

# Adaptyvios programavimo mokymosi sistemos modelis

Jūratė Urbonienė

Matematikos ir informatikos instituto doktorantė  
Institute of Mathematics and Informatics, Doctoral student  
Akademijos g. 4, LT-08663 Vilnius  
El. paštas: jurate99@gmail.com

*Straipsnyje nagrinėjami programavimo mokymo ypatumai, apžvelgiama mokymosi stilių įvairovė bei mokymosi stiliaus įtaka mokymosi rezultatams. Plačiau aptariamas Herrmanno smegenų dominavimo instrumentas<sup>1</sup> (HBDI), kuris leidžia besimokančiuosius suskirstyti pagal jų smegenų pusrutulio išsivystymą. Remiantis programavimo mokymo patirtimi ir fundamentaliais programavimo mokymo srities mokslininkų darbais, nagrinėjamos programavimo mokymosi sunkumų priežastys (programavimo srities specifškumas, mokymosi būdų ir mokymosi metodų parinkimas), išskiriamos ir apibendrinamos esminės idėjos. Pagrindinis dėmesys skiriamas adaptyvios programavimo mokymosi sistemos (APMS) koncepciniam modeliui pristatyti. Modeliuojama APMS leis priderinti mokymosi objektus, išrenkant juos iš mokymosi objektų saugyklų, ir užtikrinti atitinkamų kompetencijų pasiekimo lygį atsižvelgiant į besimokančiojo mokymosi stilių pagal Herrmanno skirstymą bei ankstesnę jo mokymosi patirtį.*

**Pagrindiniai žodžiai:** programavimo mokymasis, adaptyvi mokymosi sistema, Herrmanno mokymosi stiliai.

## Įvadas

Programavimo mokymas, ypač mokymasis – aktuali problema tiek Lietuvoje, tiek ir užsienyje. Programavimo pradedama mokytis įvairaus amžiaus: nuo pradinių klasių iki aukštosios mokyklos ir dar vėliau. Tačiau pats programavimas besimokantiejiems nėra patrauklus ir tai sukelia papildomų mokymosi problemų. Jau kelis dešimtmečius ieškoma kuo efektyvesnių mokymo bei mokymosi priemonių, metodų ir būdų. Vis labiau pabrėžiama individualizuoto mokymosi svarba – kai atsižvelgiant į individualias besimokančiojo savybes parenkami mokymosi objektai, mokymosi veiklos, mokymosi scenarijus ir pan. Už-

tikrinti individualizavimą padeda ir adaptyvios mokymosi sistemos, kurios leidžia mokymosi metu priderinti besimokančiajam tinkamiausią mokymosi turinį.

Mokymosi adaptavimas individualiems besimokančiojo poreikiams tiek užsienyje, tiek Lietuvoje analizuojamas jau gana seniai. Yra atlikta nemažai tyrimų, kurių metu įvairiais metodais – atsižvelgiant į besimokančiojo savybes bei poreikius (Weber, Brusilovsky, 2001; Baziukaitė ir kt., 2008), analizuojant jo veiklas elektroninėje mokymosi aplinkoje (Preidys, Žilinskienė, 2012), nustatant jo mokymosi stilių (Franzoni, Assar, 2009) – parenkamas arba leidžiama pačiam besimokančiajam pasirinkti tinkamiausią mokymosi turinį arba mokymosi strategiją (Preidys, Žilinskienė, 2012),

<sup>1</sup> Herrmann Brain Dominance Instrument.

mokantis aplinkoje palaikoma adaptyvi navigacija (Weber, Brusilovsky, 2001; Paromythis, Loidl-Reisinger, 2003). Mokymosi sėkmė priklauso nuo to, kaip maksimaliai pasiekiami mokymosi tikslai – įgyjamos reikiamos kompetencijos (žinios bei įgūdžiai, kurie yra aprašomi mokymosi rezultatais), bei kokios emocijos patiriamos besimokant. Mokymosi sėkmę daug lemia mokymosi efektyvumas, kuris priklauso nuo noro mokytis ir mokėjimo mokytis. Pastaruošius lemia ir individualios besimokančiojo savybės, kurios skiriasi priklausomai nuo mokymosi stiliaus. Mokslininkų darbuose dažniausiai nagrinėjamas mokymosi adaptavimas atsižvelgiant į mokymosi stilius pagal Kolbo, Honey ir Mumfordo, Felderio ir Silvermano klasifikacijas (Hawk, Shah, 2007). Tačiau nepavyko aptikti Herrmanno stilių klasifikacijos nagrinėjimo ir taikymo mokymosi kontekste.

Straipsnio tikslas – supažindinti su adaptyvios programavimo mokymosi sistemos, kuri leidžia pasiekti tam tikrą kompetencijų lygį adaptuojant mokymosi objektus pagal

besimokančiojo mokymosi stilių (remiantis Herrmanno klasifikacija), koncepciniu modeliu.

Straipsnyje naudojami mokslinių straipsnių bei dokumentų apžvalgos ir sintezės tyrimo metodai, remiamasi lyginamosios analizės metodologiniais principais.

Pirmoje straipsnio dalyje aptariama mokymosi stilių įvairovė ir besimokančiųjų skirstymas pagal Herrmanno stilių klasifikaciją (1 skyrelis), mokymosi ypatumai (2 skyrelis) ir programavimo mokymuisi tinkami mokymosi objektai (3 skyrelis). Antra straipsnio dalis (4 skyrelis) skirta adaptyvios programavimo mokymosi sistemos koncepciniam modeliui pristatyti.

## 1. Mokymosi stilių klasifikavimas

### 1.1. Mokymosi stilių įvairovė

Pastaruoju metu vis dažniau kalbama apie mokymosi individualizavimą ir pritaikymą prie besimokančiojo asmeninių poreikių, jo pažinimo ar mokymosi stiliaus. Mokymosi stilių klasifikacijų yra daug ir įvairių (Hawk,

1 lentelė. *Mokymosi stilių nustatymo instrumentai*

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

Shah, 2007). Atsižvelgiant į skirtingus kriterijus ir naudojant skirtingus instrumentus (žr. 1 lentelę), jos leidžia įvairiai klasifikuoti besimokančiuosius, parinkti geresnius mokymosi rezultatus lemiančias mokymosi veiklas bei mokymosi turinį. Dauguma mokymosi stilių yra nustatoma naudojant įvairius savianalizės klausimynus.

