Adaptive Formative Paths in a Web-based Learning Environment

Vincenza Carchiolo, Alessandro Longheu, Michele Malgeri
Dipartimento di Ingegneria Informatica e delle Telecomunicazioni
Università di Catania – V.le A. Doria 6 – 95125 Catania – Italy
Tel: +39-095-7382359
Fax: +39-095-7382397
Vincenza.Carchiolo@diit.unict.it
Alessandro.Longheu@diit.unict.it
Michele.Malgeri@diit.unict.it

ABSTRACT

During the last decade, the interest for e-learning systems has grown thanks to availability of bandwidth and powerful computers. Thus the e-learning has moved from specific environments to larger communities. The main challenge of the last generation e-learning systems is to provide courses tailored to different students backgrounds; this pushes the research to create an adaptive environment able to just-in-time craft the best path for each student. This paper deals with this problem, introducing a proposal of a system for e-learning which provides students with all formative paths moving from an initial to a desired knowledge, and where paths are adapted according to student needs and capabilities, and dynamically modified when such needs change. We also present a prototype of a web-based e-learning environment, through which students can follow dynamically adapted formative paths, also exploiting collaborative learning tools.

Keywords

E-learning, Adaptive formative paths, Web-based learning, Collaborative learning

Introduction

The increasing use of Web technologies has led to significant changes in the role covered by computers, that can be now considered as enhanced terminals through which several web-based application can be accessed.

Among these applications, E-learning (Soine, 2001; Luchini, 2001; Heinrich et al., 2001) is one of the most innovative, since it can (almost) radically change the learning process of many people. E-learning was initially developed inside specific environments (e.g., industries), where homogeneous contents were developed to homogeneous people communities.

In this paper we want to consider courses for students, where significant differences can characterize users, as the learning rate, personal capabilities and knowledge degree. This means that the same course must be adapted to each student, or to a group of them (i.e. a *class*) sharing some characteristics, as the desired (target) knowledge.

To satisfy this need, the use of new technologies, integrated in a proper environment (as a web-based one), can help the learning process. Computer-based interactions, indeed, allow the development of more active learning techniques, thus significantly improving the chance of adapting a course for each student, based on his/her capabilities and needs. This is in contrast with traditional lessons, where a single flow of information mainly exists, directed from the teacher to all students.

Moreover, a course must promote students cooperation as well as communication between students and the teacher, thus reducing the problem of students isolation that can characterize e-learning courses.

Based on these considerations and on the principles for a good educational systems described in (Chickering, 1987), the system proposed in this paper aims at promoting active learning techniques, allowing to adjust formative paths based on students needs and capabilities (named *profiles*), also exploiting feedback of each student activity; it also aims at promoting collaboration among students and between them and the teacher.

The system takes into account both the teacher and the student point of view: the teacher adjusts the course to the class, adapting it to the average capability and knowledge of the students and modifying how the lesson is proposed (examples, training, theory, etc.); the students point of view is considered through profiles, that store individual needs and capabilities.

In order to allow a greater flexibility, a course is modeled as a graph, where each node represents a subject matter and two nodes are connected only if the knowledge gained by learning the former node is needed as a pre-requisite to learn the latter node. Each oriented arc connecting two nodes is weighted in order to measure the difficulty to face the next node.

The target is to create and propose to the student all the possible paths starting from his/her knowledge and directed to the desired topic of interest, filtering such paths according to the student personal profile; weighted routes allow eliminating paths if they are too long or too difficult for that student.

The system can also adapt itself dynamically to student needs, for instance if he/she does not actually own an initially declared knowledge, system could detect these situation from exercises results and could re-evaluate and suggests a new path to that student; or the system could change the path if the student's available time varies.

This system has been implemented as a first prototype based upon open-source software, so students only need a simple browser to access the system, independently of any hardware/software architecture.

The prototype allows to attend on-line courses: each student first logs into the system, eventually specifying his membership to a class, then he/she chooses a course (or searches for a specific topic), and consequently all related formative paths are built, also based on student profile; finally, the student selects one path and attends the course (lessons, exercises, …).

Moreover, the prototype offers a set of tools to support collaboration among students and between them and the teacher during the entire learning process.

Finally, the prototype also allows the teacher to reuse educational materials already existing in other formats such as slides, electronic documents, images, or videos. To satisfy this goal, a course is built putting together all different required pieces in a web-based environment, where links among different elements are not directly embedded in the lesson itself but dynamically created and/or updated. In this way, the editor of the course avoids the troubles connected to links maintenance, having a greater degree of re-configurability of formative paths.

Next sections first describe the underlying model of the proposed e-learning system, then the prototype that has been implemented is shown, finally presenting conclusions and future work.

The Underlying Model

The system proposed in this paper is based on a model which aims at taking into account several ideas present in literature; thus it can be viewed as a general model for e-learning (Carchiolo et al., 2000). The model, shown in figure. 1, consists of several elements, described in the following subsections.

