FRAMEWORK FOR FLEXIBLE REUSE AND ASSEMBLY OF LEARNING OBJECTS – A PILOT PROJECT

Justina Lavinia Stănică[1]
Daniela Alexandra Crișan[2]

Abstract

The present paper aims to highlight certain relevant and new features revealed by the research on assisted training. The motivation is given by the necessity to efficiently develop the educational resources by combining reusable learning objects, and also by the need to customize training in order to meet specific user options. The main objective is to design a flexible framework for assembling and reusing learning objects, which can be successfully applied in developing an educational platform for mathematics teaching.

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

1. Introduction

Given the ongoing development of the e-Learning software, for both the educational and the commercial sectors, there are some concerns regarding the cost, quality, customization, and reuse of the training resources.

The learning objects approach radically changes the development of the educational materials, especially with regard to the author's style and the pedagogical and narrative style of the course. The objects functionalities (related to their autonomy, independence, format, structure, etc.) radically change the characteristics of the conventional courses. However, this approach offers significant benefits for students, teachers and educational developers, especially in terms of reusing and updating the objects, and reducing the time of creating the resources.

The advantages of the learning objects technology are clearly visible, their implementation having remarkable results for the economic efficiency, lowering the costs, reducing the risks, improving the quality of learning, and increasing the profits.

The learning objects field is rather new, the term *Learning Object* being firstly proposed by Wayne Hodgins in 1994 [POLS03]. The concept has its origins in object-oriented programming and the influence of this theory is visible in the adoption of some specific models.

The fundamental idea of this technology is based on the fact that, instead of creating new educational resources every time, teachers should use those existing in online repositories. This concept aims to facilitate the content interchange between teachers, courses or institutions.

[1] Lecturer, PhD Candidate, School of Computer Science for Business Management, Romanian-American University
[2] Associate Professor, PhD, School of Computer Science for Business Management, Romanian-American University

The learning objects are elements of a new type of training software, which is based on the creation of components that can be reused in different contexts. In addition, objects are seen as digital entities delivered over the Internet, which means that a large number of users can simultaneously access and use them. Furthermore, the training systems implementing this technology can be easily updated, by incorporating new versions of the teaching resources. These are the significant differences between the learning objects technology and other technologies used in educational software development.

2. Learning objects modeling

The objectives of this research consisted in designing an original software solution that would exploit the benefits of the learning objects, and implementing it in the development of an innovative system for teaching mathematics. Such an architecture, based on reusable components, represents a viable option, since it will provide all the functionalities required by a training application.

The success of a learning object approach depends on the design of the objects. The design requires, on one hand, an overall view to conceive the object as a component of a complex training situations (as part of a whole), and, on the other hand, a detailed vision to create the object as an independent, self-contained learning unit, which can be reused [POLS03].

The solutions chosen in the current implementation are as follows:

Establishing the optimal granularity

With the emergence of different types of learning objects, there was another concern about how the training requirements of a particular situation will affect the size or granularity of the components. The educational authors constantly comply with some certain specifications regarding the objects' size and content. Although there is no optimal granularity, the unanimous opinion is that a reduced size will increase the potential for reuse [HODG02] and will facilitate an adaptive training.

One of the solutions adopted here consisted in using components with different degrees of granularity. Thus, four categories of objects were defined: simple, elementary, web objects and lessons. The first three types have a low granularity and they can be used independently in different lessons. Their small size enables a flexible student-centered planning of training. Considering that the learning environment provides several alternatives for assembling the objects, the teacher can choose the appropriate manner of study.

An electronic lesson is designed as an ordered collection of interrelated components; this requires setting a hierarchy of objects, based on a set of well-defined educational objectives, characteristic to the field of study. This idea led to the inclusion of simple objects as a special category of components; even though they have a low granularity and do not pursue a training objective, they are used to handle some field-specific concepts and to manage the subsequent interactions of elementary components.

Providing a high level of abstraction

The learning objects approach provides a new perspective on the creation of educational materials. The success of this technology consists in separating the content of the learning resources from their use. Although pedagogical principles should underpin the objects design, they should not be linked to a specific training methodology. The optimum reusability and functionality of the learning components can be achieved only if they have a high level of abstraction [POLS03].

