LEARNING OBJECT ARCHITECTURE FOR DYNAMIC DEVELOPMENT OF MATHEMATICAL CONTENT

STĂNICĂ Justina Lavinia
CRIȘAN Daniela Alexandra[1]

ABSTRACT:

Learning objects are elements of a new architecture, which provides the educational content in a dynamic manner. The current paper wanted to highlight the advantages of implementing this technology in educational software development. This idea led to the development of an original training system for mathematics, which gives teachers an effective tool that lets them create and customize lessons by easily configuring and assembling reusable objects. The advantages of using such a system can be summarized as follows: the reduction of development efforts for creating the educational content; the ability to quickly change the training materials; the ability to create multiple versions of resources; the ease of searching and retrieving learning resources; the significant increase of reusing the learning content.

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

1. INTRODUCTION

The spread of Internet and technologies led to an exponential evolution of the assisted training. In the last decade, this has distinguished itself as another promising way to improve education, which is able to adapt to all types of learning.

Several trends have been noticed, which have increased the success of e-Learning:

- blended learning – which succeeds in combining the advantages of the traditional training, with those of e-Learning;
- web collaboration – by using the communication technologies to support training, thus facilitating the interaction between students and teachers;
- simulations and educational games – are used to acquire specific professional knowledge and skills, thus stimulating the learning process;
- *learning objects* – which are meant to increase reusability, by cutting the costs and the development time of the applications.

Learning objects are seen as the technology of the future for the design, development and delivery of the next generation of training software, because of the potential for adaptability, reusability, and scalability they provide [ALLE10].

The present paper suggests a learning object architecture that can be successfully used in the development of educational platforms. The efficiency of this approach is emphasized by the design of an integrated learning environment for the management and use of learning components for mathematics.

The main motivation comes from the need to reduce the costs and time necessary to develop educational resources, by combining reusable learning objects instead of creating new

[1] Romanian-American University, Bucharest

training materials all the time. The idea of combining existing educational components also helps customizing training in order to suit user-specific options.

The practical aspects of this research have resulted in the development of an innovative e-Learning solution for authoring and learning mathematics. The software allows the teachers to build and customize their own lessons by easily creating, configuring and combining educational components.

2. LEARNING OBJECTS APPROACH

Learning objects technology has developed as the interactive multimedia training and, afterwards, the web based training, have grown. A few decades ago, the educational sector had high expectations on how assisted instruction will revolutionize education. In reality, the cost of producing online courses with a high degree of interactivity was often excessive. In addition, educational software products were so specific that they could not be reused in a context other than the one for which they were created.

Trying to overcome these drawbacks, recent years researches have predicted how reusable, interactive, and small size learning objects will reduce the costs of assisted learning, while increasing quality and accessibility [OVER10] [ELLI08].

Learning objects are elements of a new type of training software, having its roots in object oriented programming, which is based on creating components that can be reused in different contexts. The idea behind this technology can be summarized as follows: *the authors of learning resources create small educational components that can be reused many times in various training situations.* In addition, objects are seen as digital entities delivered via the Internet, which means that a large number of users can access and use them simultaneously. The training systems designed in this manner can be easily updated by incorporating new versions of teaching resources. These are the significant differences between learning objects technology and other technologies used in the development of educational software.

The main advantages of the learning objects technology are [WILE07]:

- *interoperability* between different learning systems;
- *reusability* in order to save time and resources;
- *scalability* due to the modular development of courses and lessons, allowing the author to easily add and remove components;
- *adaptability* simplifies the process of modifying the components of a lesson;
- *cost efficiency* due to reduced development costs;
- *quick search* using the metadata that can be associated to objects or lessons, according to the modern web semantic specifications [GARA14].

The vision of learning objects is to create an educational economy, where resources are stored in commercial or open access repositories and they can be used by teachers or students for their specific training requirements [NAID06]. Another emerging idea refers to the automatic assembly of existing objects in order to create a customized training context, according to the specific needs of an individual student or even an institution.

3. DESIGNING REUSABLE LEARNING COMPONENTS

Learning objects approach radically changes the design of educational materials, since the efficiency of the learning platform depends on the design of the objects. The objects functionalities (related to autonomy, independence, format, structure, etc.) compromise the characteristics of conventional courses.

