ISSN 1392–124X (print), ISSN 2335–884X (online) INFORMATION TECHNOLOGY AND CONTROL, 2016, T. 45, Nr. 4

Personal Generative Library of Educational Resources: A Framework, Model and Implementation

Renata Burbaitė[1], Vytautas Štuikys[1], Vida Drąsutė[2], Kristina Bespalova[1], Sigitas Drąsutis[3], Giedrius Ziberkas[1]

[1] Kaunas University of Technology, Department of Software Engineering,
Studentų 50-415, LT-51368 Kaunas, Lithuania,
e-mail: renata.burbaite@ktu.lt, vytautas.stuikys@ktu.lt, kristina.bespalova@ktu.lt, giedrius.ziberkas@ktu.lt

[2] Kaunas University of Technology, Department of Multimedia Engineering,
Studentų 50-401a, LT-51368 Kaunas, Lithuania,
e-mail: vida.drasute@ktu.lt

[3] Kaunas University of Technology, Department of Multimedia Engineering,
Studentų 50-416, LT-51368 Kaunas, Lithuania,
e-mail: sigitas.drasutis@ktu.lt

crossref http://dx.doi.org/10.5755/j01.itc.45.4.14910

Abstract. We discuss the Personal Generative Library (PGL) concept that covers models to describe some structural, functional and managerial aspects. Since the concept, to some extent, was realized in our previous research, in this paper, we focus more on the managerial aspects. In this regard, we propose the feature model-driven approach to implement those aspects using meta-programming techniques. First, we present the basic idea and theoretical background of the approach. Then we discuss the PGL architecture, its functionality and management procedures that are supported by the developed meta-programs. We outline the process of designing meta-programs through the series transformations of feature models. The main contribution of the paper is the implementation of the concept itself that enables, to some extent, to resolve the well known problems: library scaling and excluding synonymy in search. Furthermore, we have extended the potential of generative reuse (meaning a higher extent of automation as compared to the component-based reuse) by applying it not only at the library entity level (a great deal of PGL items are generative learning objects (GLOs)), but also at the whole library, i.e. its management level. Therefore, the approach enables the automatic formation of annotations for PGL entities and generation of queries to support managing procedures. We have approved the approach by presenting a case study and some experimental results.

Keywords: digital library; educational resources; content personalization; learning objects; generative re-usability; automatic library management.

1. Introduction

Digital libraries (also known as Learning Object (LO) Repositories, further, we use the abbreviation DL or DLs for the first term) play a significant role in providing educational resources for the huge e-learning communities worldwide. The main intention of DLs is to support wide-scale reuse by systemizing the accumulated knowledge in order it would be possible to share and reuse those resources in multiple contexts of use as efficiently as possible. Typically the structure and functionality of the DLs is predefined by the metadata standards, such as IEEE LOM [1], Dublin Core and CanCore [2]. Standards enable to create indeed the huge spaces of available resources within DLs. On the other hand, this also leads to serious difficulties and problems for users in searching the resources that fit best in each use case. This problem is known as semantic interoperability, meaning that any information given by the library creator should be understood correctly by the library users [3]. Another problem is the incompleteness of the metadata standards (e.g., LOs for game-based learning [4]). The internal structure of DLs (such as clustering of LOs) is not always relevant to the teacher's or learner's profile (such as knowledge level, learning style [5] [6] and learning activities [7]), DLs for specialized LO collections (for teachers with limited computer skills) [8]. Therefore, the quality of LOs themselves within DLs is also a big issue [9, 10].

With regard to DLs, the list of problems is by no means full. It will be extended in the next section. As our literature review shows, there is the extremely intensive research to overcome the existing problems and difficulties in this field. Among other concepts and approaches, however, the personalization of educational resources is at the focus now [11]. It is so, because the personal space encourages the reuse of learning materials and enables the construction of unique learning processes that suit the learner's needs best [12] [13]. In fact, personalization nominates the paradigm shift from teacher-centered learning to student-centered learning [14]. Therefore, the aim of this paper is to propose the concept of *a personal generative library* (further PGL) and implement it to support this paradigm change and respond to the existing challenges.

In this context, the term ‘personal’ should be understood as the possibility to have the common and individual resources for both the teacher and students for using, extending and sharing the educational material (i.e. LOs) in the given context. The resources may be partially taken from the external DLs and, to a larger extent, created by the teacher in advance to support the course specificity (such as teaching programming using educational robots), as well by the students themselves in the course of using PGL. The term ‘generative’ means at least two things: (1) automatic generation of LO annotations; (2) automatic generation of queries in searching LO within the PGL. The other interpretation of the term may be that the great deal of PGL resources are the generative LOs (GLOs). The main contribution of the paper is the implementation of the *personal generative library concept* that enables (a) at the *item level* to resolve to some extent the *library scaling* [15] and the synonym problem in searching library items [16] and (b) lifting of the generation concept of the *library item level* (i.e. separate GLOs) to the *whole library level*, using the meta-programming techniques [17]. Though so far there were attempts to automate the DL maintenance procedures (see e.g. [18]), to our best knowledge, the meta-programming approach at the library level has been applied for the first time. It enables to achieve the high degree of automation and flexible personalization.

The structure of the paper is as follows. In Section 2, we analyze the related work. In Section 3, we present the basic idea of the approach. In Section 4, we outline the background and provide a detailed description of the approach. In Section 5, we present a case study and experiments. In Section 6, we provide a summaring discussion and evaluation of the results. Finally, in Section 7, we formulate conclusions.

2. Related work

We categorize the related work into two groups: A - the general issues related to DLs (LO repositories); B – existing solutions to respond to the emerging DLs problems.

1. H.F. Cervone [19] in his paper emphasizes the lack of social functionality (such as enabling user comments) in developing DLs for LOs. V. Dagienė and E. Kurilovas [3] in their paper focuses on the semantic interoperability problem defining it as “an information given by one actor should be understood correctly by another actor”. J.-r. Park and Y. Tosaka [20] in their paper discusses the properties and issues in sharing the educational content. The content is regarded to be shareable via the repository, if it is validated against the standards, it is reusable in multiple contexts, editing tools, runtime environments, and learning management systems. A. Cohen et al. [12] emphasizes the need of the possibility and ability of personal expression in a shared open, global, and public space. Some papers discover the problems related to the metadata: (1) selection criteria and interoperability [21]; (2) metadata modelling [22], [23]; (3) LOs accessibility profiles in metadata models [24]; (4) the need of semantic metadata [25]; (5) the incompleteness of the metadata standards (e.g., LOs for game-based learning) [4]. The other papers are related to the LOs quality problems in DLs [9], [10]. Researchers propose the evaluation of the quality of LOs taking into account the opinion of the community [26], define relationships between metadata and LOs quality metrics in repositories [27], construct statistical profiles of highly-rated learning objects [28], use LO analytics for collections, repositories & federations [29].

LOs discovery (searching, selecting) from the different DLs is also a big problem. In this regard, the paper of A. Navarro et al. [30] suggests “a theoretical approach that permits the use of a single LOR for classifying and enriching LOs according to domain-dependent information schemas, which can be dynamically changed after their definition”. The paper of P. Northrup [31] focuses on finding of the personalized learning paths, the article of authors S. Baldiris et al. [32] offers a model for recommending LOs based on item response theory, A. Zapata et al. on their paper [33] suggests the hybrid recommendation method in the LOs search system. The papers of authors S. Graf et al. [34], S. M. B. Navaro et al. [35], S. M. Baldiris et. al [36] describe the distributed LOs metadata searching process and Micro-Context based Location Process (two different possible contexts: micro-context of the LO in repository structure and micro-context in the curricular structure). The study of T. N. Teixeira et al. [37] discusses the semantic search of LOs. I.-C. Hsu in his paper [38] proposes LO Finder and an intelligent LOM (LO metadata) shell based on Semantic Web technologies that enhance the semantics and knowledge representation of LOM. R. Kawase et al. in their paper presents the strategies to gather heterogeneous learning objects from the Web of Data based on using linked data principles (these principles describe methods of publishing structured data so that it can be interlinked and become more useful through semantic queries).