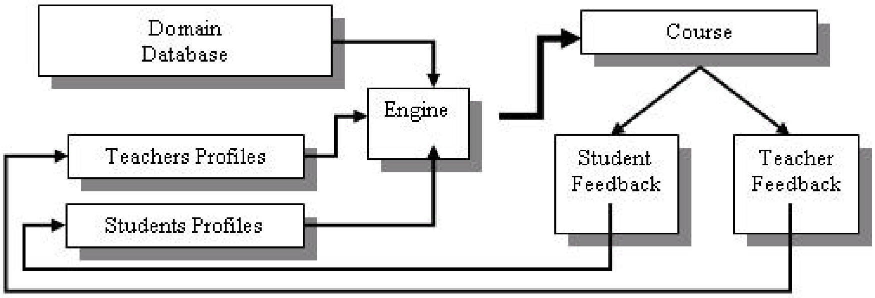


Figure 1. Engine components

The Domain Database

The first component is the domain database, which contains all information about each lesson (as its name, description and estimated time to learn it), together with links to the related educational material (e.g. slides, exercises, and so on), and with all semantics relationships among lessons, in order to exploit such links when building courses.

Several papers concerning domain databases are present in literature (Papaterpos, 2001; Cairò, 2001; Le, 2001); most of them come from artificial intelligence area, where the design of such databases has been widely analyzed as a general question.

Moreover, since the structure generally adopted for such databases is graph-based (Chiu, 2001; Fujiwara, 2001; Chan, 2001; Fung (1), 2000), we adopt the same structure for courses description. In particular, each course will be represented as a graph of Course Units (CU), each of which represents a *"one-argument lesson"* container which includes both information about that lesson and links to related educational materials.

All CUs belonging to the same course are connected together into a graph structure using oriented and weighted arcs; the orientation represents the 'precedence and succession' relation between two units, while the weight of an arc represents the difficulty to access a topic coming from a previous one. According to this semantics, not all CUs are actually connected, hence the graph is not fully connected. Arcs weights are managed by teachers, usually by establishing their initial values, which are further refined with the system help, e.g. through a statistical analysis of previous students sessions.

Students and Teachers Profiles

Two further components whose information are used for courses generation are the profiles databases, which contain all relevant data about teachers and students.

Considering the student, in literature several works are present (Abbas, 2001; Thomas, 2001; Fung (2), 2000), some are more pedagogical and/or psychological oriented (Kort, 2001; Aurum, 2001; Giraffa, 2001), others focus on students collaboration (collaborative learning), as (Ravenscroft, 2001; Wijekumar, 2001).

All these works lead to include several information into the student profile, as:

- the student's available time;
- the desired learning style, i.e. if the student prefers more practical-oriented or theoretical-oriented lessons about a given issue; this in order to select the desired level of detail and/or specific related materials (e.g., more exercises) for each unit;
- the desired level of difficulty, e.g. basic, intermediate, or advanced; this will help to select the length and/or the level of detail of formative paths;
- the desired media type, in order to select specific educational material for CUs (e.g. slides, or videos);
- the knowledge initially owned by the student, described with a set of keywords or with other indexes, obtained for instance with course admittance tests; this information is used to optimize courses building, avoiding already known CUs;
- other psychological-oriented information, as the satisfaction level and the comprehension level, both given as a numerical score, used to improve courses generation and/or the relevance order when presenting personalized formative paths to a student.

All these information are used to adapt courses to student profile, and also to assure dynamic behavior to the e-learning system, as described in the next section.

Considering the teacher, generally his role mainly consists in providing educational guidelines to be followed by students (Mora , 2001); such guidelines can include information as:

- which is the set of basic paths or sub-paths in accordance with some study programmes. In particular, the teacher obtains these paths by first accessing the system and exploiting the engine (described in the next section) to generate personalized formative paths based on the needs of the students class he conducts, and then he selects those paths which are in accordance with study programmes; a student belonging to that class will find these as "suggested" paths, in addition to his/her other personalized paths;
- which lessons (CUs) must be eventually considered as mandatory for a given class of students; this will in turn impose a constraint over personalized paths during their generation, highlighting the question of how to mediate between educational guidelines coming from teachers and courses customization coming from student needs;
- when and which exercises must be performed with respect to a given CU, set of CUs, or even suggested paths mentioned above.

All information is stored in the teacher profile and is used to build proper courses and to retrieve the desired material for CUs. Both students and teachers profiles are dynamically improved and adjusted using feedback modules shown in figure 1.

The Engine

Domain database and profiles are used to build all formative paths. To generate a formative path, several approach are possible (Galeev, 2001; Jesshope et al., 2000; De Oliveira, 2001; Mendez, 2001; Delgado, 2001). Here, paths finalized to the study of a specific matter are derived from the graph contained in the domain database thanks to the engine component shown in Figure 1.

First, the desired knowledge is provided by the student as a direct request or from search results, mapping such knowledge onto a target CU. Then, the engine accesses the domain database, retrieves that CU, and moves backward from this CU along all existing paths (semantic relationship), until a node representing a knowledge owned by the student is encountered. If such a node is not found for a given path, the search goes backward along that path until a *starting* node is found; starting nodes are those elementary CUs which are supposed to be learned without specific initial knowledge.

In a few words, the engine extracts from domain database all paths having the desired CU as final node, and starting from an initial knowledge (owned or not); note that if the desired knowledge is mapped on to a set of (not just one) target CUs, the engine perform the algorithm above for each target, retrieving paths which may or not share some of their CUs.

Figure 2 shows an example of paths retrieval for a given student. The precedence and succession relationships stored into the domain database are represented in figure 2 a). We suppose that target CU is the node 7 and that the student owns the knowledge provided by node set (2,4), thus all paths for that student will start with node 2 or 4.

