

Distance Learning and Supporting Tools at Babeş-Bolyai University

Florian M. Boian, Rareş F. Boian, Alexandru Vancea, Horia F. Pop

Babeş-Bolyai University, 1 Kogălniceanu St., Cluj-Napoca, Romania, {florin, rares, vancea, hfpop}@cs.ubbcluj.ro

Keywords

Distance learning, Tools for e-learning, Java platforms

1. Introduction

The city of Cluj-Napoca through its numerous universities is the top academic site in Transylvania and one of the largest in Romania. The Babeş-Bolyai University (BBU) is the largest Transylvanian university offering more than a hundred majors and having a student population of over 50,000.

In the computer science field, BBU is ranking among the top Romanian universities and is the recognized leader in Transylvania. The focus of the CS education at BBU is to produce highly skilled graduates, able to enroll in research projects, industry projects, and teaching/academic activities. While several local universities offer IT related programs such as computer engineering or economical informatics, only the BBU is certified to train students IT teaching. Besides the regular undergraduate studies, BBU also offers post-university conversion programs for gymnasium and high school professors.

In this context, the university faces the constant challenge of keeping high standards and graduating competitive students while coping with the following issues:

- The integration of Romania in the EU brought a large number of IT businesses in Cluj-Napoca, thus creating a large demand for skilled students.
- The number of students applying for IT education is larger every year
- As a side effect of the IT business growth in Cluj-Napoca, the students are hired well before their graduation, thus lowering their focus on studies.
- As the overall education system in Romania is growing, there is a strong need for qualified gymnasium and high-school IT teachers

The CS department adopted distance learning as part of the solution to the issues above. Distance learning and e-learning have a longer history in social and humanist education at BBU, the CS department adopted it only later and initially for IT specialization of gymnasium and high-school teachers.

The paper presents the BBU distance learning experiences in the IT field, along with the challenges, solutions, tools, and lessons learned during the process.

2. First Distance Learning Experience

The BBU CS department started distance learning in 1998 with the support of a TEMPUS grant. The program offered IT training to high-school professors of various specialisations. After graduating the program the high-school professors were certified to teach IT courses. The program spun over four semesters and was offered in six series, across eight years.

Each semester the program offered four courses, evaluated at the end through a written examination. The reference materials for the courses were created by the BBU professors, printed, and distributed to the students at the first meeting in the semester. Because the students involved in the program were people with full-time teaching jobs in Cluj-Napoca and other towns, during each semester the instructors scheduled a number of four meetings on BBU premises.

Outside these meetings, the communication between students and the BBU instructors was done through a website where the instructors posted the assignments to be solved, and through a mailing list. The solutions to the assignments were handed in and presented by the students during the last meeting before the exam.

3. Conversion Program for Gymnasium Teachers in Rural Areas

A newer CS distance learning program at BBU is targeting gymnasium professors from rural areas. Funded by a World Bank grant, the program offers the opportunity to specialize in IT post-graduation and get certified to teach gymnasium-level courses. While the initial distance learning program required the students to pay for their studies, in the current program, the World Bank also offers financial assistance to students that qualify. The program is currently offered to a second series of students.

The program is structured across three semesters, each consisting of four courses. Three of these courses require lab work under the supervision of an instructor. At the end of the last semester, the students must write a diploma thesis and pass a final examination. Due to its wide popularity, the program is organized across four centers located in cities that cover a much wider geographical area. The students can complete their lab work at any of these centers under the supervision of a local lab instructor. Twice or thrice a semester all course instructors travel from BBU to each of these locations to meet the students personally and answer their questions.

Besides the meetings enumerated above, the students and professors communicated through a website for general announcements and through a web portal developed specifically for this purpose. The web portal will be described in the upcoming sections.

4. Managing Undergraduate Courses

The regular undergraduate CS courses have been increasingly difficult due to several reasons discussed in detail in the following.

- The larger number of students together with the very practical aspect of CS course activities makes it difficult for the professor to keep track of each student's activity. Although this is a matter of management, automating it will allow the professor to focus fully on teaching.
- Due to the large number of IT companies opening business in Cluj-Napoca, a large percentage of the students have part-time or even full-time jobs before graduation. Consequently, their presence at school is much reduced which leads to a lack of communication with the professors, resulting eventually in weaker learning results or even dropouts.
- As an effect of the Bologna process, the material taught to students in four years was restructured to span only three years. In order for students to assimilate such quantity of information, the professor must keep them involved continuously. Hence, students are given homework, quizzes, and weekly or bi-weekly projects. Thus, the final grade of the course is calculated based on approximately 20 to 25 scores. Keeping track of all these scores while keeping the students informed required a specialized application.
- The grading methodology in Romania provides that grades be assigned between 1 and 10 based solely on the student's individual performance, as opposed to assigning

grades according to the distribution of individual scores. This policy does not promote a competitive spirit among the students taking the same course. Often times, weaker students try to cheat by presenting a colleague's solution as their own. To prevent and reduce the occurrence of such situations, as well as to push the students to do their own work, it is necessary to create for each assigned project numerous problems (similar in scope and difficulty) so that there will be no project collisions inside lab groups. Keeping track of the assignments became a burden that had to be automated.

