



IMPROVING COURSE INTERACTION AND MANAGEMENT WITH *TESTING ASSISTANT*

Traian ANGHEL¹
Adrian FLOREA¹

¹“Lucian Blaga” University of Sibiu, Computer Science Department, Emil Cioran Street, No. 4, 550025 Sibiu, Romania, traian_gk@yahoo.com, adrian.florea@ulbsibiu.ro

Delilah FLOREA²

²“Samuel von Brukenthal” National College of Sibiu, Informatics and Computer Science Department, Huet Square, No. 5, 550182 Sibiu, Romania, delilah_florea@yahoo.com

Abstract: *The ever growing popularity of the World Wide Web provides the possibility of extending the didactic activity over the internet. As many professors build the required competences in using modern applications, the concept of e-Learning is gaining importance as it eases the process of learning, communication and collaboration in Education. Modern web applications are nowadays fast, reliable and scalable, and implementing an e-Learning system into higher education facilities has never been easier. Because students tend to be in touch with the latest in software development, they find it easy to work on such platforms and learn more efficient because of the new teaching methods implemented. Further in this work we extend our previously started project - Testing Assistant - with new learning and student management modules, developed with the purpose of optimizing the whole process of teaching by providing a strong alternative for an e-Learning system.*

Keywords: *e-Learning system, interactive and collaborative application, 1-Bit Full Adder, coherence / consistence, e-Testing, e-Management*

I. INTRODUCTION

Computers have a great importance in our everyday lives. They have radically changed many fields, like: communication, transports, science and economy. But their impact on teaching and learning wasn't so big due to teacher training and complicated software. Many instructional software tools are neither fully sensitive to student's requirements nor flexible enough to emulate teaching. Besides that, a computer cannot fully understand more complex statements, like an essay, and cannot correctly evaluate it, like a professor can do. On the other hand, with the help of images, videos and other media content, many processes can be better explained with the help of adequate software, while instant feedback can be provided in tests where computer software is able to evaluate the answers of students. Modern intelligent tutors based on artificial intelligence help to learn about multidisciplinary approaches to using computing systems for teaching as well as gain insight through technical reports on research, development and real-world experiences [1].

Modern students (also called "digital natives" [2]) are familiar with the latest in hardware technology and gadgets (personal computers, notebooks, 3G mobile phones, palm, various embedded systems) but also with the Internet and the new information and communication technologies. They

differ from students belonging to other generations as from their educational background and their set of abilities. They prefer experimental, visual and interactive learning in a collaborative environment with the help of web applications, facilitating acquiring and information processing with the purpose of managing multiple tasks at the same time (course reading, taking notes, searching online for information). Recent studies have shown that 35% of Romanian students spend 4 to 6 hours per day in front of the computer, while 43% spend 1 to 3 hours, most of the time using the Internet [3]. The Internet is becoming a key component of young people's lives; popular activities, like online communication and social networks are gaining ground in comparison with other professional ones, such as researching and documentation. It is the main source of information, beating by far newspapers, specialty magazines and books. However, the same study reveals that 30% of the youth see the Internet as a source of distraction from more productive activities and a waste of time. Also, students are much motivated when they develop studying activities that use the new technologies, thus encouraging their learning potential [4]. Online technologies for student assessment provide a greater feedback potential helping students to change their attitude against study.

In our opinion, the main goal of educational software is to design an educational system that assesses a student's knowledge and adapts to a student's learning needs. Through our implemented application we meet the learning requirements of modern students. In this work we extend our previously started project - Testing Assistant [5, 6] - with new learning and student management modules, developed with the purpose of optimizing the whole process of teaching by providing a strong alternative for an e-Learning system. We illustrate with an interactive lesson about digital logic circuits, a fundamental topic present in Computer Science and Information Technology, Electrical Engineering and Multimedia Systems. Our application provides an interactive and collaborative documents and projects development similar with "GoogleDocs", "Yahoo Groups", facilitating also discussions and exchanges of ideas amongst teaching actors: professors, students, researchers and administrators of higher education to implement and improve the innovations in teaching and higher education management.

The organization of the rest of this paper is as follows. In section 2 we present a general description of *Testing Assistant*, the component modules (e-Learning, e-Testing and e-Management) and relationships among them. Section 3 shows what is new in Testing Assistant (the novelty in our work) – flash files and images in lessons and Presence integration. Section 4 describes the user interface and system requirements emphasizing on *1-Bit Full Adder*, an interactive lesson about digital logic circuits. In section 5 we discuss some security issues implemented in our applications, including a backup platform for lessons. Finally, section 6 suggests directions for future works and concludes the paper.

II. GENERAL DESCRIPTION OF TESTING ASSISTANT

The Testing Assistant tool consists in three ample modules: e-Learning, e-Testing and e-Management (see Figure 1). They are carefully implemented to meet the needs of three aspects of teaching: learning, testing and course management, with the purpose of easing the work of professors and to provide an easy-to-use system for the students as well.

Testing Assistant divides users into three categories, namely administrator, professor and student. They all have different access levels to different modules, each of them in strong connection to one another. The basic actions that can be done by each user type are:

- *Administrators* can create accounts for all user types and manage specializations and subjects from within a university.
- *Professors* can create lessons, exams and can manage student presence at course. In addition, they can view the marks given for any student.
- *Students* may view lessons, enter exams and have an overview of all their received marks over time.

