Adaptive Intelligent Agent Approach to Guide the Web Navigation on the PLAN-G Distance Learning Platform

Clara Inés Peña de Carrillo[*] José Luis Marzo Lázaro

Broadband Communications and Distributed Systems Group
Informatics and Applications Institute
University of Girona
Building PII, Lluís Santaló Avenue, S/N, 17071 Girona
Spain

Tel: +34 972418475
Fax: +34 972418098
E-mail: {clarenes, marzo}@eia.udg.es

Abstract

Hypermedia systems based on the Web for open distance education are becoming increasingly popular as tools for user-driven access learning information. Adaptive hypermedia is a new direction in research within the area of user-adaptive systems, to increase its functionality by making it personalized [Eklu 96]. This paper sketches a general agents architecture to include navigational adaptability and user-friendly processes which would guide and accompany the student during his/her learning on the PLAN-G hypermedia system [1] (New Generation Telematics Platform to Support Open and Distance Learning), with the aid of computer networks and specifically WWW technology [Marz 98-1] [Marz 98-2].

The PLAN-G actual prototype is successfully used with some informatics courses (the current version has no agents yet).

The proposed multi-agent system, contains two different types of adaptive autonomous software agents: Personal Digital Agents (Interface), to interact directly with the student when necessary; and Information Agents (Intermediaries), to filtrate and discover information to learn and to adapt navigation space to a specific student.

Keywords

Adaptive Hypermedia Systems, User Modelling, Multi-Agent System, WWW, Adaptation.

1 Introduction

At the University of Girona, an interdisciplinary group has developed a project to create an integrated telematic learning platform called PLAN-G, where teachers can create and publish new dynamic and interactive teaching materials that make comprehensive use of all the new possibilities offered by the information technologies and the Internet stated above. The platform has been used by the students to access these materials in a decentralized way from anywhere on the Internet,

[*] She is working at the Universidad Industrial de Santander (Colombia) and doing her PhD dissertation at the University of Girona.

[1] This research has been partially supported by the University of Girona, grant S-UdD97-197.

and at the same time by the teachers to keep track of students' utilization. It also improves and facilitates communication between students and teachers at all levels. Figure 1 depicts PLAN-G general internal open architecture based on WWW technology.

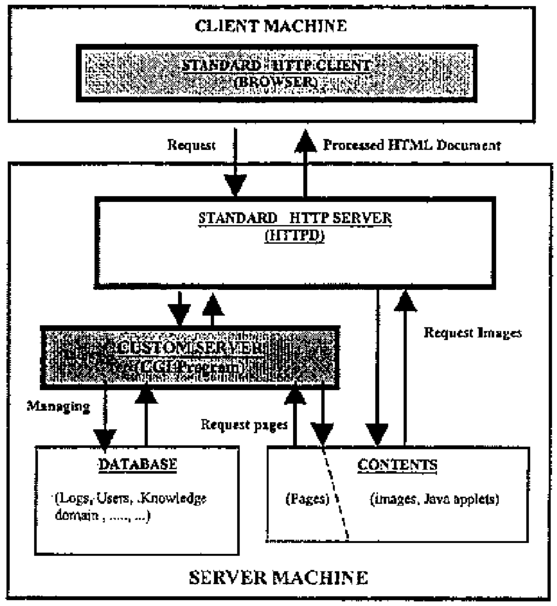

Figure 1. PLAN-G Open Telematics Learning Platform: Internal Architecture

Currently, PLAN-G is a hypermedia system that allows access to the learning knowledge in a static way. The lecturer builds a linear navigational path by means of a directed graph, using the contents stored in HTML page files and linking them based on his/her criteria for the curriculum organization.

Looking at the importance of considering the personalized learning for each student, we are working to include the adaptability characteristic to the system.

From the analysis of some adaptive educational systems based on the Web technology, we found that the key to including PLAN-G's adaptability is based mainly on the information collected to build a user model that follows the classic loop showed in Figure 2 [Brus 98].

We propose using a multi-agent system architecture (MAS-PLANG) to give adaptibility to PLAN-G.

