A Model of a New e-Learning Shell Authoring Tool

Tsvetan Hristov

Abstract: *The paper discusses the development of a new authoring tool for the needs of the new SCORM conformable e-Learning Shell (eLSe) Software Platform. The old version of the e-Learning platform and its variations is installed in 14 Bulgarian Universities and also in educational organisations in Germany, Hungary and Russia.*

Key words: *e-Learning, e-Learning Shell, Authoring, LMS.*

INTRODUCTION

According to the recommendations of the General Direction for Education and Culture of the European Commission, one of the basic criteria for evaluating the level of use of information and communication technologies in the university training process will be the number of published WEB based courses. An effective solution to comply with the requirement that every lecturer should make his or her teaching materials accessible via Internet is the use of electronic learning platforms, which are integrated environments for delivery, implementation, modification and management of WEB based teaching materials and communication between lecturers and students.

In response to this initiative a team from the University of Rousse, Bulgaria developed an e-Learning software platform named that enables lecturers to publish their own WEB-based course without having any skills and knowledge in HTML, JavaScript and other WEB-programming languages or tools. So far the platform is installed in 11 Bulgarian universities and 2 foreign institutions [3]. The user friendly interface and simple mechanisms for uploading learning materials allows building a variety of courses in various knowledge domains. Currently there are 80 courses, uploaded on the university server with 3800 registered users [4].

Currently each existing course in our e-Learning Shell (eLSe) Software Platform is described with a set of objects. They cannot be use in other courses without predefinition. To a certain degree this structure is not well optimized since the number of objects is significant, but they are not reusable.

The main goal of our work is to develop a new version of the e-Learning Shell software platform in order to avoid the disadvantages of the old one and to make it SCORM conformable. According to the project timetable at this stage of development a new authoring tool have to be developed. This tool will allow lecturers to:

- publish, update and delete reusable resources;
- import and export SCORM packages;
- create the course structure;
- assign resources to developed course structure.

LAYOUT

The learning resources can be divided into eleven major types, depending on their content:

- Text – contains plain or HTML formatted text with images.
- File – document file formats like DOC, PDF, PS, PPT etc.
- Multimedia – multimedia files like AVI, MOV, QT, MPG etc.
- Test – contains test properties and test questions. The test properties are test name and number of questions. Tree types of questions can be published: single choice, multiple choice and fill in the blank.

- Message board – this resource contains simple plain text or HTML formatted messages with images.
- FAQs – contains questions and answers. The answers can be plain text or HTML formatted with images.
- Timetable - plain text or HTML formatted messages with timestamp.
- Forum – contains a standard forum with tree structure.
- Workshop – using this resource type, lecturers can publish projects using plain or HTML formatted text and files and assign student accounts.
- Chat – this resource type will be used as to the Flash-based chat client.
- SCO – using this resource type, users will be able to publish SCORM packages.

The resources can also be divided into two groups, depending on their complexity:

- Simple resources

The management of this resource group is quite simpler. They require operations only for creation, editing and deleting. These resource types are: Text, File, Multimedia, Forum, Chat and SCO.

- Complex resources

The complex resources are not atomic resources. They contain small chunks. For example tests are created from test questions. But these chunks cannot be presented as simple resource, because they cannot exist outside the complex resources.

Lecturers have to take the following steps to create a WEB-based course:

Step 1: Create a new course object

The authoring tool has to prompt lecturers to enter the course name in all supported system languages, to choose the course template and course icon. If the information is correct the course will be registered into the database. Lecturers have to be also able to edit the course information and delete entire course.

Step2: Publish resources

When the course is created lecturers have to publish their learning resources. The process of publishing simple resources requires tools for creating, editing and deleting resources. Such tools have to be developed for each resource type, while they contain different types of information (files, text, images, media etc.).

Publishing complex resources requires more functionality. After creating the resource item lecturers have to upload additional information. For example publishing a test requires publishing the test properties (test name, number of test questions, etc.) and after that publishing the test questions. This means that the functionality, required for publishing a simple resource have to be extended with specific functions depending on the complex resource type [5].

Step 3: Creating course structure

After publishing the resources lecturers have to create the course structure. According to the requirements this is a tree structure. The authoring tool has to enable lecturers to create, edit and delete course items and sub items.

Step 4: Assign resources to the course structure

The last step is assigning the published resources to the course structure. According to the SCORM specification resources can be assigned only to the leaf elements of the tree structure [6]. Also more than one resource can be assigned to a particular leaf. The system has to enable lecturers to assign new resource, change resources' order and remove assignment.

Taking into consideration the requirements, mentioned in the four steps above, the use case model on fig. 1 of the authoring tool can be developed.