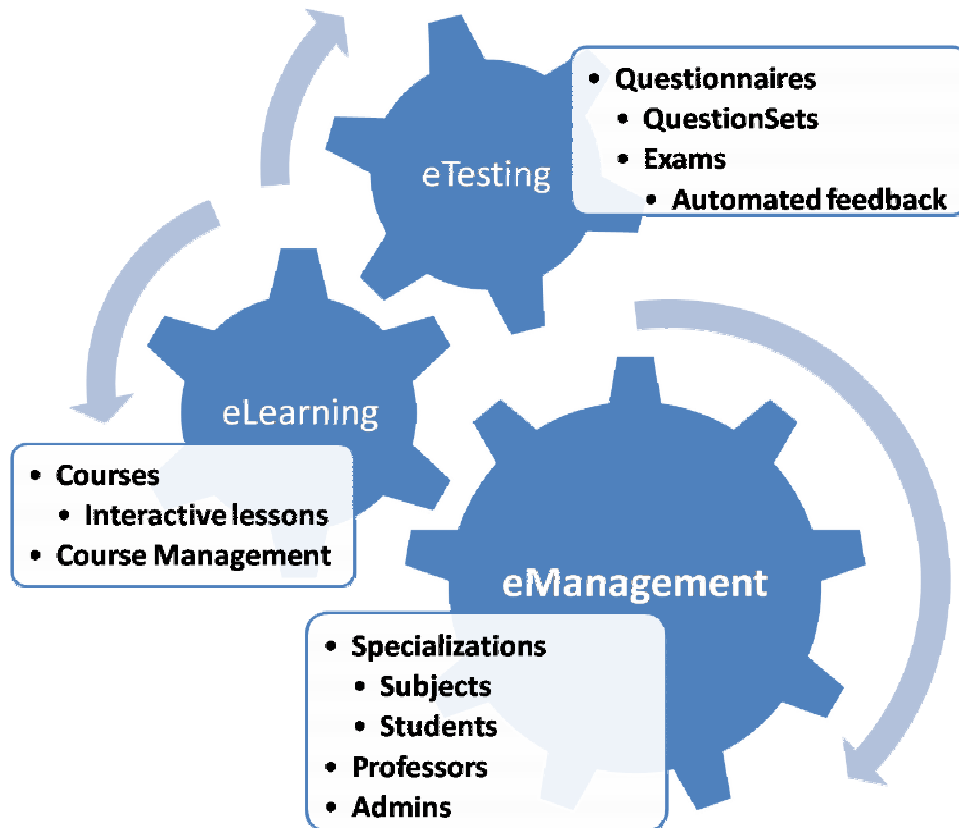


Figure 1. The modules of *Testing Assistant*

2.1 e-Learning module

This module offers the ability to create lessons which will be available for students at any time. The system provides a simple, interactive and very intuitive user interface for editing lessons with the ability of lesson editing and different object linking and embedding similar to Microsoft Word (Figure 2).

Besides that, there might be the need for co-authoring lessons within a certain course. The professor which first adds a course is the main user and has the possibility for adding one or more collaborators that are assigned to working on the same materials within that course. The collaborators (usually teaching assistants) are limited to editing the individual lessons of a course, not the course parameters, privileges which are present only for the main user of a course.

One of the problems that might appear when multiple users share the same lesson is that there could be the case when two or more users want to edit the lesson at the same time (database coherence / consistency problem). In order to avoid this situation, we implemented a "check-out" system for lesson editing. Whenever a user enters a lesson in edit mode, other users are constrained just in viewing it until that user closes the editor (sequential consistency is implemented in Testing Assistant). This system ensures that data integrity is preserved with the small price of not providing the ability of editing a lesson at the same time.

In this manner we implemented through our application an interactive and collaborative document development similar with "GoogleDocs". Because of the advantages of our lesson editing system, the authors of this work have completely developed this article using Testing Assistant, as a better alternative to sending updated documents per email. In this way, all the editing that was done on the article was done on the final version of that document (we have always kept the information coherent).

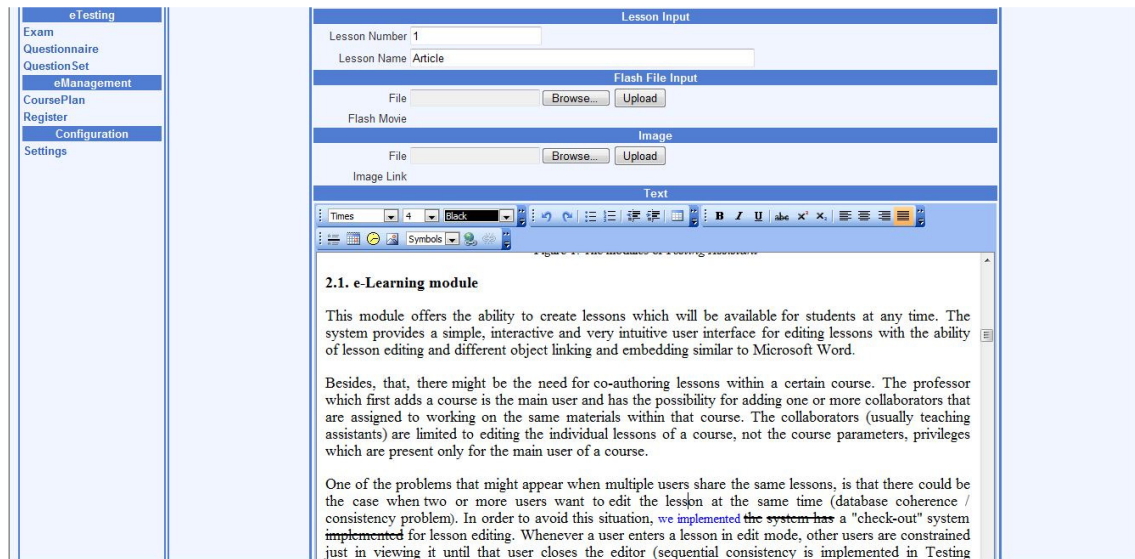


Figure 2. Lesson input

2.2 e-Testing module

Exam preparation

For an exam to take place, it must be formerly prepared by entering a *Questionnaire*. It basically consists of a set of *Questions* specified by the professor. These questions are organized in *Chapters*, and a set of chapters make up a *Questionnaire*. In its current development state, the system offers 4 *question types*:

- *Textbox*: the student provides the answer; it will be compared letter by letter (in a non case-sensitive approach) with the control answer
- *Checkbox*: the student is asked to choose one or more answers from a set of possible ones (multiple choice single answer or multiple choice multiple answer)
- *Ordered*: the student is shown a list of random items; he must choose them in the correct order
- *Essay*: the student writes a more complex answer which will be manually evaluated by the professor after exam finishing

Attached to the text of the question, it is possible to have an image (Figure 3). The same applies to individual answers for the question types *Checkbox* and *Ordered*. If the image is big (for example, photos taken with modern digital cameras), it will be automatically resized to 640x480 pixels for an image attached to a question and 320x240 pixels for an image attached to an answer. This guarantees that images will not exceed the browser space offered by the browser and no horizontal scrolling will be needed. The resizing will be done taking into consideration the aspect ratio of the image and preserving it on modification.

For further helping with the exam preparation, Testing Assistant offers the possibility of shrinking the selection from a *Questionnaire* for partial examining or exams where not all questions from a questionnaire are needed. This is done with the help of a *QuestionSet*.

A *QuestionSet* represents a subset of questions defined in a *Questionnaire*, individually chosen by the professor. This may be helpful for the case of partial examining, where the professor chooses only questions within chapters that were studied until that moment. *QuestionSets* will be used in exam planning, detailed below.