Figure 2 b) shows all paths extracted according to the procedure described above: starting from node 7, moving backward along semantic relationships, nodes 6 and 5 are anyway collected, then two paths depart from node 5, a former stops at node 4 since it belongs to the owned knowledge, while the latter collects nodes 3 and then stops at node 2. Hence, two paths are extracted, i.e. 2-3-5-6-7 and 4-5-6-7.

Similarly, if student knowledge is represented with node set (3,12), the engine extracts the paths 3-5-6-7 and 4-5-6-7 (node 12 belongs to another path), as shown in figure 2 c).

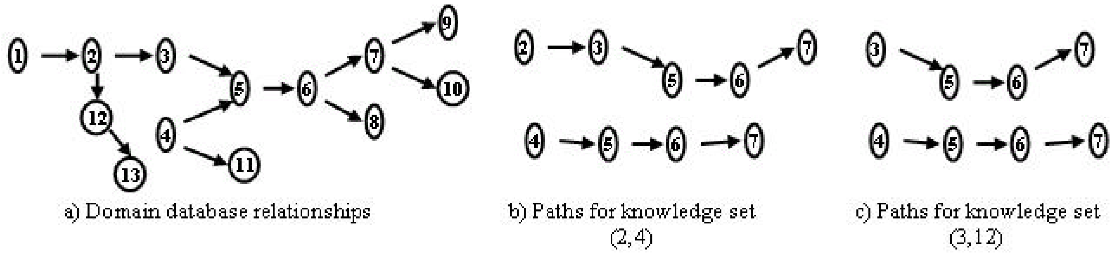
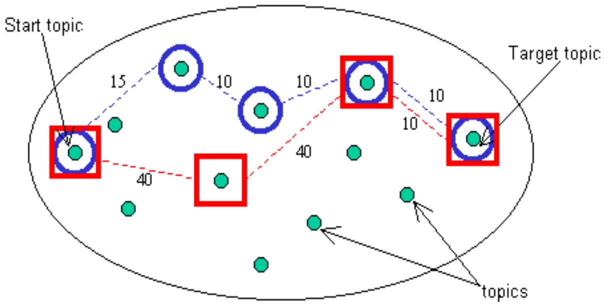

1
2
3
12
13
4
11
5
6
7
8
9
10
a) Domain database relationships
2
3
5
6
7
4
5
6
7
b) Paths for knowledge set (2,4)
3
5
6
7
4
5
6
7
c) Paths for knowledge set (3,12)

Figure 2. An example of paths retrieval for a given student

After all paths are retrieved and eventually optimized, the engine filters them comparing arc weights with information stored on student profile, as well as on teachers profiles (the system refers to the profile of the teacher which conducts the class that student belongs to).

Discarding those paths which are not in accordance with these needs, the system acts as *adaptive*.

Finally, remaining paths are presented to the student, which can select one path (personalized course) and learn its CUs.



Path	# Topics	Avg Diff	Difficulty
Circle	5	11,25	45
Square	4	30	90
…	…	…	…

Figure 3. Routing table to reach the Target CU

In figure 3, two paths are shown as examples; they belong to the same set of filtered paths and share, in this case, the same starting CU: a student can follow the path through “boxes”, which is shorter but harder, or select the path through “circles”, being a longer and easier path, given that both paths are compatible with student needs, knowledge and capabilities (e.g. both are less or equal to the desired difficulty level and available time values). Each route includes information about the difficulty for that student and with a numerical index weighting the “effort to learn” the target topic through it; however, more analysis is actually needed to better define what “difficulty” means.

The system also includes the possibility of modifying the path during the learning process, for instance suggesting some revision if exercises results highlight that a declared knowledge is not actually owned by the student, or bypassing some lesson if the student available time varies, or suggesting a shorter/harder (or longer/easier) path if total time spent denotes the student is faster (respectively, slower) than expected; all these behaviors make the system *dynamic*.

The Prototype

The architecture of the prototype of the e-learning system is shown in figure 4. Note that we designed the system adopting the "open source" approach (Open Source, 2001), assuring that just a simple browser is needed, as well as gaining simple expandability, wide support and high performance.

Based on these considerations, the prototype core of the E-learning system is based on the Apache Web Server, completed with Php (PHP Language, 2002) for server-side scripting and MySql (MySQL Database, 2002) for database management. The system is accessed from a site (named E-Learning Portal in figure 4) where students can attend courses, perform exercises and exams, as well as exploiting several collaborative tools, as described later. The E-Learning Management System includes the web server, which interacts on one hand with the E-Learning Portal, while on the other it is connected to both Php Scripts and Services (see figure 4). The former represents the main implementation of the system, as scripts implements the engine and allow domain databases and students/teacher profiles management. The latter represents the implementation of all other services, for instance Services module could include a video server together with related plug-ins, if video streaming is provided, or Java-based chat service, if collaboration is provided, and so on. The last component of the E-Learning Management System is the MySQL Database, which actually includes domain database, as well as profiles databases.

In the following, a description of services provided by this system is introduced (Carchiolo, 2000).

