e-Class Personalized: Design and Evaluation of an Adaptive Learning Content Management System

Evelthon G.Prodromou and Nikolaos Avouris
Human-Computer Interaction Group,
Electrical and Computer Engineering Dept., University of Patras
GR-26500, Patras, Greece
eprodromou@upnet.gr, avouris@upatras.gr

Abstract. This paper presents e-Class Personalized (e-CP), a new extension of the widely available open source Learning Content Management System e-Class. e-CP monitors interaction of the users of e-Class with its content and services and adapts the services to better suite the users' interests and tasks. E-CP has been tested for over a year at the University of Patras e-Class server with over 20,000 users. It drew positive response by the user population that were exposed to this version of the leaning Content Management System. In this paper we discuss architectural decisions and evaluation results of e-CP.

1 Introduction

Learning Management Systems (LMS) are software applications based on internet technologies that support management and delivery of distance learning content and services. LMSs need to be usable, reliable, affordable for their users and educationally relevant. The use of LMS in education offers definite advantages. More specifically they overcome time and space constraints, they offer flexibility in learning methods, they support extensive interaction between teachers and students whilst simultaneously allowing for inexpensive management and modification of learning resources.

The main functionality that a typical LMS needs to provide to its end users, students, teaching staff and administrators, includes: (a) *Course Management* which concerns tools that deal with the creation, administration, adaptation and supervision of courses. (b) *Classroom Administration* which includes tools that deal with handling of students and trainees, the creation of study groups, assignments etc. (c) *Communication tools* that supporting real time and asynchronous interaction between students and tutors. Such tools include email, chat rooms, voice and image conferencing, announcements and agenda management. Advanced LMS offer even

Please use the following format when citing this chapter:
Prodromou, Evelthon, Avouris, Nikolaos, 2006, in IFIP International Federation for Information Processing, Volume 204, Artificial Intelligence Applications and Innovations, eds. Maglogiannis, I., Karpouzis, K., Bramer, M., (Boston: Springer), pp. 409–416

more possibilities for cooperative learning such as sharing of archives and applications, sharing of workspaces and the whiteboard, while often more advanced services like synchronous design and modeling tools may be supported, e.g. see [1]. (d) *Students' Tools* facilitate students' access, administrate and study of the available material. Examples of such tools are private and public notes on text, indexes, personal histories, offline study, search engines etc. (e) Content management which deals with the tools that create, store and distribute learning material, the administration of archives, addition and extraction of learning material etc. (f) *Evaluation tools* deal with the administration of assignments on the internet, delivery of activities, self evaluation tests, student participation statistics etc. (g) *School administration* deals with the tools that handle student presence control, student performance, student registration, personal data of the students, financial matters, timetables etc.

Each one of these services is desirable to have an interface which adapts to the user's needs and requirements. In addition, the adaptable content of each service may affect the main system interface so that the user has overall control of the system the moment he/she connects to the system. These are some of the requirements for adaptivity of LMSs which have been tackled in the research reported in this paper. In the following we discuss general issues and state of the art of adaptive web applications and in particular adaptive LMSs and subsequently we focus in a specific case of design, development and evaluation of such a system, the e-Class Personalized, which permits adaptation of interaction according to the history of system usage by each individual user.

2 On Adaptive Learning Management Systems

The rapid development and wide use of the internet has influenced the way we access information in general and on-line learning services in particular. An internet application or an internet portal may provide large amounts of information which can meet the preferences and expectations of many different users. Most Universities and educational establishments have added such technological support to their users, facilitating and enhancing the face-to-face everyday teaching and other educational activities.

Users of such establishments are usually a heterogeneous group with widely different needs. Therefore the great quantities of information available may cause confusion and disorientation. This is what the development of web personalization services aim to solve. This approach deals with adaptation of interaction between the user and the internet application in order to serve these widely diverging users' needs and preferences.

