ACTORS AND STRUCTURE OF MODERN E-LEARNING SYSTEMS

Ionuț Reşceanu

University of Craiova, Faculty of Automation, Computers and Electronics, Department of Mechatronics, Romania

Abstract: E-learning is the continuous assimilation of knowledge and skills by adults stimulated by synchronous and asynchronous learning events — and sometimes Knowledge Management outputs — which are authored, delivered, engaged with, supported, and administered using Internet technologies. The first e-learning systems were actualy web sites which contained some electronic materials and primitive tests. Also, there were few categories of actors within those systems. Modern age e-learning systems are based on the notion of lesson. The people involved are grouped within several categories. The present paper tries to present those categories and also to highlight the structure of the main entity of modern e-learning systems – the lesson.

Keywords: learning systems, computer applications, computer systems, computer-aided instruction

1. INTRODUCTION

Training materials and instructional information that is delivered electronically over the Web, through an organization's intranet, or via CD-ROM, is known as e-Learning. An online course is a popular type of e-Learning which can include test questions. Using multimedia and interactive technology to illuminate the topic, e-Learning can engage the learner with video and audio features, self-paced navigation, and options such as pop-up hints and hyperlinks to related topics.

In a world where access to information is the key to staying competitive, organizations rely on people who are continually learning new skills, crafting innovative solutions to changing circumstances, and staying informed and responsive. With e-Learning, you can provide affordable training to as many employees or business partners as you need, at any time, from any location. You can create an instant, state-of-the-art learning system in a fraction of the time needed to implement a more traditional solution. You also have access to solid reporting tools that allow you to easily track both student progress and your training budget. (Hall, 1997)

When you compare e-Learning with traditional classroom training, e-Learning has many advantages. One advantage is cost savings. Corporations and schools often need to stretch funds while maintaining high-quality training programs. The cost of developing, distributing, and maintaining an e-Learning application is often much less than the cost of an instructor-led training program. Online learning can help organizations let the most out of their budgets and can add valuable learning tools and techniques.

Another advantage to e-Learning is the ability to update any topic in your published content and deliver the results instantly over the Internet or an intranet. Now all of your learning material can be up-to-date without the expense of reprinting a book, storing inventory, or manually distributing an update. In an online course, you can deliver an unlimited number of colorful graphics and special effects that direct attention to the important points in the learning material. Students can interact with new material right away, by answering practice questions and directing their own learning pace. (Morrison, 2003)

2. MODERN E-LEARNING SYSTEMS

Project planning is a preliminary step in which you evaluate your resources, decide your distribution strategy, develop a user profile, and then document your research and choices. This process, while sometimes time consuming, is effort well spent: the result is a road map for your project that can help you to measure expectations, activities, and results. In addition, planning can help maximize resources—if you get it right the first time, you won't have to spend time and money doing it over. This section describes some factors to consider when planning your e-Learning course.

Understanding the people for whom you're developing an e-Learning application is crucial to the success of the project. Developing a user profile will help you to keep focused on your audience and to judge the appropriateness of your content and methods. You can use both formal and informal methods to analyze your users—consider interviewing them, observing their work, visiting trade shows, or reading their professional journals or documentation materials.

When gathering information for a user profile, determine some of the following. How experienced are they with the material? What age ranges or cultural groups do they represent? What is their level of education? What are their learning styles? How computer-savvy are they? Try to evaluate their environment as well. For example, do they sit, stand, or move around? Is it noisy or quiet, dark or bright? These factors affect your choice of content, instructional approach, and page layout.

