KAUNAS UNIVERSITY OF TECHNOLOGY

RENATA BURBAITĖ

ADVANCED GENERATIVE LEARNING OBJECTS IN INFORMATICS EDUCATION: THE CONCEPT, MODELS, AND IMPLEMENTATION

Summary of Doctoral Dissertation

Physical Sciences, Informatics (09P)

2014, Kaunas

The dissertation was prepared at Kaunas university of Technology, Faculty of Informatics, Department of Software Engineering in 2010–2014.

Scientific supervisor: Prof. Dr. Habil. Vytautas ŠTUIKYS (Kaunas University of Technology, Physical Sciences, Informatics – 09P).

Dissertation Defense Board of Informatics Science Field:

Prof. Dr. Habil. Rimantas BARAUSKAS (Kaunas University of Technology, Physical Sciences, Informatics – 09P) – **chairman**;

Prof. Dr. Vacius JUSAS (Kaunas University of Technology, Physical Sciences, Informatics – 09P);

Ass. Prof. Dr. Regina KULVIETIENĖ (Vilnius Gediminas Technical University, Technological Sciences, Informatics Engineering – 07T);

Ass. Prof. Dr. Olga KURASOVA (Vilnius University, Technological Sciences, Informatics Engineering – 07T);

Prof. Dr. Alfonsas MISEVIČIUS (Kaunas University of Technology, Physical Sciences, Informatics – 09P).

The official defence of the Dissertation will be held at the open meeting of the Board of Informatics Science Field at 10 a. m. on September 18, 2014 in the Dissertation Defence Hall of the Central Building of Kaunas University of Technology (K. Donelaičio g. 73, Kaunas).

Address: K. Donelaičio g. 73-403, LT–44029, Kaunas, Lithuania.
Phone (370) 37 30 00 42, fax. (370) 37 32 41 44, e-mail doktorantura@ktu.lt

The send out date of the summary of the Dissertation is on 25 July 2014.

The Dissertation is available at
http://ktu.edu/turinys/mokslo-renginiai
and at the Library of Kaunas University of Technology (K. Donelaičio g. 20, Kaunas, Lithuania).

KAUNO TECHNOLOGIJOS UNIVERSITETAS

RENATA BURBAITĖ

IŠPLĖSTINIAI GENERATYVINIAI MOKYMOSI OBJEKTAI INFORMATIKOS MOKYMUISI: KONCEPCIJA, MODELIAI IR REALIZACIJA

Daktaro disertacijos santrauka

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).

Informatikos mokslo krypties daktaro disertacijos gynimo taryba:

Prof. habil. dr. Rimantas BARAUSKAS (Kauno technologijos universitetas, fiziniai mokslai, informatika – 09P) – **pirmininkas**;

Prof. dr. Vacius JUSAS (Kauno technologijos universitetas, fiziniai mokslai, informatika – 09P);

Doc. dr. Regina KULVIETIENĖ (Vilniaus Gedimino technikos universitetas, technologijos mokslai, informatikos inžinerija – 07T);

Doc. dr. Olga KURASOVA (Vilniaus universitetas, technologijos mokslai, informatikos inžinerija – 07T);

Prof. dr. Alfonsas MISEVIČIUS (Kauno technologijos universitetas, fiziniai mokslai, informatika – 09P).

Disertacija bus ginama viešame Informatikos mokslo krypties tarybos posėdyje, kuris įvyks 2014 m. rugsėjo 18 d. 10 val. Kauno technologijos universiteto centrinių rūmų disertacijų gynimo salėje.

Adresas: K. Donelaičio g. 73-403, LT-44249, Kaunas, Lietuva.
Tel. (+370) 37 30 00 42, faksas (+370) 37 32 41 44, el. paštas doktorantura@ktu.lt

Disertacijos santrauka išsiųsta 2014 m. liepos 25 d.

Disertaciją galima peržiūrėti
interneto svetainėje *http://ktu.edu/turinys/mokslo-renginiai*
Ir Kauno technologijos universiteto bibliotekoje (K. Donelaičio g. 20, Kaunas).

1. INTRODUCTION

1.1. Relevance of the topic

In recent years, researching in e-learning is very intensive. Among other issues, research on various aspects of the educational content is a key topic. The educational content as an independent unit of the course is usually called learning object (LO) in the scientific literature. The main intention of using LOs in multiple educational contexts is the content reuse and interoperability.

In a wider context, LO is considered as an abstraction or a model to support reusability and interoperability among extremely large e-learning communities [1]. In general, e-learning covers a wide spectrum of tools, technologies, methodologies and standards. This is the reason why, having an abstract general concept, we are able to present and exchange educational information unambiguously. Moreover, without having a general concept, it would be impossible to develop e-learning theories, to compare e-learning results, and to exchange scientific information including practical experience.

The learning objects are created and stored in external or internal repositories, contextualised and standardized; various profiles and models of LOs, applications starting with semantic network and finishing with educational modelling languages and instructional engineering exist [2]. Typically, teachers, students, researchers, course designers, groups of scientists and organizations, etc. are the users of LOs. The provided analysis of the-state-of-the art shows that research on LOs forms a separate branch which is continuously being extended and developed. This research area is also widely discussed in the Lithuanian educational community.

Among multiple ideas and approaches proposed and dealt with in this branch of research, the generative learning objects (GLOs) should be mentioned in the first place. Boyle, Leeder, Morales and their colleagues (2004) [3] have introduced the GLO concept and approaches based on it aiming to enforce the reuse potential in e-learning domain.

Here, the term ‘generative’ should be understood as a property of the learning content to be produced and handled either semi-automatically or automatically under support of some technology. The contribution of GLOs in e-learning is that the extremely wide community involved in learning has received a sign to move from the *component-based reuse model* (it basically relates to the use of LOs) to the *generative-based reuse model*, which relates to the use of GLOs. For example, the source [4] defines the generative learning object as *“an articulated and executable learning design that produces a class of learning objects”*. In general, this definition satisfies our vision in this dissertation.

The number of proponents to use GLOs is constantly growing. Our research on GLOs is different as compared to other approaches, because we use meta-programming [5] as a generative technology to implement GLOs. Despite of the effort and contribution of proponents to use the GLO-based approaches, however, this research trend is still in its infancy. There are many unsolved problems such as: (i) systematization, (ii) high-level modelling, (iii) automated design, (iv) portability of the GLOs to various learning environments, (v) the real application in teaching/learning of informatics by integrating specialized environments (educational robots-based, microcontroller-based) into the learning process. We consider a great deal of those issues in this dissertation.

Our research object is called *"advanced generative learning object"* (AGLO). We analyze the GLOs of a new generation that come from generative technology (heterogeneous meta-programming technology) with extended capabilities. This technology enables to express a variety of learning aspects (content, pedagogical, social, and technological) through parameterization explicitly. As the learning content in informatics is a program or its parts, GLOs of this type are the best choice for teaching/learning conceptually and practically.

1.2. Research object

In this dissertation, the object of research is the advanced generative learning objects, models and processes related to them.

1.3. Objective and tasks

The objective of the research is to develop and to investigate the methods that enable to formalize the designing of advanced generative learning objects and using them in teaching/learning of informatics effectively.

In order to achieve the objective, the following tasks have to be solved:

1. Analysis of the state-of-the-art as related to the learning objects in e-learning in general and in the informatics learning context.
2. Modelling of the informatics learning domain aiming at creating *feature-based general models* from which we could be able to extract *the concrete models* for designing advanced generative learning objects.
3. Formalized specification and design of the advanced generative learning objects.
4. Creation of the heterogeneous robot-based learning environments and integration of the advanced generative learning objects into the environments.
5. Experimental evaluation of the proposed methods using known technological and pedagogical criteria.

