INTERLABS: The Web-Lecturing Multiagent-Based Tool

Dr. Vladimir Uskov  
Computer Science Department, Bradley University  
1501 W. Bradley Avenue  
Peoria, IL 61625, U.S.A.  
(309) 677-3492  
Uskov@bradley.edu

Dr. Leonid Sheremetov  
Jefe del Laboratorio de Agentes  
Centro de Investigacion en Computacion  
IPN, Mexico, D.F.  
(52 5) 729-6000, ext. 56576  
Sher@cic.ipn.mx

ABSTRACT
The INTERLABS state-of-the-art multiagent-based Web-lecturing tool of the third generation has been designed, developed, and tested at the Department of Computer Science and Information Systems, and NSF-Bradley Center on Web-Based Education at Bradley University, IL, U.S.A.

Keywords  
Artificial Intelligence (AI) Applications in Education, WWW-based education, multiagent-based Web-lecturing tool, WWW-based courseware.

1. INTRODUCTION

Teaching and learning online through the Internet and World Wide Web (WWW) are bound to change the structure of traditional educational institutions. Market of online education and training estimated to be worth $19 million in 1995, is projected to be about $17 billion by 2006.

The vast majority of the U.S. educational institutions actively use various WWW-based instructional tools (WBI tools) of the 1st and 2nd generations for a design and development of WWW-based curricula and courseware. However, the leading institutions such as Stanford University, Carnegie Mellon University, University of Massachusetts, University of Michigan - Ann Arbor, recently designed, developed, and started to actively use the WBI tools of the third generation [1] in order to support the WWW-Based Education (WBE) in their institutions.

This motivated faculty and students of the Computer Science and Information Systems Department at Bradley University, and members of the NSF-Bradley Center on Web-Based Education to propose, design, develop and test the INTERLABS tool - multidisciplinary multifunctional Web-lecturing tool of the third generation.

2. CURRENT INTERLABS TOOL

Currently, the INTERLABS Web-lecturing tool (Figure 1) contains most of the usual features and functions of the Web-lecturing tools [2]. However, the INTERLABS tool also offers the additional, attractive for both students and faculty, features such as:

1) multiple "student-to-student" and "student-to-teacher" communication functions such as "call an audio-conferencing", "call a video-conferencing", "call a data-conferencing", "share an application", "send an email", "join a discussion group", "join a news group", etc., and
2) control functions to download entire course and/or a set of modules of a course from the Internet (e.g., the Interlabs Server at the InterLabs Research Institute, Bradley University).

3. INTERLABS TOOL AND ACTIVE LEARNING PARADIGM

Another distinctive feature of the INTERLABS tool is that it may be used in both synchronous and asynchronous modes,  
1) in online mode it uses the Microsoft Windows Media Player Active X control, Microsoft PowerPoint Player, Java Script, frames, Microsoft NetMeeting SDK, and several other tools, and  
2) in offline mode it uses the Microsoft Windows Media Player Active X control, Microsoft Visual Basic, and other tools.

Figure 1. Interface of the INTERLABS Web-lecturing tool.
The fundamental concept of INTERLABS tool is to truly support the active learning paradigm of education. The traditional approaches of learning have been questioned recently about their ability to provide the learner with “rich” rather than “minimalist” environments and with “authentic” learning experiences.

Active learning became a modern paradigm of learning. It is a process rather than a product. Active utilization of the INTERLABS WBI tool by students strengthens the advantages of active learning, specifically

1) student responsibility and initiative to promote ownership of learning and transferable skills,
2) intentional learning strategies, explicit methods of learning, reflection of learning processes, metacognitive skills,
3) goal-driven problem solving tasks and projects generating learning products of value,
4) teachers as facilitators, coaches and guides, not sources of knowledge, that requires discussion between teachers and learners,
5) authentic contexts for learning, anchored in real-world problems,
6) authentic assessment strategies to evaluate real-world skills, and
7) cooperative learning.

4. MULTIAGENT STRUCTURE

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 [3, 4, 5, 6]. The agent technology has a great potential to assist on construction of rich learning environment providing support for active learning paradigm. Using agents, the INTERLABS virtual learning community may include various a) pedagogic models, b) teaching and learning methodologies, c) forms of knowledge, d) collaboration and communication technologies, e) individual and team-working evaluation and assessment technologies of students/learners that are separated “in space and time”.

At the current stage of our experiments, Agent Platform (AP) and a multiagent system of Personal Assistants (PA) composed of a set of deliberative and utility agents have been developed. AP provides the framework of normative work, inside which the agents exist and operate, as well as the logical and temporal contexts for the creation, operation and destruction of agents [7]. It consists of following agent types:

1) Agent Management System (AMS). The AMS is responsible for managing activities of agents of an AP. For this purpose, it maintains a permanent list of all resident agents. This list includes, at least, the agent's unique name and its address. Each AP must have only one AMS.