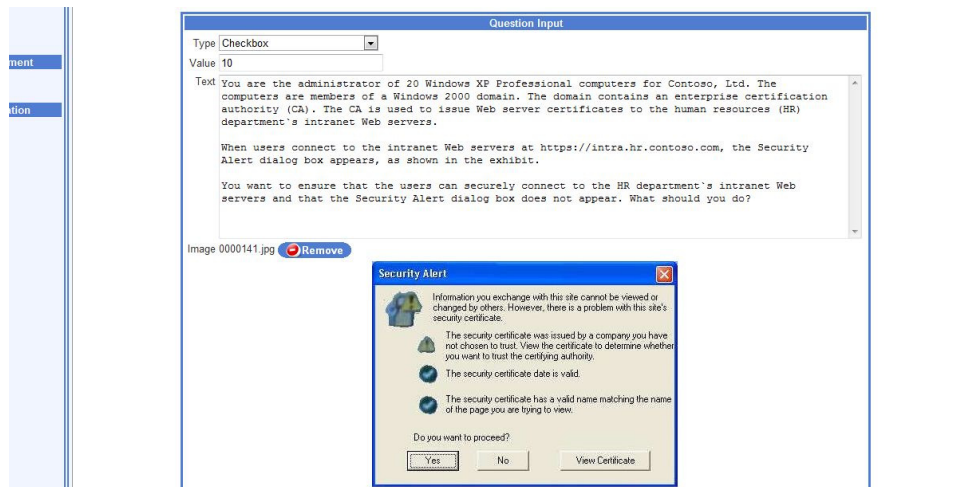


Figure 3. Question input

Exam planning

Exam planning basically consists of choosing a *Questionnaire*, how the questions will be drawn from it, one or more exam dates and students selection for each exam date (see Figure 4). Questions can be chosen in two ways, either random or manual with the help of a *QuestionSet*. For the random type, the user will specify a number of questions that will be randomly chosen individually for every student. In order to emphasize certain chapters which may have greater importance in an evaluation, the value of every chapter is taken into consideration when fetching the questions from it. For example, if there are two chapters, one with the value 10 and the other 20, the system will fetch from the second one twice as many questions than from the first one. In addition, questions from within the same chapter will be shuffled as well as answers for questions with their type set as *Checkbox* or *Ordered*, thus limiting the chances for students to copy from one another. As a result, if there are enough questions entered by the professor, the system will make sure that there is little probability for two students to share the same questions and if that happens, the order of the answers (in case of *Checkbox* or *Ordered* type questions) will not be the same.

The professor will create an exam date for each different place and time where the exam will take place (depending on groups of students). He will choose a date and time for each of them and the students that will take part. Once this step is completed, each student that has been programmed for a certain exam date will see the date and time on his personal Testing Assistant account.

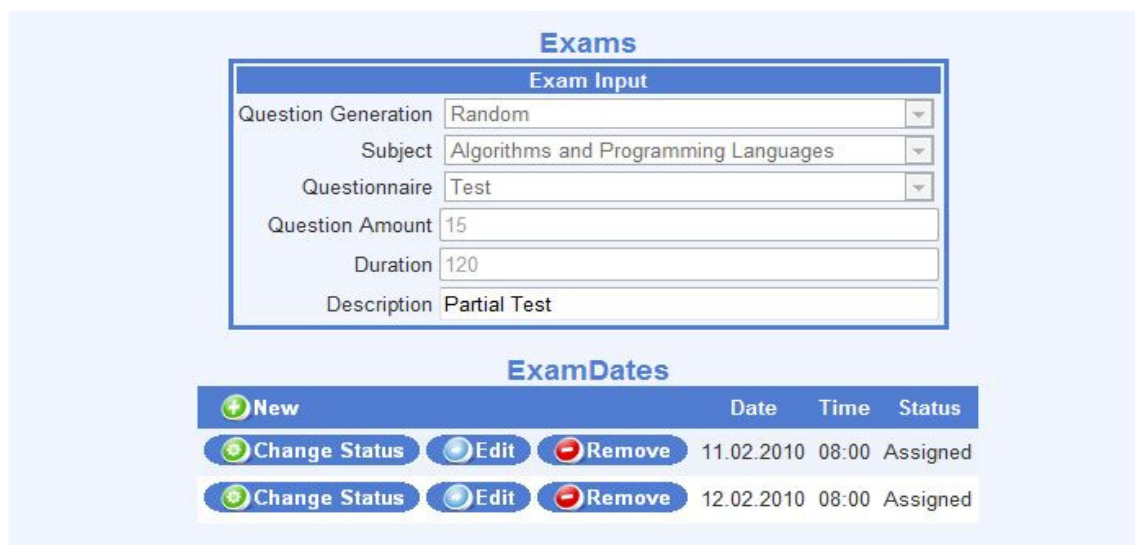


Figure 4. Planning an exam

Exam availability

When entering an exam, the student will see the time he has left until finishing it and the questions which he has to answer (Figure 5). When the remaining time reaches to zero, its exam session will be finalized, his answers are evaluated and the obtained mark will be shown, along with other statistics.

In case of disconnecting from the Internet, accidentally closing the browser or other unwanted events happens during an exam, the students are able to resume the exam and recover all of their answers. Every time the student scrolls through the questions, his answers are saved to the database, minimizing the chance of data loss.

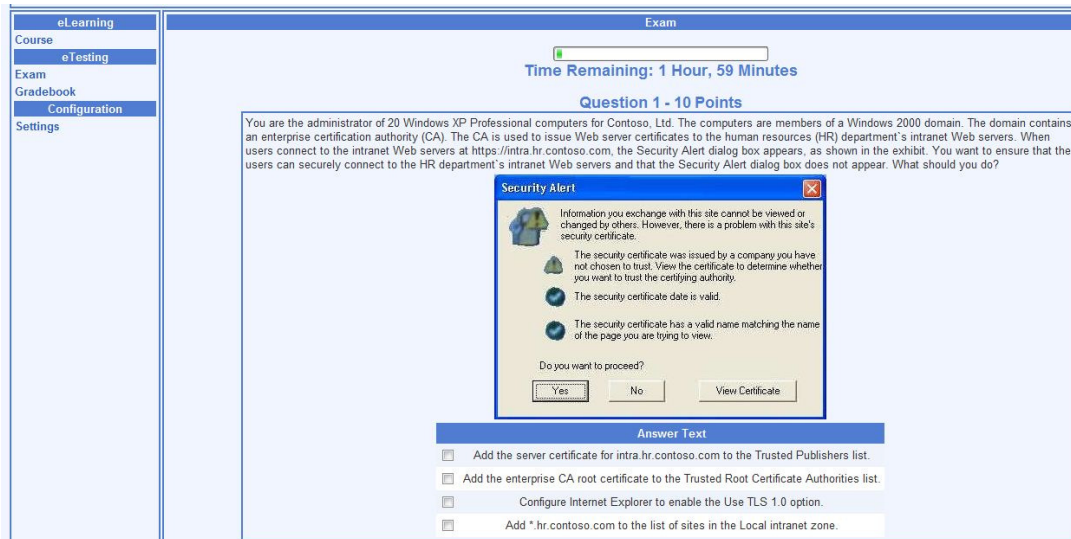


Figure 5. Exam page for student

Exam feedback

In the process of exam feedback the mark is shown instantly, with students having the possibility to view correct/incorrect answers; the professor sees the marks and answers of his students along with details on essay question type and has the possibility of changing the mark (for various reason) and bringing it back to its original state.