1.4. Methods of research

We have applied and used the following methods, theories and formalisms in the dissertation: feature-based modelling approaches, formal verification of feature models, heterogeneous meta-programming (PHP as a meta-language and

RobotC as a target (teaching) language), the first order logic theory, set theory, informal pedagogical methods and pedagogical theories (mainly constructivist).

1.5. Statements presented for defence

1. *Learning variability in informatics* is the background to design and use the advanced generative learning objects.
2. Feature-based models, at the higher level of abstraction, implement the learning variability concept.
3. Two-level models being executable specifications enables automatic content generation.
4. The heterogeneous robot-based learning environments serves for the efficient use of AGLO.

1.6. Scientific novelty

1. Advanced generative learning objects expand the informatics learning variability aspects (pedagogical, social, technological, and content). Based on those insights, it is possible to adapt and apply software engineering and computer science methods in the e-learning domain.
2. To our best knowledge, feature-based modelling in the informatics learning domain has been performed systematically for the first time. Such an approach evaluates the domain variability, aggregates and verifies the created models.
3. Formalization of the models at two levels (feature-based and executable specification) provides pre-conditions for automated tools design.
4. From the viewpoint of automatic educational content creation, advanced generative learning object extends the concept of reusability in e-learning.

1.7. Practical relevance

1. The architecture of a heterogeneous specialized learning environment based on educational robots and microcontrollers is designed, tested and used practically.
2. Advanced generative learning objects that ensure the physical visualization of the program behavior within the specialized learning environment are developed.
3. Advanced generative learning objects are integrated into the real teaching/learning process and, in this way, the objects implement contributing to interdisciplinary principles of education (in general, known as *Science, Technology, Engineering, Mathematics* – STEM).
4. The proposed methods support the possibilities to integrate processes into e-learning management systems.

5. The proposed methods have been evaluated using the known pedagogical and technological criteria. The statistics obtained through experimental research (2011-2014) enables to state that the methods are efficient enough.

1.8. Approbation of the research results

The main results of the dissertation are represented in 10 scientific publications: 4 in the periodical scientific journals (3 in ISI Web of Science), 5 in the international conference proceedings, 1 in the local conference proceedings.

1.9. The structure and volume of the dissertation

The dissertation consists of an introduction, 6 main chapters and the conclusion. A list of author publications, a list of references and 2 appendixes are given additionally. The total volume of the dissertation consists of 150 pages, including 57 figures, 27 tables and 223 references.

2. THEORETICAL BACKGROUND OF THE INFORMATICS LEARNING DOMAIN MODELLING METHOD

Three terms (programming, CS, Informatics) are treated as synonyms throughout the dissertation. We use the first in the concrete narrow context, while the remaining ones we use as general terms.

In Fig. 2.1, we present a general research framework. In the first stage, we need to perform domain analysis. Then, we specify AGLOs requirements, create AGLOs models, and describe instructional design processes. In the last stage, we evaluate AGLOs quality and their storing, searching, selecting, generating, modifying, and adapting capabilities, learning processes and feedback.

In this context, by modelling we mean the extraction from the informatics learning domain a set of models as input data to enabling then the creation of GLOs through transformations.

For successful modelling of the e-learning domain, it is necessary to express the domain explicitly. In our research, we use TPACK (*Technological Pedagogical Content Knowledge*) framework [6] (see Fig. 2.2), which describes the informatics learning domain.

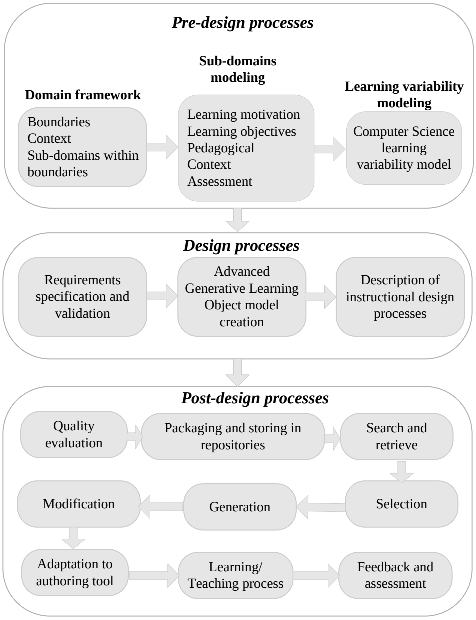


Fig. 2.1 A general research framework

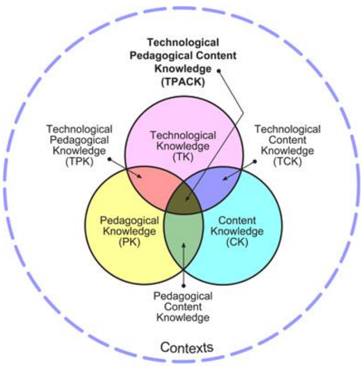


Fig. 2.2 TPACK framework [6]

2.1. The principles used to construct the method

We use the dual fundamental principles known in software engineering as "separation of concepts" (separation of concerns) and "integration of concepts" to construct our method. The dual means that principles are typically applied both: firstly separation and then integration. More generally, they perhaps can be treated similarly as analysis and synthesis.

The principle of separation of concepts might be stated as the premise that entities (e.g. in our case, concepts related to LOs or GLOs) should contain the essential attributes and behaviors inherent to their nature, but should be void of attributes and behaviors not inherent to their nature. The domain analysis methods (FODA, SCV, etc.) are actually built upon the explicit use of separation and integration of concepts. In e-learning this term is not yet so popular. However, the term is well understood for the CS researchers [7].

We use "analogy principle" to construct our method too. In the context of our research, we have an analogy between course designing and program family designing; the structure of the course is similar to software architecture. In the higher-level a set of features models the software components within an architecture. Similarly, a set of LOs models topics of the course.

2.2. Requirements for the modelling method

Requirement 1. The domain of informatics learning is heterogeneous, so the scope and boundaries have to be defined clearly.

Requirement 2. The scope and boundaries of the domain can change depending on the objectives of the analysis.

Requirement 3. As a result of Requirement 1 and Requirement 2, domain should be represented as a set of adequate models relevant to general objectives.

Requirement 4. The aims of models' usage have to be defined before creating the model.

Requirement 5. Various manipulations can be done with models: merging, splitting, aggregation, etc.

Requirement 6. All newly created models and those devised through manipulations have to be correct, therefore the model verification should be at the focus.

Requirement 7. Creating of feature diagrams and manipulating operations with models should be supported by adequate tools.

Requirement 8. For easiness of handling and managing, it is useful to introduce model hierarchies for representing them at the different levels of granularity.

Requirement 9. It is appropriate to create a feature model (FM) as a pair of the base model and its context model. In that way, a priority relation is a useful mechanism.

Requirement 10. Context model may be introduced in two forms: implicit or explicit. We use the explicit form as it is more suitable from the viewpoint of models' transformation.

2.2. Analysis methods of informatics learning domain

In the dissertation, we use FODA (*Feature-Oriented Domain Analysis*) method. Three main principles of FODA are being used: 1) identification of domain boundaries and context; 2) feature-based modelling of the context; 3) feature-based modelling of sub-domains within the domain [5, 8].

For the identification of the domain variability we use SCV (Scope-Commonality-Variability) principle based on a theory of sets [9, 10].

2.3. The modelling method of the informatics learning domain

In Fig. 2.3, we present an overall view of the modelling method. We state it as a logical sequence of high-level processes along with their outcomes. Each process is described as a goal-driven input-output relationship, according to the following scheme: *the aim-input data-process-outcome.*

Process 1. The aim is to set initial conditions for the remaining processes. As the FODA and SCV methods indicate, the identification of boundaries is the important pre-condition of modelling because it specifies the volume (scope) of the activity. The attribute IN1 includes: FODA and SCV instructions, TPACK framework. This attribute can be fulfilled through analysis of TPACK (the latter is treated as the base domain here) by an analyzer (modeler); the basis is his/her competence in the field; the use of some instructional materials and documents such as standard specifications, relevant papers, etc. are important. Context model is the outcome here. We can describe the model by encountering such domains or their influential attributes, which are close in terms of the importance of their relationships with the base domain.