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

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

Išvardytų gebėjimų išreikštumo lygis kiekvieno besimokančiojo yra skirtingas, nes kiekvienu atveju būdingos tam tikros savybės, informacija įsisavinama bei naudojama individualiu būdu. Nuo besimokančiojo stiliaus priklauso jo mokymosi veikla, naudojami mokymosi metodai, keliami klausimai, bendravimas, apibendrinimas. Čia taip pat labai svarbi protinė veikla, kuri lemia algoritminį bei loginį mąstymą.

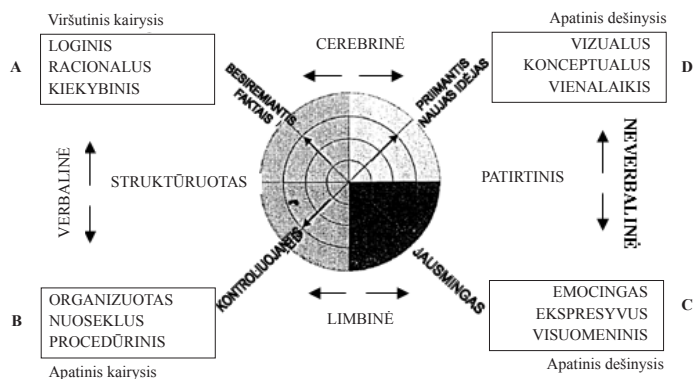
Herrmanno mokymosi stilių klasifikacija remiasi žmogaus smegenų pusrutulių darbu. Ši klasifikacija toliau ir aptariama.

## 1.2. Mokymosi stilių klasifikavimas naudojant Herrmanno smegenų dominavimo instrumentą

Visą smegenų mokymo ir mokymosi modelį 1970 metais sukūrė *Ned Herrmann*. Remiantis šiuo modeliu gali būti nustatomas asmens smegenų kiekvieno iš keturių kvadrantų (A, B, C ir D) išreikštumo laipsnis (1 pav.). Tam naudojamas specialus 120 klausimų testas bei analizuojami jo rezultatai.

Šis metodas leidžia besimokančiuosius suskirstyti pagal jų mąstymo pobūdį. Pagal smegenų fizinio funkcionavimo atliekant užduotis principus yra skiriamos keturios būsenos (kvadrantai) (žr. 2 lentelę). Tačiau gali būti taip, kad pakankamai ryškios būna du kvadrantus atitinkančios savybės arba žmogui gali būti būdingas vienodas visų keturių kvadrantų raiškos lygis.

Žinant besimokančiojo individualias mąstymo galimybes galima ieškoti individualių mokymo metodų bei priemonių arba leisti pasirinkti jam tinkamiausius. Tačiau



1 pav. Visas smegenų mokymo ir mokymosi modelis (pagal Herrmanną)

2 lentelė. *Herrmanno mokymosi stilių aprašymas*

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

ir pats programavimas priverčia atkreipti dėmesį į tam tikrus mokymosi ypatumus, kurie aptariami kitame skyrelyje.

## 2. Programavimo mokymosi ypatumai

Programavimas – sudėtingas intelektinis procesas, todėl norint juo sudominti besimokančiuosius, reikia pateikti jį paprastai, aiškiai, patraukliai. Tačiau, kad ir kaip patraukliai būtų pateikiama, vien žinių ar gerų pavyzdžių neužtenka – būtina pačiam dalyvauti procese, lavinti programavimo įgūdžius bei logiškai mąstyti. Pažinimo teorijų šalininkai bandė atsakyti į klausimą, kodėl tiek daug besimokančiųjų nesugeba išmokti programuoti? Jie nurodė šias priežastis:

- sunku suprasti programų tikslus ir jų ryšį su kompiuteriu;

- sunku suprasti specifinės programavimo kalbos sintaksę ir semantiką (Robins, Rountree, Rountree, 2003);
- įgyjamas neteisingas programavimo konstrukcijų suvokimas;
- nesugebama spęsti problemų (Jenkins, 2002; Gomes, Mendes, 2007);
- nesugebama skaityti ir suprasti programų tekstų (Lister, Adams ir kt., 2004; Mannila, 2007).

Giliau patyrinėjus programavimo mokymo problemas išskiriami penki komponentai, kurie lemia programavimo mokymo sunkumus (Dagienė, Urbonienė, 2010):

1) *Mokymo metodai*. Programavimo mokymas vis dar nėra personalizuotas, mokytojo naudojami metodai nederą su mokinių mokymosi stiliais, dažnai dinamių konceptų mokoma naudojant statinę medžiagą, mokytojas labiau susitelkęs į

programavimo kalbos ir jos sintaksės mokymą, o ne į uždavinių sprendimą taikant programavimo kalbą ir aplinką.

2) *Mokymosi metodų naudojimas*. Besimokantieji naudoja netinkamus mokymosi metodus ar metodologiją, pavyzdžiui, daugelis mano, jog gali išmokyti programuoti vien skaitydami programas, tačiau pagrindinė jų veikla turėtų būti nukreipta spręsti uždavinius. Kita tendencija – besimokantieji nepakankamai daug dirba savarankiškai, kad įgytų programavimo kompetencijos.

3) *Besimokančiųjų gebėjimai ir nuostatos*. Besimokantieji turi būti įgiję ar norėti įgyti įvairių su programų kūrimu susijusių gebėjimų: problemų supratimo, turimų žinių susiejimo su problema, uždavinio ir jo sprendimo apmąstymo (refleksijos), atkaklumo sprendžiant uždavinius, pagrindinių matematinių ir loginių žinių taikymo, konkrečių programavimo žinių taikymo (Gomes, Carmo ir kt., 2006). Pastebima, kad sunku gauti ne patį rezultatą, t. y. parašyti programą, bet pereiti patį kūrimo procesą. Daugelis pradedančiųjų netinkamai naudoja žinias rašydami žingsninę specifikaciją natūralia kalba, t. y. netinkamai transformuoja natūralios kalbos semantiką į programavimo kalbą (Bennedsen, 2008).

