Synchronous support and monitoring in web-based educational systems

Christos Fidas, Vasilios Kapsalis and Christos Tranoris
Scientific Park of Patras, Instance Ltd, Patras, Greece, and
Nikolaos Avouris
Human Computer Interaction Group, University of Patras, Patras, Greece

Abstract

Purpose – The purpose of this paper is to present the design of a context-aware service for synchronous support in web-based educational systems.

Design/methodology/approach – The proposed service supports synchronous interaction among tutors and students, based on their current context, thus increasing the communication channels, the reasoning and the exchanging of ideas.

Findings – The proposed service introduced here gives additional value to distance learning educational systems, as it transforms the web from a medium primarily used for asynchronous communication into a synchronous one.

Originality/value – As educational institutes move their content online and open universities support lessons through the internet, services like the proposed one are of general value.

Keywords E-learning, Learning methods, Worldwide web

Paper type Technical paper

Introduction

The web has been used as a suitable medium for learning since its early days and is used widely for educational purposes. It is a medium, which can be accessed easily by tutors and students and it supports multiple representations of the educational content (Hites and Ewing, 1997; Carr-Chellman and Duchastel, 2000).

During the last few years a number of web-based environments have been developed and used to support the creation and maintenance of various learning scenarios, like Blackboard (2002), Quia (2005), ScribeStudio (2005) and Claroline (Lyn, 2005). These environments use carefully designed and multiple forms of media such as hypertext, graphics, real-time audio and video and other hypermedia objects (such as Java Applets and Macromedia Flash presentations) to improve presentation and involve students in active learning activities (Weston and Barker, 2001). Recently, some have tried to adapt the educational content to the learning needs of each individual user, based on the user's context with the aim to provide the student with a complete and up-to-date picture of the subject matter, depending on their preferences and knowledge level (Lonsdale *et al.*, 2004; Brusilovsky, 2001; Harrison and Bergen, 2000).

However, the role of students remains as passive receivers of the educational content. Furthermore, the communication channels among tutors and students remain asynchronous and are thus characterized by the lack of direct communication, which makes the achievement of common understanding and the exchanging and reasoning of ideas rather a difficult task (Powell, 2001; Stahl, 2002).

Campus-Wide Information Systems
Vol. 23 No. 3, 2006
pp. 138-148

1065-0741
DOI 10.1108/10650740610674157

Modern approaches in teaching and learning puts an emphasis on learning activities that involve collaboration among students and tutors, thus encouraging construction of knowledge and building of meaning (Dillenbourg, 1999; Fidas *et al.*, 2005). E-learning is not just concerned with providing easy access to learning resources, anytime, anywhere, via a repository of learning resources, but is also concerned with supporting synchronous and asynchronous collaboration among students and tutors. For web-based systems it is important to have services that are easy to use and install and which enhance the asynchronous communication with synchronous interactions among the involved users. In addition, by using such services the tutors are able to monitor in real-time the students' interactions with the educational content, while they are performing a learning task, investigating usability problems and providing synchronous support.

This paper focuses on the architecture, the main functionality and the characteristics of a service (Edu-smile) that makes synchronous support in web-based applications possible. Special emphasis is put on the context-awareness of the tutors and students involved. Finally, a prototype implementation of the proposed architecture is presented.

Defining synchronous support in web-based educational systems

In general synchronous communication based on the web includes two main spaces of interaction:

1. the discourse space where interaction occurs via audio, text or videoconferencing among the involved participants; and
2. the task space where interaction occurs through the learning material which represents the learning concepts (Dix *et al.*, 1998).

Common understanding among the participants is established when the participants share common concepts about the learning task, the learning objects and the used environment (Stahl, 2002; Baker *et al.*, 2001; Dillenbourg, 1999).

A former step in achieving a common understanding is the exchanging of ideas and the reasoning and discussion of various aspects on the learning task. In this context there should be the ability to provide the students with synchronous support by the teachers, arguing that live communication is always important for online learners who expect immediate assistance and response (Soller, 2001; Robertson *et al.*, 1998; Barros and Verdejo, 2000).

Synchronous support in a web educational environment is not just considered as the ability of synchronous communication among tutors and students. Furthermore, it is considered to be the ability of the system to provide students and teachers with the necessary information that is important for each other regarding the past and current status of the involved entities.