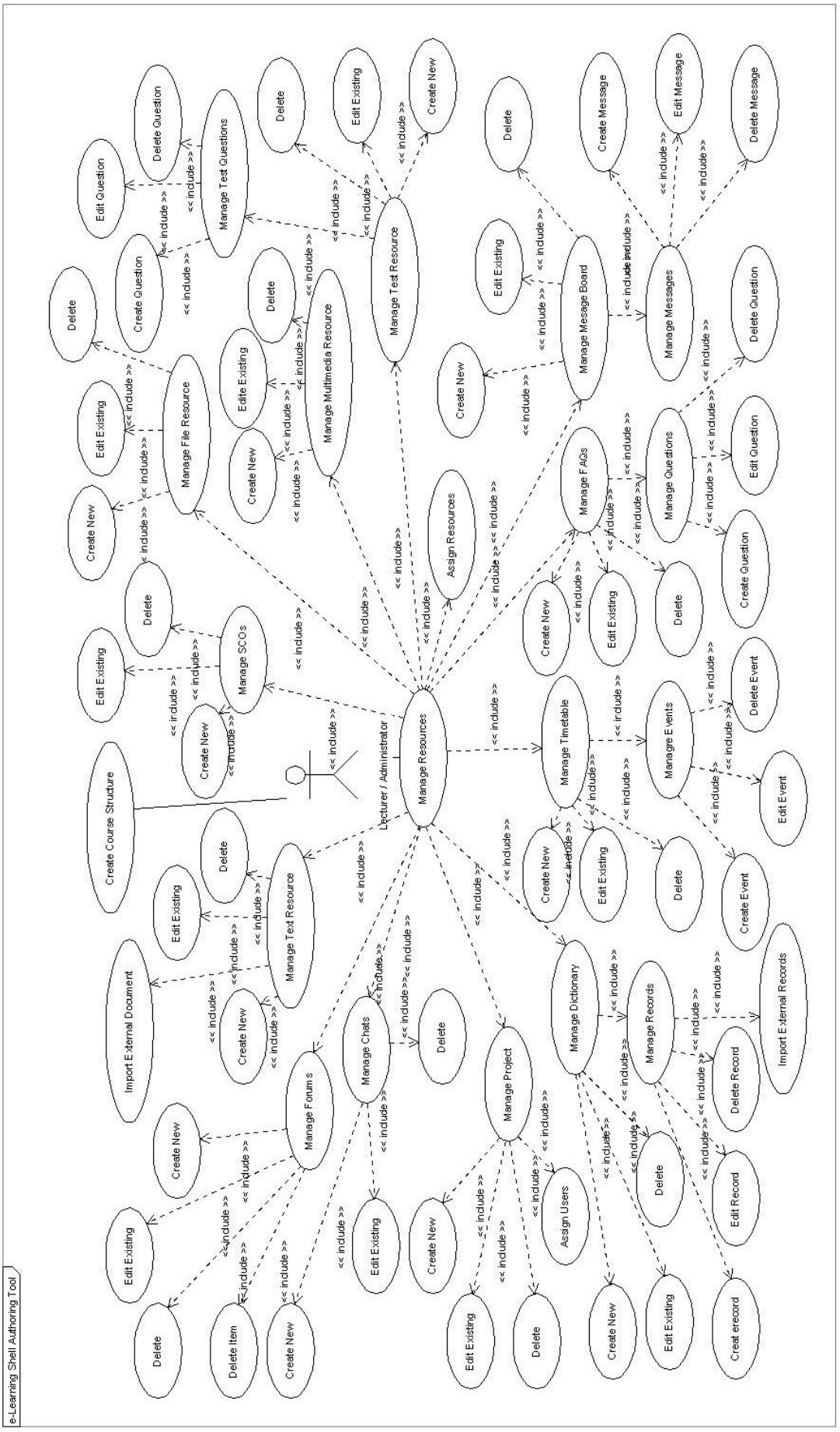

Fig. 1. Use case model

In order to make the elements of the authoring tool reusable, easy to update, to improve the system quality and productivity, the object-oriented approach for design and development will be used [1,2].

On fig. 2 the class diagram of the authoring tool is presented. Each class represents a part of the functionality described in the use case diagram on fig. 1. For example, the test class has methods for visualizing all the forms needed for creating, editing, deleting the entire test, creating, editing, deleting the test questions, and the data manipulation methods.