A learning management system assumes the existence of distinct user roles (teacher, trainee, administrator). The selection of information presented to the end-user is based on these roles. The administrator has full access to the learning management system, the teachers manages content of the courses he/she is responsible for and the student has the appropriate information for the courses he/she

is registered for. So the adaptive user interface deals with different needs of each group of end-users.

In order to be useful to individual learners, LMS, must be adaptive, since when learning from a Web-tutor there is often no availability of supporting colleagues to provide assistance as in a normal classroom situation. Minimum adaptivity of a Web- based educational application, according to [2] includes collecting, some data about the student working with the system and creating, the Student Model. This Model can be then used to adapt the presentation, of the course material, e.g. ranking content according to the user's interests, adapting navigation through it, sequencing, and annotation, to the student. Further possible levels of adaptivity are discussed in [3], while there are a number of examples of such systems and architectures, as in [4], [5]. In the rest of the paper we describe our experience with development and evaluation of an adaptive LMS for a widely available open source management system, e-Class.

3 Design of e-Class Persolized (e-CP), an adaptive open source Learning Management System

e-Class (www.eclass.gr) is an open source learning management system. The development of this software has been the result of an initiative of *GU Net* the organization that supervises the Greek Universities backbone network and provides value added services to its members. e-Class has been spawned from the Claroline project, a European open source LMS (www.claroline.net). Currently e-Class and Claroline development follows different tracks, resulting in two different platforms, despite the common origin. e-Class has been used by all major Greek Universities with many hundreds of thousands of users. Contribution to the development of this platform has been made by many developers in Greece. The University of Patras HCI lab has contributed with the e-Class Personalized (e-CP) which is discussed here. This is currently under consideration to be included in one of the forthcoming major releases of the e-Class platform. The main considerations of e-CP design are discussed first. These will provide the adaptive behavior pattern of the system. Taking these into account we then decide ono issues like the user interface to be developed as well as key architectural decisions, like the development of a user modeling system, based on users' interaction data.

3.1 Overview of requirements

One of the main aspects of e-CP is related to the navigation support and design of the user-centered interface design. The key design issue was related to the redesign of the end user *home page*. Analysis of the previous design of e-Class demonstrated that it was heavily course-centered, instead of user-centered. The user entering the system was faced with a list of courses, like in most such platforms. So any changes in a course content, for which a user is registered, are not immediately clear to the user upon entering the system. The user has to check on a daily basis for

any changes in all the courses content he/she is registered for, a tedious process, requiring a long sequence of clicks.

It is also clear that the system needs to provide a greater degree of adaptivity. The home page, central to users' interaction and conceptualization of such a system, needs to have a more meaningful content. This has to adapt to the needs of each user and user group, depending on the user and his/her role on the platform. In addition, the user needs to continue having access to the services provided by the current home page. At the same time a study needs to be carried out to identify the services to which the user should have access to directly from this home page and to identify how these services should be presented to facilitate the search and selection of information.

Considering the current system, the user should continue to have access to the list of courses he is registered for as well as the navigation menu. In addition he/she needs instant access to all important course tools and information, while a special attention focusing mechanism should be devised that permit immediate attention drawing on any changes which might have occurred, since the previous visit.

So an Adaptive Subsystem was designed and developed as a module, to meet these requirements. This subsystem, in effect takes over the control from the central system of e-class when a user logs in and adapts the user interface according to the specific characteristics of the user, found in the User Model. The most prominent characteristic of this module is to provide all relevant and timely information grouped in the same page, using a ranking mechanism to allow for the most relevant information to appear at the top of the list, taking in consideration historical data of interaction.

In addition technical considerations have to be met. An adaptive system has as its primary goal the reformulation of the content and services in order to meet the characteristics of different users. This has to occur through an automated and fast procedure.

