Edu-Smile: A context–aware service for synchronous support in web–based educational systems

Vassilis Kapsalis, Christos Fidas[1], Christos Tranoris[1] & Adrian Stoica[1]
[1]*Instance ltd, Patras Scientific Park*
GR 26504, Platani Patras, Greece
kaspalis@isi.gr, fidas@instance.gr, tranoris@instance.gr, stoica@instance.gr

Abstract

In this paper a context–aware service for synchronous support in web–based educational systems is proposed. The main purpose of the service is to support web-based learning scenarios by allowing synchronous communication and interaction among the tutors and the students, thus increasing the communication channels among them. Furthermore, the service provides a variety of tools that allows the tutors to provide real–time support to the students, discover usability and navigability problems and monitor their actions while they interact with the web-based educational environment.

1. 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, like multimedia and/or hypermedia.

During the last years a number of tools has been developed and used that support the creation and maintenance of various learning scenarios. Recent learning systems are trying to adapt the educational content to the learning needs of each individual user, based on the user's context [1]. However, the role of students remains as passive receivers of the educational content. Furthermore, the communication channels among the tutors and the 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.

Modern approaches in teaching and learning put emphasis in learning activities that involve collaboration among students and tutors, thus encouraging construction of knowledge and building of meaning [2], [3]. 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. Especially for web-based systems it is important to have easy to use and install services that 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, investigate usability problems and provide real-time help.

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 in the context-awareness of the tutors and students involved. Furthermore, a prototype implementation of the proposed architecture is presented.

2. Context definition for synchronous support in web-based educational systems

Dey [4] 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.

Usage scenarios of typical context-enabled applications found in the literature have led to the identification of the following recurrent challenges:

(a) It must be abstracted in order to make sense for the application.
(b) It may be obtained from multiple distributed and heterogeneous sources.

Proceedings of the Fifth IEEE International Conference on Advanced Learning Technologies (ICALT'05)
0-7695-2338-2/05 $20.00 © 2005 IEEE

(c) 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 [5], [6].

The advantage in the proper usage of the 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 (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 and 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 [7]. 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 it is shown in figure 1.

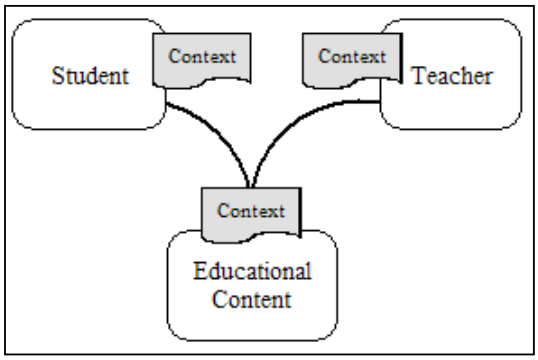

Figure 1. The General structure

For every of the three entities depicted in our system the data constituting the respective entity's context can be classified as follows: a) static information that has a very small rate of changing or updating during time (static context) and b) dynamic information that is characterized by a high volume of data and a high rate of changing and updating during time (dynamic context).

2.1. Context definition for students

Students, based on their personal needs and preferences (static context) and their current status (dynamic context) have access to a set of educational content and a list of tutors for on-line help. The static context is the information extracted by the student's profile (educational level, preferences, etc.). The dynamic context consists of relevant information, which is being captured during each student's session in an automatic way and corresponds to the student's current activity. This information includes the location, specific content that the student currently accesses, information about the visits to each relative resource, the sequence of pages he visited, 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 on-line help sessions) constitutes part of the dynamic context and is updated after the end of each student's session.

2.2. 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 offered help, 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.

2.3. Context of content

In general, the educational content is oriented around learning objects that consist of anything that can be described by a Universal Resource Locator (URL), 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 WYSIWIS to the tutors.

3. System architecture

The system, as it is illustrated in figure 2, consists of a number of distributed agents (student and tutor agents) that perform the exchange of messages regarding the changes of the contextual information.

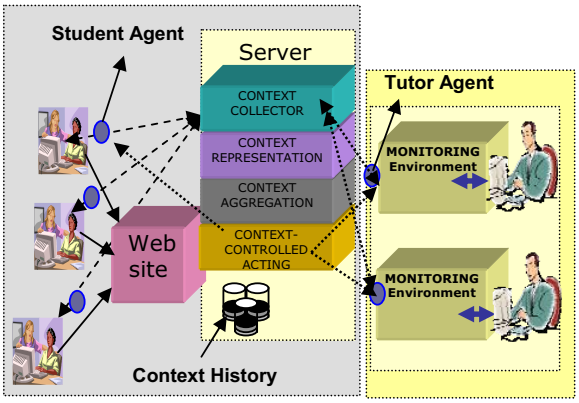

Figure 2. The System architecture

These messages are collected in the central server of the service where further processing is established through the modules explained below.

3.1. Context collection module

The first module is responsible for the detection of any change of the current status at any entity (students and tutors) 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 use the Simple Access Object Protocol (SOAP) [8] for their communication. Any relevant status changes at the students and the tutors are detected by the students' and tutors' agents, respectively, which transmit these events to the server-side collector module in the form of XML-formatted messages. The interface exposed by the server-side collector module is based on the Web Services Description Language specification (WSDL) [9] and in effect, it is a Web Service, consisting of a number of methods (operations) for receiving context updates. 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.

