# Enhanced Learning and Educational Management through Online Collaborative Technologies

Adrian Florea[1], Arpad Gellert[1], Traian Anghel[1], Delilah Florea[2]
[1]Computer Science Department
“Lucian Blaga” University of Sibiu
{adrian.florea, arpad.gellert}@ulbsibiu.ro, traian_gk@yahoo.com

[2]“Samuel von Brukenthal”
National College of Sibiu
delilah_florea@yahoo.com

*Journal of Digital Information Management*

***ABSTRACT:*** *Merging traditional ways of teaching with modern approaches as e-Learning, focused on online multimedia applications and technologies, developed in a collaborative manner, became a real necessity in order to enhance learning and educational management. In this paper we introduce our developed Education Management Tool (EDM), a visual, interactive and collaborative e-Learning solution, dedicated for optimizing the teaching, learning and testing processes in different Computer Science topics but also in any other scientific field. EDM allows geographically distributed students to contribute concurrently and collaboratively by working on the same application: designing, programming, debugging, testing and documenting. Besides that, EDM is free, easy to use, and helps to improve quality and performance in e-Learning, providing flexibility, extensibility and interactivity. The EDM Learning module allows defining lessons that consists of text-descriptions and media files such as images, flash movies or video clips. We illustrate how third-party lessons can be integrated into EDM by presenting our interactive flash applications about digital logic circuits and algorithms, two fundamental topics of Computer Science. One of the latest development achievements consists in a Management module, with proper administration of marks, averages, lessons and student’s presence at courses and laboratories, which allows generating, printing and exporting reports. Another issue, extremely necessary during collaborative writing and programming, is versioning, a mechanism which allows recovering any of the last ten versions of lessons or source codes.*

**Categories and Subject Descriptors**
**I.2.6** [Learning]; **K.3** [Computers and Education]; Collaborative learning

**General Terms:** Visualization, E-Learning, Web 2.0, Educational Management Tool

**Keywords:** e-Learning, Education Management Tool, Collaborative technologies, Web applications, Digital logic circuits, Algorithms

**Received:** 11 September 2010; Revised 31 October 2010, Accepted 3 November 2010

## 1. Introduction

Although web services and applications are, for most of us, as common as driving a car or going to a fast-food, no application really had a major impact on e-Learning communities. Thus many of today’s teachers still teach with the help of a blackboard and a piece of chalk, most of the exams are still written in ink and corrected by professors until late at night. Despite the fact that education systems have been investing in technology for more than twenty years, international indicators on technology uptake and use in education are missing (Scheuermann & Pedró, 2009). The Lifelong Learning concept involves lifelong education, which can be achieved using both by competent teachers and modern educational resources. Traditional education (classroom lectures, texts, and individual assignments) is clearly not up to the task. Current educational practices are strained to their breaking point (Woolf, 2007). Taking into account the multitudes of possibilities that the Web 2.0 has to offer, one would think that the communion between classical learning and e-Learning would be inevitable. Using simulators, different information resources such as e-books or creating software tools dedicated for testing the student knowledge, multimedia and online collaborative technologies for synchronous learning, using a virtual classroom application, and formative interaction in teaching act became a real necessity in order to enhance learning and educational management (Moreland, Jones & Chambers, 2001). Furthermore, the constructivist visions of education (Jonassen, 1999) claim a participative role for the learner. The managerial role of teacher, which deals with class and course management, requires greater attention to detail, a better lesson / testing structure, and additional student monitoring according to (Guerra, Fabbri & Maeran, 2008). The professor should document for preparing new lessons / chapters making the bridge with prerequisite courses and announcing the future topics. The information must be exposed through an attractive and interactive manner by “*bringing the student to play*”.

Still, a certain degree of reluctance exists among the teachers, especially the elder ones, in harnessing the power of the online applications and using them to their own favor. Also, the research studies indicate that the elaboration and supply of educational software are dissatisfactory for important segment of teachers, the biggest impediment being the software acquisition cost problem. Thus, visual applications developed by teachers, individually or through collaborative projects, with and for students, are really valuable and welcomed (Shen & Sun, 2000).

On the other hand, there is a theoretical limit for software that makes it so hard to overwhelm the teaching communities worldwide. Just like the semantic web that still cannot be implemented due to conceptual limitations, e-Learning applications can do everything, but cannot be as sensitive to the learners` needs as a teacher can. Teachers need to be experts in specialized disciplines, as a result of academic qualifications and research activities and continuously develop competences in teaching methodologies that focus on empowerment of increasingly applied knowledge in the process of lifelong learning and development. Every teacher has his own life experiences and wisdom to draw upon. Each of us has a unique voice which will resonate with some people, but not with everybody. No two people think exactly alike and no two people learn exactly alike either. Some of the students are ready to "receive" the information transmitted by teachers and others probably are not ready yet. However, a prerequisite of any university topics refers to the student's ability and his high motivation to learn and engaging his whole energy to support that motivation. Although applications can present information, they cannot adjust themselves to present this information in a manner that everyone understands. Also, for now, software cannot be aware of itself, but has to obey strict rules that the programmer imposes. Thus, applications cannot interpret and evaluate, for example, an essay written by a student about physics or any other topic. Just because the teaching, learning and testing processes are exponentially complex, they cannot be integrated into an application that obeys strict rules and behavioral patterns.

The solution for this conflict is as simple as it gets. Take the best from each part, combine them, and you get efficient learning and testing. Take the years of evolutionary knowledge of teachers and their presentational skills, combined with PowerPoint presentations with the help of a projector where he/she doesn`t have to spend time on drawing a complex figure on the blackboard, because it is already there with the help of a mouse button. Take the power of instant feedback examination software that usually takes less than a second to correct answers to different question types interpretable by a machine. Take the task of managing attendance for hundreds of students at several, different subjects, using pencil and paper and the easiness that Excel tables with automatic sums bring for this situation. Take online collaborative technologies that fulfill all of these tasks specific to the teachers' and learners' needs and you begin to get a grasp of the utility of e-Learning in the teaching and testing process. Combining the technical dimension of using e-Learning tools with the pedagogical dimension of putting them into practice, teachers' competences using current technologies are emphasized, whilst putting an accent on the fact that "*motivation is more important than technology*".

Coming in support of the above said, we present in this paper Education Management Tool (EDM), our visual and interactive e-Learning solution, dedicated for optimizing the whole process of teaching / learning / testing in different computer science topics. EDM promotes collaboration and professional development. Our application provides interactive and collaborative development of documents and projects, similar with "*GoogleDocs*", "*Yahoo Groups*", facilitating also discussions and exchanges of ideas amongst educational actors: professors, students, researchers and administrators of higher education to implement and improve the innovations in teaching and higher education management. Through real time collaborative programming and application development, EDM allows geographically distributed students to contribute concurrently and collaboratively by working on the same application: designing, programming, debugging, testing and documenting (Bravo C., Marcelino M.J., Gomes A., Esteves M. & Mendes A.J., 2005). Using general knowledge about social networks through a collaborative programming manner, the users of EDM will have as target not only accelerating problem resolution processes, but will substantially improve the quality of the software applications that they build. Besides that, EDM is free and easy to use, and helps to improve quality and performance in e-Learning, providing the following advantages: flexibility, extensibility, interactivity.