A modern e-learning system is based on the concept of lessons and their versions. A Lesson is in fact a discussion topic. The peoples that work on a lesson will provide several implementations of that specific topic. (Horton and Horton, 2003)

There are several types of users found within the e-learning system. Some of those users can be considered "producers" while others can be called "consumers". In the following all the users found within the system are presented:

- *Lesson Developer* The lesson developer is the creative force behind the lesson system. He can browse the lesson catalog then select and download an empty lesson for further development. He also has the possibility to request the editor for the holder right of a particular lesson An author can upload a lesson to the Collaboration Area – a lesson pool where he (and the Co-Authors) can share (upload and download) intermediate lesson versions. If the Author has Holder right, he can also submit his version for revision to the Editor as Request to Edit Version (REV). Alternatively, he can select one of the versions in the Collaboration Area as the Request to Edit Version and submit it to the Editor. Note: each lesson has its own Collaboration Area for the authors working on it. Once the lesson is submitted to the Editor for review, the Request to Edit Version is locked until an Editor either qualifies the lesson for the next level (Publishing) or returns feedback to Author on how to further improve the content. Lesson sharing is also freely available outside the Collaboration Area – the Author can export the local lesson to one file and send it by e-mail or disk to other Authors.
- *Extended Developers* A E-developer is like the Lesson Developer with additional rights granted on a lesson basis. As such, E-developer can perform all the developer actions described in the above section. In addition below are described the tasks that are specific to a E-developer.
- *Learner* The Learner is the ultimate target of the System. Upon registration within the system a learner can download the Learner Kit; he follows interactive lessons and takes administered exams and tests. He has an assigned Tutor, who creates his Learning Path and monitors his progress. Tools are provided for seamless communication between the Learner and the Tutor. The System extracts and uploads relevant data from Learner's study patterns and exams, allowing the Tutor insightful performance reviews.
- *Tutor* The Tutor is the direct supervisor of a Learner. He creates (or selects from a system list) the Learner's Learning Path, evaluates his performance and provides the Learner with feedback on the results and progress. The Tutor uses tools to freely communicate with his Learners.
- *Editors* The Editor is responsible with issuing and revoking authorization for Lesson Developers; he grants and revokes the E-Developer right to an Lesson Developer for a specific lesson. The Editor also assures the quality control for all new lessons and re-created lessons. He analyzes the requests submitted by Lesson Developers to add new lessons in the catalog and reviews a lesson submitted by an Author as Request to Edit Version.
- *Publisher* The Publisher receives the lessons approved by the Editor and performs a final verification on the content. In this he mainly checks for violations of the Law of Author Rights. If such violations are encountered, he informs the Editor and provides him with

feedback on the reasons. In case the lesson is approved for publishing, the lesson is marked as ‘Currently Published’ and thus available to be chosen in the learning path of students.

Each actor represents a distinct entity with behavioral characteristics within the e-learning system. Since the Lesson is at the core of the system organization and workflow, we will present a graphical representation of the standard Lesson workflow – from it’s inception to review and further to inclusion in the high quality, production lesson area. (Gagne and Medsker, 1996)

2.1 Lesson Workflow

Several steps have been identified within a lesson’s workflow, which are described as follows:

1. The Lesson Developer asks for an empty lesson to be inserted into the lesson catalogue and gives the necessary description for it
2. The Editor validates Lesson Developer ,srequest and inserts a new empty lesson into the Lesson Catalogue
3. Editor inserts (on self initiative) a new empty lesson into the Lesson Catalogue
4. Lesson Developer browses the Lesson Catalogue and downloads empty lesson to be modified in Author module
5. Lesson Developer (Extended Developer) selects Co-Authors for a lesson
6. Lesson Developer and Co-Lesson Developer create the lesson in the Collaboration Area
7. Extended Developer uploads the created lesson to Editor for review. This version submitted to Editor for review is called Request to Edit Version
8. Editor reviews lesson (can edit and make changes)
9. Lesson is accepted and Editor submits lesson to Publisher. This version submitted to Publisher for review is called Request to Publish Version
10. Lesson is declined and Editor informs Lesson Developers of this
11. Lesson is published by Publisher and available to students for download. At this point the lesson has status Published Version
12. Editor makes a request to Publisher to mark a lesson as ‘Out of Publish’
13. Publisher marks a lesson as ‘Out of Publish’ (on self initiative)
14. Publisher marks a lesson as ‘Archive’ when it no longer appears in any student learning path (Cooper, 1995).