4) *Programavimo prigimtis*. Programavimas reikalauja aukšto abstrakcijos lygio, programavimo kalbų sintaksė yra gana sudėtinga.

5) *Psichologiniai motyvai*. Mokiniai neturi motyvacijos, paprastai jie pradeda mokytis programavimo sudėtingu jų gyvenimo tarpsniu, pavyzdžiui, paauglystėje (Jenkins, 2002).

Norint, kad programavimo mokymasis taptų patrauklus ir paprastas, reikia mokytojo ir mokinio pastangų, palankaus nusitei-

kimo. Taip pat svarbu, kad būtų pasirinktos tinkamos mokymosi priemonės, kurios skatintų tobulinti mokinio programavimo kompetenciją. Mokymosi sėkmę paprastai lemia pasirinkti mokymosi metodai. Kadangi mokantis programuoti būtina suprasti tos srities konceptus, išmanyti jų panaudojimą, studijuoti įvairius pavyzdžius, aktyviai dalyvauti sprendžiant panašias problemas, todėl programavimo mokymuisi gali būti naudojami šie mokymosi metodai:

- informacijos šaltinių (sąvokų, terminų, konceptų) studijavimas;
- pavyzdžių studijavimas;
- grafinės informacijos studijavimas;
- vaizdo medžiagos studijavimas;
- užduočių sprendimas;
- problemų sprendimas.

Mokymosi metodai yra glaudžiai susiję su mokymuisi naudojamais mokymosi objektais. Kitame skyrelyje bus trumpai aptariami mokymosi objektai, jų saugyklos ir programavimo mokymui tinkamų mokymosi objektų skirstymas pagal mokymosi stilių.

### **3. Programavimo mokymosi objektai**

Mokymosi objektas – tai bet koks skaitmeninis išteklius, kurį galima naudoti mokymui(-si) ir taikyti iš naujo kituose mokymo(-si) kontekstuose (Dagienė, Kurilovas, 2008). Tačiau mokymosi objektas yra ne tik pats objektas (paveikslas, tekstas ir pan.), bet ir su juo susiję mokymosi medžiagą aprašantys papildomi duomenys (metaduomenys) – objekto autorius, pavadinimas, paskirtis, kalba, tema ir kt. Pagal šiuos duomenis objektą galima rasti mokymosi objektų saugyklose (angl. *Learning Objects Repositories*). Laikui bėgant tokių

saugyklų internete vis daugėja. Naudojami metaduomenų standartai (*IEEE LOM, Dublin Core, ADL SCORM, ISO Metadata Standard MLR* ir pan.) ir specifikacijos (*IMS Learning Resource Meta-data Specification* ir kt.) leidžia vieningai aprašyti adaptavimo elementus – mokymosi išteklius, besimokančiuosius, veiklos architektūrą ir turinio paketus. Yra keletas standartizavimo ir specifikavimo institucijų (*IEEE, ADL, ISO*), kurios kuria metaduomenų standartus šiems elementams aprašyti. Standartai:

- padaro skaitmeninį turinį mažiau priklausomą nuo jam pateikti naudojamų servisų bei techninės įrangos;
- uztikrina vienodą ir aiškią prieigą prie tinklinių mokymosi išteklių ir paslaugų;
- padeda pratęsti kapitalo įsigijimo ir organizacinių pokyčių gyvavimo laiką;
- palengvina sistemos komponentų ir duomenų išteklių integraciją ir priežiūrą.

Programavimo mokymuisi yra tinkami šie mokymosi objektai: tekstas, garso įrašas, vaizdo įrašas, grafinė medžiaga, interaktyvus pavyzdys, mokomoji užduotis, atsiskaitomoji užduotis, testas.

Kaip jau buvo minėta (žr. 2 skyrių), taikant Herrmanno mokymosi stilių klasifikaciją išskiriami keturi skirtingi besimokančiųjų tipai, kurie atitinka kvadrantus (A, B, C ir D). Pagal kiekvieno stiliaus

atstovo savybes bei tinkamas mokymosi veiklas (žr. 2 lentelę) besimokantiesiems galima priskirti labiausiai jiems tinkamus programavimo mokymosi objektus (žr. 3 lentelę). Šis priskyrimas taip pat priklauso nuo kvadranto išreikštumo lygio. Jei besimokančiojo kelių kvadrantų savybės vienodai ryškios, jam gali būti tinkami tuos kvadrantus atitinkantys mokymosi objektai.

## 4. Adaptyvios programavimo mokymosi sistemos koncepcinis modelis

### 4.1. Adaptyvios sistemos samprata

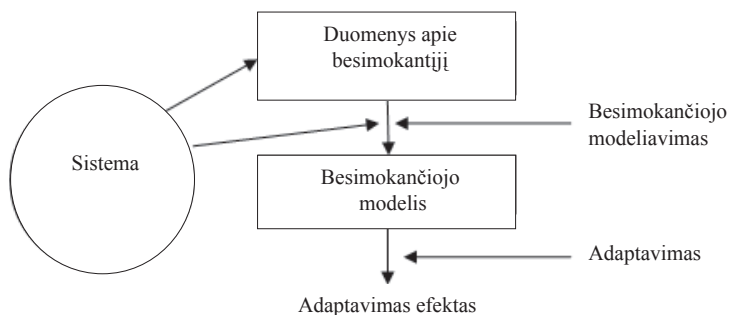
Mokymosi adaptavimas atsižvelgiant į besimokančiojo poreikius ir galimybes yra žinomas jau nuo senų laikų ir tai savo darbuose mini daugelis mokslininkų (Corno ir Show (1986), Federico ir kt. (1980), Glaser (1977), Reiser (1987), Tobias (1989), Wang ir Lindvall (1984)). Jau 1902 metais Dewey savo rašinyje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

Brusilovsky ir Maybury (2002) apibrėžia apibendrintą adaptyvios sistemos modelį (žr. 2 pav.).

