

ARCHITECTURE OF A COMPUTER BASED INSTRUCTIONAL SYSTEM

**Emilia PECHEANU, Diana STEFANESCU,
Luminita DUMITRIU, Adrian ISTRATE**

*University "Dunarea de Jos" of Galati, Department of Computer Science and
Engineering, str. Domneasca 47, Galati 6200, Romania, fax: (40) 36461353
e-mail: emilia.pecheanu@ugal.ro, diana.stefanescu@ugal.ro,
luminita.dumitriu@ugal.ro, adrian.istrate@ugal.ro*

Abstract: The paper describes the architecture of a tutorial system that can be used at various engineering graduate and postgraduate courses. The tutorial is using Internet-style WWW services to provide access to the teaching information and the evaluation exercises maintained with a RDMS. The tutorial will consist of server-side applications that process and present teaching material and assessing exercises to the student using the well-known Web interface. All information in the system will be stored in a relational database. By closely sticking to the ANSI SQL specifications, the system can take advantage of a free database managing system running on Linux, the mini-SQL. The tutorial can be used to on-line deliver any courses, creating new, continuing education opportunities. Taking advantage of the modern deployment techniques, the instructional/assessing tutorial offer high degrees of accessibility.

Keywords: Tutoring System, Assessment Component, Instructional Environment, Educational Server, RDBMS, miniSQL, Linux.

INTRODUCTION

For a school of engineers, an important objective is to find new methods of teaching in order to improve the technical expertise of graduating students. A solution is to provide distance learning and to make technical education available on the Intranet or Internet. The authors of this paperwork describe the project of an instructional system using both modern communication technologies and teaching methods. The designed instructional system is a tutorial that it will be used to teach the curricula disciplines of the Computer Science field of study. (Bumbaru, 1995).

The tutorial will be a collection of related lessons consisting of theoretical parts (the instructional environment) that can include different simulations (animated images, Java applets) and a collection of testing exercises (the testing and assessing environment). Lessons and exercises will be presented to the user as a sequence of Web pages. The Web pages forming lessons can include text, examples, images/animated images, diagrams and links to related information. The applicative (testing) environment will permit the assessment of students' knowledge and can turn into a tool to enhance learning.

CONTENTS OF THE TUTORIAL SYSTEM

The Tutorial System conceived to consist of three components that accomplish the following tasks (Ibrahim, 1995):

- **INS-ENV** (instructional environment built upon a relational database system) – that will allow interactive multi-user remote access to the lessons of various disciplines;
- **ASS-ENV** (interactive testing and self-assessing environment) – that will allow multi-user remote access to packages of interactive question sets to be answered by students in order to self-asses their domain knowledge of any discipline;
- **CONT-ED** (contents editor) that will be an interactive environment used to design and modify the appearance and contents of the available lessons and assessment packages.

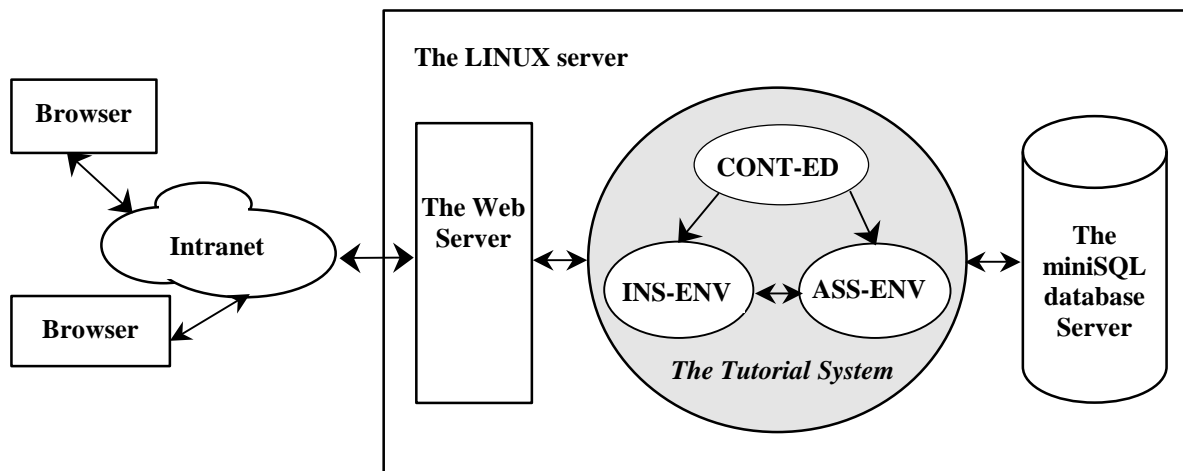


Fig.1. Architecture of the Tutorial System

1.1. The Instructional Environment

The INS-ENV environment will be an educational tool meant for the students that learn disciplines belonging to Computer Science field of study. It can be also used by anyone wishing to improve his knowledge on the subjects presented, since it is accessible to any user of the University LAN.