2.2 Lesson Structure

The lesson is at the core of the entire system. Physically a lesson can be viewed as a zip archive which contains HTML, XML and image files and a set of metadata which is stored into the local database. Each lesson contains a HTML page where a short description of the lesson is written. From the logical point of view a lesson is a collection of tasks, each of it containing a set of problems.

2.3 Lesson Tasks

A task is physically represented by a HTML page and a set of images. Inside a task theoretical aspects are presented. Each task covers a small direction of study and its purpose is to teach the student something related to the lesson topic. Based on the information contained within the HTML page there are several types of tasks:

- Entry tasks
- Main tasks
- Parallel Tasks
- Comprehensive tasks
- Explorer tasks
- Test tasks

When a task is defined by the author it is automatically added to one of the previously defined groups. For each task a name is stored and a flag which signaled that it can be added to a final test. In the viewer the test will be generated dynamically and it will contain 5 tasks chosen from a task pool which contains the test tasks defined by the author and the other tasks which have the “can be in test flag” set. The only tasks that can not be added to tests are Entry tasks and Explorer tasks.

Entry tasks. Those tasks contain information that helps the student to remember thinks that he learned in the previous lessons and that can be used to solve the current lesson. Those tasks can not be added in tests as they are not strictly related to the current lesson’s topic. This kind of tasks usually contains very simple problems and the score the user get at those problems does not count very much for the final lesson score.

Main tasks. Those tasks contain information that helps the user to understand the current lesson. They can be added to tests. The learner is advised to finish all main tasks before even starting the test, although this is not mandatory. Solving their problems well is crucial for the user to obtain a good final score from the tutor.

Parallel tasks. The parallel tasks are not strictly related to the lesson, but to a main task. This means that a main task can have several parallel tasks attached. The idea behind the parallel tasks is to present the same idea presented in the main task, but in a different form. By reading the parallel task the user can understand better the lection. Those tasks can be added in tests. Since those tasks can be considered a part of the main tasks, solving them is somehow crucial for the user to obtain a good final score from the tutor.

Comprehensive tasks. If the main tasks present the lesson from a more theoretical point of view, the comprehensive tasks help the user to understand how to apply what he learned or how to interconnect the separate main tasks information. Those tasks can be added in test. Also, solving their problems well is somehow crucial for the user to obtain a good score.

Explorer tasks. Those tasks contain information which takes the user a little further that the lesson boundaries. Usually the user can find here details about the lesson which is not actually required for passing the lesson test. Those tasks are meant for users that want to study more that it is required. Those tasks can not be added in tests, but solving their problems well can give the student some additional points in the final score.

Test tasks. Those tasks, as their name suggest, are designed only for tests. They usually contain only brief theoretical information, but solving their problem is mandatory for the user to obtain a good score. As their name suggests, those tasks can be found in tests. Usually an author chooses to create several test tasks which will for the final test instead of marking main, parallel or comprehensive tasks as tests. The reason is quite simple to understand.

2.4 Lesson problems

As I've presented earlier a task can contain several problems. In fact solving those problems is the main purpose of the entire lesson. It is not important if the student had read the task information as long as he managed to solve the attached problems. From physical point of view a problem can be seen as a collection of HTML, XML and image files. If a task contained only a single HTML page, a problem contains several, as follows:

- *A problem information page:* this page contains general information about the problem. It contains the problem body and the problem question. It is displayed when the user starts working on a problem
- *A hint page:* this page contains some hints that will help the user when solving the problem. When the problem appears in a test, the hint page can not be displayed for obvious reasons.
- *A solution page:* this page contains the solution for the problem. It can be displayed only after the user has finished solving the problem. The solution can be defined as a set of steps. The author is the one that decides this. The user will then see one step at a time.

Beside the html pages the problem contains an XML file which holds the problem options and their truth values. The structure of the XML files varies based on the problem class, as it will be presented in the following.