Tokia sistema nuolatos stebi apie besimokantįjį kaupiamus bei saugomus duomenis ir

3 lentelė. *Mokymosi objektų ir mokymosi stiliaus atitikimas*

<i>Būseną (kvadrantas)</i>	<i>Tinkami mokymosi objektai</i>
Kvadrantas A	tekstas, interaktyvus pavyzdys, mokomoji užduotis, atsiskaitomoji užduotis, testas
Kvadrantas B	tekstas, interaktyvus pavyzdys, mokomoji užduotis, atsiskaitomoji užduotis, testas
Kvadrantas C	garso įrašas, interaktyvus pavyzdys, mokomoji užduotis, atsiskaitomoji užduotis, testas
Kvadrantas D	vaizdo įrašas, grafinė medžiaga, interaktyvus pavyzdys, mokomoji užduotis, atsiskaitomoji užduotis, testas



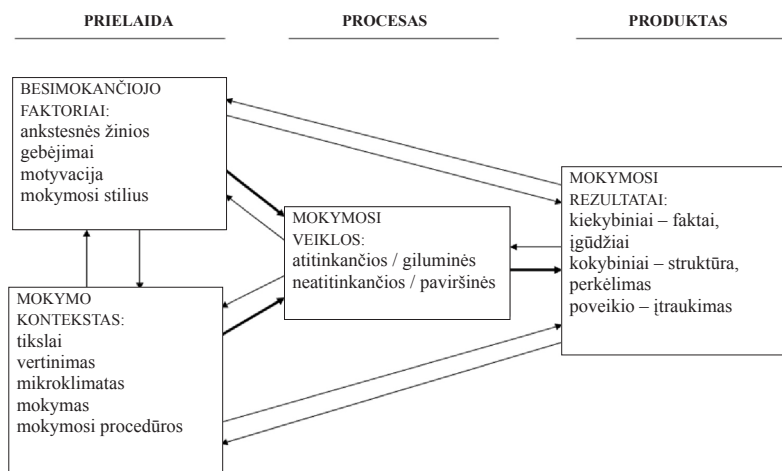
2 pav. *Adaptyvios mokymosi sistemos modelis* (Brusilovsky, Maybury, 2002)

priklausomai nuo aptinkamos informacijos bei atitinkamai realaus laiko sąlygoms aprašomų taisyklių reaguoja į esamą situaciją.

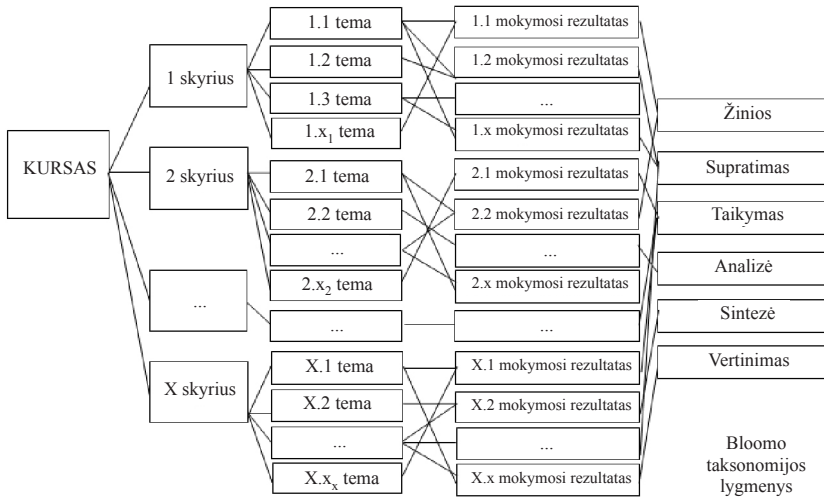
Adaptavimas elektroniniu būdu organizuotame mokymesi remiasi tiek adaptavimo procesu, tiek adaptuojamomis būsenomis. Išskiriami šie adaptavimo tipai: išankstinis (angl. *anticipatory*) arba reaguojantis (angl. *reactive*), savarankiškas (angl. *autonomous*) arba planuotas (angl. *planned*), bendras (angl. *generic*) arba rinktinis (angl. *selective*) (Moura, 2006). Mokymosi aplinka laikoma adaptyvia, jeigu ji sugeba: stebėti jos vartotojų veiksmus; interpretuoti šiuos veiksmus specifinės srities pagrindu; pagal interpretuojamus veiksmus nustatyti var-

totojo reikalavimus ir nuostatas taip, kad juos būtų galima pavaizduoti asocijuotame modelyje; paveikti leistinas žinias taip, kad atsižvelgiant į vartotoją ir temos sritį dinamiškai palengvėtų mokymosi procesas (Paramythis, Loidl-Reisinger, 2003). Pabrėžiama, jog sistema gali adaptuoti mokymo procesą kelias lygiais (Esmahi, 2009):

- Kurso turinio adaptacija (angl. *Course-Content Adaptation*): adaptyvus pristatymas įterpiant, pašalinant, rūšiuojant ir laikinai išjungiant fragmentus;
- Kurso navigacijos adaptacija (angl. *Course-Navigation Adaptation*): adaptacija palaikoma nuorodomis – nuorodos slepiamos, rūšiuojamos, uždraudžiamos



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



4 pav. Mokymosi kurso išdėstymo struktūrinė schema

ar pašalinamos, taip pat mokantis generuojamos naujos nuorodos;

- Mokymosi strategijos (angl. *Learning Strategy*): paskatomis grįstas mokymasis, atvejų analize grįstas mokymasis, probleminis mokymasis;
- Sąsajos (angl. *Interfaces*): palaikymas per sąsają – pateikiama besimokančiam labiausiai patinkanti sąsaja;
- Sąveika (angl. *Interaction*): ji turi būti intuityvi, grįsta besimokančiojo profilio informacija.

Pagal 3P mokymosi modelį (žr. 3 pav.), mokymosi procese gana svarbūs yra besimokančiojo informacija, mokymosi turinys, mokymosi veiklos ir darnus jų tarpusavio ryšys, kuris leidžia pasiekti pageidaujamą mokymosi rezultatą.

Kitas skyrelis skirtas aptarti programavimo kurso turiniui bei jo elementams.