In the academic world the professor is time-pressed due to many activities besides teaching: research grants proposals, reviewing process, conferences, writing keynotes and articles, leading PhD students, etc. In this case, EDM tool (through the Testing module) could be a solution because it reduces the time required for correcting exams. Additionally, it permits both for students and for certain professors, generally self-taught persons, to evaluate themselves and even to learn in advance various courses, providing the feedback regarding to their actual performance potential.

The EDM Learning module permits to define lessons with text, integrating media files such as images, flash movies and video clips. We illustrate with interactive lessons about digital logic circuits and algorithms and programming techniques – two fundamental topics present in Computer Science and Information Technology, Electrical Engineering and Multimedia Systems.

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

The organization of the rest of this paper is as follows. In section 2 we review the Related Work in the field of educational software. Section 3 describes the software design, the system requirements of Education Management Tool and some security issues implemented in our application, including a backup platform for lessons. Section 4 presents some of the latest developments of our application insisting on the Management module. In section 5 we present how we extended the interactive teaching section with lessons about digital circuits, algorithms and programming techniques. Finally, section 6 suggests directions for future work and concludes the paper.

## 2. Related work

For several years some social networks have been successfully used in educational environments. A social learning network consists in ensemble of mutually connected teachers, institutions and learning resources, supported by information and communication technologies. Creating and using a learning network is also important for adults who pursue continuing education as means to advance their professional and personal life. Both educators and students can create / become members of such networks. An open list that includes a significant number of social networks used in educational field can be found on the website of social networks in education (http://www.educationalnetworking.com/List+of+Networks).

International e-Learning platforms such as IBM Education Assistant (http://publib.boulder.ibm.com/infocenter/ieduasst/stgv1r0/index.jsp), Cisco Academies (http://www.cisco.com/web/learning/netacad/index.html) and Oracle (https://academy.oracle.com/), besides teaching modules, include an evaluation component based on quizzes, partial and final exams, and practice exam in Flash. Although the important names from computer science, Intel (http://www.intel.com/education/highered/modelcurriculum.htm), Hewlett-Packard (http://www.hp.com/education/), beside the formerly mentioned IBM, Oracle and Cisco, provide different courses (operating systems, microprocessors, parallel programming, business analysis, project management, database design and programming, etc.), the cost of these courses ranges from hundreds to thousands of dollars. Therefore, EDM – a free software tool – is dedicated for students and teachers from academia especially belonging to "Samuel von Brukenthal" National College and "Lucian Blaga" University, both from Sibiu, eliminating the cost impediment and providing learning opportunities for everyone.

## 3. Software Design of Education Management Tool

In this section we describe the software design, the system requirements of Education Management Tool and some security issues implemented in our application, including a backup platform for lessons.

Education Management Tool (EDM) is a web application that is capable of performing the main tasks provided by more popular e-Learning platforms, having the advantage of being free, accessible from anywhere and from any type of computing system device, not being restricted to the laboratory. The EDM tool target is primarily higher education. It can manage an entire university: specializations, subjects, administrators, professors and students (see Figure 1).

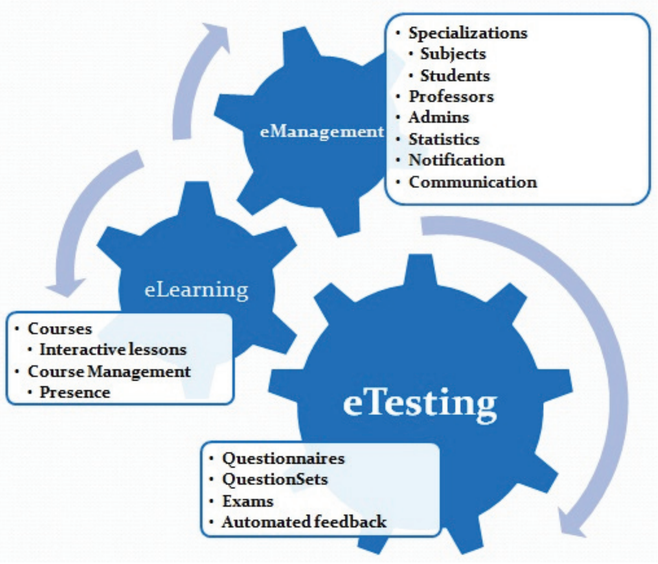


Figure 1. EDM Tool Structure

Consisting of three modules, EDM is designed to meet the basic requirements of an e-Learning environment: learning, testing and management. The learning component consists of courses, developed by teachers in a manner similar to *Google Docs* and which are easily accessible by students. The courses may contain images and flash movies. The testing module consists of exams, customized and posted by the teacher. EDM tool provides a simple and intuitive exam management system that fits any workflow, supporting exam schedule, timed exams and highly editable quizzes, with 4 types of questions. Partial and final assessments will be planned ahead and available to students 24/7 as well as online courses and lessons. When he/she activates it, the students may enter the exam, answer the questions and receive feedback after examination. The management part offers the teacher the ability to manage student grades and presence at courses together with a communication system for delivering messages to specific persons / groups.

The database implemented in EDM Tool, containing learning information as well as system management information, is organized in 39 tables. We have implemented a relational database management system that meets non-redundancy, integrity, consistency, atomicity restrictions. The lines illustrate one-to-many relationships between linked tables, with the yellow key on the "one" side of the relationship, and the infinite sign on the "many" side. For example, each *Specialization* contains many *Subjects* (topics of study). *Teaching* table contains a Professor ID and a Subject ID so that it can uniquely identify a many-to-many relationship between *Professor* and *Subject* (each professor can have one or more subjects and vice versa).

### 3.1 University Management Tables

The tables from Figure 2 refer to user types within the system (Administrators, Professors, Students) and university management entities (Subjects, Groups, Specializations). Each Professor can teach one or more subjects, while each subject can be taught by one or more Professors.

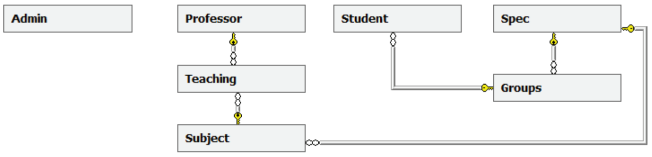


Figure 2. University Management Tables

### 3.2 E-Learning Tables

This module is built upon the idea of making learning objects available for students (see Figure 3). Professors are able to add/edit/delete Courses and Lessons with a one-to-many relationship (each Course contains one or more Lessons). Besides that, each Professor can assign one or more Professors to assist in creating learning materials for students in a collaborative manner. In this way, multiple users can share the same editing materials and edit them when they need it (database coherence / consistency problem), with only one restriction: one user can edit the same learning material (Lesson) at the same time. In order to avoid the situation that might appear when multiple users sharing the same lesson want to edit it the same time, 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 (Adve & Hill, 1990) is implemented in EDM. 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 statistically there is a low probability for two users to "want" to edit the same Lesson at the same time, this solution is a safe one and effective one, proved also by the authors by working on this article in form of a lesson in this system.

