Developing an Alarm Manager Based on Web Services [1]

Daniela Leal Musa, Marcel Weschenfelder, José Palazzo Moreira de Oliveira
Instituto de Informática - UFRGS
{musa, mweschenfelder, palazzo@inf.ufrgs.br}

Abstract

This paper describes an alarm manager aimed at monitoring students' activities in online courses over the Internet. The goal is to show an alarm manager to give more support to distance teaching, and help the teacher to have a thorough follow-up of distance students, thus minimizing the teacher's task of looking for information in long reports or complex graphs. Besides, the alarm manager also helps students during their study sessions, sending extra material, help when it considers necessary. The alarm manager proposed here was incorporated into the distance-teaching environment known as Claroline. The propodsed solution is an approach to identify the necessary requirements for alarms integration with web-based distance teaching environments situations.

1. Introduction

Today, most of the web-based distance learning environments have a great concern with the management of their courses, the display of instructional material and the availability of communication tools that foster the interaction between students and teachers. These environments lack mechanisms that would enable the teacher to perform a more complete and comprehensive follow-up of student's activities during courses. This could happen by the identification of the students' knowledge level and their learning pace.

In distance learning courses, the teacher's presence is not as effective as it is in presential courses. The evaluation of students' individual efforts is a great reason of teacher's concerns, especially when a great number of participants are involved in the course. Obtaining results that enable the assessment of the class as a whole is another worry, as in distance learning the teacher is not present to evaluate this aspect.

Usually, the process of student's evaluation in web-based courses is performed only through the analysis of information collected through tests, exercises or extensive reports that contain the log of pages accessed during the course. That is the reason why counting on appropriate tools to monitor what each student is performing is very important: problematic situations can be detected and the teacher is warned about each student's behavior. The teacher would have access to the particular features of each student, and would be able to customize classes to each one, making a closer follow-up and guiding the student during study sessions.

In some web-based teaching environments, Artificial Intelligence agents assess the student's behavior. An agent monitors the processes of interaction between students and the e-learning system, aiming at identifying the learning process of each student [1]. The higher the amount of information about each student, more suitable will be the agents' help. In the majority of teaching environments that use student's behavior monitoring techniques, the required data about student's data are extracted from an analysis of the distance learning server logs. Such records can provide the behavior of each student in what concerns files access (pages), which are stored in the server. However, it is necessary to perform several searches in these records to obtain information that are important and that can help to discover the student's behavior. A system of alarms can be used to perform such searches and warn the teacher about what is going on with some of them.

The goal of the present article is to show an alarm manager to give more support to distance teaching, and help the teacher to have a thorough follow-up of distance students, thus minimizing the teacher's task of looking for information in long reports or complex graphs. Besides, the alarm manager also helps students during their study sessions, sending extra material, notes or help when it considers necessary.

[1] The work described in this paper was partially supported by grants from CAPES.

The alarm manager proposed here was incorporated into the distance-teaching environment known as Claroline[2]. It was a way of identifying the necessary requirements for its integration with web-based distance teaching environments.

The present paper is organized as follows. Section 2 shows some studies that have been developed for students follow-up in web-based distance learning environments. Section 3 shows the alarm manager proposed, and Section 4 describes the integration with the distance-teaching environment named Claroline and the alarm manager. Section 5 brings some conclusions and future works.

2. Related works

In general, the student's behavior is assessed by using the agents' technology developed by Artificial Intelligence. Otsuka et al's [3] model is composed of interface agents. These agents learn through observation and monitoring of user's actions, they work as personal assistants, collaborating with the user and with other agents in the accomplishment of certain tasks. The model proposed by Otsuka is composed of three modules: (1) follow-up module, which tracks the student's interaction within the environment and generates reports about their participation; (2) module of behaviour analysis, which selects and presents data that can help in the analysis of student's performance in a specific activity; (3) validation module, which is responsible for the construction of the student's profile. Up to the present moment, no result has been shown about the efficacy of the model proposed, or if it has already been implemented or if it is under use.

