stiliaus nustatymo testą. Testo rezultatai išsaugomi ir naudojami mokomojo dalyko programos elementams – atitinkamiems MO – pateikti. Besimokantysis studijuoja PMK (peržiūri jam pateikiamus MO) atlikdamas numatytas mokymosi užduotis (MO_U). Studijuojant mokomojo dalyko programą besimokančiajam pateikiami jo mokymosi stilių atitinkantys MO (pateikiami

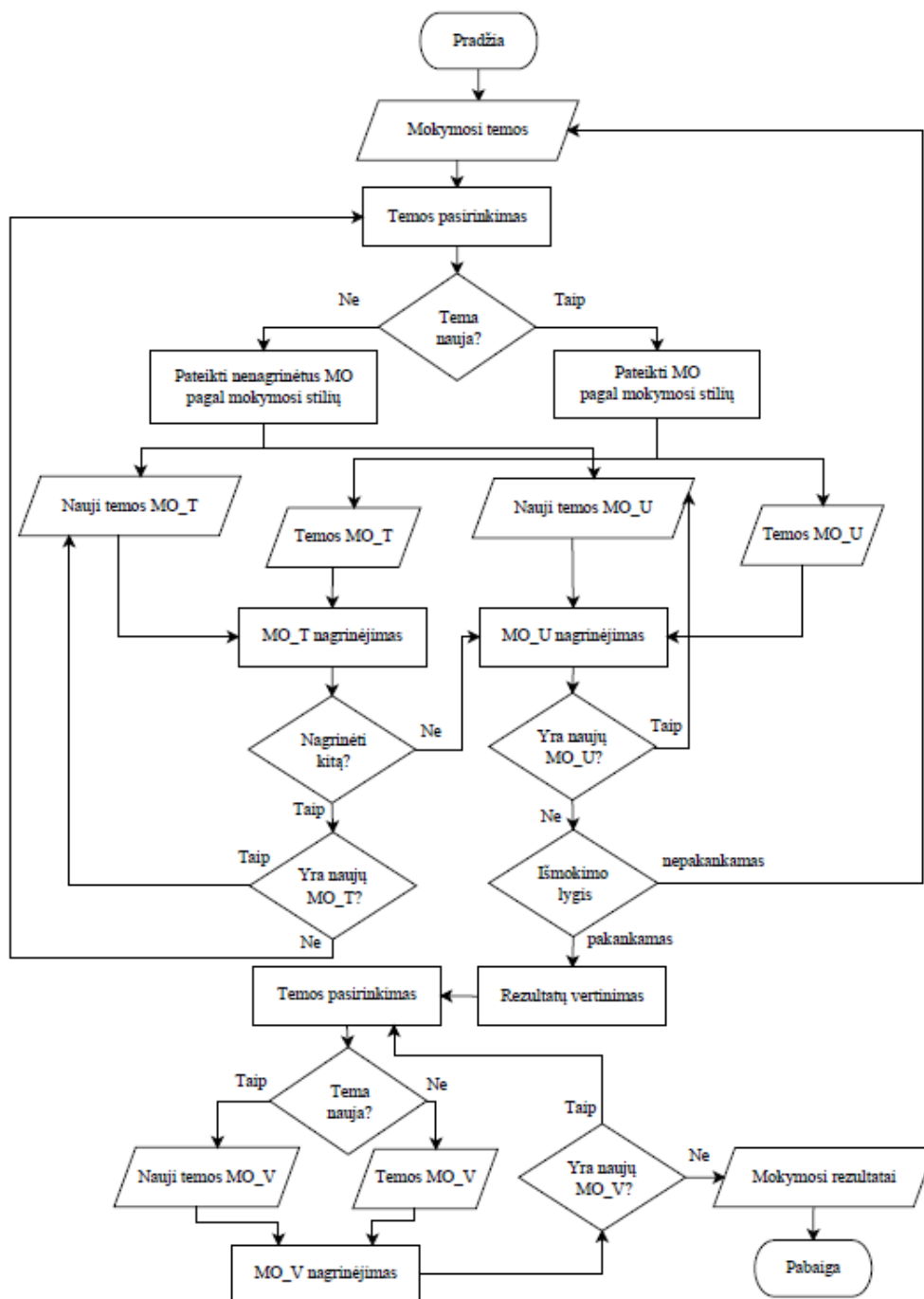
tik turinio ir užduoties rūšies MO – MO_T ir MO_U) bei SMOS fiksuojamas objekto panaudojamumas (peržiūrėjus MO pakeičiama SMOS saugoma jo panaudojamumo žymės reikšmė, kuri vėliau leidžia MO pateikti ankstesniu numeriu, nei kiti alternatyvūs tą temą ir kompetenciją atitinkantys MO). PMK naudojimo sekos diagrama pateikta 68 paveiksle.



68 pav. PMK naudojimo sekos diagrama

APMS naudojimo (besimokančiojo atžvilgiu) algoritmo schema pateikiama 69 paveiksle. Prisijungus prie sistemos yra pateikiamas PMK – mokomojo dalyko temos ir turinio bei užduoties MO. Besimokantysis rinkdamasis temas peržiūri pagal jo mokymosi stilių sąrašą pateiktus MO_T ir sprendžia MO_U. Sistema skaičiuoja mokymosi rezultatus ir pasirinktam MO pakeičia panaudojamumo reikšmę. Besimokantysis privalo išspręsti visų temų mokomąsias užduotis, t. y. panaudoti visų temų MO_U. Užduočių sprendimo rezultatai yra fiksuojami ir tikrinamas bendras temos užduočių išsprendimo lygis. Jei temos užduočių rezultatas yra didesnis už nustatytą minimalią reikšmę, leidžiama spręsti kitos temos užduotis, jei rezultatas yra mažesnis – besimokančiajam pasiūloma peržiūrėti tą temą, dalykinę kompetenciją ir mokymosi stilių atitinkančius MO, t. y. pateikiamas pagal mokymosi stilių, naujumą ir panaudojamumą reitinguotas MO_T sąrašas. Taip pat besimokantysis turi spręsti tos temos užduotis tol, kol temos užduočių sprendimų rezultatas bus didesnis už nustatytą minimalią reikšmę. Išsprendus

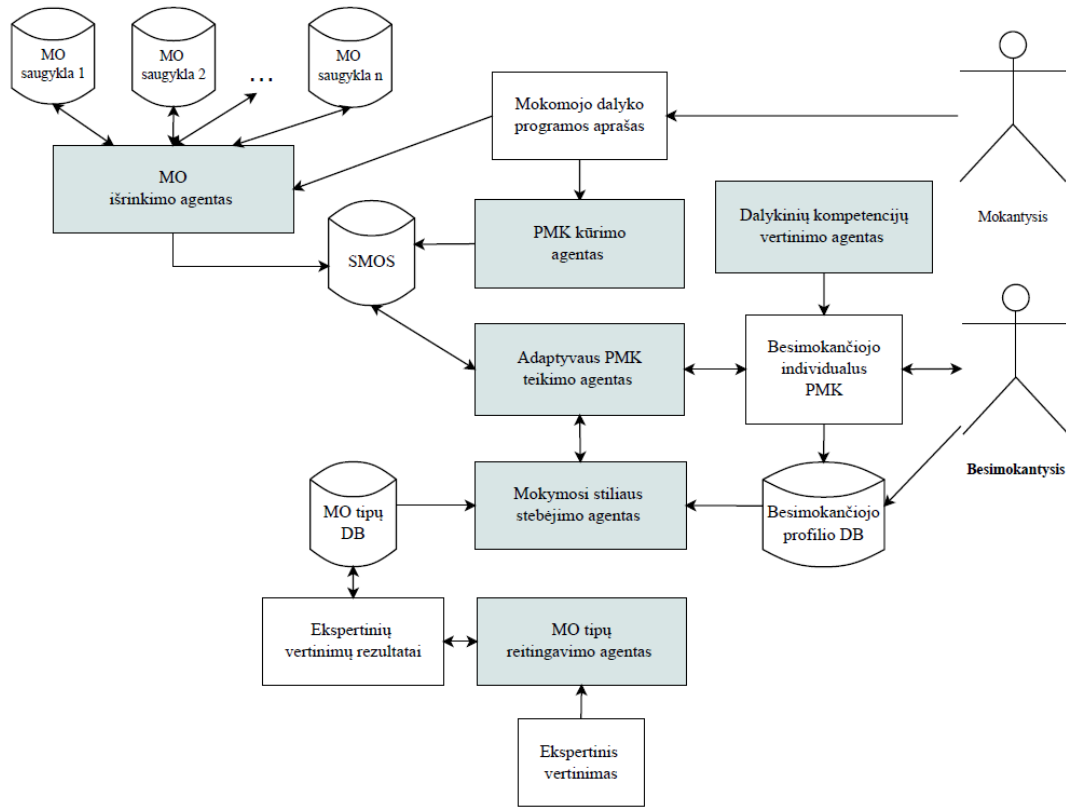
visų temų užduotis reikiamu lygiu (pasiekus nustatytą minimumą) ir pasiekus pakankamą (> 45 %) bendrą visų mokomųjų užduočių sprendimų rezultata, suteikiama prieiga prie vertinamųjų užduočių, t. y. papildomai pateikiamas kiekvienos temos pagal mokymosi stilių reitinguotas MO_V sąrašas.



69 pav. APMS naudojimo schema

APMS modelyje (70 pav.) vaizduojamas MO išrinkimo bei adaptavimo konkrečiam besimokančiajam mechanizmas. MO, panaudojant paieškos

mechanizmus bei atsižvelgiant į individualias besimokančiojo sistemoje saugomas savybes (mokymosi stilių, mokymosi patirtį ir pan.), išrenkami iš SMOS bei per sąsajos komponentą pateikiami besimokančiajam.



70 pav. APMS modelio schema

APMS buvo suprojektuota kaip daugiaagentinė sistema, kurioje veikia 6 programiniai agentai, leidžiantys sukurti adaptyvų PMK, kuris mokomojo dalyko programos elementus (MO) adaptuoja atsižvelgiant į besimokančiojo mokymosi stilių pagal *Herrmann*'o mokymosi stilių klasifikaciją. MO metaduomenys ir žymės saugomi SMOS. Besimokančiojo mokymosi stilius nustatomas naudojant 120 klausimų testą ir testo rezultatai saugomi besimokančiojo profilyje.

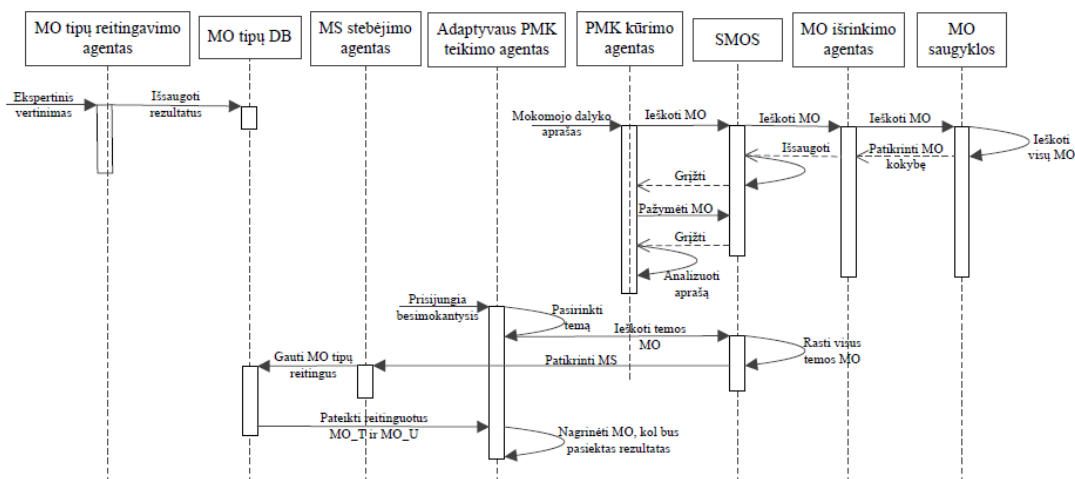
APMS veikia šie agentai:

1. *MO tipų reitingavimo agentas* – ekspertinio vertinimo būdu programavimo mokymui atrinkti MO tipai reitinguojami nustatant tinkamumo besimokančiojo mokymosi stiliui pagal *Herrmann*'o mokymosi stilių klasifikaciją procentines vertes.

2. *MO išrinkimo agentas* – panaudojant *Harvesting* metodą (2.6.4. skyrelis) programavimo mokomojo dalyko programą atitinkantys MO išrenkami iš MO saugyklų, leidžiančių saugoti vartotojų vertinimo žvaigždutėmis ir parsisiuntimų skaičiaus žymes. Prieš parsiuočiant MO į SMOS yra įvertinama MO kokybė – palyginamos vartotojų vertinimo žvaigždutėmis (turi būti > 2,5) ir parsisiuntimo skaičiaus (turi būti > 3) žymės. Kokybiškų programavimo MO metaduomenys išsaugomi SMOS.
3. *PMK kūrimo agentas* – analizuojant pateiktą programavimo mokymosi programą atliekamas programavimo MO susiejimas su programoje siūlomomis temomis, dalykinėmis kompetencijomis ir *Bloom*’o taksonomijos lygmenimis.
4. *Adaptyvaus PMK teikimo agentas* – padedantis besimokančiajam mokytiis tikslingai nukreipiant pakartoti nesėkmingai įsisavintas temas – jei temos užduotis atliekama netenkinant minimalaus apibrėžto lygio, besimokantysis gražinamas pakartoti temą – jam pateikiamas sąrašas MO, besimokančiojo mokymosi stilių labiausiai atitinkančius MO pateikiant pirmiausiai. Pradžioje besimokantysis turi prieigą tik prie turinio ir užduoties rūšies MO (MO_T ir MO_U). Vertinimo MO (MO_V) tampa prieinami tik tuomet, kai besimokantysis atlieka visas užduotis (panaudoja visus MO_U) bent minimaliu apibrėžtu lygiu ir bendras užduočių vertinimas yra > 45 %. Teikiant kursą besimokančiajam papildomai veikia *Mokymosi stiliaus stebėjimo agentas*, kuris perskaičiuoja MO tipų procentines vertes atsižvelgiant į besimokančiojo atitikimo *Herrmann*’o mokymosi stiliui išreikštumo reikšmes ir priklausomai nuo gautų reikšmių atitinkamai pateikia MO besimokančiajam.
5. *Kompetencijų vertinimo agentas* – leidžiantis įvertinti dalykinių kompetencijų pasiekimo lygį. Besimokantysis gauna prieigą prie vertinimo rūšies MO. Atlikus vertinamąsias užduotis (panaudojus vertinimo rūšies MO – MO_V) nustatomas kompetencijų lygis – aukščiausias kompetencijų lygis yra tada, kai mokymosi rezultatai (MR) sieks 85-100 % (arba atitinkamai 9-10 dešimtbalėje vertinimo sistemoje), vidutinis kompetencijų

lygis – kai MR sieks 65-84 % (arba 7-8), minimalus būtinas kompetencijų lygis – kai MR sieks 45-64 % (arba 5-6).