The solutions chosen for the current implementation aimed to [STAN11]:

- *Choose the best granularity of the learning objects* – Components with different degrees of granularity have been defined: elementary objects, educational objects, and compound objects (lessons). The elementary and educational objects show a reduced granularity; therefore they can be used independently and recombined in order to create different math lessons. This allows the training to be handled in a more flexible manner.
- *Define the ways educational objects interact* – The relations between the educational components are meant to add pedagogical value to the learning process. The interactions among components have been designed to illustrate certain mathematical concepts and the correlations between them.
- *Ensure the independence of educational objects* – The interactions between educational objects are managed using mediator elementary objects. Consequently, the coupling is minimal, so that the objects can be easily updated or reused. Their independence will reduce certain issues that may occur if some components are not intended to be used in a new learning context.
- *Ensure a high degree of abstractization* – Each educational object is created from a template (as an instance of a class), which implements the interface and controls the coupling with other objects.
- *Use compound objects to build mathematics lessons* – The compound objects, of larger granularity, were defined in order to provide the learning context by allowing the teacher to build mathematical lessons. The teacher has a very important role in creating the lessons by combining independent components, since the pedagogical aspect is determined by the process of assembling educational objects.
- *Reduce the dependence between the educational content and its components* – Each lesson is built by assembling independent educational components; this guarantees an easy way of updating it. For each lesson, only the references of the objects composing it are kept, while the educational objects are stored separately. This will allow the author to remove, add or change them without affecting the remaining content.

These design lines lay the foundation to develop adaptive lessons (compound objects) that offer comprehensive learning experiences. However, the teacher will determine the most efficient training format, the best way of assembling educational objects or the interaction between them. The user may have some predefined lessons, but has the additional possibility of reconfiguring them, by removing or adding new components. Such an approach provides significant benefits for both students and teachers, especially in terms of time saving, reusing and updating components.

4. INTEGRATED ENVIRONMENT FOR MANAGEMENT AND USE OF LEARNING OBJECTS

The practical aspects of the research intended to design a solution that would exploit the advantages of learning objects and to implement it in the development of an integrated environment for authoring and learning mathematics.

Consequently, emerged the idea of creating a learning platform that will:

- allow authoring math lessons and tests;
- draw graphical representation of math functions;
- present exercises, problems, and their solutions;
- take over the routine of some calculations.

Initially, the set of learning components, their properties and coupling possibilities have been defined. This phase was followed by the development of a platform for managing and using these components, while creating an online training system for mathematics.

4.1. The set of math learning objects

The set of learning objects was designed and defined in order to ensure the functionalities of a training software for mathematics. Depending on their granularity, three categories of objects have been defined [STAN12]:

1. *Elementary objects* – these are objects with a reduced granularity, that don't have visual representation and, consequently, don't have a training goal; they are used as mediator objects in the interaction between educational components. These are:

- *Function* – this object implements the mathematical concept of one-variable real functions; the functions are handled using their analytical expressions and the associated intervals.

- *TextLatex* – this object is used to implement the textual Latex format; this textual format is efficient for storing and handling the mathematical functions associated to the educational components; consequently, a two way conversion from Function to TextLatex has been defined.

2. *Educational objects* – they implement the mathematical concepts related to real-valued functions; each educational component includes other elementary objects and its behavior is determined by the coupling possibilities:

- *FctEval* – the function evaluator is an object that performs the validation, analysis and evaluation of the mathematical expressions of a function; the real-valued function that is evaluated is implemented through one or more analytical forms.
- *FctTable* and *MultiFctTable* – these are two similar controls that tabulate the values of one or more functions on their domain of definition; FctTable object operates with a single mathematical function, while MultiFctTable can handle more functions.
- *FctGraph* – it is used to draw the graphical representation of a function and to illustrate some related mathematical concepts, such as: derivative at a point, definite integral, asymptotes of a function.

- *MultiFctGraph* – this object draws a comparative graphical representation of multiple mathematical functions; the functions are taken from a MultiFctTable component or they can be added thru repeated interactions with FctEval, FctTable or FctGraph objects.
- *TexEquation* – it is used to visually represent mathematical formulas in an equational format; the mathematical formulas are introduced as text, using a scripting language, which is then interpreted and displayed by the editor. Besides the editing features, the TexEquation component can be used to display the mathematical functions of other educational objects.