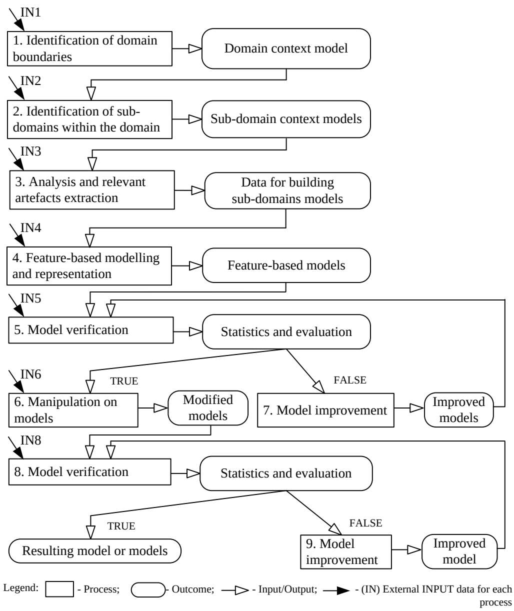


Fig 2.3 Overall view of the proposed method

Process 2. Its aim is to simplify modelling by identifying domain's boundaries and narrowing the model of the domain. IN2 covers the rules of FODA and SCV, TPACK framework, the principles of separation of concepts and analogy. The process is carried out while reviewing TPACK framework and grounding the aims of modelling of each sub-domain (pedagogical, technological, content). The outcome of the process is narrower context models of sub-domains.

Process 3. The aim is to analyze and extract the relevant artifacts for modelling. IN3 covers methods, tools, experts of the domain, knowledge, solutions, requirements, etc. The process is carried when analyser, who uses knowledge of the domain experts and his/her own experience, collects, classifies and verifies the data. The outcome of the process is sets of the data which will be used when creating the primary models of sub-domains.

Process 4. The aim is to present the models abstractly and accurately. IN4 covers feature-based language and tools such as FAMILIAR, SPLOT, knowledge and competence of the analyzer. The process is based on identification of relations and constraints among features when creating and testing models. The outcome is the set of the FMs.

Process 5. The aim is to verify models and to collect statistics. IN5 covers the model verification tools (SPLOT), knowledge and competence of the analyzer. The process is carried when using those tools. The outcome of the process is the statistics of the model features.

Process 6. The aim is to identify the objectives of the use of the multiple models. IN6 covers requirements for manipulations with models, the tool FAMILIAR. The process is carried when using this tool. The outcome of the process is multiple models.

Process 8 is analogical to Process 5. Process 9 repeats Process 7 in which models that do not satisfy the requirements are corrected and re-verified.

2.4. Informatics learning domain's feature models

By using the developed method, a set of informatics learning domain models have been created (some models are presented in Fig. 2.4-2.7).

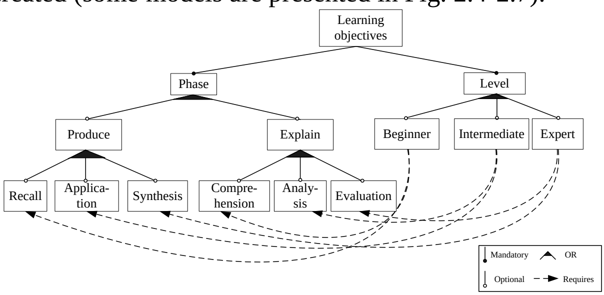


Fig. 2.4 Learning objectives model

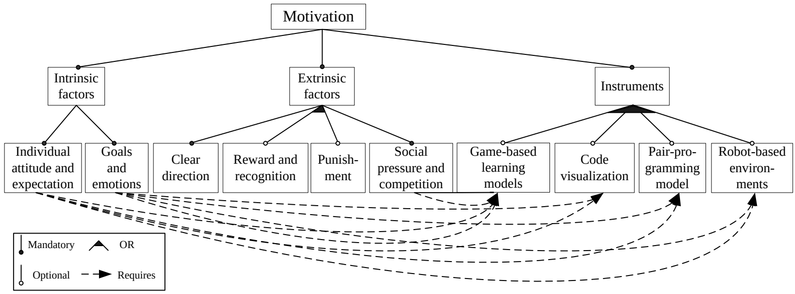


Fig. 2.5 Motivation model

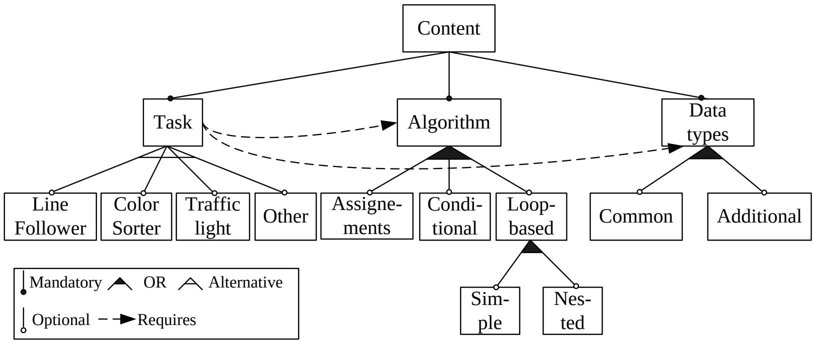


Fig. 2.6 Content model

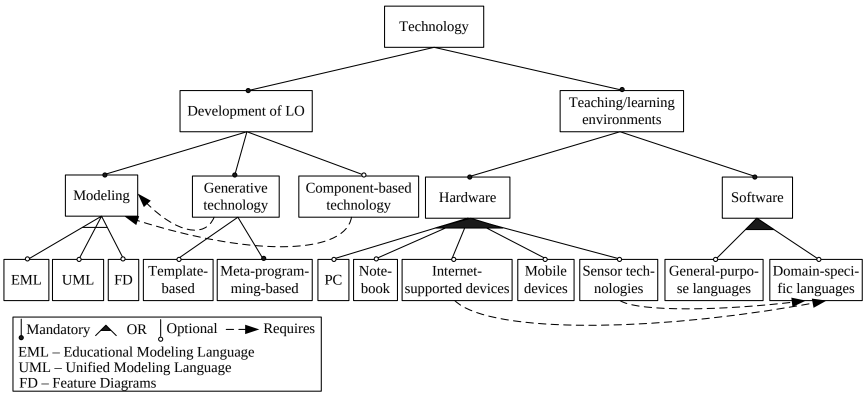


Fig. 2.7 Technological aspects model

2.5. Analysis and verification of feature models

The AGLOs' quality depends on quality of models starting with the earliest designing stages. Structural metrics of FMs are important factors of external quality. Computing methods of structural metrics are based on BDD (*Binary Decision Diagrams*). SAT Solver algorithms are used to evaluate a consistency of FMs, the number of dead features and possible configurations [11, 12].

In Tables 2.1-2.2, we present the main FMs quality's characteristics.

Table 2.1 Statistics of informatics-based Feature Models

No.	Parameter	Pedagogy (M– Motivation, LObj – Learning Objectives, TL – Teaching/Learning, A – Assessment, L - Learner)					Content	Technology
		M	LObj	TL	A	L		
1.	# Features	14	14	37	17	24	13	20
2.	# Optional	0	0	0	0	0	0	2
3.	# Mandatory	7	2	10	0	5	3	7
4.	# Grouped	6	11	26	16	18	9	10
5.	# OR groups	2	4	10	4	4	2	2
6.	# XOR groups	0	0	0	1	0	1	1
7.	#Cross-Tree Constraints (CTC)	9	6	3	3	3	2	4
8.	CTCR (%)*	0.50	0.64	0.11	0.18	0.25	0.23	0.30
9.	#CTC distinct variables	7	9	4	3	6	3	6
10.	CTC clause density**	1.29	0.67	0.75	1.00	0.50	0.67	0.67
11.	Tree Depth	3	5	9	6	3	3	4

*CTCR – constraints representativeness, number of variables in the CTC divided by the number of features in the Feature Diagram.

