

KAUNO TECHNOLOGIJOS UNIVERSITETAS

RENATA BURBAITĖ

IŠPLĖSTINIAI GENERATYVINIAI  
MOKYMOSI OBJEKTAI INFORMATIKOS  
MOKYMUI SI: KONCEPCIJA, MODELIAI IR  
REALIZACIJA

Daktaro disertacija  
Fiziniai mokslai, Informatika (09P)

2014, Kaunas

Disertacija rengta 2010-2014 m. Kauno technologijos universiteto Informatikos fakultete, Programų inžinerijos katedroje.

**MOKSLINIS VADOVAS:**

Prof. habil. dr. Vytautas ŠTUIKYS (Kauno technologijos universitetas, fiziniai mokslai, informatika – 09P).

## **PADĖKA**

Nuoširdžiai dėkoju savo moksliniam vadovui prof. Vytautui Štuikiui už kantrybę, geranoriškumą ir nuolatinį skatinimą. Vadovo idėjos, patarimai ir diskusijos buvo neįkainojamos studijuojant ir rengiant šį darbą. Taip pat dėkoju prof. Robertui Damaševičiui už vertingas pastabas ir patarimus studijuojant bei rengiant publikacijas ir Kristinai Bespalovai už bendrą darbą darbo grupėje.

Esu dėkinga darbo recenzentams doc. Danguolei Rutkauskienei, prof. Valentinai Dagienei ir prof. Aleksandrui Targamadzei. Jų pastabos ir patarimai prisidėjo prie disertacijos kokybės.

Padėka Panevėžio Juozo Balčikonio gimnazijos administracijai, suteikusiai sąlygas suderinti studijas su darbine veikla, bei gimnazijos bendruomenei už geranoriškumą ir supratingumą.

Dėkoju savo šeimai už meilę, rūpestį, nuolatinį palaikymą ir supratingumą, bei mokiniam, kaip vienam iš svarbiausių įkvėpimo šaltinių.

## TURINYS

<b>PAVEIKSLŲ SĄRAŠAS.....</b>	<b>6</b>
<b>LENTELIŲ SĄRAŠAS .....</b>	<b>8</b>
<b>TERMINŲ IR SANTRUMPŲ ŽODYNAS.....</b>	<b>9</b>
<b>1. ĮVADAS.....</b>	<b>14</b>
1.1. DARBO AKTUALUMAS .....	14
1.2. DARBO OBJEKTAS .....	16
1.3. DARBO TIKSLAS .....	16
1.4. DARBO UŽDAVINIAI .....	16
1.5. GINAMIEJI TEIGINIAI .....	16
1.6. MOKSLINIS NAUJUMAS .....	16
1.7. PRAKTINIS NAUJUMAS .....	17
1.8. DARBO APIMTIS IR STRUKTŪRA.....	17
1.9. DARBO APROBAVIMAS .....	17
<b>2. MOKYMOŠI OBJEKTŲ SRITIES ANALIZĖ IR ĮVERTINIMAS .....</b>	<b>20</b>
2.1. ĮVADAS .....	20
2.2. E.MOKYMOŠI BRUOŽAI IR IŠŠŪKIAI .....	20
2.2.1. E.mokymosi technologijos ir sistemos.....	22
2.2.2. E.mokymosi sistemose naudojami mokymosi metodai.....	22
2.2.3. E.mokymosi naudojamų mokymosi objektų savybės.....	23
2.2.4. E.mokymosi standartai.....	24
2.2.5. E.mokymosi naudojamų technologijų tendencijos.....	26
2.3. SVARBIAUSI MOKYMOŠI OBJEKTŲ EVOLIUCIJOS ETAPAI.....	27
2.4. MOKYMOŠI OBJEKTŲ PAKARTOTINIO PANAUDOJIMO PROBLEMA E.MOKYMOŠI .....	28
2.4.1. Mokymosi objektų pakartotinio naudojimo aspektai būdingi komponentinei paradigmai.....	29
2.4.2. Mokymosi objektų pakartotinio panaudojimo modeliai .....	30
2.5. E.MOKYMOŠI SRITIES IR MOKYMOŠI OBJEKTŲ KONTEKSTO ANALIZĖ .....	31
2.6. MOKYMOŠI OBJEKTŲ MODELIAI.....	34
2.7. MOKYMOŠI OBJEKTŲ STANDARTAI IR SAUGYKLOS .....	36
2.8. MOKYMOŠI OBJEKTŲ VERTINIMO METRIKOS IR INSTRUMENTAI .....	38
2.9. BENDRIEJI INFORMATIKOS MOKYMOŠI IŠŠŪKIAI 21 AMŽIUJE .....	43
2.10. PROGRAMAVIMO MOKYMOŠI IŠŠŪKIAI.....	44
2.10.1. Motyvacijos modeliai .....	45
2.10.2. Mokymosi tikslų modelis .....	45
2.10.3. Programavimo mokymosi modeliai.....	46
2.10.4. Vertinimo modeliai.....	48
2.10.5. Mokymosi aplinkos ir įrankiai.....	49
2.10.6. Programavimo mokymuisi naudojami mokymosi objektai .....	53
2.11. APIBENDRINTAS TYRIMO KARKASAS.....	55
2.12. IŠVADOS.....	56

<b>3. INFORMATIKOS MOKYMOSSI SRITIES MODELIAVIMAS.....</b>	<b>57</b>
3.1. ĮVADAS.....	57
3.2. SRITIES ANALIZĖS BENDRINĖ SCHEMA, PRINCIPAI IR METODAI .....	57
3.3. MOKYMOSSI TURINIO IR PROCESŲ MODELIAVIMO PROBLEMA .....	60
3.4. SIŪLOMO MODELIAVIMO METODO TEORINĖS PRIELAIDOS .....	61
3.4.1. <i>Modeliavimo metodo sudarymo principai</i> .....	61
3.4.2. <i>Reikalavimai modeliavimo metodui</i> .....	62
3.4.3. <i>Informatikos mokymosi srities analizės metodai</i> .....	62
3.4.4. <i>Požymių diagramų notacija</i> .....	63
3.5. INFORMATIKOS MOKYMOSSI SRITIES MODELIAVIMO METODAS .....	66
3.6. POŽYMIŲ MODELIŲ SAVYBĖS .....	68
3.7. INFORMATIKOS MOKYMOSSI SRITIES POSRİČIŲ POŽYMIŲ MODELIAI .....	69
3.8. POŽYMIŲ PASIRINKIMAS FORMUOJANT KONKREČIUS POŽYMIŲ MODELIOUS .....	72
3.9. POŽYMIŲ MODELIŲ ANALIZĖ IR ĮVERTINIMAS .....	76
3.10. POŽYMIŲ DIAGRAMŲ TAIKymo INFORMATIKOS MOKYMOSSI SRITYJE GALIMYBĖS ....	77
3.11. SANTRAUKA IR APIBENDRINIMAS .....	78
3.12. IŠVADOS.....	79
<b>4. IŠPLĖSTINIŲ GENERATYVINIŲ MOKYMOSSI OBJEKTŲ KŪRIMO METODAS .....</b>	<b>80</b>
4.1. ĮVADAS.....	80
4.2. INFORMATIKOS MOKYMOSSI SRITIES VARIANTIŠKUMO PAGRINDINĖS SĄVOKOS .....	80
4.3. IŠPLĖSTINIO GENERATYVINIO MOKYMOSSI OBJEKTO REIKALAVIMŲ MODELIS .....	82
4.4. IŠPLĖSTINIO GENERATYVINIO MOKYMOSSI OBJEKTO SPECIFIKAVIMAS POŽYMIŲ DIAGRAMA .....	82
4.5. METAPROGRAMAVIMO TECHNOLOGIJA IR IŠPLĖSTINIAI GENERATYVINIAI MOKYMOSSI OBJEKTAI .....	85
4.6. METAPROGRAMAVIMO TECHNOLOGIJOS PAGRINDINĖS SĄVOKOS .....	86
4.7. IGMO AUKŠTO LYGMENS SPECIFIKACIJOS TRANSFORMAVIMO Į VYKDOMĄJĄ SPECIFIKACIJĄ TAISYKLĖS.....	88
4.8. IŠPLĖSTINIŲ GENERATYVINIŲ MOKYMOSSI OBJEKTŲ SAVYBĖS .....	88
4.9. IGMO TECHNOLOGINIO SUDĖTINGUMO ĮVERTINIMO METRIKOS.....	90
4.10. IGMO SUDĖTINGUMO VALDYMAS VYKDOMOSIOS SPECIFIKACIJOS LYGMENYJE .....	91
4.11. IŠVADOS.....	93
<b>5. IŠPLĖSTINIŲ GENERATYVINIŲ MOKYMOSSI OBJEKTŲ INTEGRAVIMAS Į SPECIALIZUOTAS MOKYMOSSI APLINKAS .....</b>	<b>94</b>
5.1. ĮVADAS.....	94
5.2. REIKALAVIMAI MOKYMOSSI APLINKOMS .....	94
5.3. SPECIALIZUOTOS HETEROGENINĖS MOKYMOSSI APLINKOS ARCHITEKTŪRA IR FUNKCIONALUMAS .....	96
5.4. BENDRADARBIUJANČIŲ ROBOTŲ APLINKA .....	98
5.5. MOKYMOSSI APLINKŲ KOKYBĖS VERTINIMO KRITERIJAI .....	99
5.6. IŠVADOS.....	102

<b>6. EKSPERIMENTINIS ĮVERTINIMAS .....</b>	<b>103</b>
6.1. ĮVADAS .....	103
6.2. SRITIES POŽYMIŲ MODELIŲ ĮVERTINIMAS .....	103
6.3. IGMO POŽYMIŲ MODELIŲ ĮVERTINIMAS .....	104
6.4. IGMO VYKDOMŲJŲ SPECIFIKACIJŲ TECHNOLOGINIS ĮVERTINIMAS .....	105
6.5. REALIOS SPECIALIZUOTOS MOKYMOSI APLINKOS SU INTEGRUOTAIS IGMO .....	106
6.5.1. <i>Mokymosi aplinka su vienu mokomuoju robotu .....</i>	<i>106</i>
6.5.2. <i>Mokymosi aplinka su bendradarbiaujančiais IGMO .....</i>	<i>110</i>
6.5.3. <i>Realių mokymosi aplinkų vertinimas technologiniu ir pedagoginiu požiūriu ..</i>	<i>114</i>
6.6. IGMO PAŽINIMO SUDĖTINGUMO ĮVERTINIMAS PEDAGOGINIŲ POŽIŪRIU .....	116
6.7. IGMO PEDAGOGINIS ĮVERTINIMAS .....	116
6.8. IŠVADOS.....	119
<b>7. BAIGIAMASIS ĮVERTINIMAS.....</b>	<b>120</b>
<b>IŠVADOS .....</b>	<b>127</b>
<b>LITERATŪRA.....</b>	<b>129</b>
<b>PRIEDAI .....</b>	<b>143</b>

## PAVEIKSLŲ SĄRAŠAS

2.1 pav. Esminiai mokymosi projektavimo proceso elementai ir jų koreliacija.....	23
2.2 pav. Faktoriai, turintys didžiausią teigiamą įtaką mokymosi išteklių naudojimui .....	24
2.3 pav. E.mokymosi standartų tipai ir tikslai .....	25
2.4 pav. Pavyzdinis standartų modelis.....	25
2.5 pav. Svarbiausi MO pakartotinio naudojimo aspektai.....	30
2.6 pav. Kontekstualizuoto mokymosi turinio pakartotinio panaudojimo proceso modelis .....	32
2.7 pav. Besimokančiojo, turinio ir konteksto sąryšiai mokymosi situacijoje .....	32
2.8 pav. MO struktūrinis modelis .....	34
2.9 pav. Bendras MO turinio modelis.....	35
2.10 pav. Šablonais grįsto GMO modeliai: a) struktūrinis, b) elgsenos .....	36
2.11 pav. IEEE MO standartų modelis .....	37
2.12 pav. Dublin Core MO standartų modelis.....	38
2.13 pav. Metrikos, apibrėžiančios MO kokybės kriterijus.....	39
2.14 pav. Sąryšiai tarp metrikų tipų.....	39
2.15 pav. Sąryšiai tarp pakartotinio naudojimo faktorių, metrikų ir LOM metaduomenų elementų.....	42
2.16 pav. Bloomo taksonomijos metalygmenų struktūra .....	46
2.17 pav. Mokymosi kelio duomenų modelis.....	47
2.18 pav. Žaidimais pagrįstų modelių mokymosi proceso schema .....	47
2.19 pav. Programavimo aplinkos: požymių aibės.....	51
2.20 pav. Programų vizualizacijos įrankių taksonomija .....	52
2.21 pav. Apibendrintas tyrimo karkasas .....	55
3.1 pav. Srities analizės bendrinė schema .....	58
3.2 pav. TPACK karkasas informatikos mokymosi sričiai.....	61
3.3 pav. Informatikos mokymosi srities modeliavimo metodo karkasas .....	66
3.4 pav. Informatikos mokymosi srities mokymosi tikslų modelis .....	69
3.5 pav. Informatikos mokymosi srities motyvacijos modelis .....	70
3.6 pav. Informatikos mokymosi srities mokymosi metodų modelis .....	70
3.7 pav. Informatikos mokymosi srities vertinimo modelis .....	71
3.8 pav. Besimokančiojo modelis, sudarytas remiantis standartizuotu konteksto modeliu.....	71
3.9 pav. Turinio modelis.....	72
3.10 pav. Technologinių aspektų modelis .....	72
3.11 pav. Konkretaus požymių modelio išgavimas iš abstrakčių modelių.....	74
3.12 pav. IGMO „Roboto tiesiaeigis judėjimas“ požymių modeliai .....	75
4.1 pav. Apibendrintas IGMO šeimos modelis .....	83
4.2 pav.: a) – IGMO prioritetų modelis, b) – IGMO konteksto modelis, c) IGMO „Nežinomo kartojimų skaičiaus ciklas“ turinio modelis, d) apibendrintas IGMO modelis .....	84
4.3 pav. IGMO modelis .....	86
4.4 pav. IGMO „Nežinomo kartojimų skaičiaus ciklas“ vykdomoji specifikacija...	89

4.5 pav. GMO/IGMO elgsenos modelis.....	89
4.6 pav. Dviejų pakopų IGMO modelis .....	92
5.1 pav. Informatikos mokymosi konceptualusis modelis.....	96
5.2 pav. Heterogeninė specializuota mokymosi aplinka: a) – bendroji struktūra, b) – aplinkos komponentų struktūra, c) aplinkos elgsenos modelis .....	97
5.3 pav. Roboto aplinkos konstravimo etapai.....	98
5.4 pav. Mokymosi aplinka su bendradarbiaujančiais robotais.....	99
6.1 pav. IGMO „Roboto kalibravimas“: a) – vartotojo sąsaja, b) – sugeneruotas egzempliorius .....	107
6.2 pav. Roboto greičio ir nuvažiuoto atstumo priklausomybės .....	108
6.3 pav. IGMO „Ornamentų piešimas“ vartotojo sąsajos fragmentas.....	109
6.4 pav. IGMO „Ornamentų piešimas“: a) sugeneruotas MO egzempliorius, b) MO įvykdymo rezultatas.....	109
6.5 pav. IGMO „Ornamentų piešimas“: a) pradinis sugeneruotas MO egzempliorius, b) MO egzempliorius mokiniams papildžius MO trūkstamais sakiniais, c) MO įvykdymo rezultatas.....	110
6.6 pav. Piešiantis ir liniją sekantis robotai .....	111
6.7 pav. Bendradarbiaujančių robotų sistemos architektūra.....	112
6.8 pav. Bendradarbiaujantys robotai .....	113
6.9 pav. Liniją sekančio roboto judėjimas elipsės ir stačiakampio formos trajektorija .....	113
6.10 pav. Linijos sekimo algoritmų tikslumo įvertinimas .....	114
6.11 pav. IGMO pedagoginis įvertinimas .....	117
6.12 pav. Mokinių įsitraukimo pedagoginis įvertinimas .....	118
7.1 pav. Metrikos, apibrėžiančios IGMO kokybės kriterijus.....	123
7.2 pav. Į specializuotas aplinkas integruotų IGMO ir programavimo modulių turinio sąryšiai.....	124



## LENTELIŲ SĄRAŠAS

2.1 lentelė Mokymosi išteklių komponentų tipai .....	24
2.2 lentelė E.mokymosi, M.mokymosi ir U.mokymosi pagrindinės savybės .....	26
2.3 lentelė MO pakartotinio panaudojimo modelių lyginamoji analizė .....	31
2.4 lentelė Konteksto parametrai .....	33
2.5 lentelė Sąryšiai tarp MO tipų, granuliacijos, agregacijos lygių ir pakartotinio naudojimo .....	35
2.6 lentelė Mobilaus mokymosi kokybės metrikos .....	40
2.7 lentelė Motyvacijos faktoriai .....	45
2.8 lentelė Nagrinėtų modelių ir programavimo mokymosi iššūkių sąryšiai .....	48
2.9 lentelė Programavimo mokymosi sistemų atributai .....	50
3.1 lentelė Požymių diagramų grafinė ir formali notacija .....	64
3.2 lentelė FAMILIAR ir SPLOT požymių modelių notacija .....	65
3.3 lentelė IGMO „Roboto tiesiaiegis judėjimas“ požymių modelių kokybės parametrai .....	77
4.1 lentelė Vienos pakopos IGMO „Nežinomo kartojimų skaičiaus ciklas“ technologinio sudėtingumo metrikos .....	91
4.2 lentelė Dviejų pakopų IGMO „Nežinomo kartojimų skaičiaus ciklas“ technologinio sudėtingumo metrikos .....	93
5.1 lentelė Reikalavimai specializuotoms mokymosi aplinkoms su IGMO .....	95
5.2 lentelė Mokymosi aplinkų vertinimo kriterijai .....	100
5.3 lentelė Mokymosi funkcijų tipai .....	101
5.4 lentelė Sumodeliuotų heterogeninės specializuotos ir bendradarbiaujančių robotų aplinkų kokybės įvertinimas technologiniu požiūriu .....	101
5.5 lentelė Sumodeliuotų mokymosi aplinkų kokybės vertinimas pedagoginiu požiūriu .....	102
6.1 lentelė Informatikos mokymosi srities posričių požymių modelių kokybės parametrai .....	104
6.2 lentelė IGMO požymių modelių kokybės parametrai .....	105
6.3 lentelė IGMO vykdomųjų specifikacijų technologinio sudėtingumo įvertinimas .....	106
6.4 lentelė Sukurtų mokymosi aplinkų kokybės įvertinimas technologiniu požiūriu .....	115
6.5 lentelė Sukurtų mokymosi aplinkų kokybės įvertinimas pedagoginiu požiūriu .....	115
6.6 lentelė IGMO pažinimo sudėtingumo vertinimas pedagoginiu požiūriu .....	116
7.1 lentelė Įvairių MO tipų atributų lyginamoji analizė .....	120
7.2 lentelė IEEE LOM standarto pagrindiniai metaduomenų elementai, aprašantys IGMO bendrąsias ir pedagogines charakteristikas .....	122

## TERMINŲ IR SANTRUMPŲ ŽODYNAS

<b>Abstrakcijos lygmenys</b>	Kokybinis detalumo laipsnio matas. Kuo mažesnis detalumo laipsnis, tuo aukštesnis abstrakcijos laipsnis.
<b>Abstraktus požymių modelis (angl. <i>Abstract Feature Model</i>)</b>	Požymių modelis, turintis požymių, kurie kitame kontekste gali būti išskaidyti į smulkesnius.
<b>Adaptyvi mokymosi aplinka (angl. <i>Adaptive Learning Environment</i>)</b>	Mokymosi aplinka, kurioje mokinių veiklos interpretuojamos remiantis specifiniais srities modeliais ir besimokančiojo poreikiais [Nat12].
<b>Agreguotas mokymosi objektas (angl. <i>Aggregated Learning Object</i>)</b>	Mokymosi objektai, kurie sugrupuoti į didesnius mokymosi turinio rinkinius, įskaitant tradicinės struktūros kursus [NSP+12], [Nat12].
<b>Atominiis požymis (angl. <i>Atomic Feature</i>)</b>	Požymis, kuris duotame kontekste neskaidomas į smulkesnius.
<b>Bendradarbiaujantys išplėstiniai generatyviniai mokymosi objektai (angl. <i>Advanced Generative Collaborative Learning Objects</i>)</b>	Išplėstiniai generatyviniai mokymosi objektai, integruoti į mokomaisiais robotais grįstas mokymosi aplinkas ir palaikantys darbo grupėmis koncepciją.
<b>E.mokymasis (angl. <i>e-learning</i>)</b>	Mokymasis, kai jo efektyvumui didinti naudojamos informacinės ir komunikacinės technologijos. Gali būti įvairių modelių: nuo tradicinio mokymosi, papildyto informacinėmis ir komunikacinėmis technologijomis, iki visiškai pagrįsto informacinėmis ir komunikacinėmis technologijomis ir vykstančio virtualiai sukurtoje aplinkoje. E.mokymasis užtikrina mokymosi interaktyvumą ir besimokančiojo aktyvumą [DGJ08].
<b>Generatyvinė lingvistika (angl. <i>Generative Linguistics</i>)</b>	Kalbos aprašymas formaliais modeliais, kurių pagrindą sudaro transformacinės generatyvinės gramatikos. Išskiriami kalbos mokėjimo ir taikymo sluoksniai. Generatyvinės gramatikos apima kalbos mokėjimą ir turi giluminę ir paviršinę sintaksines struktūras. Generatyvinės lingvistikos principus pritaikė generatyvinių mokymosi objektų pradininkai Boyle ir jo kolegos. Transformacinėmis vadinamos generatyvinės gramatikos, turinčios rinkinį taisyklių, kurios veikia tada, kai jau nustatytos giluminės sintaksinės struktūros.

**Generatyvinė technologija**  
(angl. *Generative Technology*)

Technologija, įgalinanti automatiškai (arba pusiau automatiškai) kurti programų kodo arba aukštesnio lygmens specifikacijas.

**Generatyvinis mokymosi objektas (GMO, angl. *Generative Learning Object, GLO*)**

Pakartotinai naudojamas mokymosi objektas, turintis daugiapakopę struktūrą ir realizuojamas naudojant generatyvinę technologiją (adaptuotas iš Boyle *ir kt.* darbų).

Apibendrintas (parametrizuotas) mokymosi objektas, realizuotas panaudojus tam tikrą technologiją (pvz., metaprogramavimą) ir pasižymintis pertekliniu funkcionalumu, kad būtų galimas automatinis mokymosi objektų egzempliorių generavimas pagal vartotojo pasirinktus parametrus [Rup09].

Generatyvinis mokymosi objektas yra aiškus vykdomasis mokymosi kūrinys (projektas), kuris sukuria tam tikrą mokymosi objektų klasę [CETL14].

**Generavimas**

Automatinis mokymosi objektų egzempliorių kūrimas iš išplėstinių generatyvinių mokymosi objektų.

**Heterogeninis metaprogramavimas**

Paradigma, kai rašant metaprogramas naudojamos mažiausiai dvi nepriklausomos kalbos. Tikslo kalba *T* (angl. *target language*) yra žemesnio lygio kalba, kuri išreiškia bazinį srities funkcionalumą. Aukštesnio lygio kalba, vadinama metakalba *M* (angl. *meta-language*) per parametrus, aprašančius srities variantiškumą, užrašomas bendrinimo algoritmas.

**HTML (angl. *Hypertext Markup Language*)**

Hiperteksto ženklinimo ir tinklalapių kūrimo kalba. (Enciklopedinis kompiuterijos žodynas, <http://aldona.mii.lt/pms/terminai/term/enc.html>).

**Informatikos mokymasis (IM)**

ACM siūlo taikyti informatikos mokymosi programos vidurinėje mokykloje modelį, kuriame *informatika* apibrėžiama kaip mokslas, nagrinėjantis kompiuterių ir algoritminius procesus, techninės ir programinės įrangos projektavimą, taikymus ir įtaką visuomenei. [EAG+08]. Plačiau žr. 2.9 skyrelį. (Pabraukti terminai perdengia IM sampratą šiame darbe)

**Instrukcinė inžinerija (angl. *Instructional Engineering*, Paquette 2003)**

**Išplėstinio generatyvinio mokymosi objekto požymių modelis**

**Išplėstinis generatyvinis mokymosi objektas (IGMO, angl. *advanced generative learning object*, AGLO)**

**Kompiuterinis mąstymas (angl. *Computational Thinking*)**

**Konkretus požymių modelis (angl. *Concrete Feature Model*)**  
**Konteksto modeliavimas**

**Metaduomenys (angl. *Metadata*)**

Metodas, palaikantis mokymosi sistemos analizę, projektavimą ir pateikimo planavimą, bei integruojantis mokymosi projektavimo, programų inžinerijos ir pažinimo inžinerijos konceptus, principus ir procesus.

Modelis, sudarytas iš konteksto ir turinio požymių modelių, kuriuos semantiškai susieja prioritetų požymių modelis ir apribojimai bei sąryšiai tarp konteksto ir turinio modelių požymių.

*Semantiškai* IGMO yra mokymosi variantiškumo atvaizdavimo į generatyvinę technologiją produktas, palaikantis iš anksto aprašytų požymių realizaciją.

*Struktūriškai* IGMO sudaro iš anksto specifikuotų automatiškai pagal vartotojo poreikius generuojamų mokymosi objektų egzempliorių aibė arba konkretus egzempliorius.

*Elgsenos* požiūriu IGMO yra generatorius, generuojantis mokymosi objekto egzempliorius pagal vartotojo reikalavimus (vartotojui pasirinkus reikiamas parametrų reikšmes).

*Technologijos taikymo* programavimo mokymesi požiūriu IGMO yra traktuojamas kaip metaprograma. Mokymosi objektas, t.y. mokymosi turinys, yra tikslo kalba parašyta programa.

Gebėjimas analizuoti, abstrahuoti ir automatizuoti sprendžiant realaus gyvenimo uždavinius [IIT+12].

Požymių modelis išgaunamas iš abstrakčių požymių aibės, pasirenkant adekvačius lapus.

Esminės informacijos apie kontekstą išskyrimas ir pateikimas tolesniam apdorojimui [SB08]. Neišreikštinė konteksto vaizdavimo forma yra tokia, kai vartotojas tiesiogiai nepasirenka konteksto parametrų reikšmių. Išreikštinė konteksto vaizdavimo forma leidžia vartotojui tiesiogiai valdyti kontekstą, pasirenkant norimas konteksto parametrų reikšmes.

Struktūrizuota informacija apie MO, kuri aprašo, paaikškina, fiksuoja ar kitais būdais palengvina mokomųjų išteklių paiešką, naudojimą ir valdymą [BC10].

<b>Metaprogramavimas</b>	Aukšto lygmens programavimo paradigma, kai generatyvinis pakartotinis panaudojimas užtikrinamas sukuriant apibendrintas programas, vadinamas metaprogramomis.
<b>Metodas</b>	Pažinimo ir veikimo taisyklių, priemonių ir būdų visuma tam tikram uždaviniui išspręsti.
<b>Metodika</b>	Sisteminis metodų taikymas, apimantis metodų parengimą ir/ar parinkimą, jų įgyvendinimą ir panaudojimo nuoseklumą.
<b>Modeliavimas</b>	Modelių kūrimas ir jų analizės procesas.
<b>Modelis</b>	Sistemos esybių (objektų), procesų esminių savybių pavaizdavimas (pateikimas). [MFB+12], [DGJ08].
<b>Mokymosi aplinka (angl. <i>Teaching/Learning Environment</i>)</b>	Technologijomis pagrįsta ugdymo sistema, kuri padeda siekti apibrėžtų mokymosi tikslų naudojant specifinį mokymosi turinį [DSV04] ir apima mokymosi išteklius (mokymosi turinį ir įrankius), sąveiką ir bendradarbiavimą tarp mokymosi proceso dalyvių, mokymosi veiklas ir mokymosi palaikymo veiklas [Lom08].
<b>Mokymosi objekto kokybės metrika</b>	Mokymosi objekto atributo kokybės lygį nurodantis matas.
<b>Mokymosi kontekstas</b>	Informacija, identifikuojanti esybės būseną, t.y. besimokančiojo vietą, mokymosi veiklas naudojamus įrankius ir MO [LCW+09].
<b>Mokymosi objektas (MO, angl. <i>Learning Object, LO</i>)</b>	Bet kokia skaitmeninė esybė, kuri gali būti naudojama, pakartotinai naudojama arba nurodoma (angl. <i>referenced</i> ) e.mokymosi (adaptuota iš [LTSC02]).
<b>Mokymosi objektų saugykla</b>	Skaitmeninė biblioteka, kurioje saugomi meta-duomenimis aprašyti mokymosi ištekliai [LTSC02].
<b>Mokymosi projektavimas (angl. <i>Instructional/ Learning design</i>)</b>	Pedagoginio modelio taikymas specifiniam mokymosi tikslui, tikslinei grupei ir specifiniam kontekstui ar pažinimo sričiai [KGR06].
<b>Mokymosi variantiškumas (angl. <i>Learning Variability</i>)</b>	Pedagoginio, socialinio ir turinio variantiškumo kompozicija.
<b>Mokymosi valdymo sistema (angl. <i>Learning Management System</i>)</b>	E.mokymosi sistema, kuri pristato mokymosi modulius, užtikrina besimokančiųjų prieigos kontrolę, atlieka mokinių veiklų pažangos stebėseną [Nat12].

**Pakartotinai panaudojami mokymosi objektai (angl. *Reusable Learning Objects*)**

**PHP (angl. *Hypertext Preprocessor*)**

**Požymis**

**Problemų sritis (angl. *Problem Domain*)**

**Programavimo mokymosi modelis**

**Semantinis tankis (angl. *Semantic Density*)**

**Sprendimų sritis (angl. *Solution Domain*)**

**TRUC (angl. *Testable, Reusable Unit of Cognition*) [Mey06]**

**Užduočių sritis (angl. *Task Domain*)**

**Variantas**

**Variantinis taškas**

**Variantiškumo modeliavimas**

Maži informacijos vienetai (angl. *information chunks*), kurie pakartotinai panaudojami įvairiuose kontekstuose [NSP+12], [Nat12].

Interpretuojamoji kalba scenarijams, vykdomiems serveryje, dinaminiam žiniatinklio turiniui (parametruotiems dinaminiam tinklalapiams), o pastaruoju metu ir įvairioms taikomosioms programoms kurti (Enciklopedinis kompiuterijos žodynas, <http://aldona.mii.lt/pms/terminai/term/enc.html>). Vartotojui matoma srities charakteristika [KCH+90], kokybinė koncepto savybė arba funkcinis reikalavimas [CE00].

Disertacijos kontekste tai yra sritis, apimanti informatikos mokymąsi, pasižymintį mokymosi variantiškumu.

Konceptuali struktūra, aprašanti, kokios sisteminės procedūros turi būti atliekamos organizuojant mokymosi procesą, kad būtų pasiekti specifiniai programavimo mokymosi tikslai.

MO glaustumo laipsnis: MO informacijos kiekio ir jo dydžio arba veikimo trukmės santykis [BC10].

Sprendimų sritimi disertacijos kontekste suprantama technologija, kuri naudojama problemų srities įgyvendinimui.

Esybė, kurią sudaro konceptų aibė, praktiniai įgūdžiai ir vertinimo kriterijai, skirti žinių apie sritį ir jos mokymosi sistemos sudarymui [Mey06].

Disertacijos kontekste užduočių sritis yra programavimo mokymasis.

Variantu vadinamas požymis, kuris duotame kontekste neskaidomas į smulkesnius požymius.

Neprivalomų arba alternatyviųjų požymių grupių tėvinis požymis.

Esybių kintamų ir pastovių charakteristikų ir sąryšių bei apribojimų tarp jų modeliavimas taikant įvairias metodologijas.

# 1. ĮVADAS

## 1.1. Darbo aktualumas

Pastaraisiais metais e.mokymosi srities tyrimai labai intensyvūs. Jie apima plačią disciplinų, metodų, technologijų ir procesų erdvę. Tuose tyrimuose centrinę vietą užima mokymosi turinys. E.mokymosi sistemose nepriklausomas ir savarankiškas mokymosi turinio vienetas apibrėžiamas kaip mokymosi objektas (MO). Platesniame kontekste MO suprantamas kaip abstrakcija arba modelis, palaikantis pakartotinį panaudojimą tarp daugelio e.mokymosi bendruomenių [Lib05]. Lankstaus mokymosi turinio kūrimas, atnaujinimas ir efektyvus taikymas išlieka vienu iš didžiausių iššūkių e.mokymosi tyrimuose.

Pradedant XX a. paskutiniu dešimtmečiu, kai 1994 m. Hodgins pasiūlė šį terminą, susidomėjimas MO e.mokymesi nuolat didėja, nes sritis apima platų įrankių, metodologijų, technologijų ir standartų spektrą. Turint abstrakčią bendrinę sąvoką galima vienareikšmiškai aprašyti, pateikti ir keistis informacija. Pvz., galima aprašyti technologines operacijas su mokomaisiais kursais: išskaidyti temas į dalis, agreguoti temas iš sudėtinių dalių, nustatyti tam tikrą tų dalių seką ir panašiai. Mokymosi turinį galima modeliuoti susiejant jį su pedagoginiais ir socialiniais atributais, t.y. mokymosi teorijomis, mokinių psichologinėmis ypatybėmis ir kt. Be to, neturint bendrinio termino (t.y. mokymosi objekto), būtų neįmanoma plėtoti ir kurti e.mokymosi teorijų, lyginti e.mokymosi rezultatų, keistis moksline informacija bei praktine patirtimi. Taigi termino metodologinė ir mokslinė reikšmė didžiulė.

Su MO siejamas dar vienas labai svarbus terminas – *metaduomenys*. Metaduomenys apibrėžiami kaip struktūrizuota informacija apie MO, kuri aprašo, paaiškina, fiksuoja ar kitais būdais palengvina mokomųjų išteklių paiešką, naudojimą ir valdymą [BC10]. Vartojant *mokymosi objekto* ir *metaduomenų* sąvokas galima nagrinėti ir suprasti įvairias metaduomenų standartizavimo, išorinių mokymosi objektų saugyklų kūrimo, mokymosi objektų taksonomijų sudarymo ir kitas iniciatyvas [McG04]. Galima tvirtinti, kad MO – tai e.mokymosi centrinė ašis. MO yra kuriami, konstruojami ir saugomi išorinėse ar vidinėse saugyklose, kontekstualizuojami ir standartizuojami; egzistuoja įvairūs MO profiliai ir modeliai, įvairūs taikymai pradedant semantiniu tinklu baigiant mokomosiomis modeliavimo kalbomis (angl. *EML – Educational Modelling Languages*) ir instrukcine inžinerija (angl. *Instructional Engineering*) [McG04].

Analizė rodo, kad tyrimai apie MO e.mokymesi sudaro atskirą šaką, kuri vis plečiama ir tobulinama. MO naudotojų sąrašas yra labai platus: mokytojai, mokiniai, tyrėjai, kursų projektuotojai, mokslininkų ir organizacijų grupės ir pan.

Lietuvoje 2009-2013 m. apgintose disertacijose nagrinėjami aktyviųjų [Slo09], lanksčiai pritaikomų [Kub09] MO kūrimo metodai, sukurtas MO metaduomenų taikomasis modelis [Kub12] ir MO kokybės ekspertinio vertinimo metodas [Sēr13].

Reikšmingų pokyčių į MO sritį įnešė generatyvinio mokymosi objekto (angl. *generative learning object, GMO*) koncepcija, kurią pasiūlė Boyle su kolegomis [LBM+04]. Šie autoriai kildina GMO iš *generatyvinės lingvistikos* ir sieja su MO pakartotinio panaudojimo išplėtimu [CETL14]. Pastarajame šaltinyje pateikiamas toks GMO apibrėžimas: „GMO yra aiškus vykdomasis mokymosi kūrinys

(projektas), kuris sukuria tam tikrą mokymosi objektų klasę“ (“*An articulated and executable learning design that produces a class of learning objects*”). GMO koncepcijos pionieriai apibūdina GMO kaip naujos kartos MO [LBM+04].

Lietuvoje GMO koncepciją pradėta vystyti 2007 m. KTU mokslinio vadovo, prof. V. Štūikio mokslo grupėje. Nežiūrint to, kad per 2007-2010 m. laikotarpį čia buvo paskelbta daug straipsnių ir apginta disertacija [Rup09], su GMO susiję daug neišspręstų arba nepilnai išspręstų problemų (sistemizavimas, aukšto lygmens GMO modelių sudarymas, kūrimo automatizavimas, GMO perkeliamumas į įvairias aplinkas, realus pritaikymas mokymosi procese, įvertinimo problemos ir kt.), kurios buvo išnagrinėtos nepilnai ar visai nenagrinėtos.

Darbo tyrimo objektas – „išplėstiniai generatyviniai mokymosi objektai“. Terminas „išplėstiniai“ suprantamas kaip generatyvinių mokymosi objektų naujų pakartotinio panaudojimo dimensijų e.mokymesi plėtimas ir tobulinimas įvertinant ir integruojant pedagoginius, socialinius ir technologinius mokymosi aspektus. Mūsų nagrinėjami išplėstiniai GMO kildinami (technologiniu požiūriu) iš generatyvinės technologijos (ja laikoma heterogeninio metaprogramavimo technologija [ŠD13]). Ši technologija pasižymi tuo, kad per parametrizavimą galima unifikuotai išreikšti visus su mokymusi susijusius aspektus (turinio, pedagoginius, socialinius, technologinius). Nors metaprogramavimu grindžiami (specifikuojami) GMO iš esmės nepriklauso nuo mokomosios medžiagos, vis dėlto ir conceptualiai, ir praktiškai šio tipo GMO geriausiai tinka informatikos (programavimo) mokymuisi, kadangi automatiškai generuojamas mokymosi turinys yra programos arba jų dalys. Nors informatikos mokymasis turi daug bendrų bruožų su e.mokymosi tyrimais apskritai, tačiau dėl turinio, metodų, aplinkų ir kt. specifiškumo, informatikos mokymasis, kaip rodo atlikta apžvalga, sudaro savarankišką sritį. Darbe naudojamas ACM pasiūlytas informatikos mokymosi programos vidurinėje mokykloje modelis, kuriame *informatika* apibrėžiama kaip mokslas, nagrinėjantis kompiuterių ir algoritminius procesus, techninės ir programinės įrangos projektavimą, taikymus ir įtaką visuomenei [EAG+08] (pabraukti terminai perdengia informatikos mokymosi sampratą šiame darbe). Programavimo mokymosi terminas įvedamas apibrėžus informatikos mokymosi srities ribas.

Pastaraisiais metais iškeltos naujos problemos, susijusios su MO taikymu e.mokymesi apskritai ir informatikos mokymesi konkrečiai: 1) didinti pateikiamą informaciją vizualizuojančių mokomųjų išteklių skaičių [SDT12]; 2) kurti MO, užtikrinančius personalizuotą e.mokymąsi [RAH+09]; 3) adaptuoti mokymosi kontekstą pagal besimokančiojo poreikius: (i) gerinti turinio kontekstą, (ii) užtikrinti adaptyvų mokymosi procesą (angl. *Adaptive Learning Process*) [AAB+06], (iii) įvesti MO kontekstą, kuris apibrėžiamas kaip unikali tarpusavyje susijusių duomenų, charakterizuojančių konkrečią mokymosi situaciją, aibę [JGK+07].

Kita vertus, šaltinių analizė parodė, kad e.mokymosi srities technologinių sprendimų metodologija yra glaudžiai susijusi su informatikos mokslu. Tchounikine [Tch11] apibrėžė informatikos mokslo svarbą e.mokymosi tyrimuose trimis aspektais: 1) naujų technologijų kūrimas ar jau esančių pritaikymas sukuria naujas žmogaus veiklos galimybes; 2) plėtoja abstrakcijas; 3) įgalina vykdyti detalizuotus



modelius ir procesus automatizuotai. E.mokymasis laikomas išmaniųjų taikymų sritimi, kurioje išskyla ir sprendžiamos informatikos mokslo problemos (modelių specifikavimo, realizacijos).

Pateiktas pagrindimas ir apibrėžtos problemos leidžia teigti, kad tyrimo tema yra aktuali.

## **1.2. Darbo objektas**

Darbe tiriami informatikos (programavimo) mokymuisi skirti išplėstiniai generatyviniai mokymosi objektai (IGMO) ir su jais susiję informaciniai specifikavimo/atvaizdavimo, transformavimo modeliai ir procesai.

## **1.3. Darbo tikslas**

Darbo tikslas yra pateikti ir iširti metodiką, įgalinančią formalizuoti išplėstinių generatyvinių mokymosi objektų kūrimą ir efektyvų jų naudojimą mokant informatikos (programavimo).

Iškeltam tikslui pasiekti sprendžiami tokie uždaviniai.

## **1.4. Darbo uždaviniai**

1. Atlikti mokymosi objektų mokslinių tyrimų analizę bendrajame e.mokymosi ir informatikos mokymosi kontekstuose.

2. Modeliuoti programavimo mokymosi sritį sukuriant požymiais grindžiamus bendrinius modelius, iš kurių išgaunami konkretūs išplėstinių generatyvinių mokymosi objektų modeliai.

3. Formalizuoti išplėstinių generatyvinių mokymosi objektų specifikavimą ir kūrimą.

4. Sukurti specializuotas heterogenines mokymosi aplinkas ir į jas integruoti išplėstinius generatyvinius mokymosi objektus.

5. Eksperimentiškai įvertinti sukurtos metodikos panaudą pritaikant technologinius ir pedagoginius kriterijus.

## **1.5. Ginamieji teiginiai**

1. Informatikos mokymosi srities variantiškumo koncepcija – IGMO metodologinis pagrindas.

2. Požymiais grindžiami modeliai įgyvendina mokymosi variantiškumo koncepciją.

3. Dviejų lygmenų IGMO modelių vykdomosios specifikacijos užtikrina automatinį turinio kūrimą.

4. Specializuotos heterogeninės mokomaisiais robotais grindžiamos mokymosi aplinkos sudaro sąlygas efektyviai panaudoti IGMO.

## **1.6. Mokslinis naujumas**

1. Išplėstiniai generatyviniai mokymosi objektai išplečia informatikos mokymosi sritį naujais aspektais (pedagoginiais, socialiniais, technologiniais, turinio), aprašomais terminu *mokymosi variantiškumas*. Tai įgalino *pagrįstai*

*adaptuoti ir naujai pritaikyti* programų inžinerijos ir kompiuterijos principus ir metodus e.mokymosi sričiai.

2. Požymiais grįstas *sisteminis* informatikos (programavimo) mokymosi srities modeliavimas, mūsų žiniomis, atliktas pirmą kartą. Jis įvertina mokymosi variantiškumą ir agreguoja bei verifikuoja įvairialypius modelius (tikslų, motyvacijos, metodų, mokinio profilio, turinio ir kt.). Tai sudaro prielaidas sistemingam IGMO kūrimui.

3. Modelių formalizavimas dviejuose lygmenyse (požymių modelių ir vykdomųjų specifikacijų) sudaro sąlygas automatizuotiems įrankiams kurti.

4. Išplėstiniai generatyviniai mokymosi objektai išplečia pakartotinio panaudojimo koncepciją e.mokymosi turinio automatinio kūrimo požiūriu.

### **1.7. Praktinis naujumas**

1. Sukurta specializuota heterogeninė mokymosi aplinkos architektūra, grindžiama mokomaisiais robotais ir mikrovaldikliais.

2. Sukurti išplėstiniai generatyviniai informatikos (programavimo) mokymosi objektai, realizuojantys fizinę programų elgsenos vizualizaciją.

3. Išplėstiniai generatyviniai mokymosi objektai integruoti į realų ugdymo procesą, realizuoja tarpdalykinius mokymosi aspektus, žinomus kaip STEM (angl. *Science, Technology, Engineering, Mathematics*).

4. Sudaryta metodika *palaiko galimybes* integruoti išplėstinius generatyvinius mokymosi objektus ir procesus į plačiai naudojamas e.mokymosi valdymo sistemas.

5. Metodika įvertinta taikant žinomus pedagoginius ir technologinius vertinimo kriterijus, o eksperimentinių tyrimų 2011-2014 m. surinkta statistika įgalina tvirtinti, kad metodika yra efektyvi.

### **1.8. Darbo apimtis ir struktūra**

Darbą sudaro: terminų ir santrumpų žodynas, šeši pagrindiniai skyriai, išvados ir priedai. Apimtis be priedų – 140 puslapių, panaudoti 57 paveikslėliai ir 27 lentelės, cituojami 223 literatūros šaltiniai.

### **1.9. Darbo aprobavimas**

Darbas aprobuotas trijuose recenzuojamuose periodiniuose žurnaluose (į ketvirtą žurnalą priimtas) ir šešiose recenzuojamose konferencijose (vienoje pasaulinėje, keturiuose tarptautinėse, vienoje nacionalinėje) bei trijuose tarptautiniuose doktorantų konsorciujuose. Iš viso dešimt publikacijų.

*Straipsniai mokslinės informacijos instituto duomenų bazės „ISI Web of Science“ leidiniuose, turinčiuose citavimo indeksą:*

1. Burbaite, R., Stuikeys, V., & Marcinkevicius, R. The LEGO NXT Robot-based e.Learning Environment to Teach Computer Science Topics. *Electronics & Electrical Engineering*. ISSN 1392-1215. 2012, 18(9), p. 113-116. [ISI Web of Science; INSPEC; Computers & Applied Sciences Complete; Central & Eastern European Academic Source]

2. Štuikys, V., Bespalova, K., & Burbaitė, R. Refactoring of Heterogeneous Meta-Program into k-stage Meta-Program. *Information Technology And Control*. ISSN 1392-124X. 2014, 43(1), p. 14-27. [ISI Web of Science; INSPEC]

3. Burbaitė, R., Bespalova, K., Damaševičius, R., & Štuikys, V. Context-Aware Generative Learning Objects for Teaching Computer Science. Priimtas į žurnalą *International Journal of Engineering Education*. [ISI Web of Science; Scopus]

*Kitų tarptautinių duomenų bazių leidiniuose:*

1. Burbaite, R., & Stuikeys, V. Analysis of Learning Object Research Using Feature-based Models. Information Technologies' 2011: proceedings of the 17th international conference on Information and Software Technologies, IT 2011, Kaunas, Lithuania, April 27-29, 2011 / Edited by R. Butleris, R. Butkiene ; Kaunas University of Technology. Kaunas: Technologija. ISSN 2029-0020. 2011. p. 201-208. [Conference Proceedings Citation Index]

2. Burbaite, R., Damasevicius, R., Stuikeys, V., Bespalova, K., & Paskevicius, P. Product variation sequence modelling using feature diagrams and modal logic. CINTI 2011 [elektroninis išteklius]: 12th IEEE International Symposium on Computational Intelligence and Informatics, November 21-22, 2011, Budapest, Hungary: proceedings. Budapest: IEEE, 2011. ISBN 9781457700439. p. 73-77. [IEEE/IEE]

3. Štuikys, V., & Burbaite, R. Two-stage generative learning objects. Information and software technologies: 18th International Conference, ICIST 2012, Kaunas, Lithuania, September 13-14, 2012: proceedings / [edited by] Tomas Skersys, Rimantas Butleris, Rita Butkiene. Berlin, Heidelberg: Springer, 2012. ISBN 9783642333071. p. 332-347. [Conference Proceedings Citation Index]

4. Štuikys, V., Burbaitė, R., & Damaševičius, R. Teaching of Computer Science Topics Using Meta-Programming-Based GLOs and LEGO Robots. *Informatics in Education-An International Journal*. ISSN 1648-5831. 2013, Vol. 12, p. 125-142.[ INSPEC; CEEOL]

5. Burbaite, R., Stuikeys, V., & Damasevicius, R. Educational robots as collaborative learning objects for teaching Computer Science. ICSSE 2013: IEEE International Conference on System Science and Engineering, 4-6 July, 2013, Budapest, Hungary: proceedings. Piscataway: IEEE, 2013. ISBN 9781479900077. p. 211-216. [IEEE/IEE]

*Straipsniai kituose recenzuojamuose mokslo leidiniuose (konferencijų medžiaga):*

1. Burbaitė, R., & Štuikys, V. Mokymosi objektų pakartotinės panaudos modelių analizė. Informacinės technologijos: 16-oji tarpuniversitetinė magistrantų ir doktorantų konferencija: konferencijos pranešimų medžiaga / Kauno technologijos universitetas, Vytauto Didžiojo universitetas, Vilniaus universiteto Kauno humanitarinis fakultetas. Kaunas: Technologija. ISSN 2029-249X. 2011. p. 57-60.

2. Burbaitė, R., Damaševičius, R., & Štuikys, V. Using Robots as Learning Objects for Teaching Computer Science. WCCE 2013: 10th IFIP World Conference on Computers in Education, July 1-7, 2013, Torun, Poland. Vol. 1. Torun: Nicolaus Copernicus University Press, 2013. ISBN 9788323130901. p. 101-111.

*Dalyvavimas tarptautiniuose doktorantų konsorciumuose:*

1. 2rd International Doctoral Consortium on Informatics Engineering Education Research: Methodologies, Methods, and Practice, organized by Vilnius University Institute of Mathematics and Informatics on November 3-December 4, 2011 in Druskininkai, Lithuania.

2. 3rd International Doctoral Consortium on Informatics Engineering Education Research: Methodologies, Methods, and Practice, organized by Vilnius University Institute of Mathematics and Informatics on December 3-7, 2012 in Druskininkai, Lithuania.

3. Doctoral Consortium on Informatics Education Research, organized by Nicolaus Copernicus University of Torun and Information Processing Society of Poland in cooperation with IFIP TC 3 (Technical Committee – Computers in Education) in World Conference on Computers in Education on July 4, 2013 in Torun, Poland.

## 2. MOKYMOSI OBJEKTŲ SRITIES ANALIZĖ IR ĮVERTINIMAS

### 2.1. Įvadas

Mokslinėje literatūroje mokymosi turinys apibrėžiamas kaip *mokymosi objektas* (MO). Pradedant XX a. paskutiniu dešimtmečiu, kai 1994 m. Hodgins pasiūlė šį terminą, susidomėjimas MO e.mokymosi nuolat didėja, nes sritis apima platų įrankių, metodologijų, technologijų ir standartų spektrą. Turint abstrakčią bendrinę sąvoką galima vienareikšmiškai aprašyti, pateikti ir keistis informacija. Mokymosi turinį galima modeliuoti susiejant jį su pedagoginiais ir socialiniais atributais, t.y. mokymosi teorijomis, mokinių psichologinėmis ypatybėmis ir kt. Be to, neturint bendrinio termino (t.y. mokymosi objekto), būtų neįmanoma plėtoti ir kurti e.mokymosi teorijų, lyginti e.mokymosi rezultatus, keistis mokslinė informacija bei praktinė patirtimi. Taigi termino metodologinė ir mokslinė reikšmė didžiulė.

Tyrimai apie MO e.mokymosi sudaro atskirą šaką, kuri vis plečiama ir tobulinama. MO naudotojų sąrašas yra labai platus: mokytojai, mokiniai, tyrėjai, kursų projektuotojai, mokslininkų ir organizacijų grupės ir pan. Esminiu posūkiu šituose tyrimuose reikia laikyti generatyvinio mokymosi objekto (angl. *Generative Learning Object* - GLO) koncepciją, kurią pasiūlė Boyle su kolegomis [LBM+04]. Šie autoriai kildina GMO iš *generatyvinės lingvistikos* ir sieja su MO pakartotinio panaudojimo išplėtimu [CETL14]. Pastarajame šaltinyje pateikiamas toks GMO apibrėžimas: „GMO yra aiškus vykdomasis mokymosi kūrinys (projektas), kuris sukuria tam tikrą mokymosi objektų klasę“ (*“An articulated and executable learning design that produces a class of learning objects”*). GMO koncepcijos pionieriai apibūdina GMO kaip naujos kartos MO [LBM+04].

Nors informatikos mokymasis turi daug bendrų bruožų su e.mokymosi tyrimais apskritai, tačiau dėl turinio metodų, aplinkų ir kt. specifiškumo sudaro savarankišką sritį.

Ši trumpa įžanga ir pateiktos išvalgos suponuoja tokius sąryšius ir jų įvertinimus: (a) e.mokymasis – MO; (b) MO – GMO; (c) e.mokymasis – mokymosi aplinkos – GMO – MO; (d) informatikos mokymasis – mokymosi aplinkos – IGMO – MO. Todėl žemiau pasiūlyta literatūros analizės schema seka iš nurodytų sąryšių ir pagrindžia šio skyriaus struktūrą.

Literatūros šaltinių apžvalga padalinta į dvi dalis: 1) 2.2 – 2.8 skyreliuose aptarti šaltiniai, susiję su reikalavimais mokymosi objektams ir jų projektavimo e.mokymosi sričiai aspektais; 2) 2.9 ir 2.10 skyreliuose atlikta informatikos (programavimo) mokymosi srities analizė, kurios rezultatai tiesiogiai naudojami projektuojant IGMO programavimo mokymuisi. 2.11 skyrelyje pateiktas apibendrintas tyrimo karkasas, 2.12 skyrelyje – išvados.

### 2.2. E.mokymosi bruožai ir iššūkiai

Sparčiai vystantis informacinėms technologijoms iškyla lankstaus žinių perdavimo ir atnaujinimo būtinybė. Šių problemų sprendimui gali būti sėkmingai taikomas e.mokymasis, apimantis plačią taikymų ir procesų aibę, kurie padidina mokymosi efektyvumą. Terminas „e.mokymasis“ apibrėžiamas kaip „mokymasis, kai jo efektyvumui didinti naudojamos informacinės ir komunikacinės

technologijos. Gali būti įvairių modelių: nuo tradicinio mokymosi, papildyto informacinėmis ir komunikacinėmis technologijomis, iki visiškai pagrįsto informacinėmis ir komunikacinėmis technologijomis ir vykstančio virtualiai sukurtoje aplinkoje. Elektroninis mokymasis užtikrina mokymosi interaktyvumą ir besimokančiojo aktyvumą“ [DGJ08]. Pagrindinės e.mokymosi varomosios jėgos yra *techninių inovacijų taikymas, organizacinių ir verslo sprendimų plėtra ir individualūs besimokančiojo poreikiai* [Nat12]. E.mokymosi pokyčius lemia visų trijų komponentų kompozicija.

Specialistai e.mokymąsi siūlo numatyti kaip mokymąsi realiu laiku (angl. *real-time learning*) arba tiksliai laiku (angl. *just-in-time knowledge*) [Nat12].

Toliau šiame skyrelyje aptariami e.mokymosi bruožai ir iššūkiai. E.mokymasis pasižymi žemiau išvardintais bruožais:

1. Žinių tinklų kūrimas (angl. *Knowledge Networking*) – žinios gaunamos iš profesionalių žinių duomenų bazių.

2. Pasirenkamasis mokymasis (angl. *Arbitrary Learning*) – besimokantysis gali pasirinkti lankstų mokymosi tvarkaraštį ir vietą, įgyja daugiau mokymosi laisvės.

3. Nuolatinis mokymosi turinio atnaujinimas ir mokomųjų išteklių įvairovės didinimas saugyklose.

4. E.mokymuisi pritaikytų mokymosi metodų, išteklių, vertinimo sistemų kūrimas.

5. Šiuolaikinių išmaniųjų e.mokymosi sistemų (angl. *Intelligent E-Learning Systems*), glaudžiai susijusių su duomenų saugyklomis, kuriose naudojami išmanieji algoritmai, (angl., *intelligent algorithms*, t.y. dirbtiniu intelektu pagrįsti duomenų gavybos algoritmai, kurie e.mokymosi tyrimuose vadinami išmaniaisiais) kūrimas.

Išmaniųjų e.mokymosi sistemų architektūrų ir paslaugų lyginamoji analizė [SDT12] išryškina e.mokymosi iššūkius: 1) kurti mobiliojo agento (angl. *mobile agent*) architektūras semantiniu žiniatinkliu pagrįstoms e.mokymosi sistemoms, suteikiant agentui daugiau žinių ir išmanumo; 2) didinti pateikiamą informaciją vizualizuojančių mokomųjų išteklių skaičių; 3) suteikti naujų bendradarbiavimo (bendravimui su besimokančiuoju taikyti dirbtinio intelekto pasiekimus) ir informacijos atnaujinimo (vidinės ir internetinės duomenų bazės, wiki ir pan.) galimybių.

E.mokymosi personalizavimo problemos nagrinėjamos Essalmi *ir kt.* [EAJ+10], Mbendera *ir kt.* [MKS10], Klašnja-Miličević *ir kt.* [KVI+11] darbuose leidžia daryti išvadą, kad vieningos personalizavimo strategijos nėra. Kiekviena strategija priklauso nuo personalizavimo parametrų: besimokančiojo charakteristikų, e.mokymesi naudojamų mokymosi metodų ir MO personalizavimo problemą Nedungadi ir Raman [NR12] siūlo spręsti integruojant e.mokymąsi su mobiliuoju mokymusi, išnaudojant mobiliųjų įtaisų suteikiamą lankstumą. Ramadhani *ir kt.* [RAH+09] sukūrė MO ontologiją, palaikančią e.mokymosi personalizavimą, kuri integruojama su studento modelio ontologija (angl. *Student Model Ontology*).

Mokymosi konteksto adaptavimas pagal besimokančiojo poreikius glaudžiai susijęs su turinio konteksto gerinimu, adaptyviu mokymosi procesu (angl. *Adaptive Learning Process*) [AAB+06]. Jovanović *ir kt.* [JGK+07] problemą siūlo spręsti įvedant MO kontekstą, kuris apibrėžiamas kaip unikali tarpusavyje susijusių

duomenų, charakterizuojančių konkrečią mokymosi situaciją, aibė. Mokymosi objektų kontekstas formalizuojamas karkase, sujungiančiame besimokančiojo modeliavimo bei turinio struktūrizavimo ontologijas, ir fiksuojančiame informaciją apie specifinį MO naudojimą projektuojant mokymąsi.

### **2.2.1. E.mokymosi technologijos ir sistemos**

Programinė įranga numato tam tikrų veiklų realizaciją naudojant hipertekstą, vadovėlius, video pamokas, simulatorius, treniruoklius, mokomuosius žaidimus, pateiktis, elektroninius testus, pokalbius, forumus, wiki, dienoraščius, el. pašta, momentinius pranešimus ir t.t. Iš kitos pusės, e.mokymosi sistemos integruoja ir susieja įvairius įrankius į vieningą mokymosi platformą, pagrįstą programinės įrangos ir mokomųjų išteklių komunikavimu. E.mokymosi išskiriamos trijų tipų platformos [Nat12]:

1. Į savarankiškas studijas (angl. *self study*) orientuotos platformos, kai besimokantieji mokosi savarankiškai. Platformose naudojami įrankiai ir ištekliai yra parengiami iš anksto ir mokinys juos naudoja savo nuožiūra. Šio tipo platformose dažniausiai naudojami vadovėliai, elektroninės knygos, taip pat naudojamas modeliavimas, mokomieji žaidimai.

2. Asinchroninio e.mokymosi (angl. *asynchronous e-learning*) platformose dominuoja įrankiai, kurie nebūtinai visiems besimokantiems pateikiami vienu metu: video pamokos, klausimai skelbimų lentoje ir pan.

3. Sinchroninį e.mokymąsi palaikančiose platformose (angl. *synchronous e-learning*) besimokantieji jungiasi į virtualią sesiją tuo pačiu metu, dominuoja bendradarbiavimu pagrįstas mokymasis.

E.mokymosi sistemos klasifikuojamos į tris grupes [Nat12]: 1) mokymosi valdymo sistemas (angl. *Learning Management Systems*), kurios pristato mokymosi modulius, užtikrina besimokančiųjų prieigos kontrolę, atlieka mokinių veiklų pažangos stebėseną; 2) turinio valdymo sistemas (angl. *Content Management Systems*), sudarančias programinės įrangos šeimyną, kurioje yra turinio kūrimo, saugojimo, valdymo ir modifikavimo įrankiai. Šios sistemos turi labai svarbią savybę – kuriami pakartotinai panaudojami turinio komponentai (pakartotinai panaudojami mokymosi objektai, angl. *Reusable Learning Objects*); 3) mokymosi turinio valdymo sistemos (angl. *Learning Content Management Systems*) sujungia pirmųjų dviejų grupių sistemų galimybes. Šios sistemos dažniausiai yra internetinės, jose mokymosi turinys kuriamas naudojant elementarius pakartotinai panaudojamus komponentus, kurie pateikiami atsižvelgiant į besimokančiųjų pažangą.

