Proceedings of the 5th WSEAS Int. Conf. on APPLIED INFORMATICS and COMMUNICATIONS, Malta, September 15-17, 2005 (pp319-323)

Integrated Service Provision for Student Support

SEBASTIANO PIZZUTILO, BERARDINA DE CAROLIS, GIOVANNI COZZOLONGO,

VINCENZO SILVESTRI

Dipartimento di Informatica
Università di Bari
Via Orabona 3, 70126, Bari
ITALY
http://www.di.uniba.it/intint

Abstract: - In the last year Universities have developed systems for providing information and online services to their users. In most of the cases these services are realized by different systems and are limited to information about courses, lectures, timetable and logistic. Moreover, universities are promoting many initiatives and are developing many systems supporting e-learning and e-work. This paper discusses the design and implementation issues of a system in which many services are integrated to realize a Virtual University platform able to support students with mobile learning services.

Keywords: multiagent system, human computer interaction, e-learning.

1 Introduction

In this paper we present the architecture of a MultiAgent System (MAS) developed to integrate different services that a Virtual University can provide to its users, the students. Moreover it will support teachers and other employees in their work.

The motivation of this work concerns the possibility of having all these services always on line, and integrating them in order to reduce redundancy and conflict problems in the managing of information. Virtual University services can provide information about virtual and real courses, but also support about issues related to the student life, such as location of offices, deadlines, documents.

Another important motivation comes from the directive of the Italian Universities that aims at promoting interventions for improving mentoring and orientation services. This directive aims at following students during the entire course of study, giving them personalized suggestions about different topics and helping them to take decisions about their formative process [1]. In our Department, following this directive, has been instituted a tutoring program. In this program, a group of first year students is assigned to a teacher that follows each of them until their graduation. During this period they have to assist the students, giving them suggestion about courses and orientation choices, or helping them when they need. This role has the main aim to find out and solve those problems that prevent students from going on in their studies, and to make them more participative in their formative process.

However, we noticed that this service was not fully exploited by students and we started to investigate about the reasons related to this phenomenon. In order to assess the real motivations, we made a user study consisting in a questionnaire (150 subjects in total) aiming at understanding which was the user's expectations about the tutor role and at assessing why students were not using this orientation service. Analyzing the answers to questions about this last issue, it came out that the main reason was related to the difficulty to find the tutor always available when needed (professors teach, do research, are involved in meeting, etc.) and also to the shyness of some student that are afraid of asking questions about their problems to professors.

Taking into account these motivations we designed a Virtual University MultiAgent System (VU-MAS) able to support students during the entire course of study.

In particular, in this paper we focus on the description of the system architecture and, in particular, on end-users interaction aspects.

The paper is structured as follows: Section 2 gives an overview of the System. Section 3 shows how this architecture has been implemented. Finally conclusions and future work directions have been discussed in Section 4.

2 Architecture Overview

The Virtual University MAS is composed by agents that provide information services, and agents that represent the users. Agents related to an information service can answer to request of other agents giving information about a specific topic. Every agent is related to a specific service, like administration office or a library, or to a teacher.

Proceedings of the 5th WSEAS Int. Conf. on APPLIED INFORMATICS and COMMUNICATIONS, Malta, September 15-17, 2005 (pp319-323)

There are three type of agents (Fig. 1): the first is the personal virtual tutor (PVT), to interact with students; the second type is the agent of the tutor (VT), the third type can provide information and services on response of a request (service agents SA). The agents that represent teachers belong to the same type, but they can provide specific service for e-learning , like online exercitation, or lessons.

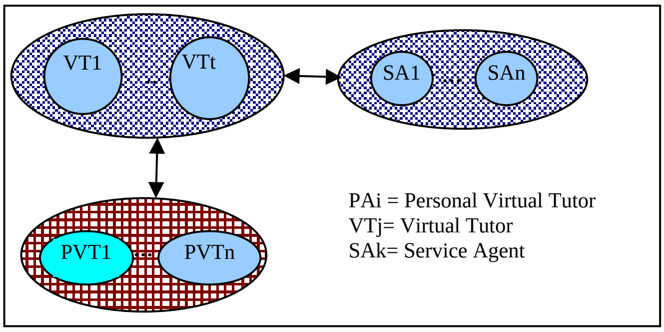

Fig. 1 VU-MAS architecture

In our system each student can interact with VU-MAS using a personal agent, the Personal Virtual Tutor (PVT), that monitors the student activities, follows his/her improvements, selects useful material according to the recognized student goals and provides him/her with useful suggestions. It combines e-learning capabilities with mobile computing, thus realizing an m-learning experience where the student can feel always in touch with his advisor. The PVT runs on a handheld computer and uses an ECA, that offer people the possibility to relate with computer media at a social level [2]. According to research and evaluation studies in the field of intelligent interfaces, ECAs [3] have shown to be a good interaction metaphor when acting in the role of counselors [4], personal trainers [5], or healthy living advisors [6].

