A multiagent architecture to support distance learning personalization on the Web

J. G. Boticario **E. Gaudioso** *
Dpto. de Inteligencia Artificial
Universidad Nacional de Educación a Distancia
Senda del Rey, 9
Madrid, Spain
e-mail: {jgb,elena}@dia.uned.es

Abstract

This article presents the description of the objectives, the structure and the functionality of an interactive system intended to focus the teaching on the performance of the student and to resolve problems detected in Internet use for distance learning. The system adapts to the information and communication needs of the different types of users. This adaption is done through user model acquisition from the data available on the students and user interaction with the system. WebDL, the system we have developed, is the result of an effective combination of techniques used in intelligent tutoring systems, adaptive hypermedia programs and learning apprentice systems for software personalization. In the initial stages, WebDL is being used for the personalization of the presentation of the exercise part of the machine learning courses at UNED (The Spanish National University for Distance Education) Computer Science School.

Introduction

One of the features that characterizes distance education is the "systematic use of communication media and technical support" (Keegan 1982) to mediate in learning experiences. In any theory about learning it is stressed that the quality of the communication between teacher and student is a decisive factor in the process. The widespread use of the Web in distance learning could help to satisfy the need for information and to mitigate the isolation that characterizes the student in this domain. However, considering the student diversity which characterizes this kind of education as well as the dispersion of the information sources (news, mailing lists, ...) the development of any kind of interactive systems, able to adapt to the information and communication needs of each student, would be of great help.

In order to solve the problems that characterize distance learning on the Web, we have constructed a multiagent architecture that is intended to be adaptable to the user's needs (Boticario & Gaudioso 1999) based on a combination of techniques applied in intelligent tutoring systems (ITS) (Weber & Specht 1997), adaptive hypermedia programs (AH) (Brusilovsky 1996) and learning apprentice systems (LA)(Dent *et al.* 1992). It falls into the category of so-called *Web-based Adaptive Educational Systems* (Brusilovsky 1998).

Considering the advantages of collaborative learning, the multitude of separate tasks involved, the unpredictability of the result and the dispersion of the resources we have opted for a multiagent architecture; specifically, we have chosen a multiagent decision system. The language used in the communication between agents is KQML (*Knowledge Query and Manipulation Language*) (Finin *et al.* 1994) and the content of the messages is represented in KIF (*Knowledge Interchange Format*) (Genesereth & Fikes 1992). The concepts exchanged in the messages depend on the ontologies used, in this case, educational ontologies (Chen & Mizoguchi 1999).

WebDL: A Personalized Distance Learning Interactive System

As a result of the combination of techniques applied (ITS, AH, LA), the system performs tasks having the following goals: correct the user's behavior (ITS) and provide effective support for the different decision tasks on the Web (LA and AH). The intention of using the learning apprentice approach is to expand the initial knowledge base in order to reduce the effort required in user decision-making (adaptive hypermedia).

WebDL is designed to be used for accessing educational services available on the Internet; it is transparent to the student and no additional specific software is required. It is based on dynamically-constructed HTML pages that interact with the user according to his/her needs. WebDL also makes use of a XML version of the same pages in order to separate document contents, the presentation itself, the page behavior and the document hypertext structure. This technique allows the local user to interact with the document with no server connection (forms which check whether they have been completely filled in before being sent to the server, tests with on-line help given in function of the user's response, etc.).

*PhD grant from UNED

Architecture

The system provides: a distributed system architecture, a communication protocol used by the agents, a distributed learning algorithm, a conflict resolution mechanism and a distributed decision-making algorithm.

The system is implemented in terms of a multiagent decision approach and is organized as follows. Two main components are involved: the user interaction and the adaptive module (see figure 1) . The first is implemented by the *interface agent* and is in charge of the organized presentation of the different types of material designed to achieve the highest *usability*. This module provides a single, integrated response to the user.

The adaptive module is composed of the following agents: user model agent, user modeling agent, material agent, pedagogical agent, contact agent, service identification agent, service model agent, service modeling agent and coordinator agent. The first four provide the basic functionality of an ITS. The next four are useful for identifying, by *collaborative filtering*, the system services that are of interest to users with similar profiles. Finally, the coordinator agent broadcasts a problem instance description (user's request) to all the agents involved in that problem. This agent is also in charge of constructing a single response for the interface agent.