### **2.2.2. E.mokymosi sistemose naudojami mokymosi metodai**

E.mokymasis apima dvi dideles klases: eksperimentinis (angl. *experimental*) ir pažintinis (angl. *cognitive*) mokymasis. Tyrimai rodo, kad besimokantieji mokosi efektyviau tuomet, kai jie yra aktyvūs mokymosi proceso dalyviai. Ši priežastis lėmė, kad į mokytoją ir turinį orientuoti mokymosi metodai keičiami į besimokantįjį orientuotais metodais. Mokytojas tampa patarėju, vedliu ir skatintoju. Daugelyje e.mokymosi sistemų naudojami projektų kūrimu, problemų sprendimu, aktyviu mokymusi ir tyrimu pagrįsti mokymosi metodai. Besimokantieji turi įdėti daug pastangų analizuodami, sintetindami ir vertindami mokomąją medžiagą [Nat12].

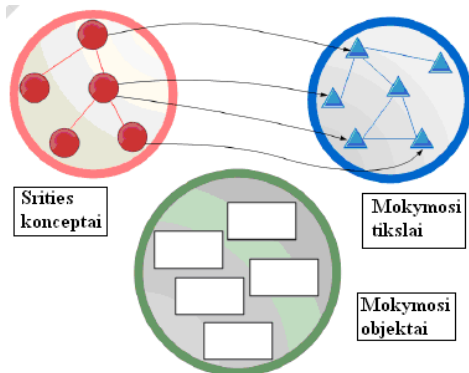
### 2.2.3. E.mokymesi naudojamų mokymosi objektų savybės

E.mokymosi sistemose MO apibrėžiamas kaip nepriklausomas ir savarankiškas mokymosi turinio vienetas, kuris būtų pakartotinai panaudojamas daugelyje mokymosi kontekstų. Pagrindinis principas, kurio laikomasi formuojant mokomąją medžiagą yra „lanksčiai pakartotinai panaudoti mažus informacijos vienetus (angl. *information chunks*) dar vadinamus pakartotinai panaudojamais MO (angl. *Reusable Learning Objects*)“ [NSP+12], [Nat12]. Šis principas glaudžiai susijęs su objektų panaudojimu objektinėje programų inžinerijoje.

Reikalavimai mokymosi objektams, naudojamiems e.mokymesi, apibrėžiami Littlejohn *ir kt.* [LFM08], Nikolopoulos *ir kt.* [NSP+12] darbuose: 1) prieinamumas (angl. *accessibility*) – MO turi būti aprašomi metaduomenimis, kad juos būtų lengva surasti saugyklose ir duomenų bazėse; 2) pakartotinis panaudojimas (angl. *reusability*) – MO naudojamas daugelyje mokymosi kontekstų; 3) tarpusavio sąveika (angl. *interoperability*) – MO turi būti nepriklausomas nuo pateikimo priemonių ir žinių valdymo sistemų; 4) mokinio motyvacijos stiprinimas – MO naudojimas turi suteikti galimybes aktyviai mokytis; 5) MO kokybės užtikrinimas.

Idealiai pakartotinai panaudojamam MO būdingi atributai: 1) modulinė MO struktūra, nepriklausoma nuo taikymų ir aplinkų [Nat12]; 2) nenuosekli modulių jungimo seka (angl. *non sequential*) [Nat12]; 3) vieno ar kelių mokymosi tikslų perdengimas [NSP+12]; 4) prieinamumas plačiai auditorijai [Nat12].

Pagrindiniai mokymosi projektavimo proceso parametrai ir jų tarpusavio sąryšiai pateikti 2.1 pav.



2.1 pav. Esminiai mokymosi projektavimo proceso elementai ir jų koreliacija [NSP+12]

E.mokymosi srityje kalbant apie turinio detalumo (angl. *granularity*) lygius plačiai naudojama agreguoto mokymosi objekto (angl. *Aggregated Learning Object*) sąvoka [NSP+12], [Nat12].

Littlejohn *ir kt.* [LFM08] e.mokymosi ištekliams adaptavo Willey taksonomiją (2.1 lentelė) ir įvertino faktorius, turinčius didžiausią įtaką jų panaudojimui (2.2 pav.).

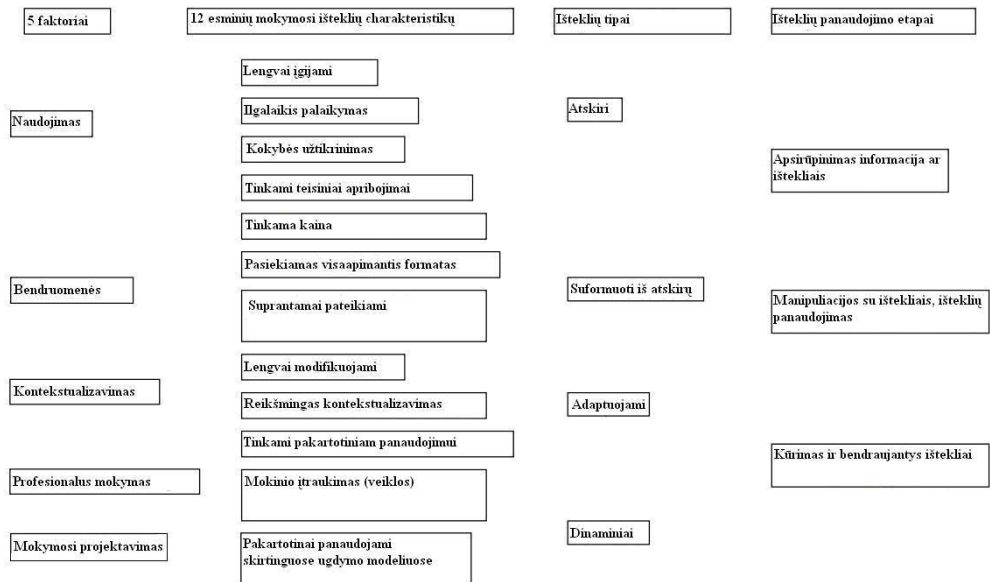
Sakarkar *ir kt.* [SDT12] lygindami išmaniąsias e.mokymosi sistemas išskiria esminius MO modelius, pagrįstus: 1) semantiniiais metaduomenimis (angl. *semantic metadata*), 2) atsakymų į klausimus teorija (angl. *Item Response Theory*), 3) pakartotiniu MO panaudojimu, 4) interneto paslaugomis (angl. *Web Service*), 5)



multimedia įrankių naudojimu, 6) bendradarbiavimu, 7) besimokančiojo profiliu, 8) savarankišku mokymusi, 9) sąryšių koncepcija.

## 2.1 lentelė Mokymosi išteklių komponentų tipai [LFM08]

MO charakteristika	Fundamentalus MO	Sudėtinis išbaigtas MO	Sudėtinis atviras MO	Generatyvinis-pristatomasis MO	Generatyvinis-instrukcinis MO
MO elementų skaičius	Vienas	Mažai	Daug	Mažai-daug	Mažai-daug
Komponento MO tipas	Fundamentalus	Fundamentalus, Sudėtinis-išbaigtas	Visi	Fundamentalus, Sudėtinis-išbaigtas	Fundamentalus, Sudėtinis-išbaigtas, Generatyvinis-pristatomasis
Pakartotinai naudojami komponento MO	Nenaudojamas	Ne	Taip	Taip/Ne	Taip/Ne



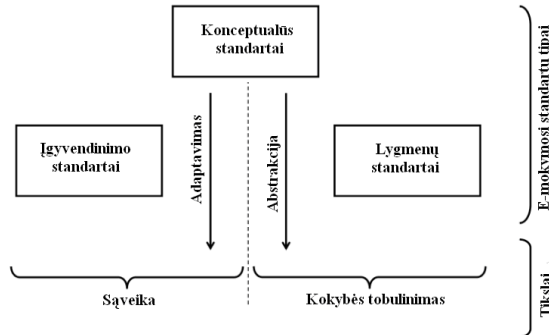
2.2 pav. Faktoriai, turintys didžiausią teigiamą įtaką mokymosi išteklių naudojimui [LFM08]

### 2.2.4. E.mokymosi standartai

E.mokymosi standartizacija apima didelę aibę aspektų, pradedant techniniais ir pedagoginiais, baigiant kokybės. Standartai yra labai svarbūs skirtingų interesų ir poreikių sistemų naudotojams. Juose daugiausia dėmesio skiriama mokymosi pasiūlymams (angl. *learning offers*) apimantiems turinį ir MO, procesams ir visuminiam organizavimui. Išskiriami trys pagrindiniai EM standartų aspektai [Str06]:

1) Standartų tipai (angl. *types of e-Learning standards*) apima įgyvendinimo (angl. *implementation*) standartus, užtikrinančius EM sričių tarpusavio sąveiką,

konceptualius (angl. *conceptual*) standartus, kurie siūlo bendrus teorinius sprendimus palyginant ir suderinant esybes ir objektus, bei lygmenų (angl. *level*) standartus, apibrėžiančius kokybės lygį. 2.3 pav. pateikiami standartų tipai ir tikslai.



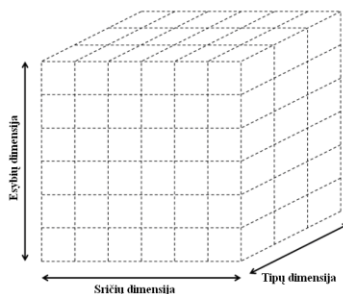
2.3 pav. E.mokymosi standartų tipai ir tikslai [Str06]

2) Standartų sritys (angl. *domains of e-Learning standards*) nurodo, koks dalykas ar tema sprendžiami iš esmės. Skiriamos šešios pagrindinės sritys: semantinės prasmės (angl. *meaning*), kokybės, didaktikos, mokymosi technologijos, mokymosi turinio ir konteksto. EM standartai gali perdengti vieną sritį arba kelių (visų) sričių derinį.

3) Standartų esybės (angl. *entities of e-Learning standards*) apima e.mokymosi aplinką, įvairių grupių vartotojų vaidmenis, e.mokymosi aplinkoje naudojamus metodus, mokymosi sistemas, išteklius, praktiką.

Pavyzdinis standartų modelis pateiktas 2.4 pav. Naudodamas šį modelį Stracke [Str06] ištyrė 14 tarptautinių standartų. Tyrimų rezultatai rodo, kad standartuose daugiausia dėmesio skiriama mokymosi technologijos sričiai ir mokymosi sistemų esybei. Tobulinimo kryptys turėtų būti susietos su konteksto srities ir praktikos esybės įtraukimu į standartus, nes nei viename iš tirtų standartų tai neužfiksuota. Tyrėjas pastebi, kad trūksta lygmenų standartų. Jų trūkumas paaiškinamas tuo, kad dėl kokybės nustatymo sunku sutarti tarptautiniu lygiu.

EM standartų tobulinimo viziją sudaro vieningų tarpdisciplininių sutarimų priėmimas, bendra terminologija, aprašanti semantiką ir taikymus, skirtingais standartais aprašomų specifikacijų suderinimas, bendras pavyzdinis tarptautiniu mastu pripažintas karkasas, sujungiantis aplinkas, architektūras, sistemas ir paslaugas.



2.4 pav. Pavyzdinis standartų modelis [Str06]

### 2.2.5. E.mokymesi naudojamų technologijų tendencijos

Modeliavimas (angl. *Simulations*) plačiai naudojamas srityse, kuriose brangu sukurti realius veikiančius modelius: karyboje, aviacijoje, aeronautikoje. Modelių naudojimas nuolat auga, jie tampa vis sudėtingesni ir efektyvesni [Nat12].

Adaptyvios mokymosi aplinkos (angl. *Adaptive Learning Environments*) užtikrina vartotojų stebėseną, kai mokinių veiklos interpretuojamos remiantis specifiniais srities modeliais ir besimokančiojo poreikiais [Nat12].

Atvirojo kodo EM įrankiai (angl. *Open Source e-learning tools*) kuriami kaip alternatyva komerciniams produktams. Manoma, kad ateityje atvirojo kodo įrankiai sudarys didžiausią EM įrankių dalį [Nat12].

Mobiliosios technologijos (angl. *Mobile Technologies*) vis sparčiau diegiamos į EM sritį dėl dviejų priežasčių: 1) besimokantiesiems, gyvenantiems nutolusiose vietovėse, besivystančiose šalyse iškyla infrastruktūros problema; 2) besimokantieji daug keliauja ir mokosi individualiai [Nat12].

M.mokymasis (angl. *Mobile Learning*, mobilusis mokymasis) ir U.mokymasis (angl. *Ubiquitous Learning*, visur esantis mokymasis) praturtina EM naujomis galimybėmis [LH10] (2.2 lentelė).

#### 2.2 lentelė E.mokymosi, M.mokymosi ir U.mokymosi pagrindinės savybės [LH10]

Teoriniai ir praktiniai kintamieji	Tradicinis e.mokymasis	M.mokymasis	U.mokymasis
Mokymosi charakteristikos	Nuotolinis, holistinis, sinchroninis ir asinchroninis	Tradicinis e.mokymasis + autentiška mokymosi aplinka, savalaikė prieiga prie informacijos	Tradicinis e.mokymasis + M.mokymasis + adaptyvus ir aktyvus mokymosi palaikymas
Naudojami įrankiai	Kompiuteris, nešiojamas kompiuteris, interneto palaikymo įrenginiai	Mobilieji įrenginiai su belaidžiu ryšiu	Jutiklių technologijos su mobiliaisiais įrenginiais ir belaidžiu ryšiu
Kontrolė: 1) vidinė, pagrįsta besimokančiojo perspektyva; 2) išorinė, pagrįsta įrankių naudojimu	Save nukreipiantis vartotojas  Vadovavimas, pagrįstas elgsena tinkle	Aktyvus vartotojas  Vadovavimas, pagrįstas elgsena belaidžiam tinkle	Aktyvus ar jutiklių motyvuojamas vartotojas Vadovavimas, pagrįstas autentiška elgsena
Informacijos šaltiniai	Serveriai	Belaidžiai serveriai ir autentiški objektai	Belaidžiai serveriai ir autentiški objektai su jutikliais
Panaudojimas ugdyme ir pramonėje	Beveik kiekvienoje srityje ir disciplinoje	Mokymosi konstatuojamasis pažinimas, tikslo objektų stebėjimas ir klasifikavimas	Mokymosi procedūrinis pažinimas, mokymasis atlikti sudėtingus eksperimentus
Dėstymo būdai	Vienas-vienas, vienas-grupė, grupė-grupė	Tradicinio e.mokymosi veiklos autentiškame kontekste, užtikrinančios konstatuojamąjį pažinimą	Tradicinio e.mokymosi veiklos autentiškame kontekste, užtikrinančios procedūrinį pažinimą

Vertinimo būdai	Vertėmis pagrįstas, sinchroninis ir asinchroninis vertinimas, kurį atlieka vienas ar keli instruktoriai, mokymosi sistemos atliekamas įvertinimas	Vertėmis pagrįstas vertinimas, kurį atlieka vienas ar keli instruktoriai, mokymosi sistemos atliekamas įvertinimas	Vertėmis pagrįstas vertinimas, kurį atlieka vienas ar keli instruktoriai, mokymosi sistemos atliekamas įvertinimas, specialiai pritaikytas realaus pasaulio veiklų vertinimui
Mokymosi scenarijai	Pasyvus mokymosi kontekstas internete	Realus pasaulis ir pasyvus mokymosi kontekstas internete	Realus pasaulis ir labiau aktyvus mokymosi kontekstas internete
Pedagginės teorijos ar konsultavimo strategijos	Beveik visos pedagoginės teorijos ir konsultavimo strategijos	Tradicinio e.mokymosi + projektais pagrįstas mokymasis, autentiškas mokymasis	Tradicinio e.mokymosi + M.mokymosi + „pameistrystė“

### 2.3. Svarbiausi mokymosi objektų evoliucijos etapai

Mokymosi objektų (MO) istorijos pradžia galima laikyti 17 amžių, kai filosofas empirikas John Locke pasiūlė MO projektavimo idėją, pateikdamas pavyzdžių, kaip žaislai gali būti panaudojami mokantis pažinti raides. Locke išvalga turėjo įtakos daugeliui filosofų, kurie išsiskyrė į tris judėjimus. „Protingos rankos“ (angl. *Intelligent Hand*) judėjimas daugiausia dėmesio skiria jutiminei sąveikai su objektais. „Eksperimentinis“ (angl. *Experimental*) judėjimas akcentuoja realaus pasaulio eksperimentus ir sąveiką su gamta. „Supaprastintos realybės“ (angl. *Simplified Reality*) judėjimas esminiais dalykais laiko patirtį ir refleksiją, demokratiją ir bendruomenę bei mokymosi aplinkas.

Mokymosi pionieriai ieškojo naujų būdų, kaip padaryti mokymąsi įdomesnę ir efektyvesnę. Jų idėjos neprarado savo vertės šiomis dienomis ir yra perkeltos į MO skaitmeninį konstravimą ir projektavimą (angl. *Digital Construction & Design*), skaitmeninę konceptualią manipuliaciją (angl. *Digital Conceptual Manipulation*) bei skaitmeninės tikrovės vaidmenų MO kūrimą (angl. *Digital Reality Role Play LOs*) [Zuc06].

MO sąvoka pirmą kartą buvo aprašyta Gerard 1967 m. [Ger67] technologijų taikymo mokymosi kontekste. Terminą „mokymosi objektas“ 1994 pasiūlė Wayne Hodgins ir jį apibrėžė kaip „skaitmeninį ar kitokį objektą, kuris gali būti naudojamas, pakartotinai naudojamas ar nurodomas taikant technologijomis pagrįstą mokymąsi“.

2000 metais Wiley [Wil00] pasiūlė MO pažinimo, adaptavimo ir palaikymo idėją, techninių standartų būtinybę ir mokymosi principais pagrįstą technologijų panaudojimą.

Sparti MO naudojimo plėtra iškėlė naujas iniciatyvas aprašyti MO metaduomenimis, kad būtų galima MO efektyviau pakartotinai panaudoti, palengvinti jų sąveiką internetinėse mokymosi valdymo sistemose (angl. *Online Learning Management Systems*) [LTSC02]. MO metaduomenų standartai akcentuoja

minimalią atributų aibę, kurios pakanka MO valdymui, talpinimui saugyklose ir įvertinimui.

Platesniame kontekste MO suprantama kaip abstrakcija arba modelis, palaikantis pakartotinę MO panaudojimą tarp didelių e.mokymosi bendruomenių [Lib05].

Kadangi mokymasis yra daugiatakslis ir daugiapakopis procesas, tradicinių MO modelių, – Metaduomenų – Turinio (angl. *Metadata – Content*) [BMO08], hierarchinių granuliacijos koncepcija turinio viduje pagrįstų [VD04], – nebepakanka. Tai lemia daug priežasčių: 1) e.mokymosi sritis vystosi labai sparčiai, todėl didėja lankstesnių, lengvai adaptuojamų personalizuotų ir kontekstualizuotų MO poreikis; 2) tradiciniai komponentiniai MO turi ribotas pakartotinio panaudojimo ir adaptavimo galimybes; 3) augant MO skaičiui, sparčiai didėja skaitmeninių bibliotekų ir saugyklų skaičius, tai susiję su valdymo ir priežiūros problemomis, analogiškoms bibliotekų mastelio (angl. *library scaling*) problemoms programų inžinerijoje [DKM04], [Nas05]; 4) e.mokymosi sritis apima platų įrankių, metodologijų, technologijų ir standartų spektrą, todėl reikia naujų MO modelių, tinkamų aukščiau išvardintoms problemoms spręsti.

2004 m. Boyle ir jo kolegų [LBM+04] pasiūlyta generatyvinio mokymosi objekto (GMO) koncepcija išskėlė pedagoginių ir socialinių aspektų svarbą ir išskyrė turinį į atskirą sluoksnį. GMO kūrėjai laikosi principų: 1) akcentuojamas GMO projektavimas ir kontekstas, kur prioritetas suteikiamas pedagoginiams ir socialiniams aspektams. Technologija sujungia pedagoginius ir socialinius aspektus su turinio adaptavimo galimybėmis. 2) Aiški koncepcijų atskirtis ir daugiapakopis požiūris į koncepcijų atskirtį. Štuikys ir Damaševičius [ŠD08] sujungė generatyvinius GMO aspektus su programų inžinerijos koncepcijomis ir papildė GMO principus dviem savybėmis: 1) bet kuri sritis konceptualiame lygmenyje gali būti išreikšta per požymius; 2) aiškiai apibrėžti požymiai gali būti susieti su generatyvine technologija. GMO evoliucija glaudžiai susijusi su mokymosi variantiškumo modeliavimu, transformavimo taisyklėmis ir požūriais. GMO modeliai turi didžiulį panaudojimo e.mokymosi srityje potencialą [LBM+04], [MLB05], [ŠD07], [ŠD08], [Old08], [Boy09], [HK09], [Boy10], [BR12].

Toliau šiame skyriuje pateikiama mokymosi objektų projektavimo analizė e.mokymosi srities ir informatikos mokymosi kontekste.

#### **2.4. Mokymosi objektų pakartotinio panaudojimo problema e.mokymesi**

Mokymosi objektų pakartotinis panaudojimas yra viena iš aktualiausių e.mokymosi srities problemų. Pakartotinę MO panaudojimą galima apibrėžti kaip strategiją, kuria siekiama pagerinti kokybę, patikimumą, sumažinti kūrimo išlaidas [Par03], [Krä05]. Programų inžinerijos kontekste pakartotinis panaudojimas apibrėžiamas kaip „egzistuojančios programinės įrangos ar jos dalių bei žinių apie programinę įrangą naudojimas, siekiant sukurti naują programinę įrangą“. E.mokymosi srityje pakartotinis MO panaudojimas interpretuojamas nevienareikšmiškai, jį skirtingai supranta MO kūrėjai, kurso sudarytojai ir besimokantieji. Mokslinėje literatūroje [Par03], [Krä05], [PS04], [Pol03], [SG03],

[AM10] nagrinėjamos teorinės MO pakartotinio naudojimo prielaidos, tačiau praktikoje pakartotinis MO naudojimas diegiamas lėtai [Par03], [Krä05].

Nagrinėjant mokymosi objektus pakartotinio panaudojimo kontekste siekiama atsakyti į klausimus:

1. Kaip pakartotinį MO panaudojimą traktuoja e.mokymosi srities bendruomenė?

2. Kaip padidinti MO pakartotinio panaudojimo galimybes?

Literatūroje dominuoja du požiūriai į MO pakartotinį panaudojimą: pirmasis požiūris akcentuoja saugojimo, paieškos, perdavimo ir nuo naudotojo priklausomo adaptavimo naujam kontekstui aspektus. Šie aspektai būdingi komponentinei MO paradigmai, kuri nekelia tikslo automatizuoti MO kūrimą ir adaptavimą. Antrasis požiūris yra platesnis, susiejantis mokymosi objektų kūrimo ir struktūros aspektus.

#### **2.4.1. Mokymosi objektų pakartotinio naudojimo aspektai būdingi komponentinei paradigmai**

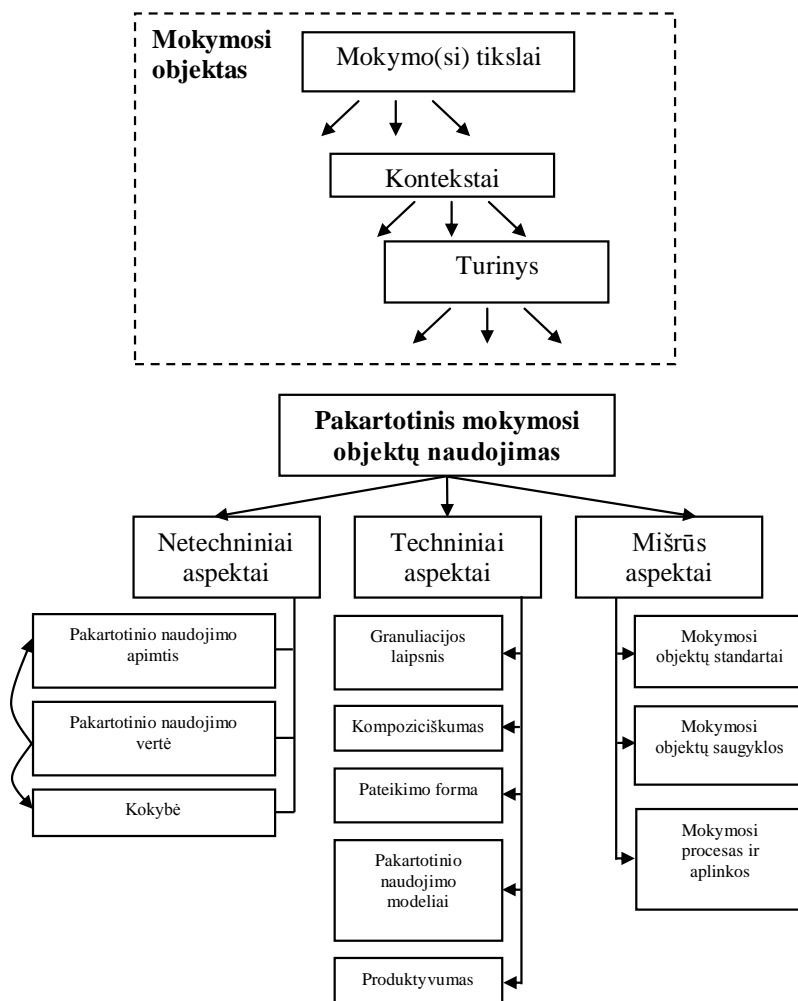
Pakartotinis MO naudojimas yra labai svarbi savybė, tiesiogiai minima MO apibrėžimuose: mokymosi objektas yra nepriklausomas ir savarankiškas mokymosi turinio vienetas, turintis vieną mokymosi tikslą ir kuriamas iš anksto numatant pakartotinio naudojimo galimybę daugelyje kontekstų [Pol03], [SG03].

Išanalizavus ir apibendrinus mokslinę literatūrą, išskirtos 3 faktorių grupės, turinčios didžiausią įtaką MO pakartotiniam naudojimui (2.5 pav.). Netechniniai aspektai susiję su MO pakartotinio naudojimo tikslais išsamiai išnagrinėti Silveira *ir kt.* [SAA+07], Quinton [Qui07] darbuose. Pakartotinio MO panaudojimo vertė glaudžiai susijusi su MO pakartotinio naudojimo apimtimi ir kokybe. Quinton [Qui07] pažymi, kad MO turinio vertė didėja kiekvieną kartą jį panaudojus pakartotinai. Kita labai svarbi savybė – galimybė naudoti MO įvairiuose kontekstuose ir mokymosi sistemose.

Išnagrinėjus ir apibendrinus MO turinio [SAA+07], [Qui07] ir konteksto [TD09], [MJ10] modelius, turinio ir konteksto ryšius [PS04], [Qui07], [DGJ+04] išskirti svarbiausi techniniai pakartotinio naudojimo aspektai. Granuliacijos (detalumo) laipsnis yra labai svarbi MO charakteristika, apibrėžianti, kaip MO formuojami iš mažesnių komponentų. Sėkmingiausiai pakartotinai panaudojami tie MO, kurie turi vieną mokymosi tikslą, t.y. paveikslėlis, teksto fragmentas, audio-video failas. Sudėtingesni MO agreguojami iš paprastesnių MO komponentų, arba jų rinkinių: pvz., jei pamokos, kursai yra formuojami iš daugelio MO, turinčių ne vieną mokymosi tikslą, tuomet galimas viso rinkinio pakartotinis panaudojimas, o galimybė pakartotinai naudoti atskirus komponentus yra maža. Churchill [Chu07a] pasiūlė MO klasifikaciją, kuri palengvina MO naudojimą įvairiuose pedagoginiuose kontekstuose.

MO standartai ir saugyklos [PS04], [HP05], mokymosi procesas ir aplinkos [Qui07] taip pat yra labai svarbūs MO pakartotiniam naudojimui. Jie priskirti mišrių aspektų grupei.

Apibendrinami mokslinę literatūrą galime teigti, kad e.mokymosi pakartotinis naudojimas suprantamas kaip MO naudojimas turinio, techniniame ir socialiniame-pedagoginiame kontekstuose.



2.5 pav. Svarbiausi MO pakartotinio naudojimo aspektai

#### 2.4.2. Mokymosi objektų pakartotinio panaudojimo modeliai

Šiame skyrelyje aptariamas sistemingesnis požiūris į pakartotinį MO naudojimą ir lyginami MO pakartotinio panaudojimo modeliai, susiejantys mokymosi objektų kūrimo ir struktūros aspektus.

Huddlestone ir kt. [HP05] pasiūlė keturių pakopų pakartotinio panaudojimo modelį (angl. *a four tie model*), kuriame strateginė pakopa apibrėžia pakartotinio panaudojimo planavimą organizacijos viduje, operacinė pakopa nusako kaip pakartotinio panaudojimo operacijos vykdomos organizacijos viduje, kontekstinėje pakopoje vertinama, kaip MO gali būti panaudojamas naujame mokymosi kontekste, struktūrinė pakopa apibrėžia MO struktūros reikalavimus: diskretumą, atskiriamumą, suderinamumą, granuliaciją, modifikavimo galimybes.

Keturių komponentų mokymosi projektavimo modelį (angl. *4C/ID Model*) [MB05] sudaro: mokymosi užduotys – mokymosi pratybos, pateikiamos mokiniams;

informacija, paaiškinanti, kaip organizuota sritis ir kokie svarbiausi srities uždaviniai; procedūrinė informacija, kuri yra būtina aprašant užduoties specifikaciją; praktika, įvertinanti įprastus mokymosi aspektus ir pasižyminti aukšto lygio automatizacija.

Ontologija pagrįsta struktūra/karkasas sujungiantis mokymosi projektavimą ir mokomąjį turinį (angl. *an Ontology-Based Framework for Bridging Learning Design and Learning Content*) [KGR06]. MO paimami iš saugyklos ir panaudojant mokymosi projektavimą integruojami į mokymosi veiklas. Šioje struktūroje tarp mokymosi projektavimo ir MO įdedamas konteksto sluoksnis, padidinantis mokymosi projektavimo ir MO pakartotinio panaudojimo galimybes.

MO pakartotinio panaudojimo struktūra/karkasą (angl. *framework for LO reusability*) [SP09] sudaro: MO paieška saugykloje; naujų MO kūrimas; MO aprašymas naudojant metaduomenis; leidimas naudoti sukurtus MO; kokybės gerinimas; publikavimas; modifikavimas; agregacija su kitais MO; integravimas; grįžtamasis ryšys; nenaudojamų MO pašalinimas.

Lyginamoji MO pakartotinio panaudojimo modelių analizė pateikiama 2.3 lentelėje. Modeliai lyginami įvertinant kaip juose išreikšti turinio/konteksto formavimo, techninės ir socialinės charakteristikos.

### 2.3 lentelė MO pakartotinio panaudojimo modelių lyginamoji analizė

Modelis (struktūra, karkasas)	Svarbiausių charakteristikų grupės + (išreikšta), ++ (stipriai išreikšta), +++ (labai stipriai išreikšta), +- (silpnai išreikšta), - (neišreikšta)		
	Turinio/konteksto formavimas	Techninės	Socialinės
Keturių pakopų pakartotinio panaudojimo modelis	++	+	++
Keturių komponentų mokymosi projektavimo modelis	++	+-	+
MO pakartotinio panaudojimo struktūra	++	++	++
Ontologija pagrįsta struktūra / karkasas	++	++	+

Modelių lyginamoji analizė rodo, kad didžiausias dėmesys kuriant modelius skiriamas turinio/konteksto formavimui. Komponentinėje paradigmoje MO pakartotinis panaudojimas maksimizuojamas pašalinant kontekstą, tačiau kontekstas yra pagrindinė mokymosi proceso dalis, be kurios MO gali tapti sudėtingu ir neefektyviu.

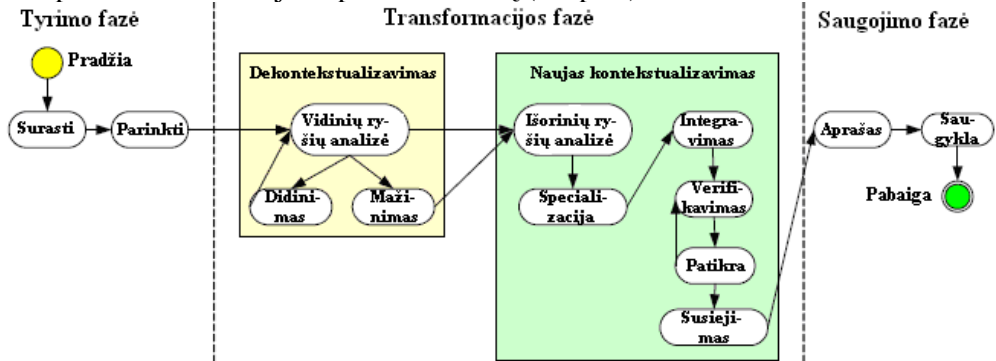
### 2.5. E.mokymosi srities ir mokymosi objektų konteksto analizė

Atliekant srities analizę ir kuriant srities modelį vienas iš svarbiausių uždavinių – konteksto analizė ir modeliavimas. Konteksto modeliavimas yra esminės informacijos apie kontekstą išskyrimas ir pateikimas tolesniam apdorojimui. Modeliuojant kontekstą labai svarbu nustatyti specifinį srities kontekstą, surasti sąryšius tarp kontekstą apibūdinančių elementų ir modeliuojant tuos elementus susieti, identifikuoti galimus dinامينius konteksto pokyčius ir



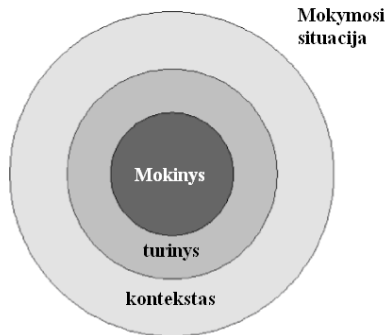
atsižvelgti į juos kuriant konteksto modelį [SB08]. E.mokymosi ir MO srityse tyrėjai akcentuoja skirtingus konteksto aspektus. Trumpai juos aptarsime.

Weitl *ir kt.* [WKG04] apibrėžė kontekstą kaip turinio fragmentą, apimančią visą projektavimo ir taikymo srities informaciją ir sąlygas bei pasiūlė kontekstualizuotų MO pakartotinio naudojimo proceso modelį (2.6 pav.).



2.6 pav. Kontekstualizuoto mokymosi turinio pakartotinio panaudojimo proceso modelis [WKG04]

Figueiredo [Fig10] kontekstą apibrėžė kaip besimokančiajam kuriančiam žinias naudojant MO tinkamų sąlygų aibę, ir pasiūlė modelį, susiejantį besimokantį su mokymosi turiniu ir kontekstu mokymosi situacijoje (angl. *learning event*) (2.7 pav.).



2.7 pav. Besimokančiojo, turinio ir konteksto sąryšiai mokymosi situacijoje [Fig10]

Safran *ir kt.* [SGG06] kontekstą apibrėžė kaip tarpusavyje susijusias sąlygas, kuriose kas nors egzistuoja arba vyksta ir mokymosi veikloms palaikyti pasiūlė sąvokomis pagrįstą konteksto modeliavimo sistemą panaudojant dinamiškai generuojamas nuo konteksto priklausančias informacijos erdves (angl. *information spaces*).

Azouaou ir Desmoulins [AD06] X elemento kontekstą išskiria kaip elemento Y savybes S, tenkinančias tris sąlygas: 1) Y yra aplink X, 2) Y įprasmina X, 3) S yra reikšmingos X atžvilgiu. Moore *ir kt.* [MHZ+07] pažymi, kad kontekstas yra sudarytas iš subkontekstų, aprašančių esybes ir sąryšius tarp jų. Tokie požiūriai į

kontekstą leidžia teigti, kad ne visos konteksto elementų savybės yra turimos omenyje, o tik tos, kurios yra reikšmingos.

Liu *ir kt.* [LCW+09] kontekstą aprašo kaip informaciją, identifikuojančią esybės būseną, t.y. besimokančiojo vietą, mokymosi veiklą, naudojamus įrankius ir MO. Allen ir Mugisa [AM10] kontekstu vadina MO bendrąją situaciją ir MO taikymo sritis. Zeng *ir kt.* [ZLZ09] pažymi, kad kontekstas apima į subjekta orientuotas, integruotas ir dinamiškas savybes. Taigi galima teigti, kad visa informacija, susieta su subjektu, gali būti vadinama nuolat kintančiu subjekto kontekstu.

Das *ir kt.* [DBC+10] pasiūlė kontekstą suprantančios e.mokymosi sistemos architektūrą, kurioje panaudojamas standartizuotas konteksto modelis (angl. *Standardized Context Model*), sukurtas apibendrinus mokslinę literatūrą (2.4 lentelė).

## 2.4 lentelė Konteksto parametrai

Konteksto kategorija	Konteksto parametrai	Konteksto subparametrai
Profilio kontekstas	Besimokančiojo profilis	Asmeninė informacija
		Pažinimo lygis
		Technologijų įvaldymo lygis
	Besimokančiojo lygis	Pradedantysis
		Vidutinis
		Aukštesnysis
Tinkamumo kontekstas	Mokymosi stilius	Video
		Audio
		Tekstas
		Animacija
		Skaidrės
	Mokymosi preferencijos	Konceptuali
		Pagrįsta pavyzdžiais
		Atvejo analizė
		Modeliavimas
	Mokymosi tikslas	Demonstravimas
		Tyrimas
		Apžvalga
		Greitos nuorodos
		Įvadas
		Projektas
		Vertinimas
		Seminaras
		Infrastruktūros kontekstas
Viešas		
Reikalavimai mokymosi aplinkoms	Funkcionalūs	
	Nefunkcionalūs	
Tinklai	Laidiniai	
	Belaidžiai	
Priemonės	Personalinis kompiuteris	
	Nešiojamas kompiuteris	
	Planšetinis kompiuteris	
	Mobilus telefonas	

Mokymosi kontekstas	Mokymosi tempas	Lėtas
		Vidutinis
		Greitas
	Mokymosi būseną	Studijuota
		Bus studijuojama
		Bus peržiūrima
	Supratimo lygis	Visiškai nesupranta
		Trupuči supranta
		Gerai supranta
		Pilnai supranta

Apibendrinami galime teigti, kad MO konteksto modelis turi apimti įvairių tipų kontekstinę informaciją, kuri adaptuojama kintančioje aplinkoje. Be to, konteksto elementai susiję tarpusavyje, t.y. vienos savybės reikšmių pokyčiai sukelia kitų savybių reikšmių pokyčius.

## 2.6. Mokymosi objektų modeliai

Vieni pirmųjų struktūrinį pakartotinai panaudojamo MO modelį (2.8 pav.) 2003 metais pasiūlė CISCO Systems [CISCO03]. Laikantis šio požiūrio MO skirtas vienam mokymosi tikslui pasiekti, sudarytas iš statinio ir interaktyvaus turinio bei mokomųjų veiklų. MO kokybė įvertinama matuojant, ar pasiektas mokymosi ar veiklos tikslas. MO viduje turinys, veiklos ir įvertinimas yra kuriami iš teksto, audio, video, animacijos fragmentų, Java kodo, apletų. MO aprašomas metaduomenimis, kurie palengvina MO paiešką ir pakartotinį panaudojimą.

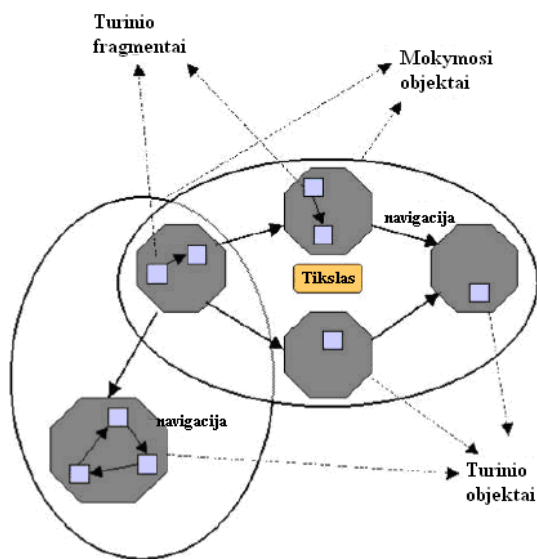


2.8 pav. MO struktūrinis modelis [CISCO03]

2004 m. Verbert ir Duval [VD04] pasiūlė bendrą MO turinio modelį (2.9 pav.). Šiame modelyje turinio fragmentai yra pagrindinių formų mokymosi ištekliai: tekstas, audio, video, paveikslėliai. Jie tarpusavyje yra nesujungti. Turinio fragmentų aibės sudaro turinio objektus. Turinio objektai agreguoja turinio fragmentus ir prideda navigaciją. Turinio fragmentai laikomi egzemplioriais, o turinio objektai – abstraktūs tipai.

Turinio fragmentai išplečiami pridedant veiklas ir žmones, turinio objektai – veiklų tipus ir vaidmenis.

Aukštesniame lygmenyje MO agreguoja turinio objektus ir prideda mokymosi tikslą.



2.9 pav. Bendras MO turinio modelis [VD04]

[SAA+07] apibrėžė sąryšius tarp mokymosi išteklių tipo, granuliacijos ir pakartotinio panaudojimo, [BMO08] pasiūlė skirtingų agregacijos lygių turinio modelius. Apibendrinti tyrimų rezultatai pateikti 2.5 lentelėje.

**2.5 lentelė** Sąryšiai tarp MO tipų, granuliacijos, agregacijos lygių ir pakartotinio naudojimo

MO tipas	Granuliacijos lygis	Agregacijos lygis	Pakartotinis panaudojimas
Pagrindiniai ištekliai: paveikslėliai, audio, video, tekstas	labai geras	žemas	Pakartotinai panaudojami pagal principą „naudok toki, koks yra“.
Vienas informacijos vienetas (angl. <i>a single piece of information</i> )	geras	žemas	Pakartotinai panaudojami kaip savarankiški vienetai. Gali būti išskaidyti į pagrindinius išteklius.
Informacijos vienetų rinkinys, susietas su vienu mokymosi tikslu	vidutinis	vidutinis	Pakartotinai panaudojami kaip savarankiški vienetai. Gali būti išskaidyti į pagrindinius išteklius arba informacijos vienetus.
Pamokos, kursai, bendri mokymosi ištekliai, sudaryti iš daugelio MO, susietų su daugeliu mokymosi tikslų	prastas	vidutinis	Pakartotinio panaudojimo galimybę mažina prastas granuliacijos lygis. Gali būti pakartotinai panaudojami kaip neskaidoma visuma.
Mokymosi aplinka – turinio ir technologijos sujungimas	labai prastas	aukštas	Pakartotinio panaudojimo galimybės priklauso nuo sugebėjimo išlaikyti sąryšius tarp turinio ir technologijos.

2006 m. Meyer [Mey06] pasiūlė TRUC (angl. *Testable, Reusable Unit of Cognition*) modelį, kurį sudaro konceptų, praktinių įgūdžių ir vertinimų rinkiniai, pasižymintys tokiomis savybėmis:

1. TRUC komponentai yra kuriami remiantis viena aiškiai nustatyta idėja.

2. TRUC ir jo komponentai yra aiškiai apibrėžti ir pritaikyti daugiopam naudojimui.

3. TRUC turi vieną ar daugiau vertinimo kriterijų aibių.

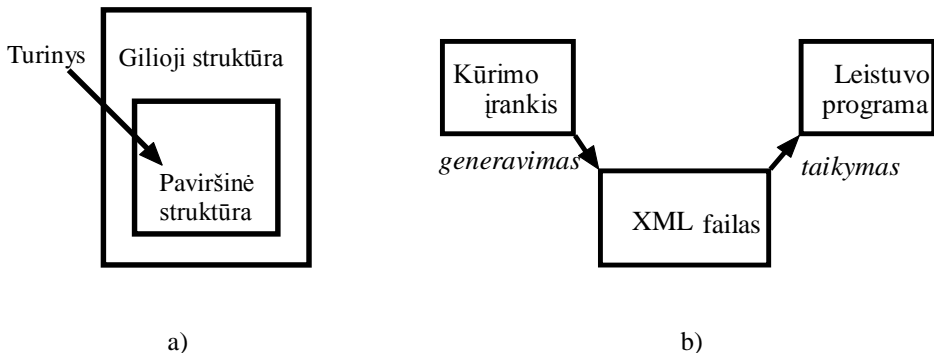
4. Nagrinėjamų temų ratas išeina už specifinio kurso ribų.

5. Jų apimtis pakankamai maža, t.y. apima vieną ar kelias pamokas.

TRUC grupuojami į hierarchinę klasterių struktūrą. TRUC, tarp kurių yra sąryšiai, priklauso vienai grupei.

Boyle [Boy06] pasiūlytą struktūrinį šablonais grįsto GMO modelį sudaro gilioji (angl. *deep structure*) ir paviršinė (angl. *surface structure*) struktūros. Gilioji struktūra apima pedagoginius ir socialinius MO aspektus, paviršinė – turinio (2.10 pav. a).

GMO elgsenos modelį sudaro trys dalys: Kūrimo įrankis (angl. *Authoring Tool*)– XML failas – Leistuvo programa (angl. *Player Program*) (2.10 pav. b). Kūrimo įrankis palaiko sąsają, leidžiančią kurti ar modifikuoti MO pagal mokytojo poreikius. Sugeneruotas rezultatas išsaugomas XML faile ir po to atkuriamas naudojant leistuvo programą. Kaip vieną siūlomo modelio didžiausių privalumų Boyle akcentuoja galimybę lanksčiai keisti XML failą naudojant kūrimo įrankį ir tokiu būdu efektyviai kurti GMO egzempliorius.



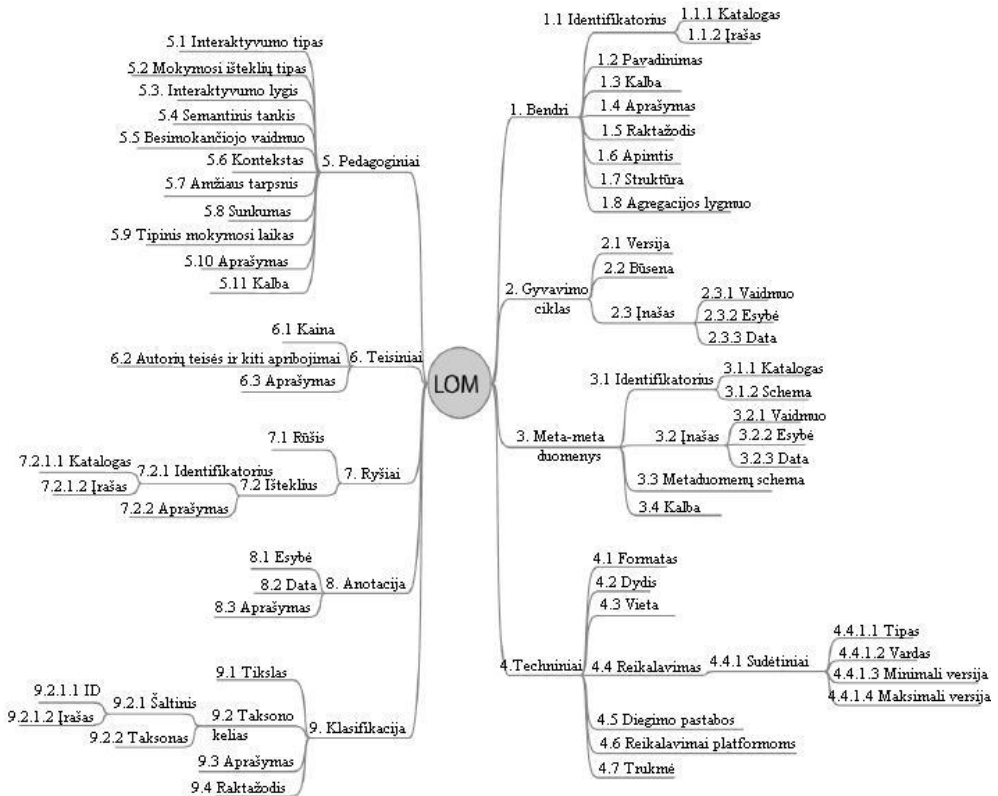
**2.10 pav.** Šablonais grįsto GMO modeliai [adaptuota iš Boy06]: a) struktūrinis, b) elgsenos

## 2.7. Mokymosi objektų standartai ir saugyklos

MO paieška saugyklose ir efektyvus pakartotinis panaudojimas glaudžiai susijęs su metaduomenų kokybės reikalavimais.

Metaduomenis NISO (angl. *National Information Standards Organization*) apibrėžia kaip struktūrizuotą informaciją, kuri aprašo, paaiškina, fiksuoja ar kitais būdais palengvina mokomųjų išteklių paiešką, naudojimą ir valdymą [BC10]. Metaduomenų reikšmės didėjimą lemia sparti e.mokymosi, MO saugyklų ir skaitmeninių bibliotekų plėtra. Roy ir kt. [RSG10] atliko lyginamąją MO metaduomenų modelių analizę, kuri parodo, kad plačiausiai naudojami IEEE LOM, Dublin Core ir CanCore standartai.

IEEE MO metaduomenų modelį (angl. *IEEE Learning Object Metadata (LOM)*) sudaro elementų hierarchija (2.11 pav.), kurioje elementų semantika apibrėžiama remiantis elementų kontekstu.

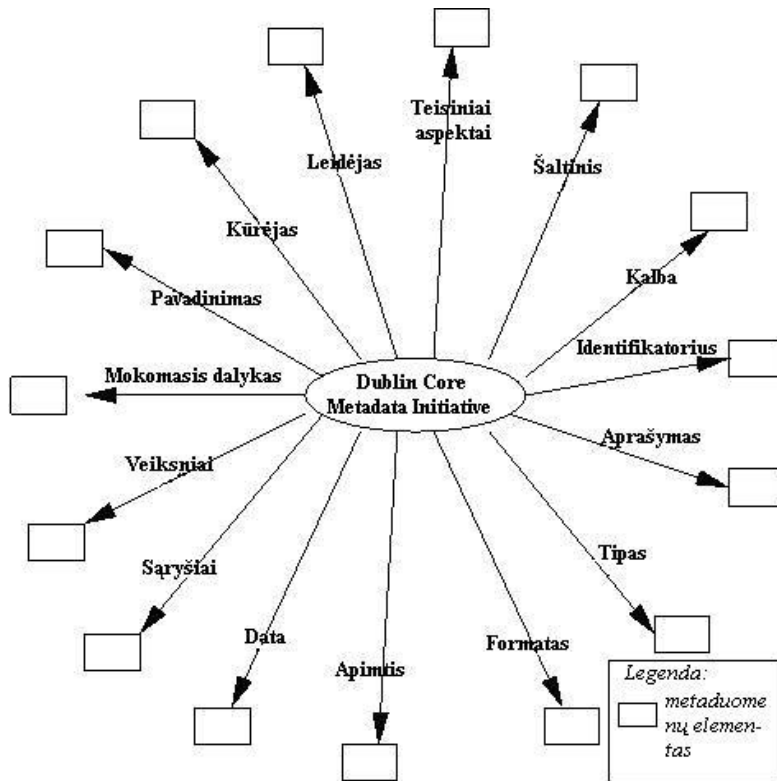


2.11 pav. IEEE MO standartų modelis [BC10]

IEEE LOM plačiai naudojamas MO saugyklose kaip tarptautinis standartas ir yra susijęs su kitomis pripažintomis specifikacijomis (IMS Global Learning Consortium, ADL (SCORM)). LOM standartai buvo adaptuoti JORUM, JISC saugyklose, Ariadne fonde, daugelyje Europos projektų. Pagrindinis IEEE LOM trūkumas yra tas, kad konceptuali duomenų schema nesusieta su abstrakčiu modeliu, sujungiančiu įvairias metaduomenų schemas ir nesuderinta su esminiais semantinės sąveikos standartais (RDF). Dar vienas trūkumas yra tas, kad skirtingose sistemose tie patys reikalavimai apibrėžiami skirtingai. Tai labai apriboja MO bendrinimo ir adaptavimo galimybes [NJN+07].

CanCore standarto, naudojamo Kanadoje, elementai atitinka IEEE LOM elementus [RSG10].

DCMI (angl. *Dublin Core (DC) Metadata Initiative*) standartai (2.12 pav.) aprašo didelę įvairiais tikslais naudojamų MO aibę ir yra apibrėžiami kaip ISO Standard 15836-2003. DCMI sudaro abstraktus modelis, metaduomenų sąvokos, Singapūro karkasas (angl. *Singapore framework*), skirtas Dublin Core taikymo profiliams (angl. *Dublin Core Application Profiles*) ir DCMI metaduomenų iškodavimo rekomendacijos RDF (angl. *Resource Description Framework*), XML ir HTML/XHTML formatams.



**2.12 pav.** Dublin Core MO standartų modelis [<http://dublincore.org/documents/dcq-rdf-xml/>]

Abstraktų DCMI modelį sudaro išteklių modelis (angl. *Resource Model*), kuris apibrėžia sąryšį tarp aprašyto ištekliu ir tarp aprašyme panaudotų išteklių. Kiekvienas DC metaduomenų aprašas aprašo vieną unikalų išteklių (vienas su vienu principas, angl. *one-to-one principle*). Aprašymų rinkinio modelis (angl. *Description Set Model*) apibrėžia sąryšius tarp atskirų išteklių. Žodyno modelis (angl. *Vocabulary Model*) aprašo abstraktaus modelio sąvokų aibę. Metaduomenų sąvokos suskirstytos į savybes, žodyno, sintaksės iškodavimo schemas ir klases. Klasės yra formalios išteklių kategorijos, kurios bendrina svarbiausias charakteristikas (bibliografinius aprašus, failų formatus). Savybės aprašo specifines ištekliu charakteristikas ir atributus, sintaksės ir žodyno iškodavimo schemas naudojamos informacijos suteikimui metaduomenų aprašuose.

Dublin Core standarte nėra pedagoginių atributų ir šis trūkumas susiaurina standarto taikymo sritį, todėl standartas naudojamas bendriesiems taikymams.

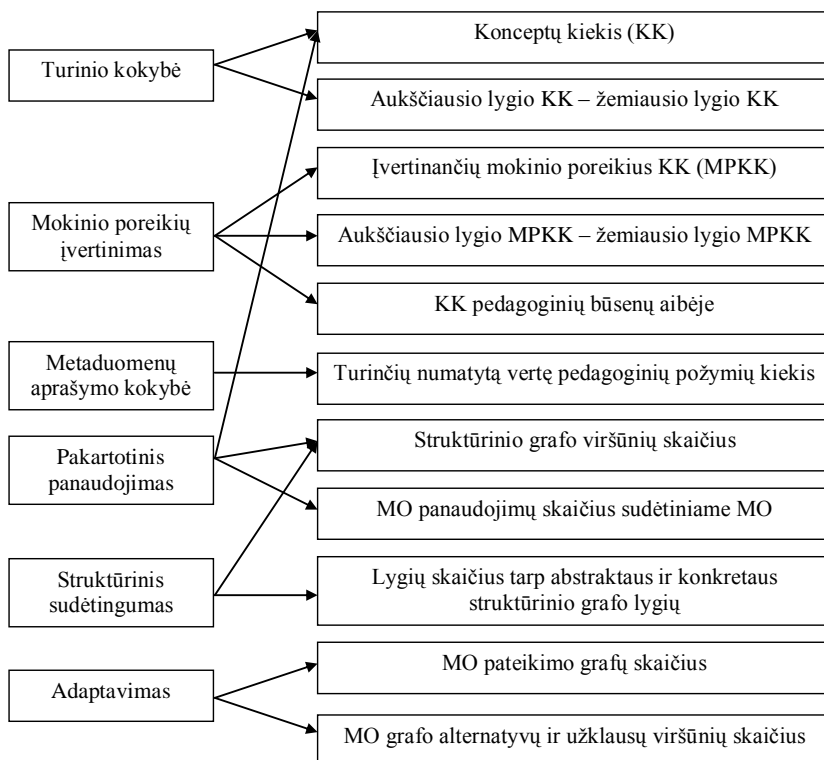
Iš Roy ir kt. [RSG10] tirtų devynių MO saugyklų, septyniose naudojamas IEEE LOM standartas, ir po vieną saugyklą, kuriose naudojami Dublin Core ir IEEE LOM Profile (SCROM) standartai.

## 2.8. Mokymosi objektų vertinimo metrikos ir instrumentai

MO paradigma glaudžiai susijusi su MO, naudojamų įvairiuose kontekstuose, kokybės įvertinimo problema.

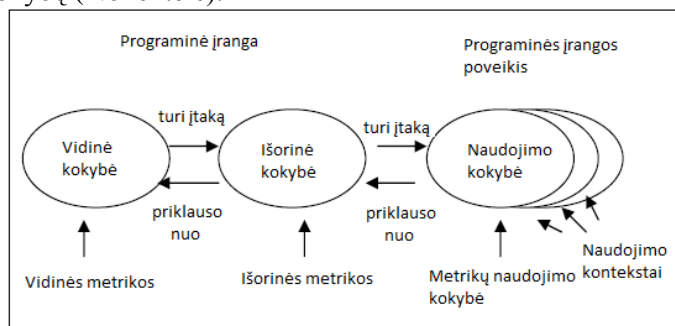
Defude ir Fargat [DF05] išskyrė 6 MO kokybės kriterijų grupes ir susiejo jas su MO kokybės metrikų aibe (2.13 pav.).

Struktūrinį agreguoto MO grafą sudaro konceptai, mokomosios veiklos ir mokinio lygis. Sudedamosios MO dalys (atominiai MO) gali būti naudojamos nepriklausomai, arba jungiamos į visumą taikant sekų, alternatyvų ir lygiagretumo operatorius. Sudaromi MO pateikimo grafai, kurie apibrėžia skirtingų agreguotų MO aibę.



2.13 pav. Metrikos, apibrėžiančios MO kokybės kriterijus [adaptuota iš DF05]

Parsons ir Ryu [PR06] susiejo skirtingus kokybės vertinimo metrikų tipus (2.14 pav.) ir ISO 9126-2 standartą papildė 3 metrikomis, įvertinančiomis mobilaus mokymosi kokybę (2.6 lentelė).



2.14 pav. Sąryšiai tarp metrikų tipų (iš ISO 9126-2) [PR06]



## 2.6 lentelė Mobilaus mokymosi kokybės metrikos

Metrika	Metrikos tikslas	Taikymo metodas	Matavimas, formali išraiška	Išmatuotos vertės interpretacija	Matavimo skalė	Matavimo tipas	Pradiniai duomenys	Vartotojai
Metafora	Ar besimokantysis turi mokymosi proceso viziją?	Apklauskos Metaforinių komponentų skaičius, kuri nurodė vartotojas	$X = A / B$ A – vartotojo nurodytų komponentų skaičius, B – kūrėjo numatytas komponentų skaičius	$0 < X < 1$ Kuo arčiau 1.0, tuo geriau	Absoliutinė	A = Count B = Count X = Count / Count	Vartotojo vadovo operacijų ataskaita	Besimokantieji, kūrėjai
Interaktyvumas	Ar besimokantysis sąveikauja su kitais mokiniais ir mokytoju?	Testavimas ir besimokančiojo elgsenos stebėjimas Sąveikų skaičius	$X = A / B$ A – besimokančiojo naudotas sąveikų skaičius, B – bendras sąveikų skaičius	$0 < X < 1$ Kuo arčiau 1.0, tuo geriau	Absoliutinė	A = Count B = Count X = Count / Count	Vartotojo vadovo operacijų ataskaita	Besimokantieji, kūrėjai
Mokymosi turinys	Ar mokinys jaučiasi naudojantis aukštos kokybės turinį?	Testavimas ir įvertinimas taikant specialią formą Mokymosi patirties atsakymų matavimas	$X = A / B$ A – maksimalus balų skaičius, B – realus balų skaičius	$0 < X \leq \text{MAX}$ Kuo arčiau MAX, tuo geriau	Absoliutinė	A = Count B = Count X = Count / Count	Vartotojo vadovo operacijų ataskaita	Besimokantieji, kūrėjai

Key ir Knaack [KK07], Leacock ir Nesbit [LN07], Akpınar [Akp08] apibrėžė MO vertinimo aspektus, taikomus įvairiuose MO vertinimo instrumentuose.