Testing Assistant contains a complex exam feedback system both for students and for professors. For the student, it shows his mark instantly when finalizing an exam. He also is able to visualize the evaluation of each of his answers and in case of incorrect answers the correct one is shown. This method is vital for the student as it helps him better understand his weaknesses and gives him the chance of a better preparation for the final exam. This feedback system for the student is very important for its response time, offering the student the ability of learning from his mistakes as his reasoning behind every answer remains fresh in his memory.

The feedback system is very important for the teacher, as well. He is able to see the mark awarded to every student and more than that, he may actually see the answers of every student (Figure 6). It is useful for the professor as he can keep a close eye on certain students that show interest to him, for example he may want to know why a good student got a low mark.

There remains one case that has to be covered - the case of correcting *Essay* type questions. For each student that has finalized his exam, the professor evaluates all questions that require an essay to be written and awards points accordingly. The mark is then automatically recalculated taking into consideration the awarded points; as a result, both the professor and the student can see the updated mark.

Welcome, Professor Traian Anghel **TESTING ASSISTANT** Log Out

eLearning		Exams					
e Testing		Students & Results					
Exam	Gradebook	Exam Details	Student Name	Student Group	Exam Status	Correction Status	Mark
			Bota Iustinian	224/1	Finalized	Not Corrected	10
			Mojic Radu Horatiu	224/1	Finalized	Not Corrected	10
			Patrusel Marius	224/2	Finalized	Not Corrected	10
			Tataroiu Bogdan Gheorghe	224/2	Finalized	Not Corrected	8.2
			Danila George Cristian	224/1	Finalized	Not Corrected	6.4
			Nitu Diana Alexandra	224/1	Finalized	Not Corrected	6.4
			Banescu Madalina	224/2	Finalized	Not Corrected	4.6
			Tescut Ovidiu	224/2	Finalized	Not Corrected	3.4
			Anghel Traian	224/2	Finalized	Not Corrected	1
			Duda Mihaela	224/1	Finalized	Not Corrected	1
			Tera Ionel	224/2	Finalized	Not Corrected	1
			Alex Govoreanu	224/2	Assigned	Not Corrected	
			Andreea Stanchi	224/1	Assigned	Not Corrected	
			Barzan Codrut Ion	224/1	Assigned	Not Corrected	
			Brad Radu	224/2	Assigned	Not Corrected	
			Bradeanu Bogdan Iulian	224/1	Assigned	Not Corrected	

Figure 6. Student results for an exam

2.3 e-Management module

This complex module is dedicated for managing parts of the system that are not directly related to teaching or testing. In our system this can be seen from three distinct perspectives.

Administrator

In the case of administrator, this module is used to set the base layer for everything that will be used later in the system. The administrator may create / edit / delete other administrators, professors, students, specializations and subjects (Figure 7). His purpose is to prepare the system to be used by the professors and students, but although he has control over them, he does not have access to their specific activities, like lessons or exams. His only job is to ensure that the organization of the university is reflected into the system.

Fullname	Email
Adrian Florea	adrian.florea@ulbsibiu.ro

Subjects for selected Professor			
Name	Year	Semester	
Computer Organization and Design	1	1	
Microprocessing Systems	3	1	
Object Oriented Programming	1	1	

Subjects			
Name	Year	Semester	Specialisation
Algebra	1	1	Computer Science
Mathematical analysis	1	1	Computer Science
Algorithms and Programming Languages	1	1	Computer Science
Special Mathematics	1	2	Computer Science
Professional Development	1	2	Computer Science

Figure 7. Assigning subjects to a professor

Professor

The e-Management module provides the professor the ability to visualize the marks of any student in a simple and reachable way, along with other properties, like subject, professor, date and time. Along with this small module called *Register*, the professor may also manage student presence in a simple, straightforward module. The presence is shown in form of a table, with the rows as students and the columns as the date. In this way he can always have a simple, accessible overview of his

student presence at any time, saving him time and effort for an otherwise daunting task, especially in ‘Lucian Blaga’ University of Sibiu, where strict rules are applied versus presence.

Register			
Marks			
Professor	Subject	Mark	Date & Time
Traian Anghel	Algorithms and Programming Languages	1	2/11/2010 11:06:15 AM
Traian Anghel	Science	1.75	1/19/2010 4:18:19 PM
Arpad Gellert	Analiza si proiectarea algoritmilor	1	1/19/2010 12:25:13 PM
Traian Anghel	Computer Organization and Design	3.12	1/9/2010 11:36:41 PM
Traian Anghel	Science	3.02	11/26/2009 8:15:22 AM
Traian Anghel	Science	1.91	11/10/2009 7:55:36 PM
Traian Anghel	Science	2.85	11/10/2009 7:53:51 PM

Figure 8. Student Grade Book from the Register module

Student

The student is provided with a *Grade book* which presents him his marks (Figure 8). Its visual part is the same like the professor’s **Register**, with the difference that only his marks are shown with no chance to see marks received by his colleagues, ensuring privacy for each and every student.

III. WHAT’S NEW IN TESTING ASSISTANT

3.1 e-Management module

Testing Assistant provides full support for independent, third party flash files as part of lessons (Figure 9). The uploaded flash file will appear before the text of the lesson and may come in three possible dimensions: small (320x200 pixels), medium (640x480 pixels) and large (800x600 pixels), depending on the requirements of the flash file. The flash format is a very popular one, providing multiple possibilities for displaying active content, ranging from videos to interactive lessons. In real-life situations, these flash files can be projected on big screens during courses, helping the professor to better explain certain aspect of the lessons, which otherwise would have demanded a lot of explanatory effort.

When using static gates as building blocks, the most fundamental latch is the simple SR latch, where S and R stand for set and reset. It can be constructed from a pair of cross-coupled **NOR** (Not OR) logic gates. The stored bit is present on the output marked Q.

Normally, in storage mode, the S and R inputs are both low, and feedback maintains the Q and \bar{Q} outputs in a constant state, with \bar{Q} the complement of Q. If S (Set) is pulsed high while R (Reset) is held low, then the Q output is forced high, and stays high when S returns to low; similarly, if R is pulsed high while S is held low, then the Q output is forced low, and stays low when R returns to low.