Starting from this idea, each elementary object is created from a template that implements its interface and controls its interactions with the other objects. The components interface is designed to standardize the interactions between different objects and forces them to comply with certain communication rules.

```
public partial class EvalExpresii : System.Windows.Forms.UserControl, IDataObject
{
        // the Express-Evaluator class
        public EvalExpresii crt;
        private string expression, latExpression;
        public LibrarieClase.Functie fct = new LibrarieClase.Functie();
        public EvalExpresii()
         {
        InitializeComponent();
         }
        public bool GetDataPresent(string fmt)
        {
        if (fmt == "Math" || fmt == "Bitmap" || fmt == "EvalExprUC.EvalExpresii"
                    || fmt == "Latex" || fmt == DataFormats.Text)
                return true;
        else return false;
        }
        // ....................
}
```

Basically, every elementary object is created as an instance of a class, depending on the author's options.

```
private void platForma_DragDrop(object sender, System.Windows.Forms.DragEventArgs e)
{
  // ....................
        if (e.Data.GetDataPresent("MathTest.paleta.itemDrag") &&
             e.Data.GetData(DataFormats.Text) == "pictBoxEvalExpUC")
         {
            // creates a new Express-Evaluator
            EvalExprUC.EvalExpresii EEucNou = new EvalExprUC.EvalExpresii();
            EEucNou.Location = new Point(e.X, e.Y - PanelPlatforma.Top - 20);
            k = 1;
            foreach (Control c in PanelPlatforma.Controls)
                if (c is EvalExprUC.EvalExpresii)
                {
                    string nume = c.Name;
                    nume = nume.Remove(0, 12);
                    k = Convert.ToInt32(nume);
                    k++;
                }
                EEucNou.Name = "EvalExpresii" + k; // the name for the new Express-Evaluator
                EEucNou.AllowDrop = true;
                EEucNou.MouseDown += new MouseEventHandler(evalExpresii_MouseDown);
```

```
               EEucNou.DragEnter                    +=                    new
DragEventHandler(evalExpresii_DragEnter);
               EEucNou.DragDrop                     +=                    new
DragEventHandler(evalExpresii_DragDrop);
               PanelPlatforma.Controls.Add(EEucNou); // adds the object to
the panel
       }
  // .....................
}
```

Abstraction ensures the objects independence of their use, enabling them to be coupled with other components in order to be used in different learning situations.

Defining the coupling possibilities between objects and ***ensuring their independence***

Determining how the individual objects interact can present certain challenges, because if the interaction patterns are wrong, the possibility of reusing and decoupling components is also reduced. Designing the components interactions through a mediator object provides a solution to this problem. Thus, a component that must communicate with another one uses an intermediate object, which then handles the communication with the destination object. Consequently, the interrelated components work only with the mediator object, without having to know the structure and the implementation details of the other components they interact with. Briefly, the mediator facilitates the decoupling. In this manner, the learning objects can be easily reused in creating new lessons, since they have a high cohesion and a low coupling level. Their independence will eliminate certain problems that may occur if there are elements that are not intended to be used in a new learning context.

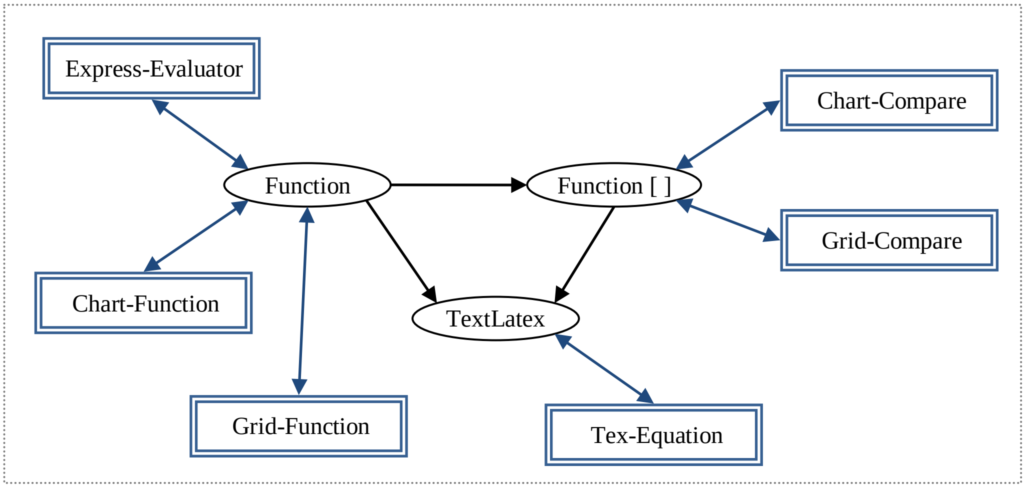

Figure 1. Learning objects interactions

The relations between the elementary components are meant to add pedagogical value to the learning process. The interactions among components were designed to illustrate certain mathematical concepts and the correlations between them.

Using compound objects and ***minimizing the dependence of the educational content on its objects***

Still, using abstract, self-contained, independent learning objects cannot generate an appropriate learning framework. Starting from this idea, one of the solutions was to create compound objects. The main advantages are:

- they bring more pedagogical value, which is not possible through simple objects;
- they support the reusability idea, because each elementary object can be reused independently.

In our implementation, the lesson type objects, of larger size, are compound objects that were introduced with the purpose of providing the learning context. Combining independent objects in order to create a lesson, is the most important step, because the pedagogical aspect is determined by the assembly process.