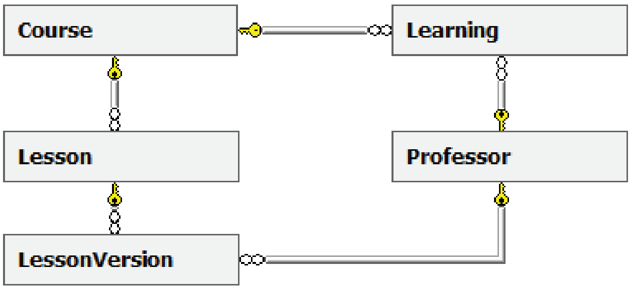


Figure 3. E-Learning Tables

### 3.3 E-Testing Tables

The testing part plays a vital role in any e-Learning environment, because it is the only part that can accurately measure the level of interest and assimilation of the learners. Because of that, extra care and extra functionality was implemented for this part. For the Questionnaire input, there is a four-level table mechanism that holds the data into place. It contains Chapters,

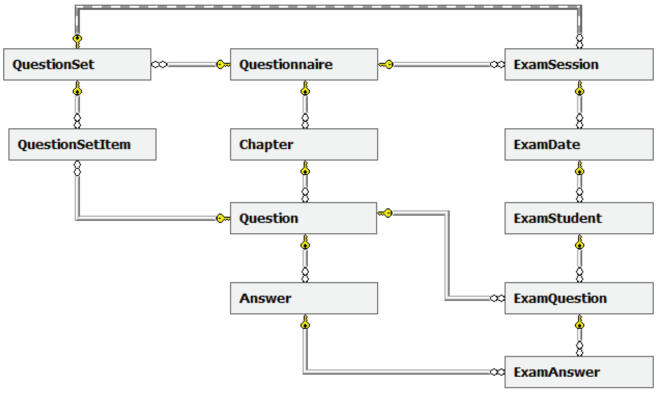


Figure 4. E-Testing Tables

each Chapter contains Questions, while each Question contains Answers. Optionally, the teacher may choose a subset of Questions from a Questionnaire, forming a QuestionSet. The examination part consists of an ExamSession, divided in ExamDates (time and date when the exam takes place along with a subset of students). The students are represented as ExamStudents, comprising ExamQuestions and ExamAnswers, holding the information about what the student answered to each Question.

### 3.4 E-Management Presence Tables

Presence (attendance) management is useful in any learning institution for holding information and statistics about how often students come to visit courses. On the top-level it is represented by CoursePlans, divided automatically in CourseGroups. When the Professors choose CourseStudents for a CoursePlan, students are automatically assigned to different CourseGroups corresponding to each different Group they are part of. LessonPlans represent dates of registering attendance, and each combination of LessonPlan and CourseStudent is represented by a Presence Item, which can have one of the PresenceItem states (present, absent, free, motivated, etc.). In time, it has been proven to be a stable, useful and reliable module, saving precious time for the professors that have used it.

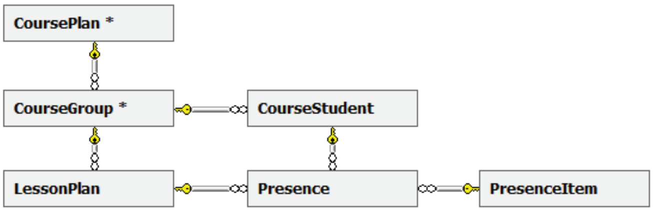


Figure 5. E-Management Presence Tables

### 3.5 The System Requirements

EDM tool was developed using evaluation versions of Visual Web Developer 2008 and Microsoft SQL Server 2008 (our database includes 39 tables) and is running on a server using IIS (Internet Information System 7.5). Programming languages that are used are ASP.NET with C# source code, external ASP.NET objects such as FreeTextBox, FlashControl, javascript libraries (mootools, jQuery) and AJAX extensions.

EDM tool is a web application that runs in client-server mode to ensure optimal performance for the client-side, having as a single requirement an internet connection, with no restrictions for platform and PC performance. From hardware point of view the client could be any type of PC or Web 2.0 capable mobile phone. For a proper use of Education Management Tool, some system requirements should be accomplished both from server and from client side. Thus, on the client side it is recommended to have a processor with at least 1 GHz frequency (x86 or x64-based PC). The recommended RAM memory is 512 MB. To have a good view, a 1024x768 minimum screen resolution is required. From *server* point of view there are the following system requirements:

- NET Framework 4.0. Can be installed on computers running Microsoft Windows operating systems. It includes a large library of coded solutions to common programming problems and a virtual machine that manages the execution of programs. It provides a large range of features including user interface, data access, database connectivity, cryptography, web application development, numerical algorithms and network communications. The space required by .NET Framework is 800 MB.
- IIS 7.5 (Internet Information Services) as web server. It is free available for users of Microsoft Windows 2008 Server.
- Microsoft SQL Server 2008 – program packages for database management system. It should be installed on web server and uses SQL languages for querying the database. Microsoft SQL Server requires 1823 MB of disk space.
- The space required for source code of EDM tool is 14 MB.

From the *client* point of view there are the following system requirements:

- Any kind of browser that interprets HTML code (Internet Explorer, Mozilla Firefox, Opera, Google Chrome, etc.) but we recommend IE8 or Mozilla Firefox 3.6.
- For accessing the application the user has to open the browser and complete in its address bar the server address: **Error! Hyperlink reference not valid.**.

As it can be seen, the system requirements do not exceed the hardware resources provided by nowadays ordinary desktop computers.

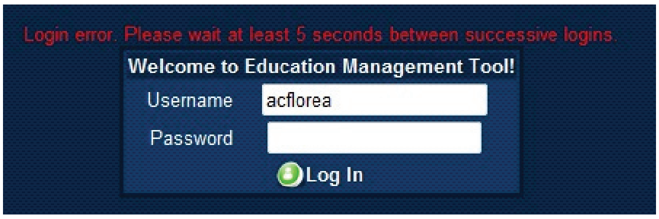
### 3.6 Security and Scalability Challenges

Security is an important issue of computing systems. Every day, millions of people use web applications for commercial, educational and entertainment purposes. Unfortunately, the more successful they are, the more they will be attacked by hackers. Confusing and clumsy code allows attackers to cause countless attacks in order to escalate the security measures. To implement a more robust application, resistant to security attacks, the developer must know both the fundamentals of the web infrastructure and the mindset of an attacker that is looking for vulnerabilities in a computer system and try to exploit them (Heiderich, Vela Nava, Heyes & Lindsay, 2010). The web developer should know and avoid the ordinary possible pitfalls in order to strengthen his defenses. Any computer applications based on Internet technologies requires the implementation of security restrictions from the outset. As later are implemented the security services and restrictions as more costly and harmful are the effects of eventual attacks. In web applications the secure transactions will be necessary while devices (environment) could differ from one place to another. A secure environment must have capability to prevent malicious attacks that intend to intrude inside. Secure environment provides guarantee to cater malicious attack in a decent way, so that no information leak would be commit.

Because the project we are working on gets bigger and bigger, from time to time there appear certain issues that would be overcome easier 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.

Starting from the previously expressed issues we developed within EDM tool an authentication service in order to assign different trust levels to users (student, teacher, administrator) based on the strength of the authentication used. A serious issue that we treated refers to what would happen if a hacker would try to break our system. The first thing he might do is to issue an *sql* injection (Clarke, 2009). 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 the 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 again.