Figure 9. Interactive lesson viewed from a student point of view

3.2 Images in lessons

Besides flash files, lessons may contain also images. Adding of images requires multiple easy-to-understand steps. In figure 10 we can see the steps needed to follow for inserting an image in a lesson. Files are uploaded on the server and are referenced within the lesson, ensuring in this way that they will always be visible when the lesson is accessible. A flash movie can be inserted in a similar manner.

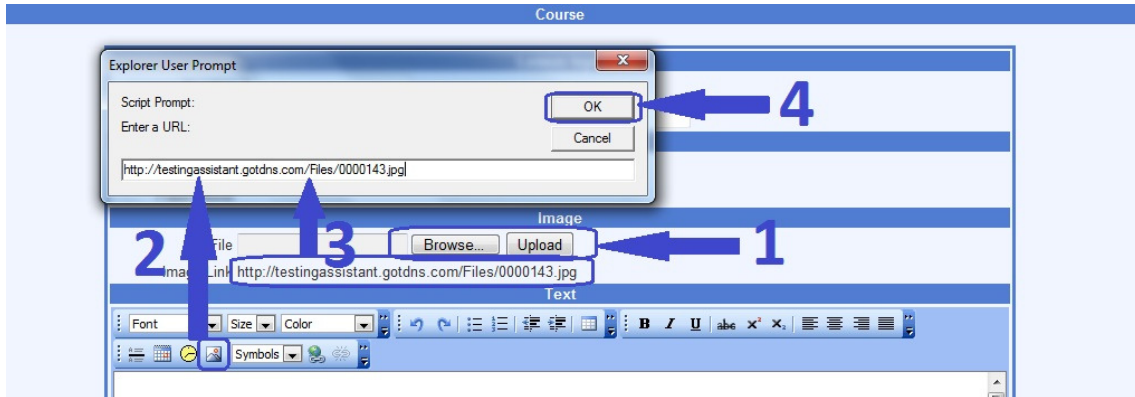


Figure 10. Snapshot with inserting an image in a lesson

3.3 Presence integration

Presence management has always been a daunting task for professors. Presence history for only a semester was kept in piles of paper, making the job even harder for him. Although Microsoft Excel is a good alternative solution for that, it still cannot automate the process by requiring to enter each student name for identification, which can be a daunting task as well if the professor teaches a course to hundreds of students.

Testing Assistant provides a straightforward management system for presence. All the professor has to do is choose the students he will teach to, then for each course session he will enter the presence marker corresponding for each student. The presence is shown in form of a table, with rows representing students, and columns individual course sessions. Each combination of student-course session will have its own presence marker, which can be customized in any way possible (e.g. the marker may have the following possible states: present, absent, motivated absent). By using this method to manage the course presence, the professor may save precious time. Besides that, it offers a good overview for the professor and as a further development, the application will also present statistics about presence in form of charts.

Welcome, Professor Traian Anghel																																																																																																																																																																																																						
TESTING ASSISTANT																																																																																																																																																																																																						
Log Out																																																																																																																																																																																																						
CoursePlan																																																																																																																																																																																																						
<ul style="list-style-type: none"> eLearning Course eTesting Exam Questionnaire QuestionSet eManagement CoursePlan Register Configuration Settings 	Filter by Group: 224/2																																																																																																																																																																																																					
	Legend: p: Present a: Absent m: Absent Motivated f: Free -: Not selected																																																																																																																																																																																																					
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Students / Lessons</th> <th>09.12.2009</th> <th>24.12.2009</th> <th>25.12.2009</th> <th>26.12.2009</th> <th>27.12.2009</th> <th>28.12.2009</th> <th>29.12.2009</th> <th>30.12.2009</th> <th>31.12.2009</th> <th>01.01.2010</th> </tr> </thead> <tbody> <tr><td>Alex Govoreanu</td><td>p</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr> <tr><td>Anghel Traian</td><td>p</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr> <tr><td>Banescu Madalina</td><td>p</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr> <tr><td>Brad Radu</td><td>a</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr> <tr><td>Buta Virgil Nicolae</td><td>p</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr> <tr><td>Conea Andrid Daniel</td><td>p</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr> <tr><td>Corcoveanu Cristian Andrei</td><td>a</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr> <tr><td>Fratila Cosmin</td><td>a</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr> <tr><td>Muntean Madalin</td><td>p</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr> <tr><td>Negrila Radu</td><td>a</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr> <tr><td>Podariu Marius</td><td>p</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr> <tr><td>Podariu Alexandru Daniel</td><td>a</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr> <tr><td>Tataroiu Bogdan Gheorghe</td><td>p</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr> <tr><td>Tera Ionel</td><td>p</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr> <tr><td>Tescut Ovidiu</td><td>p</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr> <tr><td>Tirzianu Florin Ioan</td><td>p</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr> </tbody> </table>											Students / Lessons	09.12.2009	24.12.2009	25.12.2009	26.12.2009	27.12.2009	28.12.2009	29.12.2009	30.12.2009	31.12.2009	01.01.2010	Alex Govoreanu	p	-	-	-	-	-	-	-	-	-	Anghel Traian	p	-	-	-	-	-	-	-	-	-	Banescu Madalina	p	-	-	-	-	-	-	-	-	-	Brad Radu	a	-	-	-	-	-	-	-	-	-	Buta Virgil Nicolae	p	-	-	-	-	-	-	-	-	-	Conea Andrid Daniel	p	-	-	-	-	-	-	-	-	-	Corcoveanu Cristian Andrei	a	-	-	-	-	-	-	-	-	-	Fratila Cosmin	a	-	-	-	-	-	-	-	-	-	Muntean Madalin	p	-	-	-	-	-	-	-	-	-	Negrila Radu	a	-	-	-	-	-	-	-	-	-	Podariu Marius	p	-	-	-	-	-	-	-	-	-	Podariu Alexandru Daniel	a	-	-	-	-	-	-	-	-	-	Tataroiu Bogdan Gheorghe	p	-	-	-	-	-	-	-	-	-	Tera Ionel	p	-	-	-	-	-	-	-	-	-	Tescut Ovidiu	p	-	-	-	-	-	-	-	-	-	Tirzianu Florin Ioan	p	-	-	-	-	-	-	-	-	-
	Students / Lessons	09.12.2009	24.12.2009	25.12.2009	26.12.2009	27.12.2009	28.12.2009	29.12.2009	30.12.2009	31.12.2009	01.01.2010																																																																																																																																																																																											
Alex Govoreanu	p	-	-	-	-	-	-	-	-	-																																																																																																																																																																																												
Anghel Traian	p	-	-	-	-	-	-	-	-	-																																																																																																																																																																																												
Banescu Madalina	p	-	-	-	-	-	-	-	-	-																																																																																																																																																																																												
Brad Radu	a	-	-	-	-	-	-	-	-	-																																																																																																																																																																																												
Buta Virgil Nicolae	p	-	-	-	-	-	-	-	-	-																																																																																																																																																																																												
Conea Andrid Daniel	p	-	-	-	-	-	-	-	-	-																																																																																																																																																																																												
Corcoveanu Cristian Andrei	a	-	-	-	-	-	-	-	-	-																																																																																																																																																																																												
Fratila Cosmin	a	-	-	-	-	-	-	-	-	-																																																																																																																																																																																												
Muntean Madalin	p	-	-	-	-	-	-	-	-	-																																																																																																																																																																																												
Negrila Radu	a	-	-	-	-	-	-	-	-	-																																																																																																																																																																																												
Podariu Marius	p	-	-	-	-	-	-	-	-	-																																																																																																																																																																																												
Podariu Alexandru Daniel	a	-	-	-	-	-	-	-	-	-																																																																																																																																																																																												
Tataroiu Bogdan Gheorghe	p	-	-	-	-	-	-	-	-	-																																																																																																																																																																																												
Tera Ionel	p	-	-	-	-	-	-	-	-	-																																																																																																																																																																																												
Tescut Ovidiu	p	-	-	-	-	-	-	-	-	-																																																																																																																																																																																												
Tirzianu Florin Ioan	p	-	-	-	-	-	-	-	-	-																																																																																																																																																																																												
Copyright (c) 2009 Anghel Traian																																																																																																																																																																																																						