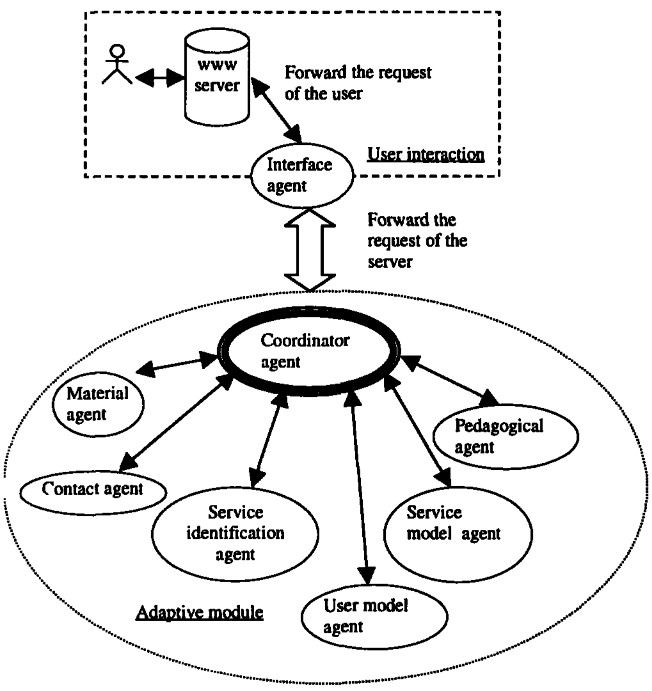

Figure 1: System's general architecture

In more detail, the adaptive module consists of a dynamic multiagent system (agents join or leave the system) with weak coupling (they may or may not participate in each task). We have chosen heterogeneous agents in order to combine the solutions learned with different bias and corresponding to different generalization methods: C5.0 (Quinlan 1997) , Naive Bayes (Smith 1988) , Progol (Muggleton 1995) , Backpropagation (Rumelhart, Hinton, & Williams 1986) , Tilde (Blockeel & Raedt 1997) and Autoclass (Cheesman *et al.* 1990) for clustering. To implement this combination a special type of agent has been introduced: the advisor agent. Several advisor agents learn the competence range of each of the other agents. A task is only distributed to those agents that have been proved to be competent in that task. In this manner we aim to establish a gradual process of agent specialization with concrete architectures for specific problems (see figure 2).

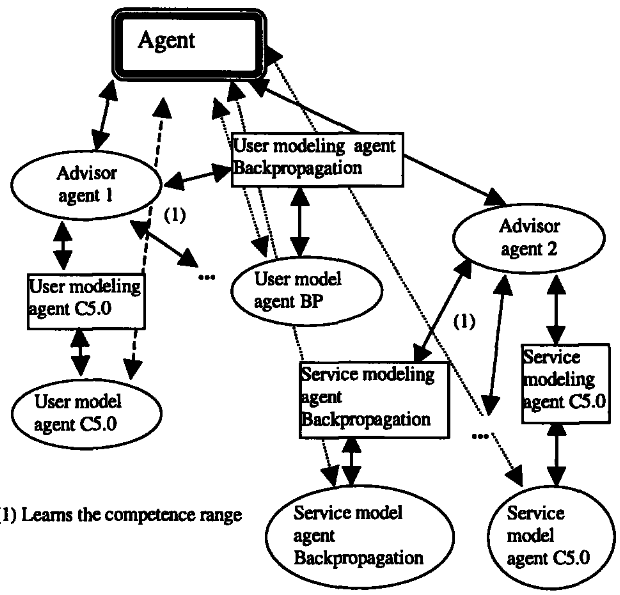

Figure 2: Two advisor agents learn the competence range of the modeling agents

We distinguish two phases: application of learned knowledge and adaptation to the individual user through learning tasks. In the first, the interface agent makes a request to the coordinator. This agent, depending on the request, asks the advisors for the agents competent in the tasks involved in that request (the intention here is to improve on other approaches that distribute the same task to every agent in the system (Giraldez, Elkan, & Borrajo 1999)). In the second phase, when the interaction with the user has finished, using the available training examples (those that have been collected), the advisor agent learns the competence of those agents involved in the adaptation task: the model agent and the service agent. The learning task of the user model and that of the service model are implemented using the different generalization paradigms that constitute the various modeling options: service modeling agent and user modeling agent. The service identification agent selects the services that interest a significant number of users through *clustering* techniques; the intention here is for the system to learn the characteristics determining which services may be of interest for a given user.