The second thing the hacker might 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, during the next 5 seconds any password he may type will not be taken into account and an error message will be shown (see Figure 6). Improving the application security by approaching more sophisticated encryption algorithms such as *md5* (http://en.wikipedia.org/wiki/MD5) or *sha-1* (http://en.wikipedia.org/wiki/SHA_hash_functions) are on our wish list.


Login error. Please wait at least 5 seconds between successive logins.

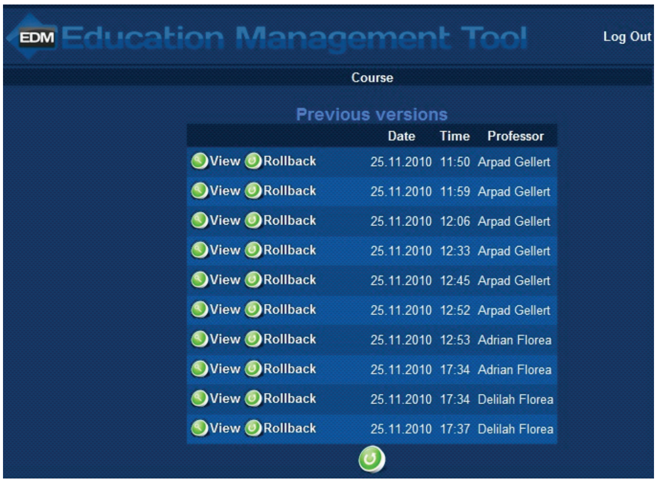
Welcome to Education Management Tool!

Username: acflorea

Password:

Figure 6. 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 has solved the problem, decreasing *view state* data to 10% of its initial size, and will be further optimized if needed.



Previous versions

| | Date | Time | Professor |
|---|---|---|---|
| View Rollback | 25.11.2010 | 11:50 | Arpad Gellert |
| View Rollback | 25.11.2010 | 11:59 | Arpad Gellert |
| View Rollback | 25.11.2010 | 12:06 | Arpad Gellert |
| View Rollback | 25.11.2010 | 12:33 | Arpad Gellert |
| View Rollback | 25.11.2010 | 12:45 | Arpad Gellert |
| View Rollback | 25.11.2010 | 12:52 | Arpad Gellert |
| View Rollback | 25.11.2010 | 12:53 | Adrian Florea |
| View Rollback | 25.11.2010 | 17:34 | Adrian Florea |
| View Rollback | 25.11.2010 | 17:34 | Delilah Florea |
| View Rollback | 25.11.2010 | 17:37 | Delilah Florea |

Figure 7. Lesson versioning

Another issue, extremely necessary during collaborative writing and programming, is versioning, a mechanism which allows recovering any of the last ten versions of lessons or source codes. Initially, if one of the authors accidentally deleted all the text that has been written until that moment, it couldn't be recovered. To make sure this situation will never happen 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 (see Figure 7). 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.

We have also solved some security and scalability problems. One of our latest challenges was when we tested the examination part of our system in a real environment. Although the system was programmed to work fine, without problems, we had some issues when many concurrent users were using the system at the same time. The first problem was with the Internet connection, because the server where we currently hosted the application was not really a server, it was a laptop that connects to the Internet via its wireless connection, which can be slow at times. The second problem we had was the fact that sometimes, the data server (SQL Server) did not "want" to retrieve results any more. As we later found out, there was a problem with the installed version of SQL Server (Express Edition), which had limitations on the number of connections and requests. The third problem was that the resources of the laptop could not cope with the needs of the web server application (first, it was hosted on Windows 7 with many other applications running at the same time). We solved this problem by installing Windows Server 2008 on the machine and stop using it as a home PC. In the future, we plan on moving the application and database to a rented server on the Internet so that all these problems can be solved entirely.

## 4. Latest development in EDM Tool

One of the latest development achievements consists in a Management module, with proper administration of marks, averages, lessons and student's presence at courses and laboratories, which allows generating, printing and exporting reports. Further, we detail all these developments and improvements.

### 4.1 e-Testing module

In the e-Testing part of the system, the Professor may use previously entered Questionnaires to generate Exams where students` knowledge will be evaluated in a manner that best suits to the testing style of the Professor. With four possible question types (Textbox, Checkbox, Ordered, Essay), almost any testing situation can be put in practice by using our system. For the first three question types, instantaneous feedback is provided to the student and the teacher, while for the Essay question type a manual input from the teacher is needed to quantify the number of points to be awarded for each answer. The exam proceeds on a limited time scale; when the specified time is over, all the students who have not yet finalized their exam, will be automatically kicked out and an evaluation of the current progress is calculated. When the exam results are complete, the teacher may draw conclusions by looking at comparative statistics between groups of students and "see" which of them have better behavior (see Figure 8). Besides that, the teacher may also look what every student has answered to individual questions, helping him to get a better picture of why for example a hard working student has got a lower mark than expected.

### 4.2 Statistics Module

The next figure illustrates the comparative performance of the students after they were tested on "Introduction in Algorithms topic". Each bar represents the average on each Group.

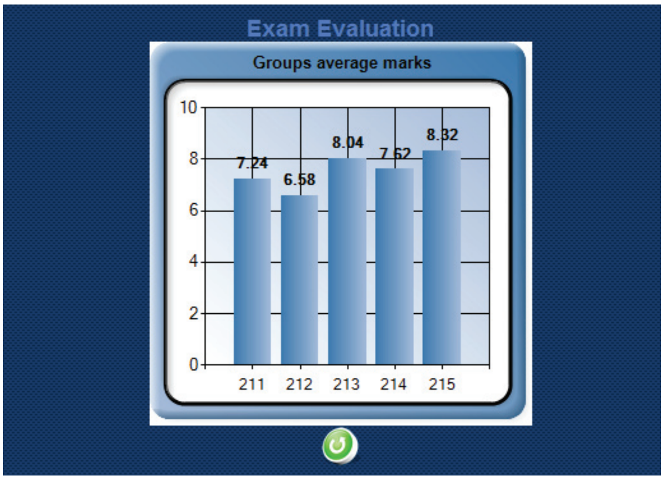
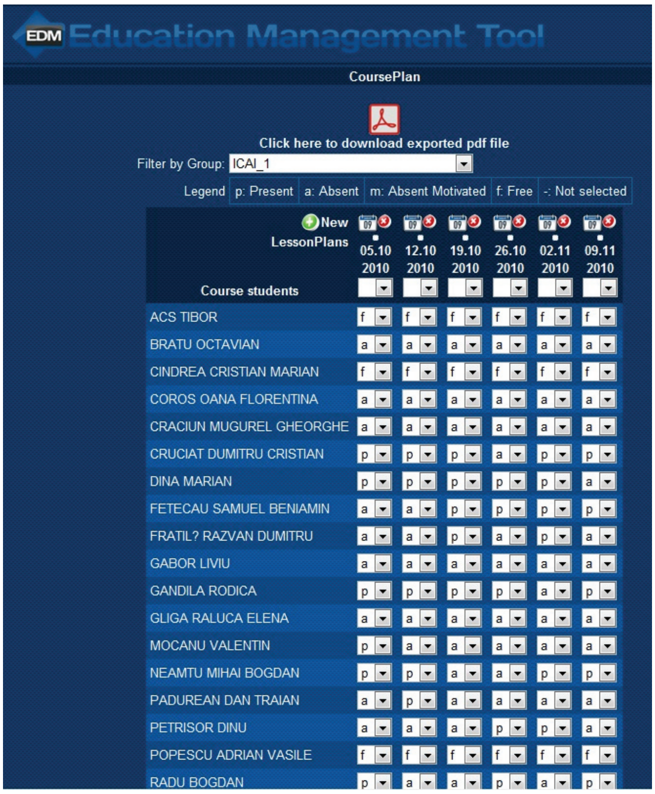


Figure 8. Statistics for exam results

### 4.3 The Presence Management Module: Collecting and Expressing Data

In a WYSIWYG (What You See Is What You Get) manner, the Professor is able to manage student attendance at his courses in an interactive manner, similar to Microsoft Excel. A table is shown, where the rows are individual students, and columns are dates of attendance. Each row-column combination contains a Presence item, which can have one or more possible states (present, absent, free, etc.).

