The 11th International Scientific Conference
eLearning and software for Education
Bucharest, April 25-26, 2015
10.12753/2066-026X-15-000

A Survey on Generative Learning Objects

Ciprian-Bogdan Chirila
Department of Computer Science and Software Engineering,
University Politehnica Timişoara, V. Pârvan Blv. no. 2, Timişoara, Romania
chirila@cs.upt.ro

***Abstract**: Learning objects (LOs) is a field of e-learning which is open for research and has a lot of potential in the creation, adaptation, testing, generation of learning content. There are standards that describe LOs in a general manner like IEEE LOM, SCORM or Dublin Core. Among these LOs there is a special kind of LOs named generative learning objects (GLOs) which present a higher degree of reuse and are considered to be the second generation LOs. The principles behind the GLO concept are taken from software engineering namely design for reuse and design with reuse. There are several approaches regarding GLOs. The first approach belongs to the pioneers of the concept and is based on reusable pedagogical patterns implemented in tools like GLOMaker. A second approach belongs to a group of researchers from the University of Vilnius which rely their implementation on parameter generated software inspired from software product lines and metaprogramming. The approach is targeted towards learning computer science disciplines, namely programming using robots. In our approach, the third one, we use metaprogramming in the context of models which we consider to be more expressive and flexible but demand a higher level of qualification for the GLO author. GLOs are not heavily used in practice because of their complexity, lack of supporting tools, testing difficulties. In this paper we intend to identify the main issues with GLOs and to discuss the main challenges, possible solutions and limitations. Understanding better GLOs problems may help in ameliorating those using new tools, software wizards and in increasing their use in practice in current learning management systems.*

Keywords: *generative learning objects, meta-programing, reusable didactic patterns*

I. INTRODUCTION

The development of Internet changed the way people approach learning: learning materials are deliverable online, they are accessible at the same time by different students, and they can be changed easily in order to create new versions, complex materials are decomposable in parts while parts can be rebuilt to a whole.

In this paper we will analyze several ideas regarding the creation, use and reuse of second generation learning objects (LO) namely generative learning objects (GLO).

Simple LOs have several definitions in the literature. Apparently, the term was coined by Wayne Hodgins in 1994 after the name given to the working group called “Learning Architectures, API’s and Learning Objects” while the concept was invented earlier in 1967 by Gerard [16].

There are also other denominations used in the literature for LOs like: knowledge object, components of instruction, instructional component, pedagogical documents, educational software components, online learning materials or simply resources.

The name has also its roots in the world of object-oriented programming where objects are designed for reuse purposes, categorized by metadata, categorization that facilitates searching and reusing them [7].

There are several types of stakeholders in the world of LOs like: instructional designers, businesses, schools. In order to increase reusability and interoperability of LOs several standards were developed by organizations like: IEEE Learning Technology Standards Committee (LTSC) [21], Advanced Distributed Learning (ADL) [1], Alliance of Remote Instructional Authoring and Distribution Network for Europe (ARIADNE) [3], Global Learning Objects Brokered Exchange (GLOBE) [17], Instructional Management System (IMS) [IMS2000] etc.

According to Cisco Systems a LO is a collection of content, practice, assessment items combined in order to serve to a single learning objective. The definition of [20] considers that a LO should have: i) a learning objective; ii) a unit of instruction that teaches the objective; iii) a unit of assessment.

In the definition of [18] a LO is an entity that can be either digital or non-digital used for learning, education and training. Its purposes are set to instructional design, development, and delivery while its potential is in reusability, generativity, adaptability, scalability.

The Educational Objects Economy [14] project define a LO to be a Java applet as an object oriented approach to computer assisted instruction. Other LO definitions include several aspects like: i) to be designed for using them in multiple training contexts; ii) to increase the flexibility of training; iii) to make the multiple course updating process much easier; iv) updating a part of a LO should reflect in all the courses using that LO.

In [27] these definitions are analyzed and criticized while the form of - any digital resource that supports learning and can be delivered across the network in large or small sizes – is considered the optimal one.

Several standards were created: Learning Object Metadata (LOM) and Sharable Content Object Reference Model (SCORM). LOM is a metadata format including a packaging wrapper around the any LO. The metadata is used for searching and retrieving LOs and for interoperability across different learning management systems. SCORM reference model is a higher level framework for creating reusable learning materials.

The paper is structured as follows. In section II we discuss general aspects regarding generative learning objects. In chapter III we present and analyze the pedagogical template based approach for generative learning objects. In chapter IV we present and analyze the parameter based generation or software product lines approach for generative learning objects. In chapter V we present the server pages like based approach for generative learning objects in comparison with the other approaches. Chapter VI concludes and set the perspectives.