MAS-PLANG will contain two different types of adaptive autonomous software agents: Personal Digital Agents or PDAs, and Information Agents or IAs. The PDA agents (Interface) will be used to interact directly with the student when necessary. This interaction will allow PDA agents to learn from the student after he/she clicks into the working environment area buttons and to suggest politely the best way to do some tasks. The IA (Intermediary) agents would filtrate and discover information about the knowledge domain to learn and the knowledge domain of user learning to adapt the navigation space to a specific student.

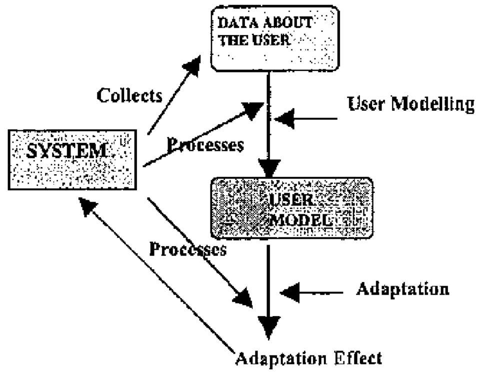

Figure 2. Classic Loop "User Modelling – Adaptation" in Adaptive Systems

If the information gathered for a specific student is not enough to suggest a positive action, agents of the same type will communicate with each other in order to achieve goals based on "wide" information.

Finally, the agents will be tested in order to evaluate their performance, taking into account their actions and students reactions to them.

This paper is organized as follows. The second section presents the state of the art and fundamental concepts about adaptive hypermedia systems and software agent systems. The third section gives a brief description of the PLAN-G hypermedia system. The fourth section presents the use of the agent technology to enable the PLAN-G as an Adaptive Hypermedia System, the major contribution of this paper and concludes with some comments and plans for future work.

2 State of the art and fundamental concepts

2.1 Traditional Hypermedia Systems and Adaptive Hypermedia Systems

Educational Hypermedia Systems, in order to be an effective educational tool, allow students to learn at their own pace, but under some direction. According to [Jona 94] there is a growing body of empirical evidence to suggest that learners tend to make poor decisions in learner controlled systems. In traditional hypermedia systems, students become lost, skip important content, choose not to answer questions, look for visually stimulating rather than informative material, and use the navigational features unwisely. To overcome traditional hypermedia systems drawback, adaptive hypermedia systems emerged, considering specific knowledge and tasks for an individual user.

[Adel 95] annotates that a hypermedia system should include dynamic and changing materials, and follow special characteristics to navigate in the navigation space (in order to avoid the student getting lost) and to promote a real, comfortable way of learning, supporting additionally, the possibility of classifying students behavior in categories depending on the ways of searching information, e.g.: direct search with a final acquaintance goal, source checking with high priority to find elements of interest or fortuity search guided by the item's attractiveness. Each type of user behavior determines which navigational facilities should wait and use and which type of design decisions the designer should adopt.

The usability and applicability of the adaptive hypermedia systems has been defined by [Brus 96-1] as when the system is expected to be used by people with different goals and knowledge and when the navigation space is reasonably large. Users with different goals and knowledge may be interested in different pieces of information presented on a hypermedia page and may use different links for navigation.

The user model can adapt the information and links being presented to a given user. Adaptation can also assist the user in a navigational sense, which is particularly relevant for a large navigation space.

By knowing the user goals and knowledge, adaptive hypermedia systems can support users navigation by limiting the browsing space, suggesting most relevant links to follow, or providing adaptive comments to visible links. In fact, [Brus 96] summarizes the ways to implement navigational adaptivity as: direct guidance, adaptive hiding or re-ordering of links, adaptive annotation or map annotation. On the basis of the user model, the system decides what is the best next node for user's visit; the "best next link" can be either one among those of the current page (and then it can be outlined) or a dynamically generated link, usually a "next" button, which is added as a complement of the current page.

To build a user model the designer has to consider static and dynamic students' information. Static information as goals, degree of knowledge with respect to the course content offered, background and navigation space experience; dynamic information as the knowledge acquired with the system, ways of interaction, etc.