The conBa system (CONcept Based) [4] is composed of interface agents able of monitoring and advising students during navigation in the instructional material. The system is composed of two main agents, the professor and the tutor. The professor agent offers the interface to the human-professors, who perform the following activities: (1) manages the course instructional material; (2) discovers, through the tutor agent, which topics the students are learning; (3) modifies the student's profile, informing the tutor agent that a given topic can be considered as acquired. The tutor agent keeps a description of the pages the student accessed, which are considered studied topics. These agents do not take decisions, and do not act in the student's favor. They only generate suggestions based on pages already visited by students and on the structure of the online course. Agents do not take into consideration the student's previous knowledge nor their style and learning pace. It would be interesting if the agent could indicate extra material when student's needs were identified, and not simply when the student reaches the module's end or accesses a specific page.

Some environments propose the use of web-server log files to collect students' and course's information. Niegemann [5] presents the EDASEQ tool (Exploratory Data Analysis for Sequential Data) to help in the analysis of student's navigation history. The tool tracks the web-server log file, converts data to an Excel file and generates graphs about the student's navigation. However, a human analysis is necessary to retrieve important information from these graphics. According to Niegemann, the tools were not used with real navigation data; fictitious data were produced to test the tool's functionality.

The monitoring model presented by Peled [6] is composed of a filter that extracts information from the information log considered not necessary, like the navigator used, the accesses generated with double clicks, accesses of teachers or course administrative personnel, administrative tasks, login, etc. Data resulting from these filters remain in text type files. This system was tested in a course with 200 students, however no tool had been developed to make visualization of these information easier. An exhaustive analysis in texts generated is necessary, making the process unfeasible.

Current studies have showed that the majority of techniques used offer final reports with access data and, in general, a human analysis is performed to collect important information from these reports. The teacher should search each student's data but is not aware of what is happening during the learning process. There is a need for other alternatives that would allow obtaining information about the student and the use of this information in the evaluation process.

3. Alarm system

A solution for the problem presented in section 2 is the use of an alarm system that monitors the processes of student's interaction with the course environment and generates a navigation history. The alarm system makes an analysis of this history. When certain situations are detected, the teacher is warned about what is going on with some students. Thus, the system is able to discover when a student stops at a certain point of the course, meaning he may not have understood the topic approached. Data from this analysis can also be used to help students in their study sessions, by stimulating and indicating extra material, exercises, etc. The coordinator of the instructional material can also benefit from an alarm system, because information of pages accessed and other about possible problems in the site structure may be sent to the coordinator. Figure 1 shows the architecture of the Alarm manager.

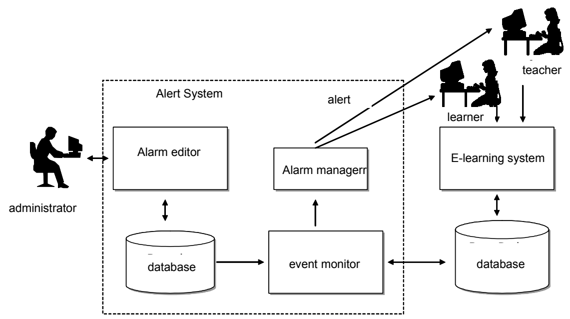

Fig.1. Alarm system architecture

The alarm system is composed of the following modules: alarm editor, events monitor, monitoring module, database, Log and alarm manager, which we present in details in the following sections.

3.1 Alarms creation and edition

Teachers or administrators use the alarm editor to define alarms used in the follow-up of student's activities. In this environment, each teacher can create alarms for their courses or select alarms that were already defined by other teachers. Figure 2 shows the user's interface for the definition of alarms in the current implementation of the system.

An alarm is composed of: (a) a triggering event, (b) one or more conditions that must be verified, and (c) an action that must be performed. Events are indicators of different situations, they can be two types: (a) environment's events, which are events the teacher or the student generates during their interaction with the teaching environment. Examples of this type of event are the student's *logon*, the end of some exercise or access to a certain page; (b) temporal events: some situations must be checked in a certain period of time, once a week, for example. This type of event depends on time and not on some situation that happened because of an interaction between the user and the environment.

An alarm is represented by ECA (event-condition-action) rules, that is, when an event takes place, a condition is checked and if it is true the action is performed. Rules are expressed as follows:

On event IF <condition> THEN<action>

When the alarm is created, the teacher selects the alarm-triggering event. In the current version of the system, the teacher cannot include a new event, and only chooses the events from a list of elements pre-defined by the alarm system.