II. GENERATIVE LEARNING OBJECTS PRINCIPLES

Generative learning objects (GLO) are a special category of LOs with a higher degree of reuse and repurposing. They are considered to be the second generation learning objects [5] where the reuse is reoriented towards the pedagogical pattern rather than towards the content of the LO. The higher level content is thus separated from the lower lever structure. As an informal definition, a GLO is considered to be the basic unit of reuse for the pedagogical pattern.

The works of [4] and [5] are the seminal papers for the analyzed concept. In [4] are set the principles inspired from software engineering that are at the foundation of GLOs, namely: modularization, cohesion and decoupling. The goal of software engineering is to design, develop and maintain large software systems, while it is considered that 70% of the allocated time is spent on maintenance.

Modularization helps dividing the software system into subsystems which are easier to maintain. LOs and GLOs fit exactly to the subsystems idea. The principle of cohesion, which states that each unit should have only one functionality, is translated into the LOs world that each LO should have only one learning objective or learning goal. The principle of decoupling states that one unit should have as less as possible bindings to other units. In the context of LOs this means that LOs should not depend on each other, should not reference external resources, thus LOs should be self-contained and it can be used independently of the other.

The idea of compound object which contain several standalone objects is set as a principle for designing attractive and pedagogically rich LOs. An example of a compound LO is given in this sense containing the core explanation and text based examples combined with expansion links to additional resources to be used in case the learner needs additional explanations on the presented topic.

The idea of this concept started from the principles of software engineering combined with pedagogy.

III. THE PEDAGOGICAL TEMPLATE APPROACH

The first approach of GLO [5] defines at the pragmatic level the GLO pattern as an "intention structure" of the tutor. They develop a GLO type based on three phases: introduction, understanding and construction. The understanding phase is based also on three stages: engage, apprehend and comprehend.

An example of a learning object is provided following these principles. The learning object is expressed using frames. The first one is an introductory one for the concept to be presented and is form of invitation for the learner. The engaging function is implemented using an analogy between a familiar concept and the new concept to be explained. The apprehension frames may contain the specific details of the new concept assisted by animation if possible. The comprehension stage is implemented through details expressing animation based on a step by step approach.

The object ends with a scaffolded construction exercise. These LOs are considered to be a source from which design patterns in the sense of [2,15] can be extracted [19] containing three sections: context, problem and solution.

From the technical point of view the implementation is based on an authoring tool which produces a representation of the GLO expressed in XML. The instance is played in an interpreter player program which interacts with the learner. The authoring tool expresses the idea of executable pedagogical pattern that the tutors can create or modify existing ones. Animation resources seem to be referenced in the model file with restricted control on the behavior.

In [Han2009] a similar approach is used to build interactive learning objects which seem very similar to GLOs. The LO phases are based on Bloom taxonomy of cognitive layers, namely: remember, understand, apply, analyze. They use an XML based formalism to express the model containing ActionScript 3 code which is further compiled with the Adobe Flex technology to generate Flash (.swf) files that can be used in most of the popular Internet browsers. The animations used in the LOs are parameterized to a certain degree and the generic animations movements are exposed through an API.

In [22] the principles of [4] are applied in the implementation of a Depreciation GLO in the field of accounting. In this case the pedagogical pattern consists in: i) title in the heading bar; ii) navigation controls in the bottom of the page; iii) an instructional panel which contains text from an XML file located in the right part of the frame; iv) the main content panel which loads and displays external files images, videos and flash animations which can be modified externally. The technologies used with this approach are chosen among the free ones for cost saving measures namely: Flash, XML, PHP.

The steps of a GLO design are presented in [6] using the GLOMaker2 tool. The tool contains a planner stage where pedagogic sequences can be created or altered. There is a palette containing three categories of elements: i) orient; ii) do task and iii) reflect from which several can be selected and sequenced thus resulting the pedagogical sequence. This sequence can be labeled with semantic explanations. The presented example is a GLO named Evaluating Multiple Interpretations (eMI) and it contains the following nodes: i) what topic; ii) main task; iii) access views; iv) multiple choice; v) reflect. Each node consists in an editable page with text and other media resources enabled by the tool. Thus, the GLO is filled with content and becomes a regular LO ready to be consumed by the learner. The designed GLO can be filled with another content resulting in a different LO following the same pedagogical pattern set in the template.

IV. THE SOFTWARE PRODUCT LINES APPROACH

This GLO approach is present in the works of Stuikys [23,24,25,26] and Damasevicius [12,13,8]. They propose a GLO model based on input knowledge which comes from the exterior (like prerequisites) and output knowledge produced by the learning component.

Structurally their LO model has: i) knowledge based interface; ii) knowledge based body; iii) declarative part; iv) procedural part; v) contextual part; vi) managerial part. The GLO has the same structure with abstracted parts.