** CTC clause density is number of constraints divided by the number of variables in the CTC.

Table 2.2 Analysis of informatics-based Feature Models

No.	Parameter	Pedagogy (M– Motivation, LObj – Learning Objectives, TL – Teaching/Learning, A – Assessment, L - Learner)					Content	Technology
		M	LObj	TL	A	L		
1.	Consistency	+	+	+	+	+	+	+
2.	# Dead Features	None	None	None	None	None	None	None
3.	# Core Features	12	5	1	1	6	4	8
4.	Count Configurations	3	61	131071	95	74803	84	828
5.	Variability Degree (%)*	1.8311E-2	3.7231E-1	9.5367E-5	7.2479E-2	4.4586E-1	1.0254E0	7.8964E-2
6.	# BDD nodes	14	49	103	95	35	16	26

*Variability Degree is the number of valid configurations divided by 2^n, where n is the number of features in the model.

2.6. Properties of the models

In this section, we formulate the most important properties of models which are related to the importance of the models to the informatics learning domain.

Property 1. As informatics teaching and learning is a heterogeneous domain, we need to use *multiple feature models* aiming to representing the domain at a higher-level of abstraction due to (i) ever-increasing requirements, (ii) complexity growth of the domain itself, (iii) need for reuse enhancement and (iv) automation purposes.

Property 2. A set of FMs presented in section 2.4 has the same semantics as the selected papers on e-learning describe, from which the feature has been extracted. The benefits of models are: preciseness, correctness, conciseness and reusability.

Property 3. FMs are highly reconfigurable items. Merging, splitting, changing, aggregating, etc. operations enable to perform the adequate reconfiguring on demand.

Property 4. In the case of using multiple models, their priority relation can be modelled by the priority levels, such as: *high*, *intermediate*, *low*.

Property 5. The developed FMs are correct with *regard to domain-based correctness* under the following assumptions: 1) the model designer has used initial data to specify models, which were created by domain experts; 2) the designer has applied *allowable manipulations* on domain initial data; 3) relationships and constraints were formed on the basis of expert knowledge.

Property 6. The developed FMs are semantically correct because the following conditions hold: 1) the models are specified using the notion accepted by the FAMILIAR language and tools; 2) the tool SPLOT we use supports formal verification of models devised with the help of FAMILIAR.

Property 7. There is no unique attribute to characterize FMs; rather multiple characteristics should be applied. The list of characteristics to evaluate models may be as follows: number of models, complexity, *degree of variability*, relevance to the requirements of a specific task such as implementation; characteristics obtained by selected tools used.

Property 8. The developed models specify and model the informatics teaching and learning domain to the extent relevant to the predefined *scope and aims* of modelling.

3. DESIGNING OF ADVANCED GENERATIVE LEARNING OBJECTS

The advanced generative learning object is the product of the implementation of the learning variability into technology. It supports predefined features. In this section, we expand the theoretical background of AGLOs.

3.1. Specification of advanced generative learning object by using feature diagrams

The FMs' complexity management problem is raised because the FM consists of a big number of features and relationships among them.

We define the terms that are required to specify AGLO using FMs.

Definition 1. AGLOs' family is a set of the LOs that are defined by common features.

Definition 2. AGLOs' FM consists of context and content FMs' that are semantically related by relationships and constraints between them, and priorities model (see Fig. 3.1).

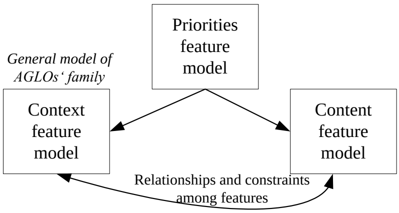


Fig. 3.1 Generalized model of AGLOs' family

Definition 3. AGLOs' context model (Context_FM) is a concrete FM which is general for AGLOs family. Context model is a result of aggregation of specialized informatics learning sub-domains FMs:

$$Context\_FM = LObj_{spec} \odot M_{spec} \odot TL_{spec} \odot A_{spec} \odot L_{spec}; \quad (3.1.)$$

where $LObj_{spec} \subset LObj$ – learning objectives FM; $M_{spec} \subset M$ – motivation FM; $TL_{spec} \subset TL$ – teaching/learning methods FM; $A_{spec} \subset A$ – assessment FM; $L_{spec} \subset L$ – learner's FM; "$\odot$" – aggregating operator of FM.

Definition 4. AGLOs' content model (Content_FM) is a concrete FM that is based on the content requirements model, and is defined as a content variability model:

$$Content\_FM = \langle CF, CE_m, CF_a, CF_o, REQ\_C, EXC\_C \rangle; \quad (3.2)$$

where $CF = (FC, CE, fc)$ is a rooted tree where FC is a finite set of content features, $CE \subseteq FC \times FC$ is a finite set of edges, $fc \in FC$ is the root content feature, $CE_m \subseteq CE$ is a set of edges that define mandatory features with their parents, $CF_a \subseteq P(CF) \times CF$; $CF_O \subseteq P(CF) \times CF$ define alternative and optional feature groups and are sets of pairs of child features together with their common parent feature, REQ_C and EXC_C are finite sets of constraints '*requires*' and '*excludes*'.

Definition 5. The AGLOs' priorities model is a concrete FM general for the AGLOs' family and is described as follows:

$$Priorities\_FM = \langle P, PE_m, REQ\_P \rangle \text{;} \qquad (3.3)$$

where $P = (PF, PE, fr)$ is a rooted tree where PF is a finite set of priorities features, $PE \subseteq PF \times PF$ is a finite set of edges, $fr \in PF$ is the root feature, $PE_m \subseteq PE$ is a set of edges that define mandatory features with their parents, REQ_P is a finite set of the constraint "*requires*".

In Table 3.1, we present the main AGLOs FMs quality's characteristics.

Table 3.1 Characteristics of AGLOs FMs obtained using FAMILIAR and SPLOT

No.	Task / Model metrics	Robot Calibration	Line Follower	Ornaments design	Scrolling text on LCD	Light follower	Traffic light
1.	# Features	38	44	51	27	41	44
2.	# Mandatory features	11	10	15	7	10	12
3.	# Core features	15	14	20	8	11	14
4.	# XOR groups	8	8	11	5	7	8
5.	# OR groups	1	1	2	2	1	2
6.	#Cross-Tree Constraints	18	12	21	12	7	14
7.	CTCR, % *	0.53	0.57	0.43	0.63	0.24	0.39
8.	Tree Depth	3	3	3	3	3	5
9.	Valid Configurations	1296	8640	62208	1440	87480	97200
10.	Variability degree, % **	4.7148 E-7	4.9113 E-8	2.7626 E-9	1.0729 E-3	3.9781 E-6	5.5252 E-7

*CTCR – constraints representativeness, number of variables in the CTC divided by the number of features in the Feature Diagram.

**Variability Degree is the number of valid configurations divided by 2^n, where n is a number of features in the model.

3.2. Advanced generative learning objects and meta-programming-based technology

In the research, we apply heterogeneous meta-programming technology which enables to implement AGLOs by expressing task's variability explicitly. In the context of the dissertation, domain variability is considered as learning variability.

Definition 6. Semantically, AGLO is an explicit mapping of learning variability onto the solution domain using heterogeneous meta-programming technology.

Definition 7. Structurally, AGLO is a set of pre-specified and automatically generated LOs or a concrete LO. Formally, the model of AGLOs can be defined:

$$AGLO = MI \times MB \text{;} \qquad (3.4)$$

where MI – meta-interface, MB – meta-body, "$\times$"– mapping.

3.3. High-level model transformation to executable specification

In this section, the transformation rules are stated.