For presence management, we have added to the latest version exporting capabilities in the management part of the system as well as allowing the professor to export *pdf* documents containing details about the student presence at course. In this manner, the professor is able to export presence in a printable manner so that he can keep track of student attendance at his courses.



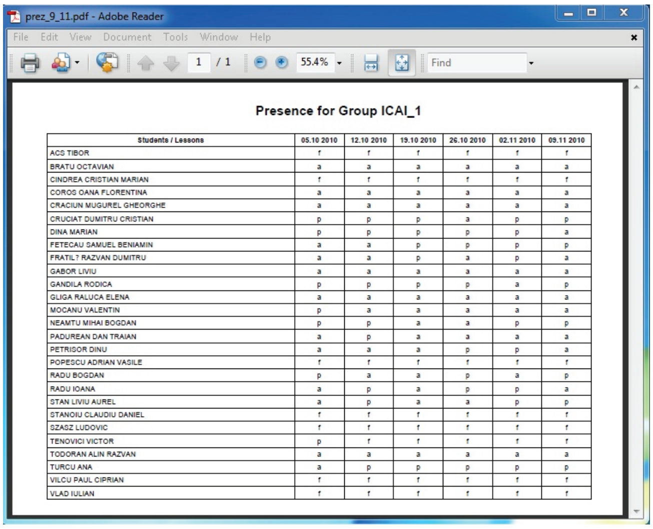
Click here to download exported pdf file

Filter by Group: ICAI_1

Legend | p: Present | a: Absent | m: Absent Motivated | f: Free | -: Not selected

| Course students | LessonPlans 05.10 2010 | 12.10 2010 | 19.10 2010 | 26.10 2010 | 02.11 2010 | 09.11 2010 |
|---|---|---|---|---|---|---|
| ACS TIBOR | f | f | f | f | f | f |
| BRATU OCTAVIAN | a | a | a | a | a | a |
| CINDREA CRISTIAN MARIAN | f | f | f | f | f | f |
| COROS OANA FLORENTINA | a | a | a | a | a | a |
| CRACIUN MUGUREL GHEORGHE | a | a | a | a | a | a |
| CRUCIAT DUMITRU CRISTIAN | p | p | p | a | p | p |
| DINA MARIAN | p | p | p | p | p | a |
| FETECAU SAMUEL BENIAMIN | a | a | p | p | p | p |
| FRATIL? RAZVAN DUMITRU | a | a | p | a | p | a |
| GABOR LIVIU | a | a | a | a | a | a |
| GANDILA RODICA | p | p | p | p | a | p |
| GLIGA RALUCA ELENA | a | a | a | a | a | a |
| MOCANU VALENTIN | p | a | a | a | a | a |
| NEAMTU MIHAI BOGDAN | p | p | a | a | p | p |
| PADUREAN DAN TRAIAN | a | p | a | a | a | a |
| PETRISOR DINU | a | a | a | p | p | a |
| POPESCU ADRIAN VASILE | f | f | f | f | f | f |
| RADU BOGDAN | p | a | a | p | a | p |

Figure 9. WYSIWYG Presence Management



Presence for Group ICAI_1

| Students / Lessons | 05.10 2010 | 12.10 2010 | 19.10 2010 | 26.10 2010 | 02.11 2010 | 09.11 2010 |
|---|---|---|---|---|---|---|
| ACS TIBOR | f | f | f | f | f | f |
| BRATU OCTAVIAN | a | a | a | a | a | a |
| CINDREA CRISTIAN MARIAN | f | f | f | f | f | f |
| COROS OANA FLORENTINA | a | a | a | a | a | a |
| CRACIUN MUGUREL GHEORGHE | a | a | a | a | a | a |
| CRUCIAT DUMITRU CRISTIAN | p | p | p | a | p | p |
| DINA MARIAN | p | p | p | p | p | a |
| FETECAU SAMUEL BENIAMIN | a | a | p | p | p | p |
| FRATIL? RAZVAN DUMITRU | a | a | p | a | p | a |
| GABOR LIVIU | a | a | a | a | a | a |
| GANDILA RODICA | p | p | p | p | a | p |
| GLIGA RALUCA ELENA | a | a | a | a | a | a |
| MOCANU VALENTIN | p | a | a | a | a | a |
| NEAMTU MIHAI BOGDAN | p | p | a | a | p | p |
| PADUREAN DAN TRAIAN | a | p | a | a | a | a |
| PETRISOR DINU | a | a | a | p | p | a |
| POPESCU ADRIAN VASILE | f | f | f | f | f | f |
| RADU BOGDAN | p | a | a | p | a | p |
| RADU IOANA | a | p | a | p | p | a |
| STAN LIVIU AUREL | a | p | a | a | p | p |
| STANOIU CLAUDIU DANIEL | f | f | f | f | f | f |
| SZASZ LUDOVIC | f | f | f | f | f | f |
| TENOVICI VICTOR | p | f | f | f | f | f |
| TODORAN ALIN RAZVAN | a | a | a | a | a | a |
| TURCU ANA | a | p | p | p | p | p |
| VILCU PAUL CIPRIAN | f | f | f | f | f | f |
| VLAD IULIAN | f | f | f | f | f | f |

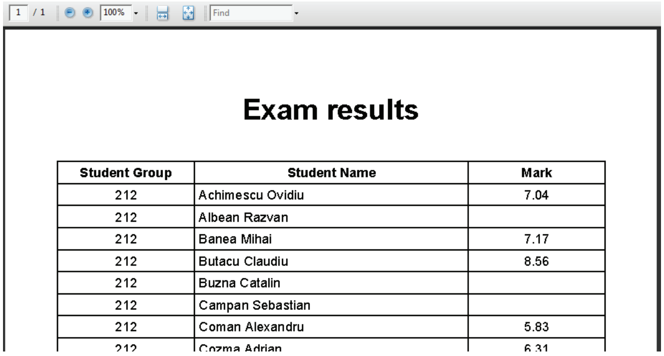
Figure 10. Presence export to *pdf* files

### 4.4 Import

The importing technique is very useful in any piece of software when users need to enter large sets of data at once. It has the advantage of not being restricted to completing web forms one at a time, but by entering data in bulk mode. In our case it is very useful for student import as an alternative to entering students one by one. The system permits import from *csv* files, which can be created and edited using any version of Microsoft Excel. The format is simple, in form of a spreadsheet, where columns represent fields and each row represents a student. The user downloads a sample *csv* file, where he/she inserts the needed students, then uploads it back to the server. The system checks the file to make sure it corresponds to the given format and if the data inside it is valid (e.g. no duplicate usernames were inserted for students). If everything is ok, the user is shown a preview and has the option of accepting the inserted data, confirming that the database may be updated with the new students.