These reasons along with the distance learning programs lead to the development of a web portal to assist professors in providing high quality education.

5. The AMS System

The Assignment Management System (AMS) has been developed to ease the management of CS courses. It has been designed to match the structures and flows of the Romanian educational process and to solve the issues presented above. The system has been developed as a web portal using J2EE technologies and the MySQL database server [1, 2].

5.1 Data Organization

The central data entity of the AMS system is the *Course* (fig. 1). A *Course* is the realization of one real life course during one semester. Each *Course* has assigned one *Professor* who teaches the lectures. For lab work, the students attending a course are divided into *Lab Groups* each of which is assigned a *Professor* (lab instructor). The lab work takes place during *Lab Hours* for which attendance is mandatory. The lab work is divided into *Projects*, each *Course* having several of them. Each *Assignment* associates a *Problem* with a *Student*. The grades are assigned to *Assignments*.

As mentioned above, each *Project* contains a pool of *Problems* assigned randomly to the students of a *Lab Group* in order to minimize overlaps. This is meant to push the students to try to solve their project by themselves, rather than duplicating a colleague's work. While most projects have problems assigned in the AMS system, a few are only placeholders. Such projects store the grades of partial or final exams.

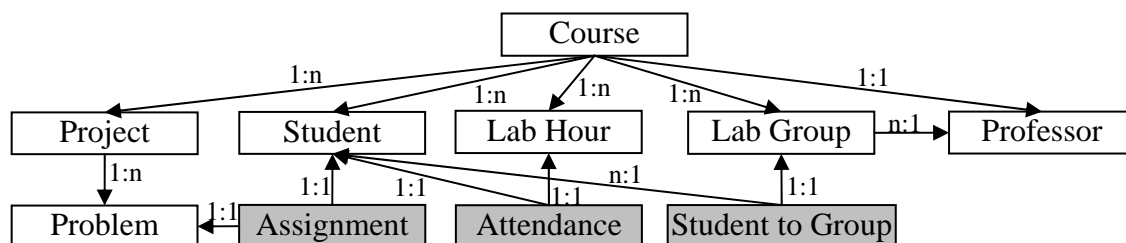


Figure 1. Data entities and relationships

5.2 User Interface

The AMS system has a different interface for each user role: one for students, one for professors, and one for the administrator.

Administrator Interface

The administrator interface is used to load the database with data such as new students, new courses, new professors, relations between students, courses, and professors, etc. This interface is used mostly at the beginning of each semester and then very rarely during the semester. Consequently, the effort to develop this interface was minimal, the aim being to bring it to a stage where it had all the needed functionality, with a minimum of effort. As of now, this interface is command line-based and consists of a series of files with simple formats that are used to modify the database. At the beginning of the semester these files

are populated with all the new data and used to insert it, while during the semester they are used to update existing data.

Student Interface

The student interface (fig. 2) presents at the top a drop-down listing the courses in which the student is enrolled. Upon selecting a course, the student is displayed a summary of the course projects.

The screenshot shows the AMS Student Interface. At the top, there's a 'Logout' button and a dropdown menu for 'Distributed Operating Systems' with a 'Select curs' button. Below this, the course name 'DOS : Distributed Operating Systems' and the semester '921-1' are displayed, along with an 'Intreaba profesorul' button. The main part of the interface is a table with the following columns: 'Proiect', 'Termen limita', 'Depune arhiva', 'Copie arhiva', 'Nota', and 'Prezenta'.