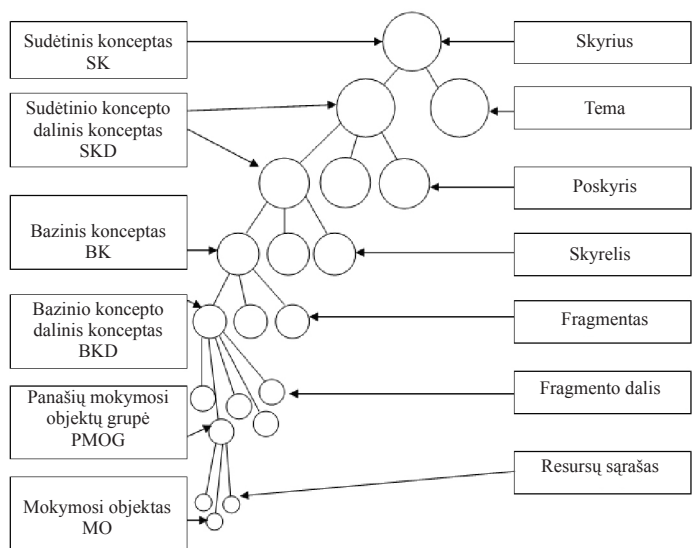
#### 4.2. Mokymosi turinio konstravimas

Siekiant išskirti pagrindinius išmokti reikiamus programavimo konceptus ir apibrėžti mokymosi turinį (dalyko žinias,

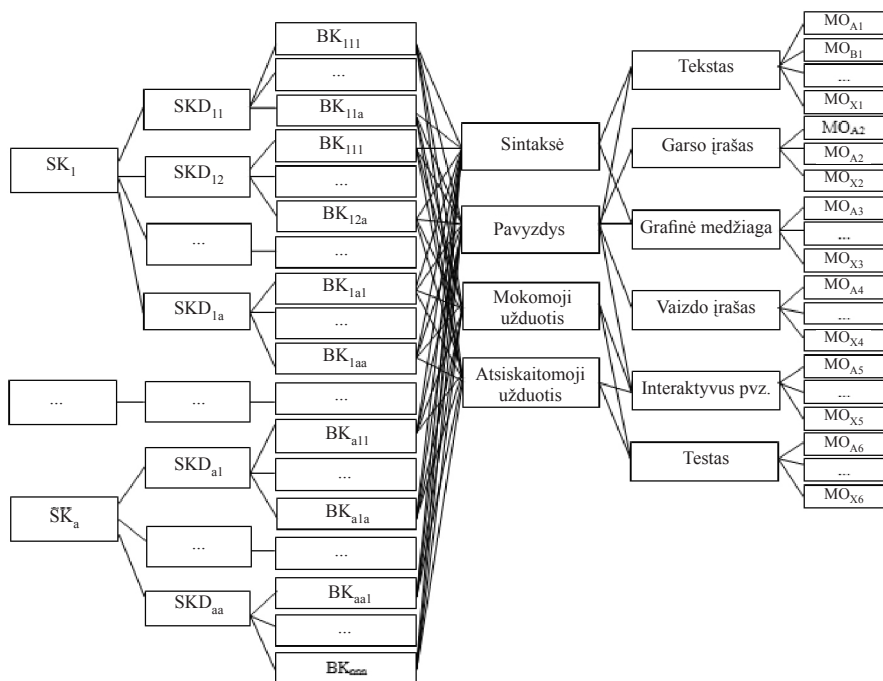
supratimą, gebėjimus, kuriuos tikimasi, kad besimokantysis įgis) remiamasi *Computer Science Curriculum 2008 (CS2008)* ir *Computer Science Curriculum 2013 (CS2013)* (parengtomis *Association for Computing Machinery, IEEE Computer Society; ACM/IEEE-CS, 2008; 2012; Pasha, Pasha, 2012*). Kadangi programavimui daugiau dėmesio skiriama CS2008<sup>2</sup>, ką pabrėžia ir autorių M. Pasha bei S. Pasha (2012) atlikta analizė, todėl daugiausia remiamasi šiuo dokumentu. Čia yra įvardytos pagrindinės kurso temos, kurios turėtų būti mokomos, bei šias temas atitinkantys mokymosi rezultatai. Taip pat atsižvelgiama į *CSTA K–12 Computer Science Standards 2011* (parengta *Computer Science Teachers Association, Association for Computing Machinery*). Kiekvieną mokymosi rezultatą galima susieti su šešiais Bloomo taksonomijos lygmenimis (Da-

<sup>2</sup> CS2008 kursai – Programavimo pagrindai (*Programming Fundamentals*), Algoritmai ir sudėtingumas (*Algorithms and Complexity*) ir Programavimo kalbos (*Programming Languages*); CS2013 kursai – Algoritmai ir sudėtingumas (*Algorithms and Complexity*) ir Programavimo kalbos (*Programming Languages*).





5 pav. Mokymosi turinio elementų išdėstymo struktūrinė schema



6 pav. Mokymosi elementų išdėstymo struktūrinė schema

gienė, Urbonienė, 2010). Taigi, mokymosi kursas gali būti dekomponuojamas atsižvelgiant į numatomus mokymosi rezultatus (žr. 4 pav.).

Nustačius mokymosi turinio, mokymosi rezultatų bei besimokančiojo pasiekiamo Bloomo taksonomijos lygmens sąsajas, svarbu apibrėžti ryšius tarp privalomų įvaldyti konceptų (žr. 5 pav.).

Konceptai (bazinis konceptas – BK) skaidomi į smulkesnes dalis (bazinio koncepto dalinis konceptas – BKD) bei grupuojami (sudėtinis konceptas – SK arba sudėtinio koncepto dalinis konceptas – SKD). Taip pat yra nustatomos konceptų atitinkančių mokymosi objektų grupių sąlygos (žr. 6 pav.).

