EVA: An Interactive Web-Based Collaborative Learning Environment

Leonid Sheremetov[1], Adolfo Guzmán Arenas

Centro de Investigación en Computación (CIC), Instituto Politécnico Nacional (IPN), México

Abstract

In this paper, a Web-based learning environment developed within the project called Virtual Learning Spaces (EVA, in Spanish) is presented. The environment is composed of knowledge, collaboration, consulting and experimentation spaces as a collection of agents and conventional software components working over the knowledge domains. All user interfaces are Web pages, generated dynamically by servlets and agents. Each EVA host has an agent associated with it, which can establish links with the other EVA hosts forming shared knowledge and collaboration space of EVA. The article focuses on the description of the system architecture, implementation of particular software modules, and the application results. A companion article describes current EVA applications in the M. Sc. program in Computer Science.

Key words: architectures for educational technology system, distributed learning environments, authoring tools and methods.

1. Introduction

The technological innovations on the one hand, and the growing popularity and availability of Internet on the other, are the main reasons of development in the last years of numerous applications and investigation projects in the field of technology-mediated education (Youngblut, 1994; O´Malley, 1995; Collis, 1999; Hamouda & Tan, 2000). The virtual teaching, integrating computer and

[1] Computer Science Research Center, National Technical University (CIC-IPN), Av. Juan de Dios Batiz esq. Othon de Mendizabal s/n Col. Nueva Industrial Vallejo, México, D.F., C.P. 07738, Mexico, Tel. +52 - 57296000 ext. 56576; Fax: +52 - 5586-2936, e-mail:, sher@cic.ipn.mx

communication technologies in different educational scenarios, is advocated as a solution to the problem of exponential growth of knowledge in the contemporary society. As a matter of fact, information technologies offer exciting opportunities to thoroughly redesign the education process and to achieve, among others, the following benefits: integration of means (text, audio, animation and videotape), interactivity, access to big quantities of information, plans and individualized work rhythms and immediate answer to the apprentice's progress.

However, the introduction of the technologies in the real education environment is a difficult problem. For example, the relatively modern and quite common methods of teaching, such as videoconferences or on-line versions of textbooks have not been very successful due to their limitations and to the negative facets that they contain, to mention some: weak integration of the means, weak interactivity among student, professor and the system, dominant paradigm of knowledge transmission, separation of the apprentices from the cooperative activities in the learning scenarios. Therefore, the intense search of new pedagogic solutions and teaching/learning technologies is a time challenge, where the advanced information technologies play the main role.

The EVA project is dedicated to the development and implementation of a software learning environment, personalized and collaborative, by means of which different academic and administrative activities can be offered in a distance manner to the students of different institutions and public and private companies (Guzmán & Nuñez, 1998; Nuñez, Sheremetov, Martínez, Guzmán & De Albornoz, 1998). As a Virtual Learning Environment (VLE), it applies the methodology and tools of Distance Learning (DL), Intelligent Tutoring Systems (ITS), Interactive Learning Environments (ILE), and Computer Supported Collaborative Learning (CSCL) to obtain a new paradigm of the Automatic Configurable Distance Education, which we have named *EVA (Espacios Virtuales de Aprendizaje - Virtual Learning Spaces).* This concept does not pretend to substitute a human tutor by an intelligent software system with one-on-one tutoring model, but tries to use the ITS technology to accomplish with some rudimentary tasks. As well as in the ILE, the intelligence invested in EVA is distributed across a range of tools rather than centralized in a tutor, but the proposed methodology does not use the basic principle of the ILE - it doesn't permit students investigate and learn topics absolutely free of external control. It also completely differs from the concept of Virtual University, introducing the artificial intelligence and interactivity in the VLE. The EVA philosophy is congruent with the existing classroom practice: mainly aims at learning goals and outcomes that are already embedded in traditional curricula, do not neglect the use of conventional learning materials, and can usually be plugged into existing curricula with minimal change to course plans. It benefits from some explicit representation of the topics that students investigate, but it doesn't need to be omniscient. Further, since EVA does not attempt to tutor, it is free of obligations to model students' cognition and to make complex pedagogical decisions.

The rest of the article has the following structure. In the next section, the EVA conceptual architecture is discussed, the concept of a Multi-book is presented and the problem of planning of learning trajectory is considered. In section 3, implementation of EVA software architecture is discussed. Multiagent virtual learning community is introduced in section 4. Development technologies and tools in EVA are described in section 5. Finally, current applications of EVA environment, conclusions and future work directions are discussed.

2. EVA conceptual architecture

EVA is a learning paradigm that considers different forms of acquiring, transmitting and exchanging knowledge among people and work groups that don't usually have physical access to the conventional sources of knowledge: books, magazines, schools, universities, laboratories, libraries, good professors, etc. EVA constitutes a conception of the education that uses advanced information technologies like, for instance: JAVA servlets and applets, Agents, Artificial Intelligence, Groupware, Multimedia and Virtual Reality.

The conceptual architecture of EVA is structured into the four essential learning elements, called Virtual Learning Spaces. These spaces are: *knowledge* - all the necessary information to learn, *collaboration* - real and virtual companions that get together to learn, *consulting* - the teachers or tutors (also real and virtual), who give the right direction for learning and consult doubts, and *experimentation* - the practical work of the students in virtual environment to obtain practical knowledge and abilities. These four spaces are complemented with the *personal space* where user-related information is accumulated. These spaces are supported by a number of tools developed using different information technologies (Fig. 1).

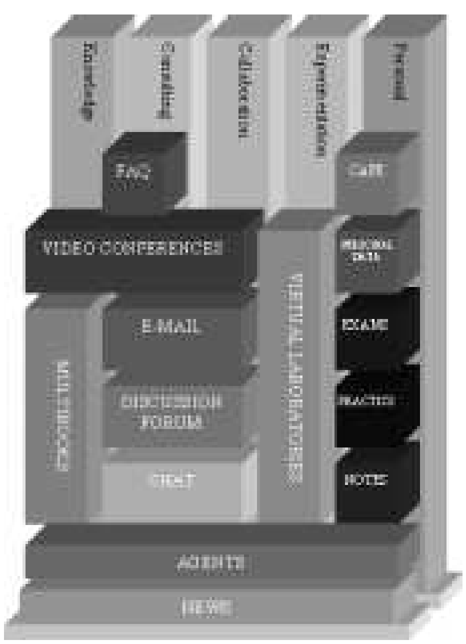


Fig. 1. Learning spaces and tools distribution

The other main concepts of EVA are the following:

1. The "Multi-book" (or POLIlibro, in Spanish) concept, a personalized electronic book, generated by concatenating selected units of learning material (ULM) along the learning trajectory for each knowledge domain.
2. A multilevel knowledge representation model of the problem domain based on concept graph formalism. The ULMs form the nodes in the knowledge space. All the learning activities are attributed to these nodes.
3. Planning of learning trajectory for students who navigate in EVA (they are called EVAnauts): EVA allows a student to establish her own route and learning rhythm, according to her scientific and professional interests, to the time and resources at her disposal, making possible to combine her more immediate requirements with the long term objectives.
4. The intensive use of agent-based tools and multiagent systems (MAS) as virtual personal assistants that help students to learn and track their intentions and performance, virtual students or learning companions, for mining information, managing, planning and scheduling learning activities, and to organize the workgroups and groups activities, and re-configure their work and learning spaces.