The next group of the papers focuses on using recommendation strategies to select suitable LOs from DLs: collaborative filtering recommendations inside DLs [40]; an algorithm to recommend LOs for students satisfying their pedagogical needs and learning goals (algorithm is based on LOs dynamic weight and similarity between LOs calculating) [41]; the authoring tools based on the use of proactive context-aware recommender [42]; evaluation and selection of group recommendation strategies for collaborative searching of learning objects [43].

2. Researchers that work in DLs problem domain suggest different solutions. Study of P. Rodríguez et. al. [44] and study of J. Vian et al. [45] proposes a multi-agent model for searching, recovering, recommendation and evaluation of LOs from Repository Federations. The study of S. Oton et al. [46] highlights that "most repositories are usually autonomous, that is, they work as portals that can be accessed through a Web-based interface, providing a search mechanism and a list of categories to conduct the search", and offer the possibility of making federated searches in distributed repositories from the original repository. A model proposed by V. L. Lopez et al. [47] for multi-label classification and ranking of LOs offers a methodology that „illustrates the task of multi-label mapping of LOs into types queries through an emergent multi-label space, and that can improve the first choice of learners or teachers". The paper of M. Lama et al. [48] presents an approach for the extraction and the annotation of the LO categories of the Universia DL that "has been transformed in a semantic LO repository following the principles of linked data". F. De la Prieta et al. in their paper [49] proposes an architecture based on a cloud computing paradigm that "will permit the evolution of current learning resource repositories by means of the cloud computing paradigm and the integration of federated search system".

D. G. Sampson, P. Zervas in their paper [50] analyzes the design and the implementation of DLs from the KMSs' (Knowledge Management Systems) perspective with an intention to support the management of implicit and explicit knowledge. The study [32] describes an architecture of the semantic DLs based on using the ontology of all e-learning artifacts and LOs, and on the representation of a domain through the logical language. The proposed architecture also includes the use of rules and concepts of semantic web services. C. Limongelli et al. in their paper [18] presents a comprehensive framework that consists of defining, retrieving and importing LOs for personalized courses. The suggested framework is partially implemented in the Moodle-based personalization system and supported the retrieval of LOs in a personalization context. The study of P. A. Rodríguez [14] focuses on the student-centered educational recommender system that combines content-based, collaborative and knowledge-based approaches. The paper of S. Tasso [51] deals with design and implementation of collaborative DLs based on the filing and retrieving distributed knowledge. The overview of the general, content, technical and quality characteristics of the existing DLs is given in [31].

We have not provided a more intensive review on technical aspects (such as feature-based modelling, meta-programming, domain-specific languages, and robotics in teaching and GLOs) to implement the library items (GLOs). The reader can learn more on those issues, for example, from [52-55].

As a result of the analysis, we are able to conclude: (1) DLs are indeed the powerful instrument and therefore is widely discussed topic to support the component-based reuse vision in the technology enhanced learning; (2) due to the complexity of this problem domain (in terms of its scope, interdependencies among the separate sub-domains, diversity of needs of different communities, etc.) there is a variety of issues and problems under intensive researching; (3) in the context of this paper, it is possible to exclude the two problems: the content personalization and more effective use of DL resources; (4) though the provided analysis by no means is exhaustive, nevertheless, it is possible to state the following: (i) so far the potential of generative reuse within this domain is yet in the infancy stage; (ii) there are little research efforts to introduce systematic studies to enforce more effective managing procedures of digital educational resources; (iii) this paper should be seen as our proposal for the explicit use of generative technology (such as meta-programming) and its benefits for this problem domain.

3. Basic idea of the approach

As it was stated, our aim is to introduce the new concept of personal digital library (PDL) and describe the approach to implement the concept. First, we describe the basic idea of the approach, presenting the main processes as it is outlined in Fig. 1. The initial data are the IEEE LOM standard for metadata. As the list of the metadata supports the wide scale of reuse, we need to specialize the list for the personalization purposes. The result of the process (it is identified by the number 1 in Fig. 1) is the subset of metadata to be used in PGL. The next activity is the transformation process, converting the selected metadata into the formal model, using the feature-based notion [52, 56, 57] (see also Section 4). The transformation process (denoted as 2*) is also applied to the user-oriented data needed for personalization. Therefore, we have two separate feature models that are to be combined into the one through aggregation (process denoted as 4 in Fig. 1).

The main requirement in constructing feature models is their consistency to be approved by using the adequate tools [53, 54]. The aggregation is to be performed on the correct models. The resulting feature

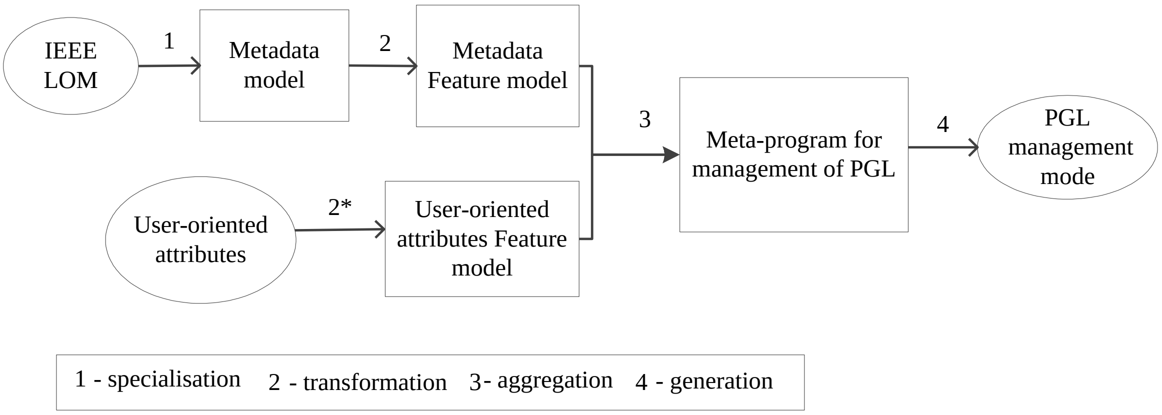


Figure 1. Process-based framework to explain main activities of the approach

model specifies the basic problem domain aspects (PDL just is treated as the problem domain in our case) needed in creating the meta-program for managing PGL. In fact, those aspects specify *metadata variability* implemented in the PGL. The external tool for managing the PGL variability is meta-programming techniques (we use heterogeneous meta-programming [17] here) are treated as the solution domain. To apply the techniques (see also Section 4), we need to introduce two languages (meta-language and target language). In our case, we use PHP in the role of the meta-language and MySQL as the target language. Note also that a large body of the library items are GLOs (or smart LOs) oriented to using robots in teaching. Those are implemented using RobotC as a target language. The meta-program is the specification that generalizes the possible modes and processes taken place in communicating with PGL. The external tool for managing the PGL variability is just the meta-program (meta-programs) we have developed along with the PGL. Finally, taking the concrete metadata values (they also treated as parameter values), it is possible to automatically generate PGL management mode on demand. In fact, this process (in Fig. 1 identified as generation numbered by 4) is the personalization process. We describe the final result, i.e. the PGL modes, in more detail, in Sections 4 and 5.

4. The background of the approach

First, we define the basic terms and their relationships treated as the theoretical background here. Note that the background is not concerned with the problem of how PDL is to be created, but rather on how it should be managed automatically as much as possible. More specifically, the background is concerned with the model-driven development of the meta-program for managing the PGL processes in order we could be able to achieve the prescribed aim of generative reuse. Next, we present more details (in Section 4.2 and 5) of the approach from the use perspective, focusing on the structural, functioning and manageable aspects of PGL.

4.1. Definitions of basic terms and relationships

Definition 1. *A digital learning library* is the set of digital educational resources created to provide the access, using the adequate technology, to the resources to support wide-scale reuse for learning communities worldwide (adapted from [58]).

Definition 2. *Metadata* is a data model, used to describe the other data, such as an LO. The purpose of LO metadata is to support the reusability of LOs, to aid discoverability, and to facilitate their interoperability (adapted [1, 2]).