3.2. 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 (RDF) [10] statements, which it stores at the central repository. Thus, a modelling of the interpretation of the context representation in terms of the semantics takes place. In order to define the context in the modelling level, an appropriate marking-up with a suitable metadata schema for each involved item that constitutes the context has been defined. Example metadata for our learning objects, include, "author", "difficulty", "ranking", etc. The student/author profiles are described as vCard Objects [11], by using semantics such as "vCard:FN" (full name), "vCard:ROLE" (distinguishes between students and tutors), etc. 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.

3.3. Context aggregation module

Elements of the context can be aggregated to form meaningful pieces of information that has a higher value and can help the system to decide for certain behavior. For example the typing speed of a tutor correlated with the internet connection speed and the average number of help sessions initiated and served can give an indicator that an incoming session should be given or not to him. The ranking a certain student gives to a certain tutor in several sessions can be abstracted to a preferential relation between the student and the tutor. Assigning to the student a preferred tutor whenever is possible will lead to an increased amount of knowledge transferred between them than a random assignment of the tutor.

3.4. 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 the location of the student, like the domain, country or IP or they can be related to data that are determining the current context of the student, like 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 (on-line) help during the students' learning session, based on the current context.

4. 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 and multiple chat sessions availability, WYSIWIS (What You See Is What I See) functionalities. A typical user view of the tutor's environment is shown in figure 3.

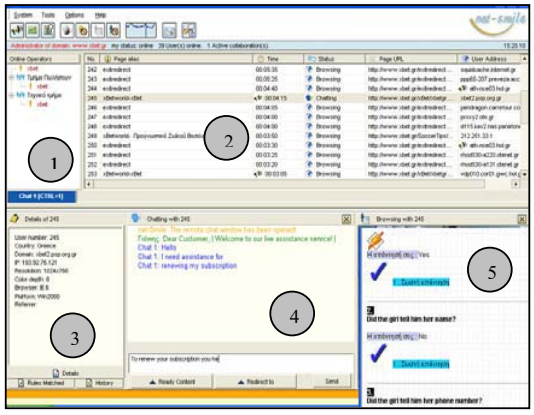

Figure 3. The tutor's monitoring environment

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, then 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 (optionally, 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 What You See Is What I See functionality.

5. A typical interaction scenario

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 on-line and meet the criteria for providing assistance to him (figure 4).

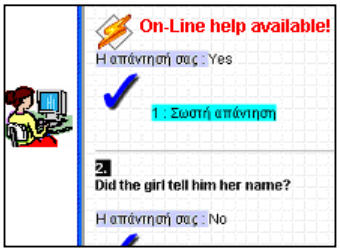

Figure 4. A student's typical view

These tutors have been selected by the service as a 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 on-line 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 it is shown in figure 5.

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 being informed, through his monitoring environment, that a new assistance session is being requested. Also, the tutor is being informed about the student's current context, as well as about the history of his previous interactions. If the tutor accepts the request then an assistance session begins.

Figure 5. List of available tutors

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 it can be seen in figure 6 the tutors are evaluated in a range from one to six where one means the lowest performance and six the highest.

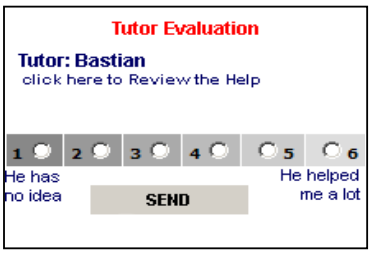

Figure 6. Tutor evaluation by the student

6. Conclusions

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 between the tutors and the students, based on the current context of all the involved entities, is one of the innovative features of our 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 on-line 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.

7. Acknowledgement

Edu-Smile combines the functionality of a generic platform (Net-Smile, http://www.net-smile.com), which permits live help over the web, with an integrated environment (WebTest) for the management of web-based student classes, developed by the tuition school "Komvos" (http://www.e-komvos.edu.gr).

8. References

[1] Lonsdale, P., Baber, C., Sharples, M. (2004), A Context Awareness Architecture for Facilitating Mobile Learning. In J. Attewell & C. Savill-Smith (eds.) Learning with Mobile Devices: Research and Development, London, Learning and Skills Development Agency, pp. 79-85.

[2] Dillenbourg P., (1999). What Do You Mean By "Collaborative Learning?". (pp. 1-19), Collaborative Learning: Cognitive Computational Approaches, Pergamon, 1999.

[3] Fidas (Ch.), Komis (V.), Tzanavaris (S.), Avouris (N.), "Heterogeneity of learning material in synchronous computer-supported collaborative modelling", Computers and Education, 2005, 44 (2), 135-154.

[4] Anind K. Dey , Understanding and Using Context, Personal and Ubiquitous Computing Journal, Volume 5 (1), 2001, pp. 4-7.

[5] Supporting Context Awareness in Highly Dynamic Network Environments, Wolfgang Beer, PHD thesis, Johannes Kepler University, May 2004.

[6] V. Christian, Pervasive Computing: Towards Self-Aware Systems, The proceedings of the 2nd Workshop on Applications of Wireless Communications, Acta Universitatis Lappeenrantaensis, Lappeenranta University of Technology, pages: 12-18, August 2004.

[7] Salber D. et al, The Context Toolkit: Aiding the Development of Context-Enabled Applications, in the proceedings of CHI'99, Pittsburgh, PA, May 15-20, 1999, ACM Press.

[8] SOAP Version 1.2 Part 1: Messaging Framework, http://www.w3.org/TR/soap12-part1/

[9] Web Services Description Language (WSDL) 1.1, http://www.w3.org/TR/wsdl

[10] Resource Description Framework (RDF), http://www.w3.org/RDF/

[11] Representing vCard Objects in RDF/XML, http://www.w3.org/TR/vcard-rdf