71 paveiksle pateikiama APMS veikiančių agentų tarpusavio sąsajos ir veikimo seka.



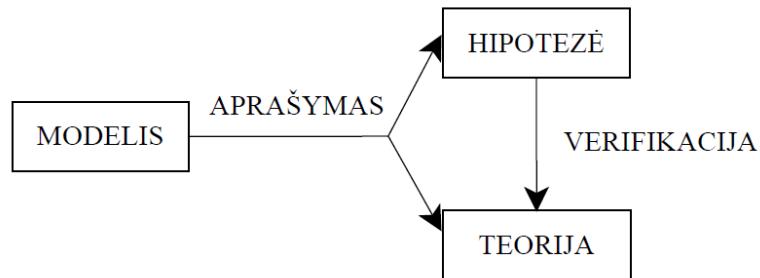
71 pav. APMS agentų veikimo sekos diagrama

3.8. Skyriaus išvados

1. Sudarytas adaptyviosios programavimo mokymosi sistemos, naudojančios intelektualias technologijas – programinius agentus, konceptualus modelis leidžia automatizuotai kurti mokomojo dalyko kursą ir vėliau adaptuoti mokymosi medžiagą prie besimokančiojo individualių savybių, atsižvelgiant į jo mokymosi stilių pagal *Herrmann*‘o mokymosi stilių klasifikaciją.
2. Modelio kūrimui naudotas ekspertinis mokomųjų objektų tipų kokybės vertinimas leidžia užtikrinti mokomojo kurso medžiagos kokybę.
3. Mokomojo dalyko kurso automatizuotas kūrimas (žymėjimas) užtikrina mažesnę klaidų, nes yra eliminuojami kurso kūrimą įtakojantys žmogiškieji faktoriai.

4. MODELIO VERTINIMAS

Sukūrus modelį pradžioje jis laikomas hipotetiniu, t. y. nepatvirtintu. Jei šio modelio elgsena atitinka modeliuojamą reiškinį, tokį modelį galima laikyti reiškinį aprašančia teorija (72 pav.). Taigi bet kuri hipotezė turi būti testuojama arba verifikuojama.



72 pav. Modelio verifikavimo schema

Atlikus paiešką Lietuvos skaitmeninių mokymo priemonių aprašų saugykloje (<http://lom.emokykla.lt>), kur centralizuotai saugomi sukurti MO metaduomenys, įvedant įvairias užklausas programavimo mokymos tematika rasta labai nedaug MO, kurie galėtų būti naudojami APMS (21 lentelė).

21 lentelė. MO paieškos saugykloje rezultatai

Naudota užklausa	Rasta iš viso MO	APMS tinkami naudoti MO	APMS netinkami naudoti MO
<i>programavimas</i>	25	3 enciklopediniai šaltiniai 1 demonstracija 3 praktinės užduotys	14 pamokų planų, 1 mokymosi svetainė – nuoroda į FPS sistemos svetainę 1 modulis, skirtas mokyti 1-2, 3-4 klasių mokinius dirbti Užbutas miškas mokymosi priemone 1 modulis skirtas gabiems 9-10, 11-12 klasių mokiniams tobulinti informacinių technologijų naudojimo įgūdžius
<i>algoritmai arba algortimavimas</i>	5	1 vaizdas 1 paveikslas	3 pamokos
<i>algoritmas</i>	9	1 paveikslas 2 vaizdai 1 praktinė užduotis	1 modulis 4 pamokos
<i>ciklas</i>	9	1 paveikslas 2 vaizdai	6 pamokos

APMS turėtų būti naudojama 15-os tipų (žr. 3.2.1. skyrelį) MO vienai mokomojo dalyko temai, o saugykloje rasta vos 16 tinkamų mokyti programuoti MO. Visi šie MO pritaikyti naudoti informacinių technologijų

pamokose mokykloje. Todėl kurti APMS prototipą ir jį išbandyti su realiais besimokančiais buvo atsisakyta.

Toliau aprašomas APMS modelio ekspertinis vertinimas.

4.1. Sistemos kokybės ekspertinio vertinimo apibrėžimas

APMS modelio kokybei naudojimo atžvilgiu vertinti pasirinktas ekspertinio vertinimo metodas. Ekspertinis vertinimas – apibendrinta ekspertų grupės nuomonė, kurios gavimui pritaikomos specialistų-ekspertų žinios, patirtis ir intuicija. Ekspertu vadinamas specialistas, turintis tam tikros srities žinių ir patirties (lot. *expertus* – patyręs). Kadangi ekspertas yra sukaupęs didelį kiekį racionaliai apdorotos informacijos (turi daug žinių ir patirties, gali remtis intuicija), todėl jis gali būti kokybinės informacijos šaltiniu. Ekspertų grupės nuomonė nedaug skiriasi nuo tikrojo problemos sprendinio.

Ekspertinio vertinimo metodas tinka tais atvejais, kai labai sudėtinga arba praktiškai neįmanoma pritaikyti objektyvius skaičiuojamuosius ar empirinio tyrimo metodus. Tai yra procesų ar reiškinių, kurie tiesiogiai negali būti išmatuoti, kiekybinis įvertinimas. Todėl jis tinkamas įvertinti sistemos modelį pagal apibrėžtus kriterijus.

Šiuo metu dar nėra vieningos moksliskai pagrįstos ekspertinių metodų klasifikacijos ir vieningų taikymo rekomendacijų. Vienais atvejais ekspertai dirba atskirai, kartais net nežinodami, kad jie yra ekspertai. Taip išvengiama žinomų autoritetų nuomonės įtakos. Kitais atvejais ekspertai renkasi kartu ir kartu aptaria problemą, įvertina išsakytus samprotavimus ir atmeta neteisingus. Vienais atvejais ekspertų skaičius griežtai fiksuotas, apskaičiuotas, jis turi tenkinti statistinių suderinamumo metodų prielaidas, kitais atvejais ekspertų skaičius didėja ekspertizės atlikimo metu.

Ekspertinio vertinimo patikimumas priklauso nuo:

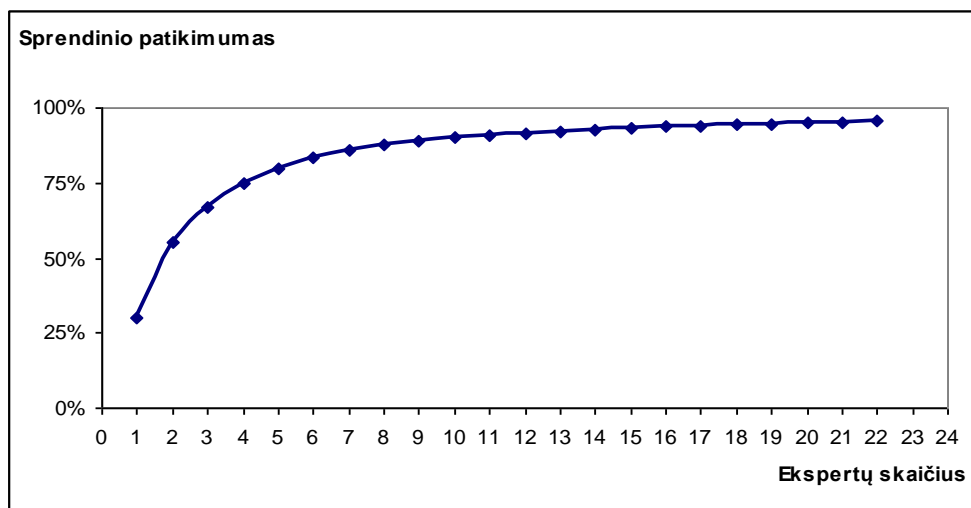
- ekspertų grupės dydžio (ekspertų skaičiaus);
- ekspertų sudėties pagal jų specialybes;
- ekspertų savybių.

Ekspertų kokybė gali būti įvertinta kaip jo objektyvaus ir subjektyvaus statuso apibendrintas rodiklis arba suderinamumo koeficientas:

$$k = 1 - \frac{\eta}{\eta_{\max}},$$

čia: η – vieno eksperto išsakytų prieštaringų vertinimų skaičius, η_{\max} – galimas maksimalus prieštaringų vertinimų skaičius.

Ekspertų skaičius nustatomas vadovaujantis klasikiniėje testų teorijoje suformuluotomis metodologinėmis prielaidomis – agreguotų sprendimų patikimumą ir ekspertų skaičių sieja greitai gėstantis netiesinis ryšys (73 pav.) (Libby, 1978).



73 pav. Ekspertų vertinimų standartinio nuokrypio priklausomybė nuo ekspertų skaičiaus (pagal Libby, 1978)

Iš pateiktos diagramos matyti, kad didžiausias sprendinio patikimumo skirtumas gaunamas esant mažesniai nei 7 ekspertų skaičiui, o esant didesniai skaičiui – patikimumo rezultatas keičiasi nežymiai (apie 10 %). Todėl sukurtą APMS modelį įvertino 7 ekspertai.

4.2. Adaptyviosios programavimo mokymo priemonės modelio kokybės ekspertinis vertinimas

Sukurto APMS modelio vertinimo ekspertams keliami atitinkami reikalavimai. Ekspertas turi turėti:

- informatikos arba informatikos inžinerijos krypties daktaro mokslinį laipsnį;
- ne mažesnę kaip 5 metų programavimo mokymo patirtį;
- supratimą apie MO naudojimą mokymuisi;
- išspausdintą bent vieną mokslinį straipsnį recenzuojamuose mokslo darbų leidiniuose.

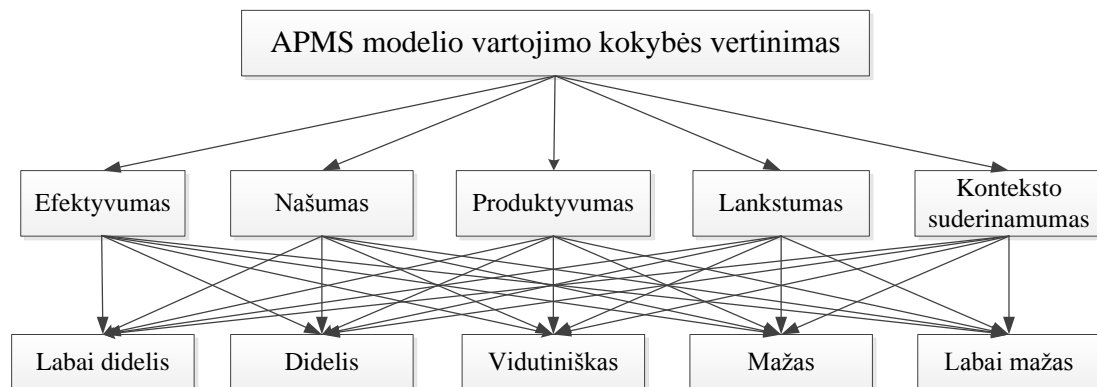
APMS modeliui vertinti pasirinktas rezultatų apdorojimo ekspertinio vertinimo metodas, t. y. atliekama rangavimo (reitingavimo) uždavinio procedūra. Tam apibrėžiama sistemos vertinimo kriterijų aibė bei nustatomos reitinguotos vertinimo reikšmės.

Sistemos modeliui įvertinti buvo parinkti vertinimo kriterijai (22 lentelė) (remiamasi standartu *ISO/IEC FCD 25010* (2009)). Siekiama išsiaiškinti APMS modelio vartojimo kokybę (74 pav.) įvertinant *efektyvumą* – ar besimokančiajam pateikiami tinkami MO, leidžiantys siekti dalyko programoje numatytų dalykinių kompetencijų, *našumą* – ar modelyje siūlomi metodai leidžia taupyti laiką renkantis tinkamus konkrečios užduotims atlikti MO, *produktyvumą* – ar siekiant efektyvumo modelyje siūlomi metodai leidžia vartotojui padidinti MO skaičių, *naudojamumą* – ar modelyje siūlomi metodai yra lankstūs, t. y. leidžiantys adaptuoti mokomojo dalyko medžiagą prie besimokančiojo individualių savybių, ir suderinami su kontekstu, t. y. gali būti naudojami įvairiuose mokymosi kontekstuose.

22 lentelė. **APMS modelio vertinimo kriterijai** (adaptuota pagal *ISO/IEC FCD 25010*)

Eil. Nr.	Charakteristika	Subcharakteristika (kriterijus)	Aprašymas	Vertinimo skalė
1.	Efektyvumas	Efektyvumas	Modelyje siūlomi metodai pateikia tinkamus MO, leidžiančius siekti dalyko programoje numatytų dalykinių kompetencijų	5 – labai didelis 4 – didelis 3 – vidutiniškas 2 – mažas 1 – labai mažas
2.	Našumas/ Veiksmingumas	Našumas	Modelyje siūlomi metodai leidžia taupyti laiką renkantis tinkamus konkrečioms užduotims atlikti MO (mokymosi proceso našumas)	5 – labai didelis 4 – didelis 3 – vidutiniškas 2 – mažas 1 – labai mažas

Eil. Nr.	Charakteristika	Subcharakteristika (kriterijus)	Aprašymas	Vertinimo skalė
3.	Produktyvumas	Produktyvumas	Siekiant efektyvumo modelyje siūlomi metodai leidžia vartotojui padidinti MO skaičių	5 – labai didelis 4 – didelis 3 – vidutiniškas 2 – mažas 1 – labai mažas
4.	Naudojamumas	Lankstumas	Modelyje siūlomi metodai leidžia adaptuoti mokomojo dalyko medžiagą prie besimokančiojo individualių savybių (Herrmann'o mokymosi stiliaus)	5 – labai didelis 4 – didelis 3 – vidutiniškas 2 – mažas 1 – labai mažas
		Konteksto suderinamumas	Modelyje siūlomi metodai gali būti naudojami įvairiuose mokymosi kontekstuose	5 – labai didelis 4 – didelis 3 – vidutiniškas 2 – mažas 1 – labai mažas



74 pav. APMS ekspertinio vertinimo kriterijų ir alternatyvų sąsajos

Apibrėžtų APMS modelio vartojimo kokybės ekspertinio vertinimo kriterijų vertinimo skalės reikšmių žymėjimai pateikti 23 lentelėje.