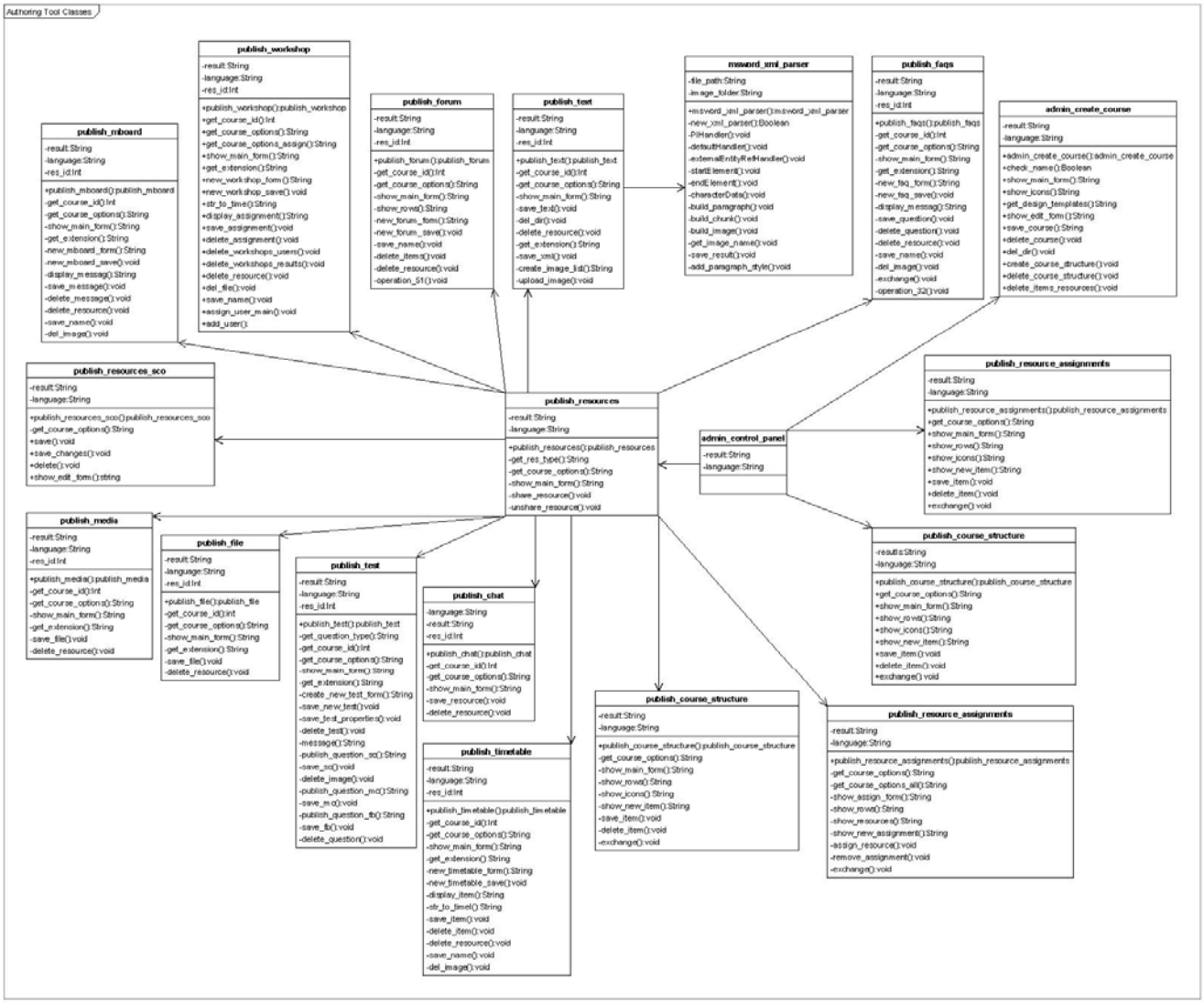

Fig. 2. Class Diagram

On fig. 3. the process of creating test resource is visualized. When a lecturer starts the authoring tool a new instance of admin_publish_resources class is created (1). If he chooses the option for creating a new test, a new instance of admin_publish_test is created (2) and the system prompts the user to enter the test name, the number of the test questions and the other test properties (3). When the form is submitted and the data is correct, the authoring tool saves the data into the database (5) and destroys the class instance. In order to publish questions the user has to open the test in edit mode. Again a new instance of admin_publish_test is created (8) and a test question management form is visualized. The authoring tool prompts the lecturer to select action for adding, editing or deleting test question. When adding new question is selected (9) the system visualizes the appropriate form depending on the selected question type. When the form is submitted

and the data is correct, the tool saves the question data into the database (11) and destroys the instance of admin_publish_test class.