Rule 1. *Variant point* in the *FM* corresponds to a *parameter name* in the executable specification.

Rule 2. Variants of a variant point within the *FM* correspond to parameter values in the executable specification.

Rule 3. The format of a *simple assignment statement* within the interface is as follows:

<*parameter*>= <*parameter_value_set*>.

Rule 4. The format of a *conditional assignment statement* within the interface is as follows:

<*parameter1*><*condition*><*parameter2*><*parameter1*>=<*parameter_value_set*.

The conditional assignment statement appears if and only if the adequate variant point has constraints <*requires*> or <*excludes*>.

Rule 5. The number of parameters in the executable specification must be equal to the number of variation points in FM.

Rule 6. Parameters in the meta-interface of executable specification of AGLOs are arranged according to priorities from high to low.

Rule 7. To form meta-body the following set of functions of the meta-language is used:

{assignment ('='), OPEN-WRITE-CLOSE, conditional, loops}.

3.4. Properties of advanced generative learning objects

Property 1. Creating of high-level (HL) AGLOs' is mapping of learning variability (LV) onto the model of heterogeneous meta-program (MP). Formally, it is expressed as:

$$AGLO_{HL} = FD_{LV} \times MP_M ; \qquad (3.5)$$

where $AGLO_{HL}$ – high-level *HL* model of *AGLO*; FD_{LV} – learning variability *LV*, expressed by concrete feature model *FD*; MP_M – models of heterogeneous meta-programming domain, "$\times$ "– mapping.

Property 2. Meta-programming based AGLOs are heterogeneous meta-programs.

Property 3. The meta-interface of AGLOs expresses a set of parameters values that allow creating an instance of LOs with selected values of parameters.

Property 4. Meta-body of AGLOs consists of a pre-provided set of meta-language functions that are included in the code of LOs according to predefined format and rules.

Property 5. From a viewpoint of the teacher and learner, AGLOs are "the black-box" entities.

Property 6. From a structural point of view, AGLOs are the high-level specification that describes a family of related LOs instances.

Property 7. From a behavioral point of view, AGLO is a generator that generates instances of LOs according to user's requirements.

From a practical point of view, a set of AGLOs forms the LO library or a part of it, in which a set of related LOs is stored as a compact package.

Property 8. From a technology application in programming (informatics) learning viewpoint, AGLO is considered as a meta - program where LO is a program written in the target language.

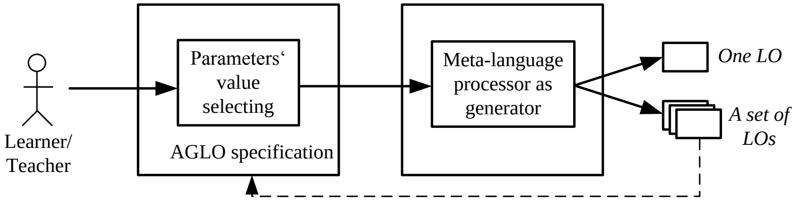


Fig. 3.2 Behavioral models of AGLO's

3.5. Advanced generative learning objects technological complexity evaluation

In Table 3.2, we present AGLOs technological complexity evaluation using metrics taken from [5].

Table 3.2 AGLOs technological complexity evaluation

No.	Task / Complexity metrics	Robot Calibration	Line Follower	Ornaments design	Scrolling text on LCD	Light follower	Traffic light
1.	Relative Kolmogorov Complexity	0.24	0.19	0.19	0.26	0.22	0.21
2.	Metalanguage Richness	0.67	0.75	0.60	0.69	0.64	0.59
3.	Cyclomatic Complexity	360	1152	27216	24	2916	2916
4.	Normalized Difficulty	0.14	0.16	0.11	0.05	0.03	0.04
5.	Cognitive Difficulty	185	473	246	220	262	213

4. INTEGRATION OF ADVANCED GENERATIVE LEARNING OBJECTS INTO LEARNING ENVIRONMENTS

AGLOs can be used effectively only if we are able to integrate them into learning environments. In this section, we present the requirements for the environments, their architectures along with quality criteria to evaluate them.

4.1. The requirements for learning environments

The learning environment helps to achieve learning objectives by using a specific learning content and covers learning resources, interaction and communication among the participants of the learning process, learning activities and learning support activities. AGLOs are the root of the informatics learning conceptual model. They interact with pedagogical activities, technological processes, knowledge transfer channels, tools and pedagogical outcomes (see Fig. 4.1).

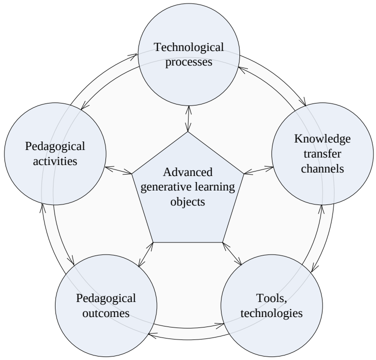


Fig. 4.1 Informatics learning conceptual model

Pedagogical activities are closely related to learning objectives, content, teaching model, selection of the tools, formulation of the task, evaluation of the pedagogical outcomes.

Technological processes start with choosing the task. Those processes allow creating AGLO, but they depend on tools, programming languages and algorithms that cover topics of the course. After the creation or selection of AGLOs from the library, the parameters' selecting and content generating processes occur. The user compiles and executes generated program, and performs the control of the task's solution.

Knowledge transfer channels connect pedagogical activities and technological processes.

The feedbacks among components ensure the flexibility of the content regeneration, modification and knowledge extraction through learning scenarios.

4.2. Functionality and architecture of a specialized heterogeneous learning environment

The specialized heterogeneous programming learning environment includes three interrelated parts: teacher's and learner's components and server (see Fig. 4.2 a). The teacher's component consists of the teacher's computer with the software for creating AGLOs and software of general use that ensures communication with the server (queries, AGLOs transfer to/from server) (see Fig. 4.2 b). Created AGLOs are transferred to AGLOs repository in the server. On the learner's computer we install software of general use that enables to generate the LO according to user's requirements. Moreover, programming language environments that create an executable specification transferring to educational robot or microcontroller must be in the learner's computer too (see Fig. 4.2 b).

In Fig. 4.2 c), we present a behavioral model of the proposed environment. Firstly, we create AGLO's specification and transfer it to the repository. The designer can modify AGLO at any moment.

The learner can find AGLO in the repository by using software of general use. He/she selects the values of parameters in the user's meta-interface and generates LO. Later, learner uploads it to programming language environment and creates executable specification, and after that transfers it into a robot or a microcontroller.

The teacher ensures monitoring and flexible feedback.

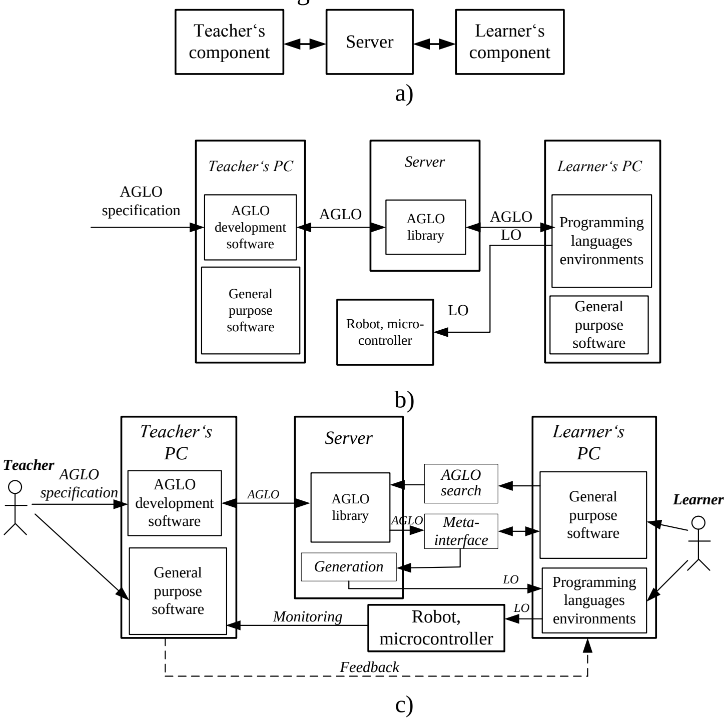


Fig. 4.2 Heterogeneous learning environment: a) – general structure, b) – environment's components structure, c) environment's behavioral model

When creating the educational robot environment, we highlight two stages: 1) preparation for the operating; 2) working mode (see Fig. 4.3).

