DEVELOPING A MATH LEARNING ENVIRONMENT – A LEARNING OBJECT APPROACH

Stanica Justina Lavinia[1]

Abstract

Implementing a software architecture, that provides the learning content in a dynamic manner, would allow educational developers to use the same content more than one time, at a very structured level. The concept underlying this architecture is that of Learning Objects, a promising technology, which allows the separation of data, logic and presentation levels, offering the potential for interoperability, combination and reusability. In this context, emerged the idea to define a learning object architecture and implement it in the development of an integrated Math learning environment. Using a system for managing and combining learning objects, would simplify the process of authoring, using and reusing educational content, being able to utilize it in a variety of e-Learning contexts.

Keywords*:* learning object, learning environment, e-Learning, learning architecture

1. Introduction

Today's e-Learning technologies development is a direct consequence of the evolution of pedagogical practices in education and the spread of ICT technologies (web, multimedia, communication technologies). Their specific features bring new dimensions to education, which may be complementary or alternative to the traditional learning methods.
One major concern of the last decade in e-Learning was the idea of reusing digital resources. This approach considers that educational content created for a specific training situation could be adapted for use in other contexts. Being derived from object-oriented programming models, the elements underlying the reusability idea are the **learning objects**: blocks of small, discrete, stand-alone educational content, which can be assembled to form more complex learning objects, and can be combined to generate new learning experiences [1][4].

The learning objects technology implies that electronic courses should be designed and developed in order to be subsequently adapted, updated and reused. This can be achieved by dividing the lessons into smaller components that correspond to each presented concept. Also, if the field of study is changing rapidly, it is necessary to update them easily. To comply with these objectives, it is essential that learning objects should have a low degree of interdependence. Consequently, each object must be autonomous, in order to be reused, deleted or modified with minimal impact on the remaining learning material.
Splitting the information into smaller chunks and reusing them in different situations and combinations, will reduce the costs of authoring learning materials. Furthermore, the training resources are easily maintained, updated and developed, mainly due to their modular design and ease of integrating or reusing prefabricated components.

[1] Romanian-American University, Bucharest

2. Object-oriented instructional architecture

Trying to develop a software architecture that best meets the application requirements for Mathematics instruction, emerged the idea of combining the object-oriented programming facilities and the learning objects technology benefits. Object-oriented programming presents the promise of creating objects with a high degree of reusability. Thus, it is possible to define classes that can be used as templates, from which individual learning objects can be created, depending on teacher's or students' preferences. Object oriented programming features can be adapted to increase the reuse of teaching materials, the same way classical software systems are successfully using this technology.

This approach is somehow independent, but does not contradict the current opinion regarding the learning object implementation in educational software development. Although some research in this area may disagree with this view, considering the applicability of object-oriented programming in the development of learning objects to be limited and counterproductive [3], such an approach could bring a plus to the learning software development. Furthermore, one can say that the facilities provided by a learning management system that supports an object-oriented implementation, would lead to less predetermined lessons, customized to suit the teacher's preferences or the class's level.

Object-oriented programming concepts are based on the idea that, in the process of software development, both the problem and its solution can be structured as collections of discrete objects, each of which must *collaborate* with other objects of the collection to meet the user's requirements.
The training system's users (teachers or students) can combine the learning objects in order to reach a learning objective. These objects are created as instances of a class. Their attributes would consist of the learning content and their methods would be the operations allowed on this content, so as to meet the training objective. The teacher or student generates lessons by interacting with a *driver* that instantiates learning objects to serve the purpose of the training. During training, other objects belonging to one or more classes will be instantiated, thus the user interacting with these instances to create a lesson.

Each class *interface* determines how different objects interact. Interactions will be generated by the teacher's actions and allow greater customization of the lesson, objects being usually created dynamically in response to the user's needs. In authoring a lesson, different objects belonging to distinct classes can be instantiated; they interact with each other to provide a training experience focused on student's or class's needs. Such an approach provides the premises of object reuse in creating new lessons; the components can be used in different educational contexts, ensuring a high degree of reusability.
The advantages of object-oriented approach for creating learning objects also provide other benefits. Dynamic instantiation of objects in response to the user's demand allows greater lessons interactivity and reduces the predetermined activities of each student. Each lesson requires teacher involvement to create learning objects according to his requests and to determine how these objects interact in order to serve the teaching purposes. The entire task of authoring the training process will be interactive and will involve constant review of the lesson's objectives. Lessons may also give up their rigid structure, caused

by a determined sequence of learning objects, the teacher being able to establish alternatives for their study.