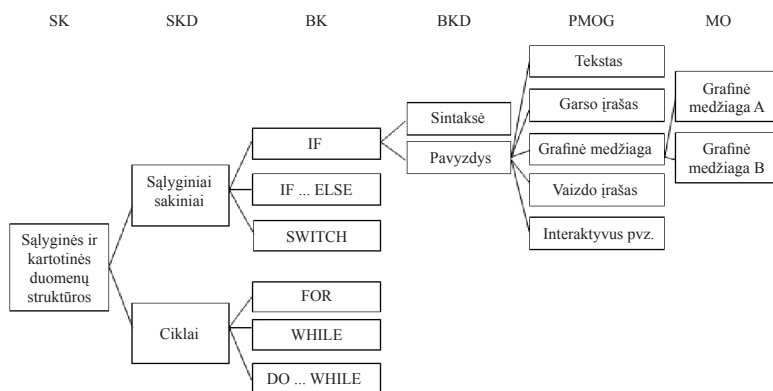
Kaip taikant šią schemą gali būti aprašomas bazinis konceptas „Ciklas For“, galima pamatyti 7 paveiksle.

Panašūs mokymosi objektai grupuojami atsižvelgiant į jų metaduomenis (žr. 8 pav.), o konkretūs objektai, priklausantys tam tikrai panašių objektų grupei, išrenkami atsižvelgiant į iš anksto aprašytas taisykles (žr. 9 pav.).

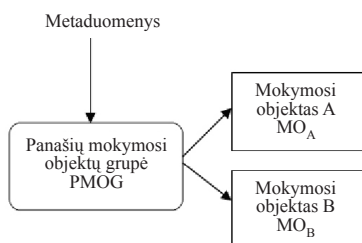
Toliau aprašoma adaptyvios programavimo mokymosi sistemos koncepcija.

### 4.3. Adaptyvios programavimo mokymosi sistemos koncepcija

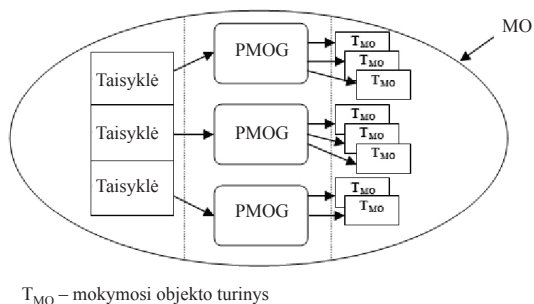
Sistemos esybių ryšių diagrama (žr. 10 pav.) nurodo esybių tarpusavio ryšius sistemoje.