Definition 3. *LO* is "any entity, digital or non-digital, which can be used, re-used or referenced during technology supported learning" [59]. In this paper, we consider digital entities. Therefore, LO is treated as an educational resource.

Definition 4. *Generative LO (GLO)* is "an articulated and executable learning design that produces a class of learning objects" [60]. Also *GLO* is the meta-level specification (meaning parameterized) to enable producing of concrete LOs on demand automatically according to the prespecified parameter values and the context of use (definition used in this paper).

Definition 5. *Personal Generative Library* (*PGL*) is the set of the entities organized so that it would be possible to provide the access and maintenance to the entities with regard to the personal needs, using the adequate generative tools (e.g. meta-program and meta-language processor, see Definitions 10, 14, 15).

PGL entities might be of the following type: (1) LO searched out from the external DL by the PGL user (typically teacher or knowledgeable student); (2) GLO created in advance by the teacher or designer; (3) smart GLO (meaning multi-stage GLO with the explicit parameter context for generating adapted GLOs [61]); (4) LO derived by the teacher or student from the scratch or specifications (2) or (3); (5) personal LO modified by the student after the derivation (generation) process.

Definition 6. *Feature* is a user-visible characteristic [62], qualitative property of the concept, or the functional requirement [63]. In the context of this paper, feature is either the entity of PGL, or an attribute of the entity.

Definition 7. *Feature model* (FM) is the specification to describe the commonality and variability aspects of a domain (in our case PGL) through the *feature types*, their *relationships* and *constraints*.

There are the *mandatory*, *optional* and *alternative* features. There are *parent-child* and AND- OR- and XOR- relationships. There are constraints of the type '*requires*' and '*excludes'* [56, 57].

A feature model is *abstract* if its features can be further decomposed into the other "smaller" features. A feature model is *concrete* if its leaf features are atomic features (atomic feature is its value).

Definition 8. *Transformation* is the process of changing one model into another according to the predefined rules.

We consider the following types of transformations in this paper:

- ***Specialization*** is such a transformation when the feature model A is transformed into the feature model B which is a sub-model of the model A [52].
- ***Aggregation*** is a composition of two or more models without common parts, which forms an output model [53]).
- ***Mapping*** is the kind of transformation when input and output models are represented by different languages.

Definition 9. *Meta-programming* is the high-level programming paradigm aiming at creating generalized programs – meta-programs [17] [64]. Here, it is also treated as *the solution domain* to implement the problem domain tasks.

Definition 10. *Heterogeneous meta-program (further meta-program)* is the executable generic program, described using at least two languages (meta-language and target language) in the same specification. Typically, the target (also domain) language serves for expressing the problem domain commonality aspects. The meta-language serves for expressing the problem domain variability aspects. Also meta-program is the generator to create *program instances automatically on demand* [17] [64].

Definition 11. *Parameter* is a syntax-driven entity within the interface or meta-body of the meta-program expressing domain variability independently from the semantics of the domain [64].

Definition 12. *Interface of a meta-program* is the set of parameters, their values and relationships among the values [64].

Definition 13. *Meta-body* is the specification to implement the functionality of the meta-program using a set of meta-language functions, where arguments of the functions are parameters, fragments of the target language (program) fragments, other functions of a combination thereof [64].

Definition 14. *Meta-language* is the subset of the functions (e.g., to specify an operation, alternatives and loops used in the mode of structured programming) of the general purpose programming language (PHP in our case) [17].

Definition 15. *Meta-language processor* is the tool to automatically generate an instance of a target program (in our case it is either the management program, or the PGL item derived from the GLO specification).

Definition 16. *Management & support meta-program* (shortly M&S MP) is the specification of the aggregated meta-program to support management processes of the PGL by generating the program instances to manage the selected mode of use.

Definition 17. *PGL* management process (mode) is any process taken from the list: (1) adding LO into PGL and annotating formation of it, (2) searching an entity in PGL, (3) deleting an entity from PGL, (4) changing attributes of the PGL entities. Each item of the list is based on using generated queries.

Therefore, the defined terms fall into two categories, belonging either to the problem domain (PD) or to the solution domain (SD). However, the process of solving the prescribed task (such as the development and then the use of the M&S MP for PGL in our case) is not a straightforward mapping of the PD entities onto the SD entities. Rather, it is the multi-level transformation process that includes various forms of transformation (T_1-T_4), which in Fig. 2 are represented graphically using the set of the Y-charts. We use Y-charts because they have three branches (left for PD, right for SD and vertical for the obtained solution) to visually express the essence of transformations. Note that Y-charts (a), (b) and (c) represent the whole meta-program development process through feature model transformations starting from the informal requirements statement for the PD (PGL in our case). The final solution is the M&S MP specification to automatically generate the needed management and support programs with regard to predefined parameter values.

4.2. The detailed description of the approach

In the previous section the framework to outline the processes for implementing PGL were described. In this section we propose the detailed approach of PGL.

The personal generative library provides the educational and managerial support in storing, updating and searching the content to realize the Computer Science (CS) curriculum objectives and tasks within the smart environment [61]. Currently we treat the library as a *personal internal database,* because it was

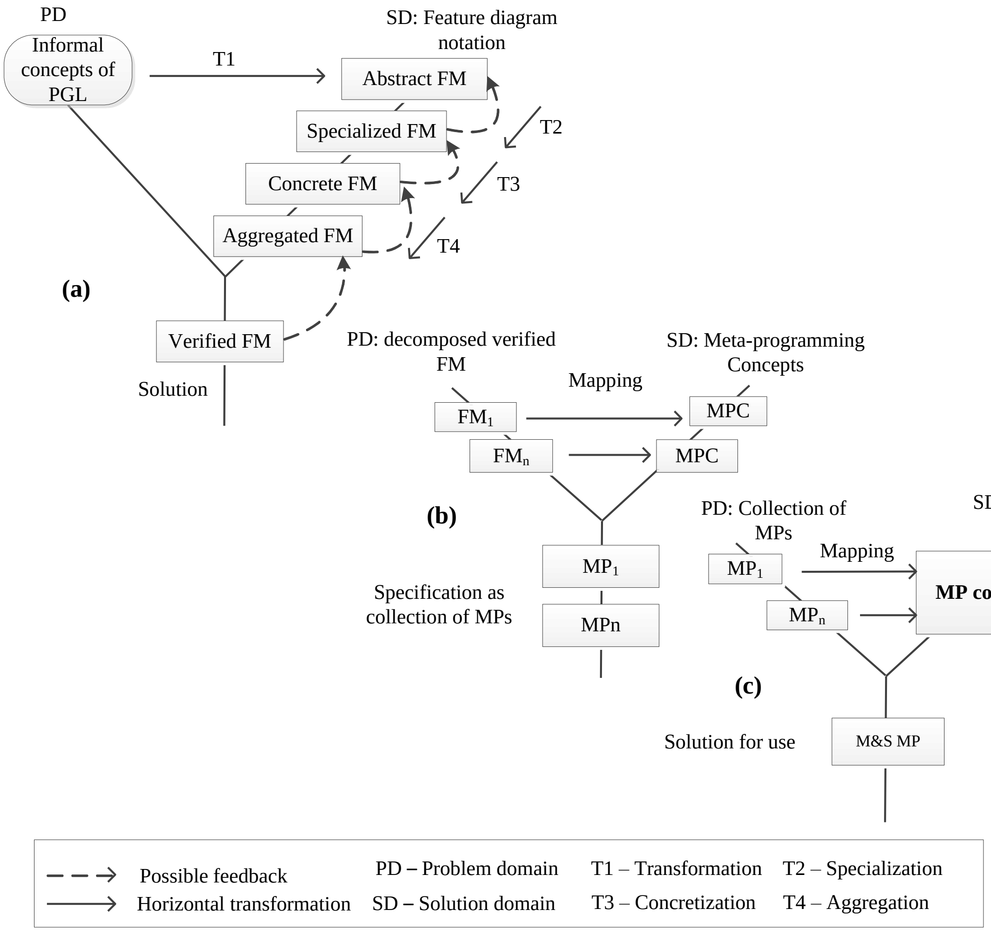


Figure 2. Y-chart based framework to outline the processes for implementing the approach