Proiect	Termen limita	Depune arhiva	Copie arhiva	Nota	Prezenta
cygwin install; X use	2006-10-06 (2006-10-20)	Depune			P
Vmware; Unix install	2006-10-13 (2006-10-27)	Depune			P
Unix signals	2006-10-20 (2006-11-03)	Depune	Extrage	10	P
Message queues	2006-10-27 (2006-11-10)	Depune	Extrage	10	P
Semaphores and shared memory	2006-11-03 (2006-11-17)	Depune	Extrage	10	P
POSIX threads	2006-11-10 (2006-11-24)	Depune	Extrage	10	P
Solaris threads	2006-11-17 (2006-12-01)	Depune	Extrage	10	P
CGI over web	2006-11-24 (2006-12-08)	Depune	Extrage	10	P
Client - server with socket	2006-12-01 (2006-12-15)	Depune	Extrage		P
Socket with standard	2006-12-08 (2006-12-22)	Depune	Extrage		P

Figure 2. Student interface

For each project, the student can see the deadline, and the grace deadline (a later date up until when the project can be handed-in with a penalty). The project entries also display the grade and links to the requirements (fig. 3(a)), solution upload page (fig. 3(b)), and solution downloading. On the right hand-side, the portal displays the list of lab hours and the student's attendance. The dates of the lab hours are visible in a tool tip.

This screenshot shows the 'Project requirements' page for the course 'SO2 : Sisteme de operare (2)'. It includes a 'Logout' button and a dropdown menu for 'Sisteme de operare (2)' with a 'Select curs' button. The page title is 'semMemP : Semafoare si memorie partajata'. Below this, there's a section for 'Prezentarea proiectului' with the text: 'IPC: Memorie partajata'. Another section, 'Prezentarea problemei', contains a detailed description of shared memory: 'Memoria Partajata este un mecanism eficient de a trimite date intre procese. Un program va crea o portiune de memorie pe care alte procese o pot accesa daca este permis.' There's a 'La precedent' button at the bottom.

(a)

This screenshot shows the 'Solution upload' page for the course 'SO2 : Sisteme de operare (2)'. It includes a 'Logout' button and a dropdown menu for 'Sisteme de operare (2)' with a 'Select curs' button. The page title is 'cozi : Cozi de mesaje'. Below this, there's a section for 'Alege arhiva de depus:' with a 'Browse...' button and a 'Depune arhiva' button. A table lists the uploaded files:

Arhiva	Numele initial	Data depunerii	Lungimea
SO2_cozi_bs29654_2006-11-04_01-08-33_laborator3.zip	laborator3.zip	2006-11-04	3104

There's a 'La precedent' button at the bottom.

(b)

Figure 3. Student interface (a) Project requirements page (b) Solution upload page

The project requirements page shows the student two text boxes: one on the top containing general guidelines and information needed for the project, and one on the bottom showing the problem assigned to them. The solution handing page allows the student to upload through a web browser a ZIP archive containing all the files relevant to their solution. Upon uploading, the server records the date and time of the upload, which would be later used by the grader to decide whether to apply a delay penalty to the grade.

At all times, the student is displayed an "Ask your professor" button in the top-right corner of the web-page. This button takes the student to a page from where he/she can send email to the lab instructor and course instructor. This feature improves the student-professor communication significantly, as they can address questions immediately upon reading the assignments. This feature proved very useful during the conversion program for gymnasium teachers in rural areas, as these students were often just starting to use a computer, and felt very comfortable sending questions whenever they felt necessary without having to look for the email address of the instructor or logging into their on-line email accounts.

Professor Interface

The professor interface is accessible to course and lab instructors. The portal displays the same course selection drop-down as it does in the student interface. A second drop-down allows the professor to select the lab group to be displayed.

The functioning of the course selector drop-down is determined by the three radio buttons on its left. Using them, the professor can choose to view one of the following:

1. The project assignments, submitted solutions for grading (fig. 4(a))
2. The lab hour attendance (fig. 4(b))
3. Download an Excel file containing all the information about assignments, grades, and attendance of each student in the selected lab group.
4. Download an archive with all the problems assign for all the projects in the selected course