The adaptive system has to be easy to use and further maintain and support. It must provide the capacity to add new functionality without major changes in the source code, especially in the context of an open source project, where maintainability of the application is one of the main considerations. To achieve these objectives, a modular architecture was defined. In addition, one of the main technical challenges was to interface to the existing system and take care not to burden the service, deteriorating performance.

The user interface of the adaptive system needs to be significantly richer to the initial one. This is a result of the fact that the user is now provided with greater amounts of information and expanded interaction possibilities. However special care needs to be taken so that there is continuity between the new and the old user interface, since such systems have large numbers of users who do not wish to be faced with major discontinuity of their already developed mental models of the application. At the same time the additional information and services in the interface has to be properly organized and provide coherent instructions.

In the following section we describe how the developed module has met these requirements. This is done through a number of evaluation tests.

4 Formative evaluation of interaction– Keystroke Level Analysis

An analytical method called Keystroke Level Analysis (KLM) [6] was initially used for testing the developed concept of the adaptive system interface. This is a technique used during the requirements development and initial planning phase in order to assess the efficiency of a system. This is part of the GOMS knowledge model. This model is based on the assumption that the user is experienced and makes no mistakes while using the system. This is a laborious method if it is to be used on extensive parts of the interface. It may though offer quite accurate results. It is particularly for comparative studies among suggested alternative designs.

For the system we used the adaptive interface which provides access to services like announcements, deadlines, schedule and a discussion forum. In addition we have menu options and the user's list of courses. The analysis method was applied on these items using the KLM model. The results are shown on table 1.

As seen from the results in table 1 there is significant access improvement (61% in average) for each tool. What is important is that the time needed (in the personalized version) may even be decreased to zero if there is no other change and if this is notified to the user.

Table 1. Results of the KLM evaluation model on the tools used for the personalization interface. T1 is the time the user needs to check the specific tool, and T2 is the same time through the personalized interface

Tool	eClass (sec)	eClass personalized (sec)
Announcements	T_1=a(5,86)	T_2=b(4,48)
Assignments	T_1=a (8,59)	T_2=d(4,48)
Agenda	T_1=a (5,86)	T_2=f(4,48)
Forum	T_1=a (11,32)	T_2=h(4,48)

As an example we can show a part of the personalized interface which regards the assignments tool.

In figure 1 we see the reason behind the significant access time improvement to the tool. Using the personalized interface the steps required to check or submit an assignment are decreased to one from three initially.

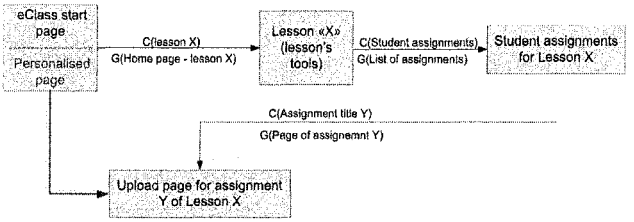
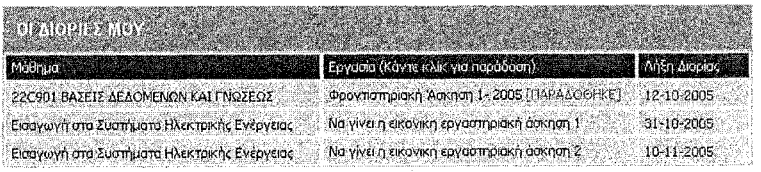


Fig. 1. The STD diagram of the the task "Deliver Assignment" when using the existing system (black path) and the personalized system (red path)

In figure 2 we see the structure of the assignments tool in the personalized module. There is clear distinction between columns and rows, clearly indicating the course, the assignment and the deadline. At the same time the second column indicates that it is active, a link to the assignments upload tool. This is made

possible by clicking on the assignment title. At the same time we are provided with delivery confirmation for the assignment. The message "delivered" appears next to the assignment title. This is based on the existing system use which allows resubmission of an assignment within the deadline.