All educational components operate with mathematical functions that are implemented through one or more elementary Function objects.

3. *Compound objects – Lessons* – these are complex objects created by assembling several educational components. They pursue a learning goal and emphasize the utility of the training platform, since they have been defined in order to provide the learning context, by creating rich mathematical lessons. The pedagogical aspects of training math are supported by the assembling operation. They are created using the lesson editor (the authoring component) and can be opened in a browser or eventually saved as a web document.

The environment facilitates the decomposition of math lessons into components, but also their recomposition. This allows the teacher to develop student-centered lessons by creating and combining new educational objects that fit particular training requirements.

4.2. Educational objects interactions

Each educational component is an object defined by a prototype. Its class implements an interface able to identify the other objects it can interact with in a predefined manner.

Determining how individual objects interact may present some difficulties, because the interactions should not reduce the objects independence; otherwise, the possibility of decoupling and reusing components will also be reduced. Designing the interactions between components through mediator objects provides a solution to this problem. In this way, a component that must communicate with another, will use an intermediate object that handles the communication with the destination. The interrelated components only work with mediator objects and they know how to handle them, without having to know the structure and the implementation details of other components they interact with. Briefly, the mediator facilitates the decoupling, so the educational components can be easily reused in building new lessons, since they have a high degree of cohesion and a low coupling level.

The elementary objects of type Function act as intermediary between the interdependent educational components. All components (except TexEquation) contain one or more Function objects, which have the role to mediate the interaction with other types of controls. In addition, the conversion from Function to latex format enables the operations of copying components to the clipboard or storing them into the database.

The interface of each learning component aims to standardize the interactions with objects of different types and forces them to obey certain communication rules. Therefore, in order to allow simple drag-and-drop interactions, all components are implemented through user controls (UserControl). In addition, a data transfer mechanism which is format independent is guaranteed via the IDataObject class.

The operation of dragging an object over another consists in interpreting the data of the source component, followed by copying the function object from the source control into the destination. In case of an interaction with objects of type TexEquation (which don't contain a function), the coupling involves converting the source function object to Latex format, that can be displayed by the destination component.

```
private void FctGraphCtrl_DragDrop(object sender, System.Windows.Forms.DragEventArgs e)
{
    // implementing the drag-drop operation over a FctGraph
    FctGraphCtrlUC.FctGraphCtrl CC = (FctGraphCtrlUC.FctGraphCtrl)sender;
    if (e.Data == null)
        mesaj("null data pointer");
    else
        if (e.Data.GetDataPresent("FctEvalUC.FctEval"))
        {
            // if a FctEval object have been dragged
            FctEvalUC.FctEval EE = null;
            EE = (FctEvalUC.FctEval)e.Data.GetData("FctEvalUC.FctEval");
            CC.fct = ClassLibrary.Function.CopyFunction(EE.fct);
            CC.DrawGraph();
            modify = true;
        }
    // .....................
}
```

Using these mediator objects, components of different types interact directly with one another in a predefined manner. The interaction patterns are designed in order to ensure the correlation between the mathematical concepts the educational objects implement. The large number of coupling possibilities is a consequence of the large variety of learning components.

Table 1. Educational objects interactions

Source component	**Destination component**	**Mediator object**	**Required conversion**	**Description**
FctEval	FctTable	Function		calculates the function table
	MultiFctTable	Function [ ]		adds the function to the current comparative table
	FctGraph	Function		draws the function graph
	MultiFctGraph	Function [ ]		adds the function to the current comparative graph
	FctEval	Function		calculates the function derivative and generates a new function
	TexEquation	TextLatex	Function → TextLatex	displays the function in visual equational format
FctTable	MultiFctTable	Function [ ]		adds the function to the current comparative table

	FctGraph	Function		draws the function graph
	MultiFctGraph	Function [ ]		adds the function to the current comparative graph
	TexEquation	TextLatex	Function → TextLatex	displays the function in visual equational format
FctGraph	MultiFctGraph	Function [ ]		adds the function to the current comparative graph
	TexEquation	TextLatex	Function → TextLatex	displays the function in visual equational format
MultiFctTable	MultiFctGraph	Function [ ]		draws the graphs of all tabulated functions
	TexEquation	TextLatex	Function [ ] → TextLatex	displays all tabulated functions in visual equational format
MultiFctGraph	TexEquation	TextLatex	Function [ ] → TextLatex	displays all plotted functions in visual equational format

All these interactions are implemented through simple drag-and-drop operations. This eases the process of combining educational components and allows the teachers to effortlessly build and configure new math lessons. The following figure provides an overview of the interaction possibilities.