2) Domain facilitator (DF). DF is responsible for service registration (roles) of PA and utility agents (yellow pages). It also facilitates communication between them (establishing a peer-to-peer connection). The DF maintains a list of services that each agent provides, and additional characteristics such as, agent's -- type, state and owner. For each service, its name, type, and ontology are stored. Agents communicate with the DF, either to register their own services, or to find out about services that other agents have to offer. A group of agents registered in a DF is known as agent domain. At least one DF exists for each INTERLABS server.

3) Agent Communication Channel (ACC). The ACC is a way by which an agent is able to communicate with other agents, including the AMS and the DF. Thus, each agent should have access to at least one ACC. The ACC not only routes messages inside a platform, but is also capable of routing messages to an ACC in another platform. Each client computer connected to INTERLABS server has ACC, which enables communication between user's PAs and utility agents.

4) Wrapper agent (WA). This agent facilitates access to the MAS and enterprise databases, as well as to the legacy software (e.g. courseware software).

5) Ontology Management Agent (OMA). This agent's responsibility is to enable communication between INTERLABS servers on the knowledge level. The system is supported by a number of ontologies: a) Service Ontologies (student model, education strategies, pedagogic models), b) Education Domain Ontologies, 3) Ontology of administration and management (FIPA management), and d) Ontology of roles (student, teacher, group monitor, administrator, etc.)

A multiagent system of PA is mounted on the AP. 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. At the moment of writing, PA play the following roles, implemented as plugins over the PA's kernel (each user can configure easily his PA agent):

1) Group Monitor Agent implements environment for collaborative problem solving. Helps tutors, coordinating collaborative activities. It can define the problem to discuss, give privileges in the acceptance of the contributions, suggest solutions, among others. Helps students, regulating their participation in the discussion, showing coordination messages, etc.

2) News Agent implements a News system for the communication between students and professors.

3) Agenda Agent implements a system of electronic agenda where all the information on the student's activities is stored.

4) Chat Agent implements the structured chat and FAQ tools with the possibility of the users' connection within one or different INTERLABS servers (Figure 2).
Agents have mental states represented in terms of beliefs, knowledge, commitments, with their behavior specified by rules. For example, 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.

Figure 2. Interface of the PA environment

Personal assistant is Java application implemented with JESS (rule-based reasoning) and Swing package of the JDK 1.2 development environment and can be loaded in any computer with Java support and Internet connection. With this, we can say that each user will have a PA agent according to the social role that plays in the cooperative environment, as a tutor or a student. The personal data for each user are stored in INTERLABS database, so, when being invoked, his PA will already have a previous knowledge about the person to whom it assists and of the social role that he plays in the environment. For any PA (teacher's or student's) a DF looks for the services provided by other agents and other assistants. WA is used in this case as DB service provider, because it administers the INTERLABS database with the purpose of consulting and its' modification in an explicit way. All the agents use Knowledge Query and Manipulation Language (KQML) for communication [8].

AP and PA multiagent system play a key role in enabling INTERLABS distributed virtual campus Each INTERLABS client uses a Web navigator to communicate with one of site servers (where he is registered as a user). 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 shared learning space.

PA agents have a modular structure. In the final version of INTERLABS WBI tool, they will play the following roles:

1) **Instructor Agent** that provides planning, design, development, management, modification, etc. of WBE courseware.
2) **Evaluator Agent** that is composed of diagnostic rules and provides student’s model by analyzing the student’s behavior, student’s self-assessing, progress tracking, building of online study skills, etc.
3) **Management Agent** that provides management functions such as automated calendaring and automatic reminding, student activity reports, student class lists, etc.
4) **Communication Agent** that provides various communication services by connecting student(s) and teacher, including a) chat rooms, b) video-, audio- and data-conferencing, c) a multi-media player for streaming video- and audio-, d) white board/document sharing, e) email management, f) bulletin boards, g) news groups, etc.
5) **Courseware Search Agent** that will provide automatic search of the Internet on additional course-related information and multimedia fragments.
6) **Group Agent** that will evaluate and enroll students into the course based on student academic performance.

5. ACKNOWLEDGMENTS

The authors of this paper would like to thank the National Science Foundation (NSF, U.S.A.) for partial support for this work within the NSF CCLI-EMD grants # DUE-0196015 (2000-2003) and # DUE-0002219 (2000-2001). Partial support for this work was also provided by CONACyT (Mexico) within the project 31851-A (2000-2002) and IPN within the project 980739 (1998-2001). The authors would like to acknowledge Mr. Alexander Uskov, CS/IS student at Bradley University, for a design and development of his part of the InterLabs tool software.

6. REFERENCES