Creating a lesson from independent objects provides an easy way of updating it. Basically, for each lesson, only the references of the objects are kept, they being stored separately. This aspect allows the removing, the adding or the changing of the content without affecting the remaining one.

Compound objects offer multiple options for the presentation of the learning content. Thus, the teacher may have a compound learning object (a lesson), but he can also reconfigure it by removing or adding new components.

All in all, this approach becomes a cornerstone for the development of adaptive compound learning objects that can offer complete training experience. Eventually the teacher will choose the best form of training, the manner of combining objects or the interactions between them.

3. The set of learning objects for mathematics

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

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

1. *Simple objects* – these are basic objects with low granularity, that don't have a training purpose, being used to manage the interaction possibilities of the elementary objects.

- *Function* – an object specific to mathematics, which allows the handling of one-variable functions, using their analytical expressions.
- *TextLatex* – this component is defined in order to ensure the conversion of other objects in textual Latex format, which can then be stored, equationally displayed or interpreted by other components.

2. *Elementary objects* – they implement various mathematical concepts; they include one or more simple objects and their behavior is determined by the coupling possibilities:

- *Express-Evaluator* – it is an object that carries out the validation, analysis and evaluation of complex mathematical expressions, being used for handling real-valued functions that are implemented through one or more analytical forms.
- *Grid-Function* and *Grid-Compare* – these are two similar controls that calculate the values of one or more functions in their domain of definition. The difference between them is the fact that the Grid-Function object operates with a single mathematical function, while Grid-Compare can handle more functions.
- *Chart-Function* – it is used for the graphical representation of a function and the illustration of some related mathematical concepts, such as: derivative at a point, definite integral, asymptotes of a function.
- *Chart-Compare* – it draws a comparative graphical representation of several mathematical functions. The functions are taken from a Grid-Compare component or they can be added as a result of the repeated interactions with Express-Evaluator, Grid-Function or Chart-Function object types.
- *Tex-Equation* – it is used to edit mathematical formulas and to visually translate them in an equational format. Thus, the mathematical formulas are given in a textual format, using a scripting language, which is then interpreted and displayed by the editor. In addition to the editing features, this type of object can be used to view the mathematical functions associated with other components.

The mathematical functions on which the elementary components operate are implemented through one or more simple objects of type Function.

3. *Web objects* – these are objects that partially pursue an assessment or learning objective, being used by the online training system:

- *MC-Question* – multiple choice question – it is used to generate assessment and self-assessment tests; is consists of a question, a correct answer and several alternative answers;
- *Problem* – it is designed to meet the need for practice and reinforcing concepts; each exercise can have a solution divided into steps.

4. *Lesson objects* – these are complex objects created through the assembling of several elementary components or web objects. They may be viewed in a browser and eventually saved as a web document.

The software allows both the decomposition of the objects into elementary components and their recomposition. This approach offers the possibility to develop a student-centered educational software that would allow authoring new learning objects and combining them in a dynamic manner.

4. Conclusion

A multitude of e-Learning solutions on a larger or a smaller scale have been continuously developed through the years. Most of the existing learning systems have all the components preprogrammed, making them to produce a deterministic behavior, with limited and rigid feedback and explanations.

In this context, generally speaking, learning object technology offers another perspective on education, particularly on the educational software development, because of its potential of reusing and customizing the educational resources, but also for the business environment as an add-on service for virtual business incubators [JOIT10].

Starting with this idea, the present research aimed to emphasize the advantages of implementing the reusable learning object technology in the educational software development. For this, an originally training system for mathematics have been developed, which provides an effective tool that allows teachers to author and to customize lessons by easily configuring and combining the learning objects.

Bibliography

[FRAN08] Francis D.E., Murphy E. *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]

[HODG02] Hodgins W., *The Future of Learning Objects*, ECI Conference on e-Technologies in Engineering Education: Learning Outcomes Providing Future Possibilities, Aug 2002, Davos, Switzerland, Lohmann J.R. & Corradini M.L. Eds, vol. P1, article 11, pp. 75-82, 2002, [http://services.bepress.com/cgi/viewcontent.cgi?article=1012&context=eci/etechnologies]

[JOIT10] Joița A.C., Cărutașu G. and Botezatu C.P., *Technology And Business Incubator Centers - Adding Support To Small And Medium Enterprises In The Information Society*, 13-14 May 2010, 17th International Economic Conference The Economic World Destiny: Crisis and Globalization?, pp. 107-115. ISBN 978-973-739-987-8

[POLS03] Polsani P.R., *Use and Abuse of Reusable Learning Objects*, Journal of Digital Information, vol. 3(4), article 164, pg. 10, ISSN: 1368-7506, 2003, [http://journals.tdl.org/jodi/article/view/89/88]

[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, 2008, ISSN 1307-6884