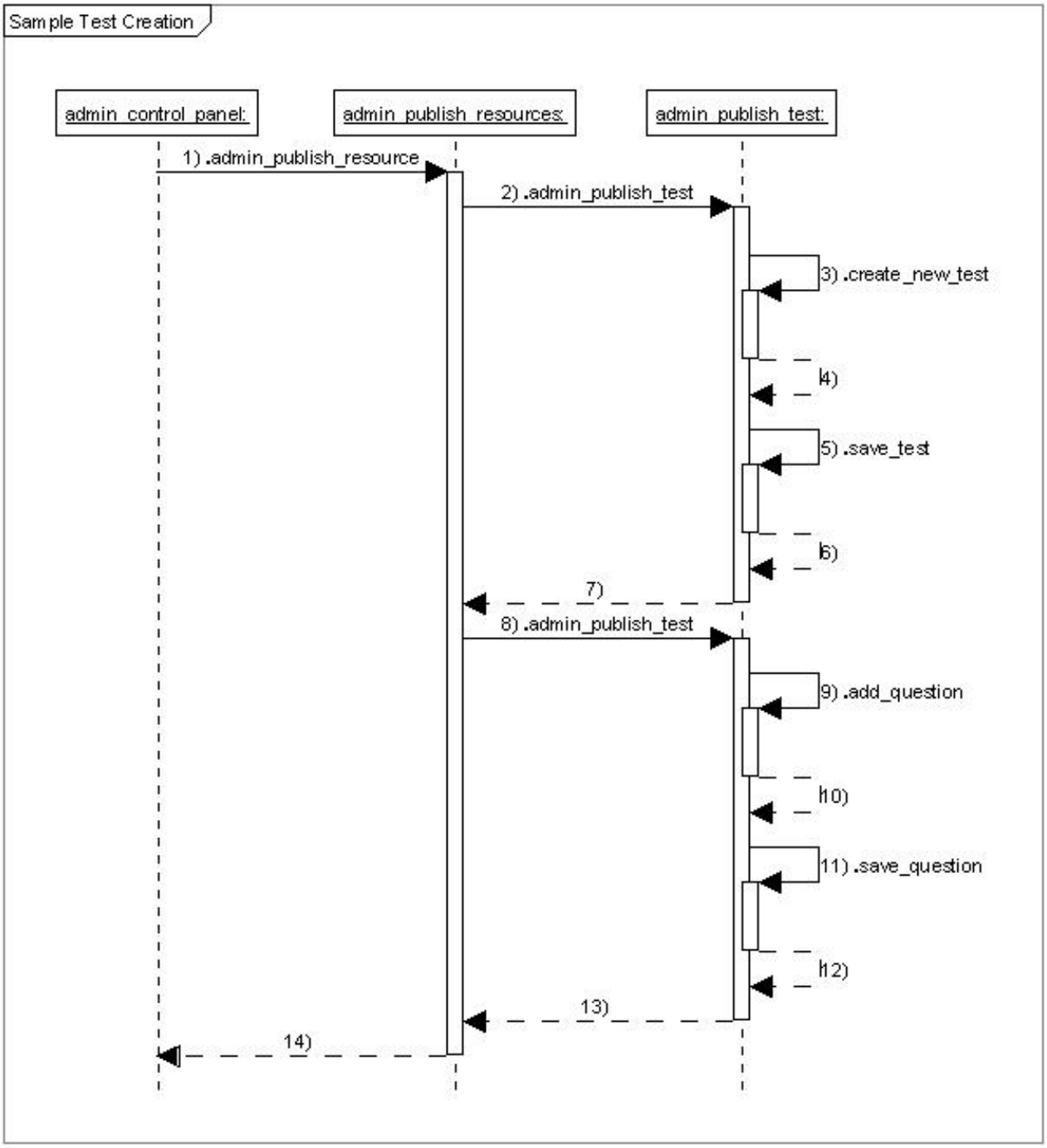

Fig. 3. Creating a test resource

CONCLUSIONS AND FUTURE WORK

The usage of the old eLSe authoring tool and the statistical information gathered up till now has proved its reliability, flexibility and usefulness [4], hence it is reasonable for the new tool to be developed based on the experience from the old system. In addition the new tool is expected to introduce a higher level of usability and reliability to comply with the contemporary requirements.

REFERENCES

[1] Domenico, F. et al., Analysis of Commercial and Experimental e-Learning Systems, Proceedings of QUIS – Quality, Interoperability and Standards in e-Learning, 2004

[2] Eriksson, H., M. Penker, Business Modeling with UML: Business Patterns at Work, John Wiley & Sons, 2000

[3] Hristov, T., S. Smrikarova, A. Vasileva, A. Smrikarov, An Approach to Development of an e-Learning Software Platform, Proceedings of CompSysTech'02, Sofia, 2002

[4] Hristov, T., T. Georgiev, M. Klaus, R. Fischer, The Statistical Features of eLSe Learning Environment, Proceedings of CompSysTech'05, Varna, 2005

[5] Rosenberg , D., K. Scott , Applying Use Case Driven Object Modeling with UML: An Annotated e-Commerce Example, Addison Wesley, 2001

[6] Rustici, M., SCORM 2004 Reference Guide, CONFORM 2 SCORM, 2006

ABOUT THE AUTHOR

Tsvetan Hristov, PhD Student, Department of Computing, University of Rousse, Phone +359 82 888 276, e-mail: thristov@ecs.ru.acad.bg