LORI (angl. *Learning Object Review Instrument*) sistemoje išskirti 9 MO vertinimo kriterijai [LN07], [Akp08]: 1) turinio kokybė (angl. *Content Quality*): MO turinys turi būti logiškai pagrįstas ir akcentuojantis svarbiausias idėjas; 2) mokymosi tikslo suderinimas (angl. *Learning Goal Alignment*) su mokymosi veiklomis, turiniu ir besimokančiojo pasiekimų bei pažangos vertinimu; 3) grįžtamasis ryšys ir adaptavimas yra labai svarbūs pritaikant MO konkretaus besimokančiojo poreikiams; 4) MO turinys turi motyvuoti mokinių siekti apibrėžtų mokymosi tikslų; 5) informacijos pateikimo stilius labai svarbus mokymosi efektyvumui; 6) naudojimo sąveika (angl. *Interaction Usability*) glaudžiai susijusi su MO vartotojo

šąsaja, apsprendžiančia sąveiką „mokinys–MO“, todėl vartotojo sąsaja turi būti nuosekli ir nuspėjama; 7) pasiekiamumas užtikrina, kad MO galima pasiekti naudojant įvairias elektronines priemones; 8) pakartotinis panaudojamumas suteikia galimybę MO panaudoti skirtinguose mokymosi kontekstuose; 9) standartų laikymasis užtikrina, kad MO bus pasiekiamas vartotojams. MO vertinimui naudojama Likerto skalė nuo 1 (žemas įvertinimas) iki 5 (aukštas įvertinimas). Tačiau reikia pažymėti ir tai, kad dažniausiai saugyklose esančius MO vertina mažas skaičius vartotojų, todėl įvertinimai gali būti nereprezentatyvūs.

MERLOT saugykloje esantys MO vertinami naudojant LORI kriterijus ir papildomai įvedant su MO standartais susietus vertinimo aspektus [Akp08]. Minėti aspektai vertinami analizuojant detalius vartotojų atsakymus į 30 klausimų. Daugiausia dėmesio skiriama trimis aspektams: turinio kokybei, potencialiam MO efektyvumui ir naudojimo patogumui.

LOEI (angl. *Learning Object Evaluation Instrument*) sukurtas apibendrinant aukščiau išvardintų sistemų galimybes ir vertina kiekvieno MO vertę taikant vientisumo, naudojamumo, turinio ir pateikimo kokybės kriterijus [Akp08].

Iš MO įvertinimų seka svarbios išvados [Akp08]: 1) MO, kuriuose numatyta daugiau mokymosi veiklų, turi reikšmingą teigiamą įtaką mokymosi išėjimams (angl. *learning outputs*); 2) geresniu grįžtamuju ryšiu ir adaptavimo galimybėmis pasižymi tie MO, kuriuose yra atsižvelgiama į besimokančiojo poreikius, t.y. įvertinamos mokinio charakteristikos (lygis, mokymosi stilius ir kt.).

Key ir Knaack [KK08] apibendrina kai kuriuos mokslinėje literatūroje pateikiamus MO vertinimo kriterijus ir jų subkategorijas: 1) interaktyvumą (konstruktyvi veikla, kontrolė, interaktyvumo lygis); 2) projektavimą (schema, personalizavimas, grafikos kokybė, esminių koncepcijų išskyrimas); 3) motyvaciją (sunkumo lygis, tema, estetika, grįžtamasis ryšys, multimedia); 4) naudojimą (visapusiškai patogus naudojimas, aiškios instrukcijos, navigacija); 5) turinį (tikslumas, kokybė).

Cuadrado ir Sicilia [CS05], Noor *ir kt.* [NYH09], Cervera *ir kt.* [CLF+09] MO pakartotinio panaudojimo vertinimui adaptavo programų inžinerijos metrikas. Klasės svertinių metodų (angl. *Weighted Method per Class*, WMC) metrika programų inžinerijoje įvertina klasės metodų sudėtingumo agregaciją, kuri panaudojama prognozuojant klasės pakartotinį panaudojimą. Klasių, turinčių daugiau metodų, panaudojimas yra labiau specifinis ir jos rečiau panaudojamos pakartotinai. WMC metrika pritaikyta MO pakartotiniam panaudojimui įvertinti, metodo analogu pasirinkus elementarius MO, iš kurių sudarytas sudėtinis MO.

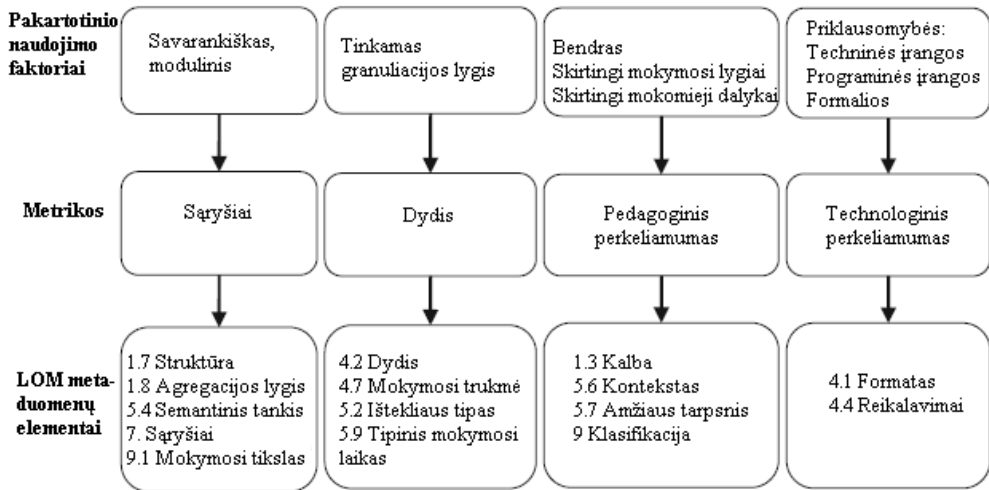
Tuomet WMC skaičiuojama pagal formulę  $WMC = \sum_{i=1}^n A_i$ , čia  $n$  – elementarių MO skaičius,  $A_i$  –  $i$ -tojo komponento agregacijos lygis pagal IEEE standartus.

Paveldėjimo medžio gylis (angl. *Depth Inheritance Tree*, DIT) metrika apibrėžia klasės paveldėjimo medžio gylį programinės įrangos karkaso viduje. Klasės, esančios paveldėjimo medyje giliau, yra sudėtingesnės ir sunkiau pakartotinai panaudojamos. MO srityje DIT metrika skaičiuojama kaip saitų gylis tarp elementarių MO. Kuo daugiau saitų, tuo MO yra sudėtingesnis ir jo pakartotinio panaudojimo galimybės mažesnės.

Objektų poravimo (angl. *Coupling Between Object*, CBO) metrika apibrėžia klasių skaičių, su kuriomis galima suporuoti pasirinktą klasę. MO srityje CBO apibrėžiama kaip MO saitų skaičius su kitais MO. Kuo mažesnis CBO, tuo lengviau MO pakartotinai panaudojamas.

Metodų sanglaudos trūkumo (angl. *Lack Cohesion of Methods*, LCOM) metrika apibrėžia iš dalies sutampančių atributų naudojimą klasės metoduose. MO srityje ši metrika apibrėžia, mokymosi tikslų, kuriuos perdengia MO, skaičių. Kuo mažesnė LCOM vertė, tuo didesnės MO pakartotinio panaudojimo galimybės.

Sanz-Rodriguez *ir kt.* [SDS11] susiejo pakartotinio panaudojimo faktorius ir metrikas su LOM metaduomenų elementais (2.15 pav.)



**2.15 pav.** Sąryšiai tarp pakartotinio naudojimo faktorių, metrikų ir LOM metaduomenų elementų [SDS11]

Chawla *ir kt.* [CGS12] pasiūlytame MO vertinimo modelyje LOQES pakartotinį panaudojimą įvertina sąryšio (angl. *cohesion*), poravimo, dydžio, perkeliamumo (angl. *portability*) ir supratimo sunkumo (angl. *Difficulty of Comprehension*) metrikos. Sąryšių skaičius priklauso nuo MO semantinio tankio, agregacijos lygio, sąryšių tarp mokymosi tikslų ir konceptų skaičiaus. Poravimas matuoja sąryšius tarp skirtingų modulių. Mažesnė poravimo reikšmė užtikrina geresnį pakartotinį panaudojimą. Pagal dydį MO skirstomi į 5 grupes nuo labai mažų (atominiai resursai, dydžio reikšmė lygi 5) iki labai didelių (labai platus kursai, dydžio reikšmė lygi 1). Perkeliamumo metrika apibrėžia galimybę naudoti MO daugelyje kontekstų. Išskiriamas technologinis ir pedagoginis perkeliamumas. Technologinis perkeliamumas priklauso nuo MO pateikimo formos ir reikalavimų techninei ir programinei įrangai. Pedagoginis perkeliamumas yra vertikalus ir horizontalus. Vertikalus pedagoginis perkeliamumas apibrėžia pakartotinio panaudojimo galimybę įvairiuose mokymosi lygiuose (pradiniam, pagrindiniam, aukštesniajame), horizontalus – tarp skirtingų mokomųjų dalykų. Sunkiai suprantamas MO atliekantis agregaciją bus mažai pakartotinai panaudojamas.

Sėrikovienė [Sėr13] sukūrė MO ekspertinio kokybės vertinimo metodą, pagrįstą neraiškiųjų skaičių teorija bei skaliarizavimo metodu, kai MO kokybė vertinama ne tik pagal kriterijų svarbą, bet ir pagal jų tarpusavio sąryšius.

## 2.9. Bendrieji informatikos mokymosi iššūkiai 21 amžiuje

Informatikos mokslas vystosi labai sparčiai, todėl vis sunkiau tiksliai apibrėžti jo turinį ir ribas. ACM siūlo taikyti informatikos mokymosi programos vidurinėje mokykloje modelį, kuriame informatika apibrėžiama kaip mokslas, nagrinėjantis kompiuterių ir algoritminius procesus, techninės ir programinės įrangos projektavimą, taikymus ir įtaką visuomenei. Informatika apima programavimo, techninės ir programinės įrangos projektavimo, kompiuterių tinklų, grafikos, duomenų bazių ir informacijos paieškos, kompiuterių saugos, programavimo kalbų, logikos, programavimo paradigmų, perėjimų tarp skirtingų abstrakcijos lygmenų, dirbtinio intelekto, kompiuterių taikymo ribų nustatymo, informacinių technologijų ir informacinių sistemų taikymų ir socialinių saugumo aspektų (saugumas ir privatumas internete, autorių teisių apsauga) sritis. Iš kitos pusės, informatika glaudžiai susijusi su teoriniais robotikos pagrindais, kompiuterine rega, intelektualiomis sistemomis, bioinformatika. Mokslininkai intensyviai dirba trimis kryptimis: 1) ieško efektyvių informatikos problemų sprendimo būdų, 2) projektuoja ir kuria programinę įrangą, 3) atlieka tyrimus, kurių tikslas – nauji kompiuterių panaudojimo būdai [EAG+08].

Informatikos mokslo pasiekimams sparčiai skverbiantis į visas gyvenimo sritis, būtina keisti informatikos dėstymą vidurinėje mokykloje. Pokyčiai būtini dėl žemiau išvardintų priežasčių ir iššūkių [EAG+08], [FL09], [HLR11], [IIT+12]:

1. Kompiuterinės sistemos vis plačiau naudojamos transporto, sveikatos, farmacijos ir kitose pramonės šakose. Mokiniai, mokyklose gavę fundamentalius informatikos pagrindus, bus ne tik išprusę technologijų vartotojai, bet ir novatoriai, gebantys panaudoti kompiuterius gyvenimo kokybės gerinimui.

2. XXI a. daugelio profesijų atstovams keliami reikalavimai būti ne tik srities profesionalais, bet ir turėti informatikos mokslo žinių, kad galėtų savo srityje tinkamai naudoti šiuolaikines technologijas.

3. Sparčiai vystantis technologijoms suteikiama vis daugiau elektroninių viešųjų paslaugų, sparčiai vystosi elektroninė prekyba ir verslas. Kadangi vis daugiau informacijos perduodama ir saugoma skaitmeniniu formatu, susiduriama su saugumo ir privatumo problemomis, kurių sprendimas reikalauja pagrindinių algoritmų supratimo. Mokydamiesi informatikos, mokiniai mokosi logiškai ir algoritmiškai mąstyti, kūrybiškai panaudoti problemų sprendimo būdus. Visos išvardintos sąvokos ir įgūdžiai yra taikomi daugelyje kontekstų, pradedant tiksliaisiais mokslais ir inžinerija, baigiant verslu bei humanitariniais mokslais.

4. Informatika moko uždavinių sprendimo būdų. Sprendžiant problemą, pirmiausia reikia aiškiai ir vienareikšmiškai apibrėžti reikalavimus. Daugelis informatikos specialistų, projektuodami ir realizuodami sudėtingas sistemas, dirba su daugelio sričių ekspertais, kartu apsvarstydami prioritetus ir apribojimus. Tik tuomet, kai problema yra aiškiai apibrėžta, gali būti kuriamas sprendimas: parenkama techninė įranga, pritaikomi arba sukuriama algoritmai, jie realizuojami ir

testuojami. Reikia atkreipti dėmesį į tai, kad dažnai į galutinę sistemą reikia integruoti modifikuotas anksčiau sukurtas sistemas. Visuose išvardintuose etapuose informatikos specialistas galvoja apie optimalų kompiuterio darbo laiko ir atminties resursų panaudojimą, be to, jis rūpinasi sistemos patikimumu, draugiška vartotojo sąsaja, priežiūra, korektišku veikimu sprendžiant inžinerines, verslo, mokslo problemas. Geriausio galimo sprendimo paieška reikalauja intensyvios analizės ir kūrybiškumo.

5. Daug informatikos specialistų dirba ne tik aukštųjų technologijų įmonėse, bet ir kitose mokslo srityse, kur naudojamas kompiuterinis modeliavimas, reikia apdoroti ir vizualizuoti didelius duomenų kiekius. Norint suprasti sudėtingų sistemų veikimo principus ir apibrėžti taisykles, naudojami kompiuteriniai modeliai. Informatikos mokslo pasiekimai taip pat padeda ieškoti atsakymų į klausimą „Kaip dirba žmogaus protas?“, tiriant žmogaus genomą.

Išvardintos priežastys ir iššūkiai rodo, kad informatikos mokymuisi mokyklose turi būti skiriama daugiau dėmesio ir siekiama, kad mokiniai naudotų kompiuterius ne tik kaip darbo įrankį, bet ir suteikti jiems fundamentalių žinių apie kompiuterių darbo principus, programinės įrangos projektavimą bei kūrimą.

Toliau nagrinėjami informatikos mokymosi aspektai, susiję su mokinių kompiuterinio mąstymo įgūdžių ir gebėjimų (analizuoti realaus gyvenimo problemas, jas abstrahuoti ir spręsti panaudojant naujausias technologijas) ugdymu mokantis programavimo [FL09], [BS11], [HLR11], [IIT+12].

## 2.10. Programavimo mokymosi iššūkiai

Daugelyje šaltinių, nagrinėjančių programavimo mokymąsi, nurodomos priežastys, sukeliančios daugiausia sunkumų pradedantiesiems programuotojams. Apibendrinant išskiriamos 4 tarpusavyje susijusios problemų grupės:

1. **Pedagoginės problemos:** (i) naudojami mokymosi modeliai neatitinka mokinių mokymosi poreikių [GM07], [Ala12], [PSM+07], [SSH+09], [BEP+09], [LY11]; (ii) mokymasis dažnai negali būti individualizuojamas dėl didelio mokinių skaičiaus klasėje [GM07], [SGM10]; (iii) tinkamo mokymosi konteksto pasirinkimo problemos [Fig06], [KS07], [CC10], [Pea10]; (iv) motyvacijos problemos [AG13], [CTT10], [GGL+12], [JCS09], [KPN08], [Lea08], [LLY10], [Pea10], [SHL+13].

2. **Pažinimo problemos:** (i) mokantis programuoti operuojama aukšto lygmens abstrakcijomis [GM07], [Chu07], [CMF+09], [CTT10], [Ala12]; (ii) programavimas reikalauja žinių ir praktinių problemų sprendimo įgūdžių suderinamumo [RRR03], [GM07], [PSM+07], [CMF+09], [SSH+09], [LY11], [SPJ+11], [SPJ+12]; (iii) atsiranda skirtumai tarp programų supratimo ir jų sąsajų su iš anksto sudarytais programų modeliais [MR02], [RRR03], [CMF+09], [CTT10]; (iv) objektinio ir struktūrinio programavimo paradigms skirtumai [Sch02], [MR02], [RRR03], [SH06], [CMF+09], [CTT10]; (v) programavimo kalbų sintaksė yra sudėtinga, pritaikyta profesionaliam naudojimui [GM07], [CMF+09], [CTT10].

3. **Mokymosi turinio problemos:** (i) programavimas yra dinamiškas, bet mokymuisi dažniausiai naudojama statinė medžiaga (angl. *static materials*) [GM07], [PSM+07], [NS09], [VBH13]; (ii) turinio adaptavimas pagal konkretaus mokinio

poreikius [AG03], [GA03], [LYW05], [AHH12]; (iii) turinio vizualizacija [Chu07], [MV07], [KPN08], [MAR08], [RöB10], [AHH11], [MT12], [Ala12];

4. **Technologinės problemos:** (i) mažai programavimo įrankių, kuriuos galima adaptuoti ir apibendrinti, nes dauguma jų yra sukurti lokalioms problemoms spręsti (kurioje nors institucijoje, kuriam nors kursui) arba tyrimams, kuriuose nagrinėjami nauji požiūriai (angl. *approaches*) ir metodai, atlikti [AG03], [PSM+07], [CMF+09]; (ii) tradicinės mokymosi valdymo sistemos (angl. *learning management systems*) neapima visų programavimo mokymosi kontekstų [CMF+09], [ADP+11]; (iii) interaktyvus programavimo mokymasis susiduria su iššūkiais, atsirandančiais dėl dalyko specifikos (aukščiau išvardintų pažinimo ir pedagoginių problemų) [BBC04], [GC06], [MV07], [GM07], [CMF+09], [BB09], [LLY10], [APH+11], [CAL12], [DAB12].

Aptarsime tyrėjų siūlomus išvardintų problemų sprendimo būdus, kurie apima motyvacijos, mokymosi modelių ir tikslų modelius, mokymosi objektų ir aplinkų kūrimą bei adaptavimą įvairiuose programavimo mokymosi kontekstuose.

### 2.10.1. Motyvacijos modeliai

Mokymasis ir motyvacija yra vieni sudėtingiausių žmogaus elgsenos aspektų. Law *ir kt.* [LLY10] pasiūlė modelį, kuriame išskirti svarbiausi mokymosi motyvacijos faktoriai. Vidiniai faktoriai sutelkia dėmesį į individą, tuo tarpu išoriniai susieti su išorine aplinka (2.7 lentelė).

Saviveiksmingumas (angl. *self-efficacy*) yra labai svarbus motyvacijos elementas, susijęs su kompetencijomis, tikslo siekimu, modeliavimu, problemų sprendimu, atlygio neapibrėžtumais (angl. *reward contingencies*), srities pažinimu, savireguliacija, socialiniais palyginimais.

### 2.7 lentelė Motyvacijos faktoriai (pagal [LLY10])

Vidiniai faktoriai	Išoriniai faktoriai
Individo nuostatos ir lūkesčiai Ambicijos ir tikslai	Aiški kryptis Atlygis ir pripažinimas Nuobaudos Socialinis spaudimas ir konkurencija

Law *ir kt.* [LLY10] atliktas empirinis tyrimas rodo, kad didžiausią poveikį mokymosi efektyvumui turi individo nuostatos ir lūkesčiai, ambicijos ir tikslai bei socialinis spaudimas ir konkurencija.

Kitų tyrėjų siūlomos motyvaciją skatinančios priemonės: žaidimais pagrįsti mokymosi modeliai (angl. *game-based learning models*) [JCS09], [SHL+13]; programų elgsenos vizualizacija [Chu07], [Pea10], [Ala12]; programavimo poromis modelio (angl. *pair-programming model*) taikymas [DSS+08], [SMG11]; robotais pagrįstos mokymosi aplinkos [CAC12], [HS12], [Tou12].

### 2.10.2. Mokymosi tikslų modelis

Starr *ir kt.* [SMS08] programavimo mokymosi tikslų ištyrimui, specifikuojant ir tobulinimui siūlo taikyti Bloom taksonomija pagrįstą modelį, sudarytą iš trijų metalygmenų: 1) įsiminimo ir pagrindų supratimo (pradedančiųjų grupė); 2) naudojimo ir kompetentingo taikymo (vidurinioji grupė) ir 3) projektavimo, kūrimo ir kritikos (pažengusiųjų grupė). Kiekviename lygmenyje yra dvi fazės: mokymosi

artefakto pateikimas ir jo paaiškinimas arba analizė. Mokiniai, siekdami mokymosi tikslo, nuosekliai pereina visus tris lygius (2.16 pav.).

Metalygmuo	Fazė	
	Pateikimas	Paaiškinimas
Pradedančiųjų grupė	Atkūrimas	Supratimas
Viduriniųjų grupė	Taikymas	Analizė
Pažengusiųjų grupė	Sintezė	Įvertinimas

**2.16 pav.** Bloomo taksonomijos metalygmenų struktūra [SMS08]

Naudojant pateiktą struktūrą aiškiai ir tiksliai specifikuojami įvairių lygių (mokymosi objekto, pamokos, kurso) mokymosi rezultatai (angl. *learning outcomes*).

### 2.10.3. Programavimo mokymosi modeliai

Programavimo mokymosi modeliu laikysime konceptualią struktūrą, aprašančią, kokios sisteminės procedūros turi būti atliekamos organizuojant mokymosi procesą, kad būtų pasiekti specifiniai programavimo mokymosi tikslai.

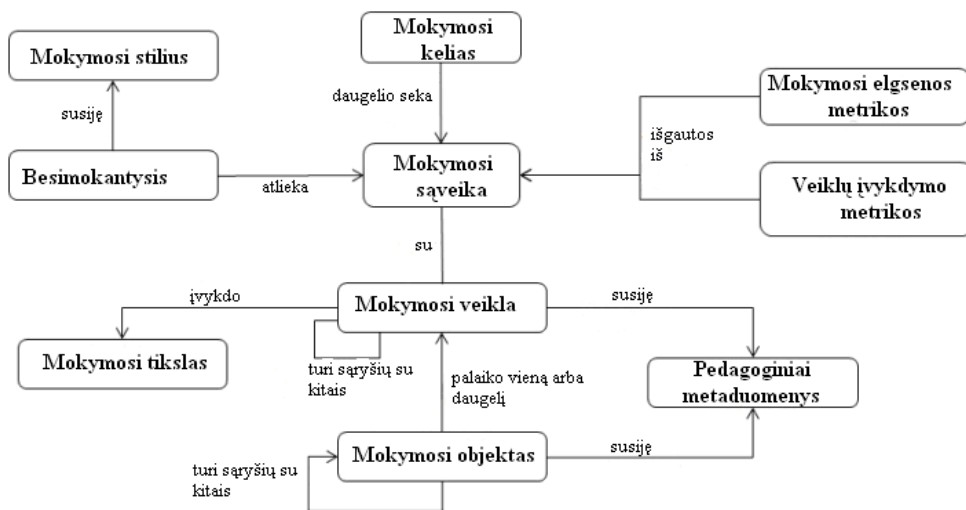
Schulte [Sch02] pasiūlytas pedagoginis informatikos mokymosi bendrinis modelis (angl. *pedagogical framework*) remiasi trimis principais, sujungiančiais teoriją ir praktiką: 1) kontekstualizuotas mokymasis susieja modeliavimo ir programavimo įgūdžius, sukuriama autentiškas kontekstas, kuriame modeliuoti mokomasi žingsnis po žingsnio; 2) laipsniškas formalizmo didėjimas leidžia mokiniams panaudoti jų ankstesnes žinias aprašant struktūras ir procesus, mokymasis prasideda nuo konkrečių ir neformalių dalykų, kurie palapsniui tampa vis labiau formalūs ir abstraktūs; 3) mokomasi tik modeliuojamo pavyzdžio kontekste naudojamų sintaksės detalių ir programavimo įgūdžių. Mokymosi procese išskiriami trys etapai: 1) besimokantieji supažindinami su kalba, įrankiais ir modeliais, nagrinėja realaus gyvenimo pavyzdžius; 2) studentai kuria modelius, remdamiesi pavyzdžiais, ir juos realizuoja; 3) dirbama poromis arba mažose grupėse savarankiškai kuriant programas.

Hadjerrouit [Had08] [Had09] mišraus programavimo mokymosi modelį (angl. *blended learning model*) sudaro trys fazės. Konceptualizacijos fazėje studentų ankstesnės žinios susiejamos su naujai išmoktomis sąvokomis naudojant palyginimus, sudarant sąryšius tarp žinomų ir naujai išmoktų sąvokų, išplečiant panaudojimo kontekstą, klasifikuojant sąvokas pagal jų bendrus požymius. Konstravimo fazėje kuriamos programos autentiškiems uždaviniams spręsti. Šioje fazėje žinios susiejamos su problemų sprendimo įgūdžiais, ankstesnių sprendimų pakartotiniu panaudojimu, sprendimų palyginimu, programų elgesio prognozavimu, daugelio sprendimų generavimu. Dialogo fazėje naudojant „paaiškink, aprašyk, diskutuok, įvertink, integruok, išplėsk, apibendrink“ pedagogines strategijas, įvertinamos studentų žinios ir įgūdžiai. Priklausomai nuo aplinkybių, ši fazė gali būti vykdoma lygiagrečiai su konceptualizacijos ir konstravimo fazėmis.

Lau ir Yuen [LY11] siūlo modelį, susiejantį mokinio charakteristikas (lytis, mokymosi stilius, mentaliniai modeliai, akademiniai gebėjimai, instrukcijų vykdymas) su programavimo charakteristikomis. Empiriniu tyrimu nustatyta, kad

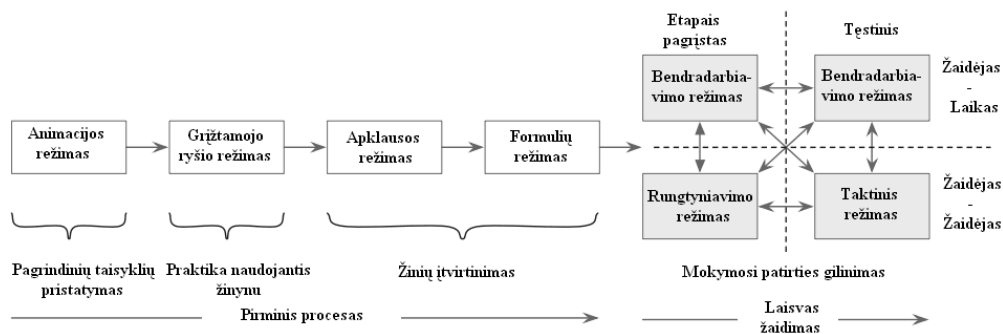
egzistuoja teigiami ryšiai: 1) tarp lyties ir mentalinių modelių (efektas 0.2, reikšmingumas  $p < 0.05$ ); 2) tarp mokymosi stilių ir programavimo charakteristikų (efektas 0.3, reikšmingumas  $p < 0.01$ ); 3) tarp mentalinių modelių ir programavimo charakteristikų (efektas 0.21, reikšmingumas  $p < 0.01$ ).

Mokymosi kelio duomenų modelis (angl. *Learning Path Data Model*) [CAL12] susieja pedagoginius, pažinimo ir technologinius programavimo mokymosi aspektus panaudojant e.mokymosi pedagoginius scenarijus, apimančius visą mokymosi procesą (2.17 pav.).



2.17 pav. Mokymosi kelio duomenų modelis [CAL12]

Schäfer ir kt. [SHL+13] pateikia žaidimais pagrįstų modelių mokymosi proceso schemą (2.18 pav.)



2.18 pav. Žaidimais pagrįstų modelių mokymosi proceso schema [SHL+13]

2.8 lentelėje pateikiama informacija kaip siūlomuose modeliuose atsispindi aukščiau įvardinti programavimo mokymosi iššūkiai.

Išanalizavus lentelėje pateiktus duomenis galima daryti išvadą, kad nei vienas siūlomas modelis pilnai neperdengia programavimo mokymosi srities probleminių vietų. Pagrindinėmis spręstinomis problemomis lieka mokymosi individualizavimas, tinkamas mokymosi konteksto parinkimas, žinių ir praktinių problemų sprendimo



įgūdžių suderinamumas, programų supratimas ir jų sąsajos su iš anksto sudarytais programų modeliais, programavimo kalbų ypatumai, turinio vizualizacija, turinio ir įrankių adaptavimas ir apibendrinimas.

## 2.8 lentelė Nagrinėtų modelių ir programavimo mokymosi iššūkių sąryšiai

Iššūkiai	Modelis	[Sch02]	[Had08] [Had09]	[LY11]	[CAL12]	[SHL+13]
<b>Pedagoginiai:</b>						
• Mokinių mokymosi poreikių patenkinimas		±	±	+	+	±
• Mokymosi individualizavimas		±	±	+	+	-
• Tinkamai parinktas mokymosi kontekstas		+	±	-	±	±
• Motyvacijos problemos		±	±	-	-	±
<b>Pažintiniai:</b>						
• Žinių ir praktinių problemų sprendimo įgūdžių suderinamumas		+	+	-	-	±
• Programų supratimas ir jų sąsajos su iš anksto sudarytais programų modeliais		+	±	-	-	±
• Programavimo kalbos ypatumai		±	±	-	-	±
<b>Mokymosi turinio pateikimo:</b>						
• Turinio adaptavimas pagal konkretaus mokinio poreikius		-	±	±	±	-
• Turinio vizualizacija		-	±	-	±	+
<b>Technologiniai:</b>						
• Galimybė adaptuoti ir apibendrinti naudojamus įrankius		-	±	-	±	±
• Mokymosi kontekstų perdengimas		±	±	±	±	±
• Dalyko specifika		±	±	±	±	±

(žymėjimai: „+“ – stipriai išreikšta, „±“ – vidutiniškai išreikšta, „-“ – neišreikšta)

### 2.10.4. Vertinimo modeliai

Viena iš svarbiausių efektyvaus mokymosi sąlygų yra tinkamai parinktas vertinimo modelis, kuris suteikia informaciją apie besimokančiojo sukauptą mokymosi patirtį, pasiekimus ir daromą pažangą, ir padeda mokiniui įvertinti savo silpnąsias ir stipriąsias puses, kelti naujus mokymosi tikslus. Mokytojui vertinimas padeda išvelgti mokinio mokymosi galimybes ir poreikius, diferencijuoti ir individualizuoti darbą parenkant tinkamą ugdymo turinį ir metodus.

Mokslinėje literatūroje aprašyti informatikos vertinimo modeliai paremti:

- šešių pažinimo lygių (žinios, supratimas, pritaikymas, analizė, sintezė, įvertinimas) Bloom taksonomija [TLW+08], [SHD09];

- SOLO (angl. *Structure of the Observed Learning Outcome*) taksonomija, sudaryta iš dviejų tarpsnių: paviršinio (kiekybinio) mokymosi, kurį sudaro ikistruktūrinis, vienstruktūrinis ir daugiastuktūrinis lygmenys ir giluminio (kokybinio) mokymosi, sudaryto iš sąryšinio ir išplėstojo abstrakto lygmenų [SHD09];

- konceptų žemėlapiu kūrimu ir vertinimu [KH08].

Visi siūlomi vertinimo modeliai papildo vienas kitą ir yra orientuoti į skirtingo pasirengimo mokinių gebėjimų vertinimą.

### 2.10.5. Mokymosi aplinkos ir įrankiai

Norint apibrėžti mokymosi aplinkos charakteristikas ir tinkamai jas suprojektuoti, labai svarbu korektiškai nustatyti besimokančiųjų mokymosi ypatumus ir sunkumus bei jų priežastis: bendruosius problemų sprendimo, algoritmų supratimo, modifikavimo ir naujų kūrimo, programavimo technikų naudojimo įgūdžius. Svarbiausios charakteristikos, apibūdinančios efektyvią mokymosi aplinką [GM07]: 1) kiekvieno besimokančiojo žinių lygio nustatymas ir nuolatinis atnaujinimas bei dominuojančio mokymosi stiliaus nustatymas suteikia galimybę mokymąsi padaryti efektyvesnę ir labiau individualizuotą; 2) programavimo modelių naudojimas (uždavinio skaidymas į smulkesnes dalis, galimybė besimokančiajam pritaikyti žinias analogiškose situacijose) ugdo problemų sprendimo gebėjimus; 3) žaidimai stiprina mokymosi motyvaciją, ugdo problemų sprendimo gebėjimus; 4) algoritmų kūrimo įrankių naudojimas aplinkoje suteikia galimybes besimokančiajam įgyti algoritmų sudarymo bei realizavimo žinių ir įgūdžių.

Toliau aptariamos programavimo mokymosi aplinkų ir įrankių taksonomijos.

Kelleher ir Pausch pasiūlyta programavimo aplinkų ir kalbų taksonomija [KP05] aplinkas suskirsto į dvi grupes:

**A. Mokymo sistemos** (angl. *Teaching Systems*), kurių tikslas yra pagalba besimokantiems programuoti. Šiose sistemose yra įdiegti programavimo įrankiai, kurie išryškina svarbiausius programavimo proceso aspektus. Dauguma mokymo sistemų turi daug panašumų su universaliomis programavimo kalbomis (angl. *general-purpose languages*) ir yra susietos su programavimo mechanika (angl. *mechanics of programming*): komandų pateikimu kompiuteriui ir jo atliekamų veiksmų supratimu. Programavimo mechanikos problemos mokymo sistemose sprendžiamos įvairiais būdais: 1) supaprastinant programavimo kalbą, 2) pritaikant kalbą specifiniams srities uždaviniams spręsti, 3) įvedant automatinę sintaksės klaidų taisymą, 4) surandant alternatyvius programų kodo pateikimo būdus: programų kodo konstravimą iš grafinių ar fizinių objektų, programų kūrimą naudojant sąsajos veiksmus (pvz., mygtukų paspaudimus, judėjimą erdvėje) komandų sekų sudarymui, programų kūrimui taikyti daugelį mechanizmų (programų specifikuojimą ir pateikimą įvairiais formatais), 5) taikant programų struktūrizavimą, kai panaudojami esami ir kuriami nauji programavimo modeliai, 8) didinant supratimą apie programų veikimą stebint programų vykdymą, naudojant mikropasaulių galimybes ir kuriant programų vykdymo modelius. Sistemose mokymosi palaikymas (angl. *learning support*) realizuojamas naudojant sąveiką tinklu (angl. *networked interaction*), numatant motyvuojantį kontekstą.

**B. Įgalinančios sistemos** (angl. *Empowering Systems*), kurių galimybėmis naudodamiesi besimokantieji gali suprojektuoti ir realizuoti jiems reikalingas programas. Šiose sistemose dėmesys sutelkiamas į programavimo kalbą ir metodus, suteikiančių besimokantiems galimybes įgyvendinti savo sumanymus, kūrimą. Kodo sudėtingumo problemos šiose sistemose sprendžiamos: 1) nurodant veiksmus ir sąlygas sąsajoje nekeičiant programos kodo, 2) tobulinant programavimo kalbas, didinant jų suprantamumą ir draugiškumą besimokančiajam. Įgalinančiose sistemose realizuojamas universaliųjų programavimo kalbų pritaikymas specifinėms sritims, kai besimokantieji gali kurti, tyrinėti ir eksperimentuoti su įvairių pažinimo sričių

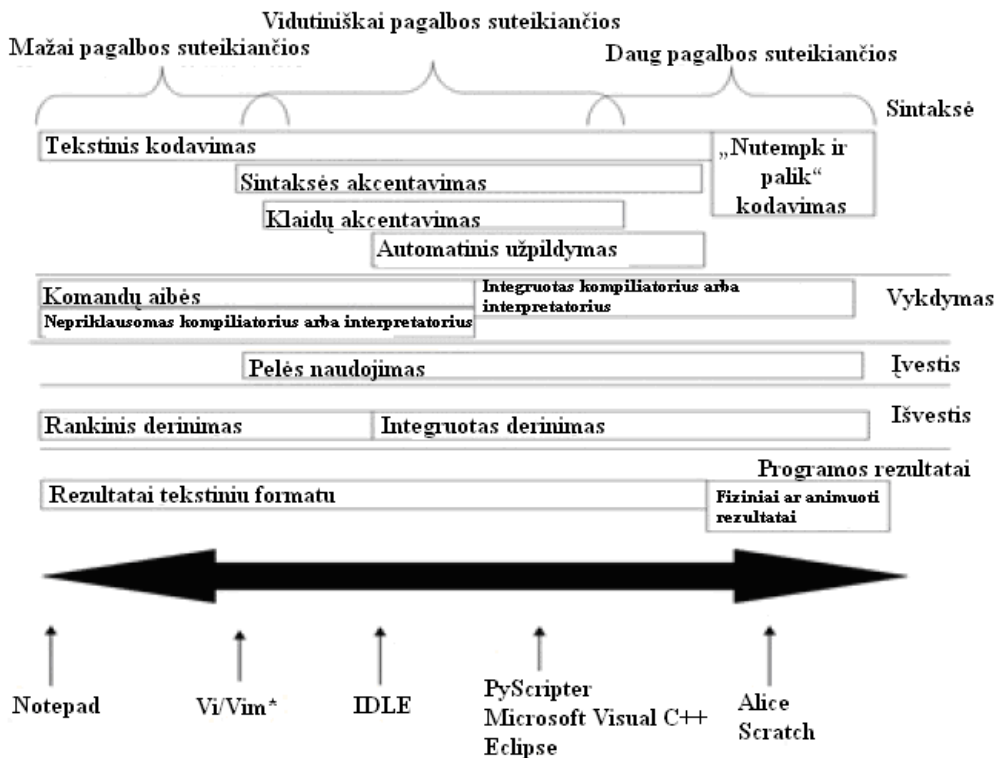
modeliais. Įdomios pramoginės veiklos (angl. *entertaining activities*) didina mokymosi motyvaciją. [KP05] išskyrė svarbiausius sistemų atributus (2.9 lentelė).

## 2.9 lentelė Programavimo mokymosi sistemų atributai [KP05]

<b>Programavimo stilius</b> (angl. <i>Style of Programming</i> )	<ul style="list-style-type: none"> <li>• Procedūrinis (angl. <i>procedural</i>)</li> <li>• Funkcinis (angl. <i>functional</i>)</li> <li>• Pagrįstas objektų naudojimu (angl. <i>object-based</i>)</li> <li>• Objektinis (angl. <i>object-oriented</i>)</li> <li>• Pagrįstas įvykiais (angl. <i>event-based</i>)</li> <li>• Pagrįstas būsenų automatų naudojimu (angl. <i>state-machine based</i>)</li> </ul>
<b>Programavimo konstruktai</b> (angl. <i>Programming Constructs</i> )	<ul style="list-style-type: none"> <li>• Sąlygos sakiniai (angl. <i>conditional</i>)</li> <li>• Ciklai FOR (angl. <i>for loops</i>)</li> <li>• Ciklai WHILE (angl. <i>while loops</i>)</li> <li>• Kintamieji (angl. <i>variables</i>)</li> <li>• Parametrai (angl. <i>parameters</i>)</li> <li>• Procedūros / metodai (angl. <i>procedures/methods</i>)</li> <li>• Vartotojo apibrėžti duomenų tipai (angl. <i>user-defined data types</i>)</li> <li>• Išankstinės ir galutinės sąlygos (angl. <i>pre and post conditions</i>)</li> </ul>
<b>Kodo pateikimas</b> (angl. <i>Representation of Code</i> )	<ul style="list-style-type: none"> <li>• Tekstas (angl. <i>text</i>)</li> <li>• Paveikslėliai (angl. <i>pictures</i>)</li> <li>• Schema (angl. <i>flow chart</i>)</li> <li>• Animacija (angl. <i>animation</i>)</li> <li>• Formos (angl. <i>forms</i>)</li> <li>• Baigtinis automatas (angl. <i>finite state machine</i>)</li> <li>• Fiziniai objektai (angl. <i>physical objects</i>)</li> </ul>
<b>Programų kūrimas</b> (angl. <i>Construction of Programs</i> )	<ul style="list-style-type: none"> <li>• Kodo rašymas (angl. <i>typing code</i>)</li> <li>• Grafinių objektų sujungimas (angl. <i>assembling graphical objects</i>)</li> <li>• Veiksmų rodymas (angl. <i>demonstrating actions</i>)</li> <li>• Pasirinkimas / Formos pildymas (angl. <i>selecting/form filling</i>)</li> <li>• Fizinių objektų sujungimas (angl. <i>assembling physical objects</i>)</li> </ul>
<b>Programų supratimo palaikymas</b> (angl. <i>Support to Understand Programs</i> )	<ul style="list-style-type: none"> <li>• Ankstesnė patirtis (angl. <i>back stories</i>)</li> <li>• Klaidų taisymas, derinimas (angl. <i>debugging</i>)</li> <li>• Fizinė interpretacija (angl. <i>physical interpretation</i>)</li> <li>• Sugeneruoti pavyzdžiai (angl. <i>generated examples</i>)</li> </ul>
<b>Sintaksės klaidų vengimas</b> (angl. <i>Preventing Syntax Errors</i> )	<ul style="list-style-type: none"> <li>• Galiojančių parinkčių pasirinkimas (angl. <i>selection from valid options</i>)</li> <li>• Į sintaksę orientuotas taisymas (angl. <i>syntax directed editing</i>)</li> <li>• Grafinių objektų tempimas į tinkamas vietas (angl. <i>dropping only in valid location</i>)</li> <li>• Geresni pranešimai apie klaidas (angl. <i>better syntax error messages</i>)</li> </ul>
<b>Tinkamų programavimo kalbų projektavimas</b> (angl. <i>Designing Accessible Languages</i> )	<ul style="list-style-type: none"> <li>• Srities ribų apibrėžimas (angl. <i>limit the domain</i>)</li> <li>• Į vartotoją orientuotų raktažodžių parinkimas (angl. <i>select user-centered keywords</i>)</li> <li>• Nereikalingų skyrybos ženklų pašalinimas (angl. <i>remove unnecessary punctuation</i>)</li> <li>• Natūralios kalbos naudojimas (angl. <i>use natural language</i>)</li> <li>• Pertekliško pašalinimas (angl. <i>remove redundancy</i>)</li> </ul>
<b>Bendravimo palaikymas</b> (angl. <i>Support Communication</i> )	<ul style="list-style-type: none"> <li>• Du ir daugiau besimokančiųjų manipuliuoja ta pačia programa būdami vienoje klasėje (angl. <i>side by side</i>)</li> <li>• Bendras manipuliavimas tinkle (angl. <i>network-shared manipulation</i>)</li> <li>• Tinkle bendrinami rezultatai (angl. <i>network-shared results</i>)</li> </ul>

<b>Uždavinio pasirinkimas</b> (angl. <i>Choice of Task</i> )	<ul style="list-style-type: none"> <li>• Motyvuojantis (angl. <i>fun and motivating</i>)</li> <li>• Naudingas (angl. <i>useful</i>)</li> <li>• Mokomasis (angl. <i>educational</i>)</li> </ul>
---	--

Dillon ir kt. [DAB12] programavimo mokymosi sistemas klasifikuoja pagal suteikiamos pagalbos besimokančiajam lygį. Mažai pagalbos suteikiančiose sistemose (angl. *low assistive*) vartotojas rašo tekstinį programos kodą, komandas ir, naudodamas nepriklausomus kompiliatorius ir interpretatorius, rankiniu būdu derina programą. Programos vykdymo rezultatas dažniausiai pateikiamas tekstiniu formatu. Vidutiniškai pagalbos suteikiančiose sistemose (angl. *moderately assistive*) besimokantysis turi daugiau galimybių. Sistemose naudojama: sintaksės išryškinimas (angl. *syntax highlighting*), klaidų išryškinimas (angl. *error highlighting*), automatinis užbaigimas (angl. *auto completion*), integruotas kompiliavimas/vykdymas (interpretavimas), integruotas programos derinimas. Šiose sistemose programos vykdymo rezultatas irgi dažniausiai pateikiamas tekstiniu formatu. Daug pagalbos suteikiančiose sistemose (angl. *highly assistive*) besimokantieji gali naudotis grafinėmis programavimo aplinkomis, kur programos vykdymo rezultatas vizualizuojamas. 2.19 pav. pateikiamos visų trijų grupių aplinkų požymių aibės.

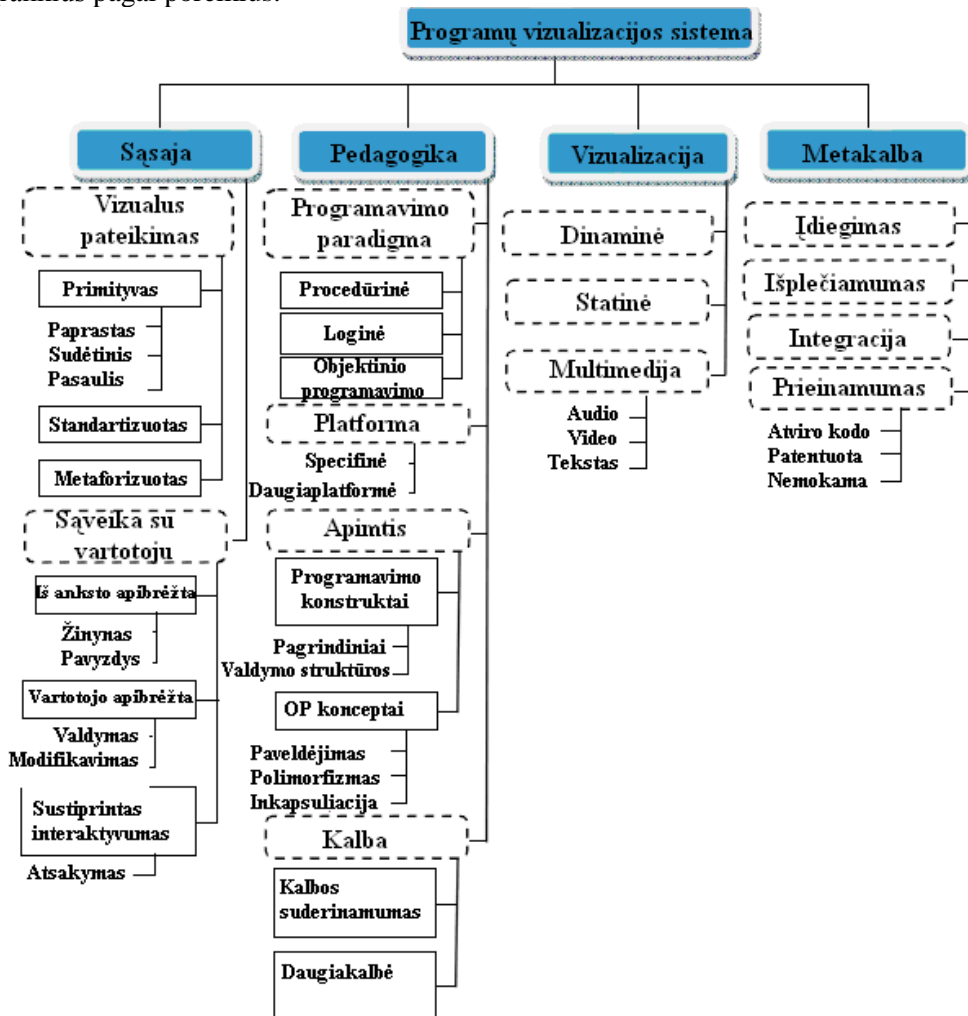


2.19 pav. Programavimo aplinkos: požymių aibės [DAB12]

Kaip atskirą mokymosi aplinkų grupę reikia išskirti mobiliųjų įrenginių (angl. *mobile devices*) panaudojimu pagrįstas aplinkas [MP10], [TMH+12], kurios leidžia

kurti programas naudojant mobiliuosius įrenginius. Tillmann *ir kt.* [TMH+12] siūlomos aplinkos branduolį sudaro struktūrizuota programavimo kalba, kuri sukurta taip, kad kodui rašyti pakaktų jutiklinio ekrano. Mobiliosiose aplinkose naudojamos programavimo kalbos panašios į universaliąsias programavimo kalbas ir sujungia būtinuosius, funkcinius ir objektinio programavimo aspektus. Programų taisymui sukurti pusiau struktūrizuoti kodo redaktoriai, kurie žymiai sumažina sintaksės klaidų skaičių. Mobilųjų aplinkų kūrėjai susiduria su platformų ir aplinkų palaikymo, mokomosios medžiagos parengimo, programinės įrangos kūrimo veiklų problemomis [TMH+12].

Mokantis programuoti labai plačiai naudojami programų vizualizacijos įrankiai. Stephen *ir kt.* [SFP+12] sudaryta programų vizualizacijos įrankių taksonomija (2.20 pav.) mokytojams ir mokiniams suteikia galimybes pasirinkti įrankius pagal poreikius.



2.20 pav. Programų vizualizacijos įrankių taksonomija [SFP+12]

## 2.10.6. Programavimo mokymuisi naudojami mokymosi objektai

Programavimo mokymosi procese mokymosi objektai (MO) naudojami supažindinant su naujomis sąvokomis ar idėjomis, suteikiant besimokantiesiems galimybes taikyti įgytas žinias praktiškai ir įveikti mokymosi sunkumus, sudarant sąlygas naudoti įvairius problemų sprendimo būdus, numatant studentų patirčių įvairovę ir papildomus išteklius, įvertinant mokinių žinias ir supratimą [Mat06], [MV07], [MAR08], [AHH11], [MHC12].

Mokslinėje literatūroje aprašomi įvairūs informatikoje naudojami MO tipai, jų modeliai ir pagrindinės savybės.

**Individualūs mokymosi objektai** (angl. *Customized Learning Objects*) [AG03], [GA03], apibūdinami kaip integruotas modulis, kurį sudaro tekstas, kodo pavyzdžiai, patikrinamieji klausimai ir kita papildoma medžiaga. Šio tipo MO yra sukurti naudojant Adaptyvios Knygos (angl. *Adaptive Book*) kūrimo priemonių rinkinį ir unikaliai aprašomi XML dokumentu, kuriame yra metaduomenys ir semantiniai sąryšiai tarp MO komponentų. Šio tipo MO leidžia mokytojams pritaikyti turinį dėstomam kursui, leidžia mokytojams ir mokiniams sukurti nepriklausomus mokymosi profilius ir juos bendrinti, sukuria bendradarbiavimu pagrįstą mokymosi aplinką, turi atvirą sąsają ir yra lengvai susiejami su kitais komponentais, atitinka IMS/SCORM standartus, pasižymi pakartotinio panaudojimo (angl. *reusability*) ir sąveikos (angl. *interoperability*) savybėmis.

**Codewitz mokymosi objektai** (angl. *Codewitz Learning Objects*) [MV07] yra interaktyvios programų vizualizacijos. MO kūrimui naudojamos Flash, Shockwave ir Applets technologijos. Codewitz MO palaikomi per naršyklę (angl. *Browse capable*), yra savarankiški (angl. *Stand-alone*), nereikalauja serverio, pakartotinai panaudojami, nesusieti su kitais MO ar ištekliais, skirti vienam specifiniam mokymosi tikslui, perdengia problemų sprendimo gebėjimų ugdymą loginiame ir algoritminiame lygmenyse.

**Mokymosi objektai kaip šablonai** (angl. *Learning Objects as Patterns*) [JB07] apibūdinami kaip pakartotinai panaudojant jau esamus MO naujai sukurti sudėtiniai MO. Šio tipo MO kūrimui taikoma šablonų technologija ir pagrįsti konstruktyvistine paradigma. MO struktūrą sudaro trys lygmenys: įvadas, supratimas (sudaro 3 sub-lygmenys: sudominimas, suvokimas, geras supratimas), taikymas, realizuojamas pakartotinis MO ir šablonų panaudojimas.

**Generatyviniai mokymosi objektai** (angl. *Generative Learning Objects*) [ŠD08], [DŠ08], [SBD09] yra specifikacija, apibūdinanti susijusių MO šeimyną. Generatyviniai MO gali būti laikomi MO objektų mini saugykla su numatyta galimybe iš saugyklos generuoti konkretų MO pagal besimokančiojo reikalavimus. Generatyvinių MO projektavimo principai apima: srities analizę, aukšto lygmens srities modelio kūrimą naudojant požymių diagramas (angl. *feature diagrams*), tinkamos generatyvinės technologijos, meta- ir tikslo kalbų parinkimą, aukšto lygmens modelio transformaciją į vykdomąją specifikaciją naudojant transformavimo taisykles, priklausančias nuo pasirinktos technologijos, sukurtos specifikacijos verifikavimą. Patikrintas generatyvinis MO perkeliamas į e.mokymosi aplinką. Sugeneruoti MO egzemplioriai integruojami su kitais MO ir sukuriama aukštesnio granuliacijos lygio MO.

**Prisitaikantys pavyzdžiai** (angl. *Adaptive Examples*) [Zha09] yra programų pavyzdžiai, kurių projektavimas pagrįstas mokymosi tikslais, kurie išskaidyti į dvi grupes: kalbos gramatika ir programų projektavimas. Kuriami trijų lygių pavyzdžiai: gramatikos pavyzdžiai demonstruoja kalbos gramatiką, aritmetiniai pavyzdžiai demonstruoja programų projektavimo teorijas, metodus ir įgūdžius, išsamūs (angl. *comprehensive*) pavyzdžiai integruoja daugelį pažinimo aspektų, kurie suteikia užuominų kaip išspręsti problemą ar padeda sukurti sistemos struktūrą. Kiekvienas pažinimo aspektas perdengiamas pavyzdžių aibe, kurios elementai yra tarpusavyje susiję. Išplėstinei grupei priklausantys pavyzdžiai turi prioritetus: išplėstinis pavyzdys papildo kitus pavyzdžius. Giminingoms grupėms priklausantys pavyzdžiai turi prioritetus, kurie išplaukia iš mokymosi metodų analogijos. Priešingų grupių pavyzdžiai turi prioritetus, kurie išplaukia iš mokymosi metodų priešybių. Numatoma tikslais pagrįsta praktinė veikla ir besimokančiojo pasirinkimo laisvė. MO derinami su interaktyviu kursu.

**Dinaminiai mokymosi objektai** (angl. *Dynamic Learning Objects*) [NS09] apibrėžiami kaip sudėtiniai MO, kurių dinamiškumas pasireiškia galimybe sukompiliuoti pilnutinį kurios nors temos MO pagal besimokančiojo poreikius. Taikomas konstruktyvistinis MO modelis. Sudėtinis MO pateikiamas kaip paprastų MO seka. Pasižymi pakartotinio panaudojimo (angl. *reusability*) ir sąveikos (angl. *interoperability*) savybėmis. Nauji dinaminiai MO sudaromi iš esančių saugykloje paprastų MO.

**Į studentą orientuoti mokymosi objektai** (angl. *Student-Centered Learning Objects*) [AHH12] projektuojami atsižvelgiant į besimokančiojo mokymosi stilių. Modelį sudaro abstrakcijos lygmens, turinio pateikimo, interaktyvumo lygmens ir sekų bei organizacinė dimensijos. MO tarpusavyje yra susiję. Dominuoja sąvokų aiškinimas naudojant animaciją.

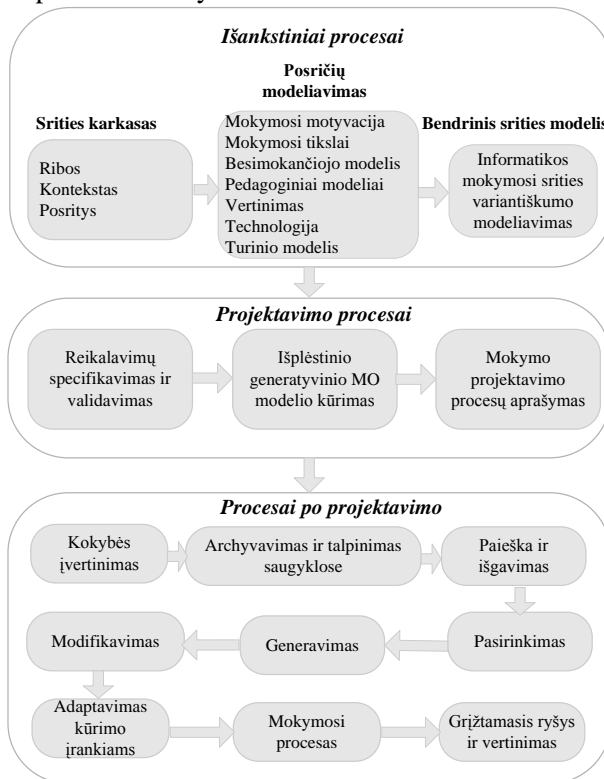
**Interaktyvūs modeliavimu pagrįsti mokymosi objektai** (angl. *Interactive Simulation-Based Learning Objects*) [TTT12] pateikia algoritmus, išreikštus schemomis (angl. *flow chart*). Besimokantysis, keisdamas kintamųjų reikšmes, seka kaip keičiasi animuota schema. MO objektuose aprašyta naudojamų išraiškų sintaksė ir semantika. Šio tipo MO kūrėjai ateityje planuoja lokalizuoti MO skirtingoms kalboms, ištirti, kokią įtaką MO naudojimas turi besimokančiųjų rezultatams, kaip atitinka jų mokymosi stilių ir nuostatas.

Alharbi ir kt. [AHH11] atliko informatikos MO, esančių saugykloje MERLOT (angl. *The Multimedia Educational Resource for Learning and Online Teaching*), kokybinę analizę ir nustatė, kad informatikos mokymuisi skirtų objektų skaičius didėja daug lėčiau, negu kitų disciplinų MO skaičius. Taip pat pažymima, kad saugykloje naudojama taksonomija neatspindi informatikos srities MO (pvz., tema „Duomenų struktūros“ neišskirta į atskirą skyrių). Populiariausi saugykloje yra animacijos tipo MO, tai rodo vizualizacijos svarbą mokantis programuoti. Tyrėjai siūlo kurti specializuotas informatikos srities MO saugyklas ir jose integruoti algoritmų vizualizacijos, automatinio pratimų vertinimo MO su interaktyviomis kompiliavimo ir derinimo aplinkomis. Naudodamiesi išvardintomis galimybėmis, MO kūrėjai galėtų kurti individualius MO pagal savo poreikius. Specializuotose saugyklose siūloma naudoti ACM Computing Classification System.

Matthews ir kt. [MHC12] išskiria mikro (angl. *micro*) ir makro (angl. *macro*) MO. Mikro MO yra mažas (3-5 min. trukmės), atskiras, nepriklausomas vidutiniško pakartotinio panaudojimo MO. Makro MO apima mokymosi turinio, praktikos ir vertinimo objektus, yra didesnis (25-30 min. trukmės), žemo pakartotinio panaudojimo. Atlikę empirinį tyrimą mokslininkai nustatė, kad didžiausią įtaką mokymosi efektyvumui turi ne MO dydis, o besimokančiųjų preferencijos. Mokymosi turinio objektai dažniausiai naudojami išankstinių žinių aktyvavimui, praktikos objektai naudojami sprendžiant uždavinius, vertinimo objektai naudojami mokantis programavimo kalbos sintaksę, nagrinėjant programų logiką, aiškinantis supratimo klaidas. Makro MO suteikia besimokančiajam didesnę pasirinkimo laisvę.

## 2.11. Apibendrintas tyrimo karkasas

Atliktų tyrimų apžvalga ir rezultatai leidžia daryti išvadą, kad kuriant MO modelius pirmiausia būtina atlikti srities analizę (angl. *domain analysis*). Atlikus srities analizę specifikuojami išplėstinio generatyvinio MO reikalavimai, sukuriama IGMO modeliai, aprašomi mokymosi projektavimo (angl. *instructional design*) procesai. Paskutiniame etape įvertinama sukurtų IGMO kokybė, jų archyvavimo ir talpinimo saugyklose, paieškos ir išgavimo, pasirinkimo, generavimo, modifikavimo ir adaptavimo galimybės, mokymosi procesai, grįžtamasis ryšys ir vertinimas. 2.21 pav. pateikiamas apibendrintas tyrimo karkasas.



2.21 pav. Apibendrintas tyrimo karkasas



## 2.12. Išvados

Atlikta literatūros analizė įgalino suformuluoti tokius pagrindinius tvirtinimus:

1. E.mokymosi metodologiniai pagrindai (pedagoginės teorijos, standartizavimo iniciatyvos, socialiniai mokymosi aspektai ir kt. ) yra bendri visai e.mokymosi sričiai, tačiau informatikos mokymasis turi savo specifiką (mokymosi modeliai, mokymosi aplinkos, mokymosi turinio pateikimas ir kt.), reikalaujančią atskiro požiūrio ir tyrimų.

2. Bendruose e.mokymosi tyrimuose mokymosi objektų tyrimai sudaro atskirą ir labai svarbią šaką, o generatyviniai mokymosi objektai išreiškia naują etapą su naujomis galimybėmis pakartotiniam panaudojimui ir mokymosi procesų efektyvumui.

3. Mokymosi objektų tyrimų technologiniam palaikymui adaptuojami gretutinių technologinių sričių koncepcijos ir metodai (pvz., mokymosi objektų modeliai, modeliavimo principai, kokybės vertinimo metrikos ir kt.).