Knowledge base

Initially, WebDL has a representation of all the domain entities together with their relations. It also has static information on the connecting student (personal data and academic record taken from the University database). All these entities are grouped in what we call *knowledge base* or *system ontology* (the concepts exchanged in the messages between system agents depend on the ontologies used, which in this case are educational ontologies (Chen & Mizoguchi 1999)) .

Taking into account our previous experience in the construction of apprentice systems (Dent *et al.* 1992), and given their effectiveness, we have developed the knowledge base of our system on THEO: an *integrated architecture* providing a generic programming environment in which different knowledge and inference representation mechanisms can be combined.

The domain entities considered are grouped into the following categories: system structure, the people it interacts with and the elements it handles (course material, concept representation of the course content ...).

Notice that the agents with learning tasks are able to expand their knowledge base dynamicaclly, adding and updating entities as the user interacts with the system (so called learning apprentice approach).

The learning tasks used to model the user modify and extend the knowledge-base entities describing the interests, preferences, skill-level ... of the person connecting to the system. They also modify and extend all those entities referring to the material elements which can be personalized (content pages, pages enhanced with annotations ...).

The elements which are not personalized (bibliographical data, system structure, node network structure ...) are not modified or extended dynamically by the system. Nonetheless, those elements considered to be static and dynamic will change in the next development phase of the system, thereby allowing personalization of new elements, such as the envisaged automatic detection of new bibliographical elements from pages accessed by the student (Craven *et al.* 1998).

User interaction

WebDL is designed to provide new information, according to the user's needs (in terms of dynamically generated web pages which in many cases resemble web portals), and new communication channels of interest to him/her (news, shared workspaces, contact with students with similar characteristics ...), during each session.

The user is aware of the dynamic aspect of the interaction through: different presentations of the same information (project, hide, annotate), changes in the order in which interface elements appear, changes in the level of interaction (in accordance with the inferred user skill level) and different information descriptions.

The user interface has two different working areas when the advice given by the system cannot be integrated in the information presented or when the advice information is contextual and has no effect on the information shown (see figure 3) . The system keeps two different interaction traces, one for each area in the interface. When the interface agent makes a request to the coordinator agent, the latter agent divides and stores the previous traces.

In order to differentiate the elements of each area, certain parameters are added to the links of those elements.

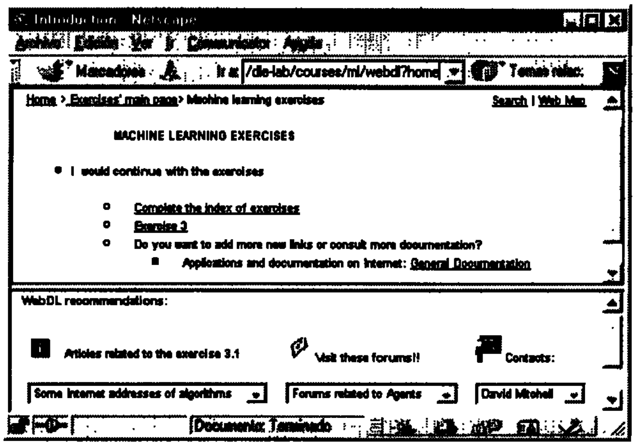

Figure 3: The two work areas of the user interface

State of development of the project

Currently, the basic architecture of the system has been implemented and it has been applied to the access to the course material of the machine learning course in the Computer Science degree of the Computer Science School of the UNED.