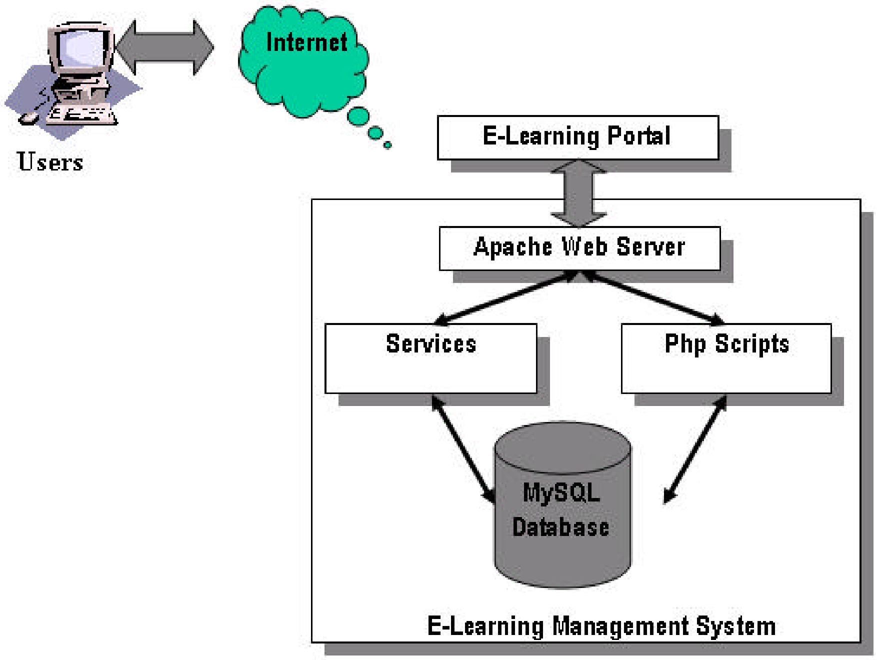


Figure 4. Architecture of the prototype of the e-learning system

Accessing the System

The system is accessed through a web based interface, which is currently the platform most widely adopted (Kuittinen, 2001). The core of the system is an e-learning portal, i.e. a web site where several services are available to build a web based community for learning activities. Such portal is on one hand accessible by final users (both students and teachers), while on the other hand it is connected to the E-learning Management System.

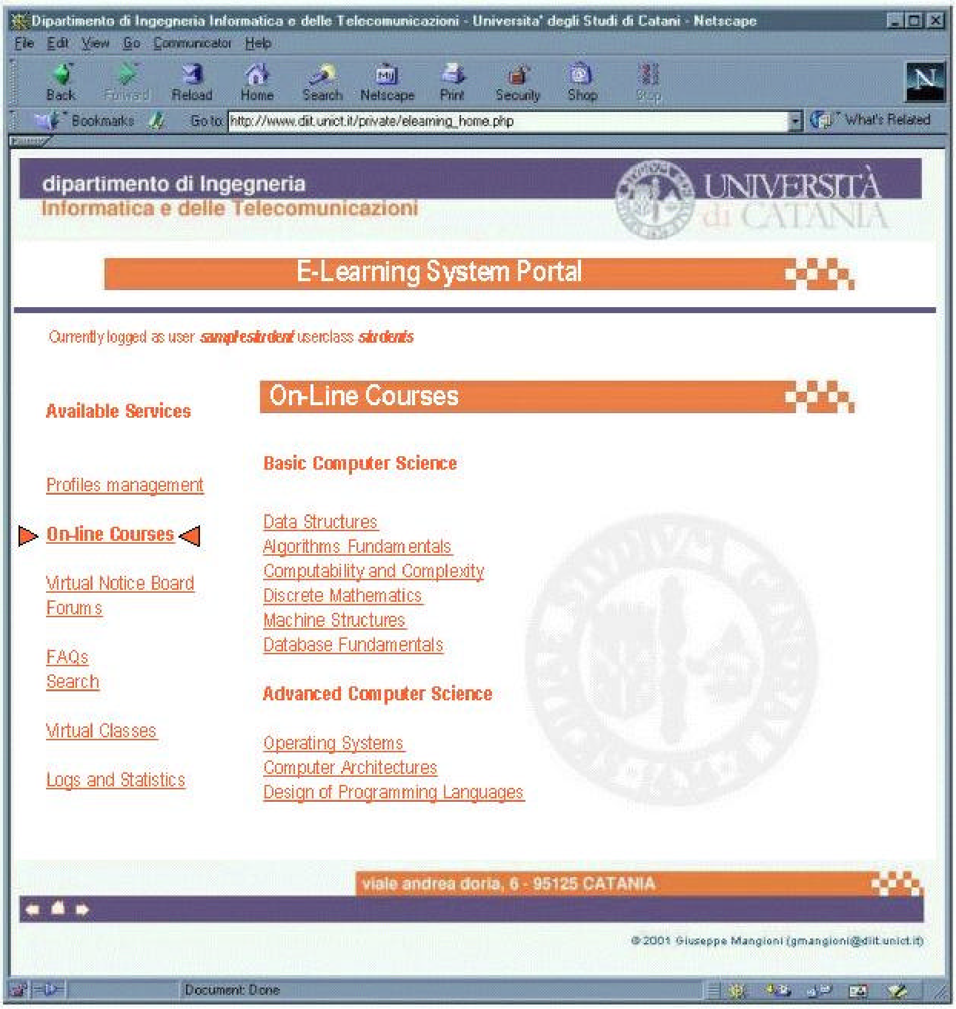
Students and teachers can access the system from anywhere, thanks to the web-based approach, provided that a proper user identifier and password must be provided in order to log in the system. The use of user-id and password:

- allows to grant access only to registered users;
- provides the system with auditing (logging) capabilities for each single users, used for further monitoring and extracting personal and statistic information about system usage;
- provides a basic security level which can be further improved by using specific protocols, e.g. secure Http (HTTPS), helpful especially when sensitive information are being exchanged (as during on-line test performance);
- allows to define user classes, in order to distinguish students from teacher and from system administrators (thus providing them with different rights and/or limitations), as well as grouping users for several purposes, as to create virtual classrooms or exams sessions, meetings, teacher update courses, and so on.