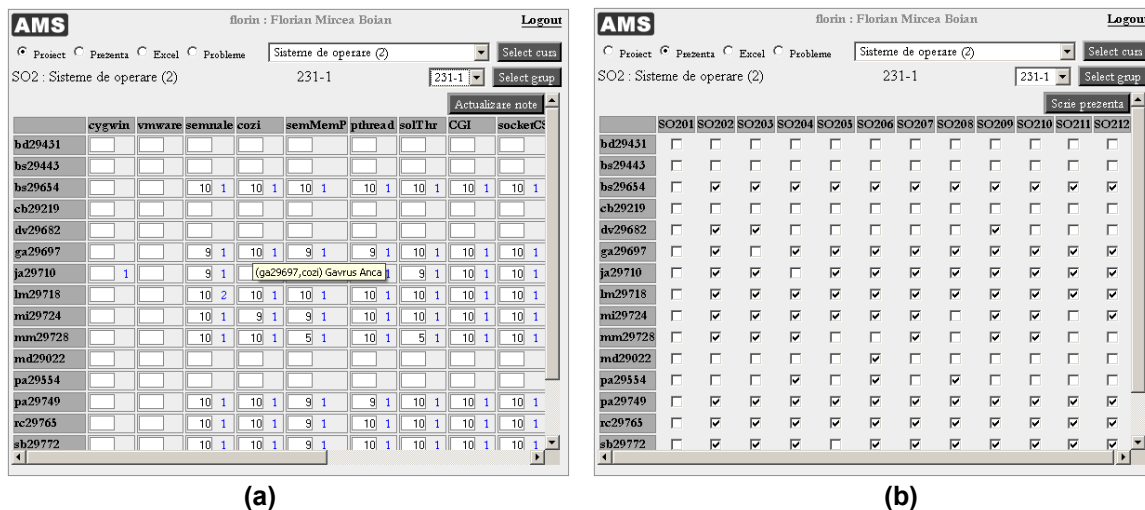


Figure 4. Professor interface (a) Project grading page (b) Attendance page

The projects page displays a table displaying on each row information about a student, and on each column information about a project. Each cell contains an input field where the professor can assign the grade. Next to the grade there is a number saying how many submissions the student made for that specific project. Clicking on this number will download an archive of all the submissions. Although not visible in the figure above, the page also contains links, through which the professor can perform on the following actions:

1. Send email to a specific student or to all the students in the lab group
2. Download all the solutions submitted by

- a. One student for all projects
- b. All students for one project
- c. All students for all projects

The attendance page displays a table similar to that of the projects page except that on columns it has the lab hours. Each cell contains a check-box which the professor can check if the student is present at that specific lab hour.

5.3 Authentication and Access Control

The AMS system uses a mixed approach to authenticating the users. This approach is designed to accommodate the following requirements raised by the location of its users.

1. For users with an account in the department's system, use the department's authentication server
2. For external users use the default AMS authentication system

The first of these requirements is necessary to simplify the student's access, by eliminating the need to remember yet another username and password. Depending on the department's network setup, the AMS system can be integrated with a Kerberos server for authentication, or if such a server is not available, it can use the department's FTP server. With this approach, students can log into the AMS portal using they existing UNIX/Windows accounts.

For external users such as the gymnasium professors in the distance learning projects, the approach above is not possible as they not given accounts in the department network. For this type of users, the AMS system uses its own authentication system relying on passwords stored in the database.

Access Control

The AMS system defines three roles, each of which results in displaying a specific interface. As already presented, these three roles are *Administrator*, *Professor*, and *Student*.

The *administrator* role is assigned to a single user created by default at system installation. The *professor* and *student* roles are assigned to users dynamically using the administrator interface. As expected, the role of a user rarely changes if ever.

A second level of access control is implemented for the professors. The course's lead professor has access to all the students and can assigned or change grades and attendance to any of the students enrolled in the course. However, the lab instructors have access limited to the lab groups assigned to them. There is one exception to this rule, in that a lab instructor can record the attendance of a student that does not belong to his lab groups. This has been implemented to solve frequent situations where students attend lab hours with other lab groups to make up for a missed lab hour. This exception does not extend to grade assignment, a student being required to present his solutions to his lab instructor.

5.4 Problem assignment

To push the student to try to solve the assignments by themselves rather than replicating a colleague's work, each project is assigned a pool of approximately fifteen problems on the same subject and with similar levels of difficulty. The aim is to assigned different problems to all the students in a lab group. The system does this automatically by cycling through the students of each lab group and choosing at random a problem from the pool. If there are more students in the lab group than problems in the pool, the system chooses at random the problems to be duplicated.

Normally, all the projects and their problem pools are prepared in advance, before the semester start, thus when the semester starts all students have their problems already assigned. There are rare occasions when students enroll late in the course, and in such cases, the system assigns them problems automatically, trying to keep the unique problem assignments within the same lab group.

5.5 Final Grade Calculation

The final grade calculation is usually a simple weighed average of all the score received for the semester assignments and the exams. Usually, the projects are all averaged in a cumulative grade, which then weighs a lower percentage of the final grade, while the partial, practical, and/or final exams will have higher weighs. However, to insure a fair evaluation the algorithm can get more complex.

Re-examination Effect on Final Grades

A first aspect to be considered is the re-examination period specific to the Romanian education system. Students that do not attend or fail the exams are given a second chance later to re-take the examination. While for students attending the regular exams the project grades are evidently important for the final grade, students taking the exam during the re-examination period cannot be given this advantage as they had more time to implement the projects. Thus, the weight of the projects needs to be distributed across the scores achieved during re-examination.