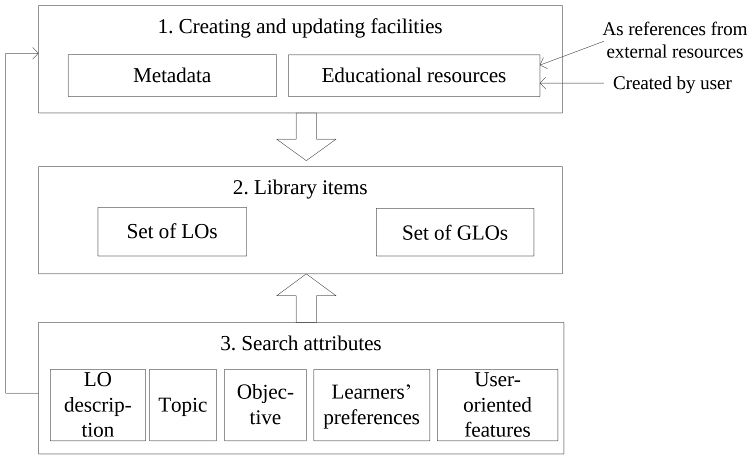


Figure 3. Architecture of the personal generative library (PGL)

created by the CS teacher (first author of this paper) to satisfy the *local needs* of the teaching institution only. However, we do not exclude the opportunity of extending the status of the library use in the future.

Currently the creator of the library is also responsible for maintenance and updating tasks. The users of the library are both teachers (there might also be other teachers as library users within the institution) and students. However, the access mode is different for the teacher-administrator, teachers as users and students. The teacher-administrator holds the highest priority: all accessibility functions are allowable. The overall structure of the library is shown in Fig. 3.

The entities within the library are those as defined by *Definition 5* (see Section 4.1). Among others, there are traditional learning objects to be obtained through linking to the external resources. They might be given in the form of text, pictures, video, etc. (e.g. to support the theoretical part of the topic, or it is the other additional material such as instructions to construct robots). Before the links being stored into the library, those links (LOs) first are enriched with metadata to enable their search procedure later, in the use time.

Note that Fig. 3 represents the common structure of the library. Using the structure, the user (i.e. teacher or student) can create his/her own PGLs. In Fig. 4, we show the interaction processes between users and PGLs. The processes 1 and 2 define the two-side interaction 'teacher-student'. The processes 3 and 4 (and also 5 and 6) define the two-side managing and supporting procedures 'user-PGL'. The procedures include the management processes (modes) as defined by *Definition 17* (see Section 4.1). The processes 7 and 8 define the modes of use the adequate educational resources from another user's PGL.

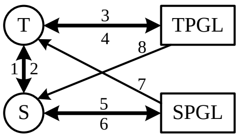


Figure 4. Interaction between users and PGLs (T- teacher, S - Student, TPGL –Teacher's PGL, SPGL – Student's PGL)

As the main focus of the paper is the development of meta-programs to support the automated management of PGL processes, we explain in more detail parts (b) and (c) of our framework (see Fig. 2). The development goes through subsequent model transformations in the following way.

The specialized concrete PD FM is transformed into the concrete SD FM using the abstract SD FM through mapping of corresponding PD items (e.g. variation points and variants) onto SD items (i.e. meta-program interface and body) using *Rules 1-4.*

Rule 1. The variation points of PD FM *correspond to parameters* within the meta-program or its model*,* and *variants* of a variation point *correspond to the parameter value.*

Rule 2. The parameter and their values are to be specified in the interface of the meta-program. The dependences among parameter values (if any) are to be *specified* in the interface too and expressed through the constraints (*requires*, *excludes)* to be implemented by the alternative meta-function (see Rule 3).

Rule 3. The SD model (i.e. meta-program model) is transformed into the executable meta-program specification (MPS) by performing the following actions: (i) selecting the concrete meta-language (ML) constructs (such as *if*-function of the meta-language, PHP in our case).; (ii) choosing the relevant target language (TL) scenario or scenarios (that depends on the task complexity, MySQL in our case); (iii) generalizing them using the ML constructs and PD variability model through coding and testing the specification.

Rule 4. The executable MPS is transformed into the application programs (such as those for PGL management in our case) via the following actions: (i) selecting a pre-programmed parameter values taken from the interface; (ii) processing (interpreting) MPS by the ML processor and generating the concrete target program; (iii) adapting it to the different use cases (if any) by the re-generation process.

We provide more details of our approach in Section 5.

5. A methodology of experiments and case study

We start describing our methodology of experiments assuming that the architecture of PGL is developed and its items (i.e. LO and GLOs) are loaded already. The experiments cover the development of meta-programs for solving the GLO managerial tasks only. Therefore, the methodology includes the following stages: (i) specification of the requirements of the tasks, using feature models; (ii) model-to-model transformation and verification, (iii) the development of meta-programs based on using transformation rules, (iv) generation of program instances from the meta-programs to solve managerial tasks; (v) obtaining the solutions of the tasks. The tasks include: (1) user-interface creation for all management modes (see *Definition 20*); (2) database table creation; (3) adding/deleting/selecting of the educational entities. This section contains two parts: Section 5.1 and Section 5.2. In Section 5.1, we explain the stages (i) and (ii) of our methodology. In Section 5.2, we present the remaining part of experiments we have carried out.

5.1. Results of modeling

Here we provide the characteristics of the verified aggregated FM (see Fig. 2(b)). Note that there are two versions of this model: one to support database creation (task 2) and the other to support selecting and deleting of an entity (task 3). Now we present the abstract metadata feature model (see *Definition 7*) based on the LOM IEEE standard (see Fig. 5). We have extracted from the standard only those attributes of metadata relevant to our purposes (see processes 1 and 2 in Fig. 1). For the model's detail, see *Definitions 7*, *8* in Section 4.1 and Legend in Fig. 5.

In Fig. 6, we present also an abstract feature model to define the user-oriented attributes (see process 2* in Fig. 1 and also Definitions 7 and 8 in Section 4.1). In Table 1, we present the characteristics of those versions. We can conclude: there are a large number of features of different types (see *Definition 8*), a huge number of valid configurations that defines the number of the possible implementations. In our approach, a valid configuration defines the annotation of an educational entity of the created PGL.

In Fig. 7, we present the modelling results obtained using the tool SPLOT [54]. Note that these results are aggregated models derived from the abstract models given in Fig. 5 and 6.