23 lentelė. APMS modelio vertinimo kriterijų reikšmės

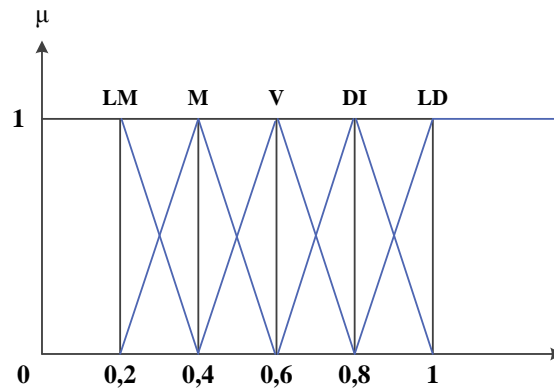
Vertė	Vertinimo kriterijų reikšmės	Žymėjimas
5	Labai didelis	LD
4	Didelis	DI
3	Vidutiniškas	V
2	Mažas	M
1	Labai mažas	LM

Apibrėžus vertinimo kriterijus buvo vykdomas pokalbis su APMS modelio vertinimo ekspertais. Pokalbio metu pristatytas sukurtas APMS modelis ir prašoma parinkti numatytą vertinimo kriterijų reikšmes. Ekspertų vertinimo rezultatai matomi 24 lentelėje.

24 lentelė. APMS ekspertų vertinimo rezultatai

Eil. Nr.	Kriterijus	Ekspertų vertinimai						
		1	2	3	4	5	6	7
1.	Efektyvumas	LD	LD	DI	LD	DI	DI	LD
2.	Našumas	LD	LD	LD	LD	DI	LD	LD
3.	Produktyvumas	LD	LD	DI	DI	V	DI	DI
4.	Lankstumas	LD	LD	LD	LD	DI	LD	DI
5.	Konteksto suderinamumas	LD	LD	LD	LD	DI	LD	DI

Pritaikius daugiakriterinę sprendimų priėmimo metodiką (Skūpienė, 2010) ir panaudojus penkis *Fuzzy* neraiškiuosius skaičius (Labai didelis (LD), Didelis (DI), Vidutiniškas (V), Mažas (M), Labai mažas (LM), 75 pav.) ekspertų vertinimas paverčiamas skaitine verte, kuri ir nurodo kriterijaus išreikštumą.



75 pav. Fuzzy lingvistinių kintamųjų reikišmių sąsajos

Tarkime, $D = \{D_1, D_2, \dots, D_n\}$ yra baigtinė sprendimo priėmėjų (ekspertų) aibė, kur $n \geq 2$, ir $K = \{K_1, K_2, \dots, K_m\}$ yra baigtinė kriterijų aibė. x_{ij} yra sprendimo priėmėjo D_i ($i = 1, 2, \dots, n$) charakteristika kriterijaus K_j ($j = 1, 2, \dots, m$) atžvilgiu. Tarkime $s_j \geq 0$ yra santykinis kriterijaus K_j ($j = 1, 2, \dots, m$) svoris ir $\sum_{j=1}^m s_j = 1$. Tada vieno sprendimo problema naudojant daugiakriterinę sprendimų priėmimo metodiką gali būti vaizduojama matrica:

$$\begin{array}{cccc}
 & K_1 & K_2 & \dots & K_m \\
 & s_1 & s_2 & \dots & s_m \\
 D_1 & x_{11} & x_{12} & \dots & x_{1m} \\
 D_2 & x_{21} & x_{22} & \dots & x_{2m} \\
 \dots & \dots & \dots & \dots & \dots \\
 D_n & x_{n1} & x_{n2} & \dots & x_{nm}
 \end{array}$$

APMS atveju visi sprendimo priėmėjai yra vienodai reikšmingi, jų vertinimai x_{ij} , kur $i = 1, 2, \dots, n$ ir $j = 1, 2, \dots, m$, turi vienodą svorį $d_t=0,14$, kur $t>2$ ir $\sum_{t=1}^n d_t = 1$. Nustatyti APMS modelio vertinimo kriterijai taip pat visi vienodai reikšmingi, todėl jų svoriai s_j , kur $j = 1, 2, \dots, m$, irgi vienodi ir jie yra lygūs $0,2$ ir $\sum_{j=1}^m s_j = 1$.

Kadangi kriterijų metrikos yra subjektyvios, t. y. priklauso nuo sprendimų priėmėjo subjektyvios nuomonės, tai argumentai yra lingvistiniai. Remiantis daugiakriterine sprendimų priėmimo metodika lingvistinės reikšmės yra konvertuojamos į trikampius neraiškiuosius skaičius (25 lentelė). APMS modelio naudojimo kokybės ekspertinio vertinimo analizei naudojami trikampiai neraiškieji skaičiai.

25 lentelė. Lingvistinės skalės konvertavimo į neraiškiuosius skaičius reikšmės

Skalės elementas	Neraiškieji trikampiai skaičiai
labai didelis	(0,8; 1,0; 1,0)
didelis	(0,6; 0,8; 1,0)
vidutiniškas	(0,4; 0,6; 0,8)
mažas	(0,2; 0,4; 0,6)
labai mažas	(0,0; 0,2; 0,2)

Toliau bus naudojamos vidurinės *Fuzzy* neraiškiųjų trikampių skaičių reikšmės: LD – 1,0, DI – 0,8, V – 0,6, M – 0,4, LM – 0,2. Taip visi lingvistiniai įverčiai pakeičiami juos atitinkančiomis skaitinėmis reikšmėmis (26 lentelė).

26 lentelė. Konvertuoti į trikampius neraiškiuosius skaičius ekspertų vertinimo rezultatai

Eil. Nr.	Kriterijus	Svoris	Ekspertų vertinimų skaitinės reikšmės							Vidurkis (v_j)
			1	2	3	4	5	6	7	
1.	Efektyvumas	0,2	1,0	1,0	0,8	1,0	0,8	0,8	1,0	0,9143
2.	Našumas	0,2	1,0	1,0	1,0	1,0	0,8	1,0	1,0	0,9714
3.	Produktyvumas	0,2	1,0	1,0	0,8	0,8	0,6	0,8	0,8	0,8286
4.	Lankstumas	0,2	1,0	1,0	1,0	1,0	0,8	1,0	0,8	0,9429
5.	Konteksto suderinamumas	0,2	1,0	1,0	1,0	1,0	0,8	1,0	0,8	0,9429

Kadangi visų ekspertų vertinimų svoriai yra vienodi (0,14), suskaičiuojamas kiekvieno kriterijaus ekspertų vertinimų aritmetinis vidurkis v_j , kur $j = 1, 2, \dots, m$ (26 lentelė). Taip pat apskaičiuojamas bendras modelio

kokybės vertinimas, kuris apibrėžiamas funkcija $f(x)$ (3 formulė) atsižvelgiant į tai, jog visų kriterijų svoriai s_j yra lygūs 0,2:

$$f(x) = \sum_{j=1}^m s_j \cdot v_j \quad (3)$$

Pritaikius formulę gaunamas rezultatas:

$$f(x) = 0,2 \times 0,9143 + 0,2 \times 0,9714 + 0,2 \times 0,8286 + 0,2 \times 0,9429 + \\ + 0,2 \times 0,9429 = 0,92$$

Ekspertinio vertinimo metu gautas rezultatas (0,92), atsižvelgiant į trikampio neraiškaus skaičiaus konvertavimo į lingvistinius kintamuosius reikšmes (27 lentelė), transformuojamas į lingvistinį kintamąjį ir gaunama, jog bendras modelio vertinimas yra labai geras, t. y. atitinka 92 % absoliučios kokybės (kuri yra lygi 100 %).

27 lentelė. **Trikampių neraiškiųjų skaičių konvertavimo į lingvistinės skalės kintamuosius reikšmės**

Trikampis neraiškūs skaičius	Lingvistinis kintamasis
(0,8; 1,0; 1,0)	labai didelis / labai gera
(0,6; 0,8; 1,0)	didelis / gera
(0,4; 0,6; 0,8)	vidutiniškas / vidutiniška
(0,2; 0,4; 0,6)	mažas / silpna
(0,0; 0,2; 0,2)	labai mažas / bloga

Gauti rezultatai taip pat rodo, kad APMS modelio efektyvumas yra „labai didelis“ (91,4 %), našumas – „labai didelis“ (97,1 %), produktyvumas – „didelis“ (82,9 %), lankstumas ir konteksto suderinamumas – „labai didelis“ (94,3 %).

4.3. Skyriaus išvados

1. Atlikus sukurto adaptyviosios programavimo mokymosi sistemos modelio naudojimo kokybės ekspertinį vertinimą nustatyta, kad pasiūlytas modelis yra „labai geros“ kokybės vertinant pagal šiuos kriterijus:

- *efektyvumą* – ar besimokančiajam pateikiami tinkami MO, leidžiantys siekti dalyko programoje numatytų dalykinių kompetencijų,
- *našumą* – ar modelyje siūlomi metodai leidžia taupyti laiką renkantis tinkamus konkrečios užduotims atlikti MO,

- *produktyvumą* – ar siekiant efektyvumo modelyje siūlomi metodai leidžia vartotojui padidinti MO skaičių,
- *lankstumą* – ar modelyje siūlomi metodai leidžia adaptuoti mokomojo dalyko medžiagą prie besimokančiojo individualių savybių,
- suderinamumą su kontekstu – ar modelyje siūlomi metodai gali būti naudojami įvairiuose mokymosi kontekstuose.

BENDROSIOS IŠVADOS IR REZULTATAI

1. Išanalizavus programavimo mokymo problematiką, programavimo mokymo ir adaptyvias sistemas, besimokančiųjų mokymosi stilių įtaką mokymosi rezultatams, nustatyta, kad:
 - 1.1. Svarbus mokymosi individualizavimas, nes besimokančiajam programuoti yra būdingas savitas informacijos apdorojimo stilius, į kurį atsižvelgiant būtų padidinamas mokymosi efektyvumas. Kaip parodė žvalgomas tyrimas, mokymosi individualizavimo poreikį akcentuoja ir patys besimokantieji.
 - 1.2. Nėra adaptyviųjų programavimo mokymosi sistemų, kurios automatiškai kurtų mokymosi kursą ir jį adaptuotų prie besimokančiojo individualių savybių atsižvelgiant į jo mokymosi stilių. Tikslinga sukurti sistemą, kuri būtų universali ir automatizuotų didžiąją dalį kurso kūrėjo rutininių darbų, atsižvelgdama į pateikiamą mokomojo dalyko apraše pateikiamą informaciją. Naudojant programavimo mokomuosius objektus ir automatinę programų žymėjimą gali būti kuriamas mokomojo dalyko apraše nurodytą turinį atitinkantis mokymosi kursas. Veiksmams ir reakcijoms į aplinkos faktorius automatizuoti sistemoje tikslinga naudoti programinius agentus.
 - 1.3. Nagrinėtuose mokslo darbuose nebuvo išskirti ir aprašyti programavimo mokymuisi tinkami mokomųjų objektų tipai ir jie susieti su mokymosi stiliais pagal *Herrmann*'o mokymosi stilių klasifikaciją. Šis susiejimas užtikrina mokymosi individualizavimą, kai atsižvelgiant į individualias besimokančiojo savybes mokymosi metu adaptuojama mokymosi medžiaga. Mokymosi kurse naudojamų programavimo mokomųjų objektų tipų parinkimą įtakoja analizės metu išskirtos mokymosi veiklos: informacijos šaltinių (sąvokų, terminų, konceptų) studijavimas; pavyzdžių studijavimas; grafinės informacijos studijavimas; vaizdo medžiagos studijavimas; modeliavimas bei

imitavimas; užduočių sprendimas; problemų sprendimas; projektų rengimas.

- 1.4. Projektuojamoje adaptyviojoje programavimo mokymosi sistemoje naudojamam dalykinių kompetencijų vertinimui, atsižvelgiant į *Bloom*'o taksonomijos lygmenis, užtikrinti turi būti atliktas mokomojo dalyko programos (temų ir dalykinių kompetencijų) tikslus susiejimas su mokomaisiais objektais ir *Bloom*'o taksonomijos lygmenimis.
2. Remiantis analizės metu nustatytais programavimo mokymo ypatumais, besimokančiųjų individualiomis savybėmis ir adaptyviųjų bei mokymosi sistemų kūrimo metodais sukurtas adaptyviosios programavimo mokymosi sistemos modelis, kuris:
 - 2.1. Susieja *Herrmann*'o mokymosi stilius su mokomųjų objektų tipais. Parengtas mokomųjų objektų tipų ekspertinio vertinimo metodas leidžia geriau adaptuoti mokomuosius objektus prie besimokančiojo mokymosi stiliaus;
 - 2.2. Suteikia galimybę įvertinti mokomųjų objektų kokybę prieš siūlant juos mokymosi kurse. Mokomųjų objektų kokybinio vertinimo metodo integravimas į adaptyviąją programavimo mokymosi sistemą leidžia užtikrinti mokymosi medžiagos kokybę, nes specializuotoje mokomųjų objektų metaduomenų saugykloje (SMOS) saugomi tik kokybiški mokomieji objektai;
 - 2.3. Leidžia susieti mokomuosius objektus su programavimo mokomojo dalyko temomis, dalykinėmis kompetencijomis ir *Bloom*'o taksonomijos lygmenimis, kas sudaro galimybę įvertinti dalykinių kompetencijų pasiekimo lygį atsižvelgiant į iš anksto apibrėžtas taisykles;
 - 2.4. Leidžia individualizuoti mokymąsi, t. y. rekomenduoti besimokančiajam jo individualias savybes (mokymosi stilių) labiausiai atitinkančią mokymosi kursą, parenkant jo mokymosi stilių atitinkančių mokomųjų objektų tipų eilę.