Course Attendance

The most important service is the course attendance, where, at the first access, the system requires the user to fill in a form where all information for student's profiles is acquired.

Then, the student can either attend one of the available courses or search for a specific topic. Courses are arranged into categories and presented as a web page, as shown in figure 5, where *Basic* and *Advanced Computer Science* categories are represented as examples; by clicking on a specific course, the related page will show all information about that course, as its name and description, its duration, its position with respect to study programmes, which knowledge is required to attend that course and which knowledge will be gained by attending it (other information can be easily added).


Dipartimento di Ingegneria Informatica e delle Telecomunicazioni - Universita' degli Studi di Catani - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks
Go to: http://www.diit.unict.it/private/elearning_home.php
What's Related
Document: Done

dipartimento di Ingegneria
Informatica e delle Telecomunicazioni

UNIVERSITÀ di CATANIA

E-Learning System Portal

Currently logged as user *samplestudent* userclass *students*

Available Services

Profiles management

▶ On-line Courses ◀

Virtual Notice Board
Forums

FAQs
Search

Virtual Classes

Logs and Statistics

On-Line Courses

Basic Computer Science

Data Structures
Algorithms Fundamentals
Computability and Complexity
Discrete Mathematics
Machine Structures
Database Fundamentals

Advanced Computer Science

Operating Systems
Computer Architectures
Design of Programming Languages

viale andrea doria, 6 - 95125 CATANIA

© 2001 Giuseppe Mangioni (gmangioni@diit.unict.it)

Figure 5. E-Learning system portal

Once the student has chosen a course or searched for a specific topic, personalized formative paths are created by the engine component using both profiles and domain database data, as described previously. Each path is presented with a set of information helpful to make the best choice, as the path length (number of CUs), its difficulty (similarly to figure 3), the required time, the style (practical/theoretical), required and gained knowledge.

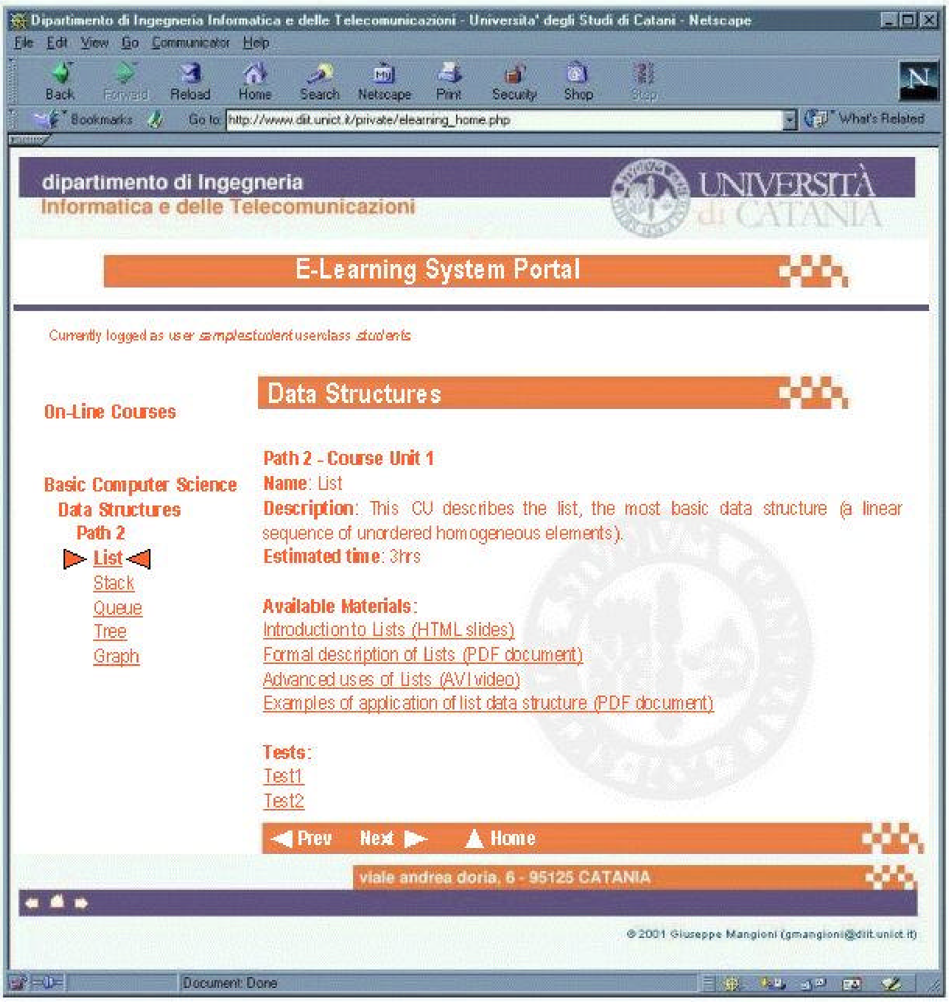
The student finally selects one of these proposed paths and attends the course. Each CU will appear (figure 6) as a web page containing information about that unit (these are similar to course information), plus all related materials, as slides, documents, videos, related links and so on. The students learns by accessing all these information, but he can also move forward along the path towards other CUs, provided that final exercises related to the current CU must be performed whenever present, or he/she can move backward to previous CUs for revising some concepts, or finally he can leave that path by selecting another path or searching for other topics.