4. Nors generatyvinio mokymosi objekto koncepcija suformuluota prieš dešimtmetį, šioje srityje yra dar daug neišspręstų problemų (sistemizavimas, aukšto lygmens GMO modelių sudarymas, kūrimo automatizavimas, GMO perkeliamumas į įvairias aplinkas, realus pritaikymas informatikos mokymuisi, integruojant sukurtus GMO į mokomaisiais robotais pagrįstas aplinkas, įvertinimo problemos ir kt.). Problemos suformuluotos ir tam tikru lygmeniu išspręstos šioje disertacijoje.

5. Nors e.mokymosi sritis apskritai ir informatikos bei programavimo mokymasis konkrečiai yra heterogeninės sritys su daugybe susijusių atributų (juos galima laikyti variantiškumu), tačiau dažniausiai jie (atributai) nagrinėjami kaip atskiros esybės. Disertaciniame darbe uždavinių ir metodikos pagrindą sudaro sisteminis informatikos mokymosi srities variantiškumo modeliavimas.

6. Analizės išdavoje suformuluotas apibendrintas tyrimo karkasas, pateikiantis bendrinę disertacinio darbo koncepciją, modelius, jų realizavimą ir įvertinimą.

### 3. INFORMATIKOS MOKYMOŠI SRITIES MODELIAVIMAS

#### 3.1. Įvadas

Trečiajame skyriuje informatikos (programavimo) mokymosi modeliavimas nagrinėjamas taikant sisteminį požiūrį, pagrįstą srities analizės metodais, plačiai taikomais programų inžinerijoje ir e.mokymėsi. Pagrindinis *modeliavimo tikslas* tyrimo kontekste – išgauti pagrindinius srities artefaktus ir žinias, reikalingus išplėstinio generatyvinio mokymosi objekto sukūrimui, analizei, įvertinimui ir naudojimui. Srities artefaktai dažnai traktuojami kaip srities modelis.

Nors literatūros apžvalgoje nurodyti atskirų informatikos mokymosi srities posričių modeliai, tačiau jie neturi formalių specifikacijų. Kai kurios posritys yra sudėtingos, todėl prieš kuriant bendrinius jų modelius, reikia išskirti esminius požymius bei nurodyti požymių tarpusavio sąryšius.

Šio skyriaus tikslas – sukurti informatikos mokymosi srities modelių, kurie bus panaudojami 4 skyriuje specifikuojant *išplėstinį generatyvinį mokymosi objektą* aukštame abstrakcijos lygmenyje, aibę.

Formuluojami tokie skyriaus uždaviniai:

- apibrėžti srities analizės ir modeliavimo metodus;
- atlikti mokymosi turinio ir procesų modeliavimo analizę;
- pagrįsti programų inžinerijos srities modeliavimo požymiais principų panaudojimą e.mokymėsi;
- sukurti informatikos mokymosi srities modeliavimo metodą;
- taikant sukurtą metodą sudaryti srities posričių modelius, juos analizuoti ir įvertinti.

3.2 skyrelyje apibrėžti srities analizės ir modeliavimo principai ir metodai; 3.3 skyrelyje iškeltos mokymosi turinio ir proceso modeliavimo problemos; 3.4 skyrelyje suformuotos informatikos mokymosi srities modeliavimo metodo sukūrimo teorinės prielaidos; 3.5 skyrelyje pateiktas informatikos mokymosi srities modeliavimo metodas; 3.6 skyrelyje apibrėžtos požymių modelių savybės; 3.7 skyrelyje pateikti informatikos mokymosi srities posričių modeliai; 3.8 skyrelyje nurodyti požymių modelių keitimo procesai, kurie leidžia pasiekti norimą rezultatą kuriant konkretų požymių modelį iš abstrakčių modelių; 3.9 skyrelyje pateikta požymių modelių analizė ir įvertinimas; 3.10 skyrelyje aptartas požymių modelių taikymas informatikos mokymosi srityje; 3.11 skyrelyje pateikiamas skyriaus apibendrinimas; 3.12 skyrelyje suformuluotos skyriaus išvados.

#### 3.2. Srities analizės bendrinė schema, principai ir metodai

Pakartotinio panaudojimo metodologijoje srities analizė vaidina labai svarbų vaidmenį, nes šiame procese informacija ir duomenys, reikalingi naujų sistemų kūrimui, nustatomi, išgaunami ir organizuojami taip, kad tiktų pakartotiniam panaudojimui kuriant naujas modernias sistemas ar jų komponentus.

Vienareikšmio srities apibrėžimo nėra. Šaltiniuose [Har02], [ŠD13] nurodoma, kad srities terminas gali būti vartojamas: 1) nagrinėjant kokią nors veiklą; 2) susiduriant su uždavinių rinkiniu; 3) turint taikymų rinkinį; 4) sritimi laikoma

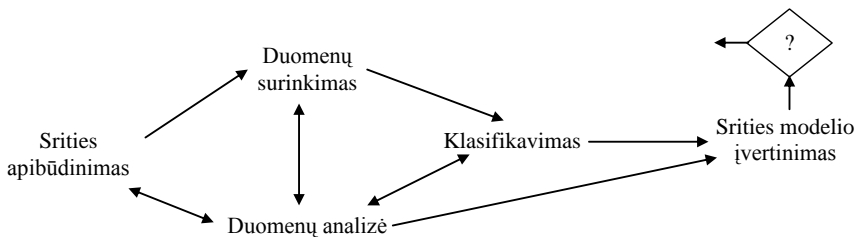
bendra terminologija susietos žinios; 5) sritis – tai veiklos ar žinių erdvė, turinti taikymų, kuriuos sieja bendros veiklos ir galimybės.

Bet kuri sritis gali būti padalinta į siauresnes sritis, vadinamas posritimis. Posritį gali sudaryti keli lygmenys. Jei srities funkcionalumas išreiškiamas vienos sistemos vienu posistemiui, sritis vadinamas sutraukta (angl. *encapsulated*). Kai srities funkcionalumas yra pasiskirstęs vienos ar kelių sistemų keliuose posistemiuose, tuomet sritis yra paskirstyta (angl. *distributed*).

Srities analizės rezultatas vadinamas srities modeliu. Harsu [Har02] pasiūlė srities modelį, kurį sudaro:

- srities apimtis (srities apibrėžimas, konteksto analizė) suranda ir apibrėžia srities ribas ir apimtį; bendrumų analizė išgauna srities taikymų bendrumus ir variantiškumus;
- srities žodynas numato ir apibrėžia srities terminus;
- notacijos (koncepcijų modeliavimas, koncepcijų pateikimas) numato būdą, kaip pristatomos srities modeliavimo koncepcijos (pvz., objektų būsenų perėjimo, esybių sąryšių, duomenų srautų diagramos ir pan.). Kai kurios notacijos vadinamos sistemos lygio notacijomis (angl. *system-level notations*) ir jomis sunku apibrėžti srities lygio (angl. *domain-level*) koncepcijas. Tinkamas notacijų pasirinkimas leidžia greičiau ir geriau suprasti sistemą ar sritį;
- reikalavimų inžineriją sudaro reikalavimų, apibrėžiančių srities taikymus, surinkimas, apibrėžimas, dokumentavimas, verifikavimas ir valdymas.

Srities analizės metodų bendri procesai pateikiami 3.1. pav. [Bat05].



3.1 pav. Srities analizės bendrinė schema [Bat05]

Praktikoje plačiausiai naudojami srities analizės metodai – FODA (*Feature-Oriented Domain Analysis*), FORM (*Feature-Oriented Reuse Method*), FeatureRSEB (*Feature Reuse-Driven Software Engineering Business*), DSSA (*Domain-Specific Software Architectures*), Sandwich, DARE (*Domain Analysis and Reuse Environment*), FAST (*Family Oriented Abstraction, Specification and Translation*), ABDA (*Analogy-Based Domain Analysis*) – išsamiai išnagrinėti [ŠD13], [Har02], [AR07], [LUM07] darbuose.

Pastaraisiais metais dominuoja dvi programinės įrangos kūrimo metodologijos: modeliais grįsta inžinerija (MDE, angl. *Model-Driven Engineering*) [Sch06] ir programų šeimynų inžinerija (PLE, angl. *Product Line Engineering*) [Bos00]. Antroji metodologija akcentuoja pakartotinį programų šeimynų bendrų kodo dalių panaudojimą ir operuoja požymiais (angl. *features*), kurie yra išorinės programų

charakteristikos ir gali būti įvairiais būdais derinami, taip keičiant programų funkcionalumą. Programų šeimynos architektūra grindžiama programų bendrumais ir variantiškumais. Iš šios architektūros skirtingi programų variantai išgaunami pakartotinai panaudojant komponentus ir struktūras pritaikant komponentinę ir generatyvinę pakartotinio panaudojimo technologijas [But01].

Modeliais grįsta inžinerija akcentuoja nuo technologijų nepriklausomų srities modelių naudojimą visuose kūrimo etapuose. Tokie modeliai gali būti įvairių abstrakcijos lygmenų, todėl naudojamas daugiapakopis (hierarchinis) modeliavimas (angl. *multilevel (hierarchical) modeling*). Modeliai kuriami naudojant srities metamodelyje apibrėžtus konceptus, sąryšius ir semantiką. Srities modeliai transformuojami į konkrečių platformų modelius naudojant transformacijos taisykles. Transformacijos taisyklės transformuoja pirminio (angl. *source*) modelio elementus į tikslo (angl. *target*) metamodelį [MM03].

Abi aukščiau minėtos metodologijos (modeliais grįsta inžinerija ir programų šeimynų inžinerija) akcentuoja modelių, modeliais grįstų procesų ir variantiškumo modeliavimo naudojimą. Jos skiriasi tuo, kad naudoja skirtingus konceptus ir modelių tipus. Modeliais grįstos inžinerijos esminis konceptas yra *objektas* ir kuriami *objektiškai orientuoti* modeliai. Programų šeimynos inžinerijos pagrindinis konceptas yra *požymis* ir kuriami *požymiais grįsti* modeliai.

Abiejų metodologijų pagrindą sudaro variantiškumo modeliavimas. Plačiausiai naudojamos variantiškumo modeliavimo metodologijos yra: 1) požymių modeliavimas (angl. *Feature Modeling, FM*), 2) Apimties-Bendrumo-Variantiškumo analizė (angl. *Scope-Commonality-Variability analysis, SCV*) [CHW98], [CBK13], kuri remiasi aibių teorija; 3) COVAMOF [SDN+04] karkasas, palaikantis keturis pagrindinius reikalavimus: i) suvienodintas variantinių taškų kaip pirmenybinių esybių atvaizdavimas visuose abstrakcijos lygmenyse; ii) hierarchinis variantiškumo atvaizdavimas; iii) priklausomybių kaip pirmenybinių elementų atvaizdavimas; iv) sąryšių tarp priklausomybių modeliavimas; 3) formali konceptų analizė (angl. *Formal Concept Analysis*) [Sne96], grindžiama konceptų, išgautų iš programos kodo, analize; 4) srities kalba FAMILIAR (angl. *FeAture Model scrIpt Language for manIpulation and Automatic Reasoning*), skirta didelės apimties modelių valdymui [CL13], kuria aprašytų modelių verifikavimas ir analizė atliekama naudojant įrankį SPLOT (angl. *Software Product Lines Online Tools*).

Toliau bus nagrinėjami požymių modeliai, kurie apibrėžiami kaip notacijų šeimyna ir programų šeimynų bendrumų ir variantiškumų modeliavimo metodas [KCH+90]. Ankstyvuosiuose programų šeimynų kūrimo etapuose požymiais grįsti modeliai numato požymius, turinčius daugiausia įtakos sistemų naudojimui, plėtros sąnaudoms, technologinei rizikai [BS99]. Požymių modeliai vaidina svarbų vaidmenį kuriant sistemų šeimynos architektūrą, kuri realizuoja požymių modeliuose apibrėžtus variantinius taškus [Bos00]. Taikymų inžinerijoje požymių modeliai gali būti naudojami aprašant ir analizuojant reikalavimus.

Požymių modeliai buvo sėkmingai pritaikyti e.mokymosi srityje konceptualiai specifikuojant generatyvinius mokymosi objektus [ŠD08], [SDB+08], sprendžiant mokymosi objektų sekų sudarymo problemą [DS09], modeliuojant e.mokymuisi skirtą vadovėlį [CNC12], kuriant e.mokymosi sistemas [DDA12].

### 3.3. Mokymosi turinio ir procesų modeliavimo problema

Šiuolaikiniuose e.mokymosi tyrimuose labai aktuali mokymosi turinio ir procesų modeliavimo problema.

Rodríguez-Artacho ir Maillo [RM04] apibrėžė tris pagrindines tyrimų kryptis, kurioms turėtų būti skiriama daugiausia dėmesio: 1) formalių specifikacijų (angl. *formal specifications*) panaudojimas, 2) mokymosi turinio atvaizdavimo modelių (angl. *representative frameworks to model educational material*) sukūrimas ir 3) pagrįstas abstrakcijos lygmens pasirinkimas siekiant užtikrinti pakartotinį MO panaudojimą.

Paquette *ir kt.* [PLL+06] mokymosi projektavimą (angl. *learning design*) apibrėžė kaip žinių inžinerijos proceso (angl. *knowledge engineering process*) rezultata, kur žinios ir kompetencijos, mokymosi projektavimas ir pateikimo modeliai (angl. *delivery models*) integruojami į bendrą karkasą.

Iki 2006 m. mokymosi turinio ir procesų modeliavimui buvo naudojamos mokymosi modeliavimo kalbos (MMK, angl. *Educational Modeling Languages*). Laforcade ir Choquet [LC06] atliko MMK analizę ir nustatė, kad MMK: 1) yra įtraukiamos paskutiniame mokymosi proceso projektavimo etape specifikuojant mokymosi scenarijų; 2) organizuojant ir struktūrizuojant mokymosi objektus MMK panaudojimas yra susijęs su standartizavimo problemomis; 3) turi XML pagrįstus atvaizdavimus, kad būtų lengvai įgyvendinamos ir realizuojamos; 4) linkusios per anksti užšaldyti vykdymo sąlygas ir tvarką; 5) neaprašo konkrečių mokymosi situacijų.

Botturi *ir kt.* [BDB+06] MMK įvertinimui pasiūlė taikyti 5 kriterijus:

- Stratifikacija (angl. *stratification*). Daugiasluoksnė (angl. *layered*) kalba siūlo įrankių ar įvairių atvaizdavimų rinkinių skirtingų tipų esybių (besimokančiųjų ir jų vaidmenų, veiklų, mokomosios medžiagos) aprašymui. Vienasluoksnė (angl. *flat*) kalba surenka visas esybes į vieną atvaizdavimą.

- Formalizavimas (angl. *formalization*). Formali (angl. *formal*) kalba apibrėžia griežtus ir uždarus konceptų ir taisyklių rinkinius, naudojamus kuriant modelius. Neformali (angl. *informal*) kalba yra atviresnė. Kalbos, turinčios ir formalių, ir neformalių kalbų bruožų, laikomos pusiau formaliomis (angl. *semi-formal*).

- Detalizavimas (angl. *elaboration*). Kiekviena kalba gali numatyti daugiau ar mažiau specifinio artefakto savybių. Konceptualiam (angl. *conceptual*) lygmenyje nagrinėjami esminiai požymiai; specifikavimo (angl. *specification*) lygmuo numato išsamesnio aprašymo galimybę; realizacijos (angl. *implementation*) lygmenyje detalizavimo lygmuo yra aukščiausias.

- Perspektyva (angl. *perspective*). Daugiaperspektyvinės (angl. *multiperspective*) kalbos naudoja skirtingus įrankius daugiau negu vieno tos pačios esybės požiūrio atvaizdavimui. Perspektyvos gali turėti tą patį detalumo lygmenį ir būti tame pačiame sluoksnyje.

- Notacija (angl. *notation*). Dažniausiai MMK naudojama tekstinė ir grafinė notacija.

Caeiro-Rodríguez *ir kt.* [CLA06] išskyrė 2 pagrindines modeliavimo naudojant MMK problemas: 1) elementų, susijusių su mokymosi vienetais (angl. *UoL, Units of*

Learning) modeliavimas (individai, tikslai, artefaktai); 2) elementų tarpusavio sąveikos modeliavimas (veiklų seka, užduotys, tikslai).

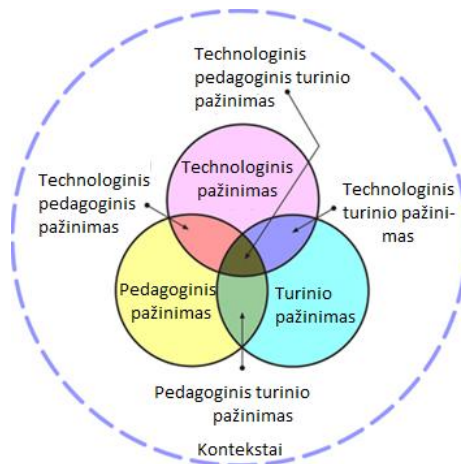
Norint sėkmingai modeliuoti e.sritį, būtina ją aiškiai išreikšti. Koehler ir Mishra [KM09] pasiūlė trijų komponentų karkasą TPACK (3.2. pav.), kuri sudaro mokytojų žinios apie turinį, pedagogiką ir technologiją. Šiame darbe TPACK karkasas naudojamas informatikos mokymosi sričiai pristatyti.

Apibendrinant du pirmuosius poskyrius formuluojamos išvados:

1. E.mokymosi srityje dirbantys tyrėjai pripažįsta srities aukštesnio lygmens modeliavimo svarbą naudojant mokymosi modeliavimo kalbas ir aukštesnio lygmens modelius. Siekiant padidinti modeliavimo galimybes, yra mėginimų taikyti e.mokymosi modeliais grįstus programų inžinerijos metodus.

2. Požymiais grįsto modeliavimo konceptai ir kalbos e.mokymosi srityje gali būti sėkmingai taikomi dėl: (1) jų paprastumo, galimybių grafinę notaciją transformuoti į tekstinę; (2) lengvai ir intuityviai suprantamų pagrindinių struktūrų; (3) galimybių kurti posričių skirtingų abstrakcijos lygmenų modelius; (4) požymio sąvokos universalumo modeliuojant skirtingus bet kurios posrities aspektus.

3. Požymių modeliai gali būti laikomi mokymosi modeliavimo kalbų šeimos sudėtine dalimi.



3.2 pav. TPACK karkasas informatikos mokymosi sričiai [KM09]

### 3.4. Siūlomo modeliavimo metodo teorinės prielaidos

Kaip minėta įvade, šio skyriaus tikslas yra sukurti metodą, tinkamą informatikos mokymosi srities analizei ir modeliavimui. Modeliavimu tyrimo kontekste suprantamas aukštesnio abstrakcijos lygmens informatikos mokymosi srities modelių, kurie bus naudojami kaip įvesties duomenys (angl. *input data*) kuriant išplėstinį generatyvinį mokymosi objektą per transformacijas, aibės išgavimas.

#### 3.4.1. Modeliavimo metodo sudarymo principai

Sudarant informatikos mokymosi srities modeliavimo metodą pritaikyti programų inžinerijoje plačiai naudojami fundamentalūs *konceptijų atskirties* (angl. *separation of concepts, separation of concerns*) ir *konceptijų integracijos* (angl.

*integration of concepts*) principai. Minėti principai paprastai taikomi abu: pirmiausia atskirtis, po to integracija. Apibendrinant šie principai gali būti apibrėžti kaip sistemų projektavimo analizės ir sintezės etapai.

Koncepcijų atskirtį Greer apibrėžia ir kaip *principą*, ir kaip *procesą* [Gre08]. Tyrimo kontekste koncepcijų atskirties principas formuluojamas kaip prielaida, kad modeliai privalo turėti būtinus atributus ir jiems būdingą elgseną, tačiau atsiribojama nuo neesminių atributų ir jų elgsenos. Koncepcijų atskirties principas suvaidino svarbų vaidmenį kuriant naujus srities analizės metodus. E.mokymesi taip pat pažymima šio principo svarba [CBS+11], tačiau daugeliu atvejų šis principas yra naudojamas netiesiogiai arba išreiškiamas kitais terminais, pvz., klasifikacija.

Kaip pavyzdžius galima pateikti: turinio ir pedagoginių požymių atskirtį; turinio ir technologinių požymių atskirtį; turinio, kaip srities pagrindo, ir srities konteksto požymių atskirtį ir pan.

Sudarant modeliavimo metodą taikytas *analogijos* principas (angl. *analogy principle*). Tyrimo kontekste mokymosi kurso projektavimą sieja analogija su programų šeimos sistemos projektavimu. Kurso struktūra turi panašumų su programinės įrangos architektūra. Aukštesniame lygmenyje pasirinkta požymių aibė modeliuoja programinės įrangos komponentus architektūros viduje. Panašiai mokymosi objektų rinkinys modeliuoja kurso temas.

### **3.4.2. Reikalavimai modeliavimo metodui**

**3.4.2.1 reikalavimas.** Informatikos mokymosi sritis yra heterogeninė, todėl turi būti aiškiai apibrėžtos srities ribos ir apimtis.

**3.4.2.2 reikalavimas.** Srities ribos ir apimtis priklauso nuo analizės tikslų.

**3.4.2.3 reikalavimas.** Iš 3.4.2.1 ir 3.4.2.2 reikalavimų seka, kad sritis turi būti vaizduojama modelių, atitinkančių tyrimo tikslus, aibe.

**3.4.2.4 reikalavimas.** Prieš kuriant modelį turi būti apibrėžti modelio naudojimo tikslai.

**3.4.2.5 reikalavimas.** Su modeliais atliekamos įvairios manipuliacijos: suliejimas, skaidymas, agregacija ir t.t.

**3.4.2.6 reikalavimas.** Naujai sukurti ir atlikus manipuliacijas gauti modeliai turi būti korektiški, todėl labai svarbu tinkamai atlikti modelių verifikaciją.

**3.4.2.7 reikalavimas.** Požymių diagramų konstravimas ir manipuliacijos su modeliais atliekamos naudojant atitinkamus įrankius.

**3.4.2.8 reikalavimas.** Kad būtų lengviau valdyti modelio sudėtingumą, naudinga taikyti modelio hierarchiją įvedant skirtingo detalumo lygmenis.

**3.4.2.9 reikalavimas.** Įvertinant konteksto svarbą e.mokymosi srityje tikslinga kurti bazinį modelį, apimantį mokymosi turinį ir jo konteksto modelį.

**3.4.2.10 reikalavimas.** Konteksto modelis gali būti pateikiamas neišreikštine (angl. *implicit*) arba išreikštine (angl. *explicit*) forma. Darbe pasirinkta išreikštine konteksto vaizdavimo forma, kuri yra tinkamesnė modelių transformacijų požiūriu.

### **3.4.3. Informatikos mokymosi srities analizės metodai**

3.3. skyrelyje buvo nurodyta, kad informatikos mokymosi sričiai apibrėžti naudojamas TPACK karkasas [KM09] (3.2 pav.). Karkasas detalizuotas pridėjus papildomus artefaktus iš kitų literatūros šaltinių: motyvacijos faktorius,

specifikuotus mokymosi tikslus, besimokančiųjų profilius, mokymosi metodus ir kt., kurie išsamiai išnagrinėti 2 skyriuje.

Darbe naudojamas FODA metodas, kuris remiasi požymių išskyrimu, jų klasifikavimu, funkcijų suskaldymu, savybių ir architektūros modeliavimu bei srities variantiškumo atvaizdavimu. Taikomi trys pagrindiniai FODA principai: 1) srities ribų ir konteksto identifikavimas; 2) požymiais grįstas konteksto modeliavimas; 3) požymiais grįstas posričių modeliavimas srities viduje.

Srities variantiškumo identifikavimui naudojamas Apimties-Bendrumo-Variantiškumo analizės metodas, pagrįstas aibių teorija: *bendrumas* suprantamas kaip prielaida, kad egzistuoja objektų aibės S atributai, kurių reikšmės yra tokios pačios visuose S elementuose. *Variantiškumas* apibrėžiamas kaip prielaida, kad egzistuoja aibės S atributai, kurių reikšmės yra vienodos tik kai kuriuose aibės S elementuose, arba mažiausiai du aibės elementai turi skirtingas atributo reikšmes.

#### 3.4.4. Požymių diagramų notacija

Požymių diagramos (angl. *Feature Diagrams*) yra požymių modelius aprašanti grafinė kalba, naudojama variantiškumo vaizdavimui ir modeliavimui aukštame abstrakcijos lygmenyje, dažniausiai pradinuose projektavimo etapuose. Kadangi yra įvairių požymių diagramų notacijų ir elementai interpretuojami skirtingai [ŠD13] [SHT+07], požymių diagramas galima laikyti grafinių modeliavimo kalbų, kurios pasižymi sąryšių tarp požymių tinkamu identifikavimu, aibe. Požymių diagramos taip pat turi tekstinę notaciją [DK02], [CL13] ir aprašomos gramatikomis bei teiginių logikos formulėmis [Bat05].

Toliau apibrėžiamos pagrindinės sąvokos, būtinos srities modeliavimui požymių diagramomis.

**3.4.4.1 apibrėžimas.** Požymis (angl. *feature*) yra vartotojui matoma srities charakteristika [KCH+90], kokybinė koncepto savybė arba funkcinis reikalavimas [CE00].

**3.4.4.2<sup>1</sup> apibrėžimas.** Požymių modelis (angl. *feature model*) – srities modelis, kai srities artefaktai modeliuojami požymiais [ŠD13].

**3.4.4.2<sup>2</sup> apibrėžimas.** Požymių modelis apibrėžia požymius, jų tarpusavio sąryšius bei apribojimus [Rie03].

**3.4.4.3 apibrėžimas.** Požymių diagrama (angl. *feature diagram*) – požymių modelius aprašanti speciali grafinė notacija [ŠD13].

Požymių diagrama tai kryptinis beciklis grafas, kurį sudaro mazgų, briaunų ir briaunų žymų aibės. Šakninis mazgas atvaizduoja aukščiausio lygmens požymį. Tarpiniai požymiai atvaizduoja sudėtinius požymius, o lapai – atominius požymius, kurie duotame kontekste neskaidomi į smulkesnius. Grafo briaunos leidžia atlikti sudėtinių požymių dekompoziciją ir apibrėžia sąryšius tarp požymių. Briaunų žymos nurodo sąryšių tipą. Požymius gali sieti sąryšis „tėvas-vaikas“ arba apribojimas.

**3.4.4.4 apibrėžimas.** Privalomu (angl. *mandatory, AND*) požymiu vadinamas požymis, kuris būtinai turi būti pasirinktas, jeigu jo tėvinis požymis yra pasirinktas. Privalomi požymiai išreiškia srities bendrumą (angl. *commonality*).

**3.4.4.5 apibrėžimas.** Neprivalomu (angl. *optional, OR*) požymiu vadinamas požymis, kuris gali būti pasirinktas, jeigu jo tėvinis požymis yra pasirinktas. Neprivalomi požymiai išreiškia srities variantiškumą (angl. *variability*).



**3.4.4.6 apibrėžimas.** Alternatyviuoju (angl. *alternative*, *XOR*) požymiu vadinamas požymis, kuris tik vienas gali būti pasirinktas, jeigu jo tėvinis požymis yra pasirinktas. Alternatyvieji požymiai išreiškia srities variantiškumą (angl. *variability*).

**3.4.4.7 apibrėžimas.** Apribojimas apibrėžia požymių apribojimus arba funkcinius sąryšius tarp požymių, kurių nesiejia „tėvas-vaikas“ ryšys.

**3.4.4.8 apibrėžimas.** Išskyrimo (angl. *excludes*) apribojimas nurodo, kad sistema gali turėti tik vieną iš šiuo sąryšių siejamų požymių.




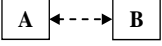
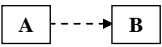
**3.4.4.9 apibrėžimas.** Jei požymius sieja reikalavimo (angl. *requires*) apribojimas, tai sistema, turinti vieną iš iš šių požymių, būtinai turi ir kitą požymį.

**3.4.4.10 apibrėžimas.** Variantu vadinamas požymis, kuris duotame kontekste neskaidomas į smulkesnius požymius.

**3.4.4.11 apibrėžimas.** Variantiniu tašku vadinamas neprivalomų (žr. 3.4.4.5 apibrėžimą) arba alternatyviųjų (žr. 3.4.4.6 apibrėžimą) požymių grupių tėvinis požymis.


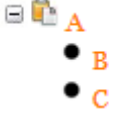
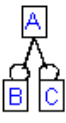
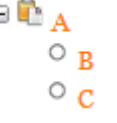

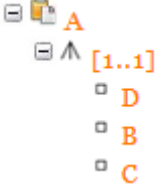

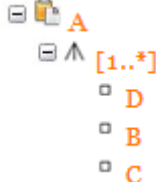
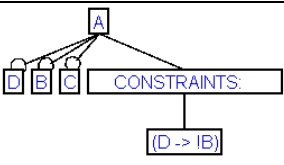
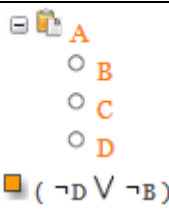
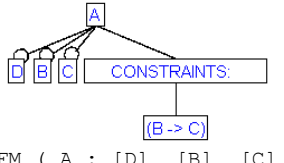
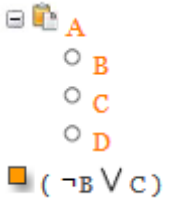
Darbe naudojama Czarnecki *ir kt.* [CHE04], Batory [Bat05] pasiūlyta grafinė notacija ir požymių išraiška teiginių logikos (angl. *propositional logic*) formulėmis [TBK09] (3.1 lentelė).

**3.1 lentelė** Požymių diagramų grafinė ir formali notacija [CHE04], [Bat05], [TBK09]

Požymiai ir sąryšiai	Aprašymas	Grafinė notacija	Išraiška teiginių logikos formulėmis *
Privalomi (AND)	Visi požymiai turi būti pasirinkti		$(P \Rightarrow \bigwedge_{i \in M} C_i) \wedge (\bigvee_{1 \leq i \leq n} C_i \Rightarrow P)$ ; $M \subseteq \{1, \dots, n\}$ <i>P</i> žymi tėvo požymį, <i>C</i> <sub>1</sub> , ..., <i>C</i> <sub><i>n</i></sub> yra jo vaikai. Jei ryšys tarp vaikų yra AND, tuomet $M \subseteq \{1, \dots, n\}$ identifikuoja privalomus požymius pagal jų indeksą.
Alternatyvieji (XOR)	Tik vienas požymis gali būti pasirinktas		$(P \Leftrightarrow \bigvee_{1 \leq i \leq n} C_i) \wedge_{i < j} (\neg C_i \vee \neg C_j)$
Neprivalomi (OR)	Vienas ar daugiau požymių gali būti pasirinkta		$P \Leftrightarrow \bigvee_{1 \leq i \leq n} C_i$
Apribojimas <išskyrus>	Požymiai <b>A</b> ir <b>B</b> vienu metu negali būti pasirinkti		$\neg A \vee \neg B$
Apribojimas <reikalauja>	Požymis <b>A</b> reikalauja, kad būtinai būtų pasirinktas požymis <b>B</b>		$\neg A \vee B$

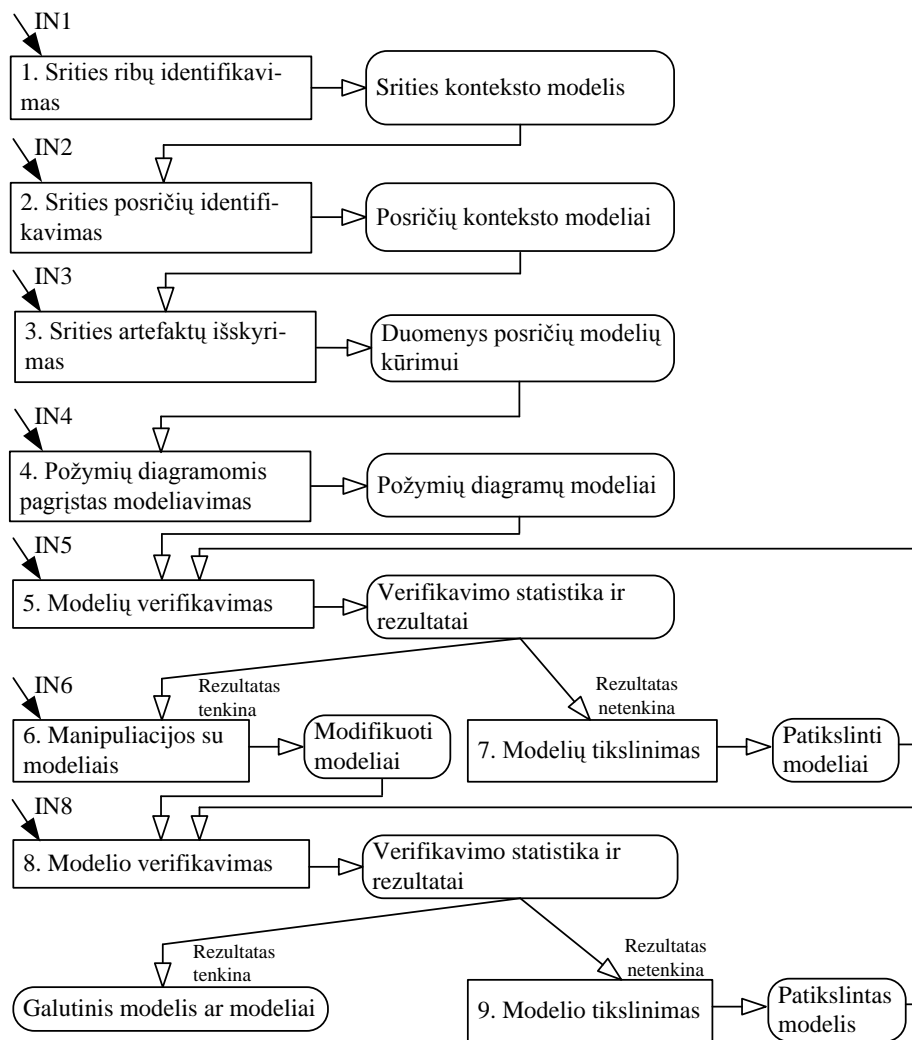
3.2 lentelėje pateikiamos FAMILIAR kalbos [CL13] ir automatizuoto požymių modelių verifikavimo ir analizės įrankio SPLOT [<http://www.splot-research.org>] požymių modelių notacijos.

### 3.2 lentelė FAMILIAR ir SPLOT požymių modelių notacija

Požymiai ir sąryšiai	Notacija		Požymių medis (xml failo fragmentas)
	FAMILIAR	SPLOT	
Privalomi (AND)	 <p>FM ( A : B C ; )</p>		<pre>&lt;feature_tree&gt; :r A(_r0)   :m B(_r1)   :m C(_r2) &lt;/feature_tree&gt;</pre>
Neprivalomi (angl. <i>optional</i> )	 <p>FM ( A : [B] [C] ; )</p>		<pre>&lt;feature_tree&gt; :r A(_r0)   :o B(_r1)   :o C(_r2) &lt;/feature_tree&gt;</pre>
XOR grupė [1..1]	 <p>FM ( A : (D B C) ; )</p>		<pre>&lt;feature_tree&gt; :r A(_r0)   :g [1,1]     : D(_r1)     : B(_r2)     : C(_r3) &lt;/feature_tree&gt;</pre>
OR grupė [1..*]	 <p>FM ( A : (D B C)+ ; )</p>		<pre>&lt;feature_tree&gt; :r A(_r0)   :g [1,*]     : D(_r1)     : B(_r2)     : C(_r3) &lt;/feature_tree&gt;</pre>
Apribojimas <išskyrus>	 <p>FM ( A : [D] [B] [C] ; (D -&gt; !B) ; )</p>		<pre>&lt;feature_tree&gt; :r A(_r0)   :o B(_r1)   :o C(_r2)   :o D(_r3) &lt;/feature_tree&gt; &lt;constraints&gt; C0:~_r3 or ~_r1 &lt;/constraints&gt;</pre>
Apribojimas <reikalauja>	 <p>FM ( A : [D] [B] [C] ; (B -&gt; C) ; )</p>		<pre>&lt;feature_tree&gt; :r A(_r0)   :o B(_r1)   :o C(_r2)   :o D(_r3) &lt;/feature_tree&gt; &lt;constraints&gt; C0:~_r1 or _r2 &lt;/constraints&gt;</pre>

### 3.5. Informatikos mokymosi srities modeliavimo metodas

3.3 pav. pateikiamas bendras informatikos mokymosi srities modeliavimo metodo karkasas. Modeliavimo metodas pagrįstas aukšto lygmens procesų logine seka kartu su jų rezultatais. Tyrimė sritimi laikoma informatikos mokymosi sritis, kuri yra heterogeninė. Kiekvienas procesas aprašomas kaip į tikslą orientuotas įėjimų-išėjimų sąryšis pagal tokią schemą: *tikslas – pradiniai duomenys – proceso eiga – rezultatas*.



Legenda: □ - Procesas; ○ - Proceso išėjimas; ▸ - Įėjimas/Išėjimas; ➔ - (IN) Išorinis įėjimas.

3.3 pav. Informatikos mokymosi srities modeliavimo metodo karkasas

**1 procesas.** Tikslas: apibrėžti likusių procesų pradinės sąlygas. Kaip reikalauja FODA ir SCV metodai, ribų apibrėžimas yra svarbi išankstinė modeliavimo sąlyga, nes tai apibrėžia veiklų apimtį. IN1 sudaro FODA ir SCV taisyklės ir TPACK karkasas. IN1 atributas užpildomas ekspertui analizuojant TPACK karkasą ir priklauso nuo eksperto žinių apie sritį: naudojami įvairūs dokumentai, pvz., standartų specifikacijos, literatūros šaltiniai ir pan. Proceso rezultatas yra konteksto modelis. Konteksto modelis gali būti išreikštas susijusių posričių tokios pačios svarbos atributais ir jų sąryšiais su baziniu (turinio) modeliu. Remiantis TPACK karkasu, konteksto modelis gali būti specifikuotas kaip trijų komponentų (posričių) – pedagoginio, technologinio ir turinio pažinimo – sąveika.

**2 procesas.** Tikslas: identifikuoti, kas yra srities ribose ir susiaurinti srities modelį, kad būtų supaprastintas modeliavimas. Šiame etape pritaikomas koncepcijų atskirties principas. IN2 apima FODA ir SCV taisykles, TPACK karkasą, koncepcijų atskirties ir analogijos principus. Procesas įvykdomas peržiūrint TPACK modelį ir pagrindžiant kiekvienos posrities (pedagoginės, technologinės, turinio) modeliavimo tikslus. Proceso rezultatas yra siauresni konteksto modeliai (pvz. pedagoginės posrities: mokymosi tikslų, motyvacijos, besimokančiojo, programavimo mokymosi metodų, vertinimo).

**3 procesas.** Tikslas: išgauti modeliavimo tikslą atitinkančius artefaktus. IN3 apima metodus, įrankius, artefaktų paieškos veikėjus (srities ekspertus), pažinimą, sprendimus, reikalavimus ir pan. Procesas vykdomas, kai analitikas, remdamasis savo ir srities ekspertų žiniomis ir patirtimi, renka, klasifikuoja, verifikuoja duomenis. Proceso rezultatas – duomenų rinkiniai, kurie bus naudojami sudarant pirminius posričių modelius.

**4 procesas.** Tikslas: abstrakčiai ir tiksliai pateikti modelius. IN4 apima požymiais grįstą kalbą ir įrankius, tokius kaip FAMILIAR, SPLOT, taip pat analitiko pažinimą ir kompetenciją. Procesas vyksta identifikuojant sąryšius („tėvas-vaikas“) ir apribojimus tarp požymių, kuriant požymių modelius ir juos testuojant. Proceso rezultatas – požymių modelių aibė.

**5 procesas.** Tikslas: patikrinti požymių modelių teisingumą ir surinkti statistiką. IN5 apima požymių modelių verifikavimo įrankius (SPLOT), modeliautojo žinias ir kompetenciją. Procesas vyksta naudojant požymių modelių analizės įrankius. Proceso rezultatas – modelio charakteristikų ir savybių statistika.

**6 procesas.** Tikslas: nustatyti, kokiems tikslams bus naudojami sudėtiniai modeliai. IN6 apima reikalavimus manipuliacijoms su modeliais, įrankį FAMILIAR. Procesas vykdomas naudojant įrankį FAMILIAR. Proceso rezultatas – sudėtiniai modeliai.

8 procesas yra analogiškas 5 procesui, 9 procesas – 7 procesui, kuriame reikalavimų neatitinkantys modeliai yra tikslinami ir pakartotinai verifikuojami.

### 3.6. Požymių modelių savybės

Šiame skyrelyje formuluojamos svarbiausios modelių savybės, susijusios su modelių svarba mokymosi sričiai.

**3.6.1 savybė.** Kadangi informatikos mokymosi sritis yra heterogeninė (kaip ir bet kurio kito mokomojo dalyko sritis), norint atvaizduoti sritį aukštesniame abstrakcijos lygmenyje, naudojami daugialypiai požymių modeliai (angl. *multiple feature models*) dėl (1) vis didėjančių reikalavimų, (2) srities sudėtingumo augimo, (3) pakartotinio panaudojimo sustiprinimo, (4) automatizavimo tikslų.

**3.6.2 savybė.** Modelių aibė, pateikta 3.5 skyrelyje, turi tokią pačią semantiką, kaip ir 2 skyriuje nagrinėtuose literatūros šaltiniuose, kuriuos išanalizavus išgauti požymiai, panaudoti kuriant modelius. Modelių privalumai: tikslumas, korektiškumas, galimybė pateikti sudėtingus modelius įvairiuose abstrakcijos lygmenyse, pakartotinis panaudojimas.

**3.6.3 savybė.** Požymių modeliai yra lengvai perkonfigūruojami. Modelių suliejimas, išskaidymas, keitimas suteikia galimybes sukonfigūruoti modelius pagal poreikius.

**3.6.4 savybė.** Modeliavimo lankstumui didelę įtaką daro du pagrindiniai modelių tipai – bazinis ir konteksto – bei jų tarpusavio sąryšiai.

**3.6.5 savybė.** Konteksto modelis, lyginant su baziniu modeliu, turi aukštesnį prioritetą. Abu modeliai gali būti sujungiami į vieną, arba naudojami kaip modelių aibė. Kokį būdą naudoti priklauso nuo tikslo konkrečioje situacijoje.

**3.6.6 savybė.** Naudojant daugialypius požymių modelius jų prioritetų sąryšiai gali būti modeliuojami įvedant prioritetų lygmenis: aukštą, tarpinį, žemą, be prioriteto. Charnecki ir kt. [CHE05] naudoja terminą *pakopinės konfigūracijos modelis* (angl. *staged configuration model*), kurio reikšmė atitinka *modelio su prioritetais* reikšmę.

**3.6.7 savybė.** Modeliai gali keisti savo vaidmenį: konteksto modelis tapti baziniu ir atvirkščiai. Tai gali atsitikti keičiantis modeliavimo tikslams arba keičiantis pateikimo lygmeniui.

**3.6.8 savybė.** Tolesniam modeliavimui ir geresniam srities supratimui galima sukurti unifikotą bendrinį informatikos mokymosi srities apibrėžtos apimties modelį.

Jei požymių modelis labai didelis, kyla sunkumų jį skaitant ir atvaizduojant. Sudėtingą modelį galima išskaidyti dalimis, išskiriant tam tikrą mazgą. Išskirtas mazgas turi būti pažymėtas nuoroda (angl. *reference*) į likusią požymių diagramos dalį. Jei yra daugiau išskirtų mazgų, galima sukurti nuorodų požymių diagramą (angl. *reference feature diagram*) susiejančią visas atskirtas sudėtingo modelio dalis.

**3.6.9 savybė.** Privalomi požymiai modeliuoja srities *bendrumą*. Variantiniai taškai (sugrupuoti alternatyvieji požymiai) ir variantai (variantinių taškų „vaikai“) modeliuoja srities *variantiškumą*. Apribojimai tarp požymių (*išskyrus, reikalauja*) modeliuoja požymių sąveiką. Sritis gali būti specifikuojama *abstrakčiai* arba *formaliai* (naudojant teiginių logikos notaciją) naudojant bendrumo-variantiškumo sąryšius.

**3.6.10 savybė.** Požymių modeliai yra korektiški vertinant srities požiūriu, jei galioja prielaidos: 1) modelio projektuotojas specifikuodamas modelį naudoja srities ekspertų sukurtus pradinius duomenis; 2) projektuotojas tinkamai manipuliuoja pradiniais srities duomenimis; 3) sąryšiai ir apribojimai tarp požymių formuojami remiantis ekspertų žiniomis.

**3.6.11 savybė.** Sukurti požymių modeliai yra semantiškai teisingi, nes: 1) modeliai yra specifikuoti naudojant FAMILIAR srities kalbą ir įrankius; 2) įrankiu SPLOT atliekamas formalus sukurtų modelių verifikavimas.

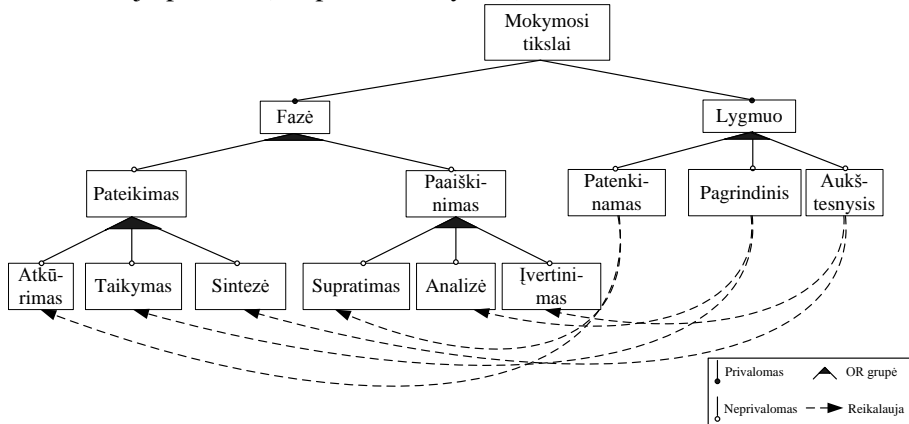
**3.6.12 savybė.** Požymių modelių įvertinimui naudojama daug charakteristikų: modelių skaičius, sudėtingumas, variantiškumo laipsnis, atitikimas specifinio uždavinio reikalavimams, verifikavimo įrankių pateikiamos charakteristikos.

**3.6.13 savybė.** Sukurti modeliai specifikuoja ir modeliuoja informatikos mokymosi sritį priklausomai nuo srities apimties ir modeliavimo tikslų.

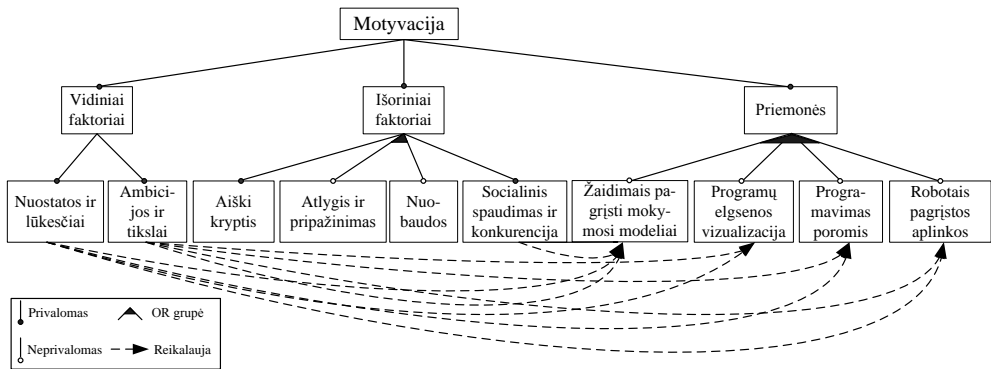
**3.6.14 savybė.** Požymių diagrama vadinama *abstrakčiu požymių modeliu* (angl. *abstract feature model*), jeigu joje yra požymių, kurie kitame kontekste gali būti išskaidyti į smulkesnius. Požymių diagrama vadinama *konkrečiu požymių modeliu* (angl. *concrete feature model*), jeigu visi požymiai yra atominiai. Atominis požymis yra „tėvo“ požymio reikšmė. Konkretus požymių modelis išgaunamas iš abstrakčių požymių aibės, pasirenkant *adekvačius lapus* (angl. *adequate leaves*) ir juos išskaidant, po to suliejant submodelius ir taip suformuojant požymių modelį, skirtą konkrečiam uždaviniui specifikuoti.

### 3.7. Informatikos mokymosi srities posričių požymių modeliai

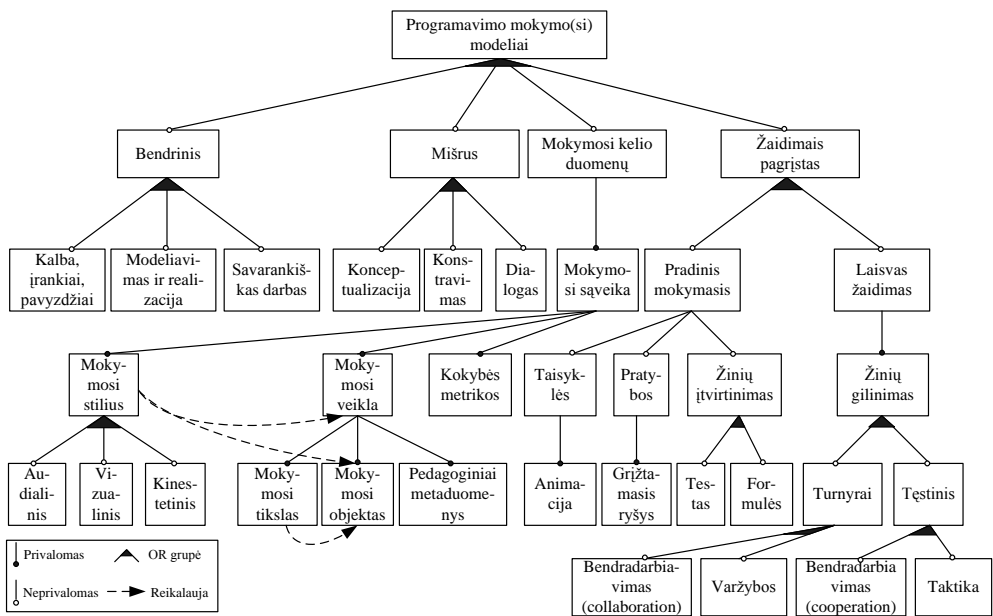
Naudojant 3.5 skyrelyje pateiktą metodą sukurta požymių modelių aibė, perdengianti informatikos mokymosi srities pedagoginius, technologinius, socialinius, turinio aspektus ir apimanti plataus masto pakartotinį panaudojimą. Abstraktūs požymių modeliai sukurti remiantis srities ekspertų patirtimi ir žiniomis (žr. 2 skyrių). Pedagoginius-socialinius srities aspektus perdengia mokymosi tikslų (3.4 pav., 2.10.2 skyrelis), motyvacijos (3.5 pav., 2.10.1 skyrelis), programavimo mokymosi metodų (3.6 pav., 2.10.3 skyrelis), vertinimo (3.7 pav., 2.10.4 skyrelis) ir besimokančiojo profilio (3.8 pav., 2.5 skyrelis) modeliai.



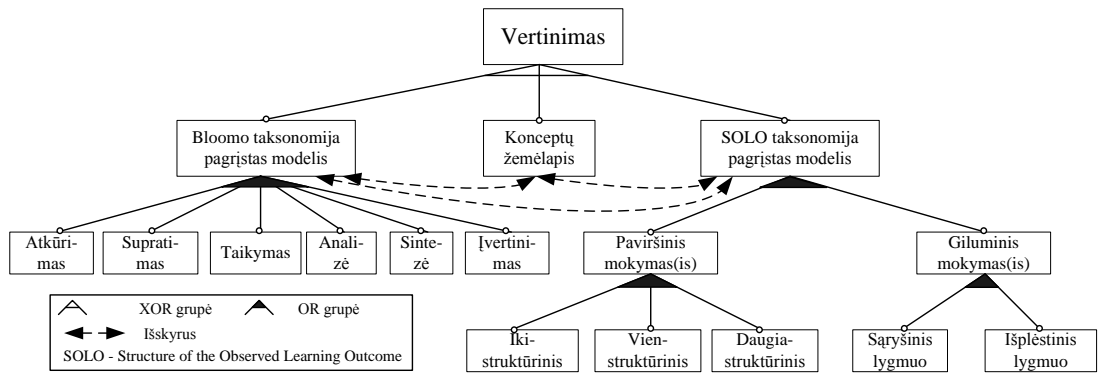
3.4 pav. Informatikos mokymosi srities mokymosi tikslų modelis



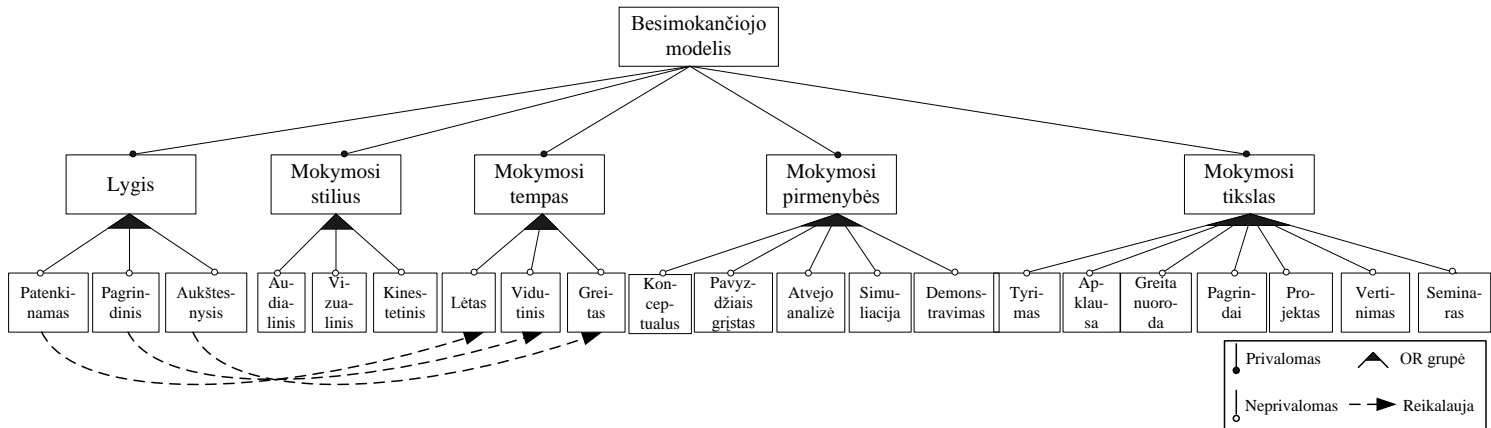
3.5 pav. Informatikos mokymosi srities motyvacijos modelis



3.6 pav. Informatikos mokymosi srities mokymosi metodų modelis



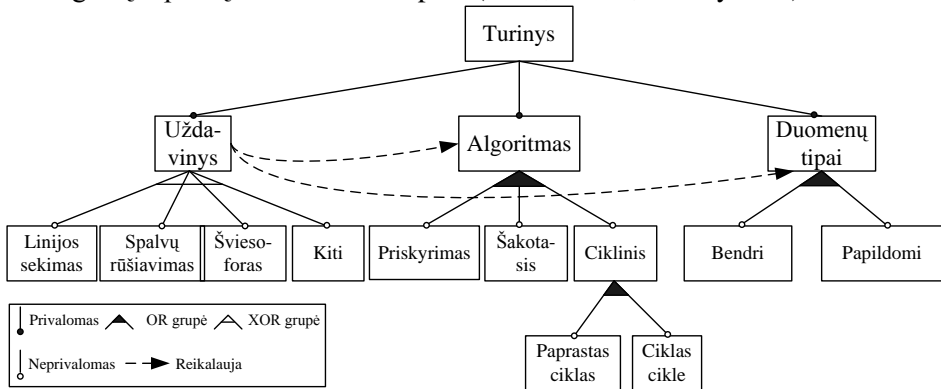
3.7 pav. Informatikos mokymosi srities vertinimo modelis



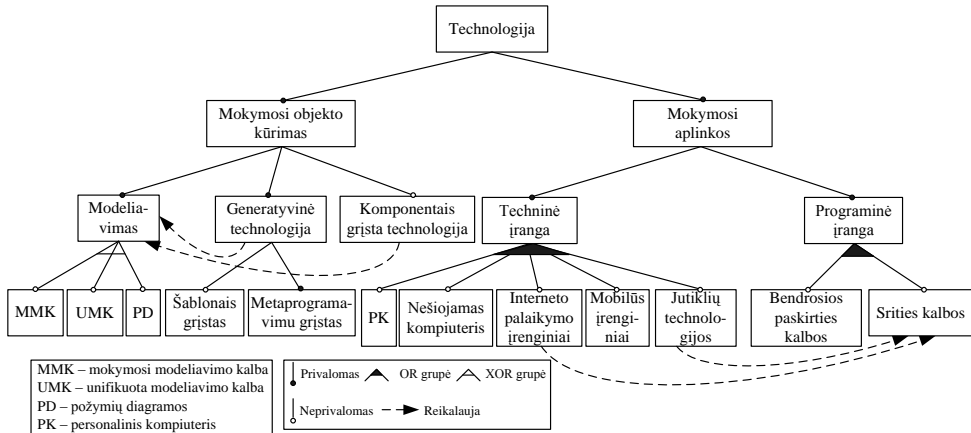
3.8 pav. Besimokančiojo modelis, sudarytas remiantis standartizuotu konteksto modeliu



Turinio aspektus apibrėžiantis modelis pateiktas 3.9 pav. (2.10.5 skyrelis), technologinių aspektų modelis – 3.10 pav. (2.2.5, 2.10.5, 3.3 skyreliai).



3.9 pav. Turinio modelis



3.10 pav. Technologinių aspektų modelis

### 3.8. Požymių pasirinkimas formuojant konkrečius požymių modelius

3.7 poskyrio modeliai yra abstraktūs ir žemiausiame lygyje neapima visų požymių (pvz., turinio, technologijos modeliai). Norint sukurti konkrečius modelius, abstraktūs modeliai turi būti pritaikyti konkrečiai situacijai.

Toliau apibrėžti požymių modelių keitimo procesai, kurie leidžia pasiekti norimą rezultatą kuriant konkretų požymių modelį iš abstrakčių modelių.

**3.7.1 apibrėžimas.** Požymių modelio specializacija (angl. *specialization*) vadinamas procesas, transformuojantis požymių modelį A į modelį B, kurio konfigūracijų skaičius yra modelio A konfigūracijų skaičiaus poaibis [TBK09].

**3.7.2 apibrėžimas.** Požymių modelio rekonstrukcija (angl. *refactoring*) vadinama transformacija, pagerinanti modelio kokybę, kai išlaikomos ar padidinamos modelio konfigūravimo galimybės [GMB11].

**3.7.3 apibrėžimas.** Požymių modelio apibendrinimas (angl. *generalization*) vadinama transformacija, padidinanti priimtinių konfigūracijų skaičių, kai buvusios konfigūracijos išlieka galioti [BM12].

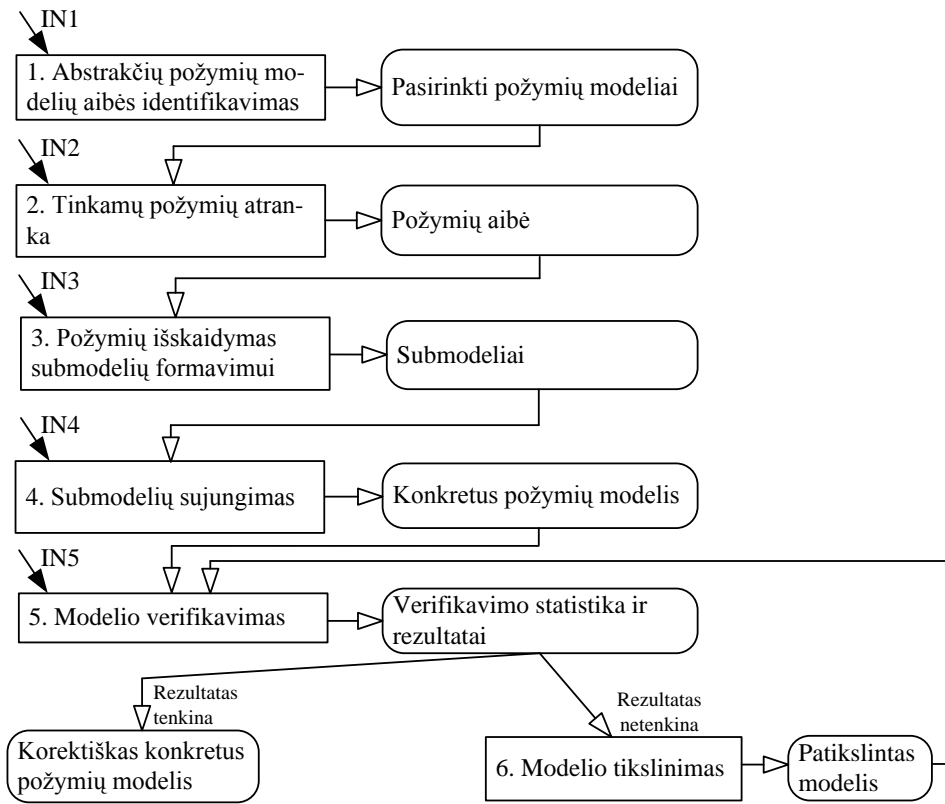
**3.7.4 apibrėžimas.** Požymių modelio dekompozicija (angl. *decomposition*) vadinamas modelio padalinimas į lokalizuotas atskiras dalis [ACL+13].