### 4.5 Export

Exporting in a web application is as crucial as importing, because the user may need to store some of the data physically on his/her computer in files for future reference or with the purpose of printing. Because printing a web page will also print the menu, the title of the application, the log out button, etc., the teacher may only want to print the data he/she needs. This is why the system offers an "*export to pdf*" option when viewing exam results for a group of students, useful in situations where the teacher needs to take out the results from EDM and use them for external purposes (e.g. lists of marks that are handed out to the secretaries of the university after exam correction, the presence situation on each course / laboratory).



Exam results

| Student Group | Student Name | Mark |
|---|---|---|
| 212 | Achimescu Ovidiu | 7.04 |
| 212 | Albean Razvan | |
| 212 | Banea Mihai | 7.17 |
| 212 | Butacu Claudiu | 8.56 |
| 212 | Buzna Catalin | |
| 212 | Campan Sebastian | |
| 212 | Coman Alexandru | 5.83 |
| 212 | Cozma Adrian | 6.31 |

Figure 11. Exam results exported to *pdf* file

### 4.6 Notifications system

The first step in transforming the application in a real-time communication and notifications system was to introduce notifications onto each user's account. One of the notification service goals is to disseminate information. This service is based on events and events channel that separate the information senders from information receivers. Administrators have the privilege to add/edit/delete notifications, while all other user types can only visualize them. These notifications are targeted to be used for general and system messages, for informing the whole network of users who use the system. If the users need to send notifications to a subset of users, they should use the communication module. Thus, when students log into their Education Management Tool account, they will be informed about new lessons introduced by professors or other interesting, important and useful information (newsletters, news and events, conference announcements, exam schedules, etc).

## 5. E-Learning Module Centered on Digital Logic Circuits and Programming Techniques

EDM courses and third-party lessons are designed and implemented with non-expert users in mind. They are accessible, dynamic, friendly and versioned and provide a very powerful and easy to use on-line help, and demo capabilities for each of its supported circuit models or implemented algorithm. Third party flash applications specifically designed for our system cover two domains of Computer Science: Logic Circuit Design (e.g. 1-bit ALU / Full Adder, Logical Shift, Latch Delay) and Algorithms and programming techniques (e.g. Divide and Conquer and Greedy Algorithms). Such visual presentation helps the teacher to better explain and the student to easier understand how a certain algorithm or a logic circuit works, avoiding thus additional descriptive explanatory effort. Our approach is motivated since we teach digital logic circuits, algorithms and programming techniques but, more than that, we consider this topic with a high degree of difficulty, especially in high school, requiring a more in depth explanation of both theoretical notions and applications that accompany these notions. Besides that, it is free and easy to use, and helps to improve quality and performance in e-Learning, providing the following advantages: flexibility, extensibility, interactivity. Our first approach related to *Divide and Conquer* technique and *Greedy* algorithms was detailed in (Florea A., Gellert A., Anghel T. & Florea D., 2010).

The third-party lessons want to be used by students in order to assimilate the notions connected with the programming techniques, to create some new applications which are to be integrated into the e-Learning platform, and also to test the acquired knowledge. The didactic materials can be further extended, so that they cover other scientific fields too. We have sought an interactive implementation, based on visual elements in programming techniques, wanting to improve the didactic process based on blackboard and chalk only. The dynamics of the area imposes the planning of teaching approach adapted to the continuous evolution/formation of students, developing their skills to work with projects in collaborative manner.

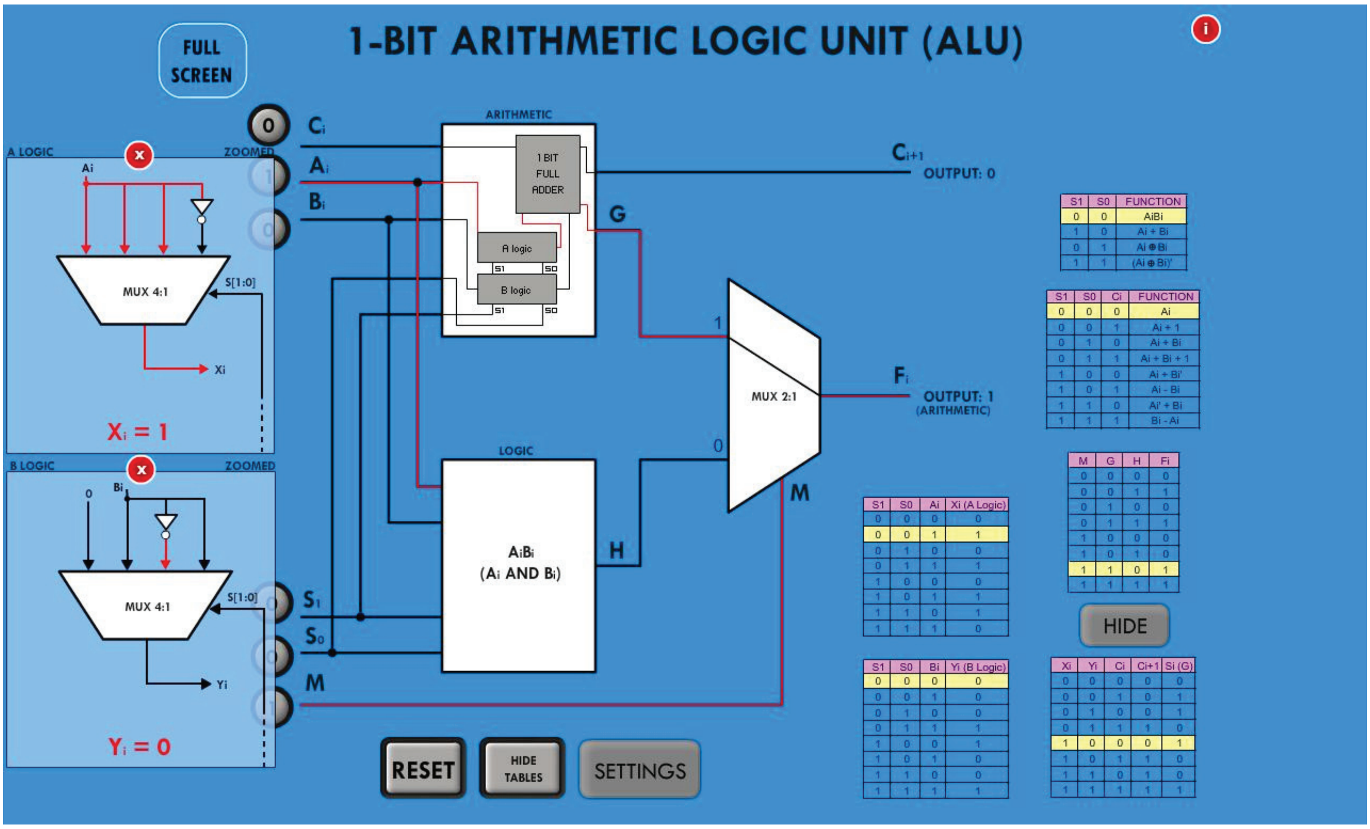
### 5.1 Digital Logic Circuits

Further, in this section we focus on the *1-Bit ALU* – an interactive lesson about digital logic circuits, a fundamental topic present in Computer Science and Information Technology, Electrical Engineering and Multimedia Systems (Warford, 2010). In order to trap learner's attention, the simulator, as shown below, was conceived to be a friendly and interactive e-Learning tool. The simulator presents a window where the user may edit the input values and start the simulation. Helped by the "*Settings*" button the user can configure the continuous "*Play*" till the end or "*Step by Step*" simulation mode. Also, after a simulation, he may "*Resets everything*" or "*Resets everything but input buttons*" remain with the previously set values. For more details the "*Show tables*" button will exhibit the truth tables of involved operations.