Figure 11. Presence management

IV. THIRD PARTY LESSON: USER GUIDE AND SYSTEM REQUIREMENTS

For a proper use of Testing Assistant tool (enclosing the independent modules attached as lessons), some system requirements should be accomplished. Thus, it is recommended to have a processor with at least 1 GHz frequency. The RAM memory recommended is 256Mbytes. To have a good view, it is required a 1024x768 minimum screen resolution (Video Memory minimum 1 MB but we recommend 8 MB). The application will run regardless of the operating system we use: Windows 2000/Millennium/XP/Vista/7. Also, any kind of browser that interprets HTML code (Internet Explorer, Mozilla Firefox, Opera, Google Chrome, etc.) can be used but we recommend Mozilla Firefox. The third party lesson about *Full-Adder* was developed using Macromedia Flash trial version.

In this paper we focus on the *1-bit Full Adder* – an interactive lesson about digital logic circuits, a fundamental topic present in Computer Science and Information Technology, Electrical Engineering and Multimedia Systems. According to [7] the Full Adder is “*a logical circuit that performs an addition operation on three one-bit binary numbers*”. The full adder produces a sum of the two inputs (X_n and Y_n) and a carry in value (C_n). Based on the Truth Table exhibited in Figure 13, the expressions of Sum bit (S_n) and Carry out bit (C_{n+1}) are:

$$S_n = \bar{X}_n \cdot \bar{Y}_n \cdot C_n + \bar{X}_n \cdot Y_n \cdot \bar{C}_n + X_n \cdot \bar{Y}_n \cdot \bar{C}_n + X_n \cdot Y_n \cdot C_n = C_n \cdot (\bar{X}_n \cdot \bar{Y}_n + X_n \cdot Y_n) + \bar{C}_n \cdot (\bar{X}_n \cdot Y_n + X_n \cdot \bar{Y}_n) = C_n \cdot \bar{X}_n \oplus \bar{Y}_n + \bar{C}_n \cdot X_n \oplus Y_n = C_n \oplus X_n \oplus Y_n \quad (1)$$

$$C_{n+1} = \bar{X}_n \cdot Y_n \cdot C_n + X_n \cdot \bar{Y}_n \cdot C_n + X_n \cdot Y_n \cdot \bar{C}_n + X_n \cdot Y_n \cdot C_n = X_n \cdot Y_n \cdot (C_n + \bar{C}_n) + C_n \cdot (\bar{X}_n \cdot Y_n + X_n \cdot \bar{Y}_n) = X_n \cdot Y_n + C_n \cdot X_n \oplus Y_n \quad (2)$$

Figure 11. Full Adder Simulator – expressions of Sum bit (S_n) and Carry out bit (C_{n+1})

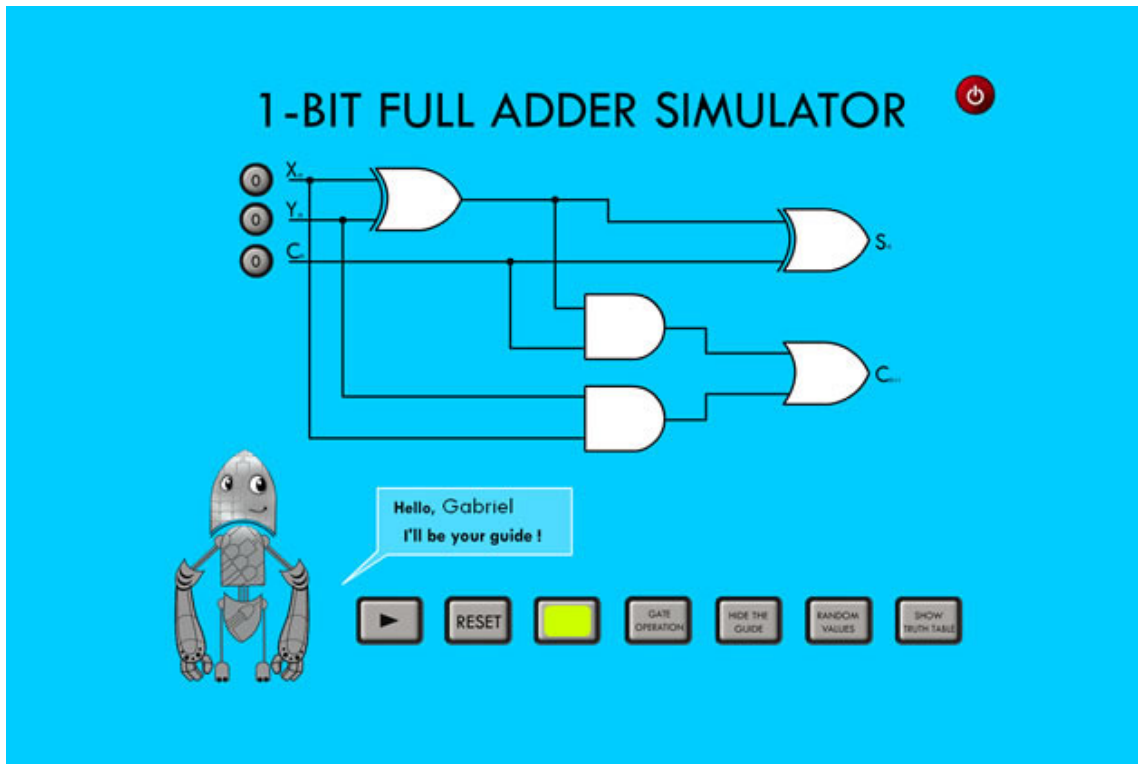


Figure 12. Simulator window on startup

The simulator as shown below was conceived to be a friendly e-Learning tool. In student view and for hitting its purpose, learning should be fun. When somebody learns a new thing, or gets a