2.1 Multi-level knowledge domain model and Multi-books

The Virtual Learning Spaces are associated with the knowledge taxonomy of Computer and Engineering Sciences. The nodes of the taxonomy are named in the same form in each space but its contents are different in each one.

In the *Knowledge spa*ce the *taxonomy of knowledge* constitutes an agglutinating center of the space, and is elaborated by "Multi-books”. A “Multi-book” is a book, where the chapters are made of units of learning material (ULM), which can be presented in several ways. An ULM can be a text in Word, Audio, PowerPoint Presentation, Video, Multimedia or Virtual Reality, etc. All the learning activities such as cooperative work, experiments, tasks, quizzes, etc. are attributed to these ULMs. The ULMs are the basic elements of knowledge structuring, they are self-sufficient that makes it possible to include the same ULM in different Multi-books.

To represent the knowledge taxonomy we have proposed a model based on the hierarchy of knowledge domains and concept graph representation of knowledge and learning activities. The model of knowledge is represented in the form of a concept graph *G*. Each node *i* has a number of attributes, including its weight (node ratio) that means the importance to achieve a final goal of learning I_i, knowledge constraints (prerequisites) $P_{i,j}$ and estimated time to learn the concept T_i. A precedence relation $i \rightarrow j$ is used to relate concepts that means that concept *'i'* has to be learned directly before the concept *'j'*. Knowledge constraint attribute $P_{k,j}$ captures a prerequisite relation *k-->j* that means that concept *'k'* is needed to learn the concept *'j'*, but not necessarily precedes it. Each arc *(i, j)* also has its weight (relation ratio) that means the strength of relationship or the necessity of the previous concept for the next one $N_{i,j}$. All the ratios are estimated from “not necessary” to “obligatory” and are represented using 5-valued numeric ranking scale from 0.2 to 1. To capture the fact that it can be necessary to learn two or more concepts to proceed with the next one, AND arcs are used.

Alternative paths mean different options for learning. Alternative routes are analyzed basing on the following decision criterion (to be maximized):

$\sum_{i=1}^{n} I_i * N_{i,j}$, where n is a number of nodes on the route.

The semantics of this criterion is that generally more long paths are preferable to be selected if and only if they are consistent with a temporal constraint:

$\sum_{i=1}^{n} T_i \leq T_c$, where T_c is a time constraint

Fig. 2 shows an excerpt from the domain model for the "Distributed Intelligent Systems" (DIS) Multi-book, adapted syllabus of which is shown in Table 1. Fig. 3 reflects the process of the conversion of the course syllabus to the Multi-book model.

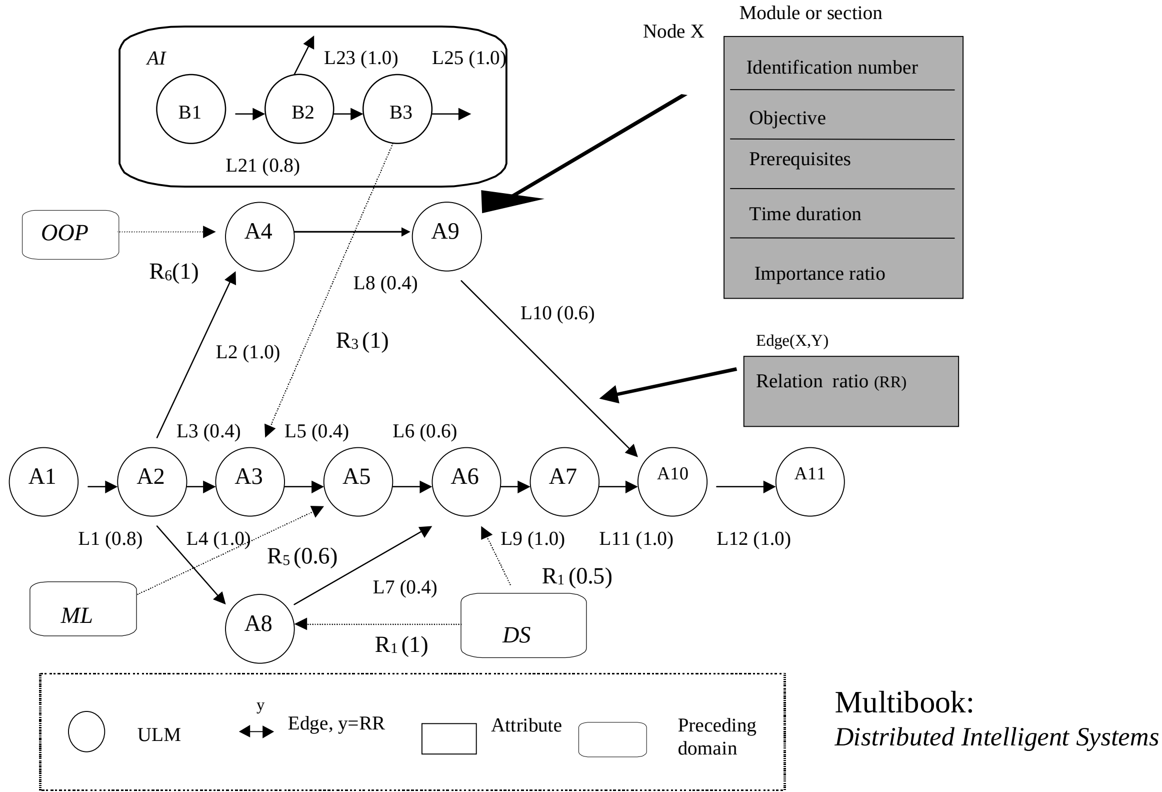


Fig. 2. A fragment of a concept graph for a Multibook on Distributed Intelligent Systems

Table 1
A fragment from the "Distributed Intelligent Systems" Multibook syllabus

Node number	ULM title	Prerequisites	Importance ratio	Time duration
A1	Introduction to agents and MAS		0.6	4
A2	Agents: definitions y classifications		1.0	4
A3	Shared knowledge and general ontology	Artificial Intelligence	0.4	6
A4	Agent oriented software engineering	Object oriented programming	0.8	4
A5	Agent oriented programming	Mathematical logic	1.0	10
A6	Communication in MAS	Distributed systems	1.0	12
A7	Interaction in MAS		1.0	6
A8	Mobile agents	Distributed systems	0.4	4
A9	Formal specifications of MAS		0.8	6
A10	MAS frameworks and development tools		1.0	6
A11	Examples of software agents		0.6	4