Their GLO models are based on feature diagrams which abstracts the subject domains. These domains are analyzed for commonality and variability.

As generative techniques they use meta-programming implemented in Open PROMOL meta-language. The meta-programming language: i) has text manipulation features; ii) use a functional approach; iii) its syntax is similar to other common programming languages. The output of the generation is plain text and HTML to be consumed in Internet browsers.

They experimented the GLO approach on computer science disciplines, namely on Boolean algebra, sorting algorithms, computer programming basics in the context of LEGO robots and Arduino systems.

V. THE SERVER PAGES LIKE APPROACH

In our approach [9,10,11] the GLO concept as previously defined is located at the level of general competence in taxonomy based on domain, general competence, specific competence and variable. For example we can consider as domain the data structures discipline. In this context the data structure tree concept together with its properties are modelled as a general competence, while its properties are modeled at the specific competence level and each property is the responsibility of a variable. Variables contain actions which deal with the variable specific learning detail.

Actions are based on textual templates which models the pedagogical and the learning content. In the templates we have inserted symbols (parameters) whose values change their contents to different learning scenarios at instantiation time. An action has a structure containing the following sections: i) a scenario section dedicated to tutors where the main idea of the learning action is explained; ii) a theory section where theoretical part of a concept can be explained together with an example; iii) a sentence section where the learner task is explained; iv) an answers section where the available responses are displayed to the learner for possible interactions (selection, writing, ordering etc); v) a feedbacks section which are shown selectively after the learner answered the sentence. All sections except for the first may use symbols, thus their generated content based on their values is different from one running instance to the other. The generated answers enable automatic result assessment.

In our approach the symbols can be set by the tutor formulas including random values generated by a pseudo random number generator available on the running platform. Thus all actions have valid values for their symbols at instantiation. The instantiation consists in executing the formulas in order to initialize the symbols and to select a view formed out of one or more frames to show the action content to the learner.

The approach was implemented in a dialog games for mathematics [9,11] in the context of a large educational project. Another use case of our approach was implemented in generating tests to be solved on paper for computer science disciplines [11] used in practice for 4 years. The templates were implemented with C programs having variables as model parameters producing HTML and LaTex source code compiled to PDF format for printing. A few examples were designed for generating questions for training in medical disciplines [Chi2014r] in order to show the generality that the approach.

The current implementation of the approach is based on a simple interpreter written in JavaScript reading action models expressed in XML format.

VI. CONCLUSIONS AND PERSPECTIVES

The advantages of the GLO derive clearly from the software engineering, namely object-oriented technology principles. The content of OOP classes is data and code description, while the content of generative learning objects is generic learning material. OOP objects are instances of OOP classes while LOs are instances of GLOs.

In this paper we reviewed the principles of dynamic and reusable LOs focusing on tree types of GLOs: pedagogical pattern (1), software product lines (2), server pages like (3). The pedagogical pattern approach seems to be at the higher level of generality than the other two models. The last two seem to fit more to the micro-contexts where detailed learning objectives are targeted. Approaches (2) and (3) were exercised more in the domain of computer science learning.

From the generative point of view approach (1) uses templates while approaches (2) and (3) use meta-programming. Approach (1) uses a proprietary script allowing GLO designer control through parameters, while approach (3) uses a JavaScript framework based on the compiling techniques concept of symbol table, allowing GLO designer directly to the script that is kept at a lower level of complexity.

Regarding content variability, approaches (1) and (2) do not support it while approach (3) relies on formulas based on random numbers generation.

From the technical point of view all approaches use the XML markup language for the representation of GLOs, thus they are human readable and can be easily modified and adapted. Approach (2) is based on a system on chip design language, namely Open Promol.

From the tool support point of view and in order to simplify the creation, editing and testing of GLOs approach (1) has an online, publicly available, published tool named GLOMaker, while approaches (2) and (3) seem to be in experimental development phases.

Approach (3) could be refined more by adding leveling parameters in order to control the level of difficulty of GLOs and instanced LOs for example.

In perspective we consider that the wide spreading of GLOs depend much on their integration in the popular LMSs assisted by tool support.

Acknowledgements

This paper is supported by the Sectorial Operational Program Human Resources Development (SOP HRD), financed from the European Social Fund and by the Romanian Government under the project number POSDRU/159/1.5/S/134378.".

References

[1] Advanced Distributed Learning Network website, http://www.adlnet.gov, 2015.

[2] C. Alexander, S. Ishikawa, M. Silverstein - A pattern language: Towns, buildings, construction. Oxford: Oxford University Press, 1977.

[3] Alliance of Remote Instructional Authoring and Distribution Networks for Europe website. http://http://www.ariadne-eu.org, 2015.

[4] Tom Boyle - Design principles for authoring dynamic, reusable learning objects. Australian Journal of Education Technology, vol. 19, no. 1, pp. 46-58, 2003.