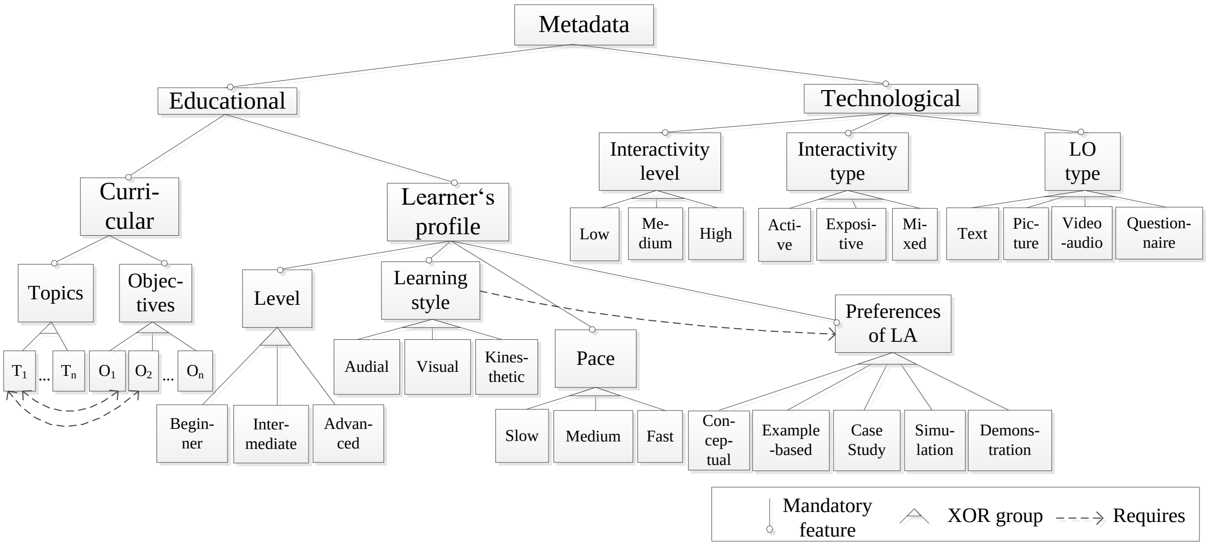


Figure 5. Abstract feature-based metadata model

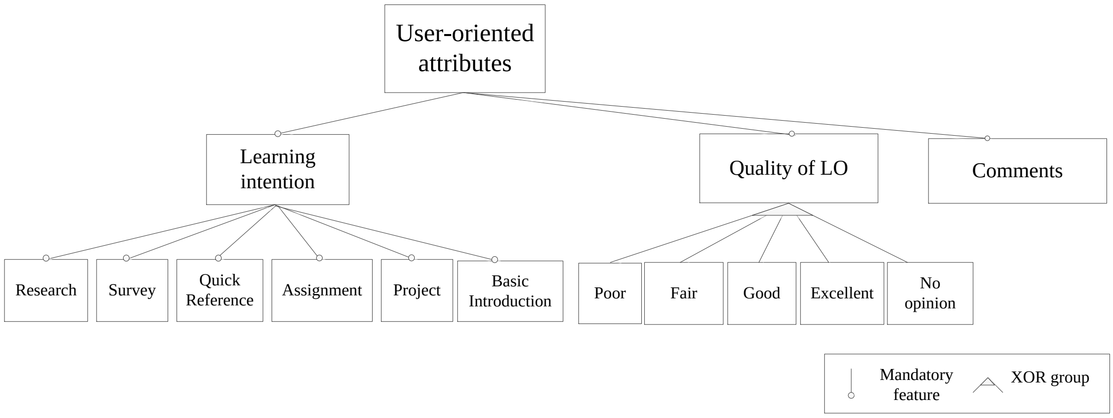


Figure 6. User-oriented attributes abstract feature model

Table 1. Characteristics of PGs aggregated FM obtained using FAMILIAR and SPLOT [53, 54]

No.	Mode Model metrics	DB creation, supplementing	Selection, deleting of the LO
1	# Features	94	94
2	# Mandatory features	14	14
3	# Optional features	10	10
4	# Core features	15	15
5	# XOR groups	24	28
6	# OR groups	4	0
7	#Cross-Tree Constraints	2	2
8	CTCR, % *	0.02	0.02
9	Tree Depth	8	8
10	Valid Configurations	1.23E12	624 490 560
11	Variability degree, % **	6.2085E-15	3.1529E-18
12	# Dead Features	None	None
13	Consistency	Consistent	Consistent

* 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.

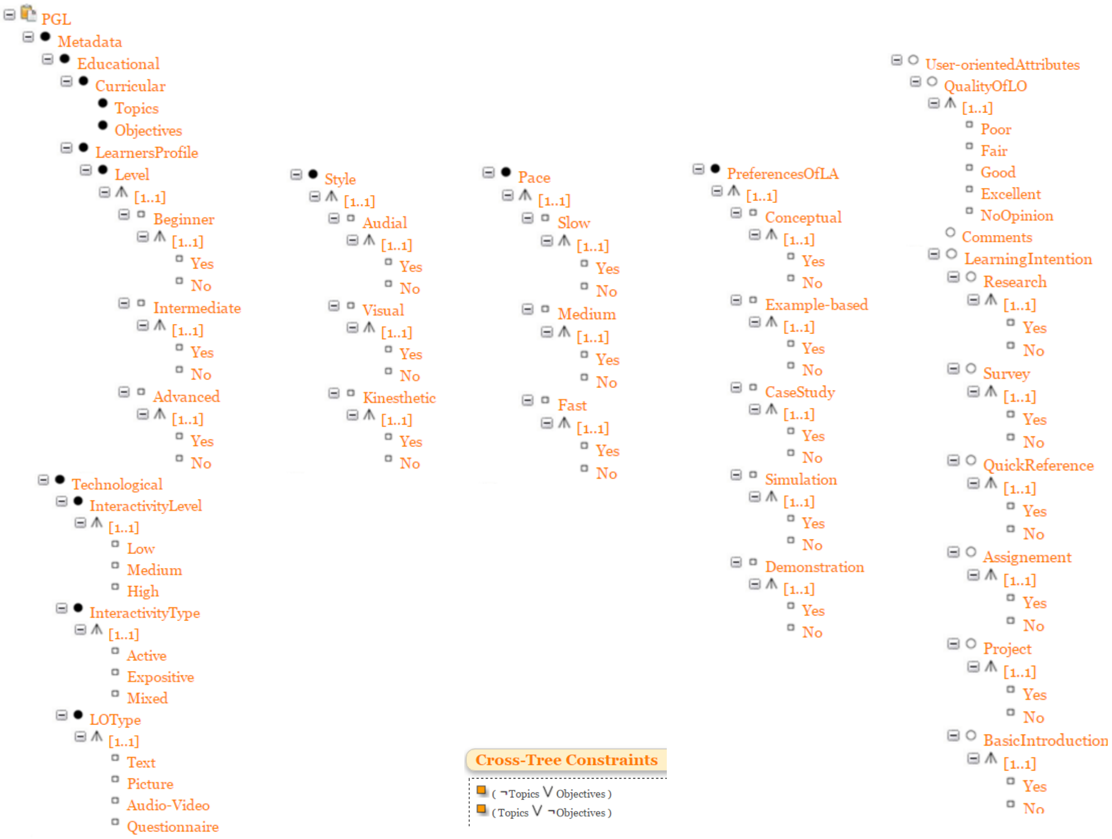
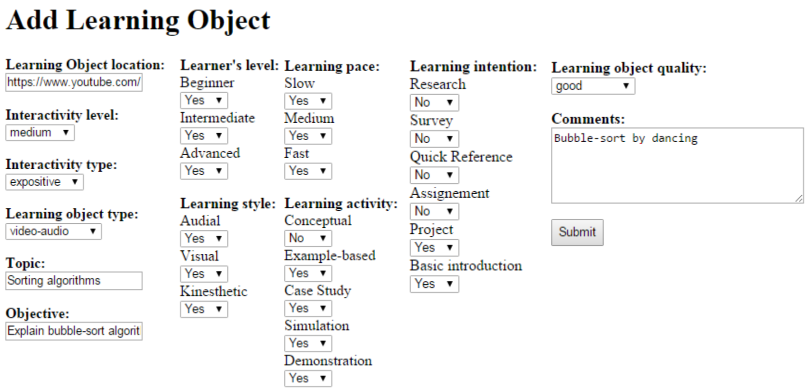


Figure 7. Concrete aggregated FM to describe annotations of the educational resources in PGL (model created using SPLOT [54])



Add Learning Object

Learning Object location: https://www.youtube.com/

Interactivity level: medium

Interactivity type: expositive

Learning object type: video-audio

Topic: Sorting algorithms

Objective: Explain bubble-sort algorit

Learner's level: Beginner Yes; Intermediate Yes; Advanced Yes

Learning style: Audial Yes; Visual Yes; Kinesthetic Yes

Learning pace: Slow Yes; Medium Yes; Fast Yes

Learning activity: Conceptual No; Example-based Yes; Case Study Yes; Simulation Yes; Demonstration Yes

Learning intention: Research No; Survey No; Quick Reference No; Assignement No; Project Yes; Basic introduction Yes

Learning object quality: good

Comments: Bubble-sort by dancing

Submit

Figure 8. A fragment of MP of the interface to add the new entity into PGL (see also the left branch in Fig. 2 (c))

5.2. A case study: results obtained by (from) the generated programs

Now we present the second part of the experiments we have carried out so far. It covers the remaining stages ((iii)-(v)) of our methodology. At the current state of the research, we have developed 3 separate meta-programs to support management activities. They include cases representing our researching tasks 1, 2 and 3 (see Section 5). Using those meta-programs, we are able to generate or derive the concrete programs automatically on demand for each task. In Fig. 8, we present a fragment of the interface for the meta-program (task 3).