Defining context for synchronous support in web-based educational systems

Dey (2001) defines context as:

> Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.

Thus, if a piece of information can be used to characterize the situation of a participant in an interaction, then that information is context. Context may consist of both static and dynamic characteristics that correspond to the current state of the involved entities.

Use of scenarios of typical context-enabled applications have led to the identification of the following recurrent challenges:

- it must be abstracted in order to make sense for the application;
- it may be obtained from multiple distributed and heterogeneous sources; and
- it must be dynamic: changes in the environment must be detected in real-time and applications must adapt to constant changes.

The development of context-aware applications has an increased difficulty imposed by the fact that the application must first obtain the contextual information and then use it to improve functionality as well as results (Christian, 2004).

The advantage in the proper use of contextual information is that the goal of the system is achieved in better terms. A metric of the success of the system can be constructed in terms of load per tutor expertise ratio (which assures a balance between tutors' load, defined as the number of concurrent assistance sessions); rate of learning (the speed of assimilating content adjusted to the students' capabilities; the relevant content for their profile) and throughput (defined as a quantitative measurement of improved knowledge transferred from the tutors to the students by taking into account their expressed preferences – e.g. through ranking).

Context information history is valuable, as shown by context-based retrieval applications. A dynamic and historical model allows applications to fully exploit the richness of context information. In our approach, we distinguish three different entities: student, tutor and content as is shown in Figure 1.

For each of the three entities depicted in our system the data constituting the respective entity's context can be classified as follows:

- static information that has a very small rate of changing or updating during time (static context); and
- dynamic information that is characterized by a high volume of data and a high rate of changing and updating during time (dynamic context).

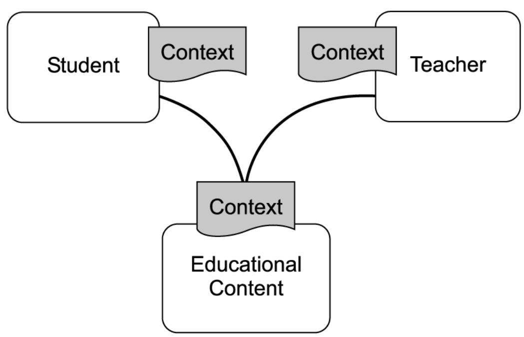

Figure 1.
The general structure

Context definition for students

Students have access to a set of educational content and a list of tutors for online help, based on the student's profile coming up from their static and dynamic context.

The static context includes the data which describe the personal information, preferences and the educational level of the student whereas the dynamic context consists of relevant information, which is being captured during each student's session automatically and corresponds to the student's current activity.

The dynamic context includes the location, specific content that the student currently accesses, information about the visits to each relative resource, the sequence of visited pages, time spent on each page and the score achieved at the self-assessment tests. Also, the history of the previous activities (e.g. data about online help sessions) constitutes part of the dynamic context and is updated after the end of each student's session.

Context definition for tutors

Tutors, in a similar way to the students, have a static context, consisting of their area of expertise, their preferences concerning the type of help offered, etc. and a dynamic context, which corresponds to their current status, including the number of concurrent help sessions offered by the tutor and the relevance of his expertise with the student's needs. Furthermore, the evaluation of each tutor's performance by the supported students (ranking) constitutes part of the dynamic context.

Context of the educational content

In general, the educational content is oriented around learning objects that consist of anything that can be described by a Universal Resource Identifier (URI), such as web pages, PDF files, etc. These objects can be modified and combined at any time by the tutors, in order to form customized learning courses, adapted to the needs of each specific group of students. Moreover, each sub-topic of the educational content includes tests with questions of several types for assessment of the students' knowledge level. The static part of the content's context includes all the relevant information regarding the thematic topic and the sub-sections that the content belongs to. Regarding the dynamic part of the context, it is defined as the content's current state, including the student's previous interactions (history) with the learning objects. This information is important for providing context-awareness, like What You See Is What I See (WYSIWIS) to the tutors.

Table I illustrates a summary of both types of context for the three involved entities that constitute our model.