7 pav. Mokymosi elementų išdėstymo pavyzdys

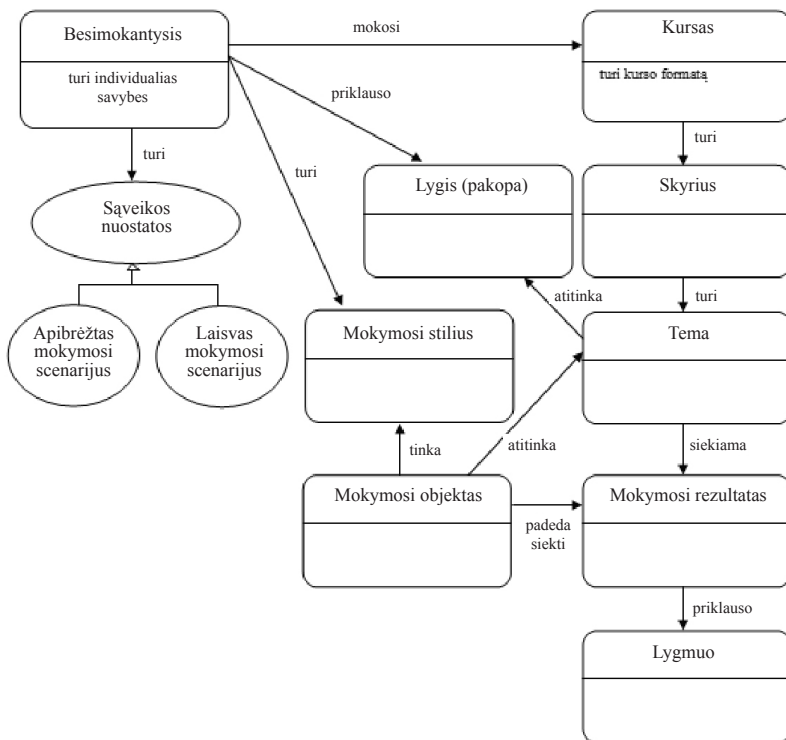


8 pav. Panašių mokymosi objektų grupės sudarymo struktūrinė schema

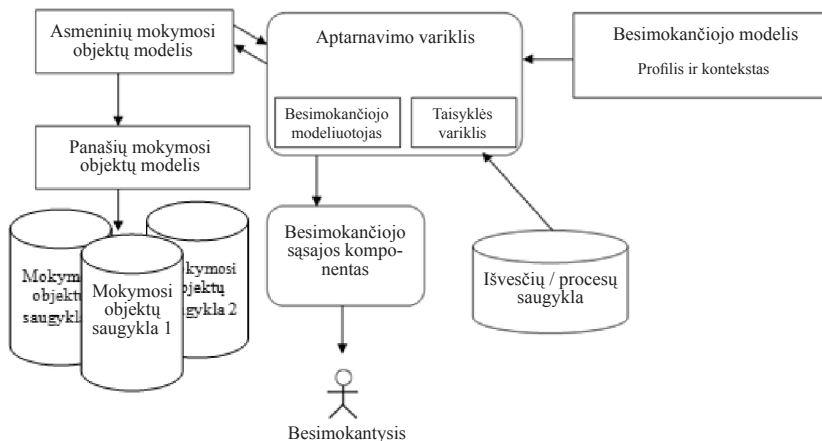


T<sub>MO</sub> – mokymosi objekto turinys

9 pav. Mokymosi objektų išrinkimo schema



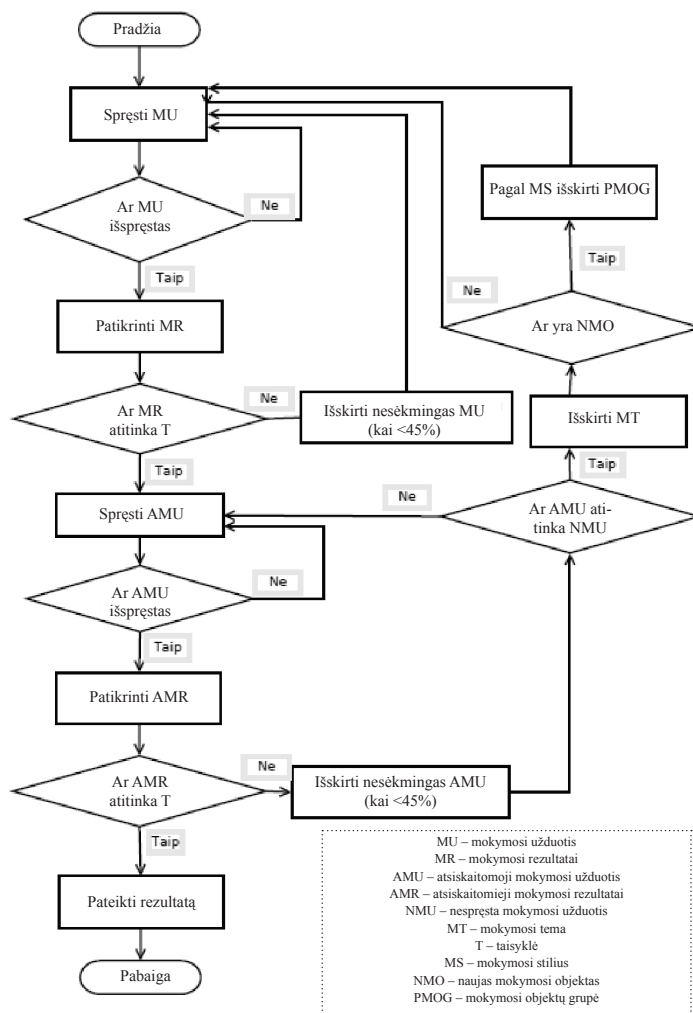
10 pav. Sistemos esybių ryšių diagrama



11 pav. Sistemos adaptavimo modelis

Taigi, besimokantysis turi individualias savybes bei jam būdingą mokymosi stilių, tačiau priklausydamas tam tikram lygiui (privalomam arba papildomam) jis mokosi atitinkamą formatą turintį mokymosi kursą. Kursas savo ruožtu turi skyrius, temas, kurios leidžia siekti numatytų atitinkamų mokymosi rezultatų. Mokymuisi naudojami mokymosi objektai turi sietis su besimokančiojo mokymosi stiliumi, numatomu mokymosi rezultatu bei mokomąja tema.

Sistemos adaptavimo modelyje (žr. 11 pav.) vaizduojamas mokymosi objektų išrinkimo bei adaptavimo konkrečiam besimokančiajam mechanizmas. Mokymosi objektai, kurie gali būti grupuojami pagal iš anksto apibrėžtas taisykles (pvz., atitinka tą patį mokymosi rezultatą, tą pačią temą, yra tos pačios formos), naudojant paieškos mechanizmus bei atsižvelgiant į individualias besimokančiojo sistemoje saugomas savybes (mokymosi stilių, mokymosi patirtį ir pan.), išrenkami iš mokymosi objektų saugyklos



12 pav. Mokymosi kurso sėkmės vertinimo algoritmo schema

bei per sąsajos komponentą pateikiami besimokančiajam.

Viena iš svarbių sistemos užduočių – padėti įvertinti besimokančiojo pasiektų mokymosi rezultatų lygį ir esant reikalui, jei mokymosi rezultatas yra nepatenkinamas, atsižvelgiant į ankstesnę mokymosi patirtį (peržiūrėtą/neperžiūrėtą mokymosi medžiagą ir sėkmingai / nesėkmingai įveiktą užduotį), pateikti tolesnį galimų atlikti veiksmų sąrašą. Sykiu sudaroma galimybė pačiam besimokančiajam rinktis sąrašo elementus, neatsižvelgiant į jų išdėstymo eiliškumą.

Mokydamasis asmuo spendžia mokymo- si užduotis, kurių rezultatai fiksuojami ir šie įrašai naudojami tolesnei mokymosi eigai generuoti. Sėkmingai įveikus mokomąsias užduotis, t. y. jei pasiekiami taisyklėmis aprašyti rezultatai, atliekamas mokymosi rezultatų vertinimas. Mokymosi rezultatams įvertinti spendžiamos juos atitinkančios atsiskaitomosios užduotys. Užduočių išspendžiamumo lygis taip pat aprašomas

taisyklėmis. Jei taisyklės netenkinamos, besimokantysis privalo grįžti ir pakartoti reikiamą medžiagą – pagal jo mokymosi stilių yra parenkami toje situacijoje tinkamiausi mokymosi objektai. Rezultatų vertinimo algoritmas vaizduojamas 12 paveiksle.

## Išvados

Išanalizavus programavimo mokymo problemas, besimokančiųjų mokymosi stilių įtaką mokymosi rezultatams bei nustačius programavimo mokymui tinkamus mokymosi objektus, buvo sukurtas adaptyvios programavimo mokymo sistemos koncepcinis modelis, kuris leis pagerinti mokymosi motyvaciją bei įvertinti kompetencijų pasiekimo lygį atsižvelgiant į iš anksto apibrėžtas taisykles. Mokymosi motyvacija pagerės, nes besimokančiajam bus parenkami jo mokymosi stilių labiausiai atitinkantys mokymosi objektai.

Ateityje ketinama sukurti šį modelį realizuojantį sistemos prototipą ir jį išbandyti realiame mokyme.

## LITERATŪRA

ACM/IEEE-CS (2008). Computer Science Curriculum 2008: An Interim Revision of CS 2001, ACM/IEEE-CS Joint Interim Review Task Force Report, *ACM Press*.

ACM/IEEE-CS (2012). *Strawman Draft: Computer Science Curricula 2013*. The Joint Task Force on Computing Curricula Association for Computing Machinery and IEEE-Computer Society.

ALLINSON, Christopher; HAYES, John (2012). *The Cognitive Style Index: Technical Manual and User Guide*. Pearson Education Ltd or its affiliate(s). 52 p.