Table 1

A fragment from the "Distributed Intelligent Systems" Multibook syllabus

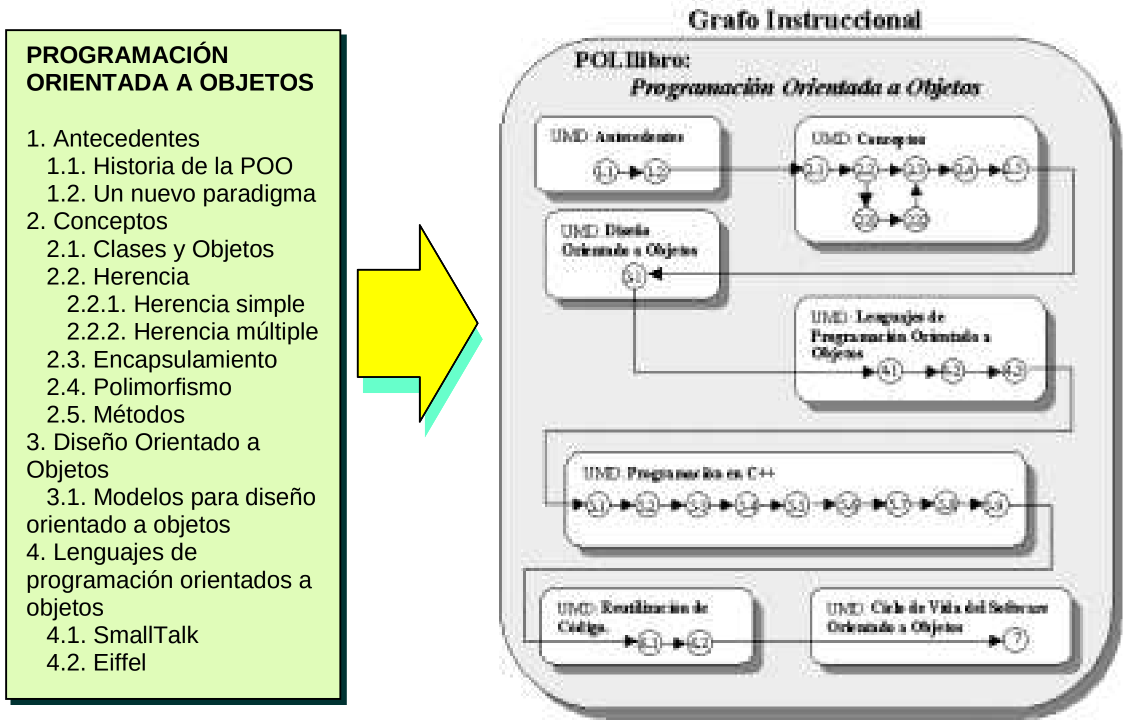


Fig. 3. An example of a conversion of a course program in a Multi-book

From fig. 2 it can be observed that the ULMs of this Multibook use concepts from a number of other Multibooks as prerequisites: OOP, AI, ML and DS (it should be mentioned that at this figure from all the prerequisites only a fragment of AI course concept model is shown). To group Multi-books, a concept of knowledge domain is introduced. A subset *G'* of graph *G* can be divided in knowledge domains if and only if the following conditions are fulfilled. There exist a collection $D_0, D_1, \ldots, D_n$, n≥2, of subgraphs G', for which:

- $D_0 \cap D_i = \varnothing$
- The subgraph D_0 has outgoing edges (of precedence) to each subgraph D_i
- The subgraph D_0 has no incoming edges (of precedence) from any subgraph D_i
- In each domain D_i there exist at least one node that has no outgoing edges (of precedence).

The subgraph D_0 is called the "common domain" for the domains $D_1, \ldots, D_n$. Note that the division of the graph *G* in knowledge domains not necessary is unique. Domains also can have common (shared) nodes or intersections. This definition can be used to obtain decompositions successively from whatever subgraph. For example, in the case of the M. Sc. program in Computer Science, common domain consists of Object Oriented Programming (OOP), Data Base Design (DBD), Discrete Mathematics (DM) courses, etc. (fig. 4). Domain of Artificial Intelligence (AI) consists of AI, Logic Programming (LP), Mathematical Logic (ML) courses, etc., and contains several sub-domains: Knowledge Based Systems (AI_1), Automatic Learning (AI_2), Natural Language (AI_3), Vision and Robotics (AI_4). Domains are connected by prerequisite relations (R_1-R_6). Usually this relation has hierarchic character, but sometimes, one domain is related with sub-domains of different specialty, which is also reflected at the level of knowledge units, e.g. the case of Distributed Systems (DS) and Distributed Intelligent Systems (DIS), related by R_1.

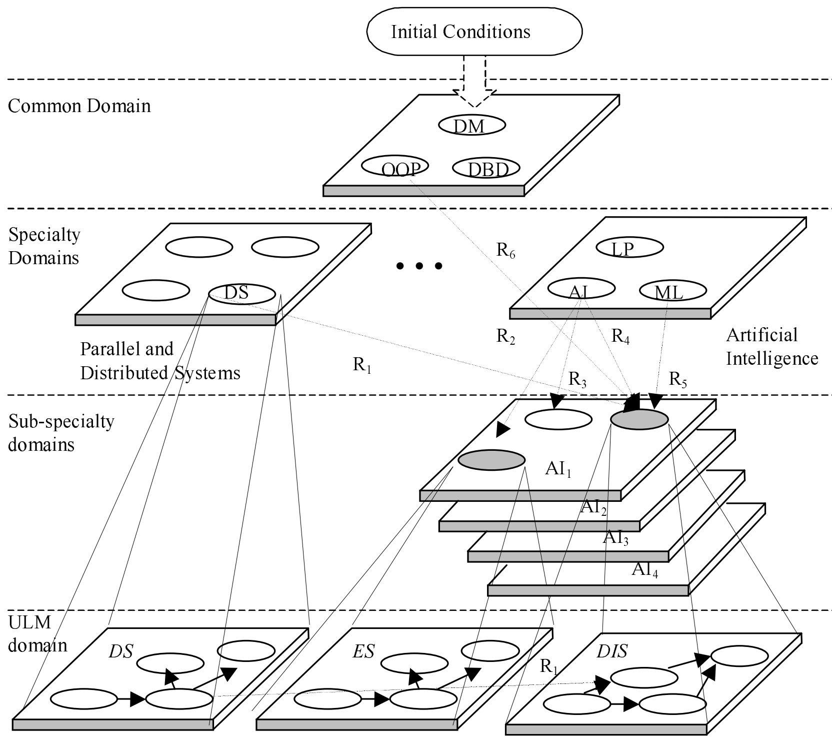


Fig. 4. Multilevel model of knowledge domains

The model of a Multi-book and all the attributes of the ULMs are stored in EVA's database and they are introduced by the professor or administrator of the system in the Professor's Environment. Other attributes of an ULM include:

- **Title** of the ULM.
- **Prerequisites:** Relate the current ULM with the basic concepts (ULM of this and other Multi-books) through the prerequisite relationships.
- **Objectives:** defined in terms of the concepts that student should understand, concepts that he should know, abilities to develop (what the student should be able to do).
- **Evaluation:** some form of exam of achievement of the objectives. There are two forms of evaluation in EVA: an automatic one and a manual one. Questions (and answers in the case of automatic evaluation) have the relationships to the concepts of the Multi-book model to infer the sources of students' misunderstandings (in the case of supported teaching). The exams and quizzes are generated based on a bank of closed questions of multiple-choice to be applied by the system

for the automatic evaluation of the student's knowledge. In the case of practical tasks, the professor puts or modifies the student's mark in the Professor's environment.