We have tested the tasks experimentally. The experiments have shown that the functionality of the meta-programs was as expected. The results given in Fig. 9 demonstrate the correct functioning of the meta-programs.

Therefore the provided experiments (meta-program testing, generation of the managerial programs, the use of the programs to support the tasks and obtained results) have validated our approach.

6. Discussion and Evaluation

Digital Libraries (DLs) are powerful instruments to support wide-scale reuse of the educational content. Typically the basic item of DLs is the educational resource called learning object LO for distribution, sharing, use and re-use. Though there is a variety of types and models to represent LOs, generative LOs (shortly GLOs) nominate a specific kind of LOs. The concept of GLO is due to the contribution of Boyle, Leeder, Morales *et al.* [65], characterizing GLOs as “the next generation learning objects”. The Center for Excellence in the design, development and use of LOs in UK (shortly, RLO-CETL) defines GLO as “an *articulated* and *executable* learning design that produces a class of learning objects” [60]. In fact, the concept GLO means the shift from the *component-based reuse model* to the *generative reuse model*. The latter has much more possibilities for adaptation and personalization, especially in the case when meta-programming is used as a generative technology [17]. In this case, it is also possible to add new features for the meta-programming-based GLOs aiming at extending their capabilities for adaptation and personalization. Such features as pre-programmed context and staging of GLO parameters for adaptation combined with the advanced capabilities of educational robots enable us to recall GLOs and treat them as smart LOs [61].

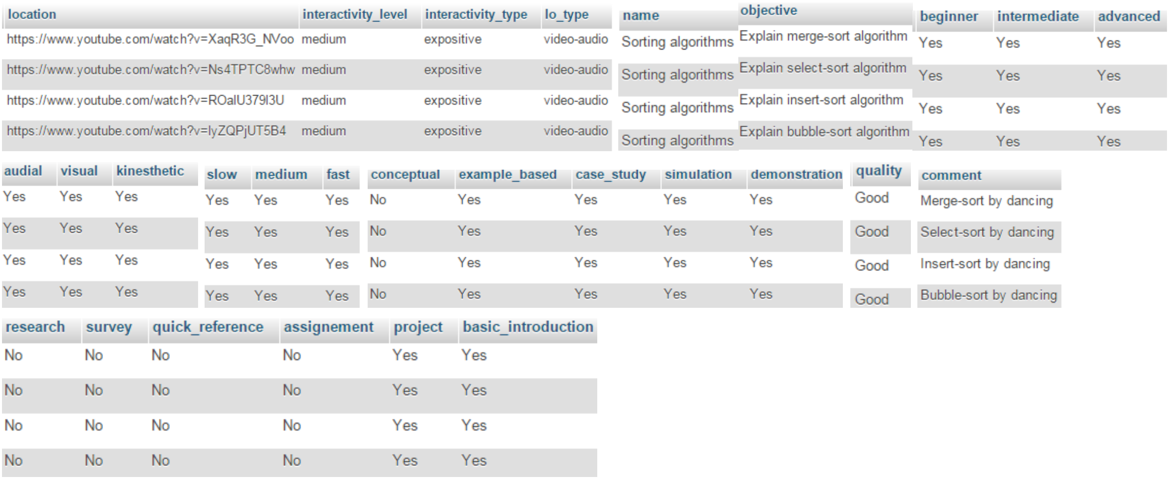
In this paper, therefore, we have extended the use of meta-programming techniques for managing the variability issues at *the library level* by introducing the concept of a *personal generative library* (PGL) to investigate and support the new capabilities for personalization of the educational content. Firstly, those capabilities are supported by *the types* of LOs the PGL contains. There are traditional LOs, for example, taken from the external DLs. There are one-stage meta-programming-based GLOs. They are context-aware multi-stage GLOs to support better adaptation and personalization. There are student created LOs derived from the both kinds of GLOs in the teaching process.

Secondly, the capabilities are supported by *the mode of use* the items of PGL along with the adequate tools (such as meta-program to maintain PGL and meta-language processor). For example, the teacher is able to derive from the context-aware multi-stage GLO a ‘smaller’ GLO being adapted for a particular group of students. Then each student of the group is able to derive from the smaller GLO a particular LO that fits to his/her context. The student is also able to repeat the process aiming to improve the content, if the generated result does not satisfy expectations, or even to modify the content manually.

Finally, the capabilities are also supported by *the mode of maintaining the content within* PGL. This is because of the *dynamic nature* of the maintaining procedure that has been implemented as a meta-program. The following modes of using the meta-program enable: (1) automatic generation of LO (GLOs) annotations in creating PGL; (2) automatic generation of queries in searching items within the PGL; (3) the items of PGL can be changed, modified, added the new ones using the adequate modes of the program.

Therefore, we summarize the contribution of the paper as follows. There are two basic results: (1) the concept, the models of the PGL and its implementation that enables to resolve, to some extent, two scientific and practical problems known as a library scaling [15] and synonymy in searching items from libraries [16]. (2) the novel maintaining procedure to support the PGL enabling automatic annotations and queries generation, as well as resulting in enhanced capabilities for personalization and adaptation of the PGL content.

The discussed approach has also some limitations: (1) currently PGL items are objects representing specific contents (such as course of programming using educational robots); (2) the students’ fraction within PGL is yet little as compared to the whole DPL content, where the fraction created by the teacher dominates now.



location	interactivity_level	interactivity_type	lo_type	name	objective	beginner	intermediate	advanced
https://www.youtube.com/watch?v=XaqR3G_NVoo	medium	expositive	video-audio	Sorting algorithms	Explain merge-sort algorithm	Yes	Yes	Yes
https://www.youtube.com/watch?v=Ns4TPTC8whw	medium	expositive	video-audio	Sorting algorithms	Explain select-sort algorithm	Yes	Yes	Yes
https://www.youtube.com/watch?v=ROalU379I3U	medium	expositive	video-audio	Sorting algorithms	Explain insert-sort algorithm	Yes	Yes	Yes
https://www.youtube.com/watch?v=lyZQPjUT5B4	medium	expositive	video-audio	Sorting algorithms	Explain bubble-sort algorithm	Yes	Yes	Yes

audial	visual	kinesthetic	slow	medium	fast	conceptual	example_based	case_study	simulation	demonstration	quality	comment
Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Good	Merge-sort by dancing
Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Good	Select-sort by dancing
Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Good	Insert-sort by dancing
Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Good	Bubble-sort by dancing

research	survey	quick_reference	assignement	project	basic_introduction
No	No	No	No	Yes	Yes
No	No	No	No	Yes	Yes
No	No	No	No	Yes	Yes
No	No	No	No	Yes	Yes

Figure 9. The obtained results (DB tables) by using meta-programs

7. Conclusions

1. The personal educational space encourages reuse of learning materials and enables the construction of unique learning processes to support student-centered learning. The *personal generative library* proposed in this paper is regarded as a tool to implement the learning paradigm more effectively as compared to the conventional digital library. It is so, because we use meta-programming, the powerful generative technology for implementing PGL, enabling (1) automatic generation of annotations of the library items; (2) automatic generation of queries in searching items within the PGL.
2. The benefits of PGL, however, are for both paradigms: teacher-centered learning and student-centered learning. It is so, because personalization and flexible content adaptation through automation are equally important in both cases, though the processes are different in objectives, scope, and results.
3. The proposed models and their implementation in creating PGL nominate the generalization of the educational content at the higher-level of abstraction as compared to the separate generative LO and existing approaches. Therefore, PGL can be thought of as a top-level content generator for management and personalization of the content on demand.
4. The scientific and practical value of PGL is that the approach enables either to eliminate the semantic interoperability problem in the whole (due to the explicit metadata at both the PDL management and GLO representation levels in the case when the library creator and user is the same actor), or to diminish its negative effect substantially in other cases due to the explicit metadata and the capabilities of rapid regeneration of queries (if needed).
5. Though we have focused on the education-based application, the obtained concepts, models and results are also applicable and may be interesting for a much larger community because the discussed problems are common in designing of many other systems.