In the first stage, we construct the educational robot that will solve the pre-defined task. The next important step within the process is the measurement of technical parameters of the robot, because these parameters are used for the robot control program.

In the second stage, we create the robot control programs automatically and transfer them into a robot, and then we implement visualization of the task.

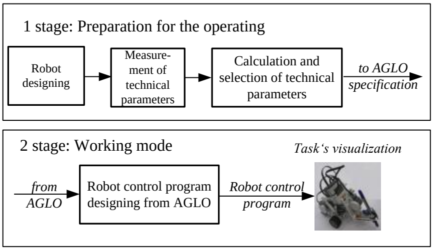


Fig. 4.3 Designing stages of educational robot-based environment

4.3. The collaborative robot-based architecture

The architecture of the collaborative robot-based learning environment refers to a classical master-slave model and includes additional components required for robot orientation in its environment (sensors, wireless cameras), communication channels to ensure the exchange of messages between communicating robots and support for different communication protocols (Bluetooth, WiFi), and control hardware/software (PC). In the master-slave model, slaves perform parallel computations and the master does sequential computations. We control sub-processes using communication between the master and slaves either by a single node broadcast from the master or by send/receive messages exchanged between the master and any slave. The principle is similar to task decomposition so that the master-slave model itself can be used as an illustrative example of practical implementation of task decomposition.

Fig. 4.4 presents a four-tiered framework to construct the collaborative robots-based environment as follows:

1) *Deliberative layer:* Central Coordinator (CC) receives initial tasks for robots from the teacher, then decomposes tasks into sub-tasks and uploads generated robot control programs (RCP) to the student PCs. In the simplest case, each task is divided into two sub-tasks (Master → Slave) and also we have two independent groups of students (GROUP1, GROUP2) assigned to work with the same task.

2) *Physical layer:* tangible mobile robots with wheels driven by servo motors.

3) *Reactive layer:* sensors allow a robot to receive information about its environment and react to its changes.

4) *Communication layer:* exchange of messages between robots and

provision of feedback to teacher's PC for monitoring and evaluation.

On real setting, the number of collaborating robot groups depends on the technical capabilities (the number of available robots and PCs in the classroom) and educational needs (the number of students, teaching and learning objectives, etc.). In order to ensure satisfaction of educational needs and improvement of technical reliability, we provide a real-time "student-teacher" feedback and monitoring of collaborative behavior of robots.

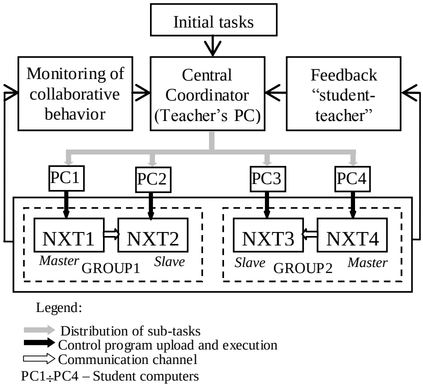


Fig. 4.4 Framework of collaborative robots based environment for e-learning [13]

4.4. Case Study: the teaching/learning process using AGLOs and educational robot-based environment

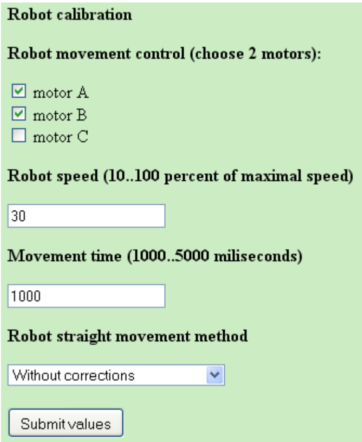
The case study demonstrates the ability to solve and visually represent a set of related graph-based tasks (given as LOs) in teaching programming (i.e. in informatics, or computer science). A particular LO adapted to the learning context is derived from the AGLO's automatically. We summarize the overall process below as follows:

1. Learning/teaching subject: *Computer Science*.
2. LO topic: *Loops and Nested Loops in a Computer Program*.
3. e-learning environment: *Lego-based DRAWBOT (drawing robot)*.
4. Learning content: *an LO derived from AGLOs*.
5. Learners: *10-11th grade secondary school students at J. Balčikonis Gymnasium*.
6. Pedagogical model used: *Constructivist*.
7. Learning objectives: *Visualization of the process and learning content*.
8. Process description by teacher: *a) design and testing of the e-learning environment; b) design and testing AGLO; c) testing-generating LO instances from AGLO to apply them in a different context of use*.
9. A learning activity by students: *a) design of the robot mechanics under the teacher guidance b) identification of robot characteristics relevant for teaching*

tasks; c) participation in the development of AGLOs, including robot control programs as AGLOs and content visualization programs as LOs.

10. Learning evaluation*: a) teacher makes observes and records students' activity actions, feedback and on this basis evaluated the gained knowledge.*

We analyze two AGLOs here. The first is "Robot calibration" (see Fig. 4.5), because these parameters are used for the robot control program. Motors are controlled for specifying a power level to apply to the motor. The programming language RobotC uses parameter named "Power level". Power levels range from –100 to +100. Negative The distance driven by the robot per time depends on the motor's Power level. The movement of the robot depends on the robot's construction and motor's technical parameters. To ensure the smooth movement there are three operating modes: 1) manual adjustment by the motor command "Power level" for the straight robot's move, 2) use of the PID (Proportional-Integral-Derivative) speed control algorithm, 3) use of the motor synchronization to ensure that both motors run at the same speed [14].




```
task main()
{
// Initial states of robot motors
   motor[motorC] = 50;
   wait1Msec(100);
   motor[motorC] = 0;
   // Straight movement of robot
   motor[motorA] = 30;
   motor[motorB] = 30;
   wait1Msec(1000);
// Final states of robot motors
   motor[motorA] = 0;
   motor[motorB] = 0;
   motor[motorC] = -50;
   wait1Msec(100);
   motor[motorC] = 0;
}
```

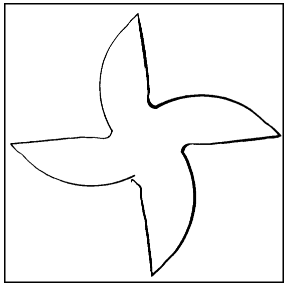
a) b)

Fig. 4.5: a) – Meta-interface of GLO "Robot calibration", b) – Generated instance as LO

Now we consider the second AGLO "Ornaments' drawing"of our case study. It deals with the task that responds to the requirement to ensure the possibility for better students' engagement in learning. The task (to teach loops in the program) is about *visualization* of the result created by the program. The program is derived from the AGLO as a LO instance (see Fig. 4.6 a)). Then the instance runs within the robot environment that makes drawing to realize the visualization (see Fig. 4.6 b)).

```
task main()
{
//---------------------------------
// Preparation for drawing
   motor[motorB] = 50;
   wait1Msec(100);
   motor[motorB] = 0;
   //------------------------------
   // Drawing
   for (int j = 0; j < 4; j++) {
      motor[motorC] = 50;
      motor[motorA] = 50;
      wait1Msec(1000);
      //---------------------------
      motor[motorC] = -50;
      motor[motorA] = 0;
      wait1Msec(1000);
   }
   //------------------------------
   // Drawing is finished
   motor[motorB] = -50;
   wait1Msec(100);
   motor[motorB] = 0;
   //------------------------------
}
```

a)



b)

Fig. 4. 6: a) – Generated LO instance (from AGLO “Ornaments' drawing”) as motivating example to cover “Loops-teaching“, b) – Result of LO execution as a material introduced by teacher for learning at initial phase through problem solving

4.5. Learning environments' evaluation

In Tables 4.1-4.2, we present technological and pedagogical evaluation of created learning environments quality. The quality's criteria are adapted from [15] (technological) and [16] (pedagogical).