surprise, there is a shot of a chemical messenger in his brain called dopamine [8]. Dopamine is famous among neuroscientists for its involvement in the reward and motivation systems of the brain. Many students are getting bored at some point because they are not receiving information in a proper way. The teachers have to be smart enough to trap learner's attention. If the student can learn something new and have fun in the same time that's a much better chance that him would actually understand what's going on, how certain stuff actually works. The visual experience of a user while sitting in front of the computer is very important. If he can run the simulation over and over again, playing with the input values, there's a higher chance that he would easily understand the working principle of the logic gates (in this case). For this reason, the lesson is presented with the help of a user Guide. The Guide's purpose is to bring a smile on learner's face and it's not just a childish thing. The Guide will tell the user what does each button when is pressed, just to follow its instructions. (Just as a note, the Guide is actually made of logic gates, take a closer look!) Thus, Have fun and learn in the same time!

The simulator presents a window where the user may edit the input values (X_n , Y_n and C_n) as shown in figure 12, start the simulation, and then the magic happens. The flow of the signal is shown through the logic gates with the result for each operation and how the final result is formed. The simulation may be repeated as many times as the user wants with different input values. Throughout the lesson, the user has a little help in the lower left part of the screen which will guide him with hints and information about certain operations within the lesson. It can be deactivated at any time if the user finds it annoying. In addition, the user has some options available to make the lesson more customizable, like showing the operation each gate performs (AND, OR, XOR, etc) or shuffling the input values. Also, a popup with the truth table may be activated, which will show all possible results for each input value (Figure 13).

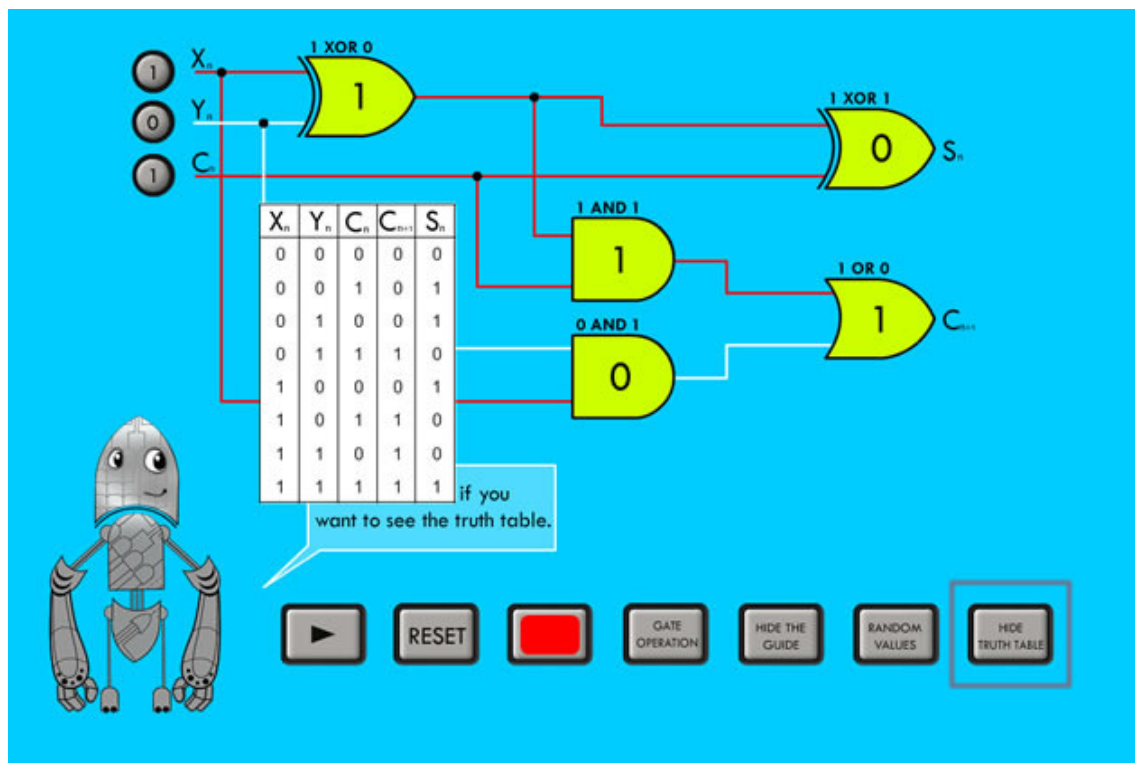


Figure 13. Simulator with gate operation enable and with truth table popup

V. SECURITY AND SCALABILITY ISSUES

Because the project we are working on gets bigger and bigger, from time to time there appear certain issues that cannot be overcome easily but with a good concept behind. There have been cases when concepts were good from the beginning but also when they had to be changed along with the development of the system.

We encountered a serious issue when we thought of what would happen if a hacker would try to break our system. The first thing he might do is issue a *sql* injection. For example, take the login page. The text from the password is compared with the password from the database. Because the hacker assumes there is a test like this on the database, he "injects" certain *sql* statements to the database, revealing him database structure or data. This has been overcome from two points of view. Even if the hacker injects *sql* statement data, he will be redirected to a safe, general page which will tell him that an error occurred (not where it occurred because that may show him sensitive data). The second issue we have overcome was to reject *sql* injection at all. This has been done by replacing all single quote (`) characters with back quote (`), thus not allowing the hacker to deliberately end the *sql* statement and issue another.

The second thing the hacker may do is trying to "guess" the password by doing repeated tries for a certain user. This has been, as well, overcome from two points of view. The first solution was not to show the hacker if the username or the password were incorrect, thus forcing him to know at least one username. The second solution was that in case of a faulty password for a user, in the next 5 seconds any password he may type will not be taken into account and an error message will be shown (Figure 14). With time it will be improved with other, more sophisticated approaches such as *md5* or *sha-1* encryption algorithms [9, 10].