3. Atlikus adaptyvios programavimo mokymosi sistemos konceptualaus modelio ekspertinį vertinimą nustatyta, kad:
- Modelyje siūlomi metodai leidžia besimokančiajam pateikti tinkamus mokomuosius objektus, tinkančius siekti dalyko programoje numatytų dalykinių kompetencijų, tad galima teigti, kad sistema yra efektyvi;
 - Modelyje siūlomi metodai leidžia taupyti laiką renkantis tinkamus konkrečios užduotims atlikti mokomuosius objektus, todėl sistema yra naši;
 - Modelyje siūlomi metodai leidžia vartotojui padidinti mokomųjų objektų kiekį, todėl metodai yra produktyvūs;
 - Siūlomi metodai leidžia adaptuoti mokomojo dalyko medžiagą prie besimokančiojo individualių savybių, todėl sistema yra lanksti;
 - Modelyje siūlomi metodai gali būti naudojami įvairiuose mokymosi kontekstuose, šitaip užtikrinamas konteksto suderinamumas.
4. Naudojant pasiūlytą adaptyviosios programavimo mokymosi sistemos konceptualų modelį galima sukurti sistemą, adaptuojančią mokomojo dalyko medžiagą prie besimokančiojo ir pasižyminčią efektyvumu, našumu, produktyvumu, naudojamumu (lankstumu ir konteksto suderinamumu).
5. Naudojant pasiūlytą adaptyviosios programavimo mokymosi sistemos konceptualų modelį galima sukurti sistemą, kuri pasižymėtų tokiais parametrais: efektyvumu, našumu, produktyvumu ir naudojamumu (lankstumu ir konteksto suderinamumu). Ši sistema galėtų būti integruojama kaip atskiras modulis į egzistuojančias ir plačiai naudojamas virtualias mokymosi aplinkas, pvz., *Moodle*.
6. Pasiūlytas modelis gali būti taikomas kitose mokymosi srityse atitinkamai išskiriant tos srities mokymuisi labiausiai tinkančius mokomųjų objektų tipus ir mokomojo dalyko programos apraše įvardinant temas, dalykines kompetencijas ir *Bloom*'o taksonomijos lygmenis.

LITERATŪRA

1. Abarius P., Liubinas V. (2014). Elektroninių mokymosi aplinkų ir akademinų informacinių sistemų integravimas. *Electronic learning, information and communication: theory and practice = Elektroninis mokymasis, informacija ir komunikacija: teorija ir praktika*. 2014 (1), ISSN 2335-2493
2. ACM/IEEE-CS, 2008. Computer Science Curriculum 2008: An Interim Revision of CS 2001, ACM/IEEE-CS Joint Interim Review Task Force Report, ACM Press
3. ACM/IEEE-CS, 2012. Strawman Draft: Computer Science Curricula 2013. The Joint Task Force on Computing Curricula Association for Computing Machinery and IEEE-Computer Society
4. Adaptive Learning Systems. Advanced Technology Program (2005). Pasiukiama: <http://www.atp.nist.gov/atp/97wp-lt.htm>
5. Agent Technology Roadmap: Overview and Consultation Report, AgentLink III, December 2004
6. Allinson Ch., Hayes J. (2012). The Cognitive Style Index. Technical Manual and User Guide. Pearson Education Ltd or its affiliate(s), 52 p.
7. Anderson J., Van Weert T. (2002). Information and Communication Technology in Education. A Curriculum for Schools and Programme of Teacher Development. Unesco, 2002
8. Avgeriou P. Adaptive Web-Based Instructional Systems – An Overview. Athens, Greece 2000
9. Baziukaitė D., Vaira Ž., Idzelytė D. (2008). A tool to support self-education in a lifelong learning, Innovative techniques in instruction technology, e-learning, e-assessment and education. New York: Springer, 2008, p. 92-97
10. Bennedsen J, Caspersen M. E., (2004) Programming in Context – A Model-First Approach to CS1. *SIGCSE BULLETIN*, 2004, VOL 36; PART 1, p. 477-481
11. Bennedsen J. (2008). Teaching and Learning Introductory Programming – A Model-Based Approach. *Dissertation for Dr. Philos degree in the Faculty of mathematics and Natural Sciences*, University of Oslo, Norway, 2008
12. Beres I., Magyar T., Turczanyi-Szabo M. (2012). Towards a Personalized, Learning Style Based Blended Learning Model with Individual Assessment. *Informatics in Education*. 2012, Vol. 11, No. 1, p. 1–28
13. Berglund A., Lister R. (2010). Introductory Programming and the Didactic Triangle. *Proceedings of 12th Australasian Computing Education Conference (ACE 2010)*, Brisbane, Australia, January 2010
14. Biggs J. (2003), *Teaching for Quality Learning at University*, 2nd Edn, Open University Press, Berkshire

15. Bishop J. M. (1997). A philosophy of teaching Java as a first teaching language, in *Proceedings of SACLA 1997 Conference*. [Interaktyvus]. [Žiūrėta 2012-03-22]. Prieiga per internetą: <http://www.cs.up.ac.za/cs/jbishop/Homepage/Pubs/Tech-reports/SACLA97.pdf>
16. Blonskis J., Dagienė V. (2003). Programavimo pagrindų mokymo vidurinėje ir aukštojoje mokyklose lyginamoji analizė. *Informacijos mokslai*, ISSN 1392-0561, 26, 2003, p. 23–28
17. Bloom B. S. (ed.) (1956). *Taxonomy of Educational Objectives, the classification of educational goals: Handbook I - Cognitive Domain*, New York: McKay
18. Brézillon P. (2011) Context and explanation in e-collaborative work, chapter in *Handbook of Research on Methods and Techniques for Studying Virtual Communities: Paradigms and Phenomena*. IGI Global ISBN: 1-111111, p. 83-102
19. Bruce K. (2004). Controversy on How to Teach CS1: A Discussion on the SIGCSE-members Mailing List. In *inroads – The SIGCSE Bulletin*, December, 2004
20. Brusilovsky P., Henze N. (2007). Open Corpus Adaptive Educational Hypermedia. In P. Brusilovsky, A. Kobsa & W. Nejdl (Eds.), *The Adaptive Web*: Springer Berlin Heidelberg. Vol. 4321, p. 671-696
21. Brusilovsky P., Maybury M. T. (2002). From adaptive hypermedia to the adaptive web. *Communications of the ACM*, vol. 45, no. 5, 2002, p. 30–33
22. Brusilovsky P., Millan E. (2007) User Models for Adaptive Hypermedia and Adaptive Educational Systems. *Lecture Notes in Computer Science*. Springer Berlin/Heidelberg. Volume 4321
23. Brustoloni J. C. (1991). *Autonomous Agents: Characterization and Requirements*. Carnegie
24. Bunderson V. C. The Validity of the Herrmann Brain Dominance Instrument. [Interaktyvus]. [Žiūrėta 2013-12-15]. Prieiga per internetą: <<http://www.greenleafconsulting.com/HBDI%20Validation.pdf>>
25. Caspersen M. E. (2007). *Educating Novices in The Skills of Programming. PhD Dissertation*, Department of Computer Science University of Aarhus, Denmark, February 2007
26. Churchill D. (2007) Towards a Useful Classification of Learning Objects. *Educational Technology Research and Development*, Volume 55, Number 5, p. 479-497
27. Clark D. (2013). Performance Analysis in Instructional System Design. [Interaktyvus] [Žiūrėta 2013-12-19]. Prieiga per internetą: http://www.nwlink.com/~donclark/hrd/isd/analyze_system.html
Communications of the ACM, 38, 11, p. 108-114
28. Conlan O., Lewis D., Higel S., O’ Sullivan D., Wade V. (2003). Applying Adaptive Hypermedia Techniques to Semantic Web Service Composition. *Workshop on Adaptive Hypermedia, WWW2003 The Twelfth International World Wide Web Conference*, Budapest, Hungary

29. Dagienė V., Kurilovas E. (2008). *Informacinės technologijos švietime: patirtis ir analizė: Monografija*. Vilnius: Matematikos ir informatikos institutas. 216 p.
30. Dagienė V., Urbonienė J. (2010). Programavimo mokymasis: lyginamoji kalbos ir aplinkos analizė. *Informacijos mokslai*. ISSN 1392-0561, VŠĮ Vilniaus universiteto leidykla, 2010 54, p. 44-62
31. Davis L. A. (2013). Using instructional design principles to develop effective information literacy instruction. The ADDIE model. [Interaktyvus]. [Žiūrėta 2013-12-19] Prieiga per internetą: <http://crln.acrl.org/content/74/4/205.full>
32. Davis N. V. (1997). Reactive and Motivational Agents: Towards a Collective Minder. *Proceeding ATAL '97 Proceedings of the 4th International Workshop on Intelligent Agents IV, Agent Theories, Architectures, and Languages*, p. 309-323
33. Dick W., Carey L. (1990). *The systematic design of instruction*. New York: Harper Collins. Chapter 1: Introduction to instructional design, p. 2-11
34. Dolphin I., Miller P. (2002). Learning Objects and the Information Environment. *Ariadne Issue 32*, 8-July-2002, ISSN: 1361-3200. [Interaktyvus]. [Žiūrėta 2012-12-12]. Prieiga per internetą: <http://www.ariadne.ac.uk/issue32/iconex>
35. Duke R., Salzman E., Burmeister J., Poon J., Murray L. (2000) Teaching Programming to Beginners Choosing the Language is just the First Step, ACE 2000, Melbourne, Australia, *ACM proceedings*, p. 79-86
36. Dushay N., Hillman D. I. (2003). Analyzing Metadata for Effective Use and Re-Use, *2003 Dublin Core Conference*, 28 September - 2 October, 2003, Seattle, USA
37. Ebel G., Ben-Ari M. (2006). Affective Effects of Program Visualization. *ICER '06 Proceedings of the second international workshop on Computing education research*. Canterbury, UK, 2006, p. 1–5. ISBN:1-59593-494-4
38. Eberhart R. C., Shi Y. (2007). *Computational Intelligence. Concepts to Implementations*. Morgan Kaufmann Publishers, United States of America. ISBN 978-1-55860-759-0
39. Esmahi L. (2009). Personalized Web-Based Learning Services. *Encyclopedia of Multimedia Technology and Networking*, Information Science reference, Hershey, New York, Volume I, 2009, p. 814-818
40. Fegen N. (2007). What is the OAI Protocol for Metadata Harvesting. *OAI-PMH Briefing Paper by Neil Fegen*. JISC CETIS, March 2007 (minor revision Oct. 2007)
41. Franklin S., Graesser A. (1996), Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents. *Proceeding of Third International Workshop on Agent Theories, Architectures and Languages*, Springer-Verlag

42. Franzoni A. L., Assar S. (2009). Student Learning Styles Adaptation Method Based on Teaching Strategies and Electronic Media. *Educational Technology & Society*, 12 (4), p. 15–29
43. Gee Q., Wills G., Cooke E. (2005) A first programming language for IT students. In, *6th Annual Conference of the LTSN Centre for ICS, York, UK, 30 Aug - 01 Sep 2005*
44. Gomes A., Mendes A. J. (2007). Learning to program – difficulties and solutions. *International Conference on Engineering Education – ICEE 2007, Coimbra, Portugal*
45. Gries D. (2008) A Principled Approach to Teaching OO First. *Proceedings of the 39th SIGCSE technical symposium on Computer science education*, 2008, p. 31-35
46. Gupta D. (2004) What is a good first programming language? *ACM proceedings*, Volume 10 Issue 4, August 2004
47. Haridi S., Van Roy P. (2004). *Concepts, Techniques, and Models of Computer Programming*. Swedish Institute of Computer Science, 2004
48. Hauger D., Köck M. (2007). State of the Art of Adaptivity in E-Learning Platforms. In: Brunkhorst, I., Krause, D. & Sitou, W. (Hrsg.), *15th Workshop on Adaptivity and User Modeling in Interactive Systems*
49. Hawk T. F., Shah A. J. (2007). Using Learning Style Instruments to Enhance Student Learning. *Decision Sciences Journal of Innovative Education*, Volume 5 Number 1, January 2007 U.S.A., p. 1-19
50. Hayes-Roth B. (1995). An Architecture for Adaptive Intelligent Systems. *Artificial Intelligence: Special Issue on Agents and Interactivity*, 72, p. 329-365
51. Herod L. (2005). Learning Styles and Strategies. *Adult Learning & Literacy*, p. 13–22
52. IMS Meta-data Best Practice Guide for IEEE 1484.12.1-2002 Standard for Learning Object Metadata. 20 May 2004. Prieiga per internetą: http://www.imsglobal.org/metadata/mdv1p3pd/imsmd_bestv1p3pd.html
53. ISO/IEC FCD 25010: Systems and software engineering - Software product Quality Requirements and Evaluation (SQuaRE) – Quality models for software product quality and system quality in use, Department of Industrial and Management Systems Eng. Waseda University, 2009-07-13
54. Jadzgevičienė V., Urbonienė J. (2013). The Possibilities of Virtual Learning Environment Tool Usability for Programming Training. *Proceedings of the 6th International Conference Innovative Information Technologies for Science, Business and Education, IIT-2013* November 14-16
55. Jenkins T. (2002). On the Difficulty of Learning to Program. *Proceedings of 3rd annual conference of the LTSN-ICS*, Loughborough University, United Kingdom, August 2002, p. 53-58
56. Johnson C. G., Fuller U. (2007). Is Bloom’s Taxonomy Appropriate for Computer Science? *6th Baltic Sea Conference on Computing Education Research Koli Calling 2006 Proceedings*, Department of Information