Exercises and Tests

The student can perform exercises and tests related to a given CU, to a set of CUs or to a whole course; such exercises are generally performed at the end of lessons, even if this is not mandatory. The role of exercises (Callear , 2001; Alvarez, 2001) is fundamental for:

- ➢ feedback purposes, thus on one hand a student can verify his acquired knowledge, while on the other the teacher can evaluate the student learning and the quality of the followed path;
- ➢ educational reasons, i.e. to allow students a better comprehension of a given issue.

To increase the configurability, any test or exercise is composed of several basic elements; a question is endowed at least with a comment box where an answer in open or closed form can be provided. If needed, a link can be shown from which to download files containing data for the question in addition to the function of uploading a file as an answer itself (for the open form case). Any question can be related to different topics, allowing, during the test, reviews on past argument in the followed path and dynamic optimization of the current route. Finally, all tests are dynamically generated, offering questions always in a random order and with different answers picked up from the database provided by the teacher.


Dipartimento di Ingegneria Informatica e delle Telecomunicazioni - Universita' degli Studi di Catani - Netscape
File Edit View Go Communicator Help
Back
Forward
Reload
Home
Search
Netscape
Print
Security
Shop
Stop
Bookmarks
Go to: http://www.diit.unict.it/private/elearning_home.php
What's Related
dipartimento di Ingegneria
Informatica e delle Telecomunicazioni
UNIVERSITÀ di CATANIA
Document: Done

E-Learning System Portal

Currently logged as user *samplestudent* userclass *students*

Data Structures

On-Line Courses

Basic Computer Science
Data Structures
Path 2
▶ List ◀
Stack
Queue
Tree
Graph

Path 2 - Course Unit 1
Name: List
Description: This CU describes the list, the most basic data structure (a linear sequence of unordered homogeneous elements).
Estimated time: 3hrs

Available Materials:
Introduction to Lists (HTML slides)
Formal description of Lists (PDF document)
Advanced uses of Lists (AVI video)
Examples of application of list data structure (PDF document)

Tests:
Test1
Test2

◀ Prev Next ▶ ▲ Home

viale andrea doria, 6 - 95125 CATANIA

© 2001 Giuseppe Mangioni (gmangioni@diit.unict.it)

Figure 6. Example course unit

In case of closed form answers, the student will receive a mark at the end of the test, while open form answers are not immediately evaluated, rather the teacher will receive the answer, evaluate it and e-mail the mark to the student.

This mark is expressed in percentage: the system calculates the maximum score obtainable as sum of the top quoted answers in each question and it compares this with the student score.

Test execution does not stop anyway the student from following the formative path in case of an inadequate mark, rather the system suggests to review again the topic.

Collaborative learning features

In addition to course attendance and exercises support, the system comes with a set of functionalities used to support collaboration among students and between them and the teacher. They can be classified into asynchronous and synchronous functionalities. Asynchronous functionalities group includes tools as:

- Virtual notice board, where messages and news about courses, exams, meetings, and other activities are accessible; messages can be managed both by teachers and authorized students;
- Forum area, where students can create virtual study groups in order to exchange messages, both by chat services or through mailing lists, documents, slides, videos, comments and questions about courses; such groups can be or not supervised by the teacher;
- FAQ databases, used to search for most frequently asked questions about courses, or concerning a specific topic;
- Search engine, to find out all information (CUs) about a given topic, or simply to search the site.

Synchronous activities mainly consist of Virtual classes, where on-line interactive lessons may be attended by students and supervised by the teacher. In this case, interaction is assured through:

- audio/video live sessions;
- providing a virtual blackboard, managed by the teacher which can perform a presentation of a lesson, or can give the control to a student (e.g. to perform an exercise);
- sharing an application, in order to work together with students.

Note that the integration of all these services allows the system to effectively support E-learning, in particular the collaborative learning, whose importance has been enforced also thanks to the combined use such technologies (Milrad , 2001).

Monitoring activities

In order to monitor didactic activities, the system has been designed to have three kinds of feedback: to the student, to the teacher, and to the system itself. Immediate test results represent the first and more direct one for to the student (so he/she can verify his/her acquired knowledge); the others are achieved through log files analysis. Log files tracks information about each student, as well as statistical information about all students; personal information include:

- which courses have been chosen;
- which kind of paths have been chosen;
- which related topics have been actually studied;
- the time spent;
- the knowledge achieved (by analyzing exerc ises results);
- the number of access to the system;

Global information include:

- the number of registered students for each course;
- the ratio between this number and the total number of students (used as a significance level for that course);
- the number of students which decrease the access or abandon a course;
- data about satisfaction and comprehension level, also coming from performance (marks distribution) analysis.

Reports coming from these logs allows feedback for the teacher, which can eventually recalibrate exercises or modify the lessons design and content, acting on the domain databases or on course materials, whereas the system updates and improves students and teachers profiles as well as course generation.

Conclusions

A great advantage for all students when learning would be to have a personal tutor adjusting the formative path for him. This dream is approached to become a reality thanks to the increasing use of E-learning systems.

The chance to supply courses inside computer-based environment is not a new issue, especially inside specific contexts (e.g., business refresher courses). The proposal of this paper represents an attempt to create a general system for E-learning facing the more common needs of a generic student; this goal imposes a dynamic and adaptive system able to satisfy student needs in order to obtain a wide-spread and effective E-learning system.