Table 4.1 Learning environments' technological evaluation*

Criteria \ Environment	A single robot-based	The collaborative robot-based
Scalability	3	4
Modularity	2	3
Reasonable performance optimizations	3	3
Robustness and stability	3	2
Reusability and portability	3	3
Localisable user interface	4	4
Localization to relevant languages	4	4
Facilities to customize for the educational institution's needs	3	3
Automatic adaptation to the individual user's needs	3	3
Automatically adapted content	3	3
Additive utility function of technological criteria	**31**	**32**

Table 4.2 Learning environments' pedagogical evaluation**

Criteria \ Environment	A single robot-based	The collaborative robot-based
Knowledge of learning content	3	3
Knowledge of learning process	4	4
Cognitive learning skills	4	4
Affective learning skills	4	4
Social learning skills	4	4
Transfer skills	4	4
Additive utility function of pedagogical criteria	**23**	**23**
Preparatory learning functions	C A M	C A M
Executive learning functions	C A M	C A M
Closing learning functions	C A M	C A M
Learning theory	Constructivism	Constructivism
Learners' roles	Cp Cm (I)	Cp Cm

*The rate range is 0÷4 (0 – no support, 1 - poor support, 2 – fair support, 3 – good support, 4 – excellent support)

**C – Cognitive, A – Affective, M – Metacognitive, Cn – constructivism, Cp – cooperative, Cm – competitive, I – individual

5. PEDAGOGICAL EVALUATION OF ADVANCED GENERATIVE LEARNING OBJECTS

Pedagogical effectiveness of using AGLOs can be evaluated by "engagement levels" using the methodology described in [17]. Fig. 5.1 explains assessment of the student engagement levels:

1. *Viewing*: Students view the programs given by teacher passively and are passive LO consumers.

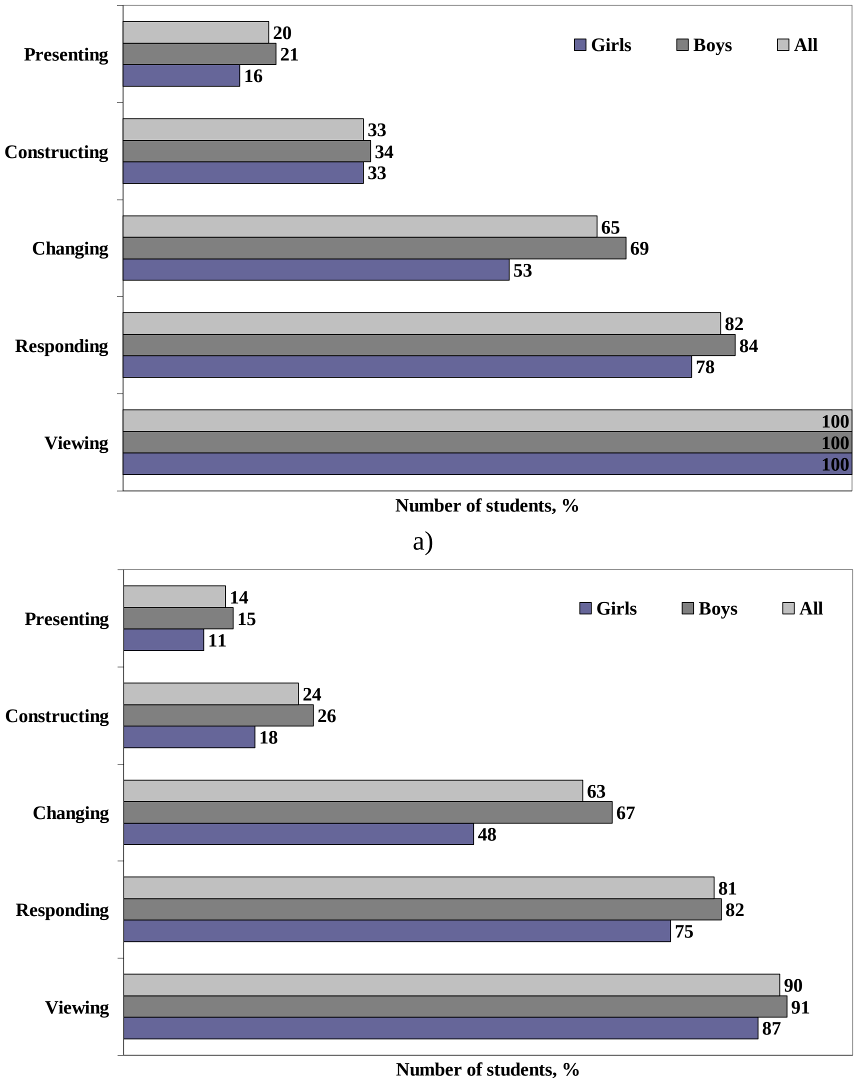
2. *Respondin*g: Students use the visualization of programs actively as a resource for answering questions given by teacher and are active LO consumers.

3. *Changing*: Students themselves modify programs by changing the meta-parameter values and are LO designers.

4. *Constructing*: Students construct their own programs introducing new meta-parameters, their values and are LO co-designers and testers.

5. *Presenting*: Students present new programs to the audience for discussion and are GLO co-designers.

The statistics are obtained through experimental research over 3 years (2011-2014).



b)

Fig. 5.1 Student engagement levels (2011 to 2014, 186 students: 141 boys, 45 girls): a) using AGLOs; b) not using AGLOs

CONCLUSIONS

1. It has been obtained through the analysis that the methodological background of e-learning (pedagogical theories, standardization initiatives, social aspects, etc.) are general; however, learning in informatics has its own specificity (teaching/learning models, learning environments, presentation of educational content, etc.), which requires a separate attitudes and research. We have proposed a new concept of advanced generative learning objects. The background of the concept is the learning variability modelling along with heterogeneous metaprogramming as implementing techniques.
2. We have developed the modelling method to model the informatics learning domain. The basis of the method is: the feature concepts, the concept separation, feature variants and their interaction as well as the goal-driven processes. The models have been created using the well-known tools (FAMILIAR, SPLOT) ensuring models' quality and presenting essential characteristics for evaluation. As a result, a *general domain model* is obtained.
3. The proposed AGLO designing method covers two levels: the development of the concrete feature-based models, and their transformation into the meta-programming-based executable specification.
   - The *concrete models* are *extracted* from the general model. The specifications of the concrete models consist of the context and content models which are semantically related by relationships and constraints, and as well as by the priorities model. The latter enables to manage the complexity of the concrete model and creates the real pre-conditions to adapt the educational content.
   - The executable specification is the tool which generates the content *automatically* for the *different educational contexts*.
4. The specialized learning environments with integrated AGLO implement the visual transformation of a real task into its physical process, thus providing a high level of motivation and effective learning.
5. Cognitive complexity evaluation according to Miller's metrics creates pre-conditions to identify the relevant parameters sequence within specifications in order we could be able to manage complexity in designing and using AGLOs.
6. The pedagogical evaluation based on Bloom's taxonomy engagement levels enables to conclude that AGLOs are most effective at the following levels: viewing, constructing and presenting levels. The statistics obtained through experimental research over 3 years (2011-2014) shows the increase of learning improvement from 6 to 15 percent.