- Technology Uppsala University, Sweden, February 2007, p. 120-122, ISSN 1404-3203
57. Kannangara D. N. P. (2013). Teaching Tools and Techniques for Efficient Teaching and Learning of Computer Programming for Beginners Using JAVA. *Thesis for the Degree of Doctor of Philosophy of Curtin University*
 58. Kinnunen P. (2009). Challenges of teaching and studying programming at a university of technology – Viewpoints of students, teachers and the university. *Dissertation for the degree of Doctor of Philosophy of the Faculty of Information and Natural Sciences at Helsinki University of Technology* (Espoo, Finland), December
 59. Kinnunen P., Malmi L. (2005). Problems in Problem-Based Learning - Experiences, Analysis and Lessons Learned on an Introductory Programming Course. *Informatics in Education*, vol 4 no 2, 2005, p. 193-214
 60. Kölling M., Koch B., Rosenberg J. (1995). Requirements For A First Year Object-Oriented Teaching Language. *Proceedings of 26th SIGCSE Technical Symposium on Computer Science Education*, 1995, Nashville, Tennessee, U.S.A., SIGCSE Bulletin 27, 1, March, p. 173-177
 61. Komlenov Z., Budimac Z., Ivanovic M. (2009). Introducing Adaptivity Features to a Regular Learning Management System to Support Creation of Advanced eLessons. *Informatics in Education*, 2010, Vol. 9, No. 1, p. 63–80
 62. Kubilinskienė S. (2012). Išplėstas skaitmeninių mokymosi išteklių metaduomenų modelis. *Daktaro disertacija*, Vilniaus universitetas, technologijos mokslai, informatikos inžinerija (07 T)
 63. Kurilovas E. (2008). Digital Library of Educational Resources and Services Components Interoperability Problems. *Doctoral Dissertation*, Vilnius Gediminas Technical University, Institute of Mathematics and Informatics, Technological Sciences, Informatics Engineering (07 T)
 64. Kurilovas E. (2012). European Learning Resource Exchange – a platform for collaboration of researchers, policy makers, practitioners, and publishers to share digital learning resources and new e-learning practices. *Chapter 14 in the book: Ahmed Cakir and Patricia Ordóñez de Pablos (Ed.) „Social Development and High Technology Industries: Strategies and Applications“*. IGI Publishing, USA, 2012, p. 200–243, ISBN 978-1-61350-192-4
 65. Kurilovas E., Šėrikovienė S., Vuorikari R. (2014). Expert centred vs learner centred approach for evaluating quality and reusability of learning objects. *Computers in Human Behavior* 30 2014, 526–534
 66. Lawlis P. K. (1997). Guidelines for Choosing A Computer Language: Support For The Visionary Organization. 2nd Edition, 1997, [Interaktyvus]. [Žiūrėta 2010-07-17]. Prieiga per internetą: <<http://archive.adaic.com/docs/reports/lawlis/content.htm>>

67. Lee J., Park Ok-Ch. (2008). *Adaptive Instructional Systems*. Handbook of Research on Educational Communications and Technology. Taylor & Francis Group, New York p. 469-484
68. Libby R., Blashfield R. (1978). Performance of a composite as a function of a number of judges. *Organizational Behavior and Human Performance*, No. 21
69. Lister R., Adams E. S., Fitzgerald S., Fone W., Hamer J., Lindholm M., McCartney R., Moström J. E., Sanders K., Seppällä O., Simon B., Thomas L. (2004). A Multi-National Study of Reading and Tracing Skills in Novice Programmers. *SIGCSE Bulletin*, December 2004, Volume 36, Issue 4, p. 119–150
70. Lister R., Corney M. W., Curran J., D'Souza D., Fidge C. J., Gluga R., Hamilton M., Harland J., Hogan J., Kay J., Murphy T., Roggenkamp M., Sheard J., Simon, Teague D. (2012) Toward a shared understanding of competency in programming : an invitation to the BABELnot Project. In de Raadt, Michael & Carbone, Angela (Eds.) *Proceedings of the 14th Australasian Computing Education Conference (ACE2012)*, Australian Computer Society, RMIT University, Melbourne, Vol. 132
71. Lister R., Leaney J. (2003). First Year Programming: Let All the Flowers Bloom. *5th Australasian Computer Education Conference (ACE2003)*, Adelaide, 2003, Vol. 20
72. Lister R., Simon B., Thompson E., Whalley J. L., Prasad Ch. (2006). Not seeing the forest for the trees: novice programmers and the SOLO taxonomy. *SIGCSE Bull.*, 2006, 38(3), p. 118–122
73. Lumsdaine M., Lumsdaine E. (1995) Thinking Preferences of Engineering Students: Implications for Curriculum Restructuring. *Journal of Engineering Education*. April, 1995, Vol. 84, No. 2. [Interaktyvus]. [Žiūrėta 2013-12-15]. Prieiga per internetą: http://www.innovationtoday.biz/pubs/Thinking_Preferences_of_Engineering_Students.pdf
74. Lupeikienė A. (2007). *Teoriniai ir technologiniai informacinių sistemų aspektai*. Išleista Matematikos ir informatikos institute. UAB „Mokslo aidai“. ISBN 978-9986-680-40-6
75. Mancordis S., Holt R. C., Penny D. A. (1993) A „curriculum-cycle“ environment for teaching programming. *In Proceedings of the Twenty-fifth SIGCSE symposium on Computer Science education*, Indianapolis, IN USA, February 18-19 1993, p. 15-19
76. Mannila L. (2007). Novices' Progress in Introductory Programming Courses. *Informatics in Education*, 2007, Vol. 6, No. 1, p. 139–152
77. Mannila L., De Raadt M. (2006). An Objective Comparison of Languages for Teaching Introductory Programming. *Proceedings*, Koli Calling 2006, p. 32-37
78. Mayer R. E.; Dyck J. L., Vilberg W. (1986) Learning to Program and Learning to Think: What's the Connection? *Communications of the ACM*, July 1986, Vol. 29, No. 7, p. 605-610

79. Merrill D. (2000). Learning Object Design and Sequencing Theory. *Doctoral Dissertation for the Degree of Doctor of Philosophy*. Brigham Young University Department of Instructional Psychology and Technology, 2000, p. 77-79
80. Milbrandt G. (1993). Using Problem Solving to Teach a Programming Language in Computer Studies. *Journal of Computer Science Education*, 1993
81. Moritz S. H., Blank G. D. A (2005) Design-First Curriculum for Teaching Java in a CS1 Course. *ACM SIGCSE Bulletin*, Volume 37 , Issue 2 (June 2005), p. 89 – 93
82. Motiejūnienė E. (2008). Ugdymo individualizavimas ir diferencijavimas. Bendrosios programos
83. Moura H. (2006). Adaptive e-Learning Environment Design. *Interactive Educational Multimedia*, Number 12, April 2006, p. 62–71
84. Nuutila E., Törmä S., Kinnunen P., Malmi L. (2008). Learning Programming with the PBL Method – Experiences on PBL Cases and Tutoring. *Teaching of Programming*, Springer-Verlag Berlin Heidelberg, p. 47-67
85. Nwana H. S. (1996) Software Agents: An Overview. *Knowledge Engineering Review*, 11(3), p. 205-244
86. Papp-Varga Z. (2008). ICT teaching methods – Programming languages. *Annales Mathematicae et Informaticae* 35, p. 163–172
87. Paramythis A., Loidl-Reisinger S. (2004). Adaptive learning environments and e-learning standards. *Electronic Journal on e-Learning* Volume 2 Issue 1, February 2004, p. 181–194
88. Parker K. R., Chao J. T., Ottaway T. A., Chang J. (2006). A Formal Language Selection Process for Introductory Programming Courses. *Journal of Information Technology Education*, 2006, Volume 5, p. 133-151
89. Parko Ok-Ch. (2005) Adaptive Instructional Systems. *Handbook of Research for Educational Communications and Technology*, p. 634-664. Taip pat pasiekama: <http://www.aect.org/edtech/ed1/>
90. Pasha M. Anwar-ur-Rehman; Pasha S. (2012). Missing Elements of Computer Science Curricula 2013. *Global Journal of Computer Science and Technology Cloud & Distributed*, Volume 12 Issue 11 Version 1.0, 2012. Online ISSN: 0975-4172, Print ISSN: 0975-4350
91. Pérez-Quiñones M. A. (1998) Teaching History of Programming Languages to Undergraduate Students. *Proceedings of 28th ASEE/IEEE Frontiers in Education Conference*, Tempe, Arizona, p. 301-306
92. Preidys S., Žilinskienė I. (2012). Nuotolinio mokymosi kurso personalizavimo modelis mokymosi veiklų atžvilgiu. *Electronic Learning, Information and Communication: Theory and Practice (Elektroninis mokymasis, Informacija ir komunikacija: Teorija ir praktika)*. Vilnius: Vilniaus universitetas, 2012, ISSN: 23352493, p. 111-132

93. Renaud K., Barrow J., le Roux P. (2001). Teaching Programming from a Distance – Problems and a Proposed Solution *ACM SIGCSE Bulletin*, Volume 33 Issue 4, December 2001 New York, NY, USA
94. Rezaei A. R., Katz L. (2004). Evaluation of the reliability and validity of the cognitive styles analysis. *Personality and Individual Differences*, 36, 2004, p. 1317–1327
95. Robins A., Rountree J., Rountree N. (2003). Learning and Teaching Programming: A Review and Discussion. *Computer Science Education*, 2003, Vol. 13, No. 2, p. 137–172
96. Russell G. (2004). Herrmann Brain Dominance Instrument. *The Ned Herrmann Group, Inc.* 23 p.
97. Sammet, J. E. (1996) “From HOPL to HOPL-II (1978-1993): 15 Years of Programming Language Development”, in *History of Programming Languages*, Bergin, T. J., Gibson, R. G. (Eds), ACM Press, New York, NY., 1996, p.16-23
98. Schute J. V., Zapata-Rivera D. (2008) Adaptive Technologies. *Handbook of Research on Educational Communications and Technology*. Taylor & Francis Group, New York, p. 277-294
99. Sebesta R. W. (2012) *Concepts of Programming Languages*. ISBN 10: 0-13-139531-9
100. Seels B., Glasgow Z. (1998). *Making Instructional Design Decisions*. (2nd ed.) Upper Saddle River, NJ: Merrill.
101. Shoham Y. (1997) An Overview of Agent-oriented Programming. In *Software Agents*, ed. J.M. Bradshaw. Menlo Park, Calif.: AAAI Press
102. Skūpienė J. (2010) Evaluation of Algorithm-Code Complexes in Informatics Contests. *Doctoral Dissertation for Physical Sciences (P 000), Informatics (09 P), Informatics, Systems Theory (P 175)*, Vytautas Magnus University Institute of Mathematics and Informatics of Vilnius University
103. Sorva J. (2012). Visual Program Simulation in Introductory Programming Education. *Doctoral Disertation for the degree of Doctor of Science in Technology*, Aalto University School of Science, 2012
104. Spinellis D. (2003). Reading, writing, and code. *ACM Queue*, 2003, Vol. 1, Issue 7, p. 84–89
105. Starr Ch. W., Manaris B., Stalvey R. H. (2008). Bloom’s Taxonomy Revisited: Specifying Assessable Learning Objectives in Computer Science. *SIGCSE ’08*, March 12–15, 2008, Portland, Oregon, USA, ACM 978-1-59593-947-0/08/0003
106. Taylor H. G., Aiken R. M., Van Weert T. J. (1993). Guidelines for Good Practice: Informatics Education in Secondary Schools, IFIP Working Group 3.1, Geneva, 1993
107. Tennant R. (2004). Bitter Harvest: Problems & Suggested Solutions for OAI-PMH Data & Service Providers, California Digital Library, Last text revision 14 May 2004

108. Teresevičienė M., Oldroyd D., Gedvilienė G. (2004). *Suaugusiųjų mokymasis*. Andragogikos didaktikos pagrindai. Vadovėlis. Kaunas: VDU
109. The Open Archives Initiative Protocol for Metadata Harvesting. Protocol Version 2.0 of 2002-06-14. Document Version 2008-12-07T20:42:00Z. [Interaktyvus]. [Žiūrėta 2014-01-15]. Prieiga per internetą: <http://www.openarchives.org/OAI/2.0/openarchivesprotocol.htm>
110. Thompson E., Luxton-Reilly A., Whalley J. L., Hu M., Robbins P. (2008). Bloom's Taxonomy for CS Assessment. *Tenth Australasian Computing Education Conference (ACE2008) in Research and Practice in Information Technology Proceedings*, Wollongong, Australia, January 2008, Vol. 78
111. Van Weert T. J. (1993). Guidelines for Good Practice: Integration of Information Technology into Secondary Education, IFIP Working Group 3.1, Geneva, 1993
112. Weber G., Brusilovsky P. (2001). ELM-ART: An adaptive versatile system for Web-based instruction. *International Journal of Artificial Intelligence in Education*, 12, 2001, p. 351-384
113. Whalley J., Clear T., Lister R. (2007). The many ways of the BRACElet project. *Bulletin of Applied Computing and Information Technology*, 5(1). [Interaktyvus]. [Žiūrėta 2012-08-15]. Prieiga per internetą: http://www.naccq.ac.nz/bacit/0501/2007Whalley_BRACELET_Ways.htm
114. Wiley D. A. (2000). Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy. In David A. Wiley (Ed.)
115. Winslow L. E. (1996). Programming pedagogy, a psychological overview. *ACM SIGCSE Bull.*, 1996, Vol. 28, Issue 3, p. 17–22
116. Yang D, Yang Q. (2005). Customizable Distance Learning: Criteria for Developing Learning Objects and Learning Model Templates. In *Proceedings of the 7th international conference on Electronic commerce (ICEC'05)*, ACM International Conference Proceeding Series, August, Xi'an, (China). ACM. (New York), 2005, p. 765-770

Pasaulinių informatikos mokymo programų (*Computer Science Curriculum 2008*, CS2008 ir *Computer Science Curriculum 2013*, CS2013) struktūros pavyzdžiai