The system presented in this paper aims at achieving such a target: it is based on a model that includes a domain database containing all Course Units, together with links to their related educational materials, as well as with semantic relationships between all such lessons. The system exploits such information, plus all preferences stored into both students and teachers profiles databases to build customized formative paths, assuring *adaptation* to specified needs; moreover, the system also allows the on-line paths evaluation, thus acting as a *dynamic* system.

These features, accessed from a web-based environment which also provides collaborative tools, can improve significantly the entire learning process.

Some issues are still open; simply put, it is important to address the question of how:

- to analyze the granularity of each CU, establishing for instance if several abstraction levels are needed or not;
- to extend profiles information, taking into account further needs coming from both students and teachers;
- to explore how to improve the algorithm used to generate courses;
- to define a common description language (e.g., XML-based) to represent educational materials, both to describe their contents, as well as to categorize and collect them when building courses;
- to exploit feedback modules, in order to improve the system behavior;

References

Abbas, J., Norris, C., & Soloway, E. (2001). Analyzing Middle School Students' Use of the ARTEMIS Digital Library. In T. Okamoto, R. Hartley, Kinshuk & J. P. Klus (Eds.) *Proceedings of IEEE International Conference on Advanced Learning Technologies (ICALT2001)* (August 6-8,2001, Madison, WI, USA), Los Alamitos, CA: IEEE Computer Society, 107-108.

Alvarez, M. (2001). Exploring Minds: Revealing Ideas Electronically. *Paper presented at the World Multiconference On Systemics, Cybernetics And Informatics,* July 22 - 25, 2001, Orlando, Florida, USA.

Aurum, A., & Handzic, M. (2001). Software Support for Creative Problem Solving. In T. Okamoto, R. Hartley, Kinshuk & J. P. Klus (Eds.) *Proceedings of IEEE International Conference on Advanced Learning Technologies (ICALT2001)* (August 6-8,2001, Madison, WI, USA), Los Alamitos, CA: IEEE Computer Society, 160-163.

Cairò, O., Barreiro, J., & Solsona, F. (2001). KAMET: An Evolved KA Methodology. *Paper presented at the World Multiconference On Systemics, Cybernetics And Informatics*, July 22 - 25, 2001, Orlando, Florida, USA.

Callear, D., Jerrams-Smith, J., & Soh, V. (2001). Bridging Gaps in Computerised Assessment of Texts. In T. Okamoto, R. Hartley, Kinshuk & J. P. Klus (Eds.) *Proceedings of IEEE International Conference on Advanced Learning Technologies (ICALT2001)* (August 6-8,2001, Madison, WI, USA), Los Alamitos, CA: IEEE Computer Society, 139-140.

Carchiolo V., Longheu A., Malgeri M. (2000). Dynamic web pages for tuning formative path, *DIIT Technical Report #1013, University of Catania*.

Chan, A., Chan, S., & Cao, J. (2001). SAC: A Self-Paced and Adaptive Courseware System. In T. Okamoto, R. Hartley, Kinshuk & J. P. Klus (Eds.) *Proceedings of IEEE International Conference on Advanced Learning Technologies (ICALT2001)* (August 6-8,2001, Madison, WI, USA), Los Alamitos, CA: IEEE Computer Society, 78-81.

Chickering W., Ehrmann, S.C. (1987). Seven Principles for Good Practice in Undergraduate Education. *AAHE Bulletin.*

Chiu, C. (2001). The Authority Structure Problem of Computer Supported Collaborative Concept Mapping System for Elementary Student. In T. Okamoto, R. Hartley, Kinshuk & J. P. Klus (Eds.) *Proceedings of IEEE International Conference on Advanced Learning Technologies (ICALT2001)* (August 6-8,2001, Madison, WI, USA), Los Alamitos, CA: IEEE Computer Society, 57-56.

Delgado, M., & Molero, J. (2001). Engineering the Teaching Support System in the Open and Distance Higher Education. *Paper presented at the World Multiconference On Systemics, Cybernetics And Informatics,* July 22 - 25, 2001, Orlando, Florida, USA.

De Oliveira, C., & Alves, J. B. M. (2001). Object Oriented Modelling on Fresh Curriculum of Computer Sciences Disciplines – An Integrative Pedagogic Proposal. *Paper presented at the World Multiconference On Systemics, Cybernetics And Informatics*, July 22 - 25, 2001, Orlando, Florida, USA.

Fujiwara, Y., Matsuzawa, B., & Okada. S. (2001). Learning Assistance Expert System with a Self-Adaptive Function Based on a Causal Network. *Paper presented at the World Multiconference On Systemics, Cybernetics And Informatics*, July 22 - 25, 2001, Orlando, Florida, USA.

Fung, I. (1) (2000). A hybrid approach to represent and deliver curriculum contents.

Fung, I. (2) (2000). On monitoring study progress with time -based course planning.

Giraffa, L., & da Costa Móra, M. (2001). Towards Student Models (Really) Based on Mental States. Paper presented at the World Multiconference On Systemics, Cybernetics And Informatics, July 22 - 25, 2001, Orlando, Florida, USA.

Heinrich, E., Jesshope, C. R., & Walker, N. (2001). Teaching Cognitively Complex Concepts: Content Representation for AudioGraph Lectures. In Montgomerie C. & Jarmo Viteli (Eds.) Proceedings of EdMedia 2001 (June 25-30, 2001, Tampere, Finland), Norfolk, VA, USA: AACE.