System architecture

The system architecture follows the application service provider model. As shown in Figure 2 we distinguish two user categories: on the left side there are the students who are visiting the web site whereas on the right side there are the tutors or the web site representatives who are trying to assist the students in order to accomplish their educational tasks. Furthermore, the service consists of the following entities:

- a number of distributed agents (student and tutor agents) that perform the exchange of messages regarding the changes of the contextual information;

- a central server who is responsible for the collection and processing of incoming messages through defined modules which are explained in the next paragraph; and
- a monitoring environment which enables the synchronous communication among the tutors and the students as well as the synchronous monitoring of the users actions.

The installation of the service is easy to accomplish as it requires only some modifications in the source code of the existing web pages. The functionalities of the main modules of the central server are explained below.

Table I.
The context types for each entity

	Context type	
Entities	Static	Dynamic
Students	Personal information Preferences Educational level	Location Currently accessed content Visits to each relative resource Time spent on each page Score at the self-assessment tests Previous help sessions (history)
Tutors	Area of expertise Preferences (e.g. type of offered help)	Current status (e.g. concurrent help sessions) Relevance of the tutor's expertise with each student's needs Tutor's ranking by the students
Content	Thematic topics Sub-sections Tests	Detailed interaction history between learning objects and students

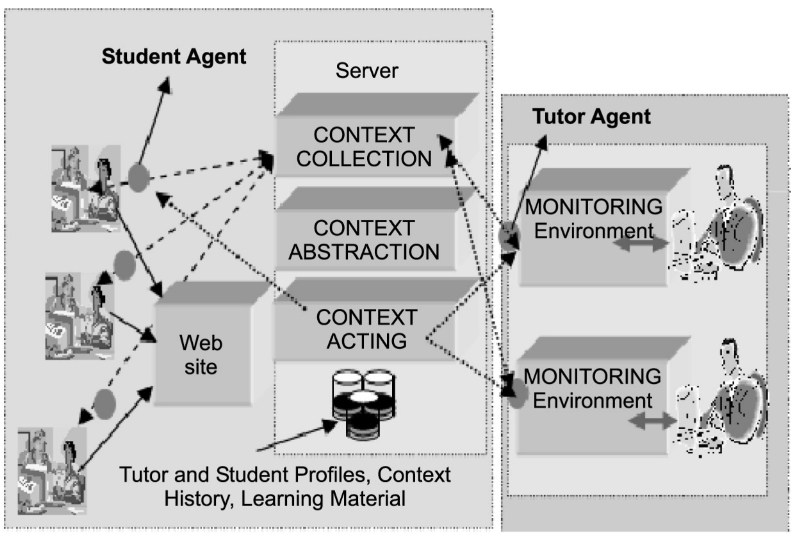

Figure 2.
The system architecture

Context collection module

The first module is responsible for the detection of any changes to the current status of any entity (students, tutors, context) that contributes to the context. The entities' attributes that constitute the context are determined at the initial stage and are specific to each educational scenario. The detection of any context change is based on an event-based mechanism, which involves distributed agents residing on the students' and tutors' application interfaces and a server-side collector module, which uses simple access object protocol messages for their communication. Any relevant status changes to the students and tutors are detected by the students' and tutors' agents, respectively, which transmits these events to the server-side collector module. As shown in Figure 3, the interface exposed by the server-side collector module is based on the web services description language and in effect, it is a web service. This module has the responsibility to transform the received data into a consistent collective data format and to forward this data to the next module.

Context abstract representation module

This module generates a unified abstraction of the real world by converting the received data to a set of resource description framework statements, which it stores at the central repository. Thus, a modeling of the interpretation of the context representation in terms of the semantics takes place. In order to define the context in the modeling level, an appropriate marking-up with a suitable metadata schema for each involved item that constitutes the context has been defined. Examples of metadata for our learning objects, includes, "author" "difficulty" "ranking", etc. Furthermore, the user's current status is combined with their history. Thus, semantic capabilities are added, allowing software agents to reason about the data and produce intelligent answers to each user's specific requirements and context.

For example, the number of simultaneously supported students by a teacher is a value that can affect whether a new students request for synchronous support will be assigned to that teacher or not. Furthermore, the evaluation of a teacher can be an indicator of his capability for supporting the students in an acceptable way.