The concept of ALU (arithmetical and logical unit) was initially introduced by John von Neumann in his paper "*First Draft of a Report on the EDVAC*" in 1945 (Patt & Patel, 2003) and is a digital combinational circuit that performs arithmetic and logical operations. ALU represents the fundamental building block of the central processing unit (CPU) from any computing system. From didactical purposes and for a better understanding we have chosen 1-Bit ALU. As you can see there are six inputs – the data values ($A_i$ and $B_i$), the initial carry ($C_i$), the control / selection values ($M$, $S_1$ and $S_0$) and two outputs ($C_{i+1}$ – the carry out for the next rank and $F_i$ – the operation result).

Through a 2:1 MUX (Multiplexer), the M input switches between the logic ($M=0$) and arithmetic ($M=1$) operating mode. The $S_1$ and $S_0$ selection bits and $C_i$ input will choose an operation to be performed, according to the $A_i$ and $B_i$ values. If M is disabled ($M=0$), a bitwise logic operation between $A_i$ and $B_i$ will be performed (such as AND, OR, XOR and XNOR). In this case $C_i$ will not influence the output. If $M$ is enabled ($M=1$), eight different arithmetic operations will be performed (such as MOVE, INCREMENT, ADD (sum), ADC (sum with carry), SUB (subtraction: $A_i$ - $B_i$ and $B_i$ - $A_i$), SUMC (sum with 1's Complement).

The "ARITHMETIC" cell consists of three blocks: one block for the "*1-Bit Full Adder*" (Anghel, Florea & Florea, 2010), and 2 logic blocks – "A Logic" and "B Logic". The latest are based on two 4:1 multiplexers, switched by $S_1$ and $S_0$ bits that allow



| S1 | S0 | FUNCTION |
|---|---|---|
| 0 | 0 | AiBi |
| 1 | 0 | Ai + Bi |
| 0 | 1 | Ai ⊕ Bi |
| 1 | 1 | (Ai ⊕ Bi)' |

| S1 | S0 | Ci | FUNCTION |
|---|---|---|---|
| 0 | 0 | 0 | Ai |
| 0 | 0 | 1 | Ai + 1 |
| 0 | 1 | 0 | Ai + Bi |
| 0 | 1 | 1 | Ai + Bi + 1 |
| 1 | 0 | 0 | Ai + Bi' |
| 1 | 0 | 1 | Ai - Bi |
| 1 | 1 | 0 | Ai' + Bi |
| 1 | 1 | 1 | Bi - Ai |

| M | G | H | Fi |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

| S1 | S0 | Ai | Xi (A Logic) |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

| S1 | S0 | Bi | Yi (B Logic) |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

| Xi | Yi | Ci | Ci+1 | Si (G) |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

Figure 12. Screenshot of 1-Bit ALU Flash application

different combinations between $A_i$ and $B_i$ bits in order to perform the eight previously expressed arithmetic operations. The full adder's carry out is actually ALU's carry out ($C_{i+1}$) and it was previously labeled as the Output 0. The Output 1 (F) is either arithmetic cell's G output (full adder's output) if M is enabled, or logic cell's H output if M is disabled.

Behind the entire "LOGIC" module we use a 4:1 MUX which selects the operation according to the selection bits.

### 5.2 Algorithms and Programming Techniques

We also included into the lesson section interactive flash applications presenting different algorithms and programming techniques. In this work we will focus on the *Divide and Conquer* programming technique, representing an important concept in computer science. This algorithm design method breaks down a problem into two or more sub-problems of the same (or related) type which are smaller in size, easier to solve, and combines their results to give the final solution to the original problem (Cormen, Leiserson, Rivest & Stein, 2009). An important such algorithm solves the *Towers of Hanoi* problem, proposed by Edouard Lucas in 1883. It consists of three rods and a number of disks of different sizes which are initially on one of the rods in descending order. The objective is to move the disks to another rod, obeying the following rules:

1. Only one disk, the upper one, may be moved at one time from a rod onto another;
2. No disk may be placed over a smaller disk.

To move $n$ disks from A (*source*) to C *(destination)* through B (*auxiliary*):

- Move *n*-1 disks from A to B through C;
- Move one (the last) disk from A to C;
- Move n-1 disks from B to C through A.

The recursive algorithm for the classical Hanoi problem (with three rods: A the source rod and C the destination rod) is the following:

```
HANOI3 (n, A, C, B)
    if n ≥ 1 then
        HANOI3 (n-1, A, B, C)
        PRINT (A → C)
        HANOI3 (n-1, B, C, A)
```

The number of disk movements can be described by the following recurrence:

$$T_3(n) = T_3(n-1) + 1 + T_3(n-1) = 2T_3(n-1) + 1 = 2^n - 1$$

meaning that the algorithm`s complexity is exponential.

There is also a generalized version of the Hanoi problem, also called Reve`s puzzle, with four or more rods (Stockmeyer, 1994). The Frame-Stewart algorithm solving the generalized Hanoi problem, with four rods (A the source rod and D the destination rod), is the following:

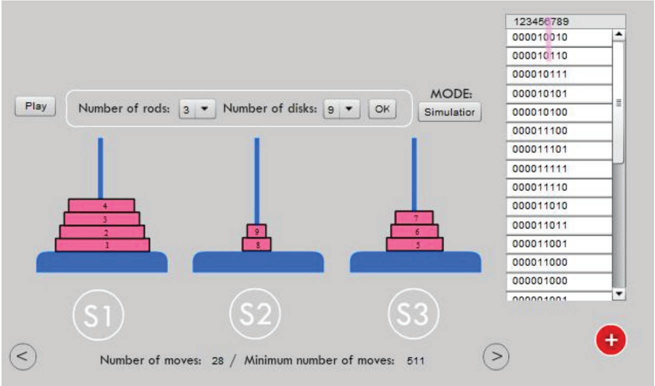
```
HANOI4 (n, A, D, B, C)
    k ← $\left\lfloor \frac{n}{2} \right\rfloor$
    HANOI3 (k, A, B, D)
    HANOI3 (n-k, A, D, C)
    HANOI3 (k, B, D, A)
```

where k can be chosen from the (1, n) interval. In (Stockmeyer, 1994) the author shows that the optimal value for k is $\sqrt{2n}$ . For simplicity in didactic explanation, we have chosen the value $\left\lfloor \frac{n}{2} \right\rfloor$ for k. Based on the complexity of HANOI3 we can determine the number of movements for HANOI4 which can be described by the following recurrence:

$$T_4(n) = T_3(k) + T_3(n-k) + T_3(k) = 2^k - 1 + 2^{n-k} - 1 + 2^k - 1$$

$$= 2^{k+1} + 2^{n-k} - 3$$

Since we have chosen k= $\left\lfloor \frac{n}{2} \right\rfloor$, we obtain $T_4(n) = 2^{\left\lfloor \frac{n}{2} \right\rfloor + 1} + 2^{n - \left\lfloor \frac{n}{2} \right\rfloor} - 3 = 2^{\left\lfloor \frac{n}{2} \right\rfloor + 1} + 2^{\left\lceil \frac{n}{2} \right\rceil} - 3$ and thus the complexity of the algorithm with four rods is still exponential.