The system's designer should adapt the presentation or else navigation, but the essence of the adaptive hypermedia systems is to adapt the navigation activity process to reduce cognitive overload in order to facilitate learning. Figure 3 summarizes the general architecture of an adaptive hypermedia system proposed by [Da Silv 98].

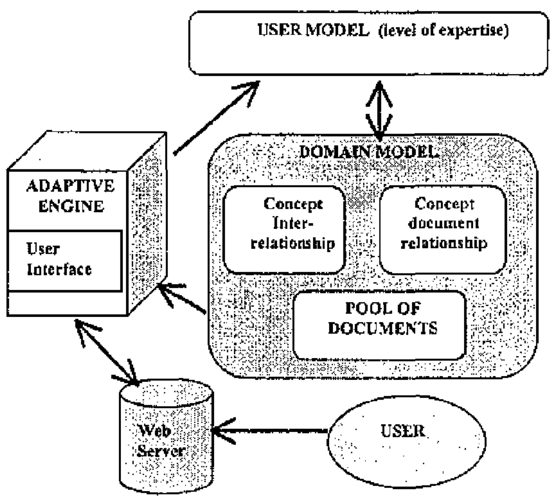

Figure 3. Architecture of an Adaptive Hypermedia System

The following hypermedia systems exhibit special adaptive characteristics in educational environments:

ELM-ART II [Webe 97] uses a combination of an overlay model and episodic user model, supporting adaptive navigation as individualized diagnosis and helping in problem solving tasks. Adaptive navigation support is achieved by annotating links and selecting the next best step in the curriculum on demand.

SPIRAL [Wild 98] adapts the navigation space based on competence, personal preferences, interest and knowledge following any degree of depth without defining explicit navigational paths. The lecturer offers a map and a set of objectives that guide student exploration.

CORAL [Chue 96] has proposed ways to gather student information to construct the user model. It calculates the system adaptability based on linearity measure.

HYPERTUTOR [Pere 95] adapts the hyperspace available depending on student knowledge. As long as the student acquires concepts, hyperspace accessibility grows, giving the opportunity to reach new parts of it.

2.2 Software Agent Systems

Recent growth of educational hypermedia systems on Internet has created a need for new tools to help handle and adapt to the profile of individual student the way of learning large amounts of information. We think that software agent systems, often called *agents,* could address these issues.

Agent technologies can be used to assist users in gathering information and can gather and select this information locally, thereby avoiding unnecessary network loads. Software agents have their roots in work conducted in the fields of software engineering, human interface research and Artificial Intelligence (AI). Software agents, up to now, can be classified under Multiple Agent Systems (MAS), one of the three branches of distributed AI research, the others being, Distributed Problem Solving (DPS) and Parallel Artificial Intelligence (PAI) [Nwan 97].

The term *agent* provides a starting definition: "one who acts for, or in the place of another". A software agent is a software package that carries out tasks for others, autonomously without being controlled by its master once the tasks have been delegated. The "others" may be human users, business processes, workflows or applications [Etzi 95]. The agents provide the users with extra eyes, ears, hands or even brains.

[Kaas 98] summarizes the characteristics of software agents as: "A software agent is **reactive** since it is able to take initiatives". One example of this may be to tell the user something that he/she needs to know but did not know how to ask. To be able to be reactive, the agent has to be able to sense its environment and act on the basis of this. An agent is **autonomous**: it does not only obey instructions but it makes its own decisions. An agent is **goal-oriented.** Software agents keep running continuously while the user is doing other things.

Additionally, [Fran 96], defines the following optional properties, which produce extensions to basic agents: The software agent is **communicative**: it communicates with its user and other agents. It is learning, changes its behavior based on its previous experience. It is **mobile**: it is able to transfer itself from one machine to another. It is **flexible**: its actions are not scripted. It has **character**: believable personality and emotional state. An intelligent agent performs tasks that have both declarative and procedural components. It can use alternative reasoning strategies, including belief management. An agent's repertoire of tasks represents its capabilities. Each task can have its procedural "how to do" component represented as rules, knowledge sources (rule sets), or methods.