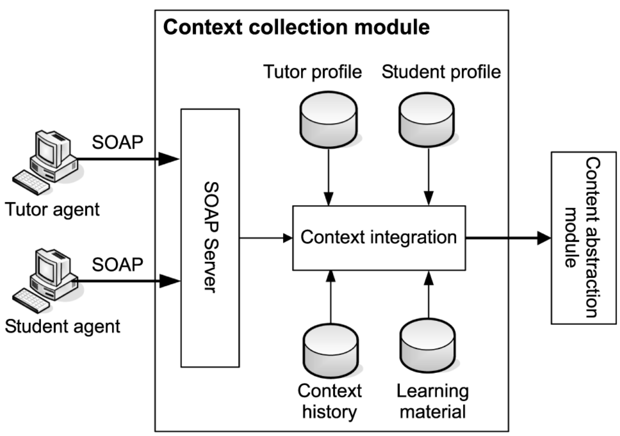

Figure 3.
The context collection module

Context-controlled acting module

This module is responsible for the initiation of the corresponding actions, as soon as an event (or a combination of events) is being detected. The events can be either user-triggered, like the initiation of a student request for assistance, or system-triggered, based on context rules, specified by the tutors.

Context rules can be related to both static and dynamic context, such as the location of the student (like the domain, country or IP) or they can be related to data that determines the current context of the student, such as the time he spends on a specific page, if he has visited a specific URL, as well as a specific sequence of page visits. The verification of any context rule that may be configured for each student separately, results into the notification of the corresponding tutor(s). Also, tutors may be notified about the beginning/ending of any student's learning session. Furthermore, the mechanism that is responsible for the initiation of actions implements a matching process, which selects tutors for synchronous (online) help during the students' learning session, based on the current context.

Tools to support synchronous communication and monitoring

The service provides facilities for synchronous interaction between tutors and students engaged in web-based learning scenarios. A number of tools have been developed with the aim to support the tutors, in order to provide synchronous help to web application students. Important aspects are team-awareness, multiple chat session availability and WYSIWIS functionalities. A typical user view of the tutor's environment is shown in Figure 4.