References

[1] **L. Standard**. Draft Standard for Learning Object Metadata, IEEE P1484. 12/D4. 0, *Consultado en: (http://ltsc.ieee.org/doc/wg12/LOM_Wd4.doc) octubre-2003*, 2000.

[2] **D. Roy, S. Sarkar, S. Ghose.** A comparative study of learning object metadata, learning material repositories, metadata annotation & an automatic metadata annotation tool. *Advances in Semantic Computing*, 2010, Vol. 2, 103-126.

[3] **V. Dagienė, E. Kurilovas.** Design of Lithuanian digital library of educational resources and services: the problem of interoperability. *Information Technology And Control*, 2007, Vol. 36, No. 4, 402-411.

[4] **M. Hendrix, A. Protopsaltis, C. Rolland, I. Dunwell, S. de Freitas, S. Arnab, P. Petridis, J. LLanas.** Defining a metadata schema for serious games as learning objects. In: *eLmL 2012, The Fourth International Conference on Mobile, Hybrid, and On-line Learning,* 2012, pp. 14-19.

[5] **D. Domazet, D. Veljković, B. Nikolić, L. Jovev,** Clustering of learning objects for different knowledge levels as an approach to adaptive e-learning based on SCORM AND DITA. In: *The Third International Conference on e-Learning (e-Learning-2012),* 2012, pp. 27-28.

[6] **A. S. Sabitha, D. Mehrotra.** User centric retrieval of learning objects in LMS. In: *Computer and Communication Technology (ICCCT), 2012 Third International Conference on, IEEE,* 2012, pp. 14-19.

[7] **N. Ignatova, V. Dagienė, S. Kubilinskienė**. ICT-based learning personalization affordance in the context of implementation of constructionist learning activities. *Informatics in Education-An International Journal*, 2015, Vol. 14, No. 1, 51-65.

[8] **A. M. Fernández-Pampillón, E. Domínguez, J. M. Lahoz, D. Romero, I. de Armas, S. Palmaz, J. Arús**. A strategy for the inductive generation of learning objects in low-Tech contexts. In: *Proceedings of ECEL*, 2011, pp. 235-245.

[9] **C. Cechinel, X. Ochoa.** A brief overview of quality inside learning object repositories. In: *Proceedings of the XV International Conference on Human Computer Interaction, ACM*, 2014, pp. 83.

[10] **X. Ochoa.** Learnometrics: Metrics for learning objects. In: Proceedings of the 1st International Conference on Learning Analytics and Knowledge, ACM, 2011, pp. 1-8.

[11] **A. Juškevičienė**, **E. Kurilovas.** On recommending Web 2.0 tools to personalise learning. *Informatics in Education-An International Journal*, 2014, Vol.13, No. 1, 17-32.

[12] **A. Cohen, S. Reisman, B. B. Sperling**. Personal Spaces in Public Repositories as a Facilitator for Open Educational Resource Usage. *The International Review of Research in Open and Distributed Learning*, 2015, Vol. 16, No. 4, 156-176.

[13] **F. A. Dorca, R. D. Araujo, V. C. De Carvalho, D. T. Resende, R.G. Cattelan**. An Automatic and Dynamic Approach for Personalized Recommendation of Learning Objects Considering Students Learning Styles: An Experimental Analysis. *Informatics in Education-An International Journal*, 2016, Vol. 15, No. 1, 45-62.

[14] **P. A. Rodríguez, D. A. Ovalle, N. D. Duque**. A Student-Centered Hybrid Recommender System to Provide Relevant Learning Objects from Repositories. In: *Learning and Collaboration Technologies*, Springer, 2015, pp. 291-300.

[15] **T. J. Biggerstaff.** The library scaling problem and the limits of concrete component reuse. In: *Proceedings of the Third International Conference on Software Reuse: Advances in Software Reusability, IEEE,* 1994, pp. 102-109.

[16] **F. Neven, E. Duval.** Reusable learning objects: a survey of LOM-based repositories. In: *Proceedings of the*

Tenth ACM International Conference on Multimedia, ACM, 2002, pp. 291-294.

[17] **V. Štuikys, R. Damaševičius**. Meta-Programming and Model-Driven Meta-Program Development: Principles, Processes and Techniques, Springer Science & Business Media, 2012.

[18] **C. Limongelli, A. Miola, F. Sciarrone, M. Temperini,** Supporting teachers to retrieve and select learning objects for personalized courses in the Moodle_LS environment, in: Advanced Learning Technologies (ICALT), 2012 IEEE 12th International Conference on, IEEE, 2012, pp. 518-520.

[19] **H. F. Cervone**, Digital learning object repositories, *OCLC Systems & Services: International digital library perspectives*, 2012, 28, 1, 14-16.

[20] **D. Kinshuk**, **R. Jesse**, Mobile authoring of open educational resources as reusable learning objects, *The International Review of Research in Open and Distributed Learning*, 2013, 14, 2, 28-52.

[21] **J.-r. Park, Y. Tosaka**, Metadata creation practices in digital repositories and collections: Schemata, selection criteria, and interoperability, *Information Technology and Libraries*, 2010, 29, 3, 104-116.

[22] **A. Casali, C. Deco, A. Romano, G. Tomé,** An assistant for loading learning object metadata: An ontology based approach, in: Proceedings of the Informing Science and Information Technology Education Conference, 2013, pp. 77-87.

[23] **F. Campos, R. Braga, N. Santos, A. C. Souza, C. Rabello**, Expanding Access to Distance Learning using Learning Objects, *Education in a technological world: communicating current and emerging research and technological efforts*, 2011, 1.

[24] **S. Green, R. Jones, E. Pearson, S. Gkatzidou.** Accessibility and adaptability of learning objects: responding to metadata, learning patterns and profiles of needs and preferences. *ALT-J Research in Learning Technology*, 2013, Vol. 14, No. 1, 117-129.

[25] **C. Ullrich, R. Shen, K. Borau.** Learning from learning objects and their repositories to create sustainable educational app environments. In: *Advanced Learning Technologies (ICALT), 2013 IEEE 13th International Conference on, IEEE*, 2013, pp. 285-287.

[26] **C. Cechinel, S. da Silva Camargo, X. Ochoa, M. Sicilia, S. Sanchez-Alonso.** Populating learning object repositories with hidden internal quality information. In: *Proceedings of the 2nd Workshop on Recommender Systems for Technology Enhanced Learning (RecSysTEL 2012)*, Manouselis, N., Draschler, H., Verber, K., and Santos, OC (Eds.). Published by CEUR Workshop Proceedings, 2012, pp. 11-22.

[27] **D. Pons, J. R. Hilera, C. Pagés.** A set of quality metrics in learning object metadata. In: Proceedings of the 2012 International Conference on e-Learning, e-Business, Enterprise Information Systems, and e-Government (Worldcomp 12), 2012.

[28] **C. Cechinel, S. Sánchez-Alonso, E. García-Barriocanal.** Statistical profiles of highly-rated learning objects. *Computers & Education*, 2011, Vol. 57, No. 1, 1255-1269.

[29] **M.-A. Sicilia, X. Ochoa, G. Stoitsis, J. Klerkx.** Learning object analytics for collections, repositories & federations. In: *Proceedings of the Third International Conference on Learning Analytics and Knowledge, ACM*, 2013, pp. 285-286.

[30] **A. Navarro, A. M. Fernandez-Pampillon, C. Fernández-Chamizo, A. Fernández-Valmayor**. A meta-relational approach for the definition and management of hybrid learning objects. *Journal of Educational Technology & Society*, 2013, Vol. 16, No. 4, 258-274.

[31] **P. Northrup**. Learning Objects for Instruction: Design and Evaluation: Design and Evaluation. *IGI Global*, 2007.