3. The set of learning objects for Mathematics

The learning objects diversity is specific to the field of study and intends to ensure the functionality of a learning environment for teaching Mathematics. Therefore, the following objects have been defined:

Expression Evaluator – is an object that allows the manipulation of Mathematical functions defined by one or more analytical expressions; this component is using a module that implements an equations editor and compiler and a mathematical expressions syntactic analyzer.

This object can be used for editing a function analytic formulae, performing a syntactic compilation and a verification of the function definition intervals. Also, for the equational description of an expression, a scripting language similar to Latex is used, being an efficient format to store and compress mathematical expressions and equations. The object allows editing the function in both formats, automatic conversion being done in both directions.

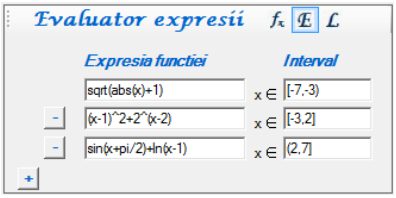

Figure 1. Expression Evaluator

Function Table – is a component used for calculating the values of a function in its domain, thus taking over the routine of some calculations. Function expressions are given through an *Expression Evaluator* object, which can mathematically interpret the analytic expressions and then calculate them over the intervals of the defined function.

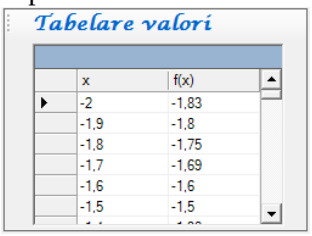

Figure 2. Function Table

Function Graph – is an object that allows the graphical representation of functions over their domains. The functions will be implemented through components of type *Expression Evaluator* or *Function Table*. The object's toolbar enables resizing and scaling the graph, to get a better visual interpretation of the graphical representation.

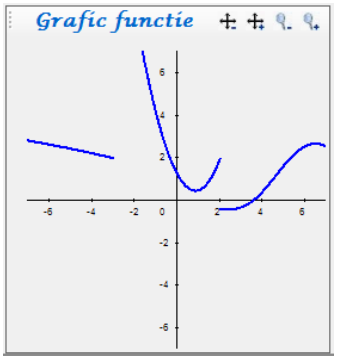

Figure 3. Function Graph

Latex Editor – is a component that can be used for editing Mathematical formulae and visual translating them in equational format. Thus, Mathematical formulae will be given in Latex textual format, which is then interpreted by the editor. Also, it can serve as a simple text editor.

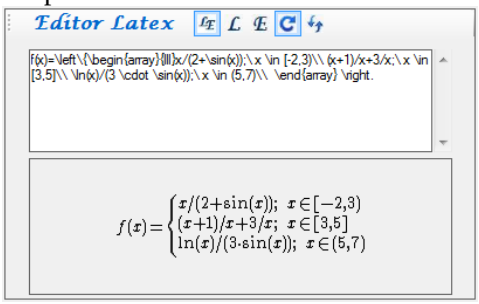

Figure 4. Latex Editor

All these objects interact directly through simple *drag-and-drop* operations, using a mediator object of type ***Function***, which implements the analytical interpretation of mathematical functions. Such an object can instantiate on demand other elementary components or can use an *Expression Evaluator* object for the syntactic validation of the function analytical form.

4. The Mathematics lessons editor

The basic functionality of the environment is to allow teachers to author their own Math lessons by combining objects at their disposal. Thus, by simple *drag-and-drop* operations, the teacher may drag the Toolbox items needed to create a lesson and then determines the interaction between them.

Figure 5. The Toolbox

This way, the application offers the possibility to configure the lesson, so that it could match the class level and the specific training requirements. Each lesson will be saved to allow its subsequent use in other training contexts. In fact, for every lesson the objects

that compose it, will be saved in the database; the objects can be used independently of the lesson they belong to, in order to be exploited in authoring another lesson.

