



Navigating through the RLATES Interface: A Web-Based Adaptive and Intelligent Educational System

Ana Iglesias, Paloma Martínez, and Fernando Fernández

Computer Science Department
University Carlos III of Madrid
Avda. de la Universidad, 30, 28911-Leganés (Madrid), SPAIN
Tel: 34-91