- **Experimentation:** the experimentation space composed of virtual laboratories is developed for each chapter or for the complete Multibook. If the experiments exist for the ULM, they are registered in the database, in the contrary case the entrance to the experimentation space is closed for this ULM.
- **URL** of the ULM: electronic address of the Web server, where the didactic material of the ULM is stored (it permits a physical distribution of ULMs).
- **Enabled:** this field allows the access to the students to all the chapters of the Multi-book from the moment of their assignment. In the contrary case, the access is controlled by the system based on the evaluations of each chapter and the corresponding instructional graph model.
- **Links** to the Web pages related to the current ULM.

2.2 Trajectory planning in the knowledge space

To navigate the learning spaces, a learner needs his personal routes (*study plans*) suggested in an automatic manner by EVA. So, the purpose on the planning system is to design a particular learning trajectory for each student in the learning spaces and schedule it in time. At the next stage, personalized books, called Multibooks, are armed by concatenating of selected ULM along the learning trajectory for each knowledge domain. In the same way, groups of students with similar interests are arranged.

Usually, courses pertaining to different levels of abstraction can be studied simultaneously. It means that they do not have prerequisite relations between them. Maximum time duration for each level in the global study plan is also constraint. These temporal constraints are used to schedule student’s learning activities. While selecting the alternatives, we are usually interested not in one and the only but in a number of them. An E-conformation is introduced, which means the possibility to initially accept first E alternatives. It is extremely difficult to generate learning trajectories in a general case even at the level of ULM, because of a concept graph dimensions. So, the idea of the proposed model is to capture the natural differences and similarities between the graph fragments, introducing the concept of knowledge domain.

Initial study plan and learning activities scheduling is generated on the following basis:

- Student’s initial knowledge state is detected by means of a knowledge prospecting evaluation in the common domain, which is also related to the areas of Computer Science at the graduate level according to the models, proposed by the ACM and the Mexican National Association of Institutions that teach Informatics (ANIEI) (ACM, 1991),

- Student’s interests in terms of sub-specialty or separated courses from the area, which defines student's final state in the knowledge space.

Initial student's knowledge is considered as initial conditions for the common domain, which means that students even with the same interests have different learning trajectories. The difference between the two types of final state definition is also very important for planning. The later case is more general one, because it can result in plan generation, initiated from different knowledge domains.

Planning process always starts from the final state in a backward chaining manner. Later on, each time a student pass through the exam, the system evaluates his knowledge and tries to infer the reasons of his misunderstandings. It can result in the goal redefinition and, as a consequence, reestablishment of a new study plan, taking his current state into account. Since a student is studying simultaneously a number of courses, the conditions which give rise to goal instantiation may be observed at more than one place on the general concept graph, and the same goal may be instantiated in two or more domains independently.

To perform planning and scheduling, we propose a Multi-agent System (MAS) for planning and associate a planning agent with each domain model. These agents have goal and belief representation capabilities to fulfill the task. Since knowledge domains are interconnected, their local goals and decisions are also interconnected. A multistage negotiation algorithm provides means by which an agent can acquire enough knowledge to reason about the impact of his local decisions and modify its behavior accordingly to construct a globally consistent decision. A special coordinator agent facilitates communication and performs managing activities. For details concerning planning MAS implementation see (Sheremetov & Núñez, 1999b).

3. EVA platform

When a study plan is successfully generated an EVAnaut, he can start his learning activities in EVA environment. In EVA environment, an EVAnaut has a series of services at his disposal, all of then are distributed between the spaces of learning and are composed of a database-generated Web pages created by means of dynamic and active documents.

The systems of virtual education traditionally use the client-server architecture since it is the basic principle of operation of the Web: a user (client) by means of a browser requests a service (pages HTML, etc.) from a computer that serves as a server. From this first conception of the HTTP server as a server of HTML files, the concept has evolved in two complementary directions: to add more intelligence to the server and to add more intelligence to the client. The most extended forms of adding intelligence to the clients (to the HTML pages) have been Javascript and the applets of Java. At the server side, the CGI programs and, recently the servlets that are executed at the server side as applets do at the client one, have been used for the same purpose.

The EVA environment is formed by the HTML pages, Java programs (Applets and Servlets), a number of MAS and a database (Sheremetov & Núñez, 1999a; Peredo, Armenta & Sheremetov, 1999). To customize the learning spaces for each user, we have adopted the technology of dynamic page and link generation and interactive pages using *servlets and applets*.

The dynamic Web documents don't exist in a predetermined form, but rather they are created by the server when they are requested by some client. When a query arrives, the server executes an application program that creates the dynamic document, which is returned as an answer to the client that requested it. Since a new document is created with each query and depends upon the parameters of the query, the content of a dynamic document can vary between different accesses. The active documents are generated by means of a program that is executed at the client site. When a client requests an active document, the server returns a copy of a program that is executed locally to generate the answer. By this, the program of the active document can interact with the user and change the presentation of the information continually. EVA user interface generated by EVA servlets is shown in fig. 5.

Fig. 5. EVA user interface: Multibook "Intelligent Distributed Systems"

For the client's request, a servlet communicates with the database by means of the SQL (Structured Query Language) and generates the answer as an HTML page in the client's navigator. The servlets, being programmed in Java operate with controller JDBC (Java Dates Base Connectivity) to access databases. Since the controller ODBC (Open Dates Base Connectivity) is used by Windows, a controller bridge between Java and OBDC is required, that allows the servlets to operate with the database. Java Web Server (JWS) is used to mount the servlets, which so far constitute the concept of virtual classroom and virtual laboratory providing flexible mechanism to communicate with the database manager that manipulates the entire page related information.

At the current stage of our experiments, the EVA environment makes use of a number of servers (Fig. 6) each playing different functional role.

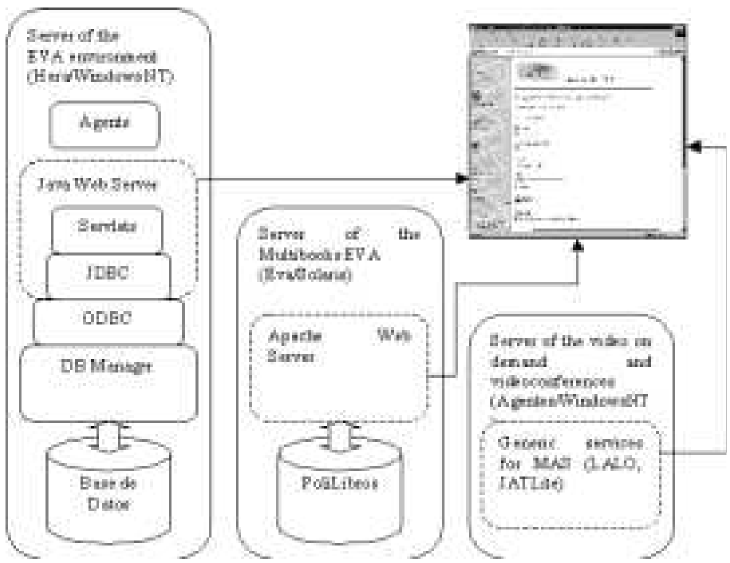


Fig. 6. Software architecture of the EVA environment

Server EVA. It uses Apache Web Server, installed in a ULTRA SUN 10 work station where the static information of the system related with the educational material included in the Multibooks, the personal information of the professors and students and the security system are managed. It serves as a gateway to the rest of the system.