The INS-ENV environment is designed to consist of a sequence of lessons referring to operating systems basics, displayed as texts, images, animated images, charts, applets. Each lesson will have links to related concepts presented in other lessons. Students' access to INS-ENV lessons will be allowed through a standard hypertext browser (Internet Explorer, Netscape) connected via Intranet to the Web server providing the instructional information.(Krol, 1992)

Each accessed lesson will be dynamically generated upon request from the user, as an HTML page built on the basis of information stored in a relational database system. User's connection to the lesson systems will demand authentication. INS-ENV will communicate with the interactive testing/assessing environment ASS-ENV to allow the students to perform the self-assessment of their acquired knowledge on the subjects presented by the lessons.

The INS-ENV environment will have two main modules: the NAVIGATOR and the ANALYZER.

The NAVIGATOR module is specified to provide the user with the following features:

1. to display of the educational contents of the lessons; the creation and the presentation of the lessons will be made upon user's request;
2. to provide the student with predefined paths for guided browsing of the lessons' system offered by the tutorial;
3. to allow random lesson browsing in an unlimited number of instructional sessions, storing each session's trace;
4. to host calls of the testing/assessing environment ASS-ENV;
5. to store in the database tables the navigation path followed by each user during every instructional session.

The **ANALYZER** module is specified to provide statistics on system's users, on instructional sessions properties, on lessons accesses and on different navigation paths followed by the students or users. All information displayed by INS-ENV component: lessons (texts, images, applets, links), as well as user's browsing paths will be stored in the tables of a relational database system.

The database tables used by the INS-ENV system are presented in figure 2 as entity-relationship diagram.

As shown in figure 2, one can notice the following:

- a chapter consists of several lessons;
- a lesson consists of several sections;
- a section consists of several components that have a certain type, such as text, image, title, subtitle, applet;
- a section component can belong to several sections, from several lessons;
- a student can perform several instructional sessions;
- an instructional session has several stages, each stage corresponding to a lesson;
- the same lesson can be associates with several stages of the same session, or different sessions.

Students' accesses of the instructional environment is designed to be counted as follows:

- in Stage table there is a record for every access at a lesson accessed during an instructional session along with the time period of the access (a lesson can be accessed several times during a session, in different stages);
- in Session file there is a record for every lesson accessed during the instructional session, along with the number of lesson accesses per session and the total amount of time associated with the lesson.

Based on the contents of Stage file, one can obtain the browsing path of the user in the lessons tree, during a certain instructional session. Phase and Session tables will be used in building the statistics on student use of the lessons system.

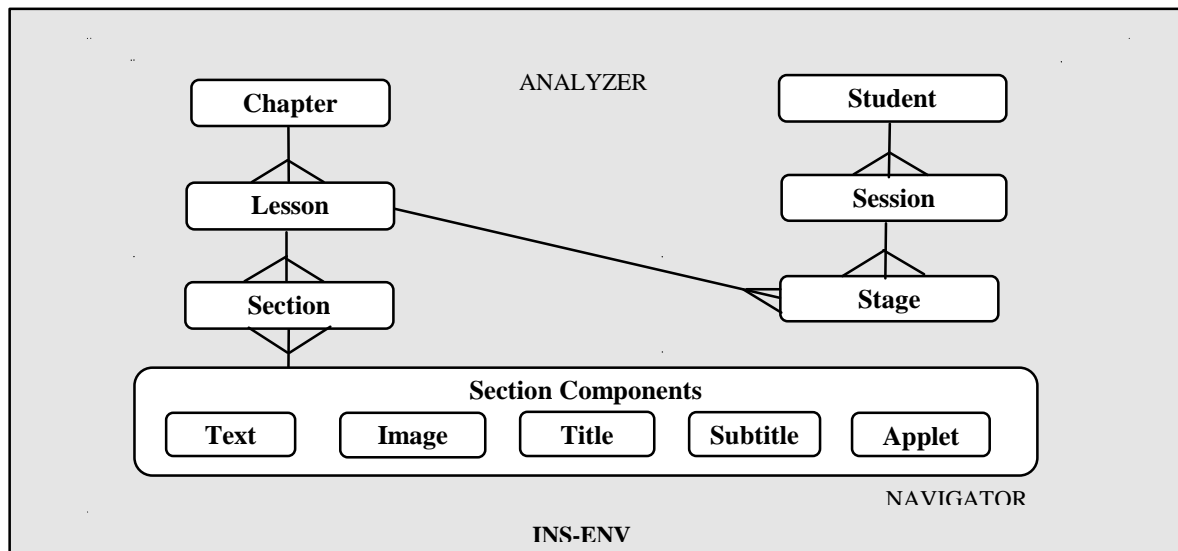


Fig.2. The entity-relationship diagram for the data structures of the INS-ENV component.