Therefore, in planning an adaptive interface one must take care to provide the functionalities already offered by the system for each tool that need to personalize to user needs.



Οι Διορίες μου

Μάθημα	Εργασία (Κάντε κλικ για παράδοση)	Λήξη Διορίας
22C901 ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΓΝΩΣΕΩΣ	Φροντιστηριακή Άσκηση 1 - 2005 [ΠΑΡΑΔΟΘΗΚΕ]	12-10-2005
Εισαγωγή στα Συστήματα Ηλεκτρικής Ενέργειας	Να γίνει η εικονική εργαστηριακή άσκηση 1	31-10-2005
Εισαγωγή στα Συστήματα Ηλεκτρικής Ενέργειας	Να γίνει η εικονική εργαστηριακή άσκηση 2	10-11-2005

Fig. 2. The assignments block when the user has assignments to deliver. There are three columns. The first column is the Course column. The second column is the Assignment column that informs us of the title of our assignment due. The third column is the Deadline column that informs us when the assignment will expire.

5. Design and performance of the Adaptive Component

In this section we discuss implementation problems and performance testing of the Adaptive CLM system developed.

The e-class environment structures data per course. This data model is reflected in the user interaction. So a new user-centered interaction model needs to be based on a transformation of this data model, without however imposing high computational load at run time, caused by the necessary queries in multiple databases. Our aim here is the construction of an algorithm which present in an efficient and timely way relevant data taking into account the user preferences. The system keeps records of user access, such as date and time of accessing a course and the user ID. The information in the log file is processed at regular intervals and are then placed in a scoreboard. Each user-course combination is a distinct entry on the board. An adaptive system needs to maintain a record of user's access and accordingly rank courses in order of importance for the user. This way, courses of high importance are easily accessible as can be found at the top of the courses list. The same principle is also applied to other services, like agenda items, assignments, scheduled events etc.

Let us consider a course classification index $ERA \in [0,1]$, where 0 (zero) indicates that the user has never visited the course and 1 that the user continually visits this course, without visiting any other.

Let us discuss the algorithm for calculating ERA for all courses of a student. Suppose that after processing the data from the logging table we are provided with the following information for a student: Total course hits:100 and Hits on the specific course: 35

A first estimation could be: $ERA = \frac{35}{100} = 0{,}35$, which is a good initial estimate.

Suppose that the scoring algorithm is executed each week. If for a course we have 35 ± 3 hits, the position of a course on the course list will not be significantly changed.

But what happens if the course is visited 35 ± 3 times per week and for one week the user needs to access it far less or far more often? In this case the position of the course on the list will change even though the particular week is an exception to the usual usage pattern. This is not a desirable change. To compensate for this we add "memory" to the system. The algorithm will be executed once a week but will "remember" user preferences. This "memory" will register the ranking of each course for the whole month (one registration per week). Each registration includes all the measurements from the time of the activation of the algorithm using an overlay between registrations. So, we now have:

$ERA_{TOT} = (ERA_{N-3} + ERA_{N-2} + ERA_{N-1} + ERA_N) \div 4$, where:

$ERAi$ takes value i=N for the period $\{-\infty < t \le t_0\}$, N-1 for $\{-\infty < t \le t_{0-1}\}$, N-2 for $\{-\infty < t \le t_{0-2}\}$ and N-3 for $\{-\infty < t \le t_{0-3}\}$

So we now have a balanced method of calculation of the scoring algorithm which disallows abrupt variations in cases of a typical interaction between user and system.

The implementation of the usage data collection system is done through the central parser of the application. This parser provides for the transfer of the user from the personalization interface to the course environment selected.

At the same time it collects usage data which can be analyzed by the algorithm. Specifically it stores dates, time, user id and course id in its own space on the database of the e-Class system. Therefore usage data tells us which user accessed which course.