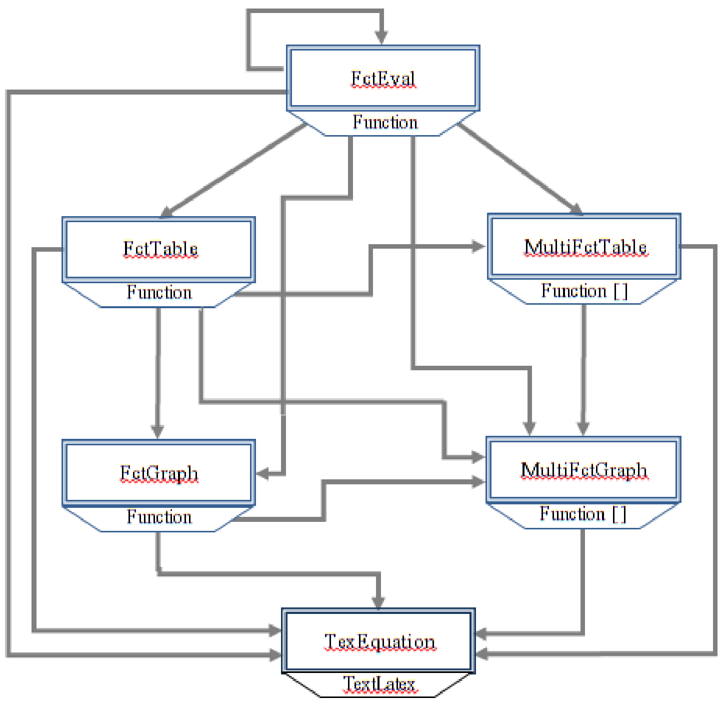

Figure 1. Educational objects interactions

5. THE FEATURES REQUIRED FOR TEACHING MATHEMATICS

Creating a reliable system for the study of mathematics raises many problems, since it has to implement the main functionalities, which are required to illustrate the concepts specific to this field of study, such as displaying formulas or rendering functions graphs. Consequently, the following modules have been defined as follows:

The *Main* module:

- includes a graphical interface that allows the user to access the software functionalities and to switch between the two operating modes (author / reader);
- implements the features necessary for building math lessons;
- allows the user to generate assessment tests, according to different criteria;
- provides functionalities for training, exercising, and assessment;
- manages the thematic search, and search depending on the training level;
- manages users and monitors students' training evolution.

The *Math* module:

- implements a mathematical expressions syntactic analyzer;
- contains an equation editor and compiler;
- performs compilations and evaluations of real-valued functions;
- calculates the values of a function on its definition intervals;
- acts as an assistant in calculating the derivative of a real-valued function.

The *Graphics* module:

- renders graphical representations of real-valued functions in a two-dimensional coordinate system;
- draws the graphical elements associated to a function: derivative at a point, definite integral, asymptotes.

The *Equation* module:

- uses a scripting language, similar to Latex, to equationally describe formulas; this is an efficient format for compressing and storing equations and mathematical expressions;
- allows editing and visual displaying equations, translating them into the scripting format.

5.1. The functionalities of the integrated environment

The application is a complete training platform that involves a Windows interface for managing learning objects and using them for authoring math lessons, and a web interface that implements the features required by an efficient teaching / learning process.

The integrated environment is developed in C#. NET and includes two different software components:

- *MathLessons* – the authoring component – developed as a Windows Application project; it implements the mathematical lessons editor, having a rich interface that includes multiple options, demonstrating user-friendliness.

- *MathPractice* – the training component – is a Web Application project; it includes the necessary features for the online training (using the authored lessons and problems) and assessment (using the tests generator).
- The platform aims to provide the required functionalities for teaching mathematics, namely: creating lessons; managing learning resources; training and exercise; assessing and self-assessing; monitoring students' evolution.
- The access to the functionalities provided by the platform is controlled based on the following three roles:
- *Administrator* – is responsible with the database management; he also manages the users and sets their access rights.
- *Teacher* – builds mathematics lessons, creates assessment tests, authors problems and explanations, organizes the chapters according to the curricula, and sets the difficulty levels of tests and exercises. Additionally, the teacher can easily track each student's individual path and can get reports and graphical representations of the students' evolution.
- *Student* – has online access to the training materials, consisting of lessons, exercises, and self-assessment tests that match his training level. Furthermore, the student can view the information about his evolution and his assessment marks.