Another information that must be informed at the moment of an alarm creation is the condition that will be tested at the moment the event is triggered. Today, the condition is informed to the system in the form of SQL queries, thus requiring the teacher has notions on the language and on the structure of the environment's data model. A solution that exports the teaching environment logic structure to the teacher and another form of the conditions representation is under study, so that the teacher can create his or her own alarms with no knowledge on SQL language.

The alarm action must be informed at the moment the alarm is created, and it can be a message in the navigator or in the e-mail. The message receiver, who can be the teacher, the student or the course administrator, must be informed. The type of action and the receiver are selected from a list previously defined, and it is not possible to insert a new action or receiver.

Data informed for the creation of an alarm are stored in the database shown in Figure 1. It is possible to make new searches, insertion and deletion of alarms through the Alarm editor. Any other teacher who participates in the system can use an alarm that has already been defined by a teacher.

3.2 Monitoring

The alarm system has a monitor of events that happen in the teaching environment database. The monitor consists basically of an interpreter of SQL commands. It receives an SQL command as parameter (Insert, Delete, Update) that will be performed in the teaching environment. The alarm server receives a warning from the monitor of events informing that some important event has taken place. After the warning, the alarm server verifies the conditions associated to that event. If the condition is true, the action indicated in the database of the alarm system is performed.

The alarm manager also stores the history of all alarms generated so that it is possible to perform statistics about the student's performance during the course.

4 Integration of the alarm system into the Claroline system

The alarm system we presented in the previous section was integrated into a distance-teaching environment so that the main requirements of integration of alarm systems into distance teaching environments could be identified. This section describes this integration in further detail. The alarm system was integrated into the Claroline environment, which is distance-teaching software developed at the UCL (*Université Catholique de Louvain*) and is open code. Currently, 500 students and 20 professors of the Computer Science Institute of Universidade Federal do Rio Grande do Sul are using the software. Professors use the environment to make instructional material, exercises, and tasks schedules available, as a complement to presential classes.

Claroline was chosen for special reasons. First, because the environment adopts the policy of GPL open code software, a very important requirement in our work. Second, Claroline was developed in PHP language and MySQL database, the same used in the implementation of the alarm system, what made the integration between systems easier. The environment's source code is easy to understand its structure has environment variables and functions libraries, what helps the addition of other software modules.

4.1 Monitoring

There is a monitor of events that occur in the teaching environment's database (Claroline). The monitor performs actions in the occurrence of a critical event. Figure 2 represents the Claroline structure integrated with the event's monitoring system.

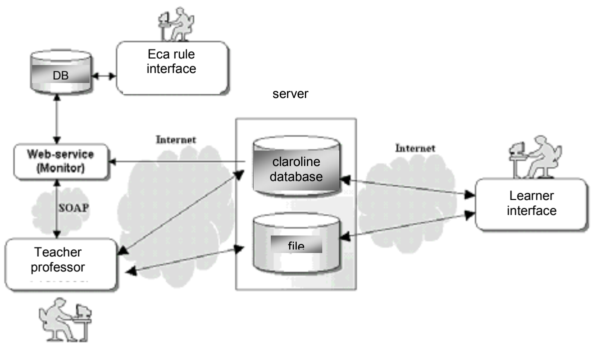

Figure 2 - Claroline integrated to the module of events monitoring

The monitor consists basically of an interpreter of SQL commands. It receives as parameters: an SQL command from Claroline, the user's ID and the course ID. With these parameters it is possible to identify the event (*INSERT, DELETE, UPDATE*) and the table where it is taking place. The monitor of events was implemented as a web service, offering a practical and efficient way for applications communication and data exchange over the Internet [7].

The communication between Claroline and the monitoring web service is carried out through SOAP (Simple Object Access Protocol); a protocol based on XML (Extensible Markup Language) and establishes a communication path among applications in different domains [7]. In the present study, SOAP uses an HTTP transfer protocol to send messages of remote functions request. HTTP facilitates the transportation of SOAP messages among systems because *firewalls* usually do not block the access to the HTTP port. The fact that XML messages are text-based justifies the possibility of communication among different platforms.

The Claroline system underwent small changes in order to call the monitoring web service. We highlight that the performance of the SQL command by Claroline remains the same. The monitoring web service only determines the action to be taken in case there is an ECA rule defined.

4.2 Implementation tests

In order to validate the alarm system integration with the Claroline environment, some alarms were defined and activated in the environment. Two types of actions were implemented: sending e-mails and calling of remote functions through a web service.