[32] **S. Baldiris, R. Fabregat, S. Graf, V. Tabares, N. Duque, C. Avila.** Learning Object Recommendations Based on Quality and Item Response Theory. In: *Advanced Learning Technologies (ICALT), 2014 IEEE 14th International Conference on, IEEE,* 2014, pp. 34-36.

[33] **A. Zapata, V. H. Menendez, M. E. Prieto, C. Romero.** A hybrid recommender method for learning objects. In: *IJCA Proceedings on Design and Evaluation of Digital Content for Education (DEDCE)*, 2011, Vol. 1, pp. 1-7.

[34] **S. Baldiris, S. Graf, R. Fabregat, N. D. D. Méndez.** Looking for Contextualized Learning Objects to support Semi-automatic Learning Design Generation. In: *Workshop on Technology-Enhanced Learning and Living (TELL 2012)*, 2012, pp. 43-49.

[35] **S. Graf, R. Fabregat, N. D. D. Méndez**. Searching for and positioning of contextualized learning objects. *The International Review of Research in Open and Distributed Learning*, 2012, Vol. 13, No. 5, 76-101.

[36] **S. M. B. Navaro, J. L. Bacca, A. N. Rojas, J. C. Guevara, R. Fabregat.** LORSE: Intelligent meta-searcher of learning objects over distributed educational repositories based on intelligent agents. In: *Frontiers in Education Conference (FIE), 2011, IEEE*, 2011, pp. F1E-1-F1E-5.

[37] **T. N. Teixeira, F. Campos, R. Braga, N. Santos, E. Mattos.** BROAD Project: Semantic Search and Application of Learning Objects. *IEEE Technology and Engineering Education (ITEE)*, 2012, Vol. 7, No. 3, 23-32.

[38] **I.-C. Hsu**. Intelligent discovery for learning objects using semantic web technologies. *Journal of Educatio nal Technology & Society*, 2012, Vol. 15, No. 1, 298-312.

[39] **R. Kawase, M. Fisichella, K. Niemann, V. Pitsilis, A. Vidalis, P. Holtkamp, B. Nunes.** Openscout: harvesting business and management learning objects from the web of data. In: *Proceedings of the 22nd international conference on World Wide Web companion, International World Wide Web Conferences Steering Committee*, 2013, pp. 445-450.

[40] **C. Cechinel, M.-Á. Sicilia, S. Sánchez-Alonso, E. García-Barriocanal**. Evaluating collaborative filtering recommendations inside large learning object repositories. *Information Processing & Management*, 2013, Vol. 49, No. 1, 34-50.

[41] **S. Qamar, S. R. Bashir**. Towards the Recommendation of Highly Relevant Learning Objects to the Learners. *VAWKUM Transactions on Computer Sciences*, 2015, Vol. 5, No. 2, 19-23.

[42] **D. Gallego, E. Barra, A. Gordillo, G. Huecas.** Enhanced recommendations for e-Learning authoring tools based on a proactive context-aware recommender. In: *Frontiers in Education Conference,* IEEE, 2013, pp. 1393-1395.

[43] **A. Zapata, V. H. Menéndez, M. E. Prieto, C. Romero.** Evaluation and selection of group recommendation strategies for collaborative searching of learning objects. *International Journal of Human-Computer Studies*, 2015, Vol. 76, 22-39.

[44] **P. Rodríguez, V. Tabares, N. Duque, D. Ovalle, R. M. Vicari.** Multi-agent Model for Searching, Recovering, Recommendation and Evaluation of Learning Objects from Repository Federations. In: *Advances in Artificial Intelligence–IBERAMIA 2012*, Springer, pp. 631-640.

[45] **J. Vian, R. L. R. Campos, C. E. G. Palomino, R. A. Silveira.** A multiagent model for searching learning objects in heterogeneous set of repositories. In: *11th IEEE International Conference on Advanced Learning Technologies (ICALT)*, IEEE, 2011, pp. 48-52.

[46] **S. Otón, A. García, A. Ortiz, E. García, J. R. Hilera, L. de-Marcos, R. Barchino, S. M. de Dios**. Developing Distributed Repositories of Learning Objects. *INTECH Open Access Publisher*, 2012.

[47] **V.F. López, F. de la Prieta, M. Ogihara, D.D. Wong.** A model for multi-label classification and ranking of learning objects. *Expert Systems with Applications*, 2012, Vol. 39, No. 10, 8878-8884.

[48] **M. Lama, J. C. Vidal, E. Otero-García, A. Bugarín, S. Barro**. Semantic linking of learning object repositories to DBpedia. *Journal of Educational Technology & Society*, 2012, Vol. 15, No. 4, 47-61.

[49] **F. De la Prieta, A. B. Gil, A. J. S. Martín, C. Zato**. Learning Object Repositories with Federated Searcher over the Cloud. In: *Methodologies and Intelligent Systems for Technology Enhanced Learning*, Springer, 2014, pp. 93-100.

[50] **D. G. Sampson, P. Zervas**. Learning object repositories as knowledge management systems. *Knowledge Management & E-Learning: An International Journal (KM&EL)*, 2013, Vol. 5, No. 2, 117-136.

[51] **S. Tasso, S. Pallottelli, R. Bastianini, A. Lagana.** Federation of distributed and collaborative repositories and its application on science learning objects. In: *Computational Science and Its Applications-ICCSA 2011*, Springer, pp. 466-478.

[52] **T. Thüm, D. Batory, C. Kästner.** Reasoning about edits to feature models. In: *IEEE 31st International Conference on Software Engineering, ICSE 2009*, pp. 254-264.

[53] **M. Acher, P. Collet, P. Lahire, R.B. France**. Familiar: A domain-specific language for large scale management of feature models. *Science of Computer Programming*, 2013, Vol. 78, No. 6, 657-681.

[54] **M. Mendonca, M. Branco, D. Cowan.** SPLOT: software product lines online tools. In: *Proceedings of the 24th ACM SIGPLAN Conference Companion on Object Oriented Programming Systems Languages and Applications*, ACM, 2009, pp. 761-762.

[55] **V. Stuikys, R. Burbaite, R. Damasevicius**. Teaching of computer science topics using meta-programming-based GLOs and LEGO robots. *Informatics in Education*, 2013, Vol. 12, No. 1, 125-142.

[56] **K. Czarnecki, S. Helsen, U. Eisenecker.** Staged configuration using feature models. In: *Software Product Lines*, Springer, 2004, pp. 266-283.

[57] **D. Batory**. Feature models, grammars, and propositional formulas. In: *SPLC*, Springer, 2005.

[58] **C. Holden**. From local challenges to a global community: learning repositories and the global learning repositories summit. *The Academic ADL Co-Lab*, 2003.

[59] **I. L. T. S. Committee**. Draft standard for learning object metadata. 2002, online access on https://biblio.educa.ch/sites/default/files/20160613/lom_1484_12_1_v1_final_draft.pdf.

[60] Reusable learning objects. What are GLO‘s?, in http://www.rlo-cetl.ac.uk/whatwedo/glos/whatareglos.php.

[61] **V. Štuikys**. Smart Learning Objects for Smart Education in Computer Science: Theory, Methodology and Robot-Based Implementation. *Springer*, 2015.

[62] **K. C. Kang, S. G. Cohen, J. A. Hess, W. E. Novak, A. S. Peterson.** Feature-oriented domain analysis (FODA) feasibility study. DTIC Document, 1990,

[63] **K. Czarnecki, U. W. Eisenecker.** Generative programming by J. Hartmanis, and J. van Leeuwen (Eds.), 2000.

[64] **V. Štuikys, K. Bespalova, R. Burbaitė.** Refactoring of Heterogeneous Meta-Program into k-stage Meta-Program. *Information Technology And Control*, 2014, Vol. 43, No. 1, 14-27.

[65] **D. Leeder, T. Boyle, R. Morales, H. Wharrad, P. Garrud.** To boldly GLO-towards the next generation of Learning Objects. In: *World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education*, 2004, pp. 28-33.

Received May 2016.