**3.7.5 apibrėžimas.** Požymių modelių suliejimu (angl. *merging*) vadinama modelių kompozicija, kai suliejamos sutampančios modelių dalys [ACL+13].

**3.7.6 apibrėžimas.** Požymių modelių agregavimu (angl. *aggregating*) vadinama modelių kompozicija, kuri suformuoja naują modelį iš modelių, kurie neturi sutampančių dalių [ACL+13].

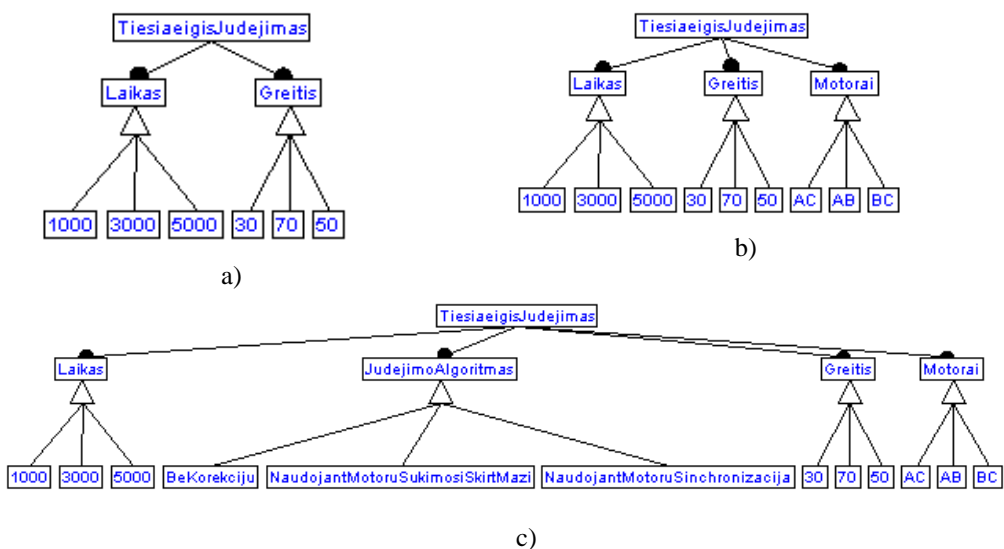
3.11 pav. pateikta konkretaus modelio išgavimo iš abstrakčių modelių schema. IN1 apima konkretaus mokymosi metodo, nagrinėjamos temos, studento profilio ir naudotinos technologijos nustatymą, IN2 – konkretaus uždavinio pasirinkimą ir jo reikalavimus, IN3 – pasirinktus konkretaus uždavinio požymius, IN4 – FAMILIAR įrankius, kuriais naudojantis kuriamas konkretus požymių modelis, IN5 – SPLOT įrankius, skirtus modelio analizei ir verifikavimui.

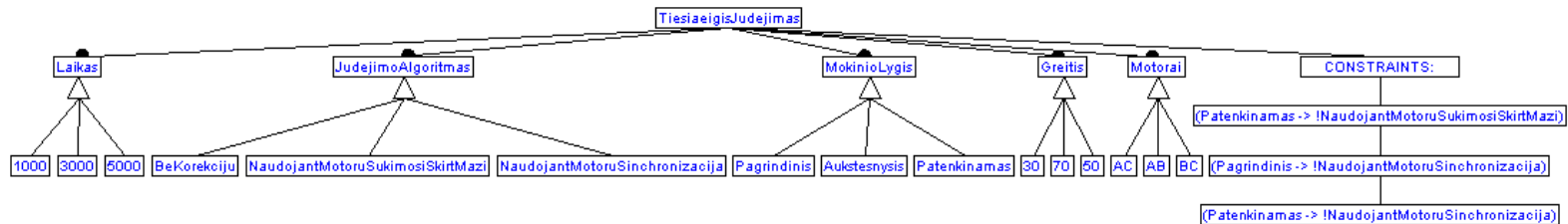
3.12 pav. pateikiami IGMO „Roboto tiesiaiegis judėjimas“ požymių modeliai sukurti naudojant įrankį FAMILIAR. Kiekvienas naujas modelis yra prieš tai buvusio modelio apibendrinimas – prieš tai buvęs modelis išlaiko buvusias konfigūracijas, o papildžius modelį naujais požymiais konfigūracijų skaičius padidėja.



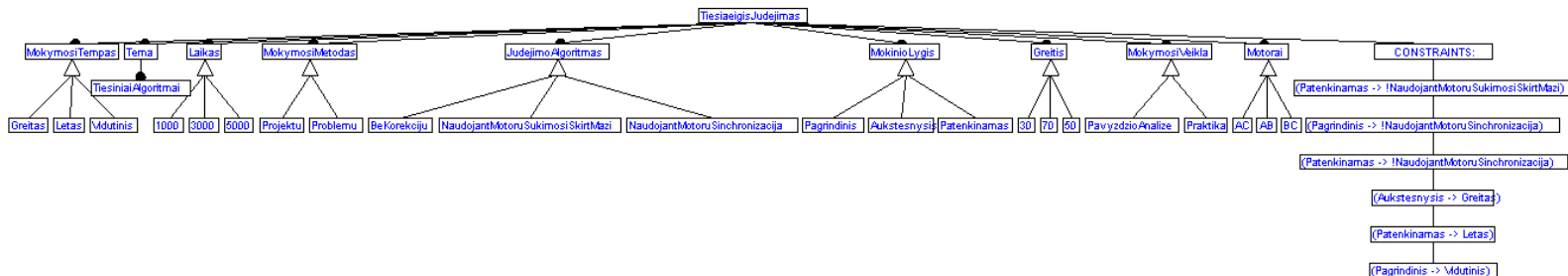
Legenda: □ - Procesas; ○ - Proceso išėjimas; → - Įėjimas/Išėjimas; → (IN) - Išorinis įėjimas.

3.11 pav. Konkretaus požymių modelio išgavimas iš abstrakčių modelių





d)



e)

**3.12 pav.** IGMO „Roboto tiesiaeigis judėjimas“ požymių modeliai: a) požymiai: roboto judėjimo greitis ir judėjimo laikas (apribojimų nėra); b) a modelis papildytas požymiu Motorai (apribojimų nėra); c) b modelis papildytas požymiu Algoritmai (apribojimų nėra); d) c modelis papildytas požymiu Besimokančiojo lygis (su apribojimais); e) apibendrintas IGMO modelis su konteksto požymiais (su apribojimais)

*Pastaba:* 3.12 pav. d) ir e) taip pat pateikiami 1 priede.

### 3.9. Požymių modelių analizė ir įvertinimas

Programų inžinerijoje kurdami kokybiškas programų sistemas, kūrėjai pirmiausia turi sukurti geros kokybės modelius, pradedant pačiais ankstyviausiais projektavimo etapais. Ta pati taisyklė galioja kuriant e.mokymosi srities ir IGMO modelius. Požymių modelių struktūrinės metrikos (angl. *structural metrics*) (3.3 lentelė, 1-12 parametrai) yra svarbūs modelio išorinės kokybės (angl. *external quality*) palaikomumo (angl. *maintainability*) atributo rodikliai [BG11]. Palaikomumas apima požymių analizuojamumą (angl. *analyzability*), kintamumą (angl. *changeability*) ir suprantamumą (angl. *understandability*). Požymių modelių struktūrinių metrikų skaičiavimo metodai pagrįsti binarinėmis sprendimų diagramomis (angl. *Binary Decision Diagrams, BDD*) [Men09]. Požymių modelio neprieštarinumas (angl. *consistency*), konfigūracijose nenaudojamų perteklinių (angl. *dead*) požymių skaičius, konfigūracijų skaičius nustatomas naudojant SAT Solver klasės algoritmus [Men09], [ABH+13].

Požymių modelių kokybės parametrai ir metrikos skaičiuojami naudojant įrankius FAMILIAR ir SPLOT. 3.3 lentelėje pateikti 1-3 ir 5-8 parametrai apibrėžti 3.4.4. skyrelyje. Pateikiami likusių parametru ir metrikų apibrėžimai, nurodant parametro/metrikos numerį 3.3 lentelėje: 4) *esminiais* laikomi požymiai, pasikartojantys visose konfigūracijose; 9) *apribojimų reprezentatyvumas* apibrėžiamas kaip apribojimų kintamųjų skaičiaus ir viso modelio požymių skaičiaus santykis; 10) *skirtingų kintamųjų skaičius apribojimuose* nurodo, kiek skirtingų požymių yra susieta ribojimais „reikalauja“ ir „išskyrus“; 11) *apribojimų išlygų tankis* apibrėžiamas kaip apribojimų skaičiaus ir apribojimų kintamųjų skaičiaus santykis; 12) *medžio gylis* nusako ilgiausią kelią nuo požymių modelio aukščiausio lygmens požymio iki atominių požymių; 13) *konfigūracijų skaičius* nurodo visų galimų konfigūracijų, kurios gali būti išgautos iš požymių modelio, skaičių; 14) *variantiškumo laipsnis* apibrėžiamas kaip galiojančių konfigūracijų skaičiaus ir  $2^n$  santykis, čia  $n$  – modelio požymių skaičius. Kuo variantiškumo laipsnis mažesnis, tuo modelyje yra daugiau apribojimų ir atvirkščiai; 15) požymių modelį aprašanti teiginių logikos formulė pervedama į binarinę sprendimų diagramą. *BDD mazgų skaičius* nurodo, kiek mazgų turi binarinė sprendimų diagrama. Šiuo parametru galima remtis tikrinant modelių ekvivalentiškumą, įvertinant galimų konfigūracijų skaičių; 15-16) *modelio neprieštarinumas* ir *perteklinių požymių skaičius* nustatomi tikrinant požymių modelį aprašančią teiginių logikos formulių aibę.

3.3 lentelėje pateikiami IGMO „Roboto tiesiaiegis judėjimas“ požymių modelių (žr. 3.12 pav. a-e) kokybės parametru skaitinės vertės, nustatytos naudojant įrankius FAMILIAR ir SPLOT.

**3.3 lentelė** IGMO „Roboto tiesiaiegis judėjimas“ požymių modelių (3.12 pav.) kokybės parametrai (parametrai apskaičiuoti naudojant įrankius FAMILIAR ir SPLOT)

Eil. nr.	Metrika / Parametras	3.12 pav. a) modelis	3.12 pav. b) modelis	3.12 pav. c) modelis	3.12 pav. d) modelis	3.12 pav. e) modelis
1.	Požymių skaičius	9	13	17	21	33
2.	Nepivalomų požymių skaičius	0	0	0	0	0
3.	Privalomų požymių skaičius	2	3	4	5	10
4.	Esminių požymių skaičius	3	4	5	6	11
5.	Sugrupuotų požymių skaičius	6	9	12	15	22
6.	OR grupių skaičius	0	0	0	0	0
7.	XOR grupių skaičius	2	3	4	5	8
8.	Apribojimų skaičius	0	0	0	3	6
9.	Apribojimų reprezentatyvumas, %	0.00	0.00	0.00	0.19	0.24
10.	Skirtingų kintamųjų skaičius apribojimuose	0	0	0	4	8
11.	Apribojimų išlygų tankis	0.00	0.00	0.00	0.75	0.75
12.	Medžio gylis	3	3	3	3	3
13.	Konfigūracijų skaičius	9	27	81	162	648
14.	Variantiškumo laipsnis, %	1.7578E0	3.2959E-1	6.1798E-2	7.7248E-3	7.5437E-6
15.	BDD mazgų skaičius	13	19	25	37	59
16.	Modelio neprieštaringumas	Nepriešt.	Nepriešt.	Nepriešt.	Nepriešt.	Nepriešt.
17.	Pertekliniai požymiai	Nėra	Nėra	Nėra	Nėra	Nėra

Išsamesni informatikos mokymosi srities ir išplėstinių generatyvinių mokymosi objektų požymių modelių kokybės tyrimai pateikiami 6 skyriuje.

**3.10. Požymių diagramų taikymo informatikos mokymosi srityje galimybės**

Informatikos mokymosi kontekste požymių diagramos turi stiprų teorinį pagrindą, nes yra pagrįstos teiginių logika, aibių ir grafų teorija. Kadangi informatikos kursuose yra daug formalių žinių, kurios turi būti išreiškiamos įvairiuose abstrakcijos lygmenyse, požymių diagramos šiuo atveju yra tinkamas įrankis, kurį galima adaptuoti vidurinėms ir aukštosioms mokykloms.

Žiūrint iš srities tyrėjų, mokymosi sistemų kūrėjų, mokytojų ir mokinių perspektyvos, pasiūlytas metodas gali būti laikomas metodologiniu instrumentu išgaunant naujas žinias ir jas sisteminant.

Srities tyrėjai požymiais grįstus modelius gali papildyti formalizavimu, žinių sisteminimu modeliuojant aukštame abstrakcijos lygmenyje, požymiais grįstu programavimu, susieti su ontologijomis, neraiškiaja logika, aspektais grįstu modeliavimu.

Mokymosi sistemų kūrėjai, naudodami sukurtą metodą, taiko programų sistemų kūrėjų naudojamas teorijas, metodus ir įrankius, standartizacijos iniciatyvas.

Iš kurso kūrėjo ir mokytojo perspektyvos požymių modeliai yra svarbūs tuo, kad: 1) suteikia galimybę specifikuoti MO skirtinguose abstrakcijos lygmenyse; 2)

adaptuoti MO skirtingiems poreikiams (mokymosi kontekstai, turinio pokyčiai ir išplėtimai); 3) požymių modelius galima traktuoti kaip specifikacijos dokumentą.

Kadangi požymių modeliai turi grafinę notaciją, intuityviai suprantami ir jų sintaksė yra paprasta, juos naudoti gali ir studentai.

IGMO kontekste požymių diagramos leidžia: 1) aiškiai išreikšti konteksto ir turinio variantiškumą, nes variantiškumas yra esminė IGMO savybė; 2) valdyti variantiškumo pokyčius; 3) valdyti IGMO sudėtingumą; 4) palaikyti IGMO evoliuciją jo gyvavimo cikle (angl. *life cycle*); 5) požymių diagramos yra specifikacijos dokumentas, norint suprasti konceptą ir jį realizuoti.

Metodo silpnoji vieta – požymių diagramų notacijų įvairovė, dėl kurios požymių modelių elementai interpretuojami skirtingai. Tai sukelia formalizavimo ir standartizavimo sunkumų. Taip pat egzistuoja įvairios tekstinės požymių modelių notacijos ir nėra standartinių modeliavimo bei modelių verifikavimo įrankių: kiekvienas tyrėjas pasirenka notaciją ir įrankius savo nuožiūra.

### **3.11. Santrauka ir apibendrinimas**

Modeliavimas yra galingas instrumentas, padedantis išgauti žinias apie sritį ankstyvuosiuose programinės įrangos sistemų kūrimo etapuose, nes suteikia galimybes projektuoti aukštame abstrakcijos lygmenyje, dalintis srities žiniomis, išplėsti pakartotinį panaudojimą per automatizavimą. Tai užtikrina didesnę produktyvumą, geresnę kokybę, trumpesnę pateikimo į rinką laiką. Darbe pateikiama informatikos mokymosi srities analizė ir modeliavimas, pagrįstas programų inžinerijoje naudojamais modeliavimo principais.

Informatikos (programavimo) mokymosi sritis modeliuojama požymiais grįstais modeliais. Plačios apimties pakartotinio panaudojimo požiūriu sujungiami dviejų sričių artefaktai ir pažinimas: požymiais pagrįsti konceptai iš programų inžinerijos ir mokymosi konceptai iš programavimo e.mokymosi. Programų inžinerijos sritis traktuojama kaip sprendimų sritis (angl. *solution domain*), programavimo mokymosi – kaip užduočių sritis (angl. *task domain*). Tuo remiantis pasiūlyta metodika, kuri formaliai gali būti suprantama kaip problemų srities (angl. *problem domain*) atvaizdavimas į sprendimų sritį. Metodiką sudaro du etapai. Pirmajame etape naudojami procesai leidžia sukurti apibendrintą požymių modelių erdvę, kurią iš anksto apibrėžia modeliavimo tikslas, kad būtų tenkinami plataus pakartotinio panaudojimo reikalavimai. Pirmojo etapo modeliavimo rezultatas – abstraktūs požymių modeliai. Antrasis etapas vykdomas sukuriant konkrečius požymių modelius, apibrėžus konkrečius reikalavimus. Pastarasis etapas dar vadinamas modelio detalizavimu (angl. *refinement*).

Detalizuoti modeliai yra aukšto abstrakcijos lygmens specifikacijos, specifikuojančios mokymosi variantiškumą – svarbiausią išplėstinio generatyvinio mokymosi objekto atributą.

### 3.12. Išvados

1. Pasiūlytas informatikos mokymosi srities modeliavimo metodas, pagrįstas koncepcijų atskirties principu ir procesais, aprašomais į tikslą orientuotais įėjimų-išėjimų sąryšiais, gali būti naudojamas modeliuojant kitas sudėtingas heterogenines sritis, pasižyminčias dideliu variantiškumu.

2. Pagal pateiktus reikalavimus konkretus požymių modelis išgaunamas iš abstrakčių modelių.

3. Detalizuoti konkretūs modeliai susiaurina pasirinkimo apimtį ir visą (abstrakčią) požymių erdvę padaro tinkamą pakartotinai panaudoti konkrečiame kontekste.

4. Konkretūs požymių modeliai specifikuoja mokymosi variantiškumą tiesiogiai per variantinių taškų ir jų variantų aibę.

5. Aiškiai išreikštas mokymosi variantiškumas yra aukšto lygmens specifikacija naudojama IGMO kūrimui.

6. Konkretaus požymių modelio variantinių taškų ir variantų aibė iš anksto specifikuoja erdvę galimų informatikos mokymosi adaptavimų valdymui.



## 4. IŠPLĖSTINIŲ GENERATYVINIŲ MOKYMOSI OBJEKTŲ KŪRIMO METODAS

### 4.1. Įvadas

3 skyriuje atlikta informatikos mokymosi srities analizė, kurios rezultatas – pedagoginės, turinio ir technologinės posričių modeliai, išryškinantys svarbiausius srities aspektus, į kuriuos reikia atsižvelgti kuriant informatikos mokymuisi skirtus šiuolaikinius IGMO. Šiame skyriuje išplečiamas teorinis IGMO pagrindimas. Požymių modeliai taip pat sudaro IGMO teorinio pagrindimo dalį, tačiau visa tai pateikta netiesiogiai, nenaudojant IGMO sąvokos.

Darbe nagrinėjama problemų sritis (angl. *problem domain*), apimanti informatikos mokymąsi, pasižymintį mokymosi variantiškumu (angl. *learning variability*). Sprendimų sritimi (angl. *solution domain*) tyrimo kontekste suprantama technologija, kuri naudojama problemų srities įgyvendinimui.

Išplėstinis generatyvinis mokymosi objektas yra mokymosi variantiškumo atvaizdavimo į įgyvendinimo technologiją produktas, palaikantis iš anksto aprašytų požymių realizaciją.

Šio skyriaus tikslas – sukurti problemų srities atvaizdavimo į sprendimų sritį metodą. Formuluojami tokie skyriaus uždaviniai:

- sukurti IGMO reikalavimų modelį;
- remiantis reikalavimų modeliu aukštame lygmenyje specifikuoti IGMO požymių diagrama;
- parinkti sprendimo sritį problemų srities įgyvendinimui;
- sukurti aukšto lygmens IGMO specifikacijos transformavimo į vykdomąją IGMO specifikaciją taisyklės;
- įvertinti sukurtą IGMO technologinį sudėtingumą.

4.2 skyrelyje apibrėžtos pagrindinės mokymosi srities variantiškumo sąvokos; 4.3 skyrelyje pateiktas IGMO reikalavimų modelis; 4.4 skyrelyje suformuotas IGMO specifikacijos požymių diagrama metodas; 4.5 skyrelyje pagrįstas metaprogramavimo technologijos tinkamumas IGMO vykdomosios specifikacijos kūrimui; 4.6 skyrelyje apibrėžtos pagrindinės metaprogramavimo technologijos sąvokos; 4.7 skyrelyje suformuluotos IGMO aukšto lygmens specifikacijos transformavimo į vykdomąją specifikaciją taisyklės; 4.8 skyrelyje pateiktos IGMO savybės; 4.9 skyrelyje įvertintas IGMO sudėtingumas; 4.10 skyrelyje išnagrinėtos IGMO sudėtingumo valdymo galimybės; 4.11 skyrelyje suformuluotos skyriaus išvados.

### 4.2. Informatikos mokymosi srities variantiškumo pagrindinės sąvokos

**4.2.1 apibrėžimas.** Mokymosi variantiškumas (MV) yra pedagoginio variantiškumo (PV), socialinio variantiškumo (SV), ir turinio variantiškumo (TV) kompozicija [ŠBD13]. Formaliai MV aprašomas 4.2.1 išraiška:

$$MV = PV \circ SV \circ TV ; \quad (4.2.1)$$

čia „ $\circ$ “ – specifinis dedamųjų integravimo operatorius.

**4.2.2 apibrėžimas.** Pedagoginio variantiškumo modeliu vadinamas konkretus požymių modelis (žr. 3.6.14 savybę) *PVM*, sudaromas naudojant požymiais grįstą kalbą ir įrankius, formaliai aprašomas 4.2.2 išraiška:

$$PVM = \langle PD, PB_m, PD_a, PD_o, REQP, EXCP \rangle; \quad (4.2.2)$$

čia  $PD = (PP, PB, pp)$  yra medis, turintis baigtinę aibę pedagoginių požymių  $PP$ ,  $PB \subseteq PP \times PP$  yra baigtinė briaunų aibė,  $pp \in PP$  yra pagrindinis pedagoginis požymis;  $PB_m \subseteq PB$  yra briaunų, apibrėžiančių privalomus pedagoginius požymius su jų tėviniais požymiais, aibė;  $PD_a \subseteq P(PP) \times PP$ ,  $PD_o \subseteq P(PP) \times PP$  apibrėžia alternatyviųjų ir pasirenkamųjų pedagoginių požymių grupes ir yra vaikų požymių bei jų bendrų tėvų požymių porų aibės; *REQP* apibrėžia pedagoginių požymių apribojimą „reikalauja“, *EXCP* – apribojimą „išskyrus“.

**4.2.3 apibrėžimas.** Socialinio variantiškumo modeliu vadinamas konkretus požymių modelis (žr. 3.6.14 savybę) *SVM*, sudaromas naudojant požymiais grįstą kalbą ir įrankius, formaliai aprašomas 4.2.3 išraiška:

$$SVM = \langle PS, SB_m, PS_a, PS_o, REQS, EXCS \rangle; \quad (4.2.3)$$

čia  $PS = (SP, SB, ps)$  yra medis, turintis baigtinę aibę socialinių požymių  $SP$ ,  $SB \subseteq SP \times SP$  yra baigtinė briaunų aibė,  $ps \in SP$  yra pagrindinis socialinis požymis;  $SB_m \subseteq SB$  yra briaunų, apibrėžiančių privalomus socialinius požymius su jų tėviniais požymiais, aibė;  $PS_a \subseteq P(SP) \times SP$ ,  $PS_o \subseteq P(SP) \times SP$  apibrėžia alternatyviųjų ir pasirenkamųjų socialinių požymių grupes ir yra vaikų požymių bei jų bendrų tėvų požymių porų aibės; *REQS* apibrėžia socialinių požymių apribojimą „reikalauja“, *EXCS* – apribojimą „išskyrus“.

**4.2.4 apibrėžimas.** Turinio variantiškumo modeliu vadinamas konkretus požymių modelis (žr. 3.6.14 savybę) *TVM*, sudaromas naudojant požymiais grįstą kalbą ir įrankius, formaliai aprašomas 4.2.4 išraiška:

$$TVM = \langle PT, TB_m, PT_a, PT_o, REQT, EXCT \rangle; \quad (4.2.4)$$

čia  $PT = (TP, TB, pt)$  yra medis, turintis baigtinę aibę turinio požymių  $TP$ ,  $TB \subseteq TP \times TP$  yra baigtinė briaunų aibė,  $pt \in TP$  yra pagrindinis turinio požymis;  $TB_m \subseteq TB$  yra briaunų, apibrėžiančių privalomus turinio požymius su jų tėviniais požymiais, aibė;  $PT_a \subseteq P(TP) \times TP$ ,  $PT_o \subseteq P(TP) \times TP$  apibrėžia alternatyviųjų ir pasirenkamųjų turinio požymių grupes ir yra vaikų požymių bei jų bendrų tėvų požymių porų aibės; *REQT* apibrėžia turinio požymių apribojimą „reikalauja“, *EXCT* – apribojimą „išskyrus“.

**4.2.5 apibrėžimas.** Mokymosi variantiškumo modeliu vadinamas konkretus požymių modelis (žr. 3.6.14 savybę) *MVM*, kuris gaunamas atlikus *PVM*, *SVM* ir *TVM* agregavimą (žr. 3.7.6 apibrėžimą, [ACL+13]), sudaromas naudojant požymiais grįstą kalbą ir įrankius, formaliai aprašomas 4.2.5 išraiška:

$$MVM = (PVM) \odot (SVM) \odot (TVM); \quad (4.2.5)$$

čia „ $\odot$ “ – požymių modelių agregavimo operatorius.

### 4.3. Išplėstinio generatyvinio mokymosi objekto reikalavimų modelis

Reikalavimų analizės (angl. *requirements analysis*) svarbą parodo tai, kad programų inžinerijoje reikalavimų inžinerija (angl. *requirements engineering*) išskiriama kaip atskira šaka ir reikalavimų analizės kūrimo etapas laikomas pačiu svarbiausiu [Lap13], nes nuo reikalavimų analizės kokybės priklauso tolesnė sistemos projektavimo sėkmė.

Aukščiausiam reikalavimų modelio lygmenyje apibrėžiama vartotojo reikalavimų aibė, išreiškianti problemų srities reikalavimus ir nepriklausanti nuo sprendimų srities. Antrajame lygmenyje apibrėžiami reikalavimai sistemai, postuluojamas abstraktus sprendimas. Trečiajame lygmenyje apibrėžiami reikalavimai posistemei. Reikalavimų lygmenų skaičius priklauso nuo kuriamos sistemos apimties ir sudėtingumo [KJ09].

Sudarant IGMO reikalavimų modeliavimo metodą taikytas *analogijos* principas. Tyrimo kontekste vartotojo reikalavimai apibrėžiami kaip vartotojo poreikių identifikavimo ir specifikavimo procesas kuriamam IGMO. Aukščiausiam lygmenyje vartotojo reikalavimų aibė sudaryta remiantis 2.9 skyrelyje atlikta šaltinių analize. Analizės rezultatas – aukšto lygmens požymių modeliai, perdengiantys programavimo mokymosi sritį (žr. 3.7. poskyrį).

Reikalavimų sistemai apibrėžimas turi panašumų su reikalavimais IGMO kontekstui. Reikalavimai IGMO kontekstui išgaunami iš 3.7 poskyrio pedagoginius-socialinius srities aspektus perdengiančių modelių atliekant modelių *specializaciją* (žr. 3.7.1 apibrėžimą, [ACL+13]) ir specializuotų modelių *agregavimą* (žr. 3.7.6 apibrėžimą, [ACL+13]) arba *suliejimą* (žr. 3.7.5 apibrėžimą, [ACL+13]).

Reikalavimai posistemei yra analogiški IGMO turinio reikalavimams. Reikalavimai IGMO turiniui išgaunami analizuojant informatikos mokymosi planą (angl. *curriculum*) ir konkrečiam tikslui pasiekti skirtų giminingų MO egzempliorių aibę.

### 4.4. Išplėstinio generatyvinio mokymosi objekto specifikavimas požymių diagrama

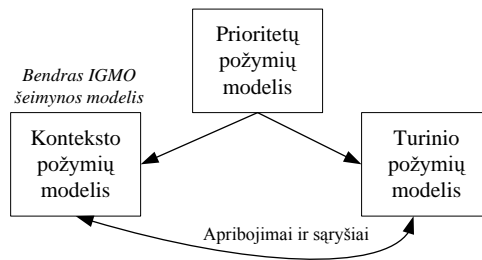
Specifikuojant IGMO požymių diagrama, susiduriama su požymių modelio sudėtingumo valdymo problema, nes požymių modelį sudaro didelis skaičius požymių ir sąryšiai tarp jų. Požymių modelių sudėtingumo valdymo problema programų inžinerijoje nagrinėjama Charnecki *ir kt.* [CHE04], [CHE05], Reiser ir Weber [RW07], Hartmann ir Trew [HT08], Classen *ir kt.* [CHH09], Bagheri *ir kt.* [BAG+10] darbuose. Požymių modelių sudėtingumas valdomas įvairiais būdais: 1) įvedant pakopinę požymių modelių konfigūraciją (angl. *staged configuration*)

atliekant specializaciją arba daugiapakopę modelių konfigūraciją, kai kiekvienos pakopos konfigūracijos pasirinkimus apibrėžia atskiri požymių modeliai [CHE04], [CHE05]; 2) formuojant daugiapakopius požymių medžius, kur „tėvo“ modelis tarnauja kaip nuorodos (angl. *reference*) modelis „vaikų“ modeliams [RW07]; 3) nustatant požymių prioritetus ir pasirenkant požymius atsižvelgiant į apribojimus ir sąryšius tarp požymių [BAG+10]; 4) sukuriant konteksto variantiškumo modelį (angl. *Context Variability model*), kuris susiejamas su standartiniu požymių modeliu ir sukuriamas naujas modelis, vadinamas produktų šeimos požymių modeliu (angl. *Multiple Product Line Feature Model, MPL-Feature Model*) [HT08].

Toliau apibrėžiamos pagrindinės sąvokos, būtinos IGMO specifikavimui požymių modeliais.

**4.4.1 apibrėžimas.** IGMO šeima vadinama aibė MO, kuriuos apibrėžia bendrų požymių, tenkinančių specifinius vartotojų reikalavimus, aibė.

**4.4.2 apibrėžimas.** IGMO požymių modeliu vadinamas modelis, sudarytas iš konteksto ir turinio požymių modelių, kuriuos semantiškai susieja prioritetų požymių modelis ir apribojimai bei sąryšiai tarp konteksto ir turinio modelių požymių (žr. 4.1 pav.).



4.1 pav. Apibendrintas IGMO šeimos modelis

**4.4.3 apibrėžimas.** IGMO konteksto požymių modeliu (*KPM*) vadinamas konkretus požymių modelis, kuris yra bendras visai IGMO šeimynai. Konteksto modelis yra informatikos mokymosi srities posričių specializuotų požymių modelių agregacijos rezultatas, formaliai aprašomas 4.4.1 išraiška:

$$KPM = MTPM_{spec} \odot MPM_{spec} \odot MMPM_{spec} \odot VPM_{spec} \odot BPM_{spec}; \quad (4.4.1)$$

čia  $MTPM_{spec} \subset MTMP$  – mokymosi tikslų PM;  $MPM_{spec} \subset MPM$  – motyvacijos PM;  $MMPM_{spec} \subset MMPM$  – mokymosi metodų PM;  $VPM_{spec} \subset VPM$  – vertinimo PM;  $BPM_{spec} \subset BPM$  – besimokančiojo PM; „ $\odot$ “ – požymių modelių agregavimo operatorius.

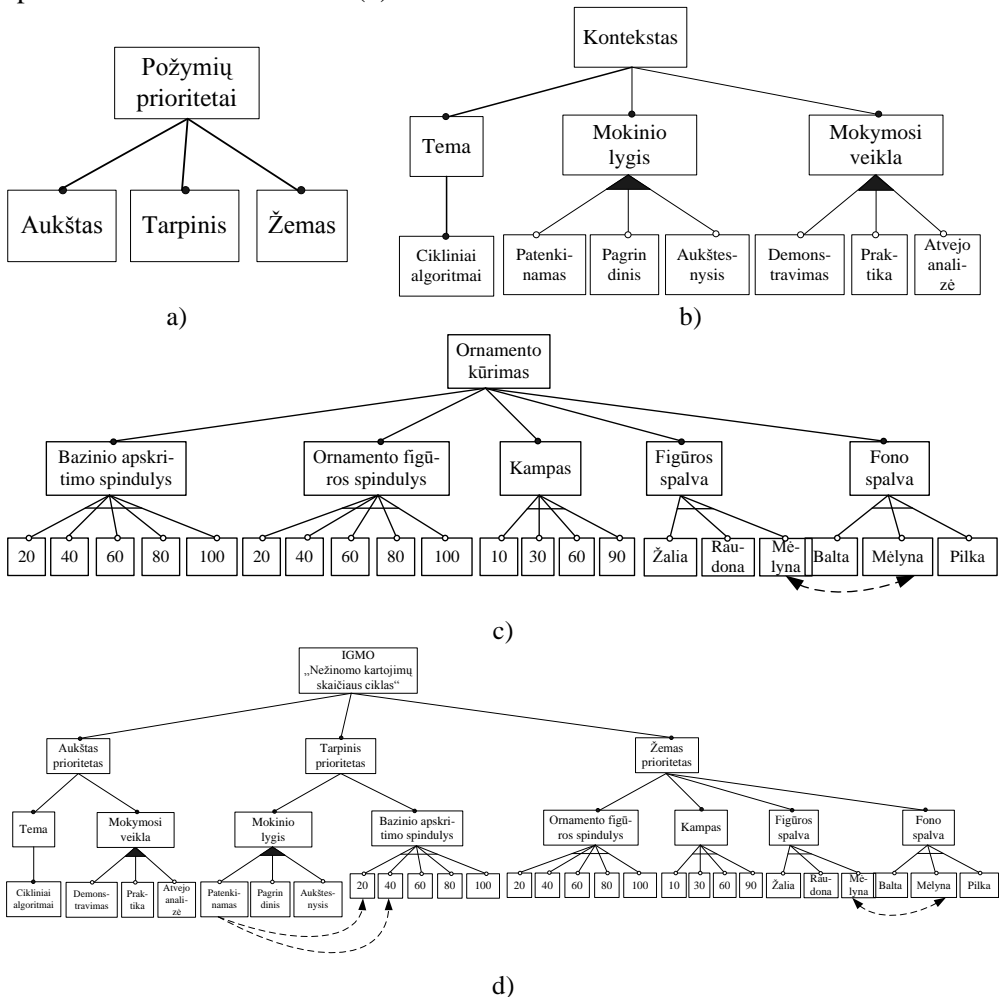
**4.4.4 apibrėžimas.** IGMO turinio požymių modeliu vadinamas konkretus požymių modelis, sudarytas remiantis IGMO turinio reikalavimų modeliu ir apibrėžiamas kaip turinio variantiškumo modelis (žr. 4.2.4 apibrėžimą).

**4.4.5 apibrėžimas.** IGMO prioritetų požymių modeliu (*PPM*) vadinamas konkretus požymių modelis bendras visai IGMO šeimynai, formaliai aprašomas 4.4.2 išraiška:

$$PPM = \langle PM, PB_m, REQ_P \rangle; \quad (4.4.2)$$

čia  $PM = (PP, PB, pp)$  yra medis, turintis baigtinę aibę prioritetą nusakančių požymių  $PP$ ,  $PB \subseteq PP \times PP$  yra baigtinė briaunų aibė,  $pp \in PP$  yra pagrindinis prioritetų požymis;  $PB_m \subseteq PB$  yra briaunų, apibrėžiančių privalomus prioritetų požymius, aibė;  $REQ_P$  apibrėžia prioriteto požymių apribojimą „reikalauja“.

4.2 pav. pateikiami prioritetų (a), IGMO konteksto (b) požymių modeliai ir IGMO „Nežinomo kartojimų skaičiaus ciklas“ turinio modelio (c) pavyzdžiai bei apibendrintas IGMO modelis (d).



**4.2 pav.:** a) – IGMO prioritetų modelis, b) – IGMO konteksto modelis, c) IGMO „Nežinomo kartojimų skaičiaus ciklas“ turinio modelis, d) apibendrintas IGMO modelis

#### 4.5. Metaprogramavimo technologija ir išplėstiniai generatyviniai mokymosi objektai

Informatikos mokymasis yra daugiatis ir daugiapakopis procesas, kurio efektyvumui užtikrinti tradicinių MO modelių nebepakanka [ŠB12]. Generatyvinio mokymosi objekto koncepcija, 2004 m. pasiūlyta Boyle ir jo kolegų [LBM+04], įnešė į e.mokymosi sritį daug pokyčių: pagrindinė GMO idėja – MO struktūros atskyrimas nuo turinio [LBM+04], [MLB05], [Boy09], [Boy10], [BR12]. Atskyrimas vykdomas panaudojant generatyvinę technologiją, todėl naujos kartos MO pavadinti generatyviniais MO.

Programavimo mokymuisi skirti IGMO, žiūrint iš mokytojo ir besimokančiojo perspektyvos, yra aukšto lygio programos (metaprogramos), generuojančios žemesnio lygio programas automatiškai naudojant įrankius. Generatyvinių mokymosi objektų idėjos pradininkai GMO realizavimui naudojo šablonais grįstą (angl. *template-based*) technologiją. Darbe taikoma heterogeninio metaprogramavimo technologija, kuri suteikia galimybę realizuoti metaprogramas tiesiogiai išreiškiant srities uždavinio variantiškumą. Tyrimo kontekste srities variantiškumas suprantamas kaip mokymosi variantiškumas (žr. 4.2.1 apibrėžimą).

Mokymosi variantiškumo sąvoką paaiškinsime nagrinėdami paprastą pavyzdį. Interpretuosime paprastą MO iš variantiškumo perspektyvos įvairiose srityse. Išnagrinėkime paprastą MO – tiesinę funkciją  $y = ax + b$ . Matematikoje toks funkcijos užrašymas išreikšta forma vaizduoja argumento  $x$ , funkcijos  $y$  bei koeficientų  $a$  ir  $b$  reikšmes:  $x, y, a, b \in R$ . Taip pat gali būti nagrinėjami atskiri tiesinės funkcijos atvejai ( $a > 0; a < 0; a = 0; b \neq 0$  ir pan.). Visa tai parodo matematikos srities variantiškumą, kai funkcija gali būti apibrėžta kaip argumentų srities atvaizdavimas į funkcijos reikšmių sritį.

Programavimo mokymosi srityje tiesinė funkcija lengvai transformuojama į kompiuterio programą, skaičiuojančią  $y$  reikšmes, kai  $x$ ,  $a$  ir  $b$  reikšmių aibės iš anksto apibrėžtos. Nagrinėjamu atveju variantiškumas programos kodo viduje gali būti vertinamas iš dalies tiesiogiai ir iš dalies netiesiogiai. Skirtumas tarp programos ir metaprogramos paaiškinamas naudojant tą patį tiesinės funkcijos pavyzdį. Programa visada grąžina apskaičiuotą konkrečią  $y$  reikšmę. Pvz., jei  $a = b = x = 2$ , tuomet  $y = 6$ . Metaprograma kaip rezultatą grąžina programą (programas): jei  $a = b = 2$ , tuomet  $y = 2 * x + 2$ . Skirtumas tarp programos ir metaprogramos suteikia galimybę išplėsti pakartotinį MO panaudojimą sukuriant vartotojo poreikius atitinkantį programos egzempliorių.

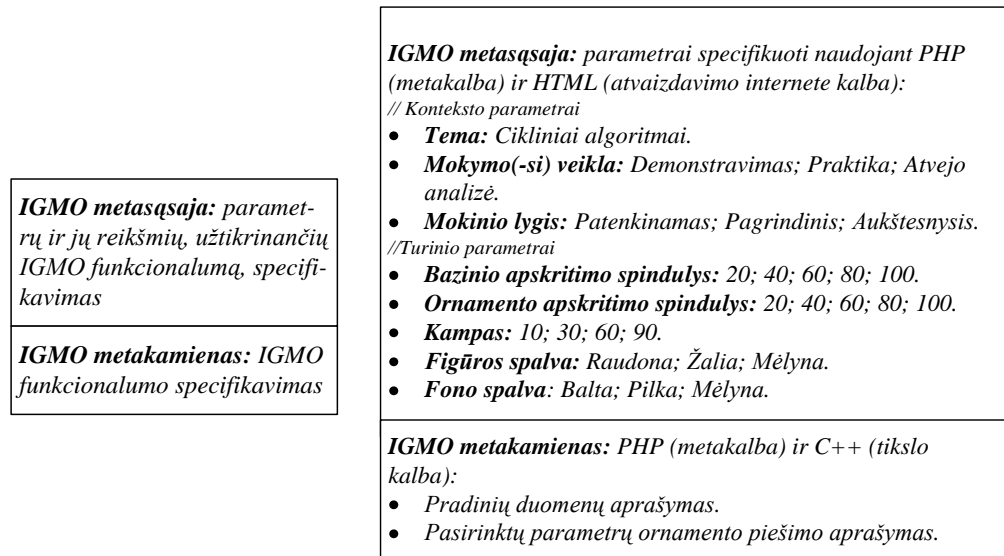
**4.5.1 apibrėžimas.** Semantiškai IGMO yra tiesioginis mokymosi variantiškumo atvaizdavimas naudojant heterogeninio metaprogramavimo technologiją [ŠBD13].

**4.5.2 apibrėžimas.** Struktūriškai IGMO sudaro iš anksto specifikuotų automatiškai pagal vartotojo poreikius generuojamų MO egzempliorių aibė arba konkretus egzempliorius [ŠBD13]. Formaliai IGMO modelis užrašomas:

$$IGMO = MS \times MK ; \quad (4.5.1)$$

čia  $MS$  – metasąsaja,  $MK$  – metakamienas, „ $\times$ “ – atvaizdavimas.

Struktūrinis apibendrintas IGMO modelis (4.3 pav. a) yra aukštesnio lygmens, nepriklausomas nuo realizacijos. Konkretaus IGMO „Nežinomo kartojimų skaičiaus ciklas“ modelis (4.3 pav. b) yra žemesnio lygmens, susietas su realizavimo technologija.



a)

b)

**4.3 pav.:** IGMO modelis: a) – apibendrinta IGMO modelio struktūra, b) IGMO „Nežinomo kartojimų skaičiaus ciklas“ modelis

Išplėstinio generatyvinio mokymosi objekto modelio realizacija pateikiama 4.8. skyrelyje ir 4.4. pav.

## 4.6. Metaprogramavimo technologijos pagrindinės sąvokos

**4.6.1 apibrėžimas.** Metaprogramavimas yra aukšto lygmens programavimo paradigma, kai generatyvinis pakartotinis panaudojimas užtikrinamas sukuriant apibendrintas programas, vadinamas metaprogramomis.

Priklausomai nuo to, kaip realizuojama koncepcijų atskirtis aukštesniame arba žemesniame metaprogramavimo lygmenyse, skiriamas homogeninis ir heterogeninis metaprogramavimas. Homogeninis metaprogramavimas remiasi netiesiogine koncepcijų atskirtimi, naudojama viena kalba, kuri padalijama į du poaibius: aukštesnį ir žemesnį. Žemesniame lygmenyje užrašomas bazinis funkcionalumas, aukštesniame – apibendrinimas.

Heterogeninis metaprogramavimas remiasi tiesiogine koncepcijų atskirtimi. Naudojamos mažiausiai dvi nepriklausomos kalbos. Žemesnio lygmens kalba užrašomas srities funkcionalumas, o aukštesnio lygmens kalba – bendrinimo algoritmas.

IGMO vykdomųjų specifikacijų kūrimui pasirinkta heterogeninio metaprogramavimo technologija. Žemiau apibrėžtos pagrindinės heterogeninio metaprogramavimo sąvokos.

**4.6.2 apibrėžimas.** Heterogeninis metaprogramavimas yra paradigma, kai rašant metaprogramas naudojamos mažiausiai dvi nepriklausomos kalbos. Tikslo kalba  $T$  (angl. *target language*) yra žemesnio lygio kalba, kuri išreiškia bazinį srities funkcionalumą. Aukštesnio lygio kalba, vadinama metakalba  $M$  (angl. *meta-language*) per parametrus, aprašančius srities variantiškumą, užrašomas bendrinimo algoritmas.

Metakalba disertacijoje pasirinkta PHP kalba, kuri plačiai taikoma ne tik interneto svetainių kūrimui, bet ir kaip universali programavimo kalba (angl. *general purpose language*). Tikslo kalbų pasirinkimą lėmė sprendžiami uždaviniai: kuriami IGMO programavimo mokymuisi ir sugeneruoti MO egzemplioriai yra programos ar jų fragmentai, kurie užrašyti Pascal, C++, RobotC programavimo kalbomis.

**4.6.3 apibrėžimas.** Heterogeninė metaprograma yra programų generatorius, iš metaprogramos specifikacijos automatiškai kuriantis programų egzempliorius (angl. *program instances*).

**4.6.4 apibrėžimas.** Struktūrinį metaprogramos modelį (angl. *structural model*)  $MPM$  sudaro tarpusavyje susiję metasąsajos  $MS$  (angl. *meta-interface*) ir metakamieno  $MK$  (angl. *meta-body*) modeliai:

$$MPM = MS \cup MK. \quad (4.6.1)$$

**4.6.5 apibrėžimas.** Metasąsajos modelį  $MS$  sudaro  $n$  dimensijų netuščia metaparametrų  $P$  ir jų  $R$  reikšmių erdvė:

$$MS = \{P; R\}; \quad (4.6.2)$$

čia  $P$  – pilna  $n$  metaparametrų vardų aibė,  $n = |P|$ ,  $R$  – nurodyta visų parametrų reikšmių aibė. Kiekvienas metaparametras  $P_i (P_i \in P)$  turi savo reikšmių aibę  $\{r_{i1}, r_{i2}, \dots, r_{im}\} \subset R$ , tuomet  $P_i := R_i = \{r_{i1}, r_{i2}, \dots, r_{im}\} \in R$ , čia  $im$  – parametro  $P_i$  reikšmių skaičius, „:=“ reiškia „yra apibrėžtas“.

**4.6.6 apibrėžimas.** Metaprogramos parametrai yra metaprogramos sąsajoje ir kamiene esančios sintaksiškai orientuotos esybės (angl. *syntax-driven entities*), išreiškiančios srities variantiškumą nepriklausomai nuo srities semantikos.

**4.6.7 apibrėžimas.** Parametrai  $P_i$  ir  $P_j (P_i, P_j \subseteq P (i \neq j))$  yra nepriklausomi (angl. *independent*), jei su bet kuria pasirinkta reikšmių pora  $\{r_{ik}, r_{js}\} (r_{ik} \in P_i, r_{js} \in P_j, \text{ čia } k \in [1, im] \text{ ir } s \in [1, jq])$  metaprograma veikia korektiškai. Kitais atvejais parametrai bus priklausomi (angl. *dependent*).

Kartais priklausomi parametrai yra traktuojami kaip sąveikaujantys (ypač požymių modelių sudarymo kontekste). Tyrimo kontekste parametrų priklausomybė išreiškia variantiškumo sąveiką (angl. *variability interaction*).

Pateiktas apibrėžimas apibrėžia sąryšį „vienas su vienu“. Norint apibrėžti visą parametrų sąveikos erdvę konstruojamas parametrų sąveikos grafas  $S(P, B)$ .  $P$  viršūnių aibė atitinka metaparametrus, briaunų  $B$  aibė apibrėžiama taip: visiems  $i$  ir  $j$  egzistuoja briauna  $b_{ij} = 1$ , jeigu  $P_i$  ir  $P_j$  yra priklausomi (žr. 4.6.7 apibrėžimą), kitais atvejais  $b_{ij} = 0$  (briauna neegzistuoja) ( $P_i, P_j \in P, b_{ij} = (P_i, P_j) \in B$ ).



**4.6.8 apibrėžimas.** Metakamieno MK modelį sudaro apibrėžta metakalbos funkcijų aibė  $f_k(a_j)$ :

$$MK = \{f_k(a_j)\}; \quad (4.6.3)$$

čia  $a_j \in P$ , arba  $a_j \in T$ .

*Pastaba:* pilnesnį formalų aprašą galima rasti [ŠBB14].

**4.6.9 apibrėžimas.** Metaprogramos elgsenos modeliu (angl. *behavioural model*) vadinamas procesas metaprogramos aplinkos viduje, apdorojantis specifikaciją taip, kad būtų sukurti programų egzemplioriai.

#### 4.7. IGMO aukšto lygmens specifikacijos transformavimo į vykdomąją specifikaciją taisyklės

**4.7.1 taisyklė.** IGMO požymių modelio variantinis taškas (žr. 3.4.4.11 apibrėžimą) atitinka IGMO vykdomosios specifikacijos (metaprogramos) parametą.

**4.7.2 taisyklė.** IGMO požymių modelio variantai (žr. 3.4.4.10 apibrėžimą) atitinka IGMO vykdomosios specifikacijos (metaprogramos) parametų reikšmes.

**4.7.3 taisyklė.** Paprastas priskyrimas IGMO vykdomosios specifikacijos metasąsajoje užrašomas:

$\langle \text{parametras} \rangle = \langle \text{parametro reikšmių aibė} \rangle$ .

**4.7.4 taisyklė.** Šakotasis priskyrimas IGMO vykdomosios specifikacijos metasąsajoje užrašomas:

$\langle \text{parametras1} \rangle \langle \text{sąlyga} \rangle \langle \text{parametras2} \rangle \langle \text{parametras1} \rangle = \langle \text{parametro reikšmių aibė} \rangle$ .

Šakotasis priskyrimas naudojamas tuomet, kai variantinis taškas turi apribojimus *reikalauja* arba *išskyrus*.

**4.7.5 taisyklė.** IGMO vykdomojoje specifikacijoje (metaprogramoje) turi būti tiek metaparametrų, kiek IGMO požymių modelyje yra variantinių taškų.

**4.7.6 taisyklė.** IGMO vykdomosios specifikacijos metasąsajoje parametrai išdėstomi prioritetų mažėjimo tvarka.

**4.7.7 taisyklė.** IGMO vykdomosios specifikacijos metakamienas formuojamas naudojant metakalbos funkcijų aibę:  $\{\text{priskyrimas } („=“), \text{ OPEN-WRITE-CLOSE, šakojimas, kartojimas}\}$ .

#### 4.8. Išplėstinių generatyvinių mokymosi objektų savybės

**4.8.1 savybė.** Aukšto lygmens (AL) IGMO kūrimas yra mokymosi variantiškumo (MV) modelių atvaizdavimas į heterogeninės metaprogramos (MP) modelį, formaliai užrašoma:

$$IGMO_{AL} = PM_{MV} \times MP_M; \quad (4.8.1)$$

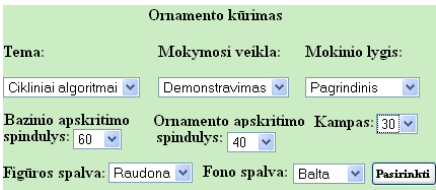
čia  $IGMO_{AL}$  – aukšto lygmens AL išplėstinio generatyvinio mokymosi objekto modelis;  $PM_{MV}$  – mokymosi variantiškumas MV, išreikštas konkrečiu požymių modeliu PM;  $MP_M$  – heterogeninės metaprogramavimo srities modeliai; „ $\times$ “ – atvaizdavimo operatorius.

**4.8.2 savybė.** Metaprogramavimu pagrįsti IGMO yra heterogeninės metaprogramos.

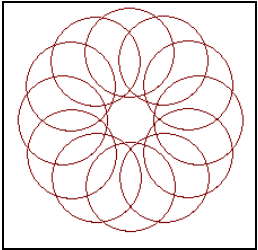
**4.8.3 savybė.** IGMO metasąsaja išreiškia parametrų reikšmių aibę, leidžiančią sukurti MO egzempliorių su pasirinktomis parametrų reikšmėmis.

**4.8.4 savybė.** IGMO metakamieną sudaro iš anksto nustatyta metakalbos funkcijų aibė, kuri įterpiama į tikslo kalba užrašytą MO kodą pagal iš anksto numatytą formatą ir taisykles.

**4.8.5 savybė.** Žiūrint iš mokinio ir mokytojo perspektyvos, IGMO yra „juoda dėžė“ (angl. *black-box*) (4.4 pav. a).



a)



c)

```

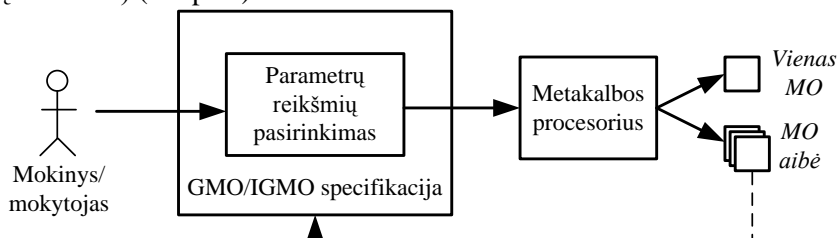
// Tema: Cikliniai_algoritmai
// Mokymosi veikla: Demonstravimas
// Mokinio lygis: Pagrindinis
// Programa Ornamentas
#include <graphics.h>
#include <cmath>
int main () {
    initwindow(600, 800);
    setcolor(WHITE);
    rectangle (0, 0, 600, 800);
    setfillstyle(SOLID_FILL, WHITE);
    floodfill (400, 50, WHITE);
    int m = 0;
    setcolor(RED);
    while (m < 360)
    {
        circle (floor (60 * cos (m * 3.14 / 180))
        + 300, floor(60 * sin (m * 3.14 / 180))
        + 300, 40);
        m = m + 30;
    }
    while(!kbhit());
    closegraph();
    return 0;
}
        
```

b)

**4.4 pav.** IGMO „Nežinomo kartojimų skaičiaus ciklas“ vykdomoji specifikacija: a) sugeneruota metasąsaja; b) su pasirinktomis parametrų reikšmėmis (4.4 pav. a) sugeneruotas MO egzempliorius; c) MO darbo rezultatas

**4.8.6 savybė.** Struktūriniu požiūriu IGMO yra aukšto lygmens specifikacija, aprašanti šeimyną susijusių MO egzempliorių.

**4.8.7 savybė.** Elgsenos požiūriu IGMO yra generatorius, generuojantis MO egzempliorius pagal vartotojo reikalavimus (vartotojui pasirinkus reikiamas parametrų reikšmes) (4.5 pav.).



**4.5 pav.** GMO/IGMO elgsenos modelis

Galima daryti išvadą, kad praktiniu požiūriu IGMO aibė sudaro MO biblioteką (angl. *LO library*) arba jos dalį, kurioje susijusių MO aibė saugoma kompaktiškai supakuota.

**4.8.8 savybė.** Technologijos taikymo programavimo mokymesi požiūriu IGMO yra traktuojamas kaip metaprograma. MO, t.y. mokymosi turinys, yra tikslo kalba parašyta programa.

#### 4.9. IGMO technologinio sudėtingumo įvertinimo metrikos

IGMO technologinio sudėtingumo matavimai naudingi IGMO struktūros motyvavimui, sąryšių tarp skirtingų IGMO dalių supratimui, IGMO įvertinimui ir palyginimui. Išskiriamos: 1) pirmos eilės metrikos, gaunamos tiesiogiai iš IGMO aprašymo naudojant paprastus matematinius veiksmus (pvz., programos dydis, simbolių skaičius faile) ir 2) antros eilės metrikos, kurios tiesiogiai negali būti gautos ir yra skaičiuojamos iš pirmos eilės metrikų [ŠD13].

Metaprogramų sudėtingumas gali būti įvertinamas skirtingais aspektais [DŠ10]: 1) informacinis aspektas – metaprograma kaip pranešimas (simbolių seka), kuriame yra informacija su nežinoma sintakse ir struktūra; 2) metakalbos aspektas – metaprograma yra paaiškintas srities pažinimas. Srities pažinimas išreiškiamas naudojant srities kalbą, kur srities variantiškumas specifikuojamas naudojant metakalbą; 3) grafo aspektas – metaprograma yra grafas, kurio šaknis yra metaprograma, mazgai – metakalbos konstrukcijos, o lapai – srities programos egzemplioriai; 4) algoritminis aspektas – metaprograma yra aukšto lygmens programos specifikacija (algoritmas), turinti funkcinį (struktūrinių) operacijų rinkinį. Operacija gali turėti vieną ar daugiau operandų, kurie specifikuojami kaip metaprogramos atributai (parametrai); 5) pažinimo aspektas – metaprograma yra informacijos vienetų rinkinys (makro, šablonas, funkcija, metaparametras). Remiantis išvardintais aspektais, darbe taikomos Štuikio ir Damaševičiaus [ŠD13] pasiūlytos metaprogramų sudėtingumo metrikos.

*Kolmogorovo sudėtingumo kriterijus* nurodo, kiek resursų reikia norint tiksliai apibrėžti objektą. Kriterijus apibrėžia objektą generuojančios mažiausios programos ilgį (arba koks mažiausias informacijos kiekis reikalingas norint sugeneruoti objektą naudojant tam tikrą algoritmą). Metaprogramos  $M$  sudėtingumas skaičiuojamas naudojant santykinį Kolmogorovo sudėtingumą (*RKC*, *Relative Kolmogorov Complexity*):

$$RKC = \frac{\|C(M)\|}{\|M\|}; \quad (4.9.1)$$

čia  $\|M\|$  – metaprogramos  $M$  apimtis,  $\|C(M)\|$  – suspaustos metaprogramos  $M$  apimtis.

Didelis *RKC* rodo didelį turinio teksto variantiškumą, mažas *RKC* reiškia, kad yra didelis pertekliškas, t.y. metaprogramos kode daug pasikartojančių fragmentų.

Metaprograma  $M$  apibrėžiama kaip srities kalbos sakinių rinkinys su atitinkamomis anotacijomis (metaduomenimis). *Metakalbos turtingumo* (*Metalanguage Richness*, *MR*) metrika skaičiuojama:

$$MR = \frac{\sum_{m \in M} \|m\|}{\|M\|}; \quad (4.9.2)$$

čia  $\|M\|$  – metaprogramos  $M$  apimtis,  $\|m\|$  - metakalbos konstruktų apimtis metaprogramoje  $M$ . Didesnė  $MR$  vertė parodo, kad metaprogramoje yra daugiau metaduomenų ir jų aprašai yra sudėtingesni.

*Ciklomatinis sudėtingumas (CC, Cyclomatic Complexity)* yra lygus programų egzempliorių, kuriuos gali sugeneruoti metaprograma, skaičiui. Didelė ciklomatinio sudėtingumo reikšmė rodo, kad metaprograma turi sudėtingesni parametrų rinkinį.

*Normalizuotas sudėtingumas (ND, Normalized Difficulty)* skaičiuojamas:

$$ND = \frac{n_1 N_2}{(N_1 + N_2)(n_1 + n_2)}; \quad (4.9.3)$$

čia  $n_1$  – metaprogramos skirtingų operatorių skaičius,  $n_2$  – metaprogramos skirtingų operandų skaičius,  $N_1$  – bendras metaprogramos operatorių skaičius,  $N_2$  – bendras metaprogramos operandų skaičius. Didelis  $ND$  rodo, kad programą yra sudėtinga suprasti.

*Pažinimo sudėtingumas (CD, Cognitive Difficulty)* yra skaičiuojamas kaip maksimalus metaprogramos metalygmenų vienetų skaičius (metaparametrai  $P$ , metakalbos konstrukcijos  $N_1$ , jų atitinkami argumentai  $N_2$ ):

$$CD = \max(P, N_1, N_2). \quad (4.9.4)$$

Metaprogramos suprantamumas ir sudėtingumas yra susiję: didėjant sudėtingumui, suprantamumas turi tendenciją mažėti.

IGMO „Nežinomo kartojimų skaičiaus ciklas“ vykdomosios specifikacijos sudėtingumo įvertinimas pateiktas 4.1 lentelėje.

**4.1 lentelė** Vienos pakopos IGMO „Nežinomo kartojimų skaičiaus ciklas“ technologinio sudėtingumo metrikos

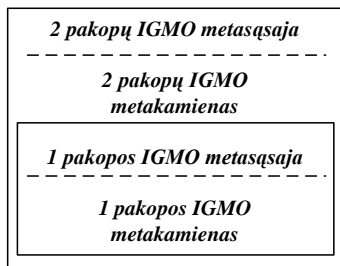
Metrika	Reikšmė
Santykinis Kolmogorovo sudėtingumas*	0.14
Metakalbos turtingumas	0.41
Ciklomatinis sudėtingumas	1920
Normalizuotas sudėtingumas	0.25
Pažinimo sudėtingumas	95

\* Metaprogramos suspaudimui taikytas BWT (*Burrows-Wheeler Transform, block-sorting compression*) algoritmas.