Figure 4, depicts some of the key components of multi-agent problem-solving. [Farh 98]

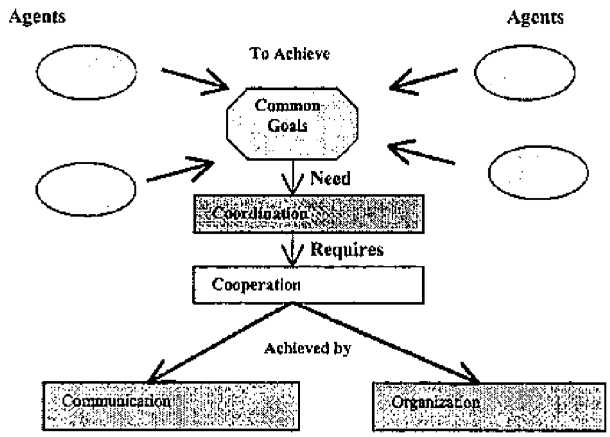

Figure 4. Multi-agent problem solving

[Nwan 97] offers some taxonomy for classifying agents. We think useful to review the following information in order to know the characteristics that the MAS-PLANG agents will exhibit.

- **Collaborative agents**: Interconnect existing legacy software, such as expert systems and decision support systems, to produce synergy and provide distributed solutions to problems that have an inherent distributed structure. MAS-PLANG agents will be collaborative when the particular information would not be enough to suggest a positive action.
- **Interface agents**: provide for personalized user interfaces, for sharing information learned from peer-observation, and for alleviating the tasks of application developers. Interface agents adapt to user preferences by imitating the user, by following immediate instructions of the user or through the Pavlov effect (learning from positive and negative responses of users). One has to realize that interface can only be effective if the tasks they perform are inherently repetitive (otherwise, agents will not be able to learn) and if the behavior is potentially different for different users (otherwise, use a knowledge base). MAS-PLANG upper level agents will be of this kind of agents acting as typical personal digital agents that learn about the environment by means of the activities depicted on Figure 7 [Maes 94].
- **Mobile Agents**: reduce communication costs and overcome limitations of local resources. Decentralization of the selection process avoids unwanted information being sent over the networks, thus economizing on network utilization. At this time, we think that some of the MAS-PLANG agents could be mobile agents.
- **Information Agents**: Circumvent 'drowning in data, but starving for information.' This corresponds to solving the problem of information overload. We select this type of agent to include in IA-agents (MAS-PLANG first level agents) to filter information about the domain model to learn and domain model of user learning and to take intelligent decisions about a given adaptavity.

3 The PLAN-G Hypermedia System

PLAN-G architecture, as a hypermedia system, has general components of generic intelligent tutoring systems. It has a hypermedia component represented by an interface, a navigation control area (buttons) and the navigation space. The tutor's component adds intelligent behavior to the system based on a pedagogic organization of the domain to learn and student preferences and activities. Figure 5, depicts PLAN-G architecture and general operation.

This system was designed to be a learning platform, in which the teacher supplies the course contents and the students may navigate through them. Each course is called "a teaching unit" and it is made up of a group of HTML page files structured by a directed graph, which allows the creation of a navigational path. By means of the administration options of the graph, it is possible to create, modify and delete "teaching units", giving in such a case the opportunity to change destinations or to define lineal paths. The navigational structure is managed independently of the contents and it is stored in a database. The user type "teacher" only has competences in dealing with the structure and the contents of the unit, to adjust them as needed.

Each student visit to web pages is registered and stored into the database, for future analysis of his/her activity in the system. The student has the opportunity to select the language and the icon structure for the working environment. Currently, the system offers 5 languages (Catalan, Spanish, English, French and German).

Additionally, there are background processes that collect the student's behavior throughout his/her movement by the buttons of the interface. This knowledge is stored into the database and will be used to feed the user model that we will incorporate into the system by the intelligent agents architecture presented in detail in section 4.

Basically, the system works around 3 concepts: users, units and a performance level given by the languages. The system associates to each user a configuration and a group of sessions and visits (static and dynamics data). Each unit has an associated group of basic exercises or bibliography web pages and a directed graph. For each page there exists an association to selected languages.

The communication area includes standard mailing systems, interactive chats and discussion lists in order to promote the collaborative and co-operative work between the course participants and to capture some information that provides feedback from the user model and help in pedagogical analysis of the course if will further.