The system tasks that are planned are: curriculum sequencing, intelligent analysis of student solutions, interactive problem solving support, example-based problem solving, adaptive presentation, adaptive collaboration support and adaptive navigation support. Those which have been implemented are: curriculum sequencing (of the course material), adaptive collaboration support (of the course practical exercises), adaptive presentation and adaptive navigation support (for every page constructed by the system).

At present, we have only experimented with potential users; these experiments have validated the proposed architecture in the adaptation tasks previously described. To obtain preliminary results concerning the effectiveness of the system a survey has been conducted with students who have used WebDL and the results obtained up to now reveal that the users obtain the information they require in the first navigation steps, each user's ideal contact medium is detected and the system provides adequate assistance in the navigation and use of the resources of the web-site. A thorough evaluation will be carried out when sufficient data is available at the end of the current academic year.

References

Blockeel, H., and Raedt, L. 1997. Top-down induction of logical decision trees. In *Proceedings of the 9th Dutch Conference on Artificial Intelligence NAIC97.*

Boticario, J. G., and Gaudioso, E. 1999. Towards personalized distance learning on the web. In Mira, J., and Sanchez-Andres, J., eds., *Foundations and Tools for Neural Modeling*, number 1607 in Lecture Notes in Computer Science. Springer Verlag. 740–749.

Brusilovsky, P. 1996. Methods and techniques of adaptive hypermedia. In *User Modeling and User-Adapted Interaction*, 87–129. Kluwer academic publishers.

Brusilovsky, P. 1998. Adaptive educational systems on the world-wide-web: A review of available technologies. In *Proceedings of Workshop WWW-Based Tutoring at Fourth International Conference on ITS (ITS'98).* San Antonio, TX: MIT Press.

Cheesman, P.; Stutz, J.; Hanson, R.; and Taylor, W. 1990. *Autoclass III.* NASA Ames Research Center: Rsearch Institute for Advanced Computer Science.

Chen, W., and Mizoguchi, R. 1999. Communication ontology for learner model agent in multi-agent architecture. In *Proceedings of the International Conference on Artificial Intelligence in Education (AI-ED99).*

Craven, M.; DiPasquo, D.; Freitag, D.; McCallum, A.; Mitchell, T.; Nigam, K.; and Slattery, S. 1998. Learning to extract symbolic knowledge from the world wide web. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence.*

Dent, L.; Boticario, J. G.; McDermott, J.; Mitchell, T. M.; and Zabowski, D. T. 1992. A personal learning apprentice. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, 96–103. San Jose, CA: MIT Press.

Finin, T.; Fritzson, R.; McKay, D.; and McEntire, R. 1994. Kqml as an agent communication language. In *Proceedings of the Third International Conference on Information and Knowledge Management*, 64–69. ACM Press.

Genesereth, M., and Fikes, R. 1992. Knowledge interchange format, version 3.0 reference manual. Technical Report KSL-92-86, Knowledge Systems Laboratory.

Giraldez, J. I.; Elkan, C.; and Borrajo, D. 1999. A distributed solution to the pte problem. In Gini, G. C., and Katritzky, A. R., eds., *Predictive Toxicology of Chemicals: Experiences and Impact of AI tools, Papers form the 1999 AAAI Spring Symposium, TR SS-99-01*, 82–85. AAAI Press.

Keegan, J. D. 1982. From New Delhi to Vancouver: trends in distance education. In *Learning at a Distance. A world perspective.* Athabasca University, Edmonton: International Council for Correspondence Education.

Muggleton, S. 1995. Inverse entailment and progol. *New Gen. Comput.* 13:245–286.

Quinlan, R. 1997. *C5.0 Data Mining Tool.* www.rulequest.com.

Rumelhart, D. E.; Hinton, G. E.; and Williams, R. J. 1986. Learning internal representations by error propagation. In Rumelhart, D. E., and McClelland, J. L., eds., *Parallel Distributed Processing: Explorations in the Microstucture of Cognition.* Cambridge, MA: MIT Press.

Smith, J. 1988. *Decision Analysis, A Bayesian Approach.* Chapman and Hall.

Weber, G., and Specht, M. 1997. User modeling and adaptive navigation support in www-based tutoring systems. In *Proceedings of the Sixth International Conference on User Modeling*, 289–300.