Attendance Effect on Final Grades

Attendance is another aspect that can affect the final grade. As lab hours are mandatory, students not having a minimum number of attended labs are not accepted in the final exam. Such students are automatically pushed in the re-examination period and their grade calculated differently.

Conditions on Project and Examination Scores

The final grade calculation can also be affected by lower limits imposed on the scores achieved during the semester. For instance, a student with a project average of less than 5 may automatically fail the class.

All these aspects make the final grade calculation difficult, especially when it involves tens of scores, as is the case with the Operating Systems courses. To simplify this, the AMS portal currently provides the possibility to download an Excel file with all the scores and attendance information necessary. Using this spread-sheet, the professor can implement the desired calculation algorithm.

A recent feature to be tested during the upcoming semester allows the professor to implement the grade calculation in JavaScript. The written algorithm is executed using the recent Java 6 support for embedded languages. This feature is not final as it restricts the professor-level user pool to people who know JavaScript. A further development is currently underway, try to provide a visual interface to setting the grade calculation algorithm using hierarchical grades.

5.6 Solution Evaluation

The evaluation of a student's solution to a project is done in person, the student having to present his work to the lab instructor. In addition to this approach, for cases where the time in class is insufficient, the professor can download the student's solutions and evaluate them before class, thus shortening the in-person interaction.

The solution evaluation is also being re-worked as to provide the professor with the possibility of viewing the student's solution in the browser rather than having to download a ZIP archive. Further more, annotation features are added to the system to allow the students and professors to make comment or raise questions on specific portions of the solution.

5.7 Detection of Plagiarism

Given the students' tendency to copy solutions from each other, it is necessary to automate the detection of plagiarism. Currently, this is not implemented and professors rely only on direct examination of each student's solution and the student's ability to answer questions about their implementation. While this approach is efficient, it causes delays and extra work,

and there are frequent situations where the cheater presents the solution before the original author, leading to discussions between the instructor and students on who the real author is. An automated module is being developed to automatically detect similarities between solutions and warn the professor as well as the student about them. In order to be efficient, the detection must provide the following features:

1. Detect code similarities even though comments and variable names are different
2. Detect code similarities regardless of the code blocks order (for instance the order of function definitions in a C program may make the code appear completely different to an unaware comparator)
3. Allow similar blocks or files to be ignored, thus eliminating the false positives caused by code provided by the instructor to help the students.
4. Support several programming languages, and detect them by more than just file extensions.
5. Support other formats than text and source code files, as assignments may require the student to write a report, which would most likely be an MSWord or PDF document.

6. AMS System Production Statistics

The AMS system has been in usage for the four semesters, supporting the university internal courses as well as the distance learning courses. Table 1 presents the system's usage statistics since it has been in production.

Table 1. AMS System production statistics

	Fall 2005	Spring 2006	Fall 2006	Spring 2007
Local Students	0	676	967	959
Remote Students	93	103	103	125
Professors	10	9	29	30
Courses	4	7	22	22
Lab Groups	4	52	109	109
Projects	16	87	138	138

During these four semesters the AMS system has handled more than 7000 hand-ins amounting to 1.7 GB of data.

7. Conclusions

Faced with the challenges of a growing educational system and of a rapidly growing business landscape, the BBU CS department has implemented changes aimed to provide high quality education. Course requirement adaptations, distance learning, and tools to support it are at the core of these changes. These new methods have been successful but are constantly monitored and adapted to the needs of the market. [3, 4]

In the context of Romania's integration in EU, dramatic changes are envisioned in the way the educational system is implemented in rural areas. The distance learning support offered by our AMS system proves though to be of a particular and significant importance for the gymnasium teachers in rural areas experiencing the IT post-graduation conversion program.

References

- 1 Boian F, Boian R, Tehnologii fundamentale Java pentru aplicații web. Ed. Albastră, Cluj, 2004
- 2 Goodwill J. B. Mastering Jakarta Struts, JSP and EJB. J. Willey. 2002
- 3 Garcia-Beltran A, Martinez R, Jaen J A, B-Learning with Engineering Students in Computer Science Courses, Proc. Conf. Informatics Education Europe, Montpellier, Nov. 9-10, 2006
- 4 Hermann C, Hurst W, Welte M., The eLecture Portal: An Advanced Archive for Lecture Recordings, Proc. Conf. Informatics Education Europe, Montpellier, Nov. 9-10, 2006