Because the tutor's component currently does not have a full organization to give adaptivity to PLAN-G, an approach of the multiagent architecture specificacion is presented.

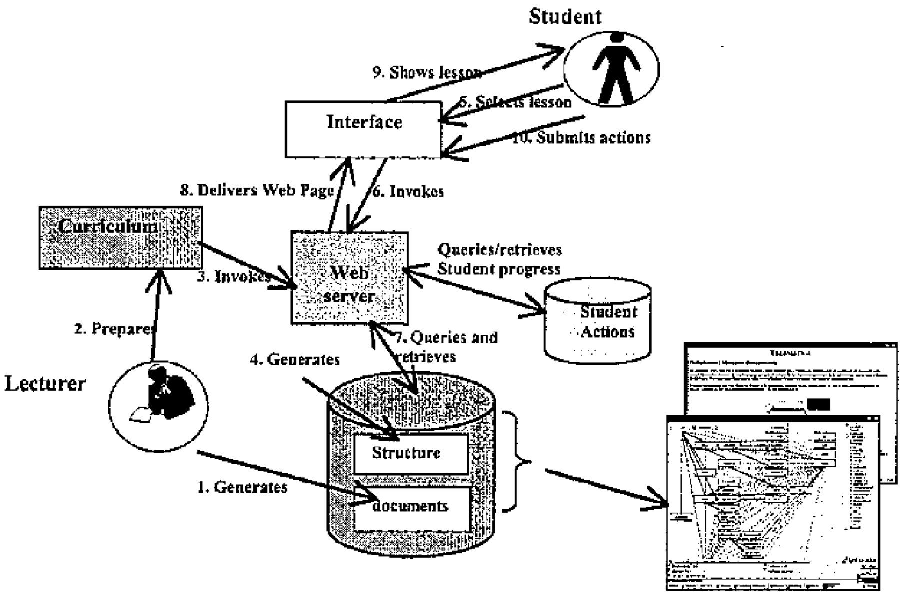

Figure 5. PLAN-G Hypermedia System: General Architecture and Operation

4 Multi-Agent architecture specification for the PLAN-G Adaptive Hypermedia System

Considering the architecture of Figure 3 and using the agent technology to include adaptivity in PLAN-G, we propose the MAS-PLANG architecture depicted in Figure 6.

The two level architecture is formed by: Intermediary agents, called IAs (Information Agents) at the lower level and Interface Agents, called PDAs at the upper one.

Upper level agents will act as typical personal digital agents that learn about the environment by means of the activities depicted on Figure 7 [Maes 94]. They will monitor and imitate the user's behavior and adapt it based on user feedback; they will be able to be trained by the user on the basis of examples, and they will be able to ask for advice from other personal digital agents.

Basically, PDA agents will assist the students when they move over the button's interface which indicate how to carry out the students" actions such as: to go forward, to go backward, to play music, to ask for help, to look at the bibliography, to consult the navigation history, to look at the multimedia material, to take a pause in learning online, to print, to search, to make exercises, to communicate with the teacher or classmates using chat or e-mail or consulting news, and to follow the navigational path.

In Figure 8 the button's area using a PLAN-G example web page is shown. The PDA assistance will consist of giving advice or suggestions on how to take better actions if the student exhibits a special behavior. Aditionally, PDA agents will help lower level agents in doing some tasks.