The PVT can handle not only information about lessons and exercises sessions, but only about timetable, calendar of examination, scheduling of teachers. It can manage information from a wide range of service provider, such as secretariat, library, etc. The interactions with those services is mediated by the virtual tutor agent, which act as a proxy. When a PVT is on line, connected to VU-MAS, it contacts the agent of the student tutor, the Virtual Tutor, passing the user profile; this agent activate its tasks, and provides information on the basis of the user model received by the personal agent. When the student asks some information about an examination or a course, the tutor looks for information asking to other agents. If it is a request about administration issues it will look for service agents providing this kind of information and then will ask for the request. Then it will give the answer back to the personal agent.

When the connection is not possible, or the user has disconnected from the system, the PVT can be used to access contents of lessons suggested by the virtual tutor, such as presentations, documents, exercises.

2.1 Student interaction with the personal agent

An ECA is the metaphor that we have chosen as interface between the student and his/her advisor-agent. It is based on the architectural schema developed in the context of the MagiCster project. In this project, the ECA was intended as an entity made up of two main components, a '*Mind*' and a '*Body*', interfaced by an I/O language, so as to overcome integration problems and to allow their independence and modularity.

During the interaction, the Agent's Mind has the purpose of deciding what to communicate, by considering the dialogue history, the conversational context and its own current cognitive state. The output of the mind module is an **APML** (Affective Presentation Markup Language) specification of the meaning to be communicated [7].

The Body has the purpose to interpret and to render this output at the surface level, according to the available communicative channels: different bodies may have different expressive capabilities and therefore may use different channels. This approach was mainly driven by the definition of *communicative function* in Poggi et al. [8]. In their theory, a communicative function is a (**meaning**, **signal**) pair, where the **meaning** item corresponds to the communicative value of the **signal** item. For instance, a smile can be the signal of a "joy" emotion.

Therefore, the Mind should convey only the meanings associated with the act to be communicated and the Body should interpret these meanings and render them into an expressive behaviour according to the interaction context.

In the first version of the architecture, the APML interpretation process was embedded in the Body Player [9]. But, since the Agent's believability is strictly related to features such as its personality and role and the cultural and social context, separating the way in which the Body will render that meaning enforces adaptivity .

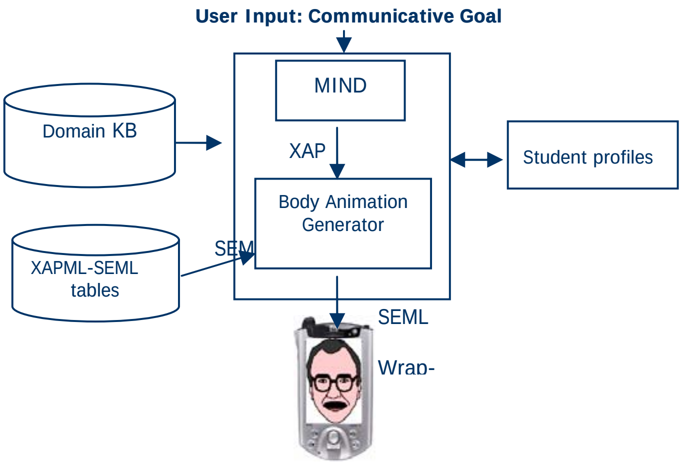

Fig. 2 Mind-Body Architecture

Mobile devices places limitations on the functionality of an embodied conversational agent since their computational and display capabilities are limited. Let's see now how this architecture structure has been adapted for the development of an ECA that runs also on an handheld computer platform (Fig. 2).

2.2 Agent's Mind

The task of the mind is to generate the specification of the Agent's move at the meaning level. Planning is a task hard to compute, then it is not feasible to have a planner running on an handheld computer, moreover the PVT has to run on and off-line. For this reason, we adopted a schema based approach. Given a communicative goal, derived by an interpretation of the user input, the Agent's Mind selects from its plan library the schema or a combination of schemas that allow to satisfy that goal. The result is an XAPML (eXtended APML) specification of meanings to be communicated. This extension of APML includes the following information that were not present in the first version: i) besides the specification of what the agent has to communicate it is possible to specify the information **background** (access to more details, visualization of relevant domain objects, etc.); ii) the ***focus*** attribute of the <performative> tag allows to establish which is the communicative focus and to update the student model with information about what has been communicated to the student; iii) the ***voice*** attribute has been added since, in case the student interact through its handheld device, the voice output has to be generated on the server side and then passed to the client for being played together with the body animation.

These recipes have been designed on the basis of the results of the proposed questionnaire. Analysing them, major interests concern information about modalities for *accessing student facilities, setting an examination test, get credits* and *recover from formative debits*. Therefore, we developed XAPML schemas for answering to these families of questions and for getting more detailed answers once selected a topic question. The following is a recipe for answering to "*how to set an exam*":

```
<XAPML-recipe goal="suggestion-how-to">
 <agent>
    <performative type="inform" affect="happy-
    for" voice="happy.mp3"> I'm happy
    for</performative>
    <performative type="inform"
    voice="decision.mp3"> your decision of
    [topic] </performative>
    <performative type="suggest"
    voice="suggestion.mp3">These are my sugges-
    tions: </performative>
    [FORALL suggestion-i DO
       <performative type="suggest" fo-
       cus="suggestion-i"
       voice="[ordinal]_sugg.mp3"> [ordinal],
       you should [suggestion-i]</performative>
    ]
  </agent>
  <background>
       [FORALL suggestion-i DO
              [IF detail <object type="link"
   label="More details on
          [suggestion-i]">[URI-suggestion-
     i]</object>
          ]
       ]
  </background>
</XAPML-recipe>
```

These schema are applied by the Mind in order to generate the XAPML specification. In the previous example, square parenthesis represent instructions to the XAPML generator that will substitute them with the appropriate text by manipulating information in the Domain Knowledge Base.

The following is an example of XAPML output generated applying the previous schema:

```
  <XAPML>
   <agent><performative type="inform" af-
   fect="happy-for" voice="happy.mp3"> I'm happy
   for </performative>
       <performative type="inform"
   voice="decision.mp3"> your decision of set-
   ting an exam</performative>
       <performative type="suggest"
   voice="suggestion.mp3">These are my sugges-
   tions:</performative>
       <performative type="suggest" fo-
   cus="contact-professor"
   voice="first_sugg.mp3">First, you should con-
   tact the professor responsible for the exam
   you want to set.</performative>
       <performative type="suggest" fo-
   cus="contact-professor"
   voice="second_sugg.mp3">Second, you should
   subscribe for the exam in the student list
   present on the Department Web
   Site.</performative>
  ..
   </agent>
   <background> <object type="link" label="More
   details on subscribe for the
   exam">http://www.di.uniba.it/esami.php</objec
   t>
   </background>
</XAPML>
```

This transformation is executed on the server side and not on the handheld computer where, as we will see later on, only instantiated XAPML specification are interpreted and rendered at a surface level.

2.3 Agent's Body

When the student download the agent from the Department Web Site she can choose its Body according to his/her preferences. The selected body, obviously, influences the way in which the agent is able to express meanings.

Since our main aim was not to develop a new ECA player but to test the validity of our approach we developed only two types of bodies (Figure 3): a face (male professor and a young female professor assistant) and a character (a funny ball). This last one could be used, for instance, for interacting with children.

Each ECA conveys the same meanings using signals that are typical of its body. For instance the two faces will use as signal channels those typical of a face (eye, eyebrow, mouth, gaze, etc.) while the funny ball, that can be used for instance in tutoring contexts suitable for children, will use movements, colors and changes in its dimensions as communicative channels. The XAPML specification is not directly interpreted by the player that is the same for all the different bodies. This is possible since the proposed architecture decouples meanings from signals. Each body has a conditional meaning-signal table that allows to appropriately translate an XAPML tag into a Signal Expression Markup Language (SEML). SEML tags define the expressions that can be performed on each channel of the Body. Table 1 specifies how a body's <act> can be specified in SEML along the appropriate channels for the *face* [3,8] and the *ball*.

The Body Player, in My Tutor application, will then read which combination of signal to play and translate it into Flash composition and animation setting up a Scene. This double translation allow to be independent from the technology used by the player since the translation of signal into specifications that can be played by the used technology is made by the SEML-PLAYER Wrapper Module.

Every Service Agent has a related knowledge base, where it stores the information, and where it looks for the links to relevant pages; those agents cyclically check for deadlines or news about administration or scheduling. Then it broadcast to VT agents that will store this information and will inform the personal agents as soon as they are connected.

3 An Example of Interaction

We will show here an example in which VU-MAS is used by a student that need some information about exams

Initially, the user has to download the PVT agent from the VU-MAS web site. Starting from that moment a virtual tutor (VT) agent will be associated to the student who will be able to interact with the PVT ECA on his/personal device. Initially, according to the student year of course or specialization level, a stereotype is assigned to him/her. This stereotype is used to set the default student background knowledge and interests and also to select a set of basic XAPML recipes that can be used to match the user goals. Together with these recipes, the related mp3 files that have been generated by the server, are also transferred on the user platform. This is important since, in a context of mobile learning in which connectivity cannot be guaranteed, the agent can use only the animation and XAPML recipes available on the client. In case the user can connect to the VU-MAS, suggestions and answers are generated, on request, by the server and transferred on the user device for interpretation and rendering.

After the agent initialization, the user stereotype becomes the user profile and it is updated according to the user interaction with the agent. This can be used for

Fig. 3 A simple example of interaction, in Italian

adapting the information to the student situation but also for storing what the user wants to know and the system was not able to communicate due to lack of resources. In this case, when the user will be connected, the PVT will be able to download the needed resources and answer to the related user requests.

Figure 3 is an example that shows a simple interaction in which the student ask for something that the Agent does not know and, after having shown that it is sorry for this inconvenient, PVT suggest to look on the VU-MAS web site performing a deictic gaze.

When the user input his/her move or look for details in a different page, it is always possible to go back and talk to the agent by selecting the icon on the bottom status bar of the handheld application.

As mentioned in the Introduction, we plan to evaluate the system in order to assess whether the number of requested intervention increases, compared to the one requested to the human one, and the provided suggestions match the user expectations. This is possible by analyzing after each semester, if students agree, their profile, measuring in this way the frequency of use of PVT, the frequency of question categories and the type of new questions asked by the students.

The study will also try to assess the student attitudes toward the Agent in terms of how much the presence of an embodied agent improved the effectiveness and engagement of interaction. This will be evaluated through the use of a questionnaire aiming at understanding how much the student liked the character and how much they found helpful PVT suggestions.

Another aspect to evaluate is the impact on queues at the administrative offices. This system will be very useful to avoid wasting the time for asking simple questions to the employers or to let student send and receive documentation to and from the administration.

4 Conclusions

We have developed VU-MAS, a MultiAgent System to support students in their academic life. It's architecture is very open and flexible and allow to provide many services to users virtually 24 ours a day.

Personal Virtual Tutor is an animated agent designed to support the interaction with the system. The agent can be consulted on a handheld device. For this reason we decided to use an extension of the Mind-Body architecture already used in the context of the MagiCster project. In particular, the Mind, running on the server side, decide what to say and generates, through the use of plan "recipes", a specification (in XAPML) of the meaning to be communicated by the Body that can be used to generate the agent behavior on the handheld computer (client side). Since the architecture can support the use of different "bodies", that can express meanings using different signals according to their available channels, a meaning-signal table specify how a specific body has to convey particular meanings. This is represented through the use of another markup language SEML that specify which combination of signal to employ in correspondence of each XAPML tag. Then, the SEML specification is interpreted by a Wrapper, developed in Flash for PVT application, that is responsible for the synchronization and the rendering of the agent animations.

The information played by the Personal Virtual Tutor are collected by the Virtual Tutor agent from Service Agents.

References

[1] http://informatica.uniba.it/info_comuni/tutorato.htm

[2] Reeves, B. and Nass, C. 1996. *The Media Equation*. New York: Cambridge University Press.

[3] Gratch, J., Rickel, J., Andre, J., Badler, N., Cassell, J., Petajan, E. "Creating Interactive Virtual Humans:Some Assembly Required", *IEEE Intelligent Systems*, July/August 2002, pp. 54-63

[4] Marsella, S.C., Johnson, W.L., & LaBore, C.M. 2003. Interactive pedagogical drama for health interventions. In U. Hoppe et al. eds., *Artificial Intelligence in Education: Shaping the Future of Learning through Intelligent Technologies*, 341-348. Amsterdam: IOS Press.

[5] Bickmore, T. 2003. Relational Agents: Effecting Change through Human-Computer Relationships. PhD Thesis, *Media Arts & Sciences*, Massachusetts Institute of Technology.

[6] F de Rosis, B De Carolis, V Carofiglio and S Pizzutilo: Shallow and inner forms of emotional intelligence in advisory dialog simulation. *In H Prendinger and M Ishizuka (Eds): "Life-like Characters. Tools, Affective Functions and Applications"*. Springer, 2003.

[7] Berardina De Carolis, Catherine Pelachaud, Isabella Poggi, Mark Steedman. APML, a Markup Language for Believable Behavior Generation. In *H Prendinger and M Ishizuka (Eds): "Life-like Characters. Tools, Affective Functions and Applications"*. Springer, 2003.

[8] I.Poggi, C. Pelachaud, and F. de Rosis. Eye communication in a conversational 3D synthetic agent. *Special Issue on Behavior Planning for Life-Like Characters and Avatars of AI Communications*. 2000

[9] W. Lewis Johnson, Catherine LaBore, and Yuan-Chun Chiu. A Pedagogical Agent for Psychosocial Intervention on a Handheld Computer. *AAAI Fall Symposium on Dialogue Systems for Health Communication*. October 22-24 2004.