BAZIUKAITĖ, Dalia; VAIRA, Žilvinas, IDZELYTĖ, Dominyka (2008). A tool to support self-education in a lifelong learning. In *Innovative techniques in instruction technology, e-learning, e-assessment and education*. New York: Springer, p. 92–97.

BENNEDSEN, Jens (2008). *Teaching and Learning Introductory Programming – A Model-Based Approach*: Dissertation for Dr. Philos degree in the Faculty of Mathematics and Natural Sciences, University of Oslo, Norway.

BERES, Ilona; MAGYAR, Timea; TURCZANYI-SZABO, Marta (2012). Towards a Personalized, Learning Style Based Blended Learning Model with Individual Assessment. *Informatics in Education*, vol. 11, no. 1, p. 1–28.

BIGGS, John (2003). *Teaching for Quality Learning at University*. 2nd edn. Open University Press, Berkshire.

BRUSILOVSKY, Peter; MAYBURY, Mark T. (2002). From adaptive hypermedia to the adaptive web. *Communications of the ACM*, vol. 45, no. 5, p. 30–33.

DAGIENĖ, Valentina; KURILOVAS, Eugenijus (2008). *Informacinės technologijos švietime: patirtis ir analizė*: Monografija. Vilnius: Matematikos ir informatikos institutas. 216 p.

DAGIENĖ, Valentina; URBONIENĖ, Jūratė (2010). Programavimo mokymasis: lyginamoji kalbos ir aplinkos analizė. *Informacijos mokslai*. VšĮ Vilniaus universiteto leidykla, t. 54, p. 44–62. ISSN 1392-0561.

ESMAHI, Larbi (2009). Personalized Web-Based Learning Services. In *Encyclopedia of Multimedia Technology and Networking*. Information Science reference. Hershey, New York, vol. I, p. 814–818.

FRANZONI, Ana Lidia; ASSAR, Said (2009). Student Learning Styles Adaptation Method Based on Teaching Strategies and Electronic Media. *Educational Technology & Society*, vol. 12 (4), p. 15–29.

GOMES, Anabela; CARMO, Lilian; BIGOTTE, Emilia; MENDES, António (2006). Mathematics and programming problem solving. In *Proc. of the 3rd E-Learning Conf. – Computer Science Education*, Coimbra, Portugal.

HAWK, Thomas F.; SHAH, Amit J. (2007). Using Learning Style Instruments to Enhance Student Learning. *Decision Sciences Journal of Innovative Education*, vol. 5, no. 1, p. 1–19.

HEROD, Lori (2005). Learning Styles and Strategies. In *Adult Learning & Literacy*, p. 13–22

JENKINS, Tony (2002). On the Difficulty of Learning to Program. In *Proceedings of 3rd annual conference of the LTSN-ICS*, Loughborough University, United Kingdom, p. 53–58.

*Learning styles – an introduction to the research literature*. Becta, Version 1.1, September 2005.

LISTER, Raymond; ADAMS, Elizabeth S.; FITZGERALD, Sue; FONE, William; HAMER, John; LINDHOLM, Morten; MCCARTNEY, Robert; MOSTRÖM, Jan Erik; SANDERS, Kate; SEPÄLLÄ, Otto; SIMON, Beth; THOMAS, Lynda (2004). A Multi-National Study of Reading and

Tracing Skills in Novice Programmers. *SIGCSE Bulletin*, December, vol. 36, iss. 4, p. 119–150.

MANNILA, Linda (2007). Novices' Progress in Introductory Programming Courses. *Informatics in Education*, vol. 6, no. 1, p. 139–152.

MOURA H. (2006). Adaptive e-Learning Environment Design. *Interactive Educational Multimedia*, no. 12, April, p. 62–71.

PARAMYTHIS, Alexandros; LOIDL-REISINGER, Susanne (2004). Adaptive learning environments and e-learning standards. *Electronic Journal on e-Learning*, vol. 2, iss. 1, February, p. 181–194.

PASHA, Muhammad Anwar-ur-Rehman; PASHA, Shaheen (2012). Missing Elements of Computer Science Curricula 2013. *Global Journal of Computer Science and Technology Cloud & Distributed*, vol. 12, iss. 11, version 1.0. Online ISSN: 0975-4172, Print ISSN: 0975-4350.

PREIDYS, Saulius; ŽILINSKIENĖ, Inga (2012). Nuotolinio mokymosi kurso personalizavimo modelis mokymosi veiklų atžvilgiu. Iš *Electronic Learning, Information and Communication: Theory and Practice (Elektroninis mokymasis, informacija ir komunikacija: teorija ir praktika)*. Vilnius: Vilniaus universitetas, p. 111–132. ISSN: 23352493.

REZAEI, Ali Reza; KATZ, Larry (2004). Evaluation of the reliability and validity of the cognitive styles analysis. *Personality and Individual Differences*, vol. 36, p. 1317–1327.

ROBINS, Anthony; ROUNTREE, Janet; ROUNTREE, Nathan (2003). Learning and Teaching Programming: A Review and Discussion. *Computer Science Education*, vol. 13, no. 2, p. 137–172.

RUSSELL, Gillian (2004). *Herrmann Brain Dominance Instrument*. The Ned Herrmann Group, Inc. 23 p.

WEBER, Gerhard; BRUSILOVSKY, Peter (2001). ELM-ART: An adaptive versatile system for Web-based instruction. *International Journal of Artificial Intelligence in Education*, no. 12, p. 351–384.

## A MODEL OF AN ADAPTIVE PROGRAMMING LEARNING SYSTEM

### Jūratė Urbonienė

#### S u m m a r y

The paper analyzes the characteristics of programming training, reviews students' learning styles and learning style influence on learning outcomes, the Herrmann brain dominance Instru-

ment (HBDI), which enables to classify learners according to their brain hemisphere development. Based on the programming learning experiences and fundamental programming training, scien-

tists' works, the programming learning difficulties (such as the programming field specificity, selection of learning styles and learning methods selection) are analyzed, and the key ideas are identified and summarized. The main focus is on adaptive programming learning system. The proposed adaptive programming learning system model

will improve students' motivation and the level of achieved results according to the pre-defined rules. The motivation will be improved by selecting a student's learning style and most appropriate learning objects. It is planned to create the system of introducing this model and testing it with real learners.