#### 4.10. IGMO sudėtingumo valdymas vykdomosios specifikacijos lygmenyje

Heterogeninių metaprogramų sudėtingumą ir suprantamumą valdyti siūloma įvedant apgražos inžinerija grindžiamą daugiapakopį metaprogramavimą [ŠD13]. Ši idėja sėkmingai pritaikyta IGMO sudėtingumo valdymui vykdomosios specifikacijos lygmenyje.

Dvių pakopų IGMO modelis pateiktas 4.6 pav.



**2 pakopų IGMO metasąsaja:** parametrai specifikuoti naudojant PHP (metakalba) ir HTML (atvaizdavimo internete kalba):

// Konteksto parametrai

- **Tema:** Cikliniai algoritmai.
- **Mokymo(-si) veikla:** Demonstravimas; Praktika; Atvejo analizė.
- **Mokinio lygis:** Patenkinamas; Pagrindinis; Aukštesnysis.
- //Turinio parametrai
- **Bazinio apskritimo spindulys:** 20; 40; 60; 80; 100.

**2 pakopų IGMO metakamienas:** PHP (metakalba) ir C++ (tikslo kalba):

**1 pakopos IGMO metasąsaja:** parametrai specifikuoti naudojant PHP (metakalba) ir HTML (atvaizdavimo internete kalba):

//Turinio parametrai

- **Ornamanto apskritimo spindulys:** 20; 40; 60; 80; 100.
- **Kampas:** 10; 30; 60; 90.
- **Figūros spalva:** Raudona; Žalia; Mėlyna.
- **Fono spalva:** Balta; Pilka; Mėlyna.

**1 pakopos IGMO metakamienas:** PHP (metakalba) ir C++ (tikslo kalba):

- Pradinių duomenų aprašymas.
- Pasirinktų parametrų ornamanto piešimo aprašymas.

a)

b)

**4.6 pav.:** Dviejų pakopų IGMO modelis: a) – apibendrinta IGMO modelio struktūra, b) IGMO „Nežinomo kartojimų skaičiaus ciklas“ modelis

**4.10.3 apibrėžimas.** Metafunkcija vadinama pasyvia (angl. *passive*), jei dabartinėje IGMO vykdymo pakopoje ji traktuojama kaip tikslo kalbos tekstas. Pasyvi metafunkcija turi deaktivacijos žymą „\“.

**4.10.4 apibrėžimas.** Deaktyvacija vadinamas procesas, kai metafunkcijos pažymimos deaktivacijos žyma „\“. Aktyvacijos procese deaktivacijos žymos pašalinamos vykdant IGMO antrąjį kartą.

Pagrindinės dviejų pakopų IGMO savybės papildo vienos pakopos IGMO savybių rinkinį.

**4.10.1 savybė.** Struktūrinis dviejų pakopų IGMO modelis išgaunamas iš vienos pakopos modelio per jo restruktūrizaciją. Vienos pakopos IGMO metasąsaja suskaidoma į du lygmenis: aukštesnis lygmuo įvertinamas antrojoje pakopoje, žemesnis – pirmojoje.

**4.10.2 savybė.** Žemesnio lygmens metasąsajos dalis yra sudedamoji dviejų pakopų IGMO metakamieno dalis.

**4.10.3 savybė.** Dviejų pakopų IGMO interpretavimo (įvykdymo) rezultatas yra vienos pakopos IGMO aibė, apibrėžta aukštesnio lygio metaparametrų reikšmėmis.

**4.10.4 savybė.** Dviejų pakopų IGMO interpretavimo procesą taip pat sudaro dvi pakopos. Vykdam aukštesnės pakopos IGMO sugeneruojama vienos pakopos IGMO aibė (4.10.3 savybė). Vykdam žemesnės pakopos IGMO sugeneruojamas MO ar MO aibė.

**4.10.5 savybė.** Metakamiene esančios metafunkcijos valdo metakamieno funkcionalumą. Metafunkcijos yra dviejų tipų: aktyvios (4.10.2 apibrėžimas) ir pasyvios (4.10.3 apibrėžimas).

**4.10.6 savybė.** Dviejų pakopų IGMO metakamieną sudaro abiejų tipų funkcijos, kai tuo tarpu vienos pakopos IGMO kamieną sudaro tik aktyvios funkcijos.

**4.10.7 savybė.** Jeigu visi  $n$  metaparametrų yra ortogonalūs, bet kuris  $(n-1)$  metaparametrų derinys gali būti perkeltas į antrą pakopą. Jei parametrai yra priklausomi, jie būtinai turi būti toje pačioje pakopoje (1 arba 2).

**4.10.8 savybė.** Realizuojant dviejų pakopų IGMO svarbiausias yra metafunkcijų aktyvacijos ir deaktyvacijos procesas (4.10.2-4.10.4 apibrėžimai).

Dviejų pakopų IGMO „Nežinomo kartojimų skaičiaus ciklas“ vykdomosios specifikacijos sudėtingumo įvertinimas pateiktas 4.2 lentelėje.

**4.2 lentelė** Dviejų pakopų IGMO „Nežinomo kartojimų skaičiaus ciklas“ technologinio sudėtingumo metrikos

Metrika	2 pakopų IGMO	1 pakopos IGMO, sugeneruotas iš 2 pakopų IGMO
Santykinis Kolmogorovo sudėtingumas*	0.18	0.12
Metakalbos turtinumas	0.33	0.43
Ciklomatinis sudėtingumas	12	160
Normalizuotas sudėtingumas	0.17	0.23
Pažinimo sudėtingumas	179	57

#### 4.11. Išvados

1. IGMO papildo GMO principus (socialiniai ir pedagoginiai aspektai yra aukštesnio prioriteto, negu turinio aspektai; tiesioginė koncepcijų atskirtis aprašo su GMO susijusius požyūrius ir procesus; kuriant GMO taikoma daugiapakopė koncepcijų atskirtis) naujomis prielaidomis: 1) e.mokymosi srities posritys konceptualiame lygmenyje gali būti išreikštos požymiais; 2) tiesiogiai apibrėžti požymiai susiejami su generatyvine technologija per transformavimo taisykles.

2. Informatikos (programavimo) mokymuisi skirti IGMO aukštame abstrakcijos lygmenyje specifikuoti požymių modeliais, sudarytais iš konteksto ir turinio modelių, kuriuos semantiškai susieja prioritetų modelis ir apribojimai bei sąryšiai tarp konteksto ir turinio modelių požymių, įgalinantys valdyti modelio sudėtingumą ir sukuriantys realias prielaidas adaptavimui modelio lygmenyje.

3. IGMO yra aukšto lygmens modeliais grįsta vykdomoji specifikacija, įgalinanti automatizuotai kurti MO egzempliorius.

4. Suformuluotos IGMO aukšto lygmens specifikacijos transformavimo į vykdomąją specifikaciją taisyklės įgalina sukurti vykdomąsias specifikacijas ir sukuria prielaidas automatiniam įrankiui kurti.

5. Suformuluotos tolesnių tyrimų gairės, susijusios su požymių prioritetais žeminant abstrakcijos lygmenį ir tolesnio IGMO kūrimo automatizavimo galimybėmis.

## **5. IŠPLĖSTINIŲ GENERATYVINIŲ MOKYMOSI OBJEKTŲ INTEGRAVIMAS Į SPECIALIZUOTAS MOKYMOSI APLINKAS**

### **5.1. Įvadas**

Sukurti išplėstiniai generatyviniai mokymosi objektai (IGMO) ugdyme gali būti naudojami įvairiais tikslais. Norint efektyviai panaudoti sukurtų IGMO galimybes, būtina juos integruoti į mokymosi aplinkas.

Šio skyriaus tikslas – sukurti specializuotas integruotas mokymosi aplinkas programavimo mokymuisi į jas integruojant sukurtus IGMO. Formuluojami tokie skyriaus uždaviniai:

- apibrėžti reikalavimus specializuotai mokymosi aplinkai;
- remiantis reikalavimais mokymosi aplinkoms ir IGMO savybėmis sukurti heterogeninės specializuotos mokymosi aplinkos architektūrą;
- sukurti bendradarbiaujančių IGMO aplinkos architektūrą;
- apibrėžti sukurtų mokymosi aplinkų kokybės įvertinimo kriterijus.

5.2 skyrelyje apibrėžti reikalavimai mokymosi aplinkoms; 5.3 skyrelyje aprašyta heterogeninės specializuotos mokymosi aplinkos architektūra; 5.4 skyrelyje pateikta bendradarbiaujančių IGMO aplinkos architektūra; 5.5 skyrelyje apibrėžti sukurtų mokymosi aplinkų kokybės įvertinimo kriterijai; 5.6 skyrelyje suformuluotos skyriaus išvados.

### **5.2. Reikalavimai mokymosi aplinkoms**

Mokymosi aplinka padeda siekti apibrėžtų mokymosi tikslų naudojant specifinį mokymosi turinį [DSV04] ir apima mokymosi išteklius (mokymosi turinį ir įrankius), sąveiką ir bendradarbiavimą tarp mokymosi proceso dalyvių, mokymosi veiklas ir mokymosi palaikymo veiklas [Lom08].

Bendrieji reikalavimai mokymosi aplinkoms su integruotais IGMO suformuluoti susiejus bendruosius [Lom08] ir programavimo mokymuisi skirtų aplinkų reikalavimus, aprašytus 2.9.5 skyrelyje, pateikiami 5.1 lentelėje.

5.1 pav. pateikiamas informatikos mokymosi konceptualusis modelis, kurio centre yra išplėstiniai generatyviniai mokymosi objektai ir jų sąveika su pedagoginėmis veiklomis, technologiniais procesais, žinių perdavimo kanalais, naudojamais įrankiais ir pedagoginiais rezultatais.

Pedagoginės veiklos yra glaudžiai susijusios su mokymosi tikslais, turiniu, mokymosi modelio, įrankių pasirinkimu, užduoties formulavimu, pedagoginių išėjimų įvertinimu. Laikant, kad mokymosi turinys yra IGMO, būtina apibrėžti vieną iš svarbiausių pedagoginių veiklų – reikalavimus IGMO ir IGMO kūrimą (žr. 4 skyrių).

Technologiniai procesai prasideda uždavinio pasirinkimu. Uždavinio pasirinkimo procesas leidžia sukurti IGMO, tačiau yra susijęs su apribojimais, atsirandančiais dėl aplinkoje naudojamų įrankių, programavimo kalbos ir algoritmų, dengiančių kurso turinį, pasirinkimo. Sukūrus, ar iš bibliotekos pasirinkus IGMO, vykdomi parametrų pasirinkimo ir turinio generavimo procesai. Sugeneruota programa kompiliuojama, paleidžiama vykdyti ir atliekama uždavinio sprendimo kontrolė.

Žinių perdavimo kanalai susieja pedagogines veiklas ir technologinius procesus.

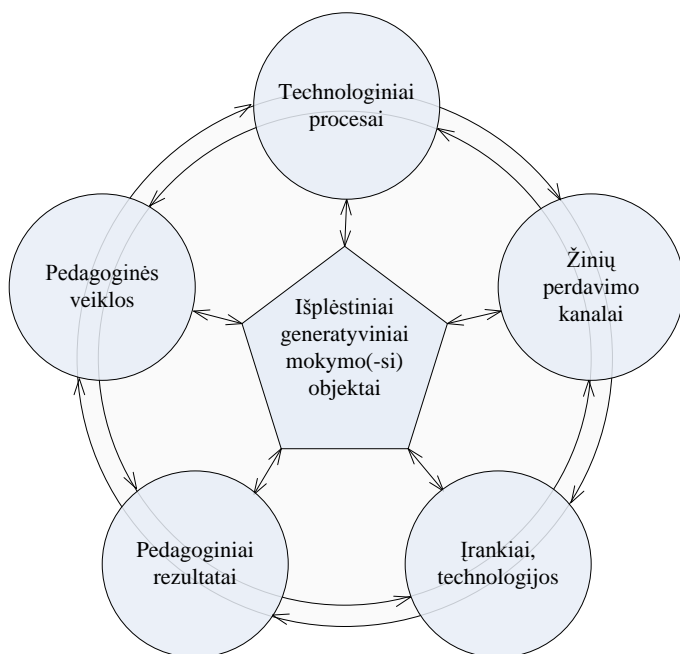
Grįžtamieji ryšiai tarp komponentų užtikrina turinio pakartotinio generavimo ir modifikavimo, žinių išgavimo per mokymosi scenarijus lankstumą.

### 5.1 lentelė Reikalavimai specializuotoms mokymosi aplinkoms su IGMO\*

Reikalavimai	Potencialūs komponentai	Pageidaujamos savybės
Mokymosi išteklių	<ul style="list-style-type: none"> <li>• Įvairialypis mokymosi turinys – MO, knygos, žinyrai ir t.t. [Lom08] + <b>GMO, IGMO</b>.</li> <li>• Įrankiai – techninė ir programinė įranga (žr. 3.10 pav.) [KP05], [SFP+12], [TMH+12] + <b>mokomieji robotai, mikrovaldikliai</b>.</li> </ul>	<ul style="list-style-type: none"> <li>• Lengvai pasiekiami.</li> <li>• Nuolat atnaujinami.</li> <li>• Ekspertų pasiūlyti ir besimokančiųjų sukurti.</li> </ul>
Sąveika ir bendradarbiavimas	<ul style="list-style-type: none"> <li>• Bendradarbiavimą užtikrinantys mokymosi išteklių: mokymosi objektai ir techninė įranga [Lom08].</li> </ul>	<ul style="list-style-type: none"> <li>• Palankios naudojimosi sąlygos.</li> <li>• Apibrėžti vaidmenys ir atsakomybė.</li> <li>• Grįžtamasis ryšys.</li> </ul>
Mokymosi veiklos	<ul style="list-style-type: none"> <li>• Formalios mokymosi veiklos mokymosi tikslui pasiekti [Lom08].</li> </ul>	<ul style="list-style-type: none"> <li>• Taikomi patikrinti ir praktikoje pasitvirtinę mokymosi principai ir modeliai [Lom08].</li> <li>• Galimybės pasirinkti veiklą pagal poreikius užtikrina mokymosi individualizavimą [Lom08].</li> <li>• Programavimo modelių naudojimas (uždavinio skaidymas į smulkesnes dalis, žinių pritaikymas analogiškoje situacijoje) ugdo problemų sprendimo gebėjimus [GM07].</li> </ul>
Mokymosi palaikymo veiklos	<ul style="list-style-type: none"> <li>• Nuolatinis grįžtamasis ryšys [Lom08].</li> <li>• Veiklos, darančios įtaką mokymuisi: nuolatinis turinio ir įrankių atnaujinimas [Lom08].</li> </ul>	<ul style="list-style-type: none"> <li>• Apibrėžtas mokytojo vaidmuo užtikrinant grįžtamąjį ryšį ir lanksčiai organizuojant mokymosi procesą per mokymosi tikslų, metodų, veiklų tikslinimą, keitimą ir papildymą [Lom08].</li> <li>• Kiekvieno besimokančiojo žinių lygio nustatymas ir nuolatinis atnaujinimas bei dominuojančio mokymosi stiliaus nustatymas suteikia galimybę mokymąsi padaryti efektyvesnį ir labiau individualizuotą [GM07].</li> <li>• Algoritmų kūrimo įrankių naudojimas aplinkoje suteikia galimybes besimokančiajam įgyti algoritmų sudarymo bei realizavimo žinių ir įgūdžių [GM07].</li> </ul>

\* Paryškintu kursyvu įrašyti reikalavimai, atsirandantys į aplinką integruojant IGMO





5.1 pav. Informatikos mokymosi konceptualusis modelis

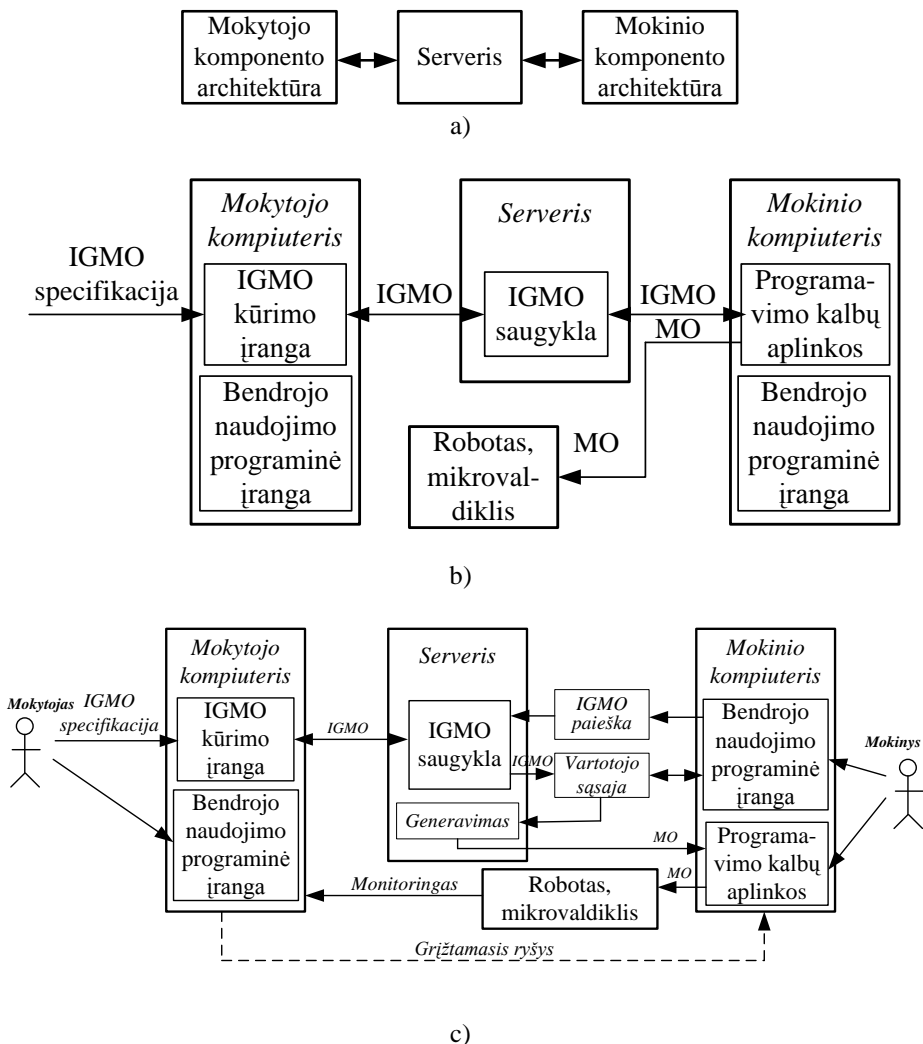
### 5.3. Specializuotos heterogeninės mokymosi aplinkos architektūra ir funkcionalumas

Specializuotą heterogeninę programavimo mokymosi aplinką sudaro trys tarpusavyje susijusios dalys: mokytojo ir mokinio komponentai bei serveris (5.2 a) pav.). Mokytojo komponentą sudaro mokytojo kompiuteris su jame įdiegta IGMO kūrimo ir bendrojo naudojimo programine įranga, kuri naudojama bendravimui su serveriu (užklausoms, sukurtų IGMO perkėlimui į/iš serverį). Sukurti IGMO perkeliama į serveryje esančią IGMO saugyklą. Mokinio kompiuteryje įdiegta bendrojo naudojimo programinė įranga, kuri leidžia iš serverio saugyklos pasirinkti IGMO ir sugeneruoti MO pagal vartotojo poreikius, programavimo kalbų aplinkos, kurios sukuria vykdomąją MO specifikaciją, kuri perkeliama į robotą, arba mikrovaldiklį (5.2 b) pav.)

5.3 c pav. pateiktas aplinkos elgsenos modelis. Pirmiausia sukuriama IGMO specifikacija ir naudojant IGMO kūrimo įrangą sukurtas IGMO perkeliama į serveryje esančią IGMO saugyklą. IGMO kūrėjas bet kuriuo momentu gali modifikuoti saugykloje esantį IGMO.

IGMO naudotojas naudodamas bendrojo naudojimo programinę įrangą saugykloje suranda IGMO, suformuotoje vartotojo sąsajoje pasirenka reikiamas parametrų vertes ir patvirtinus pasirinkimą sugeneruojamas MO (roboto valdymo programos kodas tikslo kalba), kurį naudotojas įkelia į programavimo kalbos aplinką ir sukuria vykdomąją specifikaciją, kuri perkeliama į robotą.

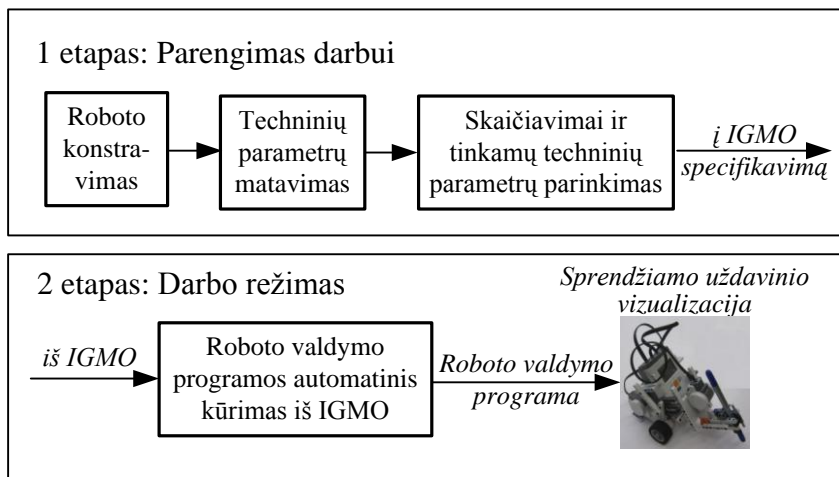
Mokytojas atlieka monitoringą ir užtikrina lankstų grįžtamąjį ryšį.



**5.2 pav.** Heterogeninė specializuota mokymosi aplinka: a) – bendroji struktūra, b) – aplinkos komponentų struktūra, c) aplinkos elgsenos modelis

Kuriant roboto aplinką (5.3 pav.) išskirti du etapai: 1) parengimas darbui; 2) darbo režimas.

Pirmajame etape iš LEGO NXT sudedamųjų dalių [BSM12] konstruojamas robotas, kuris atliks ugdymo plane numatytą užduotį. Roboto konstravimui buvo naudojama nemokama programinė įranga Lego Digital Designer (<http://ldd.lego.com/>), kuri leidžia sukurti roboto konstravimo žinyną, sudarytą iš HTML failų. Pagrindinis tokio konstravimo būdo privalumas – pakartotinai konstruojant robotą, bus optimizuota konstravimo veiksmų seka, trūkumas – nėra galimybės virtualiai išbandyti sukurtos konstrukcijos.



**5.3 pav.** Roboto aplinkos konstravimo etapai

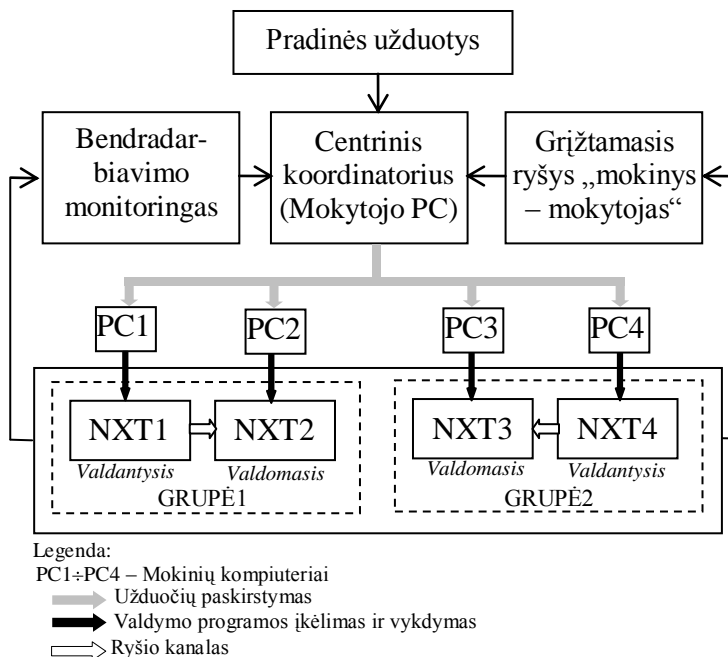
Sukonstravus robotą prieš pradėdant kurti roboto valdymo programas reikia nustatyti leistinas parametrų ribas. Dėl minėtos priežasties būtina išmatuoti roboto techninius parametrus: nuvažiuoto atstumo priklausomybę nuo roboto greičio ir judėjimo laiko, jutiklių jautrį (angl. *sensitivity*), tikslumą (angl. *accuracy*), matuojamo dydžio kitimo ribas (angl. *stimulus range*), atsako greitį (angl. *speed of response*).

Antrajame aplinkos kūrimo etape iš IGMO automatiškai kuriamos roboto valdymo programos, kurios perkeliama į robotą ir realizuojama sprendžiamo uždavinio vizualizacija.

#### 5.4. Bendradarbiaujančių robotų aplinka

Robotais pagrįstos mokymosi aplinkos, kurioje integruoti bendradarbiaujantys IGMO, architektūra yra paremta klasikiniu pavaldumo (angl. *master-slave*) modeliu su papildomais komponentais, reikalingais roboto orientavimuisi aplinkoje (jutikliai, belaidės kameros), ryšio kanalais, užtikrinančiais bendravimą tarp robotų siunčiant žinutes ir palaikančiais skirtingus bendravimo protokolus (Bluetooth, WiFi) bei valdymo technine ir programine įranga, įdiegta kompiuteryje (5.4 pav.).

Pavaldumo modelyje lygiagrečių procesų vykdymas gali būti suprantamas kaip lygiagrečių ir nuoseklių subprocesų seka. Subprocesai yra valdomi komunikuojant valdančiajam (angl. *master*) ir valdomajam (angl. *slave*) robotams, kai transliacijos mazgas yra valdantysis robotas, arba siunčiant/gaunant pranešimus (valdantysis robotas siunčia, valdomasis gauna) [YS00]. Principas yra analogiškas uždavinio skaidymo dalimis principui. 5.4 pav. pateikiama keturių pakopų struktūra, taikoma kuriant bendradarbiaujančiais robotais pagrįstą aplinką:



**5.4 pav.** Mokymosi aplinka su bendradarbiaujančiais robotais [BSD13]

1) Patariamasis (angl. *Deliberative*) sluoksnis: Centrinis koordinatorius gauna pradinės užduotis robotams iš mokytojo, po to jas išskaido į smulkesnes dalis ir perkelia sugeneruotas roboto valdymo programas į mokinių kompiuterius. Paprasčiausiu atveju kiekviena užduotis skaidoma į dvi dalis ir dvi nepriklausomos mokinių grupės atlieka kiekvieną užduoties dalį.

2) Fizinis (angl. *Physical*) sluoksnis: realūs mokomieji robotai, kurių judėjimą užtikrina servo motorai.

3) Reaguojantis (angl. *Reactive*) sluoksnis: jautikliai leidžia gauti robotui informaciją apie aplinką ir reaguoti į aplinkos pokyčius.

4) Komunikacijos (angl. *Communication*) sluoksnis: keitimasis pranešimais tarp robotų ir grįžtamojo ryšio numatymas monitoringui ir įvertinimui.

Realioje aplinkoje bendradarbiaujančių robotų grupių skaičius priklauso nuo techninių galimybių (robotų ir kompiuterių skaičiaus) ir mokymosi poreikių (mokinių skaičiaus, mokymosi tikslų ir t.t.). Norint užtikrinti mokymosi poreikių patenkinimą ir techninį patikimumą numatytas grįžtamasis ryšys „mokinys-mokytojas“ realiu laiku ir bendradarbiaujančių robotų elgsenos stebėseną.

### 5.5. Mokymosi aplinkų kokybės vertinimo kriterijai

Mokymosi aplinkų kokybės vertinimo kriterijai, suformuoti adaptavus Kurilovo ir Dagienės [KD09] (technologiniai kokybės vertinimo kriterijai) ir De Kock *ir kt.* [DSV04] (pedagoginiai kokybės vertinimo kriterijai) siūlomą metodiką, pateikiami 5.2 lentelėje.

## 5.2 lentelė Mokymosi aplinkų vertinimo kriterijai

<b>Technologiniai kokybės vertinimo kriterijai</b>	
Bendra architektūra	Lankstumas Moduliškumas Veiklos optimizavimas Patikimumas ir stabilumas Pakartotinis panaudojimas ir perkeliamumas
Internacionalizavimas ir lokalizavimas	Lokaliizuota vartotojo sąsaja Alternatyvių kalbų palaikymas
Adaptavimas	Kalbos ir kūrimo adaptavimas pagal konkrečios mokyklos poreikius Individualizavimas pagal mokinio poreikius Automatiškai adaptuojamas mokymosi turinys
<b>Pedagoginiai kokybės vertinimo kriterijai</b>	
Mokymosi tikslai	<i>Mokymosi produktai:</i> mokymosi turinio pažinimas, mokymosi proceso pažinimas, pažintiniai mokymosi įgūdžiai, emociniai mokymosi įgūdžiai, socialiniai mokymosi įgūdžiai, pritaikymo įgūdžiai. <i>Mokymosi procesas:</i> paruošiamosios, vykdomosios ir baigiamosios mokymosi funkcijos (gali būti: pažinimo, emocinės, metapažinimo)
Mokymosi teorija	Biheavioristinė, kognityvinė, konstruktyvistinė
Mokinių vaidmenys	Individualus mokymasis, bendradarbiavimas, lenktyniavimas

Technologiniai kokybės vertinimo kriterijai išreiškiami ranginiais kintamaisiais [KD09]: 0 – nėra; 1 – silpnai išreikštas; 2 – vidutiniškai išreikštas; 3 – gerai išreikštas; 4 – puikiai išreikštas. Visuminiam technologiniam aplinkos įvertinimui užrašoma *sudėtinė naudingumo funkcija* (angl. *additive utility function*) [KD09]:

$$f(X) = \sum_{i=1}^m f_i(X); \quad (5.6.1)$$

čia  $f_i(X) = \{0,1,2,3,4\}$  yra  $i$ -tojo kriterijaus reikšmė.

Laikoma, kad visi išvardinti technologiniai kriterijai yra vienodos svarbos. Kuo didesnė sudėtinės naudingumo funkcijos reikšmė, tuo geresnė mokymosi aplinkos kokybė.

Mokymosi aplinkos pedagoginiai kriterijai apibrėžiami remiantis trimis pagrindiniais aspektais: mokymosi tikslais, mokinio–mokytojo ir mokinio–mokinio tarpusavio sąveika. Remiantis [DSV04] pateikiami pagrindinių kriterijų paaiškinimai. Mokymosi turinio pažinimas suprantamas kaip tinkamų faktų ir principų žinojimas. Mokymosi proceso pažinimas apima mokymosi stilius, jų tinkamumą konkrečiam mokiniui ir situacijai. Pažintiniai mokymosi įgūdžiai parodo, kaip mokinys geba pritaikyti ankstesnes žinias ir įgūdžius. Emociniai mokymosi įgūdžiai apibrėžia motyvaciją suprasti dar neišmokus dalykus. Socialiniai mokymosi įgūdžiai apima tarpasmeninį bendradarbiavimą. Pritaikymo įgūdžiai parodo, koku laipsniu besimokantysis geba pritaikyti įgytas žinias ir įgūdžius naujuose negimininguose kontekstuose. Mokymosi funkcijos yra glaudžiai susijusios su mokymosi procesu ir jo eigoje gali būti optimizuojamos ir tikslinamos. Mokymosi funkcijų vykdymas apima integruotą specifinių žinių ir įgūdžių pritaikymą. Mokymosi funkcijų tipai pateikiami 5.3 lentelėje. Kriterijus „Mokymosi

teorijos“ apibrėžia mokinio ir mokytojo vaidmenį, „Mokinių vaidmenys“ – mokinių tarpusavio sąveiką mokymosi procese.

### 5.3 lentelė Mokymosi funkcijų tipai

Tipai Funkcijos	Pažinimo	Emocinės	Metapažinimo
Paruošiamosios	Suprasti, kas nesuprasta iš anksčiau.	Suformuluoti iššūkius.	Orientacija į mokymosi tikslus.
Vykdomosios	Praktikuotis ir taikyti.	Palaikyti motyvaciją.	Apibrėžti nesėkmių priežastis.
Baigiamosios	Igytų žinių ir įgūdžių taikymas naujuose negimininguose kontekstuose.	Darbo rezultatai.	Mokymosi procesų ir rezultatų įvertinimas.

Visi apibrėžti kriterijai yra vienodai svarbūs atliekant mokymosi sistemos pedagoginį vertinimą. Sumodeliuotų heterogeninės specializuotos ir bendradarbiaujančių robotų aplinkų kokybės įvertinimai pateikti 5.4 ir 5.5 lentelėse.

### 5.4 lentelė Sumodeliuotų heterogeninės specializuotos ir bendradarbiaujančių robotų aplinkų kokybės įvertinimas technologiniu požiūriu\*

Kokybės vertinimo kriterijai	Aplinka	Vieno roboto	Bendradarbiaujančių robotų
Lankstumas		3	4
Moduliškumas		3	3
Veiklos optimizavimas		3	3
Patikimumas ir stabilumas		3	2
Pakartotinis panaudojimas ir perkeliamumas		3	3
Lokaluota vartotojo sąsaja		4	4
Alternatyvių kalbų palaikymas		4	4
Kalbos ir kūrimo adaptavimas pagal konkrečios mokyklos poreikius		4	4
Individualizavimas pagal mokinio poreikius		3	3
Automatiškai adaptuojamas mokymosi turinys		4	4
<b>Technologinių kriterijų sudėtinė naudingumo funkcija</b>		<b>34</b>	<b>34</b>

\* Kiekvienas kriterijus vertinamas ranginiais kintamaisiais 0÷4 skalėje

Vertinant mokymosi aplinką pedagoginiu požiūriu, mokymosi produktai vertinami ranginiais kintamaisiais 0÷4 skalėje, dominuojantys mokymosi proceso funkcijų tipai, mokymosi teorijos ir mokinių vaidmenys žymimi atitinkamais simboliais: P – pažinimo, E – emocinės, M – metapažinimo, B – biheivoristinė, Kg – kognityvinė, Kn – konstruktyvistinė, I – individualus mokymasis, Bn – bendradarbiavimas, L – lenktyniavimas. Skliausteliuose pateikti žymėjimai reiškia, kad sistema tenkina kriterijų, tačiau praktikoje naudojamas retai (pvz., (I) – mokiniai gali pasirinkti individualų mokymąsi, bet praktiškai individualus mokymasis taikomas retai).

**5.5 lentelė** Sumodeliuotų mokymosi aplinkų kokybės vertinimas pedagoginiu požiūriu

	Aplinka	Vieno roboto	Bendradarbiaujančių robotų
Kokybės vertinimo kriterijai			
Mokymosi turinio pažinimas		3	3
Mokymosi proceso pažinimas		3	3
Pažintiniai mokymosi įgūdžiai		4	4
Emociniai mokymosi įgūdžiai		4	4
Socialiniai mokymosi įgūdžiai		4	4
Pritaikymo įgūdžiai		3	3
<b>Pedagoginių kriterijų (mokymosi produktų) sudėtinė naudingumo funkcija</b>		<b>21</b>	<b>21</b>
Paruošiamosios mokymosi funkcijos		P E M	P E M
Vykdomosios mokymosi funkcijos		P E M	P E M
Baigiamosios mokymosi funkcijos		P E M	P E M
Mokymosi teorija		Kn	Kn
Mokinių vaidmenys		Bn L (I)	Bn L

Realių mokymosi aplinkų vertinimai pateikti 6.5.3 skyrelyje.

**5.6. Išvados**

1. Apibrėžti reikalavimai specializuotoms heterogeninėms mokymosi aplinkoms sudaro sąlygas pasirinkti kuriamoms aplinkoms tinkamus mokymosi išteklius, užtikrinti efektyvią sąveiką ir bendradarbiavimą, mokymosi ir mokymosi palaikymo veiklas.

2. Sukurta heterogeninės specializuotos mokymosi aplinkos architektūra sudaro prielaidas efektyviam IGMO panaudojimui ugdymo procese.

3. Sukurta specializuota mokymosi aplinka, pagrįsta bendradarbiaujančių robotų panaudojimu, kuri remiasi konstruktyvistine mokymosi teorija, įgalina spręsti sudėtingesnius programavimo uždavinius, ugdo mokinių bendradarbiavimo įgūdžius ir sukuria bendradarbiaujančių IGMO kūrimo prielaidas.

4. Atliktas išankstinis sumodeliuotų heterogeninės specializuotos vieno roboto ir bendradarbiaujančių robotų aplinkų įvertinimas pritaikant mokymosi aplinkų kokybės vertinimo technologinius ir pedagoginius vertinimo kriterijus.

5. Specializuota robotais ir mikrovaldikliais pagrįsta mokymosi aplinka palaiko tarpdalykinės integracijos ugdymo procese koncepciją.

6. Suformuluotos tolesnių tyrimų gairės, susijusios su mokymosi aplinkų architektūrų išplėtimu panaudojant didesnę bendradarbiaujančių mokomųjų robotų skaičių ir didinant sprendžiamų uždavinių sudėtingumą.

## **6. EKSPERIMENTINIS ĮVERTINIMAS**

### **6.1. Įvadas**

Šeštojo skyriaus tikslas – įvertinti sukurtus išplėstinių generatyvinių mokymosi objektų (IGMO) modelius ir vykdomąsias specifikacijas technologiniu ir pedagoginiu aspektais, pateikti realių mokymosi aplinkų su integruotais IGMO pavyzdžių ir įvertinti jų tinkamumą.

Formuluojami tokie skyriaus uždaviniai:

- įvertinti sukurtus programavimo mokymosi srities posričių ir IGMO požymių modelius taikant 3.9 skyrelyje pateiktus kriterijus ir metrikas;
- įvertinti IGMO vykdomųjų specifikacijų technologinį sudėtingumą taikant 4.9 skyrelyje pateiktas metrikas;
- pateikti realių mokymosi aplinkų pavyzdžių su integruotais IGMO ir, remiantis 5.6 skyrelyje pateiktais kriterijais, įvertinti jų kokybę technologiniu ir pedagoginiu aspektais;
- įvertinti IGMO pažinimo sudėtingumą pedagoginiu požiūriu;
- atlikti sukurtų IGMO pedagoginį įvertinimą.

6.2 skyrelyje įvertinti programavimo mokymosi srities posričių modeliai; 6.3 skyrelyje atliktas IGMO požymių modelių verifikavimas; 6.4 skyrelyje pateiktas IGMO vykdomųjų specifikacijų technologinis įvertinimas; 6.5 skyrelyje pateikti realių specializuotų mokymosi aplinkų su integruotais IGMO pavyzdžiai bei įvertinta jų kokybė technologiniu ir pedagoginiu aspektais; 6.6 skyrelyje aptarta IGMO pažinimo sudėtingumo įvertinimo problema; 6.7 skyrelyje pateiktas apibendrintas IGMO pedagoginis įvertinimas; 6.8 skyrelyje suformuluotos skyriaus išvados.

### **6.2. Srities požymių modelių įvertinimas**

6.1 lentelėje pateikiamas 3.7. skyrelyje sudarytų informatikos mokymosi srities posričių požymių modelių vertinimas taikant 3.9. skyrelyje apibrėžtus kriterijus ir metrikas.



**6.1 lentelė** Informatikos mokymosi srities posričių požymių modelių kokybės parametrai (parametrai apskaičiuoti naudojant įrankius FAMILIAR ir SPLOT)

Nr.	Parametras	Pedagogika (M – Motyvacijos, MT – Mokymosi Tikslų, MM – Mokymo/Mokymosi, V – Vertinimo, Mok - Mokinio)					Turinio	Technologijos
		M	MT	MM	V	Mok		
1.	Požymių skaičius	14	14	37	17	24	13	20
2.	Neprivalomų požymių skaičius	0	0	0	0	0	0	2
3.	Privalomų požymių skaičius	7	2	10	0	5	3	7
4.	Esminių požymių skaičius	12	5	1	1	6	4	8
5.	Sugrupuotų požymių skaičius	6	11	26	16	18	9	10
6.	OR grupių skaičius	2	4	10	4	4	2	2
7.	XOR grupių skaičius	0	0	0	1	0	1	1
8.	Apribojimų skaičius	9	6	3	3	3	2	4
9.	Apribojimų reprezentatyvumas, %	0.50	0.64	0.11	0.18	0.25	0.23	0.30
10.	Skirtingų kintamųjų skaičius apribojimuose	7	9	4	3	6	3	6
11.	Apribojimų išlygų tankis	1.29	0.67	0.75	1.00	0.50	0.67	0.67
12.	Medžio gylis	3	5	9	6	3	3	4
13.	Konfigūracijų skaičius	3	61	13107 1	95	74803	84	828
14.	Variantiškumo laipsnis, %	1.8311 E-2	3.7231 E-1	9.5367 E-5	7.2479 E-2	4.4586 E-1	1.0254 E0	7.8964 E-2
15.	BDD mazgų skaičius	14	49	103	95	35	16	26
16.	Modelio neprieštaringumas	Nepr.	Nepr.	Nepr.	Nepr.	Nepr.	Nepr.	Nepr.
17.	Pertekliniai požymiai	Nėra	Nėra	Nėra	Nėra	Nėra	Nėra	Nėra

**6.3. IGMO požymių modelių įvertinimas**

6.2 lentelėje pateikti IGMO požymių modelių (žr. 2 priedą) įvertinimai. Sukurti ir verifikuoti požymių modeliai buvo transformuoti į taikomasias specifikacijas.

**6.2 lentelė** IGMO požymių modelių kokybės parametrai (parametrai apskaičiuoti naudojant įrankius FAMILIAR ir SPLOT)

Nr.	IGMO	Roboto kalibravimas	Linijos sekimas	Ornamentų kūrimas	Bėganti eilutė	Šviesos sekimas	Šviesoforas
	Metrika / Parametras						
1.	Požymių skaičius	38	44	51	27	41	44
2.	Neprivalomų požymių skaičius	0	0	0	0	0	0
3.	Privalomų požymių skaičius	11	10	15	7	10	12
4.	Esminių požymių skaičius	15	14	20	8	11	14
5.	Sugrupuotų požymių skaičius	26	33	35	19	30	31
6.	OR grupių skaičius	1	1	2	2	1	2
7.	XOR grupių skaičius	8	8	11	5	7	8
8.	Apribojimų skaičius	18	12	21	12	7	14
9.	Apribojimų reprezentatyvumas, %	0.53	0.57	0.43	0.63	0.24	0.39
10.	Skirtingų kintamųjų skaičius apribojimuose	20	25	22	17	10	17
11.	Apribojimų išlygų tankis	0.90	0.48	0.95	0.71	0.70	0.82
12.	Medžio gylis	3	3	3	3	3	5
13.	Konfigūracijų skaičius	1296	8640	62208	1440	87480	97200
14.	Variantiškumo laipsnis, %	4.7148 E-7	4.9113 E-8	2.7626 E-9	1.0729 E-3	3.9781 E-6	5.5252 E-7
15.	BDD mazgų skaičius	75	71	73	41	60	66
16.	Neprieštaringumas	Nepriešt.	Nepriešt.	Nepriešt.	Nepriešt.	Nepriešt.	Nepriešt.
17.	Pertekliniai požymiai	Nėra	Nėra	Nėra	Nėra	Nėra	Nėra

**6.4. IGMO vykdomųjų specifikacijų technologinis įvertinimas**

6.3 lentelėje pateikiami IGMO vykdomųjų specifikacijų technologinio sudėtingumo įvertinimai, atlikti remiantis 4.9 skyrelyje pateiktomis metrikomis.

Santykinio Kolmogorovo sudėtingumo vertės apibrėžia IGMO vykdomosios specifikacijos turinio variantiškumą: didelės vertės rodo, kad turinio variantiškumas (IGMO sudėtingumas) yra didelis, mažos vertės reiškia, kad IGMO specifikacija turi daug pasikartojančių turinio fragmentų.

Didelės metakalbos turtingumo reikšmės reiškia, kad metaprogramoje yra daugiau metaduomenų ir jų aprašymas yra sudėtingesnis.

Didelės normalizuoto sudėtingumo reikšmės rodo, kad norint suprasti IGMO specifikaciją reikia daug laiko ir pastangų.

Ciklomatinis sudėtingumas yra lygus MO egzempliorių, kuriuos gali sugeneruoti IGMO, skaičiui. Didelė ciklomatinio sudėtingumo reikšmė rodo, kad IGMO turi sudėtingesnę parametų rinkinį.

### 6.3 lentelė IGMO vykdomųjų specifikacijų technologinio sudėtingumo įvertinimas

IGMO	Pako- pa	Santykinis Kolmogorovo sudėtingumas	Metakalbos turtinumas	Ciklomatinis sudėtingumas	Normalizuotas sudėtingumas	Pažinimo sudėtingumas
Roboto kalibra- vimas	1	0.24	0.67	360	0.14	185
	2	0.22	0.84	8	0.16	350
	1→2*	0.34	0.72	45	0.17	88
Linijos sekimas	1	0.19	0.75	1152	0.16	473
	2	0.14	0.94	3	0.19	735
	1→2	0.20	0.74	384	0.23	315
Orna- mentų kūrimas	1	0.19	0.60	27216	0.11	246
	2	0.18	0.88	14	0.15	506
	1→2	0.28	0.72	1944	0.13	178
Bėganti eilutė	1	0.26	0.69	24	0.05	220
	2	0.26	0.75	6	0.03	291
	1→2	0.46	0.74	4	0.05	104
Šviesos sekimas	1	0.22	0.64	2916	0.03	262
	2	0.18	0.89	9	0.02	457
	1→2	0.26	0.77	324	0.05	189
Švieso- foras	1	0.21	0.59	2916	0.04	213
	2	0.19	0.84	18	0.03	368
	1→2	0.28	0.74	162	0.06	124

\* 1→2 – vienos pakopos IGMO, sugeneruotas iš dviejų pakopų IGMO.

### 6.5. Realios specializuotos mokymosi aplinkos su integruotais IGMO

Šiame skyrelyje pateikti realių mokymosi aplinkų, kuriuose integruoti sukurti IGMO, pavyzdžiai.

#### 6.5.1. Mokymosi aplinka su vienu mokomuoju robotu

##### ***Mokymosi scenarijus naudojant IGMO***

1. Mokomasis dalykas: *Informacinių technologijų modulis „Programavimo pradmenys“.*

2. Tema: *Ciklas ir ciklas cikle.*

3. E.mokymosi aplinka: *Lego NXT mokomasis robotas piešėjas.*

4. Mokymosi turinys: *Vienos ir dviejų pakopų IGMO.*

5. Mokiniai: *Panevėžio Juozo Balčikonio gimnazijos 10-11 klasių mokiniai.*

6. Pedagoginis modelis: *Konstruktivistinis mokymasis.*

7. Mokymosi tikslai: *Mokymosi turinio ir proceso vizualizacija.*

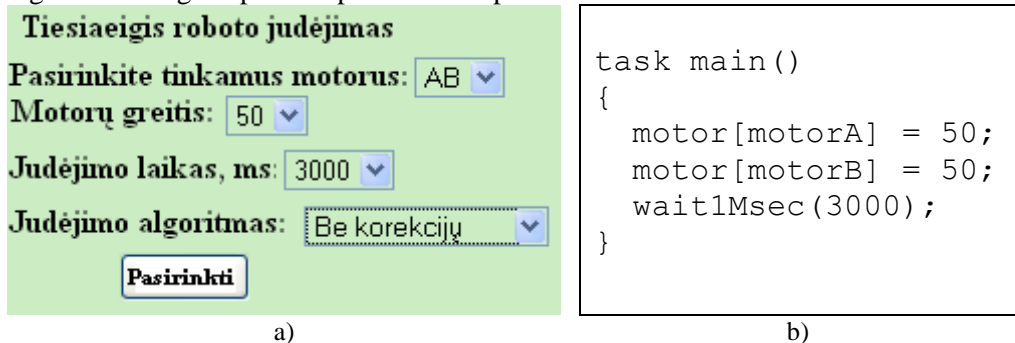
8. Mokymosi proceso aprašymas (mokytojas): *a) e.mokymosi aplinkos sukūrimas ir testavimas; b) vienos pakopos IGMO sukūrimas; c) MO egzempliorių generavimas-testavimas įvairiems mokymosi kontekstams; d) dviejų pakopų IGMO kūrimas: vienos pakopos IGMO transformavimas į dviejų pakopų ir testavimas/generavimas.*

9. Mokiųjų mokomosios veiklos: *a) roboto konstravimas naudojantis iš anksto parengtomis instrukcijomis; b) uždavinio sprendimui tinkamų roboto charakteristikų nustatymas; c) dalyvavimas IGMO kūrimo procese, MO egzempliorių (roboto valdymo programų) kūrimas ir generavimas.*

10. Mokymosi įvertinimas: a) mokytojas stebi mokinių veiklą, suteikia kokybišką grįžtamąjį ryšį: konsultuoja mokinius, stebėdamas mokinių veiklą lanksčiai organizuoja ugdymo procesą; b) mokiniai pristato atliktas užduotis.

Aprašytam scenarijui realizuoti parinkti du IGMO. Pirmasis vadinasi „Roboto kalibravimas“, nes šie parametrai naudojami kuriant roboto valdymo programą. Srities programavimo kalba RobotC naudoja parametru, vadinamą galingumo lygiu (angl. *Power level*). Roboto varikliai yra valdomi apibrėžiant paduodamą galingumo lygį. Galingumo lygis kinta nuo -100 iki 100. Neigiamos reikšmės nurodo roboto judėjimą atgal, teigiamos – į priekį. Roboto per laiko vienetą nuvažiuojamas atstumas priklauso nuo variklių galingumo lygio. Roboto judėjimas priklauso nuo roboto konstrukcijos ir variklių techninių parametru. Roboto tiesiaiegis judėjimas gali būti aprašomas vienu iš trijų algoritmu: 1) į variklio judėjimą aprašanti sakinį įrašant norimą galingumo lygį; 2) taikant variklių sukimosi skirtumų mažinimo principus (angl. *Proportional-Integral-Derivative*, trump. *PID*); 3) taikant variklių sinchronizaciją, kuri užtikrina, kad abu varikliai judės vienodais greičiais [Rob07].

IGMO „Roboto kalibravimas“ vartotojo sąsajos dalis su turinio parametrais ir sugeneruotas egzempliorius pateiktas 6.1 pav.



6.1 pav. IGMO „Roboto kalibravimas“: a) – vartotojo sąsaja, b) – sugeneruotas egzempliorius

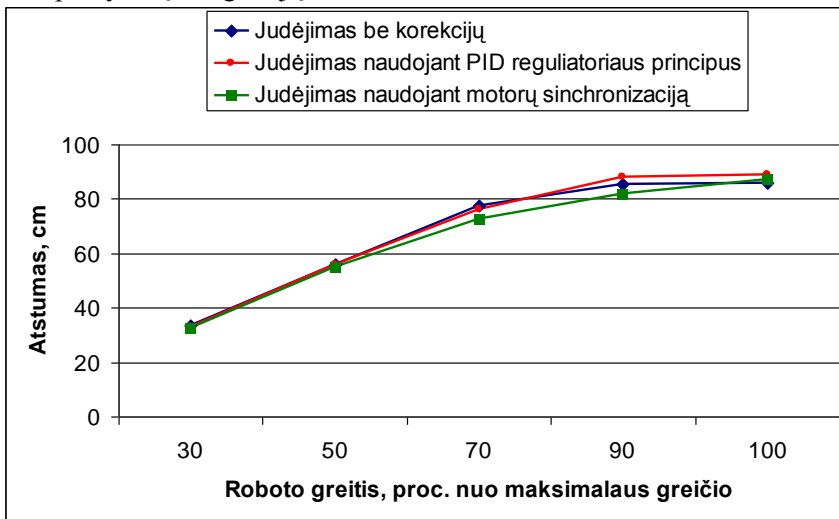
Turint roboto techninių parametru matavimų rezultatus, galima pasirinkti optimalias IGMO parametru vertes. 6.2 pav. pateikti roboto-piešėjo tiesiaiegio judėjimo ir judėjimo apskritimu greičio ir nuvažiuoto atstumo priklausomybės.

Antrasis IGMO „Ornamentų piešimas“ skirtas temos „Ciklas ir ciklas cikle“ sprendžiamų uždavinių vizualizacijai. 6.3 pav. pateikiamas IGMO vartotojo sąsajos fragmentas.

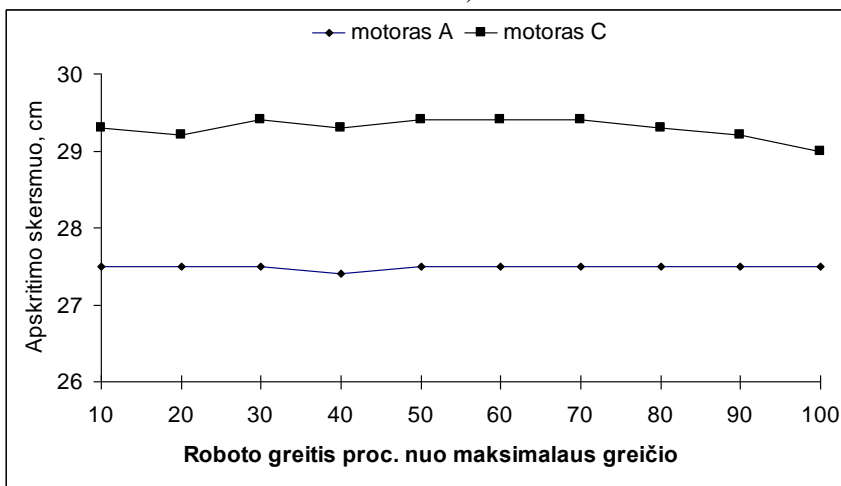
Mokiniai pradiniam etape keisdami IGMO parametru reikšmes, sukuria ir išnagrinėja MO egzempliorius (6.4 pav.). Kitame etape mokiniams pateikiami sugeneruoti MO egzemplioriai, kuriuos mokiniai turi papildyti pačių sukurtu kodu (6.5 pav.).

E.mokymosi aplinka su mokomuoju robotu supažindina mokinius su programavimo žinių ir įgūdžių taikymu realaus gyvenimo uždavinių sprendimui: mokiniai mato, kaip abstraktūs elementai (kintamieji, duomenys, tipai, programavimo struktūros) transformuojami į fizines esybes (roboto judėjimą, greitį). Metodika palaiko MTIM (mokslas, technologija, inžinerija, matematika; angl.

science, technology, engineering, mathematics (STEM)) koncepciją [Ben12] ir numato tarpdalykinę integraciją.



a)



b)

**6.2 pav.** Roboto greičio ir nuvažiuoto atstumo priklausomybės: a – tiesiaiegis judėjimas (judėjimo laikas 3 s), b – judėjimas apskritimu skirtingais greičiais, kai judėjimą valdo skirtingi motorai

**Ornamento kūrimas**

Motorai: AC ▾

Pirmo motoro piešimo greitis: 50 ▾

Antro motoro piešimo greitis: 50 ▾

Piešimo laikas: 1000 ▾

Ornamento dalių skaičius: 4 ▾

**Pasirinkti**

a)

Ornamentų skaičius: 2 ▾

Pirmo motoro judėjimo greitis: 30 ▾

Antro motoro judėjimo greitis: 30 ▾

Judėjimo laikas: 500 ▾ **Pasirinkti**

b)

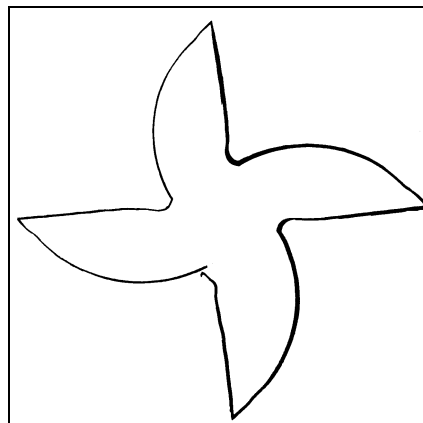
**6.3 pav.** IGMO „Ornamentų piešimas“ vartotojo sąsajos fragmentas: a) vieno ornamento piešimas, b) kelių ornamentų piešimas (vartotojo sąsają sudaro a ir b dalys)

```

task main()
{
//-----
// Pasirengimas piešimui
motor[motorB] = 50;
wait1Msec(100);
motor[motorB] = 0;
//-----
// Piešimas
for (int j = 0; j < 4; j++) {
motor[motorA] = 50;
motor[motorC] = 50;
wait1Msec(1000);
//-----
motor[motorA] = -50;
motor[motorC] = 0;
wait1Msec(1000);
}
//-----
// Ornamento piešimas baigtas
motor[motorB] = -50;
wait1Msec(100);
motor[motorB] = 0;
//-----
}

```

a)



b)

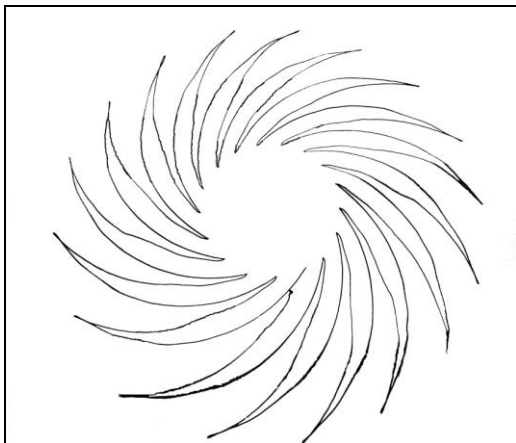
**6.4 pav.** IGMO „Ornamentų piešimas“: a) sugeneruotas MO egzempliorius, b) MO įvykdymo rezultatas

```

// Ornamento parametrai:
// Motorai: AC
// Ornamentų skaičius: 1
// Pirmo motoro piešimo greitis: 45
// Antro motoro piešimo greitis: 15
// Piešimo laikas: 1000
// Vieno ornamento dalių skaičius: 17
task main()
{
    // Įrašykite trūkstamus sakinius
}

```

a)



c)

```

task main()
{
    //-----
    // Pasirengimas piešimui
    motor[motorB] = 50;
    wait1Msec(100);
    motor[motorB] = 0;
    //-----
    // Piešimas
    for (int j = 0; j < 17; j++) {
        motor[motorC] = 45;
        motor[motorA] = 15;
        wait1Msec(1000);
    }
    //-----
    motor[motorC] = -45;
    motor[motorA] = 0;
    wait1Msec(1000);
}
//-----
// Ornamento piešimas baigtas
motor[motorB] = -50;
wait1Msec(100);
motor[motorB] = 0;
//-----
}

```

b)

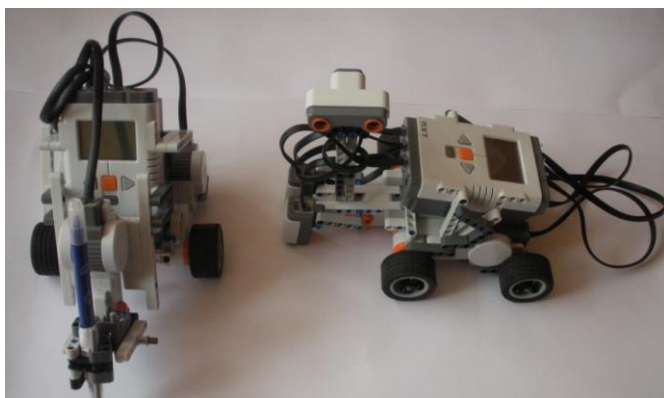
**6.5 pav.** IGMO „Ornamentų piešimas“: a) pradinis sugeneruotas MO egzempliorius, b) MO egzempliorius mokiniams papildžius MO trūkstamais sakiniiais, c) MO įvykdymo rezultatas

### 6.5.2. Mokymosi aplinka su bendradarbiaujančiais IGMO

Pateikiama e.mokymosi aplinka, skirta programavimo mokymui vidurinėje mokykloje. Naudojami heterogeniniai (su skirtingais jutikliais, skirtingų konstrukcijų) LEGO MINDSTORMS NXT robotai, kurie pavadinti pagal jų atliekamas užduotis:

1. Liniją sekantis robotas (6.6 pav., dešinėje) yra valdantysis robotas su dviem šviesos jutikliais, sekantis juodos spalvos liniją popieriaus lape, su ultragarso jutikliu, leidžiančiu nustatyti atstumą iki kliūtis. Liniją sekantis robotas siunčia pranešimą piešiančiam robotui, kai atstumas iki juodos linijos pasidaro kritinis.