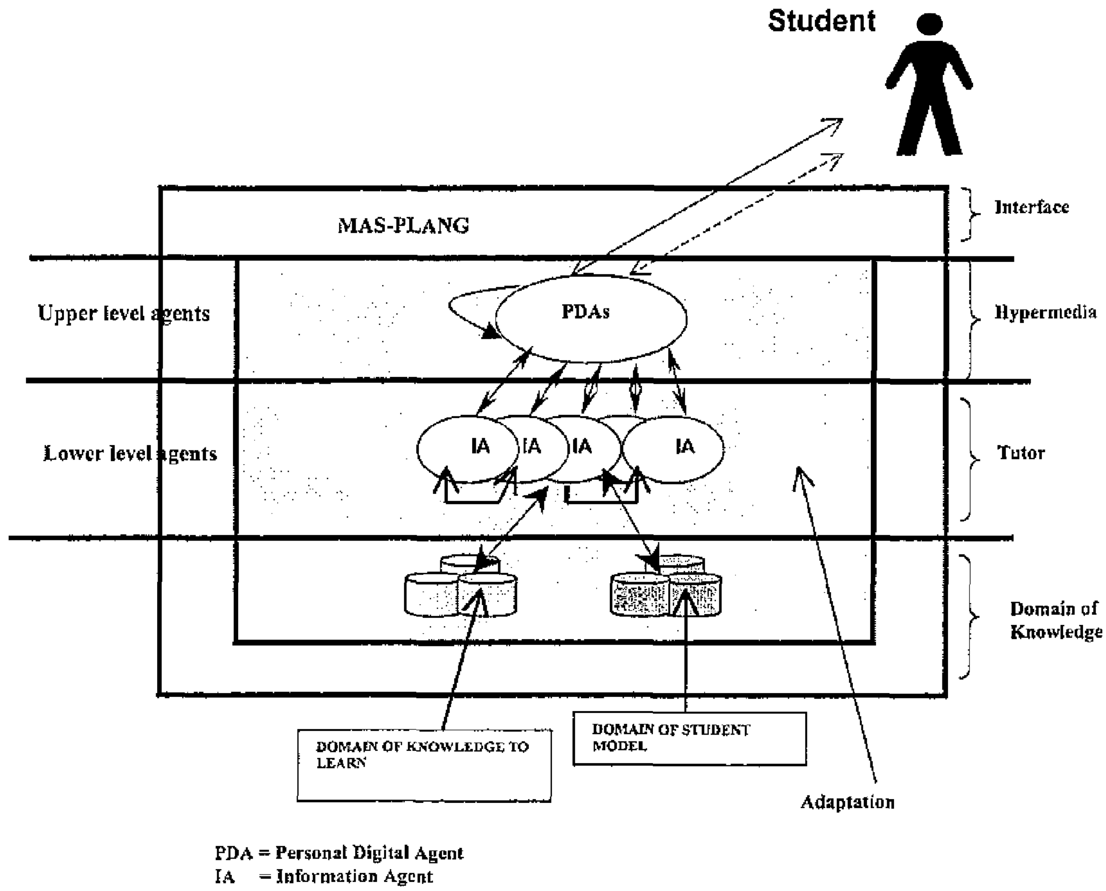

Figure 6. MAS-PLANG Architecture

Lower level agents (IAs) will act as intermediary agents between upper level agents and knowledge domain databases (knowledge domain to learn and knowledge domain of user model). They will filtrate and discover information about the knowledge domain to learn and knowledge domain of user learning, they will build the user model, adapt navigation space to particular student, update databases and communicate with upper level agents to complete the tasks that force student interactions. In conclusion, they will take the intelligent decisions that will increase the adaptivity of PLAN-G.

Each IA agent will have its corresponding PDA agent.