Server HERA. The main server of the EVA environment. It has a JWS server installed and is dedicated to the generation of dynamic pages and transmitting them via *http* to the client machine. The server is a PC computer with microprocessor Pentium III Dual at 500 MHz. This server also serves as the repository of multiagent systems of different purposes. The database for the operation of the system is installed in this server.

Server AGENTES. It uses a JWS server installed in a PC computer with microprocessor Pentium III Dual at 500 MHz. It is devoted to store the information related with the videoconferences and video-on-demand. This server also serves as the server of multiagent systems of different purposes.

To make the architecture described above more flexible, the following modifications have been made. EVA database stores all the information about all the UMD pertaining to one or more knowledge domains controlled by the corresponding EVA environment. First, all the UMDs, which form the content of Multi-books have their own location stored in the database. It means that there is no need to have all the materials on the main server (*EVA*, for example). Second, knowledge domains are interrelated between then, so usually to study in EVA means to study Multibooks pertaining to different domains. The later is illustrated in the Fig. 3. In other words, the distributed model of the knowledge space is needed. The problem is how to integrate these parts into the common knowledge space. It is done by means of the use of domain agents carrying out trajectory planning functions (see the previous section). In the same way, the MAS of the personal assistance explained in more details in the next section, permits to generate groups of students with common interests physically connected to different EVA environments.

4. Multi-agent Systems in EVA

While implementing EVA, we have detected several problems of the traditional client-server architecture that impel the search of new solutions in the organization of environments of distributed software and the software for the Internet. One of these technologies of recent creation is the technology of agents, which seems to be a promising way to approach the problem of VLE development. The notion of agents is the central part of contemporary learning environments, where they act as virtual tutors, virtual students or learning companions, virtual personal assistants that help students to learn, mine information, manage and schedule their learning activities (Müller, Wooldridge & Jennings, 1997; Chan, 1996; Gordon, & Hall, 1998). One of the main purposes of our project is to develop models, architectures and multi-agent environment for collaborative learning and experimentation.

The core of the EVA environment consists of a number of components, composed of a set of deliberative and auxiliary agents, forming three multiagent systems (MAS): a virtual learning community, a multiagent planning system and a multiagent virtual space of cooperative experimentation. The environment of collaborative education and smart personal assistance for tutors and students, consists of several agent types listed below (fig. 7).

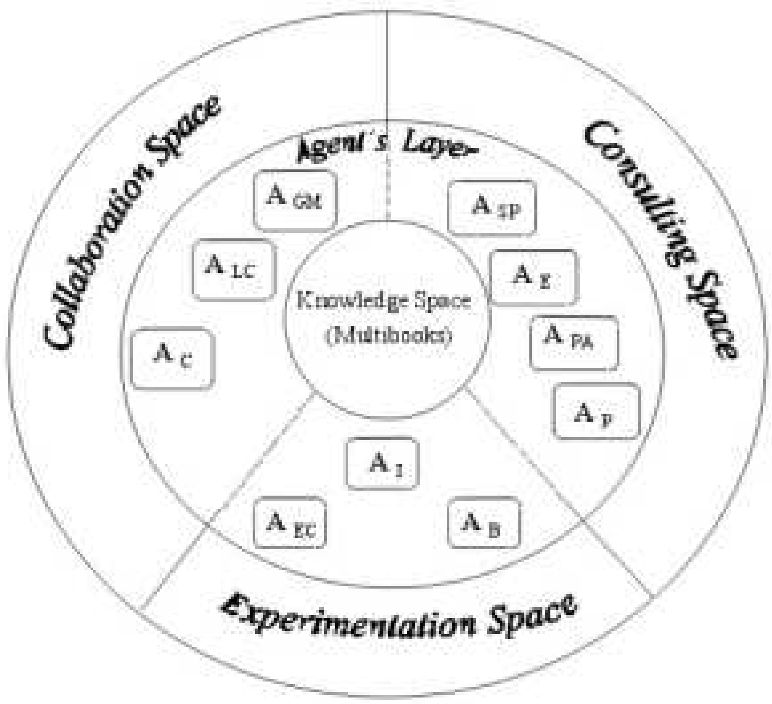


Fig. 7. EVA multi-agent conceptual architecture

In the Consulting Space, *Planning Agent (A_P)* suggests a personalized study plan to student according to its academic formation, interests, abilities and advances, and modify study plan, if it is necessary. It forms a part of Multi-agent planning system. *Internet search and filtering agent (A_{SF})*: automatically searches for additional teaching material in the Internet, not yet located in the Knowledge Space, according to student's learning trajectory and through natural language searching tools (such as Clasitex (Guzmán, 1998). Synthesizes information compiled of several sources about the same topic and filter redundant and repetitive information.

In the Collaboration Space, our prototype of learning community incorporates a *Group Monitor agent* (A_{GM}) and a *Learning companion* agent (A_{LC}). They have mental states represented in terms of beliefs, knowledge, commitments, with their behaviour specified by rules. A *Group Monitor Agent* maintains the shared knowledge model of a group and compares it with a group problem model from the knowledge base that contains the objectives, concepts, activities, etc. that characterize a group. His behavior is guided by a set of domain independent conversation rules, which refers to the interactions between the group members. During the group session, the monitor agent maintains the current problem state (shared group knowledge space) and the history record of all contributions for each participant. The problem state in terms of shared beliefs and knowledge is used to change the interaction mode, choosing one of monitoring techniques from his rule base. This result in changing behavior of learning companion from group leader (strong companion) through a week companion to a passive observer.

Companion Agents can differ in expertise level and role strategy, using 3 different pedagogical styles from supportive to non-supportive ones that can change their role in the group. Strong companions can adsorb the role of a group leaders, developing and explaining problem solution, while the weak one is used to enhance student's role in a group and stimulate his learning abilities by enforcing him to teach his companion. The conversational categories are used to classify interactions and generate learning companion's messages and are mapped into a set of communicative actions, expressed in Knowledge Query and Manipulation Language (KQML) (Finin, Labrou & Mayfield, 1997). Emotions are simply visualized by means of companion's appearance. Users interface of this collaborative environment is shown in Fig. 8.