7. It has been identified the role of AGLOs in e-learning with respect to accepted standards and taxonomies. The following juxtapositions have approved benefits of our approach:

AGLOs in e-learning satisfy:

- All four learning object creating goals defined by WBITC (Web-Based Training Information Center), including reuse, interoperability, durability, accessibility.
- Four taxonomies of learning objects (Willey, Redeker, Finlay, Churchill).
- General and pedagogical characteristics of LO as defined by IEEE LOM.

AGLOs created for informatics education satisfy the following conditions:

- Six representative AGLOs fully cover programming basis of secondary school curricula (9-10 grades) and 70 percent topics of 11-12 grades.
- AGLOs along with created environments also cover the general attributes of the Kelleher's and Pausch's programming environments and tools taxonomy.

REFERENCES

[1]. Liber, O. (2005). Learning objects: conditions for viability. *Journal of Computer Assisted Learning*, *21*(5), 366-373.

[2]. McGreal, R. (Ed.). (2004). *Online education using learning objects*. Psychology Press.

[3]. 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).

[4]. Reusable learning objects. What are GLO's [viewed: 2014-04-16]. Online: http://www.rlo-cetl.ac.uk/whatwedo/glos/whatareglos.php.

[5]. Štuikys, V., & Damaševičius, R. (2012). Meta-Programming and Model-Driven Meta-Program Development: Principles, Processes and Techniques (Vol. 5). Springer.

[6]. [KM09] Koehler, M., & Mishra, P. (2009). What is technological pedagogical content knowledge (TPACK)?. *Contemporary Issues in Technology and Teacher Education*, *9*(1), 60-70.

[7]. [Gre08] Greer, D. (2008). The Art of Separation of Concerns [viewed: 2014-04-16]. Online: http://ctrl-shift-b.blogspot.com/2008/01/art-of-separation-of-concerns.html

[8]. [Har02] Harsu, M. (2002). *A survey on domain engineering*. Tampere University of Technology.

[9]. Coplien, J., Hoffman, D., & Weiss, D. (1998). Commonality and variability in software engineering. *Software, IEEE*, *15*(6), 37-45.

[10]. Capilla, R., Bosch, J., & Kang, K. C. (2013). *Systems and Software Variability Management*. Springer.

[11]. [Men09] Mendonça, M. (2009). *Efficient reasoning techniques for large scale feature models* (Doctoral dissertation, University of Waterloo).
[12]. [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.
[13]. Burbaite, R., Stuikys, 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.
[14]. RobotC – Improved Movement. (2007). Robotics Academy.
[15]. Kurilovas, E., & Dagienė, V. (2009). Multiple criteria comparative evaluation of e-Learning systems and components. *Informatica*, *20*(4), 499-518.
[16]. De Kock, A., Sleegers, P., & Voeten, M. J. (2004). New learning and the classification of learning environments in secondary education. *Review of educational research*, *74*(2), 141-170.
[17]. [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.

LIST OF PUBLICATIONS ON THE SUBJECT OF DISSERTATION

Publications in journals included into the Institute for Scientific Information (ISI) database

1. Burbaite, R., Stuikys, 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].

3. 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].

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

Articles referred in other international databases

1. Burbaite, R., & Stuikys, 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., Stuikys, V., Bespalova, K., & Paskevicius, P. Product variation sequence modelling using feature diagrams and modal logic. CINTI 2011: 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., Stuikys, 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].

Articles published in the other reviewed scientific publications

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: Nocolaus Copernicus University Press, 2013. ISBN 9788323130901. p. 101-111.

ACKNOWLEDGEMENTS

My sincerest gratitude to my supervisor Prof. Vytautas Štuikys for all his support, encouragement, patience, discussions and guidance during my PhD studies. I also acknowledge Prof. Robertas Damaševičius for his valuable feedback and for raising my research to new levels. My thanks also go to Kristina Bespalova for collaboration at work group.

Thanks to reviewers Ass. Prof. Danguolė Rutkauskienė, Prof. Valentina Dagienė and Prof. Aleksandras Targamadzė for their valuable advice and remarks which made my work more successful.

My special acknowledgement goes to administration of Panevėžys Juozas Balčikonis gymnasium for giving me opportunities to coordinate my studies with working activities, and community for helpfulness and understanding.

To my family for her love, understanding and care, and my students for being a great source of inspiration.

INFORMATION ABOUT THE AUTHOR OF THE DISSERTATION

Education:

1986-1991: gained master degree in physics at Vilnius University.
2001-2003: gained master degree in informatics at Šiauliai University.
2010-2014: doctoral studies in informatics at Kaunas University of Technology.

Work experience:

1991-1997: engineer, Joint Stock Company "Ekranas".
1997-2001: IT teacher, Panevėžys trading and service business school.
2001-2004: IT teacher, Panevėžys Kazimieras Paltarokas secondary school.
Since 2003: IT teacher, Panevėžys Juozas Balčikonis gymnasium.

Research interests: e-learning and formal methods in computer science education, generative learning objects, educational robotics.

E-mail: renata.burbaite@gmail.com

REZIUMĖ

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 [1]. Lankstaus mokymosi turinio kūrimas, atnaujinimas ir efektyvus taikymas išlieka vienu iš didžiausių iššūkių e-mokymosi tyrimuose. 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. 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.

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 (Slotkienė, 2009), lanksčiai pritaikomų (Kubiliūnas, 2009), generatyvinių (Rupšienė, 2009) MO kūrimo metodai, sukurtas MO metaduomenų taikomasis modelis (Kubilinskienė, 2012) ir MO kokybės ekspertinio vertinimo metodas (Sėrikovienė, 2013).

Reikšmingų pokyčių į MO sritį įnešė generatyvinio mokymosi objekto (angl. *generative learning object, GMO*) koncepcija, kurią pasiūlė Boyle su kolegomis [2]. Šie autoriai kildina GMO iš *generatyvinės lingvistikos* ir sieja su MO pakartotinio panaudojimo išplėtimu [3]. Pastarajame šaltinyje pateikiamas toks GMO apibrėžimas: „GMO yra aiškus vykdomasis mokymosi kūrinys (projektas), kuris sukuria tam tikrą mokymosi objektų klasę“.

Su GMO susiję daug neišspręstų arba nepilnai išspręstų problemų: (i) sistematizavimas, (ii) aukšto lygmens GMO modelių sudarymas, (iii) kūrimo automatizavimas, (iv) GMO perkeliamumas į įvairias aplinkas, (v) realus pritaikymas informatikos mokymuisi integruojant į mokymosi procesą specializuotas aplinkas, (vi) įvertinimo problemos ir kt., kurios buvo išnagrinėtos nepilnai ar visai nenagrinėtos. Dalis išvardintų problemų nagrinėjama šioje disertacijoje.

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 naujos kartos išplėstiniai GMO technologiniu požiūriu kildinami iš generatyvinės technologijos (ja laikoma heterogeninio metaprogramavimo technologija [5]). 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 konceptualiai, ir praktiškai šio tipo GMO geriausiai tinka informatikos (programavimo) mokymuisi, kadangi automatiškai generuojamas mokymosi turinys yra programos arba jų dalys.

Darbo objektas

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

Darbo tikslas

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

Iškeltam tikslui pasiekti sprendžiami tokie uždaviniai.

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. Integruoti išplėstinius generatyvinius mokymosi objektus į specializuotas heterogenines mokymosi aplinkas.
5. Eksperimentiškai įvertinti sukurtos metodikos panaudą pritaikant technologinius ir pedagoginius kriterijus.

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.

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.mokymesi turinio automatinio kūrimo požiūriu.

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.

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 sukuria 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.

UDK 004.9 : 37.091.3 (043.3)
SL344. 2014-07-21 2,5 leidyb. apsk.1. Tiražas 70 egz. Užsakymas xxx.
Išleido leidykla „Technologija", Studentų g.54, 44029 Kaunas
Spausdino leidyklos „Technologija" spaustuvė, Studentų g. 54, 51424 Kaunas