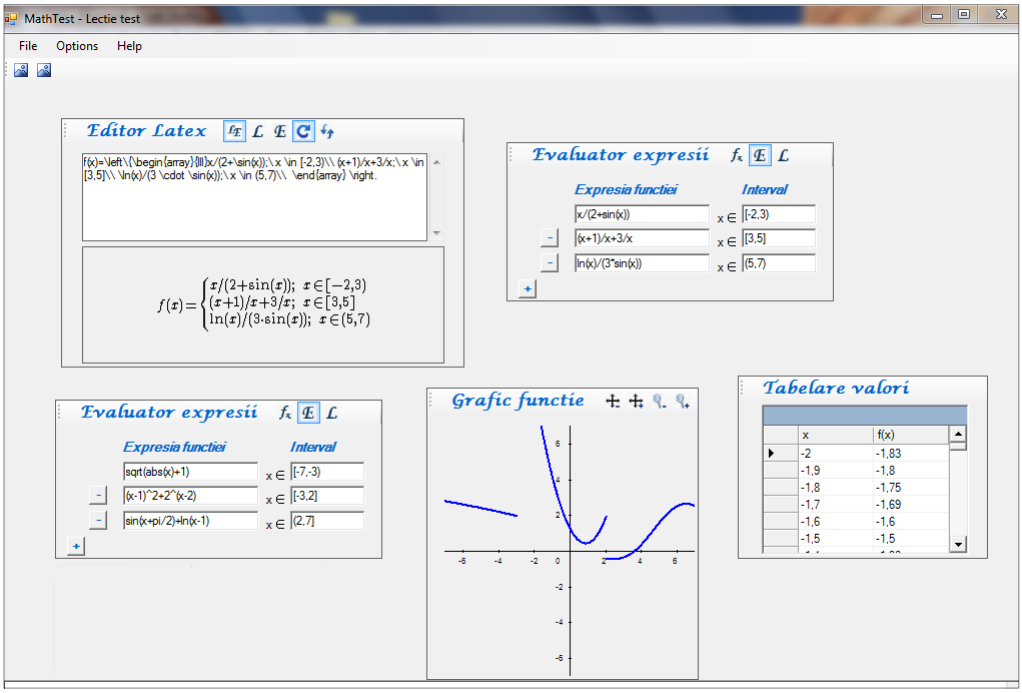

Figure 6. A Math lesson

The objects can be coupled, configured, moved, and deleted so as to better meet the lesson requirements.
Briefly, a lesson is composed of such elementary learning objects. This approach offers the possibility of developing a student-centered educational software that would allow authoring new learning objects and combining them in a dynamic manner. This type of architecture promotes the creation of new learning content and more complex lessons.

5. Conclusions

Most of the existing learning systems have all the components preprogrammed, which leads them to produce deterministic behavior, with limited and rigid feedback and explanations. In this manner, the educational software cannot take into account the specific needs of a particular user, having no possibility to adapt the learning material to the student's preferences and skills.
Consequently, the learning objects offer another perspective on education in general, and on educational software development in particular. Recently, there has been a proliferation of this concept, especially due to the fact that it offers a way of reusing the educational material.

Therefore, the present study aimed to define a computer architecture based on learning objects technology, to be implemented in the development of an integrated environment for learning Mathematics. The innovations come from two directions:

Using the learning objects technology in the software development. The advantages of this approach are high, enabling the reuse of objects in different contexts, and ensuring, through components combining, the multiple functions required by a virtual learning environment. The main argument for the use of learning objects is that they offer flexibility in creating educational materials. Additionally, the use of such an architecture, allows creating a software system with greater potential for customization.

The software allows teachers to configure the lessons. Most educational software offers predefined lessons, the teachers having no possibility to adapt to the class's needs or create new ones. But, by using this learning environment they will be able to define and combine learning objects in different ways, thus creating new lessons.
Concerning the pedagogical aspect, this will be the teachers' responsibility; they are going to choose the best ways of training, the architecture offering them the ability to customize their lessons and the possibility to combine different types of objects.

6. Bibliography

[1] Francis, David E. and Murphy, Elizabeth. *Instructional designers' conceptualisations of learning objects*. Australasian Journal of Educational Technology, vol. 24(5), pp. 475-486, 2008, ISSN 1449-5554, ISSN 1449-3098, http://www.ascilite.org.au/ajet/ajet24/francis.html

[2] Smeureanu, Ion, Dârdală, Marian and Reveiu, Adriana. *Component Based Framework for Authoring and Multimedia Training in Mathematics*. Proceedings of World Academy of Science, Engineering and Technology, vol. 29, pp. 230-234, 2008, ISSN 1307-6884

[3] Sosteric, Mike and Hesemeier, Susan. *When is a Learning Object not an Object: A first step towards a theory of learning objects*. The International Review of Research in Open and Distance Learning, vol. 3(2), pg. 16, 2002, ISSN 1492-3831, http://www.irrodl.org/index.php/irrodl/article/viewArticle/106/185

[4] Wiley, David A. *Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy*. in: *The Instructional Use of Learning Objects*. Wiley D.A. (Ed.), Agency for Instructional Technology, Association for Educational Communications and Technology, pg. 35, 2002, ISBN 0-7842-0892-1, http://reusability.org/read/chapters/wiley.doc

[5] Wiley, David A. *RIP-ping on learning objects*. in Wiley Blog: *Iterating toward openness*. 2006, online, http://opencontent.org/blog/archives/230

[6] Wiley, David A. *The Learning Objects Literature*, in Wiley Blog: *Iterating toward openness*, pp. 345-353, 2007, online, http://opencontent.org/docs/wiley-lo-review-final.pdf