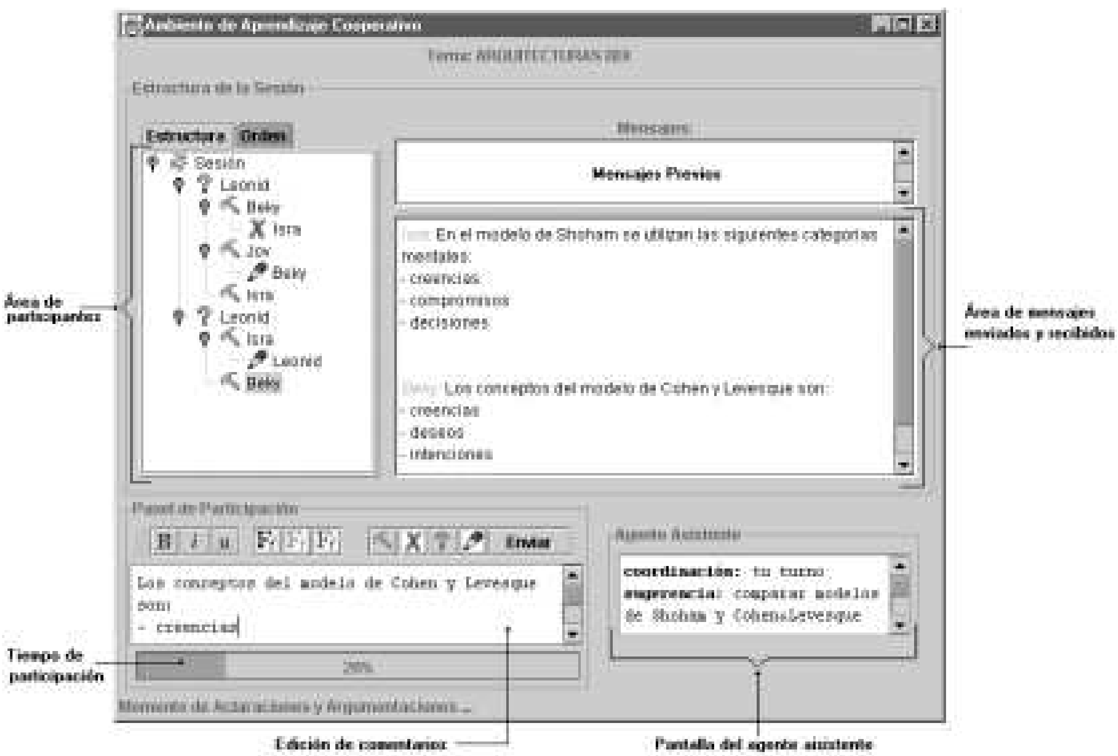


Fig. 8. User interface of the collaborative environment maintained by the Assistant Agent

In the both spaces, students and tutors receive help from their *Personal Assistants (A_{PA})*. Personal assistants (PA) is a class of intelligent agents that act semi-autonomously on behalf of a user, modeling his interests and providing services to the user or other PA's that require it. This agent communicates with the search agent to find information according to the topic, it communicates with collaboration agent to establish communication with the rest of the group members. *Personal Assistant* has the following functions:

- Helps tutors, coordinating collaborative activities. It can give privileges in the acceptance of the contributions, define the problem to discuss, among others.
- Helps students, regulating their participation in the discussion, showing coordination messages, etc.
- Implements a News system for the communication between students and professors.
- Implements a system of Electronic Agenda where stays all the information on the student's activities.
- Implements the structured Chat tool with the possibility of the users' connection registered in different EVA environments.

This software agent is implemented as Java Application and can be loaded in any computer with Java support and Internet connection. With this, we can say that each user will have a personal assistant agent according to the social paper that plays in the cooperative environment, as a professor or a student. To enter to the EVA environment, each user has to register, so it's personal data are stored in EVA's database. So, when being invoked, his personal assistant will already have a previous knowledge about the person to whom it assists and of the social paper that he plays in the environment. Two types of system Agents are also used in this MAS, a *Broker Agent* (or Directory Facilitator) and a *Wrapper Agent*. The first of them is considered a service facilitator, since for any Assistant (Professor or Student) it will look for the services provided by other agents and other assistants. Wrapper Agent can be defined as DB service provider, because it administers the EVA database with the purpose of consulting and it's modification in an explicit way.

The observations discussed above make the architecture of EVA more flexible. At the current stage of our experiments, the architecture illustrated in the Fig. 9 is used. This Fig. shows an excerpt from the model actually implemented in the IPN creating groups of users from different colleges having similar study plans. Each client uses a Web navigator to communicate with his EVA server (where he is registered). However, his personal assistant agent establish communication with the personal assistants of his working group (possibly registered to other servers) and, if needed, with the domain planning agents to create virtual distributed learning space for the particular user.

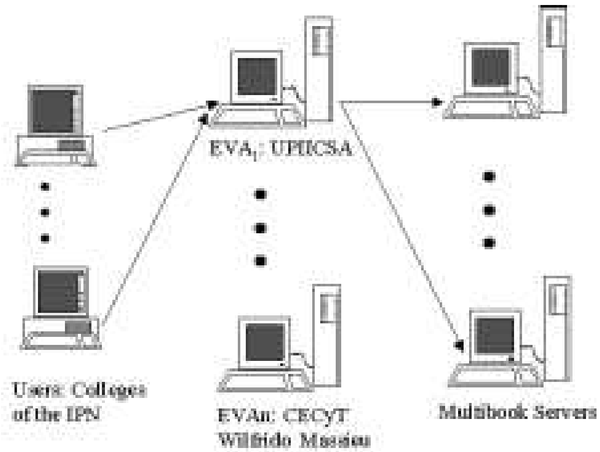
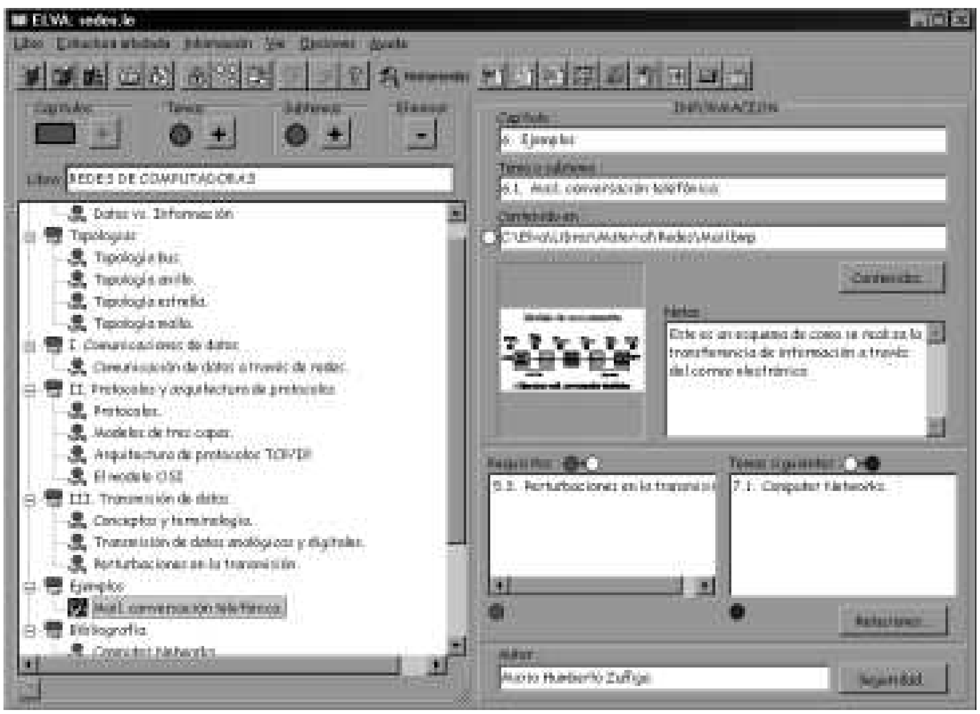


Fig. 9. EVA distributed architecture



In the Experimentation Space, the visualization of agents behavior is performed in VRML allowing generating 3D representations of scenes through a browser. We have also developed a framework of distributed software environment for the virtual education built on foundations of agents' and distributed objects technologies as the base for integration of components. For more details concerning this framework implementation see also (Sheremetov & González, 2000).