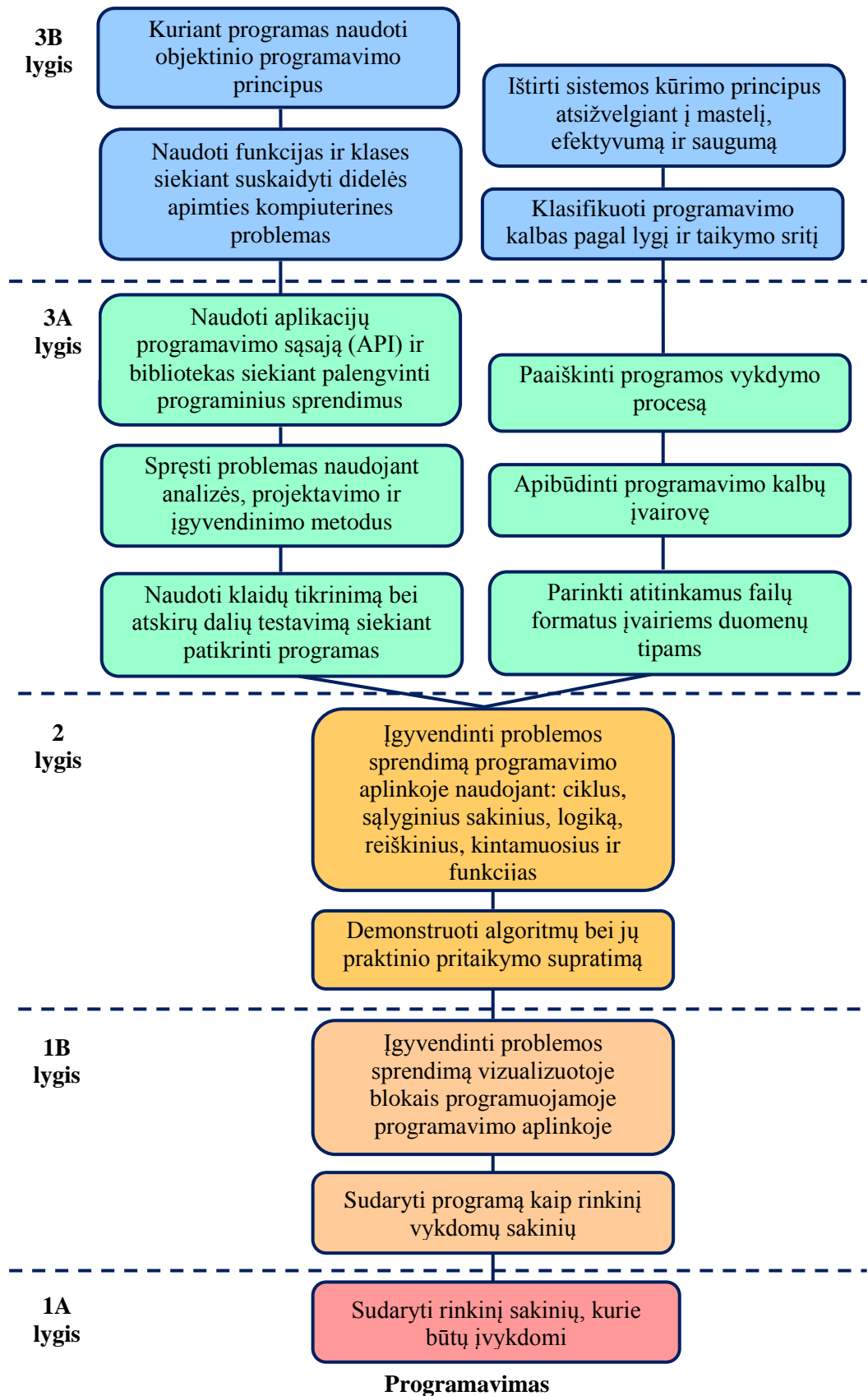
Siūloma dalyko *Programavimo pagrindai* sandara:

- Pagrindiniai konstruktai (angl. *Fundamental Constructs*);
- Algoritminis problemų sprendimas (angl. *Algorithmic Problem Solving*);
- Duomenų struktūros (angl. *Data Structures*);
- Rekursija (angl. *Recursion*);
- Įvykiais grįstas programavimas (angl. *Event Driven programming*);
- Objektinis programavimas (angl. *Object Oriented*);
- Informacijos saugumo pagrindai (angl. *Foundations Information Security*);
- Saugus programavimas (angl. *Secure Programming*).

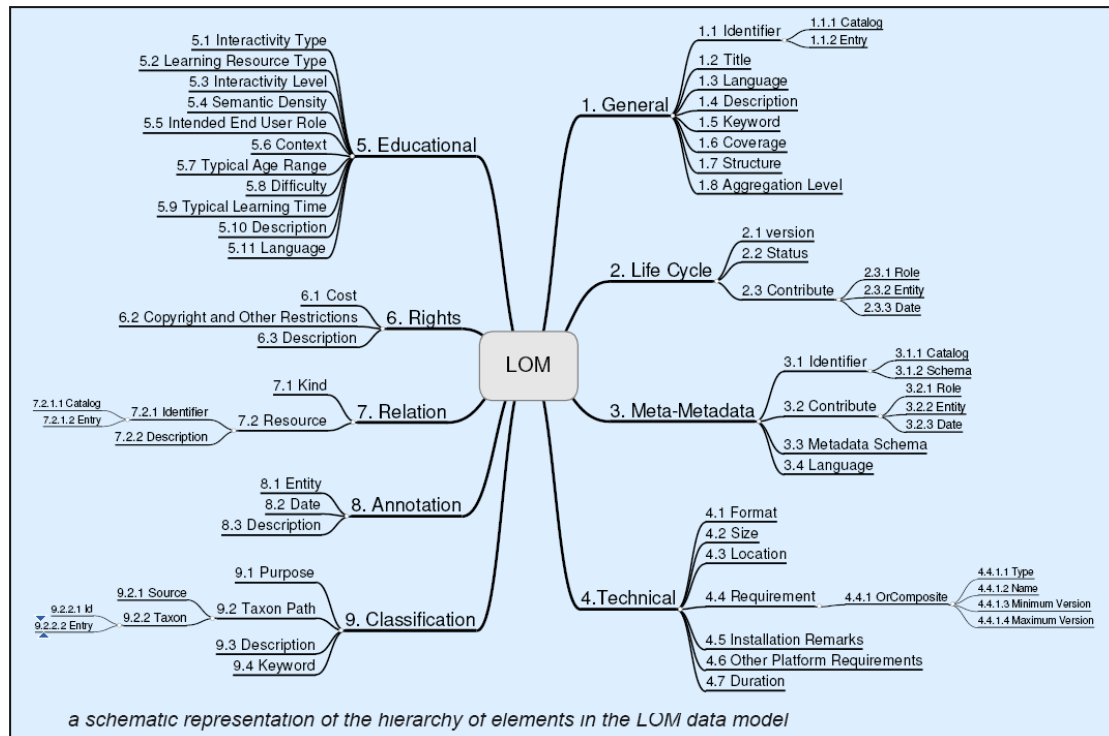
Dalyko *Algoritmai ir sudėtingumas* sandara:

- Pradinė analizė (angl. *Basic Analysis*);
- Algoritmavimo strategija (angl. *Algorithmic Strategies*);
- Pagrindiniai algoritmai (angl. *Fundamental Algorithms*);
- Paskirstyti algoritmai (angl. *Distributed Algorithms*);
- Pagrindiniai skaičiavimai (angl. *Basic Computability*);
- P prieš NP (angl. *P versus NP*);
- Automatų teorija (angl. *Automata Theory*);
- Sudėtinga analizė (angl. *Advanced Analysis*);
- Kriptografiniai algoritmai (angl. *Cryptographic Algorithms*);
- Geometriniai algoritmai (angl. *Geometric Algorithms*);
- Lygiagretūs algoritmai (angl. *Parallel Algorithms*).

CSTA K-12 Computer Science Standards 2011 m. apibrėžiamos „Kompiuterijos praktikos ir programavimo“ dalykinės kompetencijos



LOM metaduomenų, naudojamų mokymosi objektams aprašyti, schema



76 pav. LOM metaduomenų schema

Paimta iš *IMS Meta-data Best Practice Guide for IEEE 1484.12.1-2002 Standard for Learning Object Metadata*. Prieiga internetu:

http://www.imsglobal.org/metadata/mdv1p3pd/imsmd_bestv1p3pd.html

3 PRIEDAS.

Mokymosi objekto metaduomenims aprašyti klasifikatoriai

Direktorius Vaino Brazdeikis

PATVIRTINTA

Švietimo informacinių technologijų centro direktoriaus 2010 m. gruodžio 06 d.

įsakymu

Nr. V1-175

MOKYMOŠI OBJEKTŲ TIPŲ KLASIFIKATORIUS

Klasifikatoriaus pavadinimas lietuvių kalba	Mokymosi objektų tipai
Klasifikatoriaus pavadinimas anglų kalba	Learning Resources Types
Klasifikatoriaus pavadinimo lietuvių kalba abėcėlinė santrumpa	LOM_OBJTIPAI
Klasifikatoriaus paskirtis	Klasifikuoti mokymosi objektus į tipus
Klasifikatoriaus tipas	Lokalus

Klasifikatoriaus duomenys:

Eil. Nr.	Kodas	Pavadinimas lietuvių kalba	Pavadinimas anglų kalba	Aprašymas
1.	5.2.01.0	testas	test	Testas, anketa, apklausa ir pan. žinių patikrinimui. Pagal <i>LRE Learning resours types</i> žodyno reikšmę <i>assessment</i>
2.	5.2.02.0	Klausimų rinkinys	question bank	Klausimų aibė testui sudaryti. Pagal <i>LRE Learning resours types</i> žodyno reikšmę <i>assessment</i>
3.	5.2.03.0	testo elementas	assessment item	Testo dalis: klausimas ir jo teisingas atsakymas. Pagal <i>LRE Learning resours types</i> žodyno reikšmę <i>assessment</i> .
4.	5.2.04.0	demonstracija	demonstration	Mokymosi objekto vizuali priemonė. Pagal <i>LRE Learning resours types</i> žodyno reikšmę <i>demonstration</i>
5.	5.2.05.0	pratybos, praktinė užduotis	drill and practice	Praktinius įgūdžius formuojanti priemonė Pagal <i>LRE Learning resours types</i> žodyno reikšmę <i>drill and practice</i>
6.	5.2.06.0	tyrinėjimas	exploration	Mokymosi objekto pažinimas, jį stebint, tiriant, bandant, sprendžiant iškeltas problemas ir formuluojant išvadas (pažinimas eksperimentuojant). Pagal <i>LRE Learning resours types</i> žodyno reikšmę <i>exploration</i>
7.	5.2.07.0	modeliavimas (imitavimas)	simulation	Modelis, iliustracija, animacija, imituojantis pavyzdys ir pan. Pagal <i>LRE Learning resours types</i> žodyno reikšmę <i>simulation</i>
8.	5.2.08.0	edukacinis žaidimas	educational game	Edukacinis žaidimas, suteikiantis dalykinių žinių ir lavinantis gebėjimus. Pagal <i>LRE Learning resours types</i> žodyno reikšmę <i>educational game</i>

Eil. Nr.	Kodas	Pavadinimas lietuvių kalba	Pavadinimas anglų kalba	Aprašymas
9.	5.2.09.0	išteklų paketas	resource pack	Kursui, programai, moduliui, pamokos temai skirtas rinkinys priemonių su instrukcijomis mokytojui ir mokiniui Pagal <i>LRE Learning resours types</i> žodyno reikšmę <i>course</i>
10.	5.2.10.1	garso įrašas (komponentė)	learning asset / audio	Garso įrašas, skirtas mokymuisi arba mokymosi objekto kūrimui.. Pagal <i>LRE Learning resours types</i> žodyno reikšmę <i>audio</i>
11.	5.2.10.2	paveikslas (komponentė)	learning asset / image	Paveikslas, skirtas mokymuisi arba mokymosi objekto kūrimui.. Pagal <i>LRE Learning resours types</i> žodyno reikšmę <i>image</i>
12.	5.2.10.3	vaizdo įrašas (komponentė)	learning asset / video	Videofilmas, skirtas mokymuisi arba mokymosi objekto kūrimui. Pagal <i>LRE Learning resours types</i> žodyno reikšmę <i>video</i>
13.	5.2.11.0	taikomoji programa	application	Kompiuterinė programa mokymui ir praktinėms užduotims, pvz: tekstų rengyklė, skaičiuoklė, grafikos programa, interneto naršyklė ir pan. Pagal <i>LRE Learning resours types</i> žodyno reikšmę <i>application</i>
14.	5.2.12.0	įrankis	tool	Redagavimo, piešimo ir kitų funkcijų kompiuterinės programos naudotojo kūrybai. . Pagal <i>LRE Learning resours types</i> žodyno reikšmę <i>tool</i>
15.	5.2.13.0	programos, modulio, skyriaus planas	course/ module programme	Programos, jos modulio, skyriaus, temos nustatytam laikotarpiui mokymo planas, nurodant tikslus, uždavinius, mokytojo ir mokinių veiklą, priemones. Pagal <i>LRE Learning resours types</i> žodyno reikšmę <i>course</i>
16.	5.2.14.0	pamokos ar pamokų ciklo planas	lesson plan	Dalyko pamokos ar pamokų ciklo planas, nurodant tikslus, uždavinius, mokytojo ir mokinių veiklą, priemones. Pagal <i>LRE Learning resours types</i> žodyno reikšmę <i>lesson plan</i>
17.	5.2.15.0	pedagogo vadovas	educator guide	Rekomendacinė paremta gerąja patirtimi pagalbos knyga pedagogui. Pagal <i>LRE Learning resours types</i> žodyno reikšmę <i>guide</i>
18.	5.2.16.0	vertinimas (atestavimas)	evaluation form	Vertinimo metodai ir jų taikymo priemonė. Pagal <i>LRE Learning resours types</i> žodyno reikšmę <i>assessment</i>
19.	5.2.17.0	naudotojo vadovas	user guide	Instrukcija naudotojui kaip dirbti su mokymosi objekto priemone. Pagal <i>LRE Learning resours types</i> žodyno reikšmę <i>guide</i>
20.	5.2.18.0	atvejo tyrimas	case study	Atvejo, proceso, temos tyrimo aprašymas, pritaikytas diskusijai, analizei, pristatymui Pagal <i>LRE Learning resours types</i> žodyno reikšmę <i>case study</i>