The system permits 3 problem classes:

- Multiple choice
- Matching
- Open Answer.

Multiple choice. Multiple choice problems, as their name suggests, contain a set of clickable options each having an attached true/false value. The user can select several options. A selected option can not be unchecked neither clicked again. Thus the truth value is strictly related to the option. The answer XML file looks as follows

```
<?xml version="1.0" encoding="UTF-8"?>
<choice
        xmlns="http://www.elearning_site.net"
        xmlns:xsi="http://www.w3.org/2001/XML
        Schema-instance"
        xsi:noNamespaceSchemaLocation="./sche
        ma.xsd">
        <options>
           <option optionid="1" truth="1">
                OPTION 1 TEXT
           </option>
           <option optionid="2" truth="0">
                OPTION 2 TEXT
           </option>
           <option optionid="3" truth="1">
                OPTION 3 TEXT
           </option>
        </options>
        <feedbacks>
           <feedback feedbackid="1">
                FEEDBACK 1 TEXT
           </feedback>
           <feedback feedbackid="2">
                FEEDBACK 2 TEXT
           </feedback>
           <feedback feedbackid="3">
                FEEDBACK 3 TEXT
           </feedback>
        </feedbacks>
</choice>
```

As it can be seen from the previous XML each option has an attached id and a truth value. Also, for each option, there is a feedback which will be displayed to the user after the option has been selected. (Roberts, 1999; Schmeiser, 1997, William and Tollet, 1998)

The MC class contains 3 types of problems: classical multiple choice (MC), select (SLT) and self check (SC).

Select problems are a particular case of Multiple Choice problems. The main difference is that the options are only text.

Self check problems consist on two phases. In phase one the student has to freely write a text based on a given topic. After this writing is done, the student is asked several questions about the content of his writing. Each question has a truth value attached. Based on its answers, the system will provide a score. The actual score of this problem will be computed by the tutor after analyzing the writing and comparing it to the answers.