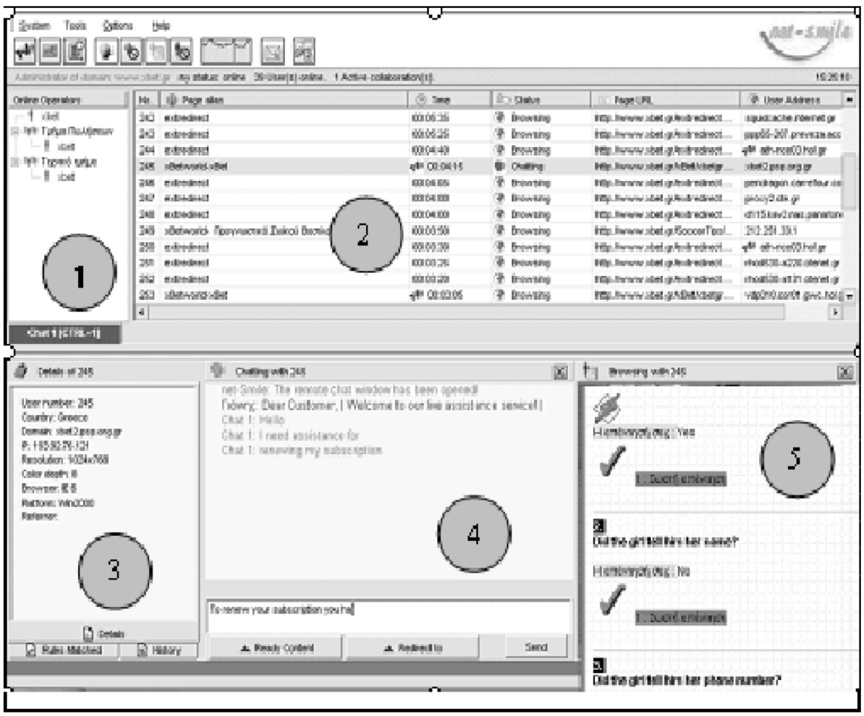

Figure 4.
Typical view of the tutors monitoring environment

Area 1 represents the presence awareness of the tutors who are logged in the system and are able to provide help to the students. Area 2 gives an overview of the students who are on the web site and are involved in various learning tasks. Every line represents a student. By selecting a student, the bottom part of the window (Area 3) will be updated with information concerning the selected student. Area 3 displays the context data of the current selected student. Details are comprised of three pages: static details like resource capabilities, dynamic details like rules matched (appears only if the student matched at least one rule) and history details where the students' interaction history can be seen. Area 4 displays the chat space, which is available to the tutor while he provides real-time help to one or more students. The chat space supports free and predefined phrases and allows the tutor to redirect the student to another tutor who is online in case his expertise is not enough. In order to have absolute awareness on the chat sessions in this area, tabs are displayed with active conversations. When a message is received and the user is not selected, then the corresponding tab will blink. Area 5 displays the "BrowseWithU" functionality, which allows the tutor to watch exactly the same page the student is currently browsing, supporting the WYSIWIS functionality.

A typical scenario of use

A typical interaction scenario of both tutors and students with the proposed service involves the steps outlined in this section. While a student interacts with the educational content of a web page he is being informed about the existence of tutors who are online and can meet the criteria for providing assistance to him (Figure 5).

These tutors have been selected by the service as the best match to the student's context history and the current status of his learning task. This process takes place if the student has sent a request for online assistance. Optionally, the service may suggest the list of the proposed tutors, even though the student did not make a request. The returned information consists of the tutors' names, their subject of expertise and their ranking which is calculated from the evaluation done by the students at the end of the given support as is shown in Figure 6.

As soon as the student decides which tutor is the most appropriate to provide assistance to him, he makes a request to the specific tutor for an assistance session. Then the tutor is informed, through his monitoring environment, that a new assistance session is being requested. The tutor is also informed about the student's current

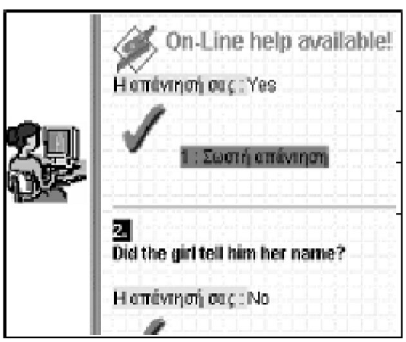

Figure 5.
A student's typical view

List of On-line Tutors

Name	Profile	Ranking	Chat Now
Bastian	C++	excellent	click here
Christos	Java	middle	click here

Figure 6.
List of available tutors

context, as well as about the history of his previous interactions. If the tutor accepts the request then an assistance session begins.

As soon as the assistance session finishes the student is asked by the system to evaluate the tutor's performance and his capability to answer the student's questions and to provide the appropriate assistance to him. As can be seen in Figure 7 the tutors are evaluated in a range from one to six where one means the lowest performance and six the highest.

Conclusion

An innovative service enabling synchronous communication in web-based educational systems has been introduced in this paper. The proposed service gives additional value to distance learning educational systems, as it transforms the web from a medium primarily used for asynchronous communication into a synchronous one.

The support of synchronous interaction among the tutors and the students, based on the current context of all the involved entities, is one of the innovative features of the proposed service.

Despite the capability provided to the students for getting real-time assistance, this service can be a powerful tool for the tutors, too. It gives additional value to their relationship with the students and it allows discovering usability and navigability issues of their web-based educational material, while monitoring the students' interactions in real-time.

As educational institutes move their content online and open universities support lessons through the internet, services like the proposed one are of general value. Edu-smile provides enhanced collaborative capabilities and opens a new area for further study in web-based educational systems.

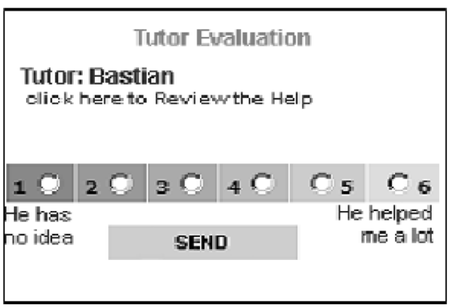

Figure 7.
Tutor evaluation by the student

References

Baker, M.J., de Vries, E., Lund, K. and Quignard, M. (2001), “Computer epistemic interactions for co-constructing scientific notions: lessons learned from a five years research programme”, in Dillenbourg, P. and Eurelings, A. (Eds), *Proceedings of the Euro Computer Supported Collaborative Learning, Maastricht, March 22-24*, pp. 89-96.