[5] Tom Boyle - The design and development of second generation learning objects. Invited talk at Ed Media 2006, World Conference on Educational Multimedia, Hypermedia and Telecommunications, Orlando, Florida, June 28, 2006.

[6] Tom Boyle, Claire Bradley - User Guide for the GLO Maker 2 Authoring Tool, http://www.glomaker.org, 2009.

[7] Bratina, T., Hayes, D., & Blumsack, S. - Preparing teachers to use learning objects. In The Technology Source, November/December 2002.

[8] Renata Burbaite, Kristina Bespalova, Robertas Damasevicius, Vyutatas Stuikys - Context Aware Generative Learning Objects for Teaching Computer Science, International Journal of Engineering Education, vol. 30, no. 4, pp. 929-936, 2014.

[9] Ciprian-Bogdan Chirila - A Dialog Based Game Component for a Competencies Based E-Learning Framework, In proceedings of SACI 2013 8-th IEEE International Symposium on Applied Computational Intelligence and Informatics, pp. 055--060, Timisoara, Romania, May, 2013.

[10] Ciprian-Bogdan Chirila - Educational Resources as Web Game Frameworks for Primary and Middle School Students, In proceedings of eLSE 2014 International Scientific Conference eLearning and Software Education, Bucharest, Romania, April, 2014.

[11] Ciprian-Bogdan Chirila - Generative Learning Object Assessment Items for a Set of Computer Science Disciplines, In proceedings of SOFA 2014 6-th International Workshop on Soft Computing Applications - Advances in Intelligent and Soft Computing, Springer Verlag, ISSN 1867-5662, Timisoara, Romania, July, 2014.

[12] Robertas Damasevicius, Vyutatas Stuikys - On the Technological Aspects of Generative Learning Object Development, Third International Conference on Informatics in Secondary Schools - Evolution and Perspectives (ISSEP 2008), pp. 337-348, Torun, Poland, July, 2008.
[13] Robertas Damasevicius, Vyutatas Stuikys - Specification and Generation of Learning Object Sequences for e-Learning Using Sequence Feature Diagrams and Metaprogramming Techniques, In proceedings of 2009 9-th Intenrational Conference on Advanced Learning Technologies, 2009.
[14] Educational objects economy website. Available: http://www.eoe.org/eoe.htm, 2002.
[15] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides - Design Patterns: Elements of Reusable Object-Oriented Software, Pearson Education, 2004.
[16] Ralph W. Gerard - Shaping the mind: Computers in education, In National Academia of Sciences, Applied Science and Technological Progress, 1967.
[17] Global Learning Objects Brokered Exchange http://globe-info.org, 2015.
[18] IEEE Learning Technology Standards Committee - LOM working draft v4.1 Available: http://ltsc.ieee.org/doc/wg12/LOMv4.1.htm
[19] Ray Jones, Tom Boyle - Learning Object Patterns for Programming. Interdisciplinary Journal of Knowledge and Learning Objects, vol. 3, 2007.
[20] James J. L'Allier - A Frame of Reference: NETg's Map to Its Products, Their Structures and Core Beliefs. http://www.netg.com/research/whitepapers/index.asp, 1997.
[21] Learning technology standards committee website http://ieee-sa.centraldesktop.com/ltsc/ 2013.
[22] James D. Oldfield - An implementation of the generative learning object model in accounting. In proceedings of Asciilite, Melbourne, 2008.
[23] Vyutatas Stuikys, Robertas Damasevicius - Towards Knowledge-Based Generative Learning Objects,Information Technology and Control, vol. 36, no. 2, ISSN 1392-124X, 2007.
[24] Vyutatas Stuikys, Robertas Damasevicius - Development of Generative Learning Objects Using Feature Diagrams and Generative Techniques, Journal of Informatics in Education, vol. 7, no. 2, pp. 277-288, Institute of Matematics and Informatics, Vilnius, 2008.
[25] Vyutatas Stuikys, Ilona Brauklyte - Aggregating of Learning Object Units Derived from a Generative Learning Object Journal of Informatics in Education, vol. 8, no. 2, pp. 295-314, Institute of Mathematics and Informatics, Vilnius, 2009.
[26] Vyutatas Stuikys, Renata Burbaite, Robertas Damasevicius - Teaching of Computer Science Topics Using Meta-Programming-Based GLOs and LEGO Robots, Journal of Informatics in Education, vol. 12, no. 1, pp. 125-142, Institute of Mathematics and Informatics, Vilnius, 2013.
[27] David Wiley - Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy. In D. Wiley (Ed.), The instructional use of learning objects: Online version, 2000. Retrieved January 05, 2015 from http://reusability.org/read/chapters/wiley.doc.