1.2. The Assessing Environment

The ASS-ENV component is designed as an interactive tool that allows students, by multi-user remote access (via Intranet), the self-assessment and examination of their knowledge on a discipline. Knowledge assessment will be performed through the presentation of packages of multiple choice question sets. By answering the question set, the student can appreciate the quality of his acquired knowledge on a specific subject.

Students access to the question sets will be interactive through a standard HTML browser (Internet Explorer, Netscape); the question sets will be dynamically generated as HTML pages, and are sent on request from Web server via Internet/Intranet. The contents of question sets, as well as students answer choices, will be stored in the relational data base tables. ASS-ENV

also will allow teachers adding or updating the question sets, through an interactive remote access based on the same Web technology.

ASS-ENV subsystem consists of three main modules: EVAL, QUEST, REPORT. The three components will work on the same data, stored in the database tables, providing the main features of the ASS-ENV self-assessment and testing environment (figure 3).

The EVAL module is designed to provide the following features:

1. students access to the question sets of the system; the access will require authentication;
2. students knowledge self-assessment/examination through test sessions; during a test session the student is presented HTML pages with multiple choice questions, where he has to select the correct answer(s);

3. each testing session will present several questions, randomly selected from the entire set of questions. EVAL will allow a student to perform an unlimited number of assessment sessions, keeping track of each one of them.

The **QUEST** module is designed to provide the following features:

1. teachers access to the question sets of the system; the access requires authentication;
2. interactive adding and updating of questions and answers;

3. teachers can grant students the access to the correct answers of the questions in the set.

The **REPORT** module is designed to accomplish the following tasks:

1. will provide, on request, statistics of the activity of a certain student or group of students that have performed test sessions;
2. will provide, on teacher's request, statistics on the manner in which certain questions or groups of questions have been answered;
3. will provide, on request, the correct answers, if students access was granted by the teacher.

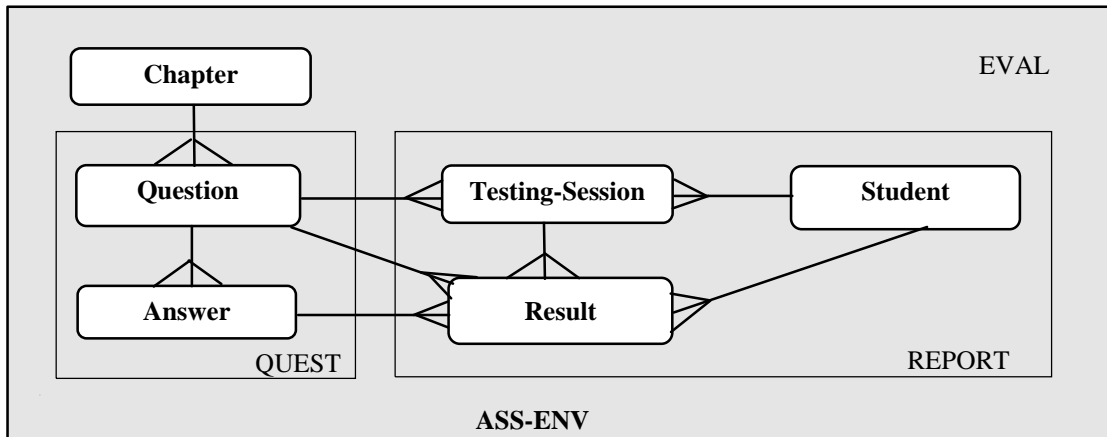


Fig.3. The entity-relationship diagram for the data structures of ASS-ENV.

As shown in figure 3, one can notice that:

- a chapter has several associated questions;
- a question has several associated answers (correct and incorrect answers);
- a student has several associated test sessions; a question can be shown in several test sessions.

The Results table contains all answers for the choices of all questions, for all testing sessions, for all students. During a testing session a clock for measuring time will enable the student to know the time remaining for completion the test. At the end of a testing session an intermediate assessment of student knowledge will be made.

The score S is calculated considering the correct answers and the difficulty level of the test achieved. The score S_i for a choice of a question is established using the following rules (Thomas, 1996):

1. correct answer: $S_i = 1$;
2. incorrect answer: $S_i = 0$;
3. no answer for a correct choice: $S_i = 0$;
4. no answer for an incorrect choice: $S_i = 1$.