Eil. Nr.	Kodas	Pavadinimas lietuvių kalba	Pavadinimas anglų kalba	Aprašymas
21.	5.2.19.0	paieška	enquiry-oriented activity	Informacijos paieškai, apdorojimui, tvarkymui, arba pritaikymui skirtas aprašymas. Pagal <i>LRE Learning resours types</i> žodyno reikšmę <i>enquiry-oriented activity</i>
22.	5.2.20.0	eksperimentas	experiment	Eksperimento, hipotezės, tiesos patikrinimo arba naujo atradimui skirtas aprašymas. Pagal <i>LRE Learning resours types</i> žodyno reikšmę <i>experiment</i>
23.	5.2.21.0	projektas	project	Projekto aprašymas, turintis kokybinius ir kiekybinius rodiklius, unikalų tikslo pasiekimų kelią.. Pvz., tarpdalykinių ryšių, bendradarbiavimo, klasės veiklos, konkurso, egzaminų, nacionalinis arba Europos projektas mokymosi tikslais ir kt. Pagal <i>LRE Learning resours types</i> žodyno reikšmę <i>project</i>
24.	5.2.22.0	vaidmenų žaidimas	role play	Mokymosi iš žaidimo vaidmenų aprašymas. Pagal <i>LRE Learning resours types</i> žodyno reikšmę <i>role play</i>
25.	5.2.23.0	popamokinė veikla (neformalusis ugdymas)	after-school activities	Organizuotos popamokinės veiklos (neformaliojo vaikų ugdymo) aprašymas, skirtas asmeninėms, socialinėms, edukacinėms, profesinėms kompetencijoms ugdyti. Pagal <i>LRE Learning resours types</i> žodyno reikšmę <i>open activity</i>
26.	5.2.24.0	straipsnis	article	Metodinis leidinio straipsnis, jo aprašymas Pagal <i>LOM Learning resours types</i> žodyno reikšmę <i>reference</i>
27.	5.2.25.0	skaitmeninis leidinys	digitize publication	Knyga, vadovėlis, straipsnis skaitmenine (elektronine) forma. Pagal <i>LOM Learning resours types</i> žodyno reikšmę <i>referencet</i>
28.	5.2.26.0	glosarijus (aiškinamųjų terminų žodynas)	glossary	Vienakalbis, dvikalbis arba daugiakalbis paaiškinantis terminus, žodžius, ženklus žodynas Pagal <i>LRE Learning resours types</i> žodyno reikšmę <i>glossary</i>
29.	5.2.27.0	žodynas	vocabulary	Žodžių arba dalykinių sričių terminų sąrašas su vienos ar kelių kalbų atitikmenimis. Pagal <i>LRE Learning resours types</i> žodyno reikšmę <i>reference</i>
30.	5.2.28.0	enciklopedinis šaltinis	reference	Enciklopedija, duomenų bazė, informacinė sistema ir kt. šaltinis temos informacijai gauti. Pagal <i>LRE Learning resours types</i> žodyno reikšmę <i>reference</i>
31.	5.2.29.0	pateiktis	presentation	Skaidrėmis teikiama informacija, apipavidalinant ją programine įranga. Pagal <i>LRE Learning resours types</i> žodyno reikšmę <i>presentation</i>
32.	5.2.30.1	interneto tinklaraštis	web resource / weblog	Autoriaus arba grupės kūrėjų leidinys skelbiamas kompiuterių tinkle (žiniatinklyje) Pagal <i>LRE Learning resours types</i> žodyno reikšmę <i>weblog</i>

Eil. Nr.	Kodas	Pavadinimas lietuvių kalba	Pavadinimas anglų kalba	Aprašymas
33.	5.2.30.2	interneto vikis	web resource / wiki	Žiniatinklio rūšis, kurios turinį naudotojas gali papildyti, patikslinti, kurti (Vikiu taip pat vadinama ir programinė įranga tokių svetainių kūrimui). Pagal <i>LRE Learning resours types</i> žodyno reikšmė <i>wiki</i>
34.	5.2.30.3	mokymosi svetainė	web resource / learning web site	Tarpusavyje susieti keli arba daugiau tinklapių, turintys apibendrinantį pradinį puslapį (home page) mokymosi tikslais. Pagal <i>LRE Learning resours types</i> žodyno reikšmė <i>web page</i>
35.	5.2.30.4	teminė mokymosi svetainė	web resource / thematic website	Visuotinė vienos srities, temos svetainė skirta visuomenės ugdymui ir švietimui. Pagal <i>LRE Learning resours types</i> žodyno reikšmė <i>web page</i>
36.	5.2.30.5	konkursų, olimpiadų svetainė	web resource / competitionwebsite	Informacinė interneto svetainė apie nuolat vykstančius konkursus, olimpiadas, varžybas, egzaminus. Pagal <i>LRE Learning resours types</i> žodyno reikšmė <i>web page</i>
37.	5.2.30.6	kitas interneto išteklius	other web resource	Kita klasifikatoriuje nenurodytas žiniatinklis skirtas mokymui. Pagal <i>LRE Learning resours types</i> žodyno reikšmė <i>other web resource</i>
38.	5.2.31.0	kitas išteklius	other	Kitas klasifikatoriuje nenurodytas mokymosi objekto tipas. Pagal <i>LRE Learning resours types</i> žodyno reikšmė <i>other</i>

Pastaba:

Klasifikatoriaus kodą sudaro 5 skaitmenys:

I ir II skaitmenys (5.2) atitinka IEEE LOM standarto lauko numerį 5.2,

III ir IV skaitmenys nurodo standarto lauko 5.2 įrašus eilės tvarka (nuo 01 iki 31),

V skaitmuo nurodo III ir IV skaitmenimis pažymėto įrašo skaidymą (hierarchiją).

MOKYMOŠI OBJEKTO TURINIO APIMTIES LYGMENS KLASIFIKATORIUS

Klasifikatoriaus pavadinimas lietuvių kalba	Mokymosi objekto turinio apimties lygmuo
Klasifikatoriaus pavadinimas anglų kalba	The content granularity of learning object
Klasifikatoriaus pavadinimo lietuvių kalba abėcėlinė santrumpa	LOM_TALYGM
Klasifikatoriaus paskirtis	Klasifikuoti mokymosi objektą pagal turinio apimtį į lygmenis
Klasifikatoriaus tipas	Lokalus

Klasifikatoriaus duomenys:

Eil. Nr.	Kodas	Pavadinimas lietuvių kalba	Pavadinimas anglų kalba	Aprašymas
1.	1.8.1	1 - mokymosi komponentas, jų aibė	1: learning asset, a set of asset	Pirmas (žemiausias) turinio apimties lygmuo yra mokymosi elementas arba jų aibė, pvz. tekstas, piešinys, nuotrauka, grafikas, trumpas filmas, garso įrašas, užduotės sprendimas. Pagal <i>LOM Aggregation Levels</i> žodyno reikšmė

Eil. Nr.	Kodas	Pavadinimas lietuvių kalba	Pavadinimas anglų kalba	Aprašymas
				„1“.
2.	1.8.2	2 - pamoka, pamokų ciklas	2: lesson	Antras turinio apimties lygmuo yra mokymosi objektų rinkinys, pvz. pamoka, pamokų ciklas. Pagal <i>LOM Aggregation Levels</i> žodyno reikšmę „2“.
3.	1.8.3	3 - kursas, modulis	3: course, modul	Trečias turinio apimties lygmuo yra antro lygmens mokymosi objektų rinkinys, pvz. kursas, skyrius. Reikšmė Pagal <i>LOM Aggregation Levels</i> žodyno reikšmę „3“.
4.	1.8.4	4 - programa, kursų rinkinys	4: program, a set of courses	Ketvirtas (aukščiausias) turinio apimties lygmuo, pvz kursų rinkinys, kuris suteikia sertifikavimo teisę Pagal <i>LOM Aggregation Levels</i> žodyno reikšmę „4“.

Pastaba:

Klasifikatoriaus kodą sudaro 3 skaitmenys:

I ir II skaitmenys (1.8) atitinka IEEE LOM standarto lauko numerį 1.8,

III skaitmuis nurodo turinio lygmens numerį.

HBDI klausimynas

Prašome atsakyti į kiekvieną klausimą taip, kaip yra nurodoma šalia jo. Kiekvienas atsakymas pateikia tam tikrus duomenis, jeigu liks neatsakytų klausimų, nebus galima tinkamai interpretuoti gautus rezultatus.

BIBLIOGRAFINĖ INFORMACIJA

1. Vardas, pavardė _____ 2. Lytis vyras moteris
 3. Išsilavinimas _____
 4. Užsiėmimas ar pareigos darbovietėje _____
 Aprašykite savo darbą _____
 (pasistenkite aprašyti kuo tiksliau) _____

RANKOS DOMINAVIMAS

5. Kuris paveikslėlis geriausiai apibūdintų tai, kaip Jūs laikote pieštuką? Pažymėkite **A, B, C** arba **D** variantą



6. Kaip stipriai ir kuri Jūsų ranka dominuoja? Pažymėkite **A, B, C, D** arba **E** variantą

A pagrindinė kairė B pagrindinė kairė, šiek tiek dešinė C abi rankos lygios
 D pagrindinė dešinė, šiek tiek kairė E pagrindinė dešinė

GERIAUSIAS/BLOGIAUSIAS DALYKAS

Pagalvokite, kuris žemiau nurodytas dalykas buvo geriausias/blogiausias mokykloje. Išreitinguokite visus dalykus šalia pažymėdami **1** – jei tai geriausias dalykas, **2** – jei tai antrasis pagal gerumą arba **3** – jei tai trečiasis.

7. _____ matematika 8. _____ užsienio kalba 9. _____ gimtoji kalba

PATIKRINKITE: AR SKAIČIUS 1, 2 IR 3 YRA ĮRAŠYTAS TIK VIENĄ KARTĄ?

PATAISYKITE, JEI REIKIA!

DARBO ELEMENTAI

Įvertinkite kiekvieną žemiau pateiktą darbo elementą pagal jų stiprumą Jūsų veikloje, naudokite skalę:

5 = darbą atlieku geriausiai, **4** = darbą atlieku gerai, **3** = neutralu, **2** = darbą atlieku mažiau gerai, **1** = darbą atlieku mažiausiai gerai.

10. _____ analitinį 16. _____ techninius aspektus 21. _____ inovacijas
 11. _____ administracinį 17. _____ įgyvendinimą 22. _____ mokymą
 12. _____ konceptualizavimą 18. _____ planavimą 23. _____ organizavimą
 13. _____ idėjų reiškimą 19. _____ tarpasmeninius aspektus 24. _____ kūrybinius aspektus
 14. _____ integravimą 20. _____ problemų sprendimą 25. _____ finansinius aspektus
 15. _____ rašymą

PRAŠOME PERŽIŪRĖTI: 5-TŲ4-TŲ 3-TŲ 2-TŲ 1-TŲ SKAIČIUS. JEI NORS VIENOJE KATEGORIJOJE JŲ YRA DAUGIAU KAIP 4, PERSKIRSTYKITE IŠ NAUJO

SVARBIAUSI DESKRIPTORIAI

Pažymėkite aštuonis būdvardžius, kaip Jūs geriausiai apibūdintumėte save. Parašykite po 2 prie kiekvieno Jūsų pasirinkimo. Tada ties vienu būdvardžiu, kuris geriausiai apibūdina Jus, 2 pakeiskite į 3.

- | | | |
|-------------------------|----------------------------|-------------------------|
| 26. _____ loginis | 35. _____ emocionalus | 43. _____ simbolinis |
| 27. _____ kūrybiškas | 36. _____ erdvinis | 44. _____ dominuojantis |
| 28. _____ muzikalus | 37. _____ kritiškas | 45. _____ holistinis |
| 29. _____ nuoseklus | 38. _____ artistiškas | 46. _____ intuityvus |
| 30. _____ sintezatorius | 39. _____ dvasiškas | 47. _____ kiekybinis |
| 31. _____ verbalinis | 40. _____ racionalus | 48. _____ skaitytojas |
| 32. _____ konservatyvus | 41. _____ kontroliuojantis | 49. _____ vienalaikis |
| 33. _____ analitiškas | 42. _____ matematinis | 50. _____ faktiškas |

PRAŠOME SUSKAIČIUOTI: SEPTYNI 2-TAI IR VIENAS 3-TAS

HOBII

Pažymėkite daugiausiai šešis labiausiai Jums tinkančius hobius. Parašykite 3 prie Jūsų mėgstamiausio hobbio, 2 – prie pagrindinių hobių ir 1 prie papildomų hobių.

- | | | |
|---------------------------|--------------------------------|----------------------------------|
| 51. _____ menas/amatas | 59. _____ sodininkystė/augalai | 67. _____ siuvimas |
| 52. _____ irklavimas | 60. _____ golfas | 68. _____ „sirgaliaus“ |
| 53. _____ stovyklavimas | 61. _____ namų tobulinimas | 69. _____ plaukimas/
nardymas |
| 54. _____ kortos | 62. _____ muzikos klausymas | 70. _____ tenisas |
| 55. _____ kolekcionavimas | 63. _____ muzikavimas | 71. _____ kelionės |
| 56. _____ kepimas | 64. _____ fotografija | 72. _____ medžio darbai |
| 57. _____ kūryba | 65. _____ skaitymas | _____ kita _____ |
| 58. _____ žvejyba | 66. _____ buriavimas | _____ |

PRAŠOME PERŽIŪRĖKITE: TIK VIENAS 3-TAS IR NE DAUGIAU KAIP ŠEŠI HOBIAI. PATAISYKITE, JEI REIKIA!

ENERGIJOS LYGIS

73. Pagalvokite apie savo energijos lygį ar jėgas. Pažymėkite variantą, kuris labiausiai Jus atitinka – **A**, **B** arba **C**.

- A dienos žmogus B tiek dienos, tiek nakties žmogus C nakties žmogus

SUPIMO LIGA

74. Ar jūs kada nors judant transporto priemone (kai judama automobiliu, valtimi, lėktuvu, autobusu, traukiniu, karuselėse) patyrėte pykinimo (pykinimo, vėmimo) jausmą? Kartų skaičiui nurodyti pažymėkite **A**, **B**, **C** arba **D**

- A niekada B 1-2 C 3-10 D daugiau kaip 10

75. Važiuojant automobilyje ir skaitant Jūs nejaučiate skrandžio dirginimo, nepykina ar nevemiame? Pažymėkite **A** arba **B** variantą.

- A taip B ne

BŪDVARDŽIŲ POROS

Kiekvienai žemiau pateiktai elementų porai pažymėkite žodį ar frazę, kuri geriausiai apibūdina Jus. Kiekvienu atveju pažymėkite **A** arba **B** variantą, net ir tuo atveju, jei pasirinkti yra sudėtinga.