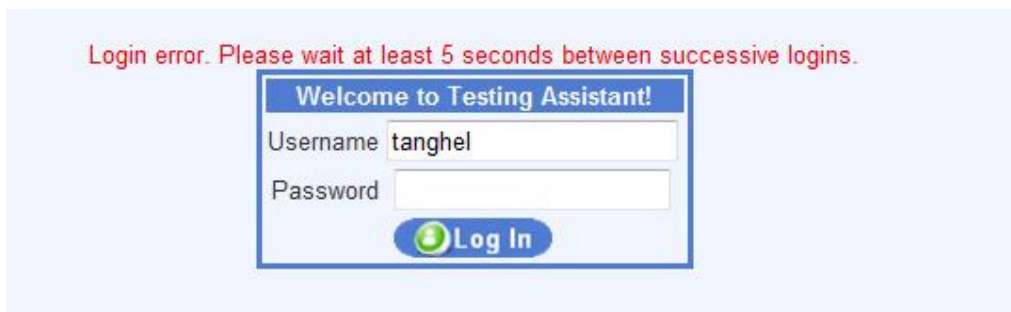


Figure 14. Login error

Another issue we encountered during development was *view state* data. This basically contains properties of all items present on the webpage, properties which will be accessible to the programmer in source code, encrypted and stored in the html of the page in form of a hidden field. As the application was becoming more and more complex, more and more data was stored in the *view state*, becoming as big as 20 kb for every request. It is a huge amount of information taking into account that for every request (button click, for example) it is posted back to the server and downloaded again, along with the html. Besides that, the basic html of the page is as big as 10 kb, half of the content of the *view state*. The source of this problem was that when loading the login page, all the items present in the system were loaded but not visible. Although hackers would never have gotten to them, it overloaded the html source with useless information (the only useful information is the one that is visible). Implementing a method of runtime loading for every item that was needed solved the problem, decreasing *view state* data to 10% of its initial size, and will be further optimized if needed.

The most recent issue we had was during the writing of this article. One of the authors accidentally deleted all the text that has been written until that moment and it couldn't be recovered. To make sure this situation never happened in the future, a versioning mechanism was implemented. It basically keeps the previous 10 updates of each lesson and if something bad happens to the current document, its content can be rolled back to one of its previous states (Figure 15). During the writing of the article after this feature was implemented, the same situation happened again but the data could be recovered with minimum loss.




















TESTING ASSISTANT				
Course				
Previous versions				
		Date	Time	Professor
 View	 Rollback	11.02.2010	09:59	Adrian Florea
 View	 Rollback	11.02.2010	10:13	Adrian Florea
 View	 Rollback	11.02.2010	10:21	Adrian Florea
 View	 Rollback	11.02.2010	10:22	Adrian Florea
 View	 Rollback	11.02.2010	12:00	Adrian Florea
 View	 Rollback	11.02.2010	12:00	Adrian Florea
 View	 Rollback	11.02.2010	12:00	Adrian Florea
 View	 Rollback	11.02.2010	14:01	Adrian Florea
 View	 Rollback	11.02.2010	14:03	Traian Anghel
 View	 Rollback	15.02.2010	19:46	Traian Anghel

Figure 15. Lesson versioning

VI. CONCLUSIONS AND FURTHER DEVELOPMENT

Through our implemented application we meet the learning requirements of *digital natives*' students. Thus, in this work we continue our project – Testing Assistant tool (started about 12 month ago) extending it with new learning and student / course management modules, developed with the purpose of optimizing the whole process of teaching by providing a strong alternative for an e-Learning system.

We improved the e-Learning module permitting to define lessons with text, integrating media files such as images, flash movies and video clips. We illustrate with an interactive lesson about digital logic circuits, a fundamental topic present in Computer Science and Information Technology, Electrical Engineering and Multimedia Systems. Our application provides an interactive and collaborative documents and projects development similar with “*GoogleDocs*”, “*Yahoo Groups*”, facilitating also discussions and exchanges of ideas amongst teaching actors: professors, students, researchers and administrators of higher education to implement and improve the innovations in teaching and higher education management.

Also, we developed an e-Management module with proper administration of marks, averages, lessons and student's presence at courses / laboratories. Finally, these reports could be printed.

As a natural next step of our work we will implement further a notification and communication system such as (e-mail, RSS feeds, personal blogs, Forum and FAQ – frequently asked questions, wikis, podcasts and other powerful web tools for learning in a collaborative manner) very useful in any e-Learning system. Thus, every student when he logs in his Testing Assistant account will be informed about new lessons introduced by professors or other interesting, important and useful information (newsletter, news and events, conference announcement, exam schedules, etc).

ACKNOWLEDGEMENTS

We like to thank to our students Ciprian Pascu and Gabriel Cioflica from Multimedia Systems specialization for providing useful and competently help in technical developing of some third party lessons.

REFERENCES

- [1] Woolf B.P., *Building Intelligent Interactive Tutors*, Elsevier, 2007, ISBN-13: 978-0-12-373594-2.
- [2] Anghel T., *Instrumente WEB 2.0 utilizate in educatie*, Editura Albastra, 2009, ISBN 978-973-650-243-9, Cluj Napoca.
- [3] Istrate Olimpius, Studiu Intel: Tendinte in utilizarea tehnologiei de catre studentii romani, Elearning.Romania, 2010-01-03. Bucuresti: TEHNE- Centrul pentru Dezvoltare si Inovare în Educatie, www.elearning.ro/resurse/Intel_2009_studiu_studenti_Hermes.pdf.
- [4] Istrate Olimpius, *Modalitati de utilizare a TIC pentru evaluarea studentilor*, Elearning.Romania, 2009-03-22. București: TEHNE- Centrul pentru Dezvoltare și Inovare în Educație (<http://www.elearning.ro>).
- [5] Florea A., Anghel T., Florea D., **Error! Hyperlink reference not valid.**, International Conference on Embedding Innovation in Teaching and Management of Higher Education (ICITM 2009), Shah Alam, Selangor, Malaysia, December, 2009, PST Enterprise SDN BHD Publishing Hall, ISBN 978-983-41527-1-0, pp.87 (1-13).
- [6] Anghel T., Florea D., Florea A., **Error! Hyperlink reference not valid.**, The 7th National Conference for Virtual Learning - Virtual Reality, News Technologies in Education and Research, Models & Methodologies, Technologies, Software Solutions, pp.136-143, ISSN 1842-4708, Iasi, 2009.
- [7] *Full Adder*, [http://en.wikipedia.org/wiki/Adder_\(electronics\)](http://en.wikipedia.org/wiki/Adder_(electronics))
- [8] *Dopamine receptor*, http://en.wikipedia.org/wiki/Dopamine_receptor
- [9] <http://en.wikipedia.org/wiki/MD5>
- [10] http://en.wikipedia.org/wiki/SHA_hash_functions