Prototypes of agents have been developed using Microsoft VC++, LALO, JAVA, and JATLite with rule-based inference capabilities, programmed in Jess (Friedman-Hill, 1998; Gauvin, 1995; JATLite, 1998). *Personal Assistant and Planning* agents are Java applications implemented with Swing package of the JDK 1.2 development environment and with the integration of a Java component for the creation of JATLiteBean agents.

5. Development technologies and tools in EVA

The EVA project is composed of a number of authoring tools for virtual teaching and learning according to the EVA methodology. In the previous section, a number of MAS and MAS development tools have been described. Two more authoring systems must be also mentioned:

1. Visual environment for the development of the Multibooks: *ELVA* - an auxiliary tool facilitating the development of learning materials, which is developed in collaboration with CENIDET (Fig. 10). This tool permits a domain expert to create a Multi-book (in a generalized form) in a user-friendly visual environment according to the Multi-book knowledge representation model described above and to publish developed materials in the server's space.

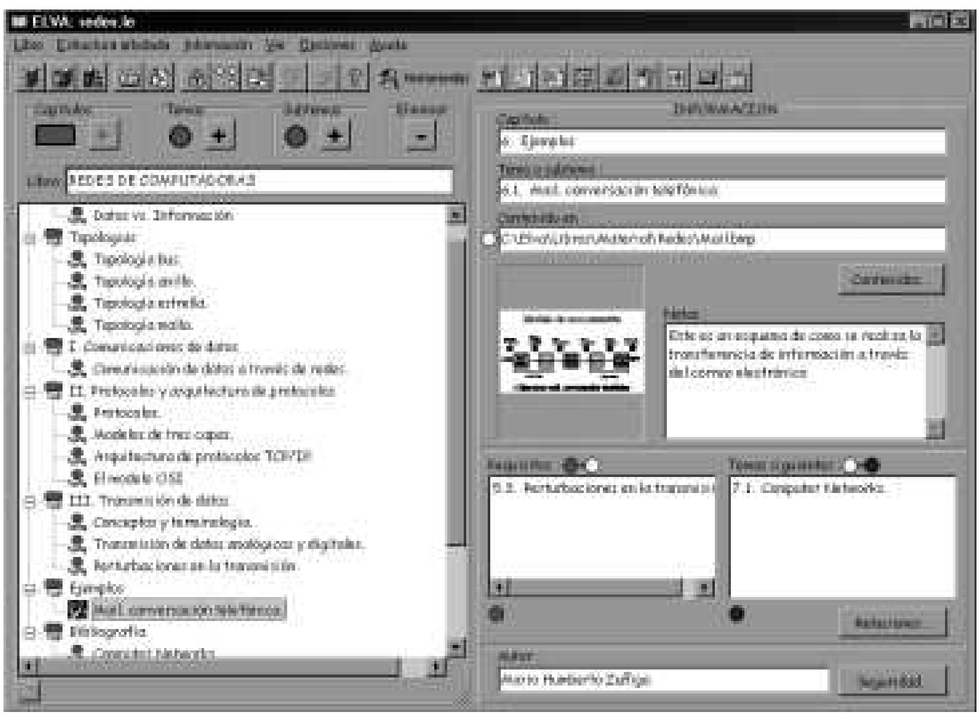


Fig. 10. ELVA user interface

2. A CASE tool for the development of virtual laboratories en Virtual Reality (VR): EASYVRML. The VRML allows to change the object and agents features of one scene (shape, color, position, size, orientation, etc) in function of determined events through the script languages, in other words to assign constraint behaviors (Mitra, Honda, Matsuda, Bell, Yu, & Marrin, 1995; Nadeau & Moreland, 1995). We've developed a CASE tool that permits to assign more complex behaviors to the VRML scenarios in a more simple form then conventional tools, using C++ and Java languages

(Quintero, Armenta, Balladares, & Nuñez, 1998) facilitating the development of virtual laboratories and simulations with VR. EASYVRML, is based in:

- Visualization algorithms and three-dimensional manipulation of objects in the WEB.
- Model of representation of objects in Distributed Virtual Environments (DVE).
- Model of behavior sharing and interaction in DVE.
- Model of mechanisms based in multi-casting for the solution of support of DVE for VRML.
- Models for the representation of intelligent agents in virtual environments on the WEB.

This tool has also been used for the visualization of agents behaviour in the VR scenarios.

6. Current experiments and conclusions

At the moment of writing, the first version of EVA is in the implementation and testing phase. The following results have been obtained:

1. In the fall semester of 1998, the EVA environment was first used to impart the seminar of Software Agents in the CIC. The server was accessed by near 20 students from the PC's of the LAN of the CIC and from remote computers from the periphery of Mexico City, from an approximate distance of 50 miles. Since then, the EVA environment and the Multibooks have been used by the professors of the CIC as an additional didactic support in the Postgraduate courses in Computer Sciences.
2. From October 15, 1999 to February, 2000, the first International course in EVA called "THE DIGITAL DOCUMENTATION IN MEANS OF SOCIAL COMMUNICATION" was imparted jointly by the Center of Training of Educational Television (CETE) of the General Direction of Educational Television of Mexico and the Complutense University of Madrid, with the participation of students from Mexico, Colombia, Argentina and the Dominican Republic.
3. Starting from October 1999, under the supervision of the Coordination of Academic Computation of the IPN, the development of the institutional project "INSTALLATION OF THE MODEL AND SYSTEM EVA (VIRTUAL SPACES OF LEARNING) IN THE IPN" has begun. This project has as objective to support the educational modernization of the IPN, taking advantage of the methodology and the EVA software environment, within the framework of the Program on Educational Technology for Academic Innovation. The EVA environment is installed at the servers of diverse schools and colleges of the IPN, as, for example, UPIICSA, CECyT Wilfrido Massieu, CECyT 6, which will serve as nodes in the training network and in developing of the corresponding Multibooks.