The platform also provides access to the online training component for unauthenticated users, but with limited rights.

6. CONCLUSION

Learning objects technology delivers a new architecture that can be successfully applied in educational software development. This architecture has its roots in object oriented programming which gives the utmost importance to components that can be reused in different contexts.

This approach based on reusing educational components has many benefits in creating learning resources and also in developing training platforms. However, building a lesson by combining low granularity independent objects is problematic, given that the author should ensure the pedagogical quality of training. In the current context, the design of the learning components aimed to implement some important mathematical concepts, while the objects interactions reflected the relationships between them.

The platform is very useful, providing a superior training experience that offers students a better understanding of the mathematical concepts by combining different textual or graphical components. The software also facilitates the creation of new lessons by integrating some prefabricated objects or reusing the existing materials.

The use of learning objects in the practical development of educational resources has many advantages and disadvantages relevant to the future use of this technology in creating educational software. The main motivation for their use derives from the considerable efficiency they can offer in creating e-Learning courses.

BIBLIOGRAPHY

[ALLE10] Allen C.A., Mugisa E.K., *Improving Learning Object Reuse Through OOD: A Theory of Learning Objects*, Journal of Object Technology, vol. 9, no. 6, pp. 51–75, ISSN 1660-1769, 2010, [http://www.jot.fm/issues/issue_2010_11/article3.pdf]

[ELLI08] Elliott K., Sweeney K., *Quantifying the Reuse of Learning Objects*, Australasian Journal of Educational Technology, vol. 24(2), pp. 137-142, ISSN: 1449-5554, ISSN: 1449-3098, 2008, [http://www.ascilite.org.au/ajet/ajet24/elliott.html]

[GARA14] Garais G.E., *SEO Codding Guidelines for a Reliable Attraction of Visitors to Relevant WEB Content*, Journal of Information Systems and Operations Management (JISOM), Vol. 8 No. 1 – May 2014, ISSN 1843-4711, 2014, [http://jisom.rau.ro/ISOM_Nr14.html]

[NAID06] Naidu S., *eLearning - a Guidebook of Principles, Procedures and Practices*, Second Revised Edition, for Commonwealth of Learning, Commonwealth Educational Media Center for Asia, pg. 100, ISBN: 81-88770-04-3, 2006, [http://www.cemca.org/e-learning_guidebook.pdf]

[OVER10] Overton L., Howarth N., Merritt R., Basiel A., Howarth M., *Delivering Results with Learning Technologies in the Workplace*, Towards Maturity Enterprises Ltd, on behalf of BECTA, pg. 76, online, 2010, [http://www.e-learningcentre.co.uk/Resource/CMS/Assets/5c10130e-6a9f-102c-a0be-003005bbceb4/form_uploads/delivering_results_with_learning_technologies.pdf]

[SMEU08] Smeureanu I., Dârdală M., Reveiu A. *Component Based Framework for Authoring and Multimedia Training in Mathematics*. Proceedings of World Academy of Science, Engineering and Technology, vol. 29, pp. 230-234, ISSN 1307-6884, 2008

[STAN11] Stănică J.L., Crișan D.A., *Framework for Flexible Reuse and Assembly of Learning Objects – A Pilot Project*, Journal of Information Systems and Operations Management (JISOM), Vol. 5 No. 2.1 – Special Issue / December 2011, pp. 478-484, ISSN 1843-4711, 2011

[STAN12] Stănică J.L., Crișan D.A., *Dynamic Development and Assembly of Learning Objects in a Math Learning Environment*, Journal of Information Systems and Operations Management (JISOM), Vol. 6 No. 1 / May 2012, pp., ISSN 1843-4711, 2012

[WILE07] Wiley D.A., *The Learning Objects Literature*, in Wiley Blog: "Iterating toward openness", pg. 10, online, 2007, [http://opencontent.org/docs/wiley-lo-review-final.pdf]