2. Piešiantis robotas (6.6 pav., kairėje) yra valdomasis robotas, gaunantis pranešimus iš liniją sekančio roboto ir piešiantis linijas ant popieriaus lapo. Piešiantis robotas buvo adaptuotas iš Burbaitė *ir kt.* [BSM12], kur jis buvo taikomas programavimo uždavinių sprendimų vizualizacijai.



**6.6 pav.** Piešiantis (kairėje) ir liniją sekantis (dešinėje) robotai

Mokytojo kompiuteris suformuluoja užduotis ir paskirsto jas mokinių kompiuteriams. Mokytojo kompiuteris naudoja belaidę interneto kamerą, kad būtų galima stebėti robotų elgseną ir gauti kokybinį grįžtamą ryšį „mokinys-mokytojas“ apie mokinių darbą atliekant užduotis.

Sistemos programinės įrangos dalis sudaryta iš uždavinio skaidymo ir paskirstymo įrangos mokytojo kompiuteryje ir roboto valdymo programų kūrimo aplinkos mokinių kompiuteriuose. Robotų valdymo programos vykdomos RobotC aplinkoje ir perkeliamos į Lego NXT robotus.

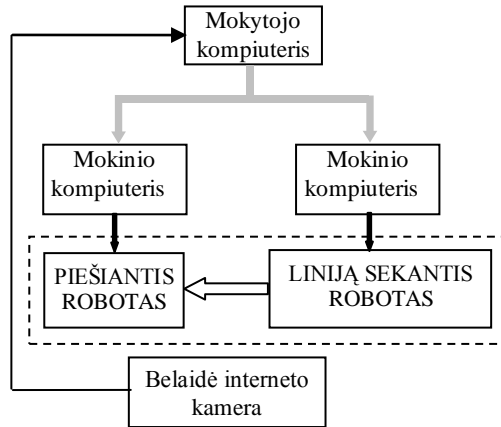
Piešančio roboto valdymo programos sugeneruojamos pasirinkus IGMO parametrų norimas reikšmes. Robotas, vykdydamas įkeltą programą, popieriaus lape piešia figūras.

Besimokantieji gali pasirinkti linijos sekimo algoritmą („Vienas viduje“, „Vienas atšokant“, „Du iš kraštų“, „Du viduje“) [Gra03], išbandyti jį, bei stebėdami roboto elgseną modifikuoti roboto valdymo programą, pridėdami komunikavimo funkcijas, kad liniją sekantis ir piešiantis robotai galėtų bendrauti. „Vienas viduje“ ir „Vienas atšokant“ algoritmai naudoja šviesos jutiklį įtaisytą roboto priekyje, kuris aptinka linijos kraštą. „Du iš kraštų“ ir „Du viduje“ algoritmai naudoja du šviesos jutiklius, įtaisytus roboto priekyje. Taikant algoritmą „Du iš kraštų“ jutikliai yra išdėstyti abiejose linijos kraštuose, o taikant algoritmą „Du viduje“ jutikliai išdėstyti linijos viduje.

Komunikavimas. NXT palaiko Bluetooth taikymo protokolą SPP (angl. *Serial-Port Profile*). Jis yra vykdomas žemo lygio RFCOMM (angl. *Radio Frequency Communication Protocol*) protokolo, kuris vartotojui pateikia patikimą duomenų srautą. Bluetooth užtikrina komunikaciją tarp: 1) kompiuterių ir robotų (daugiau negu trys, bet tik vienas norimu laiko momentu); 2) mokytojo ir mokinių kompiuterių; 3) tarp kitų Bluetooth palaikančių įrenginių (mobiliųjų telefonų, planšetinių kompiuterių). Sujungus įtaisus, nustatomas pavaldumo ryšys. Valdantysis įtaisas sukuria Bluetooth ryšį ir inicijuoja komunikaciją. Kad ryšys būtų patikimas, laikomasi taisyklių, suformuluotų Toledo [Tol06].



Realizuotos sistemos architektūra pateikta 6.7 pav.



6.7 pav. Bendradarbiaujančių robotų sistemos architektūra

Naudojant piešiantį ir liniją sekantį robotus ir aukščiau aprašytus principus, sukurti robotais pagrįsti bendradarbiaujantys MO (angl. *Collaborative Learning Objects, CLOs*) skirti temų „Ciklai“, „Ciklas cikle“, „Šakotieji algoritmai“, „Uždavinio skaidymas į dalis“, „Uždavinių sujungimas“ mokymuisi.

Naudojant aplinką taikytas Reid ir kt. [RFC90] pasiūlytas mokymosi bendradarbiaujant modelis, kurį sudaro 6 etapai:

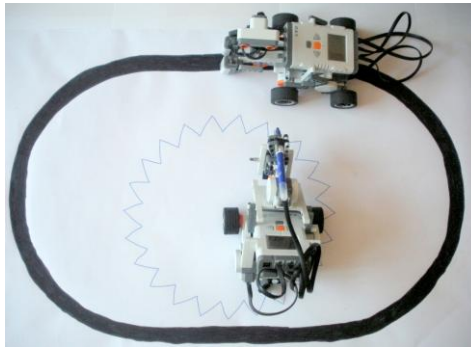
1) *Sudominimas*. Mokiniam pateikiamas sudėtingas uždavinys ir jo skaidymo dalimis, dalių paskirstymo ir koordinavimo idėja. Kaip pavyzdys pateikiamas liniją sekantis robotas, kuris atpažįsta kliūtis. Paaiškinamas uždavinio skaidymo dalimis principas, uždavinio sąlygos ir apribojimai: statinės ir dinaminės kliūtys, uždavinių prioritetai ir pan.

2) *Detalizavimas*. Mokiniai dirbdami grupėse detalizuoja mokytojo pateiktą informaciją. Identifikuojamos dalinės problemos ir formuluojami uždaviniai, kuriuos turės atlikti valdantysis ir valdomasis robotai: linijos sekimas, kliūčių aptikimas ir žinučių siuntimas, atliekami valdančiojo roboto; piešimas, žinučių gavimas ir piešimo stabdymas, kuriuos atlieka valdomasis robotas.

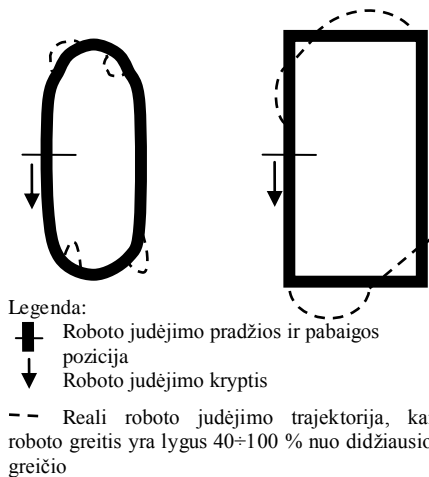
3) *Transformacija*. Mokiniai analizuoja roboto valdymo programų bibliotekoje esančias programas ir jas adaptuoja dalinių uždavinių sprendimui: kliūčių aptikimo, linijos sekimo, ornamentų piešimo.

4) *Tyrimas*. 6.8 pav. pateikiama tyrimo aplinka. Robotas piešėjas piešia ornamentą plote, kurį riboja juodos spalvos linija. Liniją sekantis robotas seka juoda liniją ir tuo pačiu stebi atstumą iki roboto piešėjo. Jeigu atstumas tarp robotų pasidaro per didelis arba per mažas, liniją sekantis robotas siunčia robotui piešėjui žinutę sustoti. Naudoti du skirtingi keliai (6.9 pav.): elipsė, kurios spinduliai yra 21 ir 32 cm, ir stačiakampis, kurio kraštinių ilgiai lygūs 42 ir 70 cm.

Punktyrinė linija vaizduoja eksperimentiškai gautą realų roboto, sekančio liniją, kelią. Kai roboto greitis yra 10 ÷ 30 procentų didžiausio galimo greičio, roboto trajektorija sutampa su juoda linija. Kai roboto greitis yra didesnis, jis nukrypsta nuo trajektorijos.



6.8 pav. Bendradarbiaujantys robotai

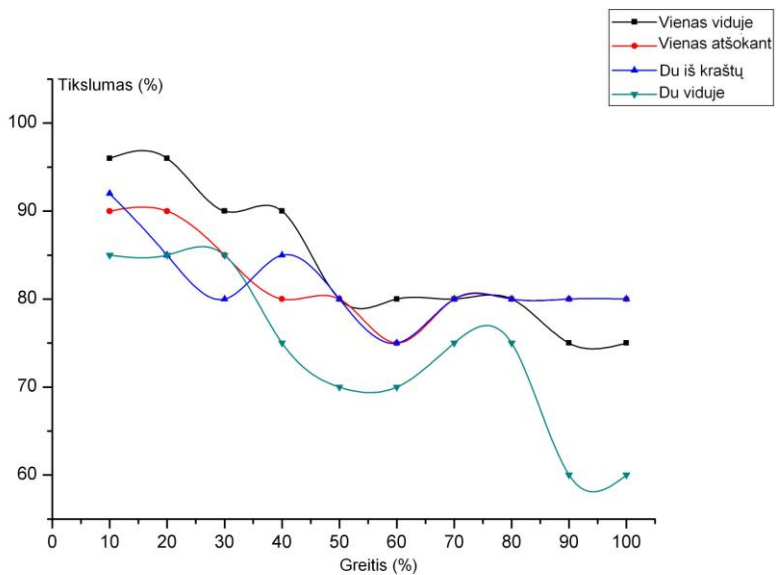
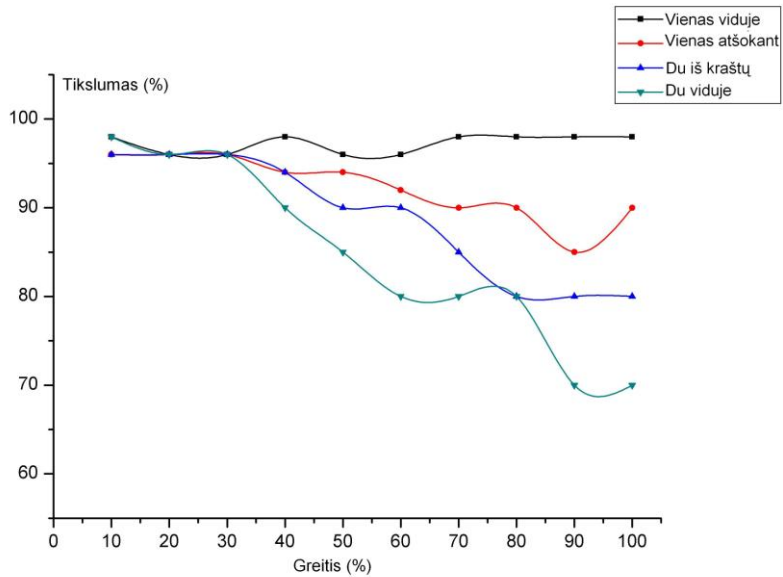


6.9 pav. Liniją sekancio roboto judėjimas elipsės ir stačiakampio formos trajektorija

5) *Pristatymas*. Mokiniai pristato grupių darbo rezultatus ir gauna grįžtamąjį ryšį iš mokytojo ir draugų. Kaip pavyzdys pateikiami linijos sekimo tikslumo, kai robotas juda pagal skirtingus algoritmus, rezultatai (6.10 pav.). Tikslumas skaičiuotas įvertinant, kurią kelio dalį robotas įveikia nenukrypdamas nuo juodos linijos, judėdamas skirtingu greičiu (greitis išreikštas procentais nuo maksimalaus galimo roboto greičio).

6) *Refleksija*. Mokiniai įvertina temos nagrinėjimo, naudotų pedagoginių metodų privalumus ir trūkumus užpildydami mokytojo parengtą klausimyną.

E.mokymosi aplinka su bendradarbiaujančiais IGMO: 1) numato tarpdalykinių mokymosi aspektų realizaciją (užduotys susijusios su mechanika, fizika, matematika, programavimu); 2) didina mokinių motyvaciją; 3) ugdo kritinio mąstymo gebėjimus, problemų sprendimo įgūdžius (pateiktu atveju linijos sekimo algoritmų analizė); 4) skatina mokinius domėtis tiriamąja veikla ir mokytis tinkamai pristatyti atlikto darbo rezultatus.



**6.10 pav.** Linijos sekimo algoritmų tikslumo įvertinimas, kai robotas juda elipsės (viršuje) ir stačiakampio (apačioje) formos keliu

### 6.5.3. Realių mokymosi aplinkų vertinimas technologiniu ir pedagoginiu požiūriu

Šiame skyrelyje pateikiamas darbo autorės atliktas 6.5.1 ir 6.5.2 skyreliuose išnagrinėtų realių mokymosi aplinkų technologinis ir pedagoginis įvertinimas, taikant 5.6 skyrelyje aprašytą metodą. Ekspertinių tyrimų atlikti nebuvo galimybės

dėl aplinkų naujumo. Lietuvoje mokant programavimo robotais ir mikrovaldikliais pagrįstos aplinkos naudojamos labai retai.

6.5 lentelėje pateiktas aplinkų technologinis, 6.6 lentelėje – pedagoginis įvertinimas.

#### 6.4 lentelė Sukurtų mokymosi aplinkų kokybės įvertinimas technologiniu požiūriu\*

Kokybės vertinimo kriterijai	Aplinka	Vieno roboto	Bendradarbiaujančių robotų
Lankstumas		3	4
Moduliškumas		2	3
Veiklos optimizavimas		3	3
Patikimumas ir stabilumas		3	2
Pakartotinis panaudojimas ir perkeliamumas		3	3
Lokaluota vartotojo sąsaja		4	4
Alternatyvių kalbų palaikymas		4	4
Kalbos ir kūrimo adaptavimas pagal konkrečios mokyklos poreikius		3	3
Individualizavimas pagal mokinio poreikius		3	3
Automatiškai adaptuojamas mokymosi turinys		3	3
<b>Technologinių kriterijų sudėtinė naudingumo funkcija</b>		<b>31</b>	<b>32</b>

\* Kiekvienas kriterijus vertinamas ranginiais kintamaisiais 0÷4 skalėje

#### 6.5 lentelė Sukurtų mokymosi aplinkų kokybės įvertinimas pedagoginiu požiūriu

Kokybės vertinimo kriterijai	Aplinka	Vieno roboto	Bendradarbiaujančių robotų
Mokymosi turinio pažinimas*		3	3
Mokymosi proceso pažinimas*		4	4
Pažintiniai mokymosi įgūdžiai*		4	4
Emociniai mokymosi įgūdžiai*		4	4
Socialiniai mokymosi įgūdžiai*		4	4
Pritaikymo įgūdžiai*		4	4
<b>Pedagoginių kriterijų (mokymosi produktų) sudėtinė naudingumo funkcija</b>		<b>23</b>	<b>23</b>
Paruošiamosios mokymosi funkcijos**		P E M	P E M
Vykdomosios mokymosi funkcijos**		P E M	P E M
Baigiamosios mokymosi funkcijos**		P E M	P E M
Mokymosi teorija**		Kn	Kn
Mokinių vaidmenys**		Bn L (I)	Bn L

\* Mokymosi produktai vertinami ranginiais kintamaisiais 0÷4 skalėje.

\*\* Dominuojantys mokymosi proceso funkcijų tipai, mokymosi teorijos ir mokinių vaidmenys žymimi simboliais: P – pažinimo, E – emocinės, M – metapažinimo, B – biheavioristinė, Kg – kognityvinė, Kn – konstruktyvistinė, I – individualus mokymasis, Bn – bendradarbiavimas, L – lenktyniavimas. Skliausteliuose pateikti žymėjimai reiškia, kad sistema tenkina kriterijų, tačiau praktikoje naudojamas retai (pvz., (I) – mokiniai gali pasirinkti individualų mokymąsi, bet praktiškai individualus mokymasis taikomas retai).

## 6.6. IGMO pažinimo sudėtingumo įvertinimas pedagoginiu požiūriu

IGMO pažinimo įvertinimui naudojami Bloomo taksonomijos lygmenys, susieti su pažintiniais procesais [AK01], [MVS+10]. Procesai įvertinami įvedant skalę, susietą su Milerio pažinimo sudėtingumu [Mil56] ( $7 \pm 2$ ):  $< 5$  – lengvas (L), 5..7 – normalus (N), 8..9 – sudėtingas (S),  $> 9$  – labai sudėtingas (LS).

### 6.6 lentelė IGMO pažinimo sudėtingumo vertinimas pedagoginiu požiūriu

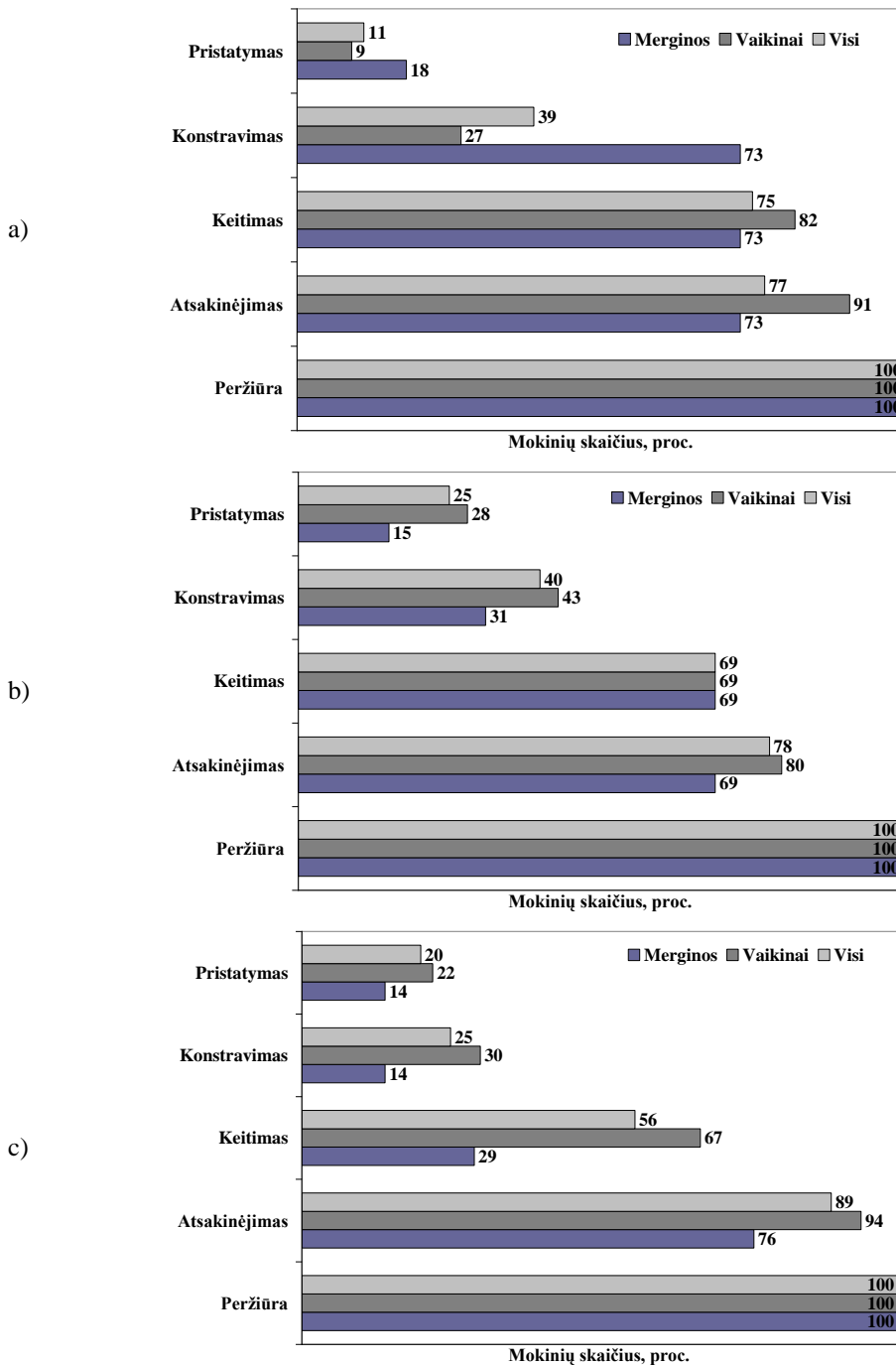
IGMO	Bloomo taksonomijos lygmenys	Pažinimo sudėtingumas, išreikštas per IGMO turinio parametrus, susietus su Milerio metrika					
		Žinios	Supratimas	Pritaikymas	Analizė	Įvertinimas	Kūrimas
Roboto kalibravimas		1 L	3 L	4 L	4 L	4 L	4 L
Linijos sekimas		1 L	2 L	3 L	3 L	3 L	3 L
Ornamentų kūrimas		1 L	5 N	7 N	8 S	9 S	9 S
Bėganti eilutė		1 L	2 L	3 L	3 L	3 L	3 L
Šviesos sekimas		1 L	2 L	3 L	3 L	4 L	4 L
Šviesoforas		1 L	4 L	5 N	5 N	6 N	7 N

Atliktas IGMO pažinimo sudėtingumo vertinimas, išreikštas per IGMO turinio parametrus, susietus su Milerio pažinimo sudėtingumo metrika, sudaro prielaidas tinkamai nustatyti parametų seką vienpakopiame IGMO, arba parametų prioritetus kelių pakopų IGMO.

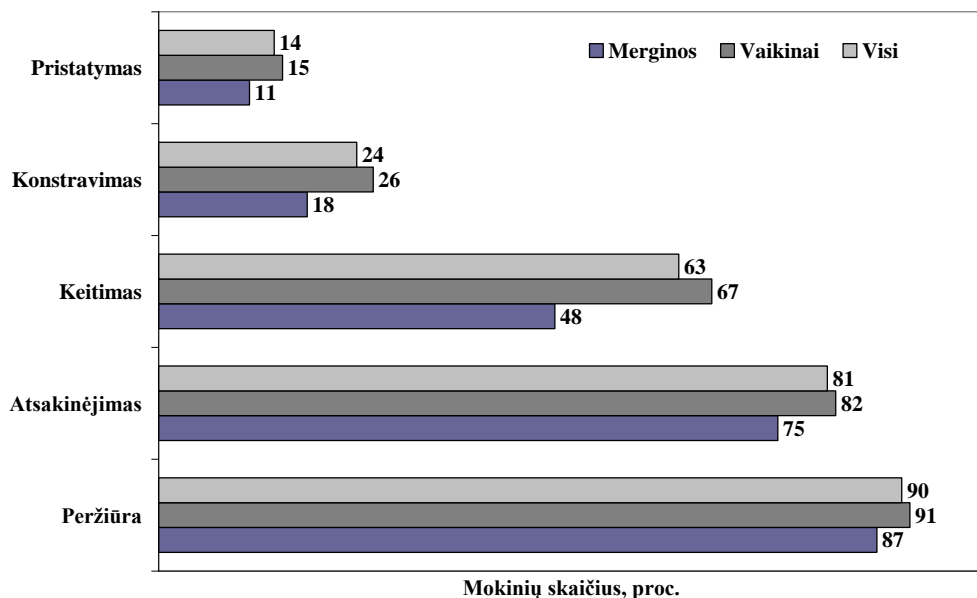
## 6.7. IGMO pedagoginis įvertinimas

IGMO pedagoginiam įvertinimui pasirinkta Urquiza-Fuentes ir Velázquez-Iturbide [UV09] pasiūlyta metodika, pagrįsta Bloomo taksonomija, kai vertinami mokinių išitraukimo lygmenys (angl. *engagement levels*): 1) **peržiūra** – mokiniai peržiūri mokytojo pateiktą programą ir yra pasyvūs mokymosi objekto naudotojai; 2) **atsakinėjimas** – mokiniai naudodami programos vizualizaciją atsako į mokytojo klausimus ir yra aktyvūs mokymosi objekto naudotojai; 3) **keitimas** – mokiniai modifikuoja programą keisdami parametų vertes; 4) **konstravimas** – mokiniai kuria programas patys, naudodamiesi išnagrinėtais pavyzdžiais ir pritaikydami įgytas žinias ir įgūdžius naujuose kontekstuose, pvz. įvesdami naujus parametrus, įvertindami apribojimus; 5) **pristatymas** – mokiniai pristato atliktus darbus, paaiškina gautus rezultatus, apibrėžia nesėkmių priežastis, formuluoja iššūkius ir tolesnių darbų tikslus.

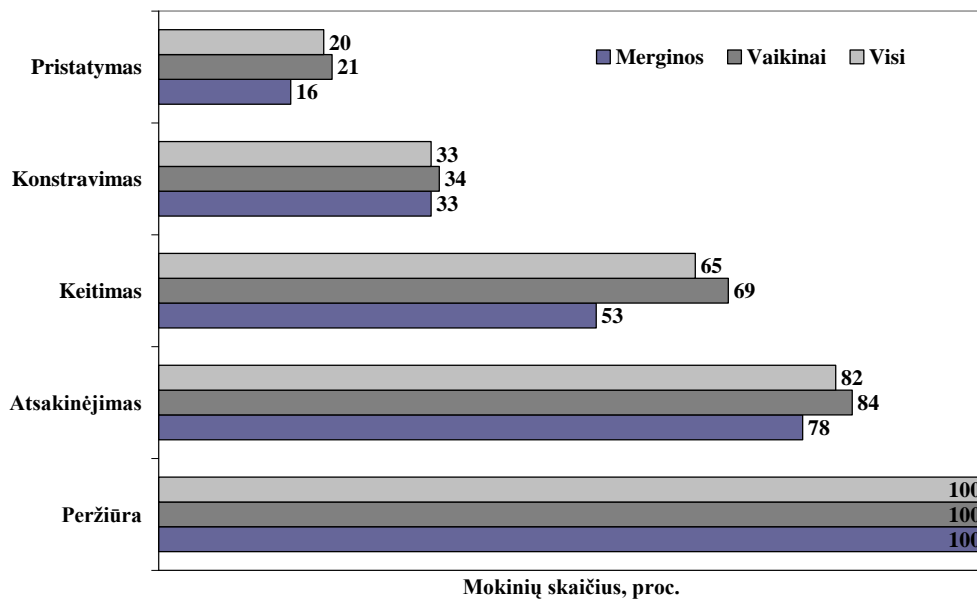
Tyrimas atliktas 2011-2014 m. Panevėžio Juozo Balčikonio gimnazijoje. Tyrimo rezultatai pateikiami 6.11 ir 6.12 pav.



**6.11 pav.** IGMO pedagoginis įvertinimas: a) 2011-2012 m.m. (44 mokiniai, 33 vaikinai, 11 merginų); b) 2012-2013 m.m. (67 mokiniai, 54 vaikinai, 13 merginų); c) 2013-2014 m.m. (75 mokiniai, 54 vaikinai, 21 mergina)



a)



b)

**6.12 pav.** Mokinių išitraukimo pedagoginis įvertinimas (2011-2014 m., 186 mokiniai, 141 vaikas, 45 merginos): a) nenaudojant IGMO; b) naudojant IGMO

## 6.8. Išvados

1. Atlikus informatikos mokymosi posričių ir IGMO požymių modelių įvertinimą, naudojant įrankius FAMILIAR ir SPLOT, nustatyta, kad modeliai yra neprieštaringi, neturi perteklinių požymių ir gali būti pagrįstai naudojami.

2. Atliktas IGMO vykdomųjų specifikacijų technologinio sudėtingumo įvertinimas pagal metaprogramų sudėtingumo metrikas ir nustatyta, kad dvipakopių IGMO santykinis Kolmogorovo ir normalizuotas sudėtingumas keičiasi nežymiai, o metakalbos turtingumas ir pažinimo sudėtingumas padidėja iki 2 kartų ir todėl dvipakopių (daugiapakopių) IGMO kūrimą tikslinga automatizuoti.

3. Sukurtų dviejų specializuotų mokymosi aplinkų pedagoginis įvertinimas parodė, kad aplinkos turi tokius bruožus: i) realizuoja tarpdalykinius mokymosi aspektus; ii) didina mokinių mokymosi motyvaciją; iii) skatina mokinių domėjimąsi tiriamąja veikla ir ugdo kritinio mąstymo gebėjimus, problemų sprendimo, tinkamo darbo pristatymo įgūdžius. Sukurtų aplinkų technologiniai rodikliai, apibūdinantys bendros architektūros, internacionalizavimo ir lokalizavimo, adaptavimo kokybę, yra *geri ir labai geri* pagal pripažintą Likerto vertinimo skalę. Aplinkos su vienu mokomuoju robotu technologinių kriterijų sudėtinė naudingumo funkcija yra lygi 31, su bendradarbiaujančiais robotais – 32 (maksimali funkcijos reikšmė – 40).

4. IGMO pažinimo sudėtingumo vertinimas, išreikštas per turinio parametrus, susietus su Milerio pažinimo sudėtingumo metrika, sudaro prielaidas tinkamai nustatyti parametrų seką vienpakopiame ir prioritetus daugiapakopiame IGMO.

5. Pedagoginiu vertinimu, pritaikius Bloomo taksonomija pagrįstą mokinių įsitraukimo lygmenų metodiką, nustatyta, kad IGMO yra efektyviausi peržiūros, konstravimo ir pristatymo lygmenyse. Pagal 2011-2014 m. eksperimento duomenis skirtinguose įsitraukimo lygmenyse mokymosi efektyvumas pagerėja nuo 6 iki 15 procentų (žr. 6.11-6.12 pav.).



## 7. BAIGIAMASIS ĮVERTINIMAS

Baigiamojo skyriaus tikslas – nustatyti pasiūlytos metodikos ir naudojant ją sukurtų išplėstinių generatyvinių mokymosi objektų vietą e.mokymosi ir programavimo mokymosi srityse. Šiame skyriuje siekiama atsakyti į šitokius klausimus:

1. Kokie yra įvairių tipų mokymosi objektų atributai?
2. Ar IGMO atitinka MO kūrimo tikslus?
3. Kaip IGMO tenkina MO taksonomijas?
4. Ar IGMO tenkina MO standartuose apibrėžtas svarbiausias charakteristikas?
5. Ar IGMO kokybės vertinimui galima pritaikyti standartinių MO kokybės vertinimo metrikas?
6. Kokias programavimo mokymosi temas perdengia sukurti IGMO?
7. Kokie yra mokymosi aplinkų su integruotais IGMO atributai?
8. Ar IGMO gali būti laikomi programų vizualizacijos įrankiais?
9. Kokie pagrindiniai GMO, sukurtų naudojant šablonais grįstą technologiją, ir IGMO, sukurtų taikant heterogeninį metaprogramavimą, panašumai ir skirtumai?
10. Kokie IGMO trūkumai ar ribojimai?
11. Kokios galėtų būti tolesnės darbe pateiktos metodikos vystymo kryptys?

### **1. Kokie yra įvairių tipų mokymosi objektų atributai?**

7.1 lentelėje pateikiama įvairių tipų MO atributų lyginamoji analizė.

**7.1 lentelė** Įvairių MO tipų atributų lyginamoji analizė

MO tipas ir apibrėžimas	Tikslas	Kilmė	Pagrindinė idėja
MO – bet kokia skaitmeninė esybė, kuri gali būti naudojama, pakartotinai naudojama arba nurodoma (angl. <i>referenced</i> ) e.mokymosi (adaptuotas iš [LTSC02])	Sistematizavimas, sąveika, naudojimas ir pakartotinis naudojimas	Iš modulinio projektavimo objektiškai orientuotų idėjų	Tiesiogiai mokymosi naudojamų mokymosi išteklių abstrahavimas
TRUC – esybė, kurią sudaro konceptų aibė, praktiniai įgūdžiai ir vertinimo kriterijai, skirti žinių apie sritį ir jos mokymąsi sistemos sudarymui [Mey06]	Sistematizavimas ir pažinimas	Iš programinės įrangos projektavimo šablonų	Sudaryti sistemą apie srities, kuri taikoma mokymosi procesuose, pažinimą
GMO – pakartotinai naudojamas MO, turintis daugiapakopę struktūrą ir realizuojamas naudojant generatyvinę technologiją (adaptuotas iš Boyle <i>ir kt.</i> darbų)	Pakartotinis naudojimas ir adaptavimas	Iš šablonais grįstos technologijos ir generatyvinių metodų	Adaptavimas per pusiau automatinį turinio variantų generavimą
IGMO – GMO, kuriame įvertinti integruoti konteksto variantiškumo aspektai panaudojant generatyvinę metaprogramavimo technologiją	Išplėstas pakartotinis naudojimas ir adaptavimas	Iš požymių modelių ir generatyvinių metodų	E.mokymosi srities tinkamų požymių parinkimas, modelių sudarymas, verifikavimas ir transformavimas į generatyvinės technologijos parametrus

## **2. Ar IGMO atitinka MO kūrimo tikslus?**

Nustatyta, kaip IGMO atitinka WBITC (*Web-Based Training Information Center*) [[http://www.webbasedtraining.com/trends\\_objects.aspx](http://www.webbasedtraining.com/trends_objects.aspx)] apibrėžtus MO kūrimo tikslus:

1. Pakartotinis naudojimas (angl. *reusability*, “*learning content modularized into small units of instruction suitable for assembly and reassembly into a variety of courses*”) – mokymosi turinys, modularizuotas į smulkius vienetus, kuriuos galima agreguoti ir daug kartų panaudoti. IGMO sustiprina pakartotinį naudojimą, nes atsiranda generatyvinis pakartotinis naudojimas.

2. Tarpusavio sąveika (angl. *interoperability*, “*instructional units that interoperate with each other regardless of developer or learning management system*”) – turinio vienetai sąveikauja tarpusavyje nepriklausomai nuo mokymosi valdymo sistemos ar projektuotojo. IGMO sąveika specifikuota per parametrų reikšmes ir jų sąryšius.

3. Ilgaamžiškumas (angl. *durability*, “*units of instruction that withstand ever evolving delivery and presentation technologies without becoming unusable*”) – turinio vienetai prisitaiko prie nuolat kintančių technologijų. IGMO užtikrina lankstų prisitaikymą per parametrizavimą.

4. Prieinamumas (angl. *accessibility*, “*learning content that is available anywhere, any time – learning content that can be discovered and reused across networks*”) – mokymosi turinys turi būti pasiekiamas bet kurioje vietoje bet kuriuo laiku. IGMO prieinamumą užtikrina generavimo procesą palaikantys ir bendro naudojimo įrankiai.

## **3. Kaip IGMO tenkina MO taksonomijas?**

Nagrinėjant, kaip IGMO tenkina MO taksonomijas, buvo pasirinktos 4 tyrimuose plačiai taikomos MO taksonomijos: 1) Willey MO tipų [Wil00]; 2) Redeker MO pedagoginė [Red03]; 3) Finlay MO naudojimo [Fin04]; 4) Churchill MO tipų [Chu07a].

Willey [Wil00] MO tipų taksonomijoje IGMO įsiterpia tarp generatyvinių pateikiamųjų (angl. *generative-presentation*) ir generatyvinių mokomųjų (angl. *generative-instructional*) MO, turinčių aukštą turinio pakartotinio naudojimo laipsnį skirtinguose kontekstuose.

Redeker [Red03] taksonomijoje IGMO apima viduje interaktyvius (angl. *internally interactive*) ir bendradarbiaujančius (angl. *cooperative*) MO. Viduje interaktyviais suprantami MO, integruojantys mokinį į aktyvią sąveiką „žmogus-kompiuteris“. Bendradarbiaujantys MO užtikrina, kad bus tenkinami darbo grupėmis reikalavimai taikant problemų sprendimo metodą.

Finlay [Fin04] taksonomijoje IGMO perdengia 3 tipus: taikymu pagrįstus (angl. *application-based*), bendradarbiaujančius (angl. *cooperative*) ir individualizuotus (angl. *individualised*) MO.

Remiantis Churchill [Chu07a] MO taksonomija, IGMO apima: 1) pateikiamuosius objektus (angl. *presentation objects*), kuriuos mokytojas naudoja perteikdamas mokiniams naujas žinias; 2) veiklos objektus (angl. *practice objects*), kurie didina besimokančiųjų motyvaciją, užtikrina grįžtamąjį ryšį, suteikia

galimybes pasirinkti užduotis pagal lygį, ugdo problemų sprendimo įgūdžius; 3) konceptuales modelius (angl. *conceptual models*), atvaizduojančius susijusius konceptus ar idėjas, naudojant vizualizacijos technikas; 4) kontekstinius atvaizdavimus (angl. *contextual representations*), kuriuos naudodami besimokantieji dirba pagal realius scenarijus, surenka ir apdoroja duomenis, reikalingus problemai išspręsti.

#### 4. Ar IGMO tenkina MO standartuose apibrėžtas svarbiausias charakteristikas?

IGMO geriausiai aprašomi IEEE LOM standarto metaduomenų elementais. 7.2 lentelėje pateiktas IEEE LOM (*Learning Object Metadata*) standarto bendrųjų ir pedagoginių charakteristikų, aprašančių IGMO struktūrą ir funkcionalumą, rinkinys.

**7.2 lentelė** IEEE LOM standarto pagrindiniai metaduomenų elementai, aprašantys IGMO bendrąsias ir pedagogines charakteristikas

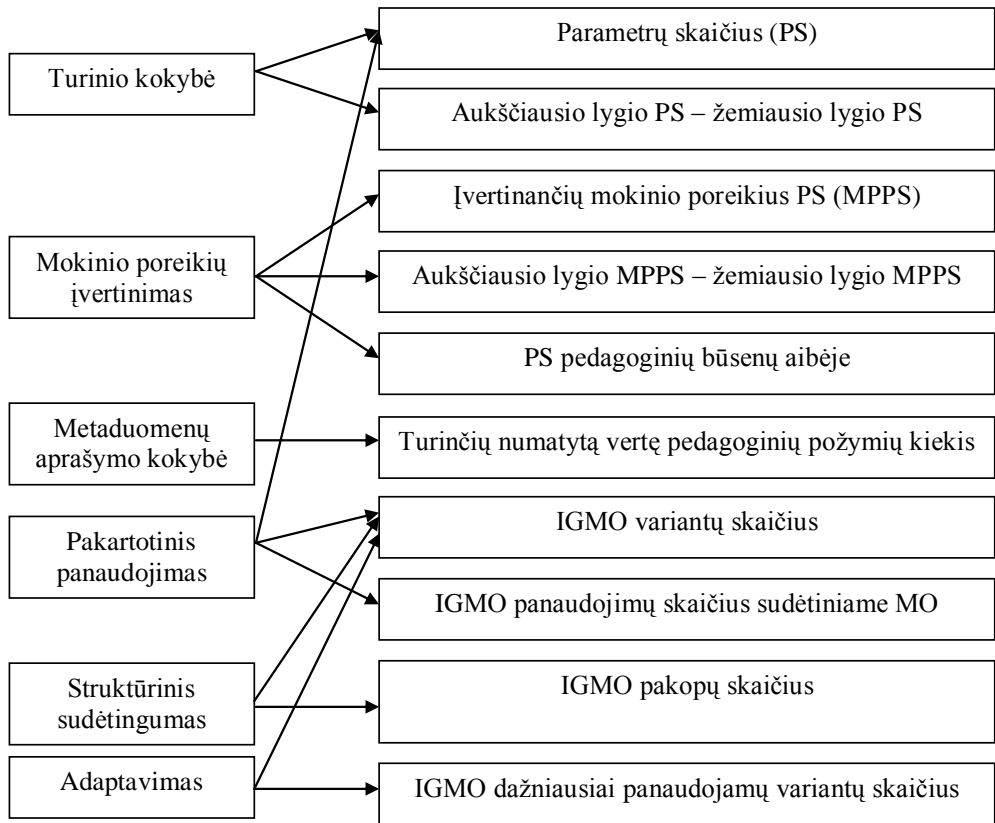
Elemento pavadinimas	Paaiškinimas (remiantis standartais)	Perdengiamas įverčių intervalas (iš standartų)	Paaiškinimas (kaip susijęs su IGMO)
1.8 Agregacijos lygis	Funkcinis MO detalumas (angl. <i>the functional granularity of LO</i> )	1: mažiausias agregacijos lygmuo; 2: rinkinys, sudarytas iš 1 lygmens MO, pvz. pamoka	1: jei MO yra IGMO egzempliorius; 2: jei IGMO yra sudėtinė pamoka kaip MO dalis
5.1 Interaktyvumo tipas	Dominuojantis mokymosi metodas	Aktyvus	Manipuliacijos pasirenkant parametų reikšmes, problemų sprendimas, praktika
5.3 Interaktyvumo lygis	Besimokančiojo įtakos MO elgsenai laipsnis	Vidutinis Aukštas, labai aukštas	Parametų verčių pasirinkimas Parametų verčių korekcija
5.4 Semantinis tankis	MO glaustumo (angl. <i>conciseness</i> ) laipsnis, išreiškiamas per MO apimtį, trukmę	Vidutinis Aukštas, labai aukštas	Vienpakopė vartotojo sąsaja Dvipakopė (daugiapakopė) vartotojo sąsaja
5.5 Galutinio vartotojo vaidmuo	Dominuojantis galutinis MO vartotojas	Mokytojas Mokinys	Parinkdamas tinkamus IGMO konteksto parametrus mokytojas suformuoja specializuotą IGMO mokiniui, kurį mokinys toliau aktyviai naudoja
5.6 Kontekstas, 5.7 Amžiaus tarpsnis	Pagrindinė aplinka, kurioje vyksta mokymasis ir naudojami sukurti MO	Vidurinė mokykla 14-19 metų	IGMO taikomi mokant programavimo vidurinėje mokykloje
5.8 Sunkumas	Kaip vidutinių gebėjimų mokiniai supranta MO	Labai lengvas, lengvas Vidutinio sunkumo Sunkus, labai sunkus	IGMO sunkumas valdomas įvedant konteksto parametą, kuris įvertina mokinių pasirengimo lygį ir pagal jį generuojamas atitinkamas IGMO egzempliorius

#### 5. Ar IGMO kokybės vertinimui galima pritaikyti standartinių MO kokybės vertinimo metrikas?

Remiantis Defude ir Fargat [DF05] išskirtomis šešiomis MO kokybės kriterijų grupėmis ir jų sąsajomis su MO kokybės metrikomis (2.12 pav.) suformuotos metrikos, apibrėžiančios IGMO kokybės kriterijus (7.1 pav.).

7.1 pav. apibrėžtos metrikos glaudžiai susijusios su 6 skyriuje atliktais IGMO modelių ir vykdomųjų specifikacijų vertinimais, juos papildoma naujais aspektais ir

sudaro prielaidas IGMO struktūrinio sudėtingumo valdymui ir adaptavimo galimybių didinimui.



7.1 pav. Metrikos, apibrėžiančios IGMO kokybės kriterijus

### 6. Kokias programavimo mokymosi temas perdengia sukurti IGMO?

Sukurtų ir gimnazijoje ugdymo procese naudotų IGMO ir programavimo modulių, dėstomų Lietuvos bendrojo lavinimo mokyklose, turinio sąsajos pateiktos 7.2 pav.

IGMO „Ornamentų kūrimas“ ir „Linijos sekimas“ gali būti naudojami nepriklausomai, arba sudaryti bendradarbiaujantį IGMO, skirtą temai „Procedūros, funkcijos“ mokyti.

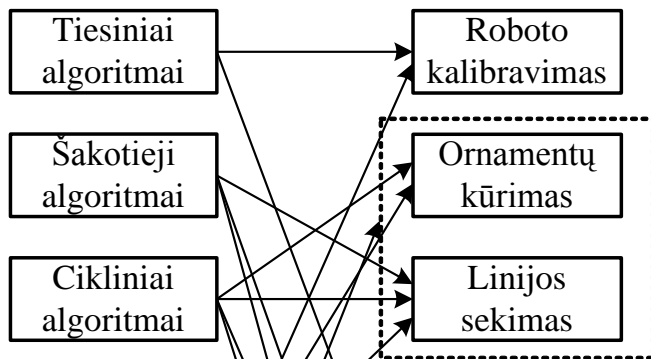
Sukurtų 6 reprezentacinių IGMO rinkinys pilnai perdengia 9-10 klasių programavimo pradžmų modulio kursą ir dalį 11-12 klasės kurso temų.

**Programavimo modulių turinys**

**Į specializuotas mokymosi aplinkas integruoti IGMO**

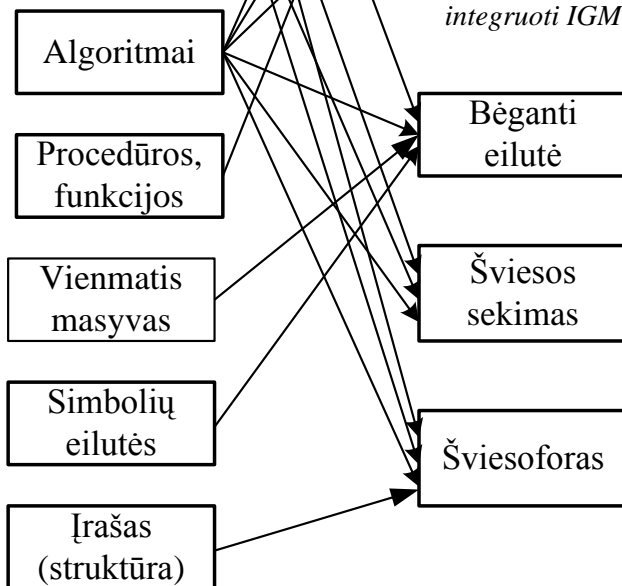
9-10 klasė

Į LEGO NXT aplinką integruoti IGMO



11-12 klasė

Į ARDUINO aplinką integruoti IGMO



7.2 pav. Į specializuotas aplinkas integruotų IGMO ir programavimo modulių turinio sąryšiai

### **7. Kokie yra mokymosi aplinkų su integruotais IGMO atributai?**

Atsakant į iškeltą klausimą programavimo mokymosi sistemų atributai susieti su Kelleher ir Pausch [KP05] nagrinėtais atributais (2.10.5 skyrelis, 2.9 lentelė):

Programavimo stilius – *procedūrinis, funkcinis*.

Programavimo konstruktai – *kintamieji, parametrai, vartotojo apibrėžti duomenų tipai, sąlygos sakiniai, FOR ir WHILE ciklai, procedūros*.

Kodo pateikimas – *tekstas, fiziniai objektai*.

Programų kūrimas – *kodo rašymas, pasirinkimas/formos pildymas, fizinių objektų sujungimas*.

Programų supratimo palaikymas – *ankstesnė patirtis, klaidų taisymas, derinimas, fizinė interpretacija, sugeneruoti pavyzdžiai*.

Sintaksės klaidų vengimas – *galiojančių parinkčių (parametrų verčių) pasirinkimas, į sintaksę orientuotas taisymas*.

Tinkamų programavimo kalbų projektavimas – *srities kalbos pasirinkimas priklausomai nuo specializuotos aplinkos (LEGO NXT aplinka – RobotC, ARDUINO aplinka – Arduino)*.

Bendravimo palaikymas – *darbas grupėse, bendradarbiaujantys IGMO, uždavinio sprendimo suskaidymas į smulkesnes dalis ir išspręstų dalių sujungimas į visumą*.

Uždavinio pasirinkimas – *motyvuojantis, naudingas, mokomasis*.

### **8. Ar IGMO gali būti laikomi programų vizualizacijos įrankiais?**

Remiantis Stephen ir kt. [SFP+12] programų vizualizacijos įrankių taksonomija (2.19 pav.) IGMO galima laikyti programų vizualizacijos mini sistema, kurios sąsaja yra vizualiai paprastai pateikta, sistemos sąveika su vartotoju yra iš anksto apibrėžta, tačiau vartotojas turi galimybę pasirinkti norimas parametrų reikšmes. Sistemoje palaikoma procedūrinė programavimo paradigma, apimanti pagrindinius programavimo konstruktus ir valdymo struktūras. IGMO palaiko dinaminę realiu laiku vykstančią vizualizaciją.

### **9. Kokie pagrindiniai GMO, sukurtų naudojant šablonais grįstą technologiją, ir IGMO, sukurtų taikant heterogeninį metaprogramavimą, panašumai ir skirtumai?**

GMO, sukurti naudojant šablonais grįstą technologiją, ir IGMO, sukurti taikant heterogeninį programavimą, panašūs tuo, kad abiejų tipų GMO specifikaciją kuria žmogus, tačiau IGMO specifikacija yra sintaksiškai orientuota, o šablonais grįstų GMO – neformalizuota.

Šablonais grįsti GMO kuriami rankiniu būdu užpildant šabloną (ikeliant į šabloną teksto, garso, vaizdo failus, nuorodas į turinio fragmentus), IGMO – parenkant tinkamas parametrų reikšmes. Didžiausias skirtumas tarp šablonais grįstų GMO ir IGMO yra tas, kad IGMO mokymosi turinys generuojamas automatiškai, tuo tarpu kuriant šablonais grįstus GMO naudojamas jau sukurtas mokymosi turinys.

Šablonais grįsti GMO kuriami naudojant įrankį GLO Maker [<http://www.glovermaker.org>], IGMO – metakalbos ir tikslo kalbos kompiliatorius.

Šablonais grįsti GMO plačiausiai taikomi mokant medicinos (Boyle ir kt. darbai), buhalterinės apskaitos [Old08], skaitmeninio raštingumo įgūdžių

[<http://www.glomaker.org/docs/DigiLitSkillsFlyerAug11.pdf>]. IGMO taikomi programavimo mokymuisi. IGMO kūrimo ir integravimo į mokomasias aplinkas metodika palaiko MTIM (mokslas, technologija, inžinerija, matematika; angl. *science, technology, engineering, mathematics (STEM)*) koncepciją [Ben12] ir numato tarpdalykinę integraciją.

### **10. Kokie IGMO trūkumai ar ribojimai?**

Išplėstinių generatyvinių mokymosi objektų kūrimas reikalauja aukštos mokytojo technologinės ir pedagoginės kompetencijos.

Sukurti išplėstiniai generatyviniai mokymosi objektai apima pirmus du agregacijos lygius, 3 ir 4 lygiai nepasiekti, nors metodas tai įgalina (7.2 lentelė).

Specializuotų mokomųjų aplinkų kūrimas reikalauja ne tik informatikos, bet ir kitų technologinių disciplinų žinių, turi savo įdiegimo specifiką.

Bendradarbiaujančių mokomųjų robotų panaudojimo galimybės nėra pilnai išnagrinėtos.

### **11. Kokios galėtų būti tolesnės darbe pateiktos metodikos vystymo kryptys?**

Tolesnės darbe pateiktos metodikos vystymo kryptys *galėtų būti*: (1) gilesnis ir platesnis IGMO integravimas į mokymosi aplinkas tarpdalykiniam mokymuisi pagal STEM (angl. *Science, Technology, Engineering, Mathematics*) koncepciją; (2) automatizuoti įrankiai IGMO kūrimui; (3) IGMO daugiapakopių struktūrų tyrimai siekiant lankstesnio mokymosi turinio (MO) adaptavimo.

## IŠVADOS

1. Atlikta literatūros analizė rodo, kad e.mokymosi metodologiniai pagrindai yra bendri visai e.mokymosi sričiai, tačiau informatikos (programavimo) mokymasis reikalauja atskiro požiūrio ir tyrimų. Pasiūlyta nauja generatyvinių mokymosi objektų su išplėstinėmis galimybėmis koncepcija. Jos metodologinis pagrindas – srities variantiškumo modeliavimas pritaikant metaprogramavimu grindžiamą realizaciją.

2. Sukurtas programavimo mokymosi srities modeliavimo metodas, pagrįstas požymių konceptais, jų atskirties principu, požymių variantais, jų sąryšiais bei sąveika bei tikslui orientuotais procesais. Modeliavimui pritaikyti žinomi įrankiai (FAMILIAR, SPLOT) užtikrina modelių korektiškumą ir pateikia esmines jų charakteristikas įvertinimui. Modeliavimo išdavoje gaunamas *bendrinis srities modelis*.

3. Pasiūlyta išplėstinių generatyvinių mokymosi objektų (IGMO) sudarymo metodika apima du lygmenis: konkrečių modelių kūrimo (*išgavimo* iš bendrinio modelio) ir tų modelių transformavimo į metaprogramavimu grindžiamas vykdomąsias specifikacijas:

- Konkrečių modelių specifikacijos sudarytos iš konteksto ir turinio modelių, kuriuos semantiškai susieja prioritetų modelis su sąryšiais ir apribojimais. Prioritetų modelis įgalina valdyti konkretaus modelio sudėtingumą ir sukurti sąlygas turinio adaptavimui.
- Vykdomosios specifikacijos yra įrankis, įgalinantis *automatiškai kurti* mokymosi turinį *skirtingiems ugdymo kontekstams*.

4. Sukurtos specializuotos heterogeninės mokymosi aplinkos, į kurias integruoti IGMO, įgyvendina realaus uždavinio vizualinę transformaciją į fizinį procesą bei užtikrina aukštą mokinių motyvaciją ir efektyvų mokymąsi.

5. Sukurtų objektų pažinimo sudėtingumo vertinimas, išreikštas per turinio parametrus, susiejus juos su Milerio pažinimo metrika, įgalina nustatyti tinkamą parametrų seką specifikacijose, kad būtų galima valdyti sudėtingumą projektuojant ir naudojant IGMO.

6. Atliktas IGMO pedagoginis vertinimas taikant Bloomo taksonomija pagrįstą mokinių įsitraukimo lygmenų metodiką leidžia daryti išvadą, kad IGMO yra efektyviausi peržiūros, konstravimo ir pristatymo lygmenyse. Pagal 2011-2014 m. eksperimento duomenis skirtinguose įsitraukimo lygmenyse mokymosi efektyvumas pagerėja nuo 6 iki 15 procentų.



7. Nustatyta IGMO esminių atributų vieta e.mokymesi apskritai ir programavimo mokymesi konkrečiai, remiantis visuotinai pripažintais standartais bei taksonomijomis. Sukurtos metodikos privalumus patvirtina atlikti šitokie palyginimai.

IGMO e.mokymesi atitinka:

- Visus 4 WBITC (*Web-Based Training Information Center*) apibrėžtus MO kūrimo tikslus (pakartotinio panaudojimo, tarpusavio sąveikos, ilgaamžiškumo, prieinamumo).
- 4 plačiai naudojamas MO taksonomijas (Willey, Redeker, Finlay, Churchill).
- IEEE LOM standartuose apibrėžtas *svarbiausias* bendrąsias ir pedagogines MO charakteristikas.

IGMO sukurti programavimo mokymuisi tenkina tokias sąlygas:

- 6 reprezentaciniai IGMO 100 % perdengia vidurinės mokyklos 9-10 klasės programavimo pradmenų modulį ir 70 % 11-12 klasės programavimo modulio temų.
- Integruoti į mokomaisiais robotais grįstas aplinkas atitinka Kelleher ir Pausch programavimo aplinkų ir įrankių taksonomijos *pagrindinius* atributus.

## LITERATŪRA

1. [AAB+06] Abarca, M. G., Alarcon, R. A., Barria, R., & Fuller, D. (2006, January). Context-based e-learning composition and adaptation. In *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops* (pp. 1976-1985). Springer Berlin Heidelberg.
2. [ABH+13] Acher, M., Baudry, B., Heymans, P., Cleve, A., & Hainaut, J. L. (2013, January). Support for reverse engineering and maintaining feature models. In *Proceedings of the Seventh International Workshop on Variability Modelling of Software-intensive Systems* (p. 20). ACM.
3. [ACL+13] Acher, M., Collet, P., Lahire, P., & France, R. B. (2013). Familiar: A domain-specific language for large scale management of feature models. *Science of Computer Programming*, 78(6), 657-681.
4. [AD06] Azouaou, F., & Desmoulins, C. (2006, July). A Flexible and Extensible Architecture For Context-Aware Annotation in E-Learning. In *ICALT* (pp. 22-26).
5. [ADP+11] Antonis, K., Daradoumis, T., Papadakis, S., & Simos, C. (2011). Evaluation of the effectiveness of a web-based learning design for adult computer science courses. *Education, IEEE Transactions on*, 54(3), 374-380.
6. [AG03] Adamchik, V., & Gunawardena, A. (2003, April). A learning objects Approach to teaching programming. In *Information Technology: Coding and Computing [Computers and Communications], 2003. Proceedings. ITCC 2003. International Conference on* (pp. 96-99). IEEE.
7. [AG13] Anderson, N., & Gegg-Harrison, T. (2013, March). Learning computer science in the comfort zone of proximal development. In *Proceeding of the 44th ACM technical symposium on Computer science education* (pp. 495-500). ACM.
8. [AHH11] Alharbi, A., Henskens, F., & Hannaford, M. (2011, December). Computer science learning objects. In *e-Education, Entertainment and e-Management (ICEEE), 2011 International Conference on* (pp. 326-328). IEEE.
9. [AHH12] Alharbi, A., Henskens, F., & Hannaford, M. (2012). Student-Centered Learning Objects to Support the Self-Regulated Learning of Computer Science. *Creative Education*, 3, 773.
10. [AK01] Anderson, L. W., & Krathwohl, D. R. DR, et al (Eds.) (2001) A taxonomy for learning, teaching, and assessing: A revision of bloom's taxonomy of educational objectives.
11. [Akp08] Akpinar, Y. (2008). validation of a learning object Review Instrument: Relationship between Ratings of learning objects and Actual learning outcomes. *Interdisciplinary Journal of E-Learning and Learning Objects*, 4(1), 291-302.
12. [Ala12] Ala-Mutka, K. (2003). Problems in learning and teaching programming. *Codewitz Needs Analysis*.
13. [AM10] Allen, C. A., & Mugisa, E. K. (2010). Improving Learning Object Reuse Through OOD: A Theory of Learning Objects. *Journal of Object Technology*, 9(6), 51-75.
14. [APH+11] Alharbi, A., Paul, D., Henskens, F., & Hannaford, M. (2011, December). An investigation into the learning styles and self-regulated learning strategies for computer science students. In *Proceedings Ascilite*.

15. [AR07] Alaña, E., & Rodriguez, A. (2007). Domain Engineering Methodologies Survey. *GMV Innovating Solutions*.
16. [BAG+10] Bagheri, E., Asadi, M., Gasevic, D., & Soltani, S. (2010). Stratified analytic hierarchy process: Prioritization and selection of software features. In *Software Product Lines: Going Beyond* (pp. 300-315). Springer Berlin Heidelberg.
17. [Bat05] Batory, D. (2005). *Feature models, grammars, and propositional formulas* (pp. 7-20). Springer Berlin Heidelberg.
18. [BB09] Bygholm, A., & Buus, L. (2009). Managing the Gap between Curriculum Based and Problem Based Learning: Deployment of Multiple Learning Strategies in Design and Delivery of Online Courses in Computer Science. *International Journal of Education & Development using Information & Communication Technology*, 5(1).
19. [BBC04] Boyle, T., Bradley, C., & Chalk, P. (2004). Improving the teaching of programming using a VLE enhanced with learning objects. In *Information Technology: Research and Education, 2004. ITRE 2004. 2nd International Conference on* (pp. 74-78). IEEE.
20. [BC10] Barker, P., & Campbell, L. M. (2010). Metadata for Learning Materials: An Overview of Existing Standards and Current Developments. *Technology, Instruction, Cognition & Learning*, 7.
21. [BDB+06] Botturi, L., Derntl, M., Boot, E., & Figl, K. (2006, July). A classification framework for educational modeling languages in instructional design. In *6th IEEE International Conference on Advanced Learning Technologies (ICALT 2006), Kerkrade (The Netherlands)* (pp. 1216-1220).
22. [Ben12] Benitti, F. B. V. (2012). Exploring the educational potential of robotics in schools: A systematic review. *Computers & Education*, 58(3), 978-988.
23. [BEP+09] Berglund, A., Eckerdal, A., Pears, A., East, P., Kinnunen, P., Malmi, L., ... & Thomas, L. (2009). Learning computer science: Perceptions, actions and roles. *European Journal of Engineering Education*, 34(4), 327-338.
24. [BG11] Bagheri, E., & Gasevic, D. (2011). Assessing the maintainability of software product line feature models using structural metrics. *Software Quality Journal*, 19(3), 579-612.
25. [BM12] Barreiros, J., & Moreira, A. (2012). Soft Constraints in Feature Models: An Experimental Assessment. *International Journal On Advances in Software*, 5(3 and 4), 252-262.
26. [BMO08] Balatsoukas, P., Morris, A., & O'Brien, A. (2008). Learning Objects Update: Review and Critical Approach to Content Aggregation. *Journal of Educational Technology & Society*, 11(2).
27. [Boy06] Boyle, T. (2006). The design and development of second generation learning objects. In *World Conference on Educational Multimedia, Hypermedia and Telecommunications* (Vol. 2006, No. 1, pp. 2-12).
28. [Boy09] Boyle, T. (2009, March). Generative learning objects (GLOs): design as the basis for reuse and repurposing. In *First International Conference 'e-Learning and Distance Education*.
29. [Boy10] Boyle, T. (2010). Layered learning design: Towards an integration of learning design and learning object perspectives. *Computers & Education*, 54(3), 661-668.
30. [Bos00] Bosch, J. (2000). *Design and use of software architectures: adopting and evolving a product-line approach*. Pearson Education.