4. From October of 1999, the tools for the experimentation space of EVA as a platform of Distance Education at the Master level of Mathematical Education of the Autonomous University of the State of Hidalgo, Mexico has begun. To the date, a number of virtual laboratories for the basic courses of this career have been developed.
5. Now, the development of the project on the implementation of EVA as a platform of distance education in the system that integrates about one hundred of Technological Institutes in Mexico has also begun. For details see an accompanying paper in this issue (Guzmán & Sheremetov, 2000).

EVA is a rich learning environment using conventional software and agents with the following innovative characteristics:

- It considers in an integrated manner all the methods and channels, by means of which students acquire information and communicate to learn, to exchange knowledge, to solve problems, to experiment and, in general, to realize academic activities.
- It also contemplates the development of the environment capable of delivering to the user in an integrated and virtual way the main services, work and entertainment spaces usually offered by the institutions of high education, including the interactions outside of the classroom.
- The environment is dynamically configured, generated and delivered through the user's WEB browser based on the models stored in the database.
- This environment makes intensive use of agent technology, where agents plan, configure, evaluate, help and administer the learning process.
- The knowledge is represented in a multimodal way, i.e. logic, linguistic, graphic and/or animated, audio, to facilitate its communication and assimilation by students with different learning styles.
- The technological platform is developed permitting the integration of EVA environments in the virtual network of learning spaces.

Our experiments to-date have demonstrated the high flexibility for the EVA MAS system as well as the interoperability with non-agent software modules. We have also detected some difficulties while working with the proposed architecture. For example, all the agents and protocols must exist (be active or able to be initialized automatically) for a service to be provided. Also, working with different databases, we have detected some problems when accessing them with JDBC drivers on WindowsNT platform. Nevertheless, our expectation is that as the set of agents grows, the development will be easier.

The work on virtual learning community, composed of learning companions and personal learning assistants is in process. We are also working on animation assignation to learning companions and tutors (especially in the experimentation space) to obtain their strong visual presence in a 3D learning environment. We have developed a number of prototypes but a lot of work is still to be done to convert them into a real VLE.

Acknowledgements

The EVA project is the effort of many people. The authors would particularly like to thank their colleagues and students of the Agents Laboratory for their determination and contribution of ideas and talent to EVA programming. This work was supported in part by REDII CONACyT and CGEPI IPN.

References

ACM, (1991) Computing as a Discipline, ACM report.

Chan, T.W. (1996) Learning companion Systems, Social Learning Systems, and Intelligent Virtual Classroom, In *Proc. of the World Congress on AI in Education*, 1996, Vol. 7, No. 2, 125-159.

Collis, B. (1999) Applications of Computer Communications in Education: An Overview. *IEEE Communications Magazine*, March 1999, 82-86.

Finin, T., Labrou, Ya., & Mayfield J. (1997) KQML as an agent communication language. In J. Bradshaw, (Ed.), *Software Agents*, AAAI/MIT Press, 1997.

Friedman-Hill, E. J. (1998), Jess, The Java Expert System Shell, Version 4.3, Technical report SAND98-8206 Sandia National Laboratories, Livermore, CA, http://herzberg.ca.sandia.gov/jess.

Gauvin, D. (1995).Un environnement de programmation orienté agent. *Dans Troisièmes journées francophones sur l'intelligence artificielle distribuée et les systèmes multiagents*, France.

Gordon, A. & Hall, L. (1998) Collaboration with Agents in a Virtual World. *In Proc. of the Workshop on Current Trends and Artificial Intelligence in Education, 4 World Congress on Expert Systems*, Mexico-city, Mexico, 1998.

Guzmán A. (1998). Finding the main themes in a Spanish document. *Journal Expert Systems with Applications*, 14(1-2): 139-148.

Guzmán, A., & Nuñez, G. (1998) Virtual Learning Spaces in Distance Education: Tools for the Eva Project. *International Journal Expert Systems with Applications (Special Issue)*, Pergamon Press, 15: 205-210.

Guzmán, A. & Sheremetov, L. (2000) Applying EVA to tailored virtual, distance, on-line education via Internet. In this issue.

JATLite Beta Complete Documentation (1998), Stanford University, http://java.stanford.edu

Hamouda A.M.S. & J.P. Tan. (2000) Development of Virtual Classroom for Interactive Education on the Web, In *Proc. of the International Conference in Digital University*, 16-17 May 2000, KL, Malaysia.

Mitra, Ya. Honda, K. Matsuda, G. Bell, Ch. Yu, & Ch. Marrin (1995) Moving Worlds: behaviors for VRML, http://www.vrml.sgi.com/spec.html

Müller, J., Wooldridge, M. J. & Jennings, N. R., eds. (1997) Intelligent Agents III: Proc. of the 3rd Int. Workshop on Agent Theories, Architectures, and Languages, *Lecture Notes in Artificial Intelligence*, Vol. 1193, Springer-Verlag.

Nadeau R. D., Moreland L. J., (1995) The Virtual Reality Behavior system (VRBS): A Behavior Language Protocol for VRML, San Diego SuperComputer Center. http://www.sdsc.edu/projects/vrml/vrbs_proto.html.

Nuñez, G., Sheremetov, L., Martínez, J., Guzmán, A., De Albornoz, A. (1998) The EVA Teleteaching Project - The concept and the first experience in the development of virtual learning spaces. In Gordon Davies (ed.) *Teleteaching'98 Distance Learning, Training and Education: Proceedings of the 15th IFIP World Computer Congress "The Global Information Society on the Way to the Next Millennium"*, Vienna and Budapest, 31 August - 4 September 1998, Part II, pp. 769-778.

O´Malley, C. (1995) *Computer Supported Collaborative Learning*. Springer Verlag.

Peredo R., Armenta A. & Sheremetov L. (1999) Arquitectura Computacional para Sistemas de Aprendizaje Colaborativo, In Proc. of the *Simposium Internacional CIC'99*, 15-19 de noviembre 1999, pp. 628-636. (in Spanish).

Quintero, R., Armenta, A., Balladares, L. & Nuñez G. (1998) EasyVRML Herramienta CASE para la asignación de comportamiento a componentes de mundo VRML. In Proc. *XI Congreso Nacional ANIEI*, Mexico, 1998.

Sheremetov, L. & González, J. (2000) Desarrollo y Construcción de una Plataforma Multiagente Basada en el Modelo de Referencia de FIPA y DCOM, In Proc. of *Simposium Internacional de Sistemas Distribuidos Avanzados (SISDA)*, CD ROM, 2000, (in Spanish).

Sheremetov, L. & Núñez, G. (1999a) Multi-agent framework for virtual learning spaces. *Int. Journal of Interactive Learning Research (JILR)*, 10(3-4):301-320.

Sheremetov, L. & Núñez, G. (1999b) Multi-Stage Cooperation Algorithm and Tools for Agent-Based Planning and Scheduling in Virtual Learning Environment, In Proc. *of The First International Workshop of Central and Eastern Europe on Multi-agent Systems (CEEMAS'99)*, 30th May-3rd June 1999, St. Petersburg, Russia, pp. 211-223.

Youngblut, Ch. (1994) Government - Sponsored Research and Development Efforts in the Area of Intelligent Tutoring Systems. *Institute for Defense Analyses*, USA.