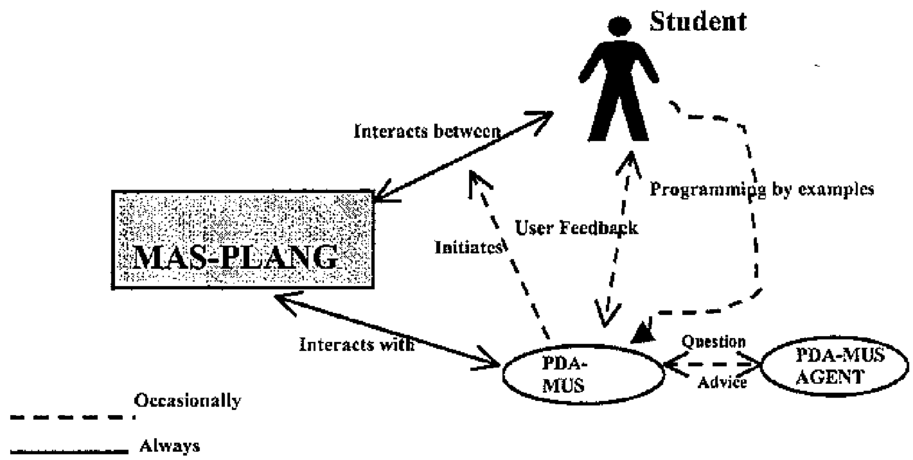

Figure 7. Ways of Learning for Personal Digital Agents

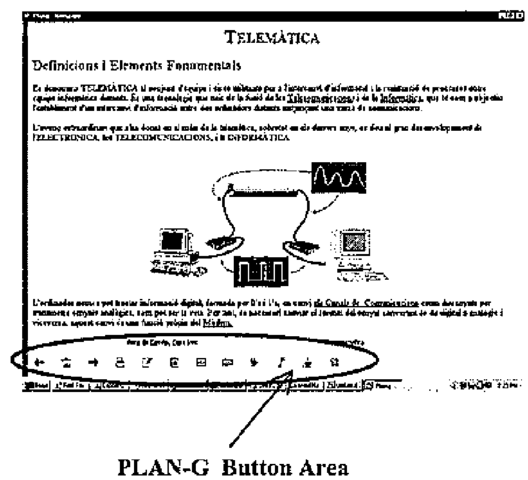

Figure 8. Button Area in PLAN-G Working Environment

The IA-NAVIGATION agent will build the learning navigation space for a specific student based on characteristics of domain model of knowledge and student model. The following example shows some lower level agents behavior:

If it is the first student session, the IA-NAVIGATION agent will have to build a navigation space composed by a multiple-choice exercise in order to classify the student according his/her level of expertise (lets say beginner, intermediate or expert). To achieve this goal, it will ask its IA-EXERCISES neighbor agent to select the corresponding exercise. The IA-EXERCISES agent will select a suitable exercise from the exercises domain database and will tell the IA-NAVIGATION agent to build a special navigation space with the information selected. The IA-NAVIGATION agent will build the corresponding navigation space and will tell the PDA-NAVIGATION agent to show it to the student. The PDA-EXERCISES agent will collect the student's answers and will send them to the IA-EVALUATION and IA-EXERCISES agents. The IA-EVALUATION agent will have to evaluate the exercise, communicate the student level, and update student model. The IA-EXERCISES agent will have to update the student model (See Figure 6 for agents relationships).

With a student profile, the IA-NAVIGATION agent will build the corresponding navigation space based on model domain of knowledge and stucent model, and will be concerned with the student progress in order to extend navigation space to new hyperdocuments (to enable new links).

The IA-SEARCH agent will gather information from the knowledge database of student learning and knowledge database to be learned by a specific student, in order to look for a particular piece of information requested by the student. The IA-SEARCH agent will tell PDA-SEARCH agent to show the information.

The IA-HISTORY agent will gather information from records of student learning based on knowledge domain of user model. It will build the navigation space of known concepts as they were learned by the student including solved exercises and evaluations. The IA-HISTORY agent will tell PDA-HISTORY agent to show the information to the student.

Conclusions and future work

We have presented a proposal to convert PLAN-G into an Adaptive Hypermedia System, using a two-level multi-agent system (MAS-PLANG). The agent technology will offer a useful field of experimentation for the PLAN-G system.

Currently, the agent programming language and the agent communication language are being selected to allow the best implementation taking into account the compatibility with the current system.

To experiment with the new prototype, the use of the same informatics courseware prepared for the PLAN-G hypermedia system, with a body of 450 students, is planned.

We are especially interested in knowing the performance of the agents and the students' reactions to agents' actions.

References

[Adel 95] Adell Jordi. 1995. La navegación hipertextual en el World Wide Web: Implicaciones para el diseño de materiales educativos. Universidad de las Islas Baleares, Spain. http://nti.uji.es/docs/nti/edutec95.htm
http://www.igd.fhg.de/www/www95/proceedings/papers/52/www3.htm

[Brus 96] Peter Brusilovsky, Elmar Schwarz, Gerhard Weber. 1996. A Tool for Developing Adaptive Electronic Textbooks on WWW. Webnet96 proceedings. http://www.contrib.andrew.cmu.edu/~plb/WebNet96.html.