Matching. The Matching problems contain a set of clickable options called "leaders" and another set called "followers". Each leader is logically connected to one and only one follower. The user has to first click on a leader and then on a follower. If the two are logically connected, the selection is marked as true. Otherwise the selection is considered false. The answer XML file looks as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<matching
       xmlns="http://www.elearning_site.net"
       xmlns:xsi="http://www.w3.org/2001/XML
       Schema-instance"
       xsi:noNamespaceSchemaLocation="./sche
       ma.xsd">
       <leaders>
           <leader leaderid="1">
              Leader 1
           </leader>
           <leader leaderid="2">
              Leader 2
           </leader>
           <leader leaderid="3">
              Leader 3
           </leader>
       </leaders>
       <followers>
           <follower followerid="1">
              Follower 1
           </follower>
           <follower followerid="2">
              Follower 2
           </follower>
           <follower followerid="3">
              Follower 3
           </follower>
       </followers>
       <links>
          <link leaderid="1" followerid="2">
          </link>
          <link leaderid="2" followerid="1">
          </link>
          <link leaderid="3" followerid="3">
          </link>
       </links>
       <feedbacks>
           <feedback feedbackid="11"
              leaderid="1" followerid="1">
              leader 1 -> follower 1
           </feedback>
           <feedback feedbackid="21"
              leaderid="2" followerid="1">
              leader 2 -> follower 1
           </feedback>
           <feedback feedbackid="31"
              leaderid="3" followerid="1">
              leader 3 -> follower 1
           </feedback>
           <feedback feedbackid="12"
              leaderid="1" followerid="2">
              leader 1 -> follower 2
           </feedback>
           <feedback feedbackid="22"
              leaderid="2" followerid="2">
              leader 2 -> follower 2
           </feedback>
           <feedback feedbackid="32"
              leaderid="3" followerid="2">
              leader 3 -> follower 2
           </feedback>
           <feedback feedbackid="13"
              leaderid="1" followerid="3">
              leader 1 -> follower 3
           </feedback>
           <feedback feedbackid="23"
              leaderid="2" followerid="3">
              leader 2 -> follower 3
           </feedback>
           <feedback feedbackid="33"
              leaderid="3" followerid="3">
              leader 3 -> follower 3
           </feedback>
       </feedbacks>
</matching>
```

The structure of this file is more complicated compared to the previous one. First there are the two option lists: leader and followers. For each of them an id is known. The connection between leaders and followers is stored in the link fields. For each leader there is an attached link field which holds the information about the attached follower. Link fields contain no data. The last group of tags is the feedbacks group. If for MC problems there were a feedback for each option, here there is a feedback for each leader-follower pair. Every time the user selects such a pair the appropriate feedback is displayed. (Roberts, 1999; Schmeiser, 1997, William and Tollet, 1998)

The MC class contains 4 types of problems: classical matching (MT), select and write (SW), recall and write (RW) and drag and drop (DD).

The select and write problems are a special case of Matching problems. The main difference is that leaders represent input text areas. The user sees the list of followers and then writes one of those inside a leader text field. If that follower and that leader are identical, truth is signaled.

Recall and Write problems are identical to the previous ones, the only difference being that the followers are not displayed. The user has to write text inside the leaders without knowing a list of possible options.

Drag and Drop problems are a special case of matching problems. In this case the leaders are fragments of an image, while the followers are text. The user have to click on a follower and then drag and drop it on a leader.

Open Answer. In those problems the learner has to write freely a text based on a given topic. This text is then sent to the tutor who estimates it and it gives the student a score. Those problems do not have an answer XML file as there is no expected answer. Also no solution or hint page is available for the same considerations. The system can not compute any estimate score and it is only the tutor who will give any estimation. (Walter and Carey, 1996)

3. CONCLUSIONS AND FUTURE WORK

Modern e-learning systems are no longer based on a simple web-site with a few types of users. The main entity is now the lesson. Also, several types of actors can be found in those modern systems. Each user have a limited role within the system and thus he does not require an intensive training for using the entire system.

Once the application is placed on a Web server, it is available immediately to your audience. And the exchange is mutual: e-Learning allows course developers to track and evaluate student scores, giving them the information they need to measure the effectiveness of their courses. (The American Journal of Distance Education, 1997)

At this moment an implementation of the ideas presented in this paper exists. The product uses a client-server architecture. There are implemented two tools: an authoring tool and a viewing and evaluation tool.

The authoring tool used by authors (Lesson Developers) to create the lessons. The lessons are created as a collection of html pages and images. The authoring tool contains an integrated complex html editor.

The vieweing and evaluation tool is in fact a modified web-browser. With it the students can view and solve the problems from a lesson. At the end the application computes their score and it displays it.

In the future I intend to improve the existing tools and also to create similar application for the other types of users (editor, tutor, publisher etc.)

4. REFERENCES

Cooper, A. (1995). About face: the essentials of interface design, IDG Books, Foster City, CA.

Gagne, R. and K. Medsker (1996). *The conditions of learning: training applications.* Fort Worth, TX: Harcourt, Brace.

Hall B.(1997) *Web-Based Training Cookbook.* John Wiley & Sons, Inc.

Horton W. and K. Horton (2003*), E-learning Tools and Technologies*, Wiley Publishing

Morrison D.(2003), *E-learning Strategies* , Wiley & Sons, Inc.

Roberts S. (1999), *Programming Microsoft Internet Explorer 5,* Microsoft Press, ISBN: 0735607818

Schmeiser, L. (1997). *Web design templates sourcebook,* New Riders Publishing, Indianapolis.

The American Journal of Distance Education *Vol. 11, No. 3 1997, p. 76 - 78*

Walter D. And L. Carey (1996). The Systematic Design of Instruction, 4th ed. Harper Collins, New York, NY.

Williams, R. and J. Tollett (1998). *The non-designer's web book,* Peachpit Press, Berkley, CA.