Number of rods: 3
Number of disks: 9
Number of moves: 28 / Minimum number of moves: 511

```
123456789
000010010
000010110
000010111
000010101
000010100
000011100
000011101
000011111
000011110
000011010
000011011
000011001
000011000
000001000
000001001
```

Figure 13. Towers of Hanoi Flash application

The interactive flash application asks for the number of disks, the number of rods (3 for the classical problem or 4 in the case of the generalized problem) and to select the running mode. There are two important ways to run the application: in *user* mode or in *simulation* mode. In *user* mode the disks can be manually moved by drag and drop from one rod to another. In *simulation* mode there are other two different possibilities to run the application, *step by step* and *play*, both are using the above presented recursive algorithm to move the disks. The *step by step* running mode can be performed by clicking on the *step forward* or *step back* buttons. In *play* mode, the application runs the entire algorithm and uses a timer in order to delay the movements.

An alternative solution for the *Towers of Hanoi* problem can be provided based on the Gray codes. In the Gray system, each binary code differs from its neighbors by only one bit. If the number of bits in the Gray code is equal to the number of disks, with the least significant bit representing the smallest disk and the most significant bit the largest one, then the bit changed in each code corresponds to the disk to move. The Gray codes show which disk to move but not where to move it, however, in the most cases there is just one movement possible (due to the rules of the problem). For the smallest disk there are always two possibilities to move. If we denote the source rod with *s*, the target rod with *t* and the auxiliary rod with *a*, the correct movement (destination) of the smallest disk is provided by the following rule: s, t, a, s, t, a, etc. if the number of disks is odd and s, a, t, s, a, t, etc. when the number of disks is even. The above presented flash application illustrates also the Gray codes and highlights their connections with the corresponding disk movements.

## 6. Conclusions and Further Work

Considering that the future in education will be based mainly on the freedom of choosing an accessible way to study according to the necessities of each person, which should eliminate the physical and mental borders, the development of e-Learning platforms certainly becomes a priority of the modern education. The evolution of the digital, multimedia and communication technologies has created the possibility of using the e-Learning way in education on a large scale.

In this paper, we presented EDM tool, our visual, interactive, functional and collaborative e-Learning solution, dedicated for optimizing the whole process of teaching / learning / testing in different computer science topics. The third party lessons connected to our developed online visual application are targeted to be used by students and teachers in order to assimilate the notions connected with the digital logic circuits and programming techniques: Divide and Conquer, Greedy and the graphs theory, to create some new applications which are to be integrated into the e-Learning platform, and last, not least, to test the acquired knowledge. As a next step, we will extend the interactive teaching section with other lessons about algorithms: binary tree operations, heap operations, graph algorithms, etc. The didactic materials can be further extended, so that they cover other scientific fields too.

In future phases of the project, we will extend the system with new modules for more advanced communication (RSS feeds, personal blogs, forum, wikis, podcasts and other powerful web tools for learning). Improving the application security by approaching more sophisticated encryption algorithms such as *md5* or *sha-1* are on our wish list as well, together with more rigorous scalability testing.

## 7. Acknowledgements

We express our gratitude to Gabriel Cioflica, student at Multimedia Systems specialization, for providing his useful and competent help in the technical development of some third party lessons.

## References

[1] Adve, S. V., Hill M.D. (1990). Implementing Sequential Consistency *In:* Cache-Based Systems. The Proceedings of the 1990 International Conference on Parallel Processing, 47–50, Pennsylvania State University Press.

[2] Anghel, T., Florea A., Florea D. (2010). Improving Course Interaction and Management with Testing Assistant. *6th International Conference eLSE – eLearning and Software for Education*, Bucharest, Romania, 161-174.

[3] Bravo, C., Marcelino M.J., Gomes A., Esteves M., Mendes A.J. (2005). Integrating Educational Tools for Collaborative Computer Programming Learning. *Journal of Universal Computer Science,* 11 (9) 1505-1517, Know Center, Springer Verlag.

[4] Clarke, J. (2009). SQL Injection Attacks and Defense. *Elsevier*.

[5] Cormen T., Leiserson C., Rivest R., Stein C. (2009). Introduction to Algorithms. *MIT Press, Third Edition*.

[6] Florea A., Gellert A., Anghel T., Florea D. (2010). Online Collaborative Education Management Tool, *In*: Proceedings of the 5th International Conference on Virtual Learning (2010 - Towards a Learning and Knowledge Society - 2030), Targu Mures, Romania, p. 367-374.

[7] Guerra L., Fabbri M., Maeran G. (2008). A problematic teaching model for the implementation of e-Learning formative pathways, *In*: Proceedings of the Conference Knowledge Construction in Elearning Context: CSCL, ODL, ICT and SNA in Education, Cesena, Italy.

[8] Heiderich M., Vela Nava, E. A., Heyes, G., Lindsay, D. (2010). Web Application Obfuscation. *Elsevier.*

[9] Jonassen D. (1999). Designing constructivist learning environments, In C. M. Reigeluth (ed.), Instructional Design Theories and Models: A New Paradigm of Instructional Theory, V. 2, Lawrence Erlbaum Associates, 215–239.

[10] Moreland J., Jones A., Chambers M. (2001). Enhancing student learning in technology through enhancing teacher formative interactions. *Research Information for Teachers*, 3, Wellington: New Zealand.

[11] Patt Y., Patel S. (2003). Introduction to Computing Systems: from bits & gates to C & beyond. *McGraw-Hill Higher Education, 2nd Edition*.

[12] Scheuermann F., Pedró F. (2009). Assessing the Effects of ICT in Education: Indicators, Criteria and Benchmarks for International Comparisons, Luxembourg: Publications Office of the European Union.

[13] Shen H., Sun C. (2000). RECIPE: a prototype for Internet-based real-time collaborative programming. *2nd* International Workshop on Collaborative Editing Systems, Philadelphia, USA, 3-4.

[14] Stockmeyer P. K. (1994). Variations on the Four-Post Tower of Hanoi Puzzle. *Congressus Numerantium 102*, p. 3-12.

[15] Warford S.J. (2010). Computer Systems, Fourth Edition. Jones & Bartlett, Publishers.

[16] Woolf B.P. (2007). Building Intelligent Interactive Tutors. Student-centered strategies for revolutionizing e-learning, Morgan Kaufmann.