We evaluated the algorithm by using 21298 events from the logging table of the database. The algorithm was programmed in java and used threads to control the program run time and CPU usage. Our purpose was to create a light algorithm that has minimum CPU and memory requirements of the web server. The algorithm has an initial peak when it fetches the logs for processing. After this, it is hardly noticeable. This was made possible since the algorithm is not time-critical and more logs can be added to the logging table as the algorithm is active.

A benchmarking experiment was conducted of this algorithm, using simulations of server balances in the form of eight different thread sleep times ranging from 5ms up to 40ms with a step of 5ms. For each thread sleep-time we performed 10 simulations and received the average duration. In figure 3 the results of the simulation are shown. The algorithm is almost linear and gives good performance results for the amount data processed.

6. Conclusions

In this paper the e-Class Personalized prototype has been presented. The prototype was tested for performance at the keystroke level, simulating improvement of user keystroke activity. An average improvement of 61% was measured according to this test. In addition, the ranking algorithm that adapts content presentation to the users' usage patterns was tested for performance A linear behavior of the algorithm in terms of server load, simulated through a thread sleep was demonstrated, while the absolute load was not found noticeable in typical server conditions. After the

reported tests, the new version of the e-class software was installed on the Network Centre of the University[1] serving over 20,000 users. In the first phase a limited number of users was exposed to this software. Through a focus group we measured their reaction to the new system and the first findings have been very positive. As we approach the end of the first year of operation of the prototype, we plan a more systematic evaluation using the logfiles of the system itself.

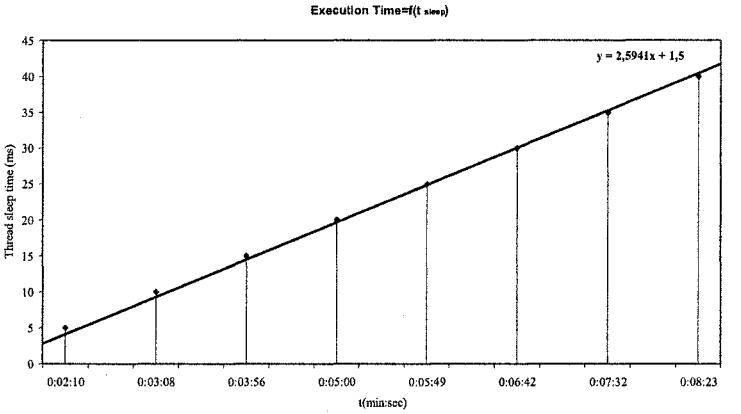


Fig. 3. Ranking algorithm execution time vs thread sleep time

References

1. Avouris N., Margaritis M., and Komis V., Modelling interaction during small-group synchronous problem-solving activities: The Synergo approach, 2nd International Workshop on Designing Computational Models of Collaborative Learning Interaction, ITS2004, 7th Conference on Intelligent Tutoring Systems, Maceio, Brasil, Sept. 2004.
2. Devedzic V. B., Key Issues in Next-Generation Web-Based Education, IEEE Trans. On Systems, Man and Cybernetics-Part C, Vol 33 (3), August 2003, pp. 339-349.
3. Brusilovsky P., "Adaptive and intelligent technologies for web-based education," Künstliche Intell., no. 4, pp. 19–25, 1999.
4. Virvou M. Automatic reasoning and help about human errors in using an operating system. Interacting with Computers, 11, pp. 545 – 573, 1999.
5. Avgeriou, A. Papasalouros and S. Retalis: Learning Technology Systems: issues, trends, challenges, proceedings of the 1st IOSTE symposium in Southern Europe, Science and Technology Education, Paralimni, Cyprus (2001)
6. Card S.K., Moran T.P., Newell A., The keystroke-level model for user performance with interactive systems, Com. of ACM, vol 23, pp. 396-410, 1980.

[1] Special thanks are due to the Network and Computing Center of the University of Patras for their support and in particular to Dr. V. Daskalou and A.G. Voyiatzis