Jesshope, C., Heinrich, E. & Kinshuk (2000). Technology Integrated Learning Environments for Education at a Distance. Paper presented at the DEANZ 2000 Conference, 26-29 April 2000, Dunedin, New Zealand.

Kort, B., Reilly, R., & Picard, R. W. (2001). An Affective Model of Interplay between Emotions and Learning: Reengineering Educational Pedagogy – Building a Learning Companion. In T. Okamoto, R. Hartley, Kinshuk & J. P. Klus (Eds.) Proceedings of IEEE International Conference on Advanced Learning Technologies (ICALT2001) (August 6-8,2001, Madison, WI, USA), Los Alamitos, CA: IEEE Computer Society, 43-48.

Kuittinen, M., Pöntinen S., & Sutinen E.,(2001). How to Design Web-Based Counseling Systems. In T. Okamoto, R. Hartley, Kinshuk & J. P. Klus (Eds.) *Proceedings of IEEE International Conference on Advanced Learning Technologies (ICALT2001)* (August 6-8,2001, Madison, WI, USA), Los Alamitos, CA: IEEE Computer Society, 178-179.

Le, T., & Nguyen, T. H. (2001). Toward an Universal Knowledge Management: Using Ontological Approach with Methods and Tools. *Paper presented at the World Multiconference On Systemics, Cybernetics And Informatics*, July 22 - 25, 2001, Orlando, Florida, USA.

Luchini, K., Oehler, P., Quintana, C., & Soloway, E. (2001). An Engineering Process for Constructing Scaffolded Work Environments to Support Student Inquiry: A Case Study in History. In T. Okamoto, R. Hartley, Kinshuk & J. P. Klus (Eds.) *Proceedings of IEEE International Conference on Advanced Learning Technologies (ICALT2001)* (August 6-8,2001, Madison, WI, USA), Los Alamitos, CA: IEEE Computer Society, 55-56.

Mendez, G., de Antonio, A., & Herrero, P. (2001). PRVIR: An Integration Between an Intelligent Tutoring System and a Virtual Environment. *Paper presented at the World Multiconference On Systemics, Cybernetics And Informatics*, July 22 - 25, 2001, Orlando, Florida, USA.

Milrad, M. (2001). From Concrete Experiences to Abstract Formalisms: Learning with Interactive Simulations that Combine Physical and Computational Media. In T. Okamoto, R. Hartley, Kinshuk & J. P. Klus (Eds.) *Proceedings of IEEE International Conference on Advanced Learning Technologies (ICALT2001)* (August 6-8,2001, Madison, WI, USA), Los Alamitos, CA: IEEE Computer Society, 141-144.

Mora, M., & Moriyón, R. (2001). Collaborative Analysis and Tutoring: The FACT Framework. In T. Okamoto, R. Hartley, Kinshuk & J. P. Klus (Eds.) *Proceedings of IEEE International Conference on Advanced Learning Technologies (ICALT2001)* (August 6-8,2001, Madison, WI, USA), Los Alamitos, CA: IEEE Computer Society, 82-85.

MySQL Database (2002). http://www.mysql.org.

Open Source (2002). http://www.opensource.org/osd.html.

Papaterpos, C., Georgantis, N P., & Papatheodorou, T. S. (2001). An Ontology for Modeling Ill-Structured Domains in Intelligent Educational Systems. In T. Okamoto, R. Hartley, Kinshuk & J. P. Klus (Eds.) *Proceedings of IEEE International Conference on Advanced Learning Technologies (ICALT2001)* (August 6-8,2001, Madison, WI, USA), Los Alamitos, CA: IEEE Computer Society, 41-42.

PHP Language (2002). http://www.php.net.

Ravenscroft, A., & Matheson, M. P. (2001). Carpe Diem: Models and Methodologies for Designing Engaging and Interactive e-leaming Discourse. In T. Okamoto, R. Hartley, Kinshuk & J. P. Klus (Eds.) *Proceedings of IEEE International Conference on Advanced Learning Technologies (ICALT2001)* (August 6-8,2001, Madison, WI, USA), Los Alamitos, CA: IEEE Computer Society, 74-77.

Soine, R (2001). Instructional Design in a Technological World: Fitting Learning Activities into the Larger Picture. In T. Okamoto, R. Hartley, Kinshuk & J. P. Klus (Eds.) *Proceedings of IEEE International Conference on Advanced Learning Technologies (ICALT2001)* (August 6-8,2001, Madison, WI, USA), Los Alamitos, CA: IEEE Computer Society, 49-52.

Thomas, P., & Paine, C. (2001). How Students Learn to Program: Observations of Practical Tasks Completed. In T. Okamoto, R. Hartley, Kinshuk & J. P. Klus (Eds.) *Proceedings of IEEE International Conference on Advanced Learning Technologies (ICALT2001)* (August 6-8,2001, Madison, WI, USA), Los Alamitos, CA: IEEE Computer Society, 170-173.

Wijekumar, K. (2001). Implementing Collaborative Learning Research in Web-Based Course Design and Management Systems. In T. Okamoto, R. Hartley, Kinshuk & J. P. Klus (Eds.) *Proceedings of IEEE International Conference on Advanced Learning Technologies (ICALT2001)* (August 6-8,2001, Madison, WI, USA), Los Alamitos, CA: IEEE Computer Society, 86-89.