31. [BR12] Boyle, T., & Ravenscroft, A. (2012). Context and deep learning design. *Computers & Education*, 59(4), 1224-1233.
32. [BS11] Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community?. *ACM Inroads*, 2(1), 48-54.
33. [BS99] DeBaud, J. M., & Schmid, K. (1999, May). A systematic approach to derive the scope of software product lines. In *Software Engineering, 1999. Proceedings of the 1999 International Conference on* (pp. 34-43). IEEE.
34. [BSD13] Burbaite, R., Stuikeys, V., & Damasevicius, R. (2013, July). Educational robots as collaborative learning objects for teaching Computer Science. In *System Science and Engineering (ICSSE), 2013 International Conference on* (pp. 211-216). IEEE.
35. [BSM12] Burbaite, R., Stuikeys, V., & Marcinkevicius, R. (2012). The LEGO NXT Robot-based e-Learning Environment to Teach Computer Science Topics. *Electronics & Electrical Engineering*, 18(9).
36. [But01] Butler, G. (2001). Generative techniques for product lines. *ACM SIGSOFT Software Engineering Notes*, 26(6), 74-76.
37. [CAC12] Costa, C. J., Aparicio, M., & Cordeiro, C. (2012, June). A solution to support student learning of programming. In *Proceedings of the Workshop on Open Source and Design of Communication* (pp. 25-29). ACM.
38. [CAL12] Campos, A. M., Alvarez-Gonzalez, L. A., & Livingstone, D. E. (2012). ANALYZING EFFECTIVENESS OF PEDAGOGICAL SCENARIOS FOR LEARNING PROGRAMMING A LEARNING PATH DATA MODEL. *Editor: Ion Mierluș-Mazilu*, 51.
39. [CBK13] Capilla, R., Bosch, J., & Kang, K. C. (2013). *Systems and Software Variability Management*. Springer.
40. [CBS+11] Campos, F., Braga, R., Souza, A. C., Santos, N., Matos, E., & Nery, T. (2011). Projeto broad: Busca semântica por objetos de aprendizagem. In *ESUD 2011-VIII Congresso Brasileiro de Ensino Superior a Distância, Ouro Preto*.
41. [CC10] Cooper, S., & Cunningham, S. (2010). Teaching computer science in context. *ACM Inroads*, 1(1), 5-8.
42. [CE00] Czarnecki, K., & Eisenecker, U.W. (2000). *Generative programming: Methods, Tools, and Applications*. Addison-Wesley.
43. [CETL14] CETL. Reusable learning objects. What are GLO's. Prieiga per internetą: <http://www.rlo-cetl.ac.uk/whatwedo/glos/whatareglos.php>. Žiūrėta: 2014-04-16
44. [CGS12] Chawla, S., Gupta, N., & Singla, R. K. (2012). LOQES: Model for Evaluation of Learning Object. *International Journal of Advanced Computer Science & Applications*, 3(7).
45. [CHE04] Czarnecki, K., Helsen, S., & Eisenecker, U. (2004). Staged configuration using feature models. In *Software Product Lines* (pp. 266-283). Springer Berlin Heidelberg.
46. [CHE05] Czarnecki, K., Helsen, S., & Eisenecker, U. (2005). Staged configuration through specialization and multilevel configuration of feature models. *Software Process: Improvement and Practice*, 10(2), 143-169.
47. [CHH09] Classen, A., Hubaux, A., & Heymans, P. (2009, January). A Formal Semantics for Multi-level Staged Configuration. In *VaMoS* (pp. 51-60).

48. [Chu07] Chudá, D. (2007, June). Visualization in education of theoretical computer science. In *Proceedings of the 2007 international conference on Computer systems and technologies* (p. 84). ACM.
49. [Chu07a] Churchill, D. (2007). Towards a useful classification of learning objects. *Educational Technology Research and Development*, 55(5), 479-497.
50. [CHW98] Coplien, J., Hoffman, D., & Weiss, D. (1998). Commonality and variability in software engineering. *Software, IEEE*, 15(6), 37-45.
51. [CISCO03] CISCO S (2003) Reusable learning object strategy: Designing and developing learning objects for multiple learning approaches. White paper, CISCO.
52. [CL13] Collet, P., & Lahire, P. (2013, July). Feature modeling and separation of concerns with FAMILIAR. In *Comparing Requirements Modeling Approaches Workshop (CMA@RE), 2013 International* (pp. 13-18). IEEE.
53. [CLA06] Caeiro-Rodríguez, M., Llamas-Nistal, M., & Anido-Rifón, L. (2006, October). A separation of concerns approach to educational modeling languages. In *Frontiers in Education Conference, 36th Annual* (pp. 9-14). IEEE.
54. [CLF+09] Cervera, J. F., López-López, M. G., Fernández, C., & Sánchez-Alonso, S. (2009). Quality metrics in learning objects. In *Metadata and Semantics* (pp. 135-141). Springer US.
55. [CMF+09] Castillo, J. F., Montes de Oca, C., Flores, E. S., & Elizondo, P. V. (2009, February). Toward an Approach to Programming Education to Produce Qualified Software Developers. In *Software Engineering Education and Training, 2009. CSEET'09. 22nd Conference on* (pp. 101-104). IEEE.
56. [CNC12] Castro, J., Nazar, J. M., & Campos, F. (2012). EasyT: Apoiando a Construção de Objetos de Aprendizagem para uma Linha de Produtos de Software. *Conferencias LACLO*, 3(1).
57. [CS05] Cuadrado, J. J., & Sicilia, M. A. (2005). Learning object reusability metrics: Some ideas from software engineering.
58. [CTT10] Corney, M., Teague, D., & Thomas, R. N. (2010, January). Engaging students in programming. In *Proceedings of the Twelfth Australasian Conference on Computing Education-Volume 103* (pp. 63-72). Australian Computer Society, Inc.
59. [DAB12] Dillon, E., Anderson, M., & Brown, M. (2012). Comparing feature assistance between programming environments and their effect on novice programmers. *Journal of Computing Sciences in Colleges*, 27(5), 69-77.
60. [DBC+10] Das, M., Bhaskar, M., Chithralekha, T., & Sivasathya, S. (2010). Context Aware E-Learning System with Dynamically Composable Learning Objects. *International Journal on Computer Science and Engineering*, 2(4), 1245-1253.
61. [DDA12] Diez, D., Díaz, P., & Aedo, I. (2012). The ComBLA Method: The Application of Domain Analysis to the Development of e-Learning Systems. *Journal of Research & Practice in Information Technology*, 44(3).
62. [DF05] Defude, B., & Farhat, R. (2005, July). A framework to design quality-based learning objects. In *Advanced Learning Technologies, 2005. ICALT 2005. Fifth IEEE International Conference on* (pp. 23-27). IEEE.
63. [DGJ+04] Doorten, M., Giesbers, B., Janssen, J., Daniels, J., & Koper, R. (2004). Transforming existing content into reusable learning objects. *Online education using learning objects*, 116-127.

64. [DGJ08] Dagienė, V., Grigas, G., & Jevsikova, T. (2008). *Enciklopedinis kompiuterijos žodynas, II papildytas leidimas*. Vilnius. TEV.
65. [DK02] Van Deursen, A., & Klint, P. (2002). Domain-specific language design requires feature descriptions. *CIT. Journal of computing and information technology*, 10(1), 1-17.
66. [DKM04] Dreher, H., Krotzmaier, H., & Maurer, H. A. (2004). What we Expect from Digital Libraries. *J. UCS*, 10(9), 1110-1122.
67. [DS09] Damasevicius, R., & Štuikys, V. (2009, July). Specification and generation of learning object sequences for E-learning using sequence feature diagrams and metaprogramming techniques. In *Advanced Learning Technologies, 2009. ICALT 2009. Ninth IEEE International Conference on* (pp. 572-576). IEEE.
68. [DSS+08] Dorairaj, S. K., Singh J., Shanmugam, M., & Shamini, S. (2008). Experimenting with Industry's Pair-Programming Model in Teaching and Learning Programming. In *Proceedings of the 4th International Conference on Information Technology and Multimedia at UNITEN (ICIMU' 2008)*, Malaysia.
69. [DSV04] De Kock, A., Slegers, P., & Voeten, M. J. (2004). New learning and the classification of learning environments in secondary education. *Review of educational research*, 74(2), 141-170.
70. [DŠ08] Damaševičius, R., & Štuikys, V. (2008). On the technological aspects of generative learning object development. In *Informatics Education-Supporting Computational Thinking* (pp. 337-348). Springer Berlin Heidelberg.
71. [DŠ10] Damaševičius, R., & Štuikys, V. (2010). Metrics for evaluation of metaprogram complexity. *Computer Science and Information Systems*, 7(4), 769-787.
72. [EAG+08] Ericson, B., Armoni, M., Gal-Ezer, J., Seehorn, D., Stephenson, C., & Trees, F. (2008). Ensuring exemplary teaching in an essential discipline: Addressing the crisis in computer science teacher certification. *Final Report of the CSTA Teacher Certification Task Force*. ACM.
73. [EAJ+10] Essalmi, F., Ayed, L. J. B., Jemni, M., & Graf, S. (2010). A fully personalization strategy of E-learning scenarios. *Computers in Human Behavior*, 26(4), 581-591.
74. [Fig06] de Figueiredo, A. D. (2006). From content to context in technology supported computer science education. In *e-learning conference Computer Science Education*, University of Coimbra.
75. [Fig10] Dias de Figueiredo, A. (2010). Learning Contexts: a Blueprint for Research. *Digital Education Review*, (11), 127-139.
76. [Fin04] Finlay, J. (2004). CONTEXT-NEUTRAL ELearning OBJECTS: A TALE OF TWO PROJECTS. In *the 7th HCI Educators Workshop: Effective Teaching and Training in HCI*.
77. [FL09] Fletcher, G. H., & Lu, J. J. (2009). Education Human computing skills: rethinking the K-12 experience. *Communications of the ACM*, 52(2), 23-25.
78. [GA03] Gunawardena, A., & Adamchik, V. (2003). A customized learning objects approach to teaching programming. *ACM SIGCSE Bulletin*, 35(3), 264.
79. [GC06] Gulatee, Y., & Combes, B. (2006). Identifying the Challenges in Teaching Computer Science Topics Online. In *Proceedings of the EDU-COM 2006 International Conference*. Engagement and Empowerment: New Opportunities for Growth in Higher Education, Edith Cowan University, Perth Western Australia.

80. [Ger67] Gerard, R. W. (1967). Shaping the mind: Computers in education. *Applied Science and Technological Progress: A Report to the Committee on Science and Astronautics, US House of Representatives*, 207.
81. [GGL+12] Goldberg, D. S., Grunwald, D., Lewis, C., Feld, J. A., & Hug, S. (2012, July). Engaging computer science in traditional education: the ECSITE project. In *Proceedings of the 17th ACM annual conference on Innovation and technology in computer science education* (pp. 351-356). ACM.
82. [GM07] Gomes, A., & Mendes, A. J. (2007, June). An environment to improve programming education. In *Proceedings of the 2007 international conference on Computer systems and technologies* (p. 88). ACM.
83. [GMB11] Gheyi, R., Massoni, T., & Borba, P. (2011). Automatically Checking Feature Model Refactorings. *J. UCS*, 17(5), 684-711.
84. [Gra03] Gray, J.A. Toeing the Line: Experiments with Line-following Algorithms. [Online]. Available: [http://www.fll-freak.com/misc/01-jgray\\_report.pdf](http://www.fll-freak.com/misc/01-jgray_report.pdf)
85. [Gre08] Greer, D. (2008). The Art of Separation of Concerns. <http://ctrl-shift-b.blogspot.com/2008/01/art-of-separation-of-concerns.html>
86. [Had08] Hadjerrouit, S. (2008). Towards a blended learning model for teaching and learning computer programming: A case study. *Informatics in Education-An International Journal*, (Vol 7\_2), 181-210.
87. [Had09] Hadjerrouit, S. (2009). Teaching and learning school informatics: a concept-based pedagogical approach. *Informatics in Education-An International Journal*, (Vol 8\_2), 227-250.
88. [Har02] Harsu, M. (2002). *A survey on domain engineering*. Tampere University of Technology.
89. [HK09] Han, P., & Kramer, B. J. (2009, February). Generating interactive learning objects from configurable samples. In *Mobile, Hybrid, and On-line Learning, 2009. ELMML'09. International Conference on* (pp. 1-6). IEEE.
90. [HLR11] Hazzan, O., Lapidot, T., & Ragonis, N. (2011). *Guide to Teaching Computer Science: An Activity-Based Approach*. Springer.
91. [HP05] Huddleston, J., & Pike, J. (2005). Learning object reuse-a four tier model.
92. [HS12] Hamada, M., & Sato, S. (2012). A Learning System for a Computational Science Related Topic. *Procedia Computer Science*, 9, 1763-1772.
93. [HT08] Hartmann, H., & Trew, T. (2008, September). Using feature diagrams with context variability to model multiple product lines for software supply chains. In *Software Product Line Conference, 2008. SPLC'08. 12th International* (pp. 12-21). IEEE.
94. [IIT+12] Informatikos, informacinių technologijų ugdymo nuo 2015 metų gairės. (2012). [http://www.upc.smm.lt/ugdymas/dokumentai/svarstomi/it/Informatikos\\_ir\\_IT\\_ugdymo\\_nuo\\_2015\\_m.\\_gaires\\_2012-05-21.pdf](http://www.upc.smm.lt/ugdymas/dokumentai/svarstomi/it/Informatikos_ir_IT_ugdymo_nuo_2015_m._gaires_2012-05-21.pdf), žiūrėta 2013-10-26
95. [YS00] Yalçinkaya, Y., & Steihaug, T. (2000, January). An analytical model for a class of architectures under master-slave paradigm. In *High Performance Computing and Networking* (pp. 601-604). Springer Berlin Heidelberg.
96. [JB07] Jones, R., & Boyle, T. (2007). Learning object patterns for programming. *Interdisciplinary Journal of E-Learning and Learning Objects*, 3(1), 19-28.

97. [JCS09] Jiau, H. C., Chen, J. C., & Ssu, K. F. (2009). Enhancing self-motivation in learning programming using game-based simulation and metrics. *Education, IEEE Transactions on*, 52(4), 555-562.
98. [JGK+07] Jovanović, J., Gašević, D., Knight, C., & Richards, G. (2007). Ontologies for Effective Use of Context in e-Learning Settings. *Journal of Educational Technology & Society*, 10(3).
99. [KCH+90] Kang, K. C., Cohen, S. G., Hess, J. A., Novak, W. E., & Peterson, A. S. (1990). *Feature-oriented domain analysis (FODA) feasibility study* (No. CMU/SEI-90-TR-21). CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST.
100. [KD09] Kurilovas, E., & Dagienė, V. (2009). Multiple criteria comparative evaluation of e-Learning systems and components. *Informatica*, 20(4), 499-518.
101. [KGR06] Knight, C., Gašević, D., & Richards, G. (2006). An Ontology-Based Framework for Bridging Learning Design and Learning Content. *Journal of Educational Technology & Society*, 9(1).
102. [KH08] Keppens, J., & Hay, D. (2008). Concept map assessment for teaching computer programming. *Computer Science Education*, 18(1), 31-42.
103. [KJ09] Krueger, C.W., & Jackson, K. (2009). Requirements engineering for systems and software product lines. *Technical report*, IBM Corporation.
104. [KK07] Kay, R. H., & Knaack, L. (2007). Evaluating the learning in learning objects. *Open Learning*, 22(1), 5-28.
105. [KK08] Kay, R. H., & Knaack, L. (2008). A multi-component model for assessing learning objects: The learning object evaluation metric (LOEM). *Australasian Journal of Educational Technology*, 24(5), 574-591.
106. [KM09] Koehler, M., & Mishra, P. (2009). What is technological pedagogical content knowledge (TPACK)?. *Contemporary Issues in Technology and Teacher Education*, 9(1), 60-70.
107. [KP05] Kelleher, C., & Pausch, R. (2005). Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Computing Surveys (CSUR)*, 37(2), 83-137.
108. [KPN08] Kasurinen, J., Purmonen, M., & Nikula, U. (2008). A study of visualization in introductory programming. In *Proc. 20th annual Meeting of Psychology of Programming Interest Group, Lancaster, UK*.
109. [Krä05] Krämer, B. J. (2005). Reusable Learning Objects: Let's give it another trial. *Forschungsbericht Fern Universität, Hagen*.
110. [KS07] Knobelsdorf, M., & Schulte, C. (2007, November). Computer science in context: pathways to computer science. In *Proceedings of the Seventh Baltic Sea Conference on Computing Education Research-Volume 88* (pp. 65-76). Australian Computer Society, Inc.
111. [Kub09] Kubiliūnas, R. (2009). Lanksčiai pritaikomų mokymosi objektų formavimo metodas ir jo tyrimas. Kauno technologijos universitetas, daktaro disertacija.
112. [Kub12] Kubilinskienė, S. (2012). Išplėstas skaitmeninių mokymosi išteklių metaduomenų modelis. Vilniaus universitetas, daktaro disertacija.
113. [KVI+11] Klačnja-Miličević, A., Vesin, B., Ivanović, M., & Budimac, Z. (2011). E-Learning personalization based on hybrid recommendation strategy and learning style identification. *Computers & Education*, 56(3), 885-899.



- 114.[Lap13] Laplante, P. A. (2013). *Requirements engineering for software and systems*. CRC Press.
115. [LBM+04] Leeder, D., Boyle, T., Morales, R., Wharrad, H., & Garrud, P. (2004). To boldly GLO-towards the next generation of Learning Objects. In *World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education* (Vol. 2004, No. 1, pp. 28-33).
- 116.[LC06] Laforcade, P., & Choquet, C. (2006, July). Next Step for Educational Modeling Languages: The Model Driven Engineering and Reengineering Approach. In *ICALT* (pp. 745-747).
- 117.[LCW+09] Liu, L., Chen, H., Wang, H., & Zhao, C. (2009, December). Construction of a student model in contextually aware pervasive learning. In *Pervasive Computing (JCPC), 2009 Joint Conferences on* (pp. 511-514). IEEE.
- 118.[Lea08] Leavens, G. T. (2008). Use concurrent programming models to motivate teaching of programming languages. *ACM Sigplan Notices*, 43(11), 93-98.
- 119.[LFM08] Littlejohn, A., Falconer, I., & McGill, L. (2008). Characterising effective eLearning resources. *Computers & Education*, 50(3), 757-771.
- 120.[LH10] Liu, G. Z., & Hwang, G. J. (2010). A key step to understanding paradigm shifts in e-learning: towards context-aware ubiquitous learning. *British Journal of Educational Technology*, 41(2), E1-E9.
- 121.[Lib05] Liber, O. (2005). Learning objects: conditions for viability. *Journal of Computer Assisted Learning*, 21(5), 366-373.
- 122.[LY11] Lau, W. W., & Yuen, A. H. (2011). Modelling programming performance: Beyond the influence of learner characteristics. *Computers & Education*, 57(1), 1202-1213.
- 123.[LYW05] Lee, M. C., Ye, D. Y., & Wang, T. I. (2005, July). Java learning object ontology. In *Advanced Learning Technologies, 2005. ICALT 2005. Fifth IEEE International Conference on* (pp. 538-542). IEEE.
- 124.[LLY10] Law, K. M., Lee, V., & Yu, Y. T. (2010). Learning motivation in e-learning facilitated computer programming courses. *Computers & Education*, 55(1), 218-228.
- 125.[LN07] Leacock, T. L., & Nesbit, J. C. (2007). A Framework for Evaluating the Quality of Multimedia Learning Resources. *Journal of Educational Technology & Society*, 10(2).
126. [Lom08] Lombardozzi, C. (2008). Learning environment design. *E-learning Guild's Leading Solutions e-magazine*.
- 127.[LTSC02] Learning Technology Standards Committee. (2002). IEEE standard for learning object metadata. *IEEE Standard*, 1484(1), 2007-04.
- 128.[LUM07] Lung, C. H., Urban, J. E., & Mackulak, G. T. (2007). Analogy-based domain analysis approach to software reuse. *Requirements Engineering*, 12(1), 1-22.
- 129.[MAR08] Mercado, C. A. A., Andrade, E. L. M., & Reynoso, J. M. G. (2008). The Effect of Learning Objects on a C++ Programming Lesson. *Proceedings of the 19th Annual International Information Management Association, San Diego, CA, October*, 13-15.
- 130.[Mat06] Matthiasdottir, A. (2006). Usefulness of Learning Objects in Computer Science Learning. In *Proceedings of Codewitz Open Conference Methods, Materials and Tools for Programming Education, Tampere, Finland*.

- 131.[MB05] van Merriënboer, J. J., & Boot, E. (2005). A holistic pedagogical view of learning objects: Future directions for reuse. *Innovations in instructional technology*, 43-64.
- 132.[McG04] McGreal, R. (Ed.). (2004). *Online education using learning objects*. Psychology Press.
- 133.[Mey06] Meyer, B. (2006). Testable, reusable units of cognition. *IEEE Computer*, 39(4), 20-24.
- 134.[Men09] Mendonça, M. (2009). *Efficient reasoning techniques for large scale feature models* (Doctoral dissertation, University of Waterloo).
- 135.[MFB+12] Muller, P. A., Fondement, F., Baudry, B., & Combemale, B. (2012). Modeling modeling modeling. *Software & Systems Modeling*, 11(3), 347-359.
- 136.[MHC12] Matthews, R., Hin, H. S., & Choo, K. A. (2012, December). Merits and pitfalls of programming learning objects: a pilot study. In *Proceedings of the 10th International Conference on Advances in Mobile Computing & Multimedia* (pp. 293-296). ACM.
- 137.[MHZ+07] Moore, P., Hu, B., Zhu, X., Campbell, W., & Ratcliffe, M. (2007, November). A survey of context modeling for pervasive cooperative learning. In *First IEEE International Symposium on Information Technologies and Applications in Education, ISITAE'07* (pp. K5-1). National Institute of Standards and Technology.
- 138.[Mil56] Miller, G. A. (1956). The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychological review*, 63(2), 81.
- 139.[MJ10] Man, H., & Jin, Q. (2010, April). Putting adaptive granularity and rich context into learning objects. In *Information Technology Based Higher Education and Training (ITHET), 2010 9th International Conference on* (pp. 140-145). IEEE.
- 140.[MKS10] Mbendera, A. J., Kanjo, C., & Sun, L. (2010, February). Towards development of personalised knowledge construction model for e-learning. In *Mobile, Hybrid, and On-Line Learning, 2010. ELML'10. Second International Conference on* (pp. 29-35). IEEE.
- 141.[MLB05] Morales, R., Leeder, D., & Boyle, T. (2005, June). A case in the design of generative learning objects (GLOs): applied statistical methods. In *World Conference on Educational Multimedia, Hypermedia and Telecommunications*(Vol. 2005, No. 1, pp. 2091-2097).
- 142.[MM03] Miller, J., & Mukerji, J. (2003). MDA Guide Version 1.0. 1. *Object Management Group*, 234, 51.
- 143.[MP10] Mahmoud, Q. H., & Popowicz, P. (2010, October). A mobile application development approach to teaching introductory programming. In *Frontiers in Education Conference (FIE), 2010 IEEE* (pp. T4F-1). IEEE.
- 144.[MR02] Milne, I., & Rowe, G. (2002). Difficulties in learning and teaching programming—views of students and tutors. *Education and Information technologies*, 7(1), 55-66.
- 145.[MT12] Mtebe, J., & Twaakyondo, H. (2012). Are Animations Effective Tools for Teaching Computer Science Courses in Developing Countries?. *International Journal of Digital Information and Wireless Communications (IJDIWC)*, 2(2), 202-207.
- 146.[MV07] Mierlus-Mazilu, I., Vaduva, M. A. (2007, April 12-14). Learning objects for programming. In *ICTA '07* (pp. 167-172). Hammamet, Tunisia.

- 147.[MVS+10] Martin, S., Vallance, M., van Schaik, P., & Wiz, C. (2010). Learning spaces, tasks and metrics for effective communication in Second Life within the context of programming LEGO NXT Mindstorms robots: towards a framework for design and implementation. *Journal of Virtual Worlds Research*, 3(1).
- 148.[Nas05] Nash, S. (2005). Learning objects, learning object repositories, and learning theory: Preliminary best practices for online courses. *Interdisciplinary Journal of E-Learning and Learning Objects*, 1(1), 217-228.
- 149.[Nat12] Nath, J. (2012). E-learning methodologies and its trends in modern information technology. *Journal of Global Research in Computer Science*, 3(4), 48-52.
- 150.[NYH09] Noor, S. F. M., Yusof, N., & Hashim, S. Z. M. (2009, November). A Metrics Suite for Measuring Reusability of Learning Objects. In *Intelligent Systems Design and Applications, 2009. ISDA'09. Ninth International Conference on* (pp. 961-963). IEEE.
- 151.[NJN+07] Nilsson, M., Johnston, P., Naeve, A., & Powell, A. (2007). The future of learning object metadata interoperability. *Learning Objects: Standards, Metadata, Repositories, and LCMS*, 255-313.
- 152.[NR12] Nedungadi, P., & Raman, R. (2012). A new approach to personalization: integrating e-learning and m-learning. *Educational Technology Research and Development*, 60(4), 659-678.
- 153.[NS09] Narasimhamurthy, U., & Al Shawkani, K. (2009, August). Teaching of programming languages: An introduction to dynamic learning objects. In *Technology for Education, 2009. T4E'09. International Workshop on* (pp. 114-115). IEEE.
- 154.[NSP+12] Nikolopoulos, G., Solomou, G., Pierrakeas, C., & Kameas, A. (2012, September). Modeling the characteristics of a learning object for use within e-learning applications. In *Proceedings of the Fifth Balkan Conference in Informatics* (pp. 112-117). ACM.
- 155.[Old08] Oldfield, J. D. (2008). An implementation of the generative learning object model in accounting. *Proceedings of ASCILITE, Melbourne*.
- 156.[Par03] Paris, M. (2003). Reuse in practice: Learning objects and software development. In *Interact, Integrate, Impact: Proceedings of the 20th Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education (ASCILITE)*.
- 157.[Pea10] Pears, A. N. (2010, October). Enhancing student engagement in an introductory programming course. In *40th Frontiers in Education Conference, ser. Proceedings of the Frontiers in Education Conference* (No. 40).
- 158.[PLL+06] Paquette, G., Léonard, M., Lundgren-Cayrol, K., Mihaila, S., & Gareau, D. (2006). Learning design based on graphical knowledge-modeling. *Journal of Educational technology and Society*, 97-112.
- 159.[Pol03] Polsani, P. R. (2006). Use and abuse of reusable learning objects. *Journal of Digital information*, 3(4).
- 160.[PR06] Parsons, D., & Ryu, H. (2006, April). A framework for assessing the quality of mobile learning. In *Proceedings of the International Conference for Process Improvement, Research and Education* (pp. 17-27).
- 161.[PS04] Pitkanen, S. H., & Silander, P. (2004, August). Criteria for pedagogical reusability of learning objects enabling adaptation and individualised learning processes. In *Advanced Learning Technologies, 2004. Proceedings. IEEE International Conference on* (pp. 246-250). IEEE.

- 162.[PSM+07] Pears, A., Seidman, S., Malmi, L., Mannila, L., Adams, E., Bennedsen, J., ... & Paterson, J. (2007, December). A survey of literature on the teaching of introductory programming. In *ACM SIGCSE Bulletin* (Vol. 39, No. 4, pp. 204-223). ACM.
- 163.[Qui07] Quinton, S. R. (2007). Contextualisation of learning objects to derive meaning. *Learning objects: Theory, praxis, issues, and trends*, 113-79.
- 164.[RAH+09] Ramadhania, M. A., Aminah, S., Hidayanto, A. N., & Krisnadhi, A. A. (2009). Design and Implementation of Learning Object Ontology for e-Learning Personalization. In *International Conference on Advanced Computer Science and Information System* (pp. 428-433).
- 165.[Red03] Redeker, G. H. (2003, July). An educational taxonomy for learning objects. In *Advanced Learning Technologies, IEEE International Conference on* (pp. 250-250). IEEE Computer Society.
- 166.[RFC90] Reid, J. A., Forrestal, P., & Cook, J. (1990). *Small group learning in the classroom*. Heinemann.
- 167.[Rie03] Riebisch, M. (2003). Towards a more precise definition of feature models. *Modelling Variability for Object-Oriented Product Lines*, 64-76.
- 168.[RM04] Rodríguez-Artacho, M., & Maillo, M. F. V. (2004). Modeling educational content: the cognitive approach of the PALO language. *Educational Technology & Society*, 7(3), 124-137.
- 169.[Rob07] RobotC – Improved Movement. (2007). Robotics Academy.
- 170.[Röß10] Rößling, G. (2010). A family of tools for supporting the learning of programming. *algorithms*, 3(2), 168-182.
- 171.[RRR03] Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer Science Education*, 13(2), 137-172.
- 172.[RSG10] Roy, D., Sarkar, S., & Ghose, S. (2010). A Comparative Study of Learning Object Metadata, Learning Material Repositories, Metadata Annotation & an Automatic Metadata Annotation Tool. *Advances in Semantic Computing*, 2.
- 173.[Rup09] Rupšienė, I. (2009). Generatyvinių mokymosi objektų kūrimo metodai, pagrįsti aukšto lygmens abstrakcijomis. Kauno technologijos universitetas, daktaro disertacija.
- 174.[RW07] Reiser, M. O., & Weber, M. (2007). Multi-level feature trees. *Requirements Engineering*, 12(2), 57-75.
- 175.[SAA+07] Silveira, I. F., Araujo Jr, C. F., Amaral, L. H., Oliveira, I. C. A., Schimiguel, J., Ledón, M. F. P., & Ferreira, M. A. G. V. (2007). Granularity and reusability of learning objects. *Learning objects and instructional design*, 1, 139-170.
- 176.[SB08] Schmohl, R., & Baumgarten, U. (2008, March). Context-aware computing: a survey preparing a generalized approach. In *Proc. of the Int. MultiConference of Engineers and Computer Scientists*.
- 177.[SBD09] Stuiškys, V., Brauklyte, I., & Damasevicius, R. (2009). How to integrate generative learning objects into teaching and learning processes.
- 178.[Sch02] Schulte, C. (2002). Towards a pedagogical framework for teaching programming and object-oriented modelling in secondary education. *Proceedings of SECIII 2002*, 22-26.
- 179.[Sch06] Schmidt, D. C. (2006). Model-driven engineering. *COMPUTER-IEEE COMPUTER SOCIETY-*, 39(2), 25.

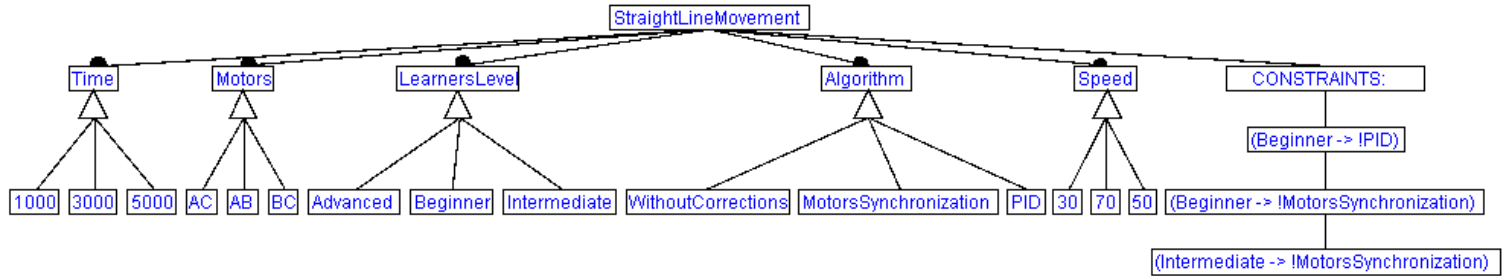
- 180.[SDB+08] Stuikys, V., Damasevicius, R., Brauklyte, I., & Limanauskiene, V. (2008, June). Exploration of learning object ontologies using feature diagrams. In *World Conference on Educational Multimedia, Hypermedia and Telecommunications* (Vol. 2008, No. 1, pp. 2144-2154).
- 181.[SDN+04] Sinnema, M., Deelstra, S., Nijhuis, J., & Bosch, J. (2004). Covamof: A framework for modeling variability in software product families. In *Software Product Lines* (pp. 197-213). Springer Berlin Heidelberg.
- 182.[SDS11] Sanz-Rodriguez, J., Dodero, J. M., & Sanchez-Alonso, S. (2011). Metrics-based evaluation of learning object reusability. *Software Quality Journal*, 19(1), 121-140.
- 183.[SDT12] Sakarkar, G., Deshpande, S. P., & Thakare, V. M. (2012). Intelligent Online e-Learning Systems: A Comparative Study. *International Journal of Computer Applications*, 56.
- 184.[Sër13] Šerikoviėnė, S. (2013). Mokomųjų objektų daugkartinio panaudojamumo kokybės vertinimo metodų taikymo tyrimas. Vilniaus universitetas, daktaro disertacija.
- 185.[SFP+12] Stephen, M., Franklin, W., Patrick, O., Peter, A., & Elizabeth, A. (2012). Classifying Program Visualization Tools to Facilitate Informed Choices: Teaching and Learning Computer Programming. *International Journal of Computer Science and Telecommunications*, 3(2), 42-48.
- 186.[SG03] Sicilia, M. A., & Garcia, E. (2003). On the concepts of usability and reusability of learning objects. *The International Review of Research in Open and Distance Learning*, 4(2).
- 187.[SGG06] Safran, C., García Barrios, V. M., & Gütl, C. (2006, March). A Concept-based Context Modelling System for the Support of Teaching and Learning Activities. In *Society for Information Technology & Teacher Education International Conference* (Vol. 2006, No. 1, pp. 2395-2402).
- 188.[SGM10] Santos, Á., Gomes, A., & Mendes, A. J. (2010). Integrating new technologies and existing tools to promote programming learning. *Algorithms*, 3(2), 183-196.
- 189.[SH06] Sajaniemi, J., & Hu, C. (2006). *Teaching Programming: Going Beyond" objects First"*. University of Joensuu.
- 190.[SHD09] Shuhidan, S., Hamilton, M., & D'Souza, D. (2009, January). A taxonomic study of novice programming summative assessment. In *Proceedings of the Eleventh Australasian Conference on Computing Education-Volume 95* (pp. 147-156). Australian Computer Society, Inc.
- 191.[SHL+13] Schäfer, A., Holz, J., Leonhardt, T., Schroeder, U., Brauner, P., & Ziefle, M. (2013). From boring to scoring—a collaborative serious game for learning and practicing mathematical logic for computer science education. *Computer Science Education*, 23(2), 87-111.
- 192.[SHT+07] Schobbens, P. Y., Heymans, P., Trigaux, J. C., & Bontemps, Y. (2007). Generic semantics of feature diagrams. *Computer Networks*, 51(2), 456-479.
- 193.[Slo09] Slotkienė, A. (2009). Aktyviojo mokymosi objekto projektavimo metodas ir jo tyrimas. Kauno technologijos universitetas, daktaro disertacija.
- 194.[SMG11] Salleh, N., Mendes, E., & Grundy, J. (2011). Empirical studies of pair programming for CS/SE teaching in higher education: A systematic literature review. *Software Engineering, IEEE Transactions on*, 37(4), 509-525.

- 195.[SMS08] Starr, C. W., Manaris, B., & Stalvey, R. H. (2008, March). Bloom's taxonomy revisited: specifying assessable learning objectives in computer science. In *ACM SIGCSE Bulletin* (Vol. 40, No. 1, pp. 261-265). ACM.
- 196.[Sne96] Snelling, G. (1996). Reengineering of configurations based on mathematical concept analysis. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 5(2), 146-189.
- 197.[SP09] Sampson, D. G., & Papanikou, C. (2009, July). A framework for learning objects reusability within learning activities. In *Advanced Learning Technologies, 2009. ICALT 2009. Ninth IEEE International Conference on* (pp. 32-36). IEEE.
- 198.[SPJ+11] Saeli, M., Perrenet, J., Jochems, W. M., & Zwaneveld, B. (2011). Teaching programming in secondary school: a pedagogical content knowledge perspective. *Informatics in Education-An International Journal*, (Vol 10\_1), 73-88.
- 199.[SPJ+12] Saeli, M., Perrenet, J., Jochems, W. M., & Zwaneveld, B. (2012). Programming: Teachers and Pedagogical Content Knowledge in the Netherlands. *Informatics in Education-An International Journal*, (Vol11\_1), 81-114.
- 200.[SSH+09] Sheard, J., Simon, S., Hamilton, M., & Lönnberg, J. (2009, August). Analysis of research into the teaching and learning of programming. In *Proceedings of the fifth international workshop on Computing education research workshop* (pp. 93-104). ACM.
- 201.[Str06] Stracke, C. M. (2006). Interoperability and Quality Development in e-Learning. Overview and Reference Model for e-Learning Standards. *Proceedings of the Asia-Europe e-Learning Colloquy. e-ASEM, Seoul*.
- 202.[ŠB12] Štuikys, V., & Burbaitė, R. (2012). Two-stage generative learning objects. In *Information and Software Technologies* (pp. 332-347). Springer Berlin Heidelberg.
- 203.[ŠBB14] Štuikys, V., Bepalova, K., & Burbaitė, R. (2014). Refactoring of Heterogeneous Meta-Program into k-stage Meta-Program. *Information Technology And Control*, 43(1), 14-27.
- 204.[ŠBD13] Štuikys, V., Burbaitė, R., & Damaševičius, R. (2013). Teaching of Computer Science Topics Using Meta-Programming-Based GLOs and LEGO Robots. *Informatics in Education-An International Journal*, (Vol12\_1), 125-142.
- 205.[ŠD07] Štuikys, V., & Damaševičius, R. (2007). Towards knowledge-based generative learning objects. *Information technology and control*, 36(2), 202-212.
- 206.[ŠD08] Štuikys, V., & Damaševičius, R. (2008). Development of generative learning objects using feature diagrams and generative techniques. *Informatics in Education-An International Journal*, (Vol 7\_2), 277-288.
- 207.[ŠD13] Štuikys, V., & Damaševičius, R. (2012). *Meta-Programming and Model-Driven Meta-Program Development: Principles, Processes and Techniques* (Vol. 5). Springer.
- 208.[TBK09] Thum, T., Batory, D., & Kastner, C. (2009, May). Reasoning about edits to feature models. In *Software Engineering, 2009. ICSE 2009. IEEE 31st International Conference on* (pp. 254-264). IEEE.
- 209.[Tch11] Tchounikine, P. (2011). *Computer Science and Educational Software Design: A Resource for Multidisciplinary Work in Technology Enhanced Learning*. Springer.
- 210.[TD09] Tankeleviciene, L., & Damasevicius, R. (2009, July). Towards a conceptual model of learning context in e-learning. In *Advanced Learning Technologies, 2009. ICALT 2009. Ninth IEEE International Conference on* (pp. 645-646). IEEE.

- 211.[TLW+08] Thompson, E., Luxton-Reilly, A., Whalley, J. L., Hu, M., & Robbins, P. (2008, January). Bloom's taxonomy for CS assessment. In *Proceedings of the tenth conference on Australasian computing education-Volume 78* (pp. 155-161). Australian Computer Society, Inc.
- 212.[TMH+12] Tillmann, N., Moskal, M., de Halleux, J., Fahndrich, M., Bishop, J., Samuel, A., & Xie, T. (2012, July). The future of teaching programming is on mobile devices. In *Proceedings of the 17th ACM annual conference on Innovation and technology in computer science education* (pp. 156-161). ACM.
- 213.[Tol06] Toledo, S. (2006). Analysis of the NXT Bluetooth-Communication Protocol. *Technical paper.[Online]. Available: <http://www.tau.ac.il/~stoledo/lego/btperformance.html>*.
- 214.[Tou12] Touretzky, D. S. (2012, February). Seven big ideas in robotics, and how to teach them. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education* (pp. 39-44). ACM.
- 215.[TTT12] Tuparov, G., Tuparova, D., & Tsarnakova, A. (2012). Using interactive simulation-based learning objects in introductory course of programming. *Procedia-Social and Behavioral Sciences*, 46, 2276-2280.
- 216.[UV09] Urquiza-Fuentes, J., & Velázquez-Iturbide, J. Á. (2009). Pedagogical effectiveness of engagement levels—a survey of successful experiences. *Electronic Notes in Theoretical Computer Science*, 224, 169-178.
- 217.[VBH13] Vincenti, G., Braman, J., & Hilberg, J. S. (2013). Teaching introductory programming through reusable learning objects: a pilot study. *Journal of Computing Sciences in Colleges*, 28(3), 38-45.
- 218.[VD04] Verbert, K., & Duval, E. (2004). Towards a global architecture for learning objects: a comparative analysis of learning object content models. In *World Conference on Educational Multimedia, Hypermedia and Telecommunications*(Vol. 2004, No. 1, pp. 202-208).
- 219.[Wil00] Wiley DA (2000) Learning Object Design and Sequencing Theory (Ph.D. dissertation). Brigham Young University, Utah, USA.
- 220.[WKG04] Weigl, F., Kammerl, R., & Göstl, M. (2004). Context aware reuse of learning resources. In *World Conference on Educational Multimedia, Hypermedia and Telecommunications* (Vol. 2004, No. 1, pp. 2119-2126).
- 221.[Zha09] Zhang, J. (2009, July). A model on adaptive example recommendation for programming learning. In *Computer Science & Education, 2009. ICCSE'09. 4th International Conference on* (pp. 1689-1692). IEEE.
- 222.[ZLZ09] Zheng, Y., Li, L., & Zheng, F. (2009). Context-awareness Support for Content Recommendation in e-learning Environments. In *2009 International Conference on Information Management, Innovation Management and Industrial Engineering*(Vol. 3, pp. 514-517).
- 223.[Zuc06] Zuckerman, O. (2006). Historical overview and classification of traditional and digital learning objects.

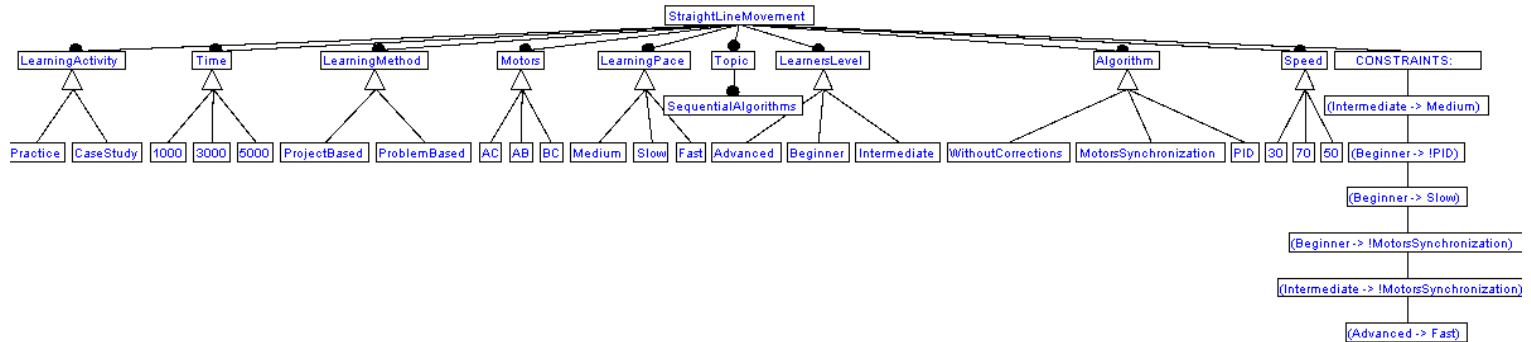
## PRIEDAI

### 1 priedas. IGMO „Roboto tiesiaegis judėjimas“ požymių modeliai



Legend: Algorithm: Proportional-Integral-Derivative – PID.

### 1 pav. IGMO turinio modelis papildytas požymiu Besimokančiojo lygis (su apribojimais)

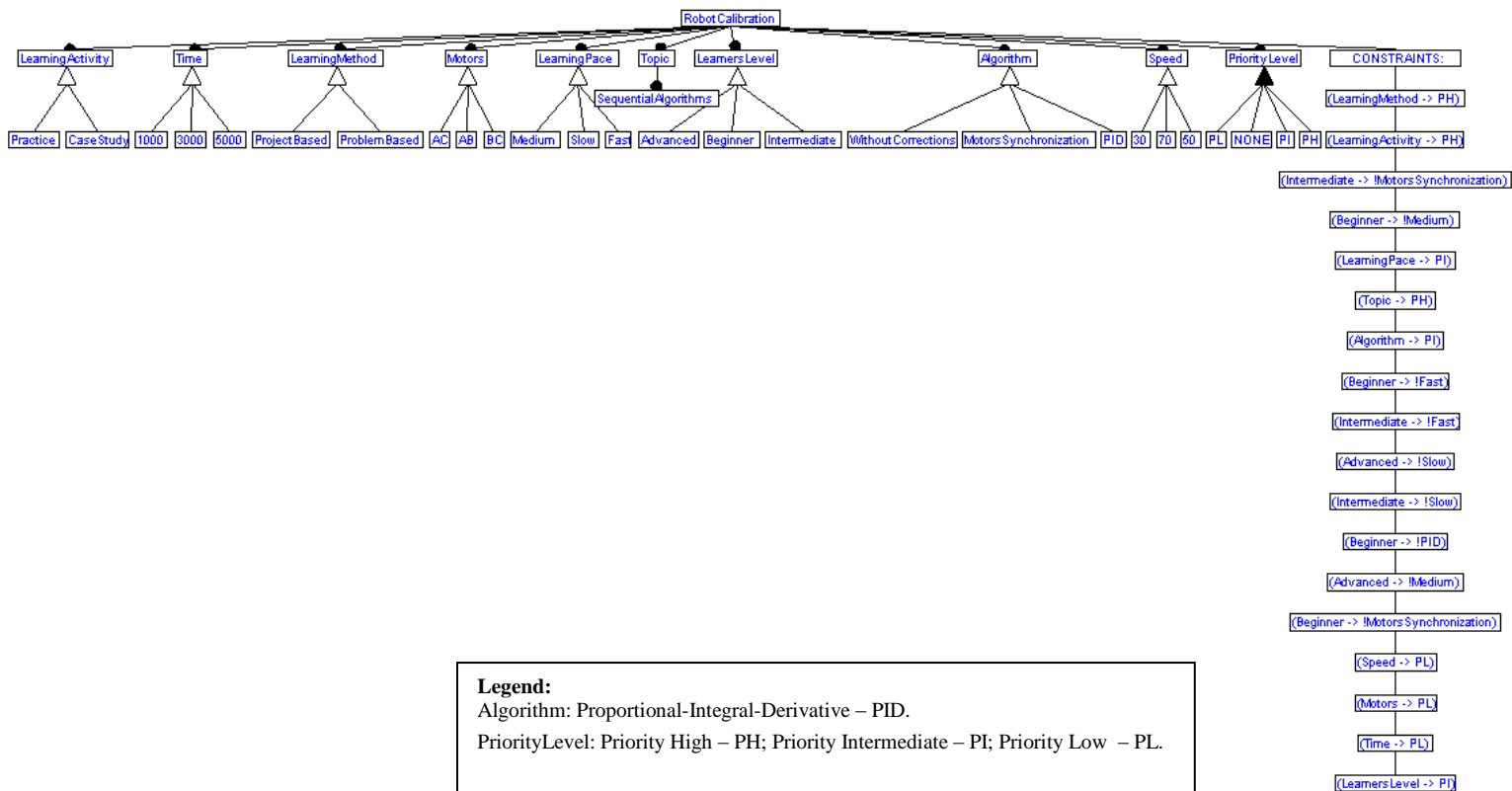


Legend: Algorithm: Proportional-Integral-Derivative – PID.

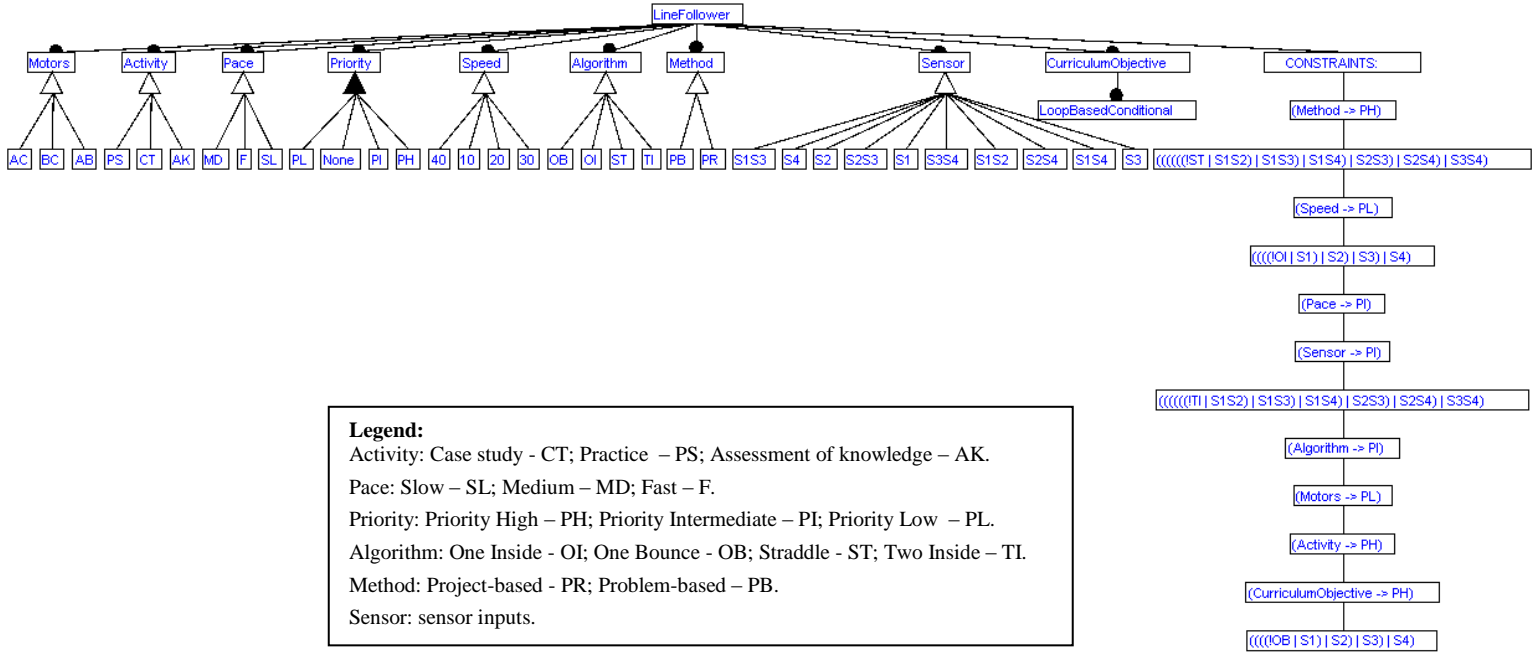
### 2 pav. Apibendrintas IGMO modelis su konteksto požymiais (su apribojimais)



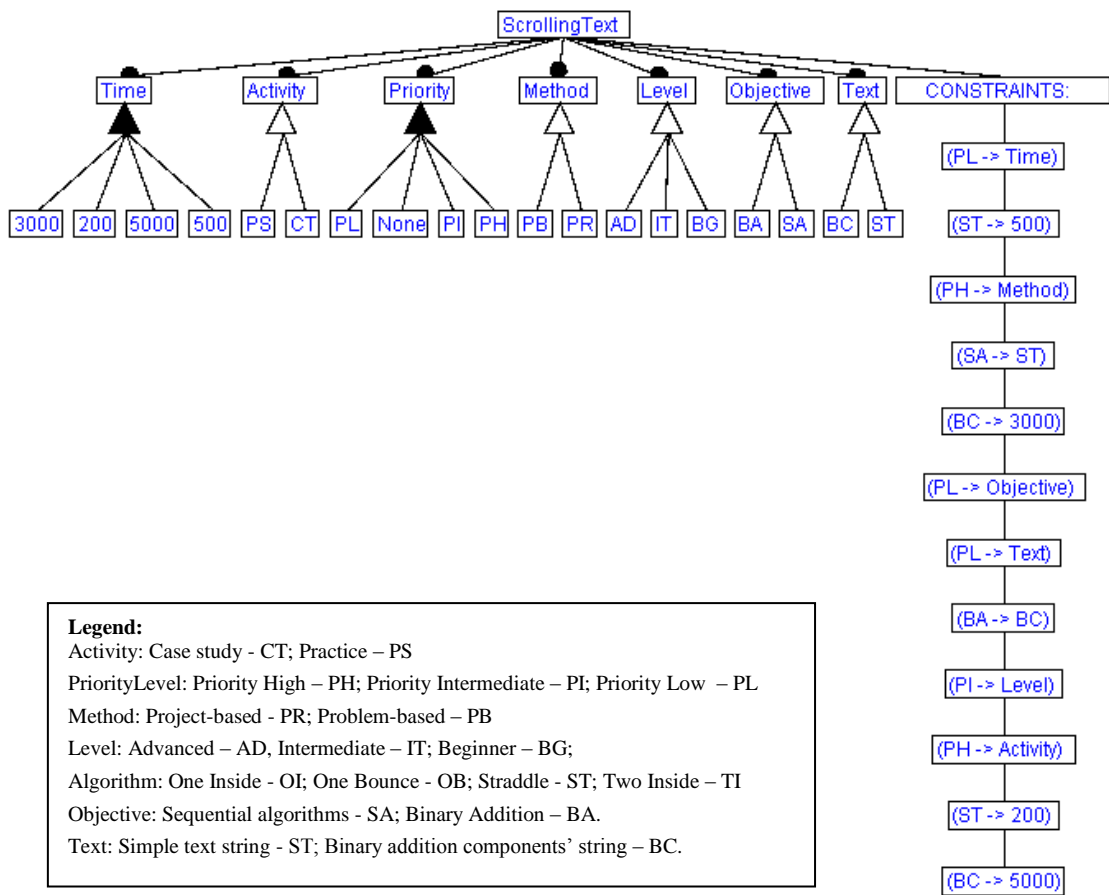
**2 priedas.** Sukurtų ir į mokymosi aplinkas integruotų išplėstinių generatyvinių mokymosi objektų požymių modeliai



**3 pav.** IGMO „Roboto kalibravimas“ požymių modelis



4 pav. IGMO „Linijos sekimas“ požymių modelis



5 pav. IGMO „Bėganti eilutė“ požymių modelis

- OrnamentsDesign
- Objective
  - LN
- Activity
  - [1..1]
    - CT
    - PS
- Level
  - [1..1]
    - BG
    - IT
    - AD
- Method
  - [1..1]
    - PR
    - PB
- Motor
  - [1..1]
    - AB
    - AC
    - BC
- Speed1
  - [1..1]
    - 10
    - 30
    - 50
- Speed2
  - [1..1]
    - 10
    - 30
    - 50

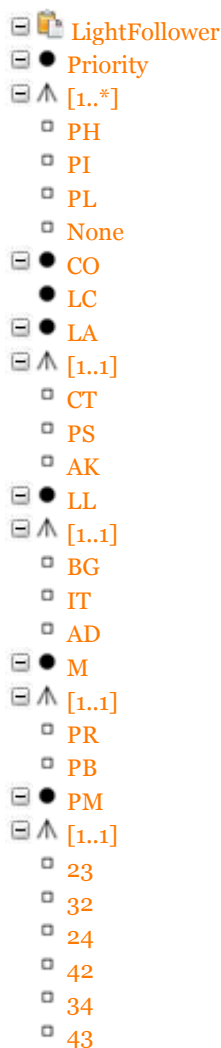
- Time
  - [1..1]
    - 1000
    - 3000
    - 5000
- P
  - [1..1]
    - 4
    - 5
    - 6
- D1
  - [1..1]
    - 10
    - 30
- D2
  - [1..1]
    - 10
    - 30
- T1
  - [1..1]
    - 200
    - 500
- P1
  - [1..\*]
    - 1
    - 2
    - 3
- Priority
  - [1..\*]
    - PH
    - PI
    - PL
    - None

### Cross-Tree Constraints

- ( $\neg$ Objective  $\vee$  PH)
- ( $\neg$ BG  $\vee$  1)
- ( $\neg$ AD  $\vee$  1)
- ( $\neg$ Speed1  $\vee$  PL)
- ( $\neg$ AD  $\vee$  2)
- ( $\neg$ Speed2  $\vee$  PL)
- ( $\neg$ Time  $\vee$  PL)
- ( $\neg$ IT  $\vee$  1)
- ( $\neg$ Method  $\vee$  PH)
- ( $\neg$ P  $\vee$  PL)
- ( $\neg$ IT  $\vee$  2)
- ( $\neg$ Motor  $\vee$  PL)
- ( $\neg$ D1  $\vee$  PL)
- ( $\neg$ BG  $\vee$   $\neg$ 3)
- ( $\neg$ Level  $\vee$  PI)
- ( $\neg$ D2  $\vee$  PL)
- ( $\neg$ T1  $\vee$  PL)
- ( $\neg$ AD  $\vee$  3)
- ( $\neg$ Activity  $\vee$  PH)
- ( $\neg$ P1  $\vee$  PL)
- ( $\neg$ BG  $\vee$   $\neg$ 2)

**Legend:**  
 Objective: Loops and nested loops – LN.  
 Activity: Case study – CT; Practice – PS.  
 Learner's level: Beginner-BG; Intermediate – IT; Advanced – AD.  
 Method: Project-based - PR; Problem-based – PB.  
 Motor – selected motor, Speed1, Speed2– *drawing velocity* of motors, T – robot's drawing time, P – number of ornament's parts, D1, D2 – *moving velocity* of motors, T1 – robot's moving time, P1 – number of ornaments.  
 PriorityLevel: Priority High – PH; Priority Intermediate – PI; Priority Low – PL.

6 pav. IGMO „Ornamentų kūrimas“ požymių modelis



### Cross-Tree Constraints

- (  $\neg$ PH  $\vee$  CO )
- (  $\neg$ PL  $\vee$  DL )
- (  $\neg$ PL  $\vee$  MD )
- (  $\neg$ PI  $\vee$  PM )
- (  $\neg$ PI  $\vee$  PS )
- (  $\neg$ PH  $\vee$  LA )
- (  $\neg$ PH  $\vee$  M )

#### Legend:

PriorityLevel: Priority High – PH; Priority Intermediate – PI; Priority Low – PL.  
 CO – curriculum objective; LC – Loop-based and conditional algorithms.  
 LA – learning activity; CT – Case study; PS – Practice; AK – Assessment of knowledge.  
 LL – learner’s level; BG – Beginner; IT – Intermediate; AD – Advanced.  
 M – learning method; PR – Project-based - PR; PB – Problem-based.  
 PM – output pins on Arduino to control left and right motors.  
 PS – left and right light sensors input pins on Arduino.  
 DL – difference of the lightning.  
 MD – movement direction (R – right; L – left; S – straight).

7 pav. IGMO „Šviesos sekimas“ požymių modelis



**Legend:**

PriorityLevel: Priority High – PH; Priority Intermediate – PI; Priority Low – PL.

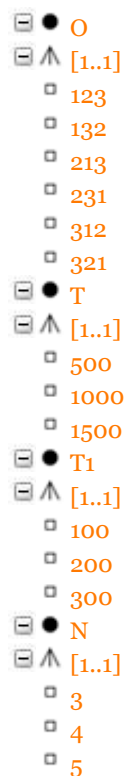
CO – curriculum objective; LN – Loops and nested loops.

LA – learning activity; CT – Case study; PS – Practice.

LL – learner’s level; BG – Beginner; IT – Intermediate; AD – Advanced.

M – learning method; PR – Project-based - PR; PB – Problem-based.

A – algorithm’s type (FT – fixed time control; DB – demand button control), I – input pin on Arduino for button (NO – without button), O – output pins on Arduino for red, yellow and green LEDs, T – duration of light phase in ms, T1 – duration of warning in ms, N – number of warning signals.



**Cross-Tree Constraints**

- (  $\neg$  PH  $\vee$  CO )
- (  $\neg$  PL  $\vee$  O )
- (  $\neg$  PL  $\vee$  T )
- (  $\neg$  PL  $\vee$  T1 )
- (  $\neg$  PI  $\vee$  A )
- (  $\neg$  PL  $\vee$  N )
- (  $\neg$  PI  $\vee$  LL )
- (  $\neg$  BG  $\vee$  FT )
- (  $\neg$  PH  $\vee$  M )
- (  $\neg$  IT  $\vee$  FT )
- (  $\neg$  IT  $\vee$  DB )
- (  $\neg$  AD  $\vee$  FT )
- (  $\neg$  AD  $\vee$  DB )
- (  $\neg$  PH  $\vee$  LA )

8 pav. IGMO „Šviesoforas“ požymių modelis