The first action consists basically of sending e-mail to all students of the same class about the scheduling of a new task. In Claroline, scheduling works as follows: the teacher accesses the system and schedules a test or homework. Scheduling data are stored in the Claroline database and can be seen when the student accesses the course's page. To make the teacher's work easier and help the student, the monitor of events identifies the addition of a new task in the scheduling table and sends e-mail to all students with the new item. This happens in a transparent mode to the teacher, who does not need to interfere in the process. The administrator registers the message that will be sent.

Besides, an action that calls a remote procedure, a web service, was also implemented. In order to test this functionality we used the model described in [8], which defines a central data repository that contains information about students from different distance teaching environments. It was necessary to develop an action that enabled Claroline to send its student's data to the repository, so a web service was created and named Notify_Data. The function implemented to perform this action is specific for events in the Claroline's table that stores the student's registers. Its main goal is to call the remote procedure named Notify_PersonalData from the *Web-service* defined by Musa. This procedure warns the repository that new data have been inserted in the

Claroline database and requests authorization to insert them in the repository.

During informal conversation with some students and teachers who used the system and received alarms, we noticed they found some alarms interesting and some inconvenient. Alarms that suggest content to be studied or exercises were considered positive. Alarms that send the students an e-mail reminding them to access the course after an idle period were considered inopportune.
The teachers suggested an alarm should be sent after a certain period of time, i.e., the alarms generated should be sent to the user gathered in a batch. This suggestion can be applied to some kinds of alerts.

5 Conclusions and Future work

The alarm system presented in this article was implemented as a generic module, so that it can be attached to other web-based teaching environments. Aiming at validating the use of this system in a e-learning system, it was integrated into the Claroline environment, which is being tested by a great number of users, students and professors, of the Computer Science Institute of UFRGS. An advantage was the choice for an environment developed within the policy of open code software. The entire source-code and the database could be fully accessed and changed whenever required. One of the main difficulties found during the integration was the fact that the alarm system needed the e-learning system to inform data about the teaching environment database. Besides, the alarm conditions are expressed as SQL queries, what constrains the use of the environment. A different way of solving these drawbacks is under investigation. The information of data regardless of the administrator's presence is under study too, as well as a language or editor that has conditions already defined to be used at moment of the alarms edition.

As the alarms are being generated in the Claroline environment, it is possible to capture data that are essential for the validation of this model both for student's activities follow-up and as a way of keeping them motivated in the course they are taking. Some results have already been obtained and were shown to be quite promising, being important indicators of the system's success. Now we are offering the possibility of alarms management for the full set of courses being teatched at II-UFRGS to gather a big amount of real experimental data. We hope this experience can give a step further on the contributions to make Distance Education a feasible alternative in the educational process.

6. References

[1] Viccari, Rosa. (1998) Artificial Intelligence and Educacional Systems. Journal of School n.2, Olivais, vol I, 1998.

[2] Claroline System. Available at: http://www.claroline.net

[3] Joice Lee Otsuka, Heloisa Viera da Rocha: An Agent-Based Approach to Support Formative Assessment. ICCE 2002: 1486-1487

[4] Roda, C. A multi-agent system for advising and monitoring students navigating instructional Web sites." In 6th International Conference on Information Systems Analysis and Synthesis (ISAS 2000*),* Special session on Cooperative and Distance Learning. Orlando, Florida. July 2000. 505-509

[5] Niegemann, H. (2002) EDASEQ – A *log* file analysis program for assessing navigation processes – Proceedings of the 8th International Conference on Computers in Education / International Conference on Computer-Assisted Instruction 2000. Taipei, Taiwan.

[6] Peled, A. (1999) Logging for Sucess - Advancing the Use to Improve Computer Mediated Distance Learning - Journal of Educational Computing Research. v. 21, n. 3, 1999.

[7] Tsalgatidou, A. and Pilioura, T., An Overview of Standards and Related Technology in Web Service. In Distributed and Parallel Databases, vol. 12, p. 135 – 162, ISSN:0926-8782, 2002.

[8] Musa, D.; Palazzo, J. M.O.; Sharing Learner Information through a Web services based Learning Architecture. IN WEB INFORMATION SYSTEMS MODELING, WISM, 2004, Riga, Latvia. CAiSE Workshops , Riga, 2004. p.122–131.