Barros, B. and Verdejo, F. (2000), “Analysing student interaction processes in order to improve collaboration. The DEGREE approach”, *International Journal of Artificial Intelligence in Education*, Vol. 11, pp. 221-41.

Blackboard (2002), *Blackboard Instructor Manual*, Blackboard Inc., Washington, DC, available at: www.blackboard.com

Brusilovsky, P. (2001), “Adaptive hypermedia”, *User Modeling and User-Adapted Interaction*, Vol. 11 Nos 1/2, pp. 111-27.

Carr-Chellman, A. and Duchastel, P. (2000), “The ideal online course”, *British Journal of Educational Technology*, Vol. 31 No. 3, pp. 229-41.

Christian, V. (2004), “Persuasive computing: towards self-aware systems”, *The Proceedings of the 2nd Workshop on Applications of Wireless Communications, Acta Universitatis Lappeenrantaensis, Lappeenranta University of Technology*, pp. 12-18.

Dey, A.K. (2001), “Understanding and using context”, *Personal and Ubiquitous Computing Journal*, Vol. 5 No. 1, pp. 4-7.

Dillenbourg, P. (1999), “What do you mean by ‘collaborative learning?’”, *Collaborative Learning: Cognitive Computational Approaches*, Pergamon Press, Oxford, pp. 1-19.

Dix, A., Finlay, J., Abowd, G. and Beale, R. (1998), *Human-Computer Interaction*, Prentice-Hall, Englewood Cliffs, NJ.

Fidas, C., Komis’, V., Tzanavaris, S. and Avouris, N. (2005), “Heterogeneity of learning material in synchronous computer-supported collaborative modelling”, *Computers and Education*, Vol. 44 No. 2, pp. 135-54.

Harrison, N. and Bergen, C. (2000), “Some design strategies for developing an online course”, *Educational Technology*, Vol. 40 No. 1, pp. 57-60.

Hites, J. and Ewing, K. (1997), “Designing and implementing instruction on the world wide web: a case study. Center for Information Media”, available at: http://lrs.stcloud.msus.edu/cim/cimindex.html

Lonsdale, P., Baber, C. and Sharples, M. (2004), “A context awareness architecture for facilitating mobile learning”, in Attewell, J. and Savill-Smith, C. (Eds), *Learning with Mobile Devices: Research and Development*, Learning and Skills Development Agency, London, pp. 79-85.

Lyn, A. (2005), *Claroline 1.3 Teaching Manual*, available at: www.claroline.net/dlarea/claroline_manual_teacher_english.doc

Powell, G. (2001), “The ABCs of online course design”, *Educational Technology*, Vol. 41 No. 4, pp. 43-7.

Quia (2005), “Quia web presentation”, available at: www.quia.com/company/quia-presentation.pdf

Robertson, J., Good, J. and Pain, H. (1998), “BetterBlether: the design and evaluation of a discussion tool for education”, *International Journal of Artificial Intelligence in Education*, Vol. 9, pp. 219-36.

ScribeStudio (2005), *ScribeStudio User Guide Version 2*, ScribeStudio, New York, NY, available at: www.scribestudio.com

Soller, A. (2001), "Supporting social interaction in an intelligent collaborative learning system", *International Journal of Artificial Intelligence in Education*, Vol. 12, pp. 40-62.

Stahl, G. (2002), "Introduction: foundations for a CSCL community", in Stahl, G. (Ed.), *Computer Support for Collaborative Learning: Foundations for a CSCL Community*, Laurence Earlbaum Associates, Hillsdale, NJ, Proceedings of CSCL 2002, Boulder, CO, pp. 1-2.

Weston, T. and Barker, L. (2001), "Designing, implementing, and evaluating web-based learning modules for university students", *Educational Technology*, Vol. 41 No. 4, pp. 15-22.

Corresponding author
Christos Fidas can be contacted at: fidas@ee.upatras.gr