If we consider n as the number of choices of a question, the score S_q for a question is calculated with formula (1). If we consider $n1, n2, n3$ the total number of questions corresponding to a particularly

difficulty level, the total score S is calculated with formula (2).

$$S_q = \frac{\sum_{i=1}^n S_i}{n} \quad (1)$$

$$S = \frac{\sum_{i=1}^{n1} S_{q_i}}{n1} + 2 * \frac{\sum_{i=1}^{n2} S_{q_i}}{n2} + 3 * \frac{\sum_{i=1}^{n3} S_{q_i}}{n3} \quad (2)$$

The final assessment of student knowledge can be established by the teacher as special testing and examination session.

1.3. INS-ENV and ASS-ENV implementation

INS-ENV and ASS-ENV will be implemented on an IBM PC, running the Linux operating system. The HTML pages showing lessons and question sets will be dynamically built on user's request, request intercepted and treated by the Web server running on the computer. The information included in the

dynamic pages can be obtained as a result of a query to the relational data base system miniSQL.

All information needed by INS-ENV, namely texts, images, links, charts, sessions histories, or ASS-ENV, namely questions, answers, students choices, will be stored and maintained in the files of the relational database server miniSQL. The structures of these files can be implemented with the creation table tool, available on the miniSQL server.

The lessons and the interactive questionnaires will include Java Scripts that give emphasis to the visual presentation and help to build the predefined paths. The connection between the Web server and the miniSQL database server is to be achieved through CGI scripts of the two modules: the instructional module, INS-ENV, and the assessing module, ASS-ENV. The two modules, INS-ENV and ASS-ENV are designed to be a collections of C modules, using the miniSQL data base server's API, as well as the communication specific CGI calls of the WWW server on Linux. (Speh, 1992)

1.4. The CONT-ED environment

The CONT-ED environment will be integrated in the tutorial system, along with the INS-ENV and ASS-ENV environments. CONT-ED is conceived to assist the interactive design of the educational information used by INS-ENV and ASS-ENV, as well as its storing in a structured format (database tables).

The CONT-ED application is specified to provide the accomplishment of two main tasks:

1. as an interactive environment, to assist the teacher in conceiving and realizing both the appearance and the contents of educational material to be presented to the user (student), namely lessons, multiple-choice questions;
2. to convert and store the educational contents along with appearance properties in the database files, in order to make them available to INS-ENV and ASS-ENV environments.

Using the CONT-ED module the teacher will be able to model the logical structure of the instructional material at his wish, and to choose the type of its visual presentation (layout, shape and disposition of push-buttons, colors, font size, etc.) from the templates offered by CONT-ED.

CONT-ED's implementation will be as Java programs on PCs, on Windows platform (Win'95, Win98 or Win NT). Considering the platform

independence of Java language and the portability of AWT package classes, we will be able to transport it, with minimal changes or no change at all, on other platform, namely Unix.

CONCLUSIONS

While designing INS-ENV and ASS-ENV, authors used an object approach, meaning that each database table that is unique and consistent is treated as a class. Each record in a database table is an object (instantiation of the class associated with the data base table in question), which is uniquely identified. This approach offered a higher productivity in the design and implementation processes of the two environments, INS-ENV and ASS-ENV.

The choice for the miniSQL relational database system as the tool for structured data storage led to a high performance management of the instructional material. The components of Tutoring System, INS-ENV, ASS-ENV and CONT-ED, will be separately developed, so they can be run on different platforms, and via Internet/Intranet they can still function together as an item.

The Tutorial System will join the information spread technology WWW-specific with the relational data base systems capabilities in storing and managing structured data. This can lead to multiple advantages in exploiting, maintaining and updating the computer-aided instructional system, offering a higher degree of generality approach in implementing and using such a system.

REFERENCES

- Bumbaru, S., Stinga, O., Pecheanu, E., Dumitriu, L., Tudorie C. (1996) A CBTS built upon Oracle RDBMS. *Proceedings of The 3-rd East-West Congress on Engineering Education*, vol. 2, pp. 378-384, Monash University Press, Poland
- Ibrahim, B., et al, (1995) Advanced Educational Uses of the WWW, *Proceedings of ICWWW '95*, pp 297-305 Elsevier Science, Amsterdam
- Krol, E., *The Whole Internet* (1992) O'Reilly and Associates, New York
- Speh, M., et al, (1994) WWW for the Globewide Network Academy, *Proceedings of ICWWW '94*, pp 337-346, Elsevier Science, Amsterdam
- Thomas, I., *Principles of Teaching/Learning Process. Application of computer-assisted training programs in engineering education* (1996) University Press, Melbourne