NEPRALEISKITE NEI VIENOS POROS

- | | | | | | | | |
|-----|---|------|--|-----|--|------|--|
| 76. | konservatyvus <input type="checkbox"/> / <input type="checkbox"/> | A/ B | <input type="checkbox"/> empatiškas | 88. | lakios vaizduotės <input type="checkbox"/> / <input type="checkbox"/> | A/ B | <input type="checkbox"/> nuoseklus |
| 77. | analitikas <input type="checkbox"/> / <input type="checkbox"/> | | <input type="checkbox"/> sintezatorius | 89. | originalus <input type="checkbox"/> / <input type="checkbox"/> | | <input type="checkbox"/> patikimas |
| 78. | kiekinis <input type="checkbox"/> / <input type="checkbox"/> | | <input type="checkbox"/> muzikalus | 90. | kūrybingas <input type="checkbox"/> / <input type="checkbox"/> | | <input type="checkbox"/> logiškas |
| 79. | problemų sprendėjas <input type="checkbox"/> / <input type="checkbox"/> | | <input type="checkbox"/> planuotojas | 91. | susitvardantis <input type="checkbox"/> / <input type="checkbox"/> | | <input type="checkbox"/> emocionalus |
| 80. | valdomas <input type="checkbox"/> / <input type="checkbox"/> | | <input type="checkbox"/> kūrybingas | 92. | muzikalus <input type="checkbox"/> / <input type="checkbox"/> | | <input type="checkbox"/> detalus |
| 81. | originalus <input type="checkbox"/> / <input type="checkbox"/> | | <input type="checkbox"/> emocionalus | 93. | vienalaikis <input type="checkbox"/> / <input type="checkbox"/> | | <input type="checkbox"/> empatiškas |
| 82. | jausmingas <input type="checkbox"/> / <input type="checkbox"/> | | <input type="checkbox"/> mąstantis | 94. | kommunikatorius <input type="checkbox"/> / <input type="checkbox"/> | | <input type="checkbox"/> konceptualizatorius |
| 83. | tarpasmeninis <input type="checkbox"/> / <input type="checkbox"/> | | <input type="checkbox"/> organizatorius | 95. | besidomintis <input type="checkbox"/> / <input type="checkbox"/> | | <input type="checkbox"/> besidomintis |
| 84. | dvasiškas <input type="checkbox"/> / <input type="checkbox"/> | | <input type="checkbox"/> kūrybingas | 96. | techniniais dalykais <input type="checkbox"/> / <input type="checkbox"/> | | <input type="checkbox"/> žmonėmis |
| 85. | detalus <input type="checkbox"/> / <input type="checkbox"/> | | <input type="checkbox"/> holistinis | 97. | gerai organizuotas <input type="checkbox"/> / <input type="checkbox"/> | | <input type="checkbox"/> logiškas |
| 86. | generuoti idėjas <input type="checkbox"/> / <input type="checkbox"/> | | <input type="checkbox"/> išbandyti ir patvirtinti idėjas | 97. | griežtai mąstantis <input type="checkbox"/> / <input type="checkbox"/> | | <input type="checkbox"/> mąstantis |
| 87. | šiltas, draugiškas <input type="checkbox"/> / <input type="checkbox"/> | | <input type="checkbox"/> analitinis | 98. | patinka suplanuoti dalykai <input type="checkbox"/> / <input type="checkbox"/> | | <input type="checkbox"/> patinka tikslūs dalykai |
| | | | | 99. | specialus <input type="checkbox"/> / <input type="checkbox"/> | | <input type="checkbox"/> vyraujantis |

PRAŠOME PERŽIŪRĖKITE: AR PAŽYMĖJOSTE VEINĄ IR TIK VIENĄ KIEKVIENOJE POROJE? PATAISYKITE, JEI REIKIA!

INTRAVERTIŠKUMAS/EKSTRAVERTIŠKUMAS

100. Pateiktoje intraverto-ekstraverto skalėje pažymėkite tik vieną variantą, kuris tinka Jums.

Intravertas

|

Ekstravertas

--

|

|

|

DVIDEŠIMT KLAUSIMŲ

		Visiškai sutinku	Sutinku	Tarp	Nesutinku	Visiškai nesutinku
101.	Man atrodo, kad žingsnis po žingsnio metodas yra geriausias problemų sprendimui	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
102.	Fantazija man suteikia impulsą spręsti svarbias problemas	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
103.	Man patinka žmonės kurie yra tvirtai apsisprendę dėl savo išvadų	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
104.	Verčiau aš būsiu žinomas kaip patikimas žmogus, negu kaip žmogus su fantazija	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

		Visiškai sutinku	Sutinku	Tarp	Nesutinku	Visiškai nesutinku
105.	Genialios idėjos man ateina tada, kai aš praktiškai nieko neveikiu	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
106.	Sprendžiant problemą aš vadovaujuosi nuojauta ir supratimu kas yra bloga ir gera	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
107.	Man kartais patinka laužyti taisykles ir daryti ne taip kaip yra siūloma	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
108.	Daug svarbių dalykų neįmanoma išreikšti žodžiais	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
109.	Dažniausiai aš konkuruoju su kitais negu su savimi	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
110.	Aš su malonumu praleisčiau visą dieną „vienas su savo mintimis“	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
111.	Aš nemėgstu netikrų ir sunkiai nuspėjamų dalykų	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
112.	Aš labiau mėgstu dirbti komandoje su kitais nei vienas	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
113.	Man svarku, kad viskas turėtų savo vietą ir viskas būtų savo vietoje	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
114.	Neįprastos idėjos ir drasios koncepcijos domina ir intriguoja mane	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
115.	Man patinka konkretūs nurodymai, kuriuose pateikiama papildoma informacija	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
116.	Žinojimas „kodėl“ yra svarbesnis už žinojimą „kaip“	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
117.	Kruopštus planavimas ir laiko organizavimas yra privalomas sprendžiant sudėtingas problemas	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
118.	Aš dažnai galiu numatyti savo problemų sprendimus	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
119.	Aš linkęs pasikliauti pirmu išpūdžiu ir nuojauta priimant sprendimus, o ne nuodugnia situacijos analize	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
120.	Aš norėčiau, kad įstatymų būtų griežtai laikomasi	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Prašome patikrinkite ar atsakėte į visus klausimus šioje dalyje.

PAPILDOMI KLAUSIMAI

Kokie dalykai Jus motyvuoja, leidžia judėti pirmyn?

Ką norėtum savyje pakeisti?

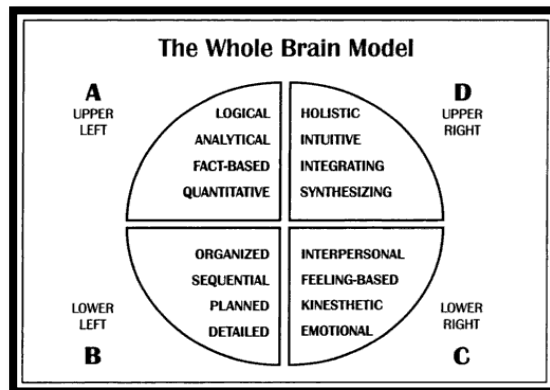
Ar yra dalykų kuriuos norėtum patobulinti (galite parašyti kelis)?

Kuris iš aukščiau išvardytų dalykų yra svarbiausias Jums?

Mokymosi objektų tipų tinkamumo ekspertinio vertinimo anketa

Learning object types evaluation form

This is Description of 4 different selves. Ned Herrmann (1996) believes that thinking styles can be best described as a coalition of four different thinking selves. Left defines ratio, right defines emotion. In addition there are different layers: the outer layer stands for thinking (cognitive) and emotions (affective), the inner layer stands for instinct (conative). In the Whole Brain Model, these four different thinking selves are characterized as follows:



<ul style="list-style-type: none"> Lectured facts, details Research findings Higher order reasoning Critical thinking Textbooks, readings Case studies Use of experts Apply logic Metacognition Theories Thinking strategies <div style="text-align: center; border: 1px solid black; width: 40px; height: 40px; margin: 0 auto; background-color: white; color: blue; font-weight: bold; font-size: 24px;">A</div>	<ul style="list-style-type: none"> Holistic exercises (big picture) Mind mapping, synthesis Brainstorming Mental pictures Metaphors Active imagination Creativity Illustrations/pictures Pretending Painting/drawing Patterns/designs <div style="text-align: center; border: 1px solid black; width: 40px; height: 40px; margin: 0 auto; background-color: white; color: yellow; font-weight: bold; font-size: 24px;">D</div>
<ul style="list-style-type: none"> Outlining Graphic Organizer Checklists, Worksheets Number Sequences Policies, Procedures organization, Summaries Who, What Why, Where, When Exercises with steps Problem Solving with steps <div style="text-align: center; border: 1px solid black; width: 40px; height: 40px; margin: 0 auto; background-color: white; color: green; font-weight: bold; font-size: 24px;">B</div>	<ul style="list-style-type: none"> Cooperative learning Group discussions Role playing Drama/mime Body language Sharing personal experiences Listening and sharing ideas Storytelling Musical and rhythmic Interviews Physical activities <div style="text-align: center; border: 1px solid black; width: 40px; height: 40px; margin: 0 auto; background-color: white; color: red; font-weight: bold; font-size: 24px;">C</div>

A Quadrant: Analytical thinking.

- Key word: logical, factual, critical, technical and quantitative.
- Preferred activities: collecting data, listening to informational lectures, reading textbooks. Judging ideas based on facts, criteria and logical reasoning.

B Quadrant: Sequential thinking.

- Key word: conservative, structured, organised, detailed, and planned.
- Preferred activities: following directions, repetitive detailed homework problems, time management and schedules.

C Quadrant: Interpersonal thinking

- Key word: kinaesthetic, emotional, spiritual, sensory, feeling.
- Preferred activities: listening to and sharing ideas, looking for personal meaning, sensory input, and group study.

D Quadrant: Imaginative thinking.

- Key word: Visual, holistic, intuitive, innovative, and conceptual.
- Preferred activities: Looking at the big picture, taking initiative, simulations (what if questions), visual aids. Appreciate beauty of a problem, brainstorming.

Please evaluate each learning object type by marking 4 – if it is **very appropriate** for Quadrant in the column, 3 – if it is **appropriate**, 2 – if it is **good**, 1 – if it is **can be used**.

Code	Name in English	Name in Lithuanian	Quadrant			
			A	B	C	D
5.2.01.0	test	testas	1	1	1	1
5.2.04.0	demonstration	demonstracija	1	1	1	1
5.2.05.0	drill and practice	pratybos, praktinė užduotis	1	1	1	1
5.2.06.0	exploration	tyrinėjimas	1	1	1	1
5.2.07.0	simulation	modeliavimas (imitavimas)	1	1	1	1
5.2.10.1	learning asset/ audio	garso įrašas (komponentė)	1	1	1	1
5.2.10.2	learning asset/ image	paveikslas (komponentė)	1	1	1	1
5.2.10.3	learning asset/ video	vaizdo įrašas (komponentė)	1	1	1	1
5.2.21.0	project	projektas	1	1	1	1
5.2.26.0	glossary	glosarijus (aiškinamųjų terminų žodynas)	1	1	1	1
5.2.27.0	vocabulary	žodynas	1	1	1	1
5.2.28.0	reference	enciklopedinis šaltinis	1	1	1	1
5.2.29.0	presentation	pateiktis	1	1	1	1
5.2.30.1	web resource/ weblog	interneto tinklaraštis	1	1	1	1
5.2.30.2	web resource/ wiki	interneto vikis	1	1	1	1

Thank You for Helping!

6 PRIEDAS.

28 lentelė. MO atitikimo mokymosi stiliui ekspertinio vertinimo rezultatai

Kodas	Mokomojo objekto tipo pavadinimas	Pasirinkimų skaičius																							
		Kvadrantas A						Kvadrantas B						Kvadrantas C						Kvadrantas D					
		4	3	2	1	Vid	%	4	3	2	1	Vid	%	4	3	2	1	Vid	%	4	3	2	1	Vid	%
5.2.01.0	testas	7	4	4	3	2,8	71%	10	5	2	1	3,3	83%	0	1	6	11	1,4	36%	0	1	5	12	1,4	35%
5.2.04.0	demonstracija	4	8	3	3	2,7	68%	5	6	5	2	2,8	69%	2	8	2	6	2,3	58%	5	9	3	1	3	75%
5.2.05.0	pratybos, praktinė užduotis	4	8	3	3	2,7	68%	12	4	1	1	3,5	88%	1	2	5	10	1,7	42%	3	2	5	8	2	50%
5.2.06.0	tyrinėjimas	13	2	2	1	3,5	88%	4	5	3	6	2,4	60%	4	8	3	3	2,7	68%	10	3	2	3	3,1	78%
5.2.07.0	modeliavimas (imitavimas)	7	7	2	2	3,1	76%	4	5	3	6	2,4	60%	4	8	3	3	2,7	68%	11	3	2	2	3,3	82%
5.2.10.1	garso įrašas (komponentė)	4	3	6	5	2,3	58%	4	3	8	3	2,4	61%	6	6	3	3	2,8	71%	3	7	5	3	2,6	64%
5.2.10.2	paveikslas (komponentė)	3	6	4	5	2,4	60%	3	4	3	8	2,1	53%	3	8	3	4	2,6	64%	14	3	1	0	3,7	93%
5.2.10.3	vaizdo įrašas (komponentė)	2	7	4	5	2,3	58%	2	3	9	4	2,2	54%	6	9	2	1	3,1	78%	10	5	2	1	3,3	83%
5.2.21.0	projektas	7	5	4	2	2,9	74%	10	1	6	1	3,1	78%	11	4	2	1	3,4	85%	12	5	1	0	3,6	90%
5.2.26.0	glosarijus (aiškinamųjų terminų žodynas)	11	4	2	1	3,4	85%	8	8	1	1	3,3	82%	0	1	4	13	1,3	33%	0	0	6	12	1,3	33%
5.2.27.0	žodynas	9	4	4	1	3,2	79%	9	6	2	1	3,3	82%	0	0	5	13	1,3	32%	0	1	4	13	1,3	33%
5.2.28.0	enciklopedinis šaltinis	12	2	3	1	3,4	85%	7	8	2	1	3,2	79%	0	0	6	12	1,3	33%	1	1	6	10	1,6	40%
5.2.29.0	pateiktis	5	7	2	4	2,7	68%	5	7	4	2	2,8	71%	2	8	3	5	2,4	60%	4	6	5	3	2,6	65%
5.2.30.1	interneto tinklaraštis	4	5	4	5	2,4	61%	2	6	2	8	2,1	53%	7	5	3	3	2,9	72%	5	7	4	2	2,8	71%
5.2.30.2	interneto vikis	4	5	2	7	2,3	58%	6	2	5	5	2,5	63%	7	5	5	1	3	75%	6	7	3	2	2,9	74%