[Brus 96-1] Peter Brusilovsky. 1996. Methods and Techniques of Adaptive Hypermedia. User Modelling and User Adapted Interaction, 1996, Vol. 6, N 2-3, pp 87-129

[Brus 98] Peter Brusilovsky, Alfred Kobsa, Julita Vassileva. 1998.Adaptive Hypertext and Hypermedia. Kluwer Academic Publishers.

[Chue 96] Chuen-Tsai Sun, Chien Chou. 1996. Experiencing CORAL: Design and Implementation of Distant Co-operative Learning. IEEE Transactions on Education. Vol 39, N-3.

[Da Silv 98] Denise Pilar DaSilva, Rafael Van Durm, Erik Duval, Henk Olivie. 1998. Concepts and Documents for Adaptive Educational Hypermedia: A model and a Prototype. Proceedings of the 2nd Workshop on Adaptive Hypertext and Hypermedia HYPERTEXT'98. Pittsburg, USA. http://wwwis.win.tue.nl/ah98/Pilar/Pilar.html

[Etzi 95] Oren Etzioni & Daniel S. Weld. 1995. Intelligent Agents on the Internet: Fact, Fiction and Forecast. Department of Computer Science and Engineering, University of Washington.

[Farh 98] Faramarz Farhoodi and Peter Fingar. 1998. Competing for the Future with Intelligent Agents. http://www.trcinc.com/news/iagents/index.html.

[Fran 96] Franklin, S. And Graesser, A. 1996 Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents Proceedings of the Third International Workshop on Agent Theories, Architectures and Languages. Springer-Verlag.

[Kaas 98] Eija Kaasinen. 1998. Usability Issues in Agent Applications: What Should the Designer be Aware of. VTT Information Technology, Finland. Usinacts 1.6.1998. http://atwww.hhi.de/USINACTS/agent.html

[Jona 94] David H. Jonassen. 1994. Technology as Cognitive Tools: Learners as Designers. Pennsylvania State University. ITForum Paper #1. http://itech1.coe.uga.edu/itforum/paper1/papaer1.htm

[Maes 94] Pattie Maes. 1994. Agents that Reduce Work and Information Overload. MIT, Media Laboratory. http://lcs.www.media.mit.edu/people/pattie/CACM-94/CACM-94.p1.html.

[Marz 98-1] J.L. Marzo-Lázaro, T. Verdú-Carbó, R. Fabregat-Gesa. 1998. User Identification and Tracking into an Educational Web Environment. ED-MEDIA & ED-TELECOM 98. World Conference on Educational Multimedia and Hypermedia and World Conference on Educational Telecommunications. Freiburg, GERMANY. June, 1998.

[Marz 98-2] J.L. Marzo, M. Estebanell, R. Fabregat, F. Ferrés, T. Verdú. Support Units for University Teaching Based on WWW. ED-MEDIA & ED-TELECOM 98. World Conference on Educational Multimedia and Hypermedia and World Conference on Educational Telecommunications. Freiburg, GERMANY. June, 1998.

[Nwan 97] Hyacinth S. Nwana, Nader Azarmi (Eds.). 1997. Software Agents and Soft Computing – Towards Enhancing Machine Intelligence – Concepts and Applications. Lecture Notes in Artificial Intelligence 1198. Springer.

[Pere 95] Tomás A. Pérez, Julian Gutierrez and Philippe Lopistéguy. 1995. An Adaptive Hypermedia System. Departamento de Lenguajes y Sistemas Informáticos, Facultad de Informática, Universidad del País Vasco, Spain. Artificial Intelligence in Education, AIED'95. AACE: Charlottesville, USA.

[Webe 97] Gerhard Weber and Marcus Specht. 1997. User Modelling and Adaptive Navigation Support in WWW-based Tutoring Systems. Proceedings of User Modelling 1997. 289-300.

[Wild 98] Chris Wild. 1998. Adaptive Web-based Learning for Students with Diverse Backgrounds.- Case Study in an Introductory Programming Language Course. Old Dominion University – Department of Computer Science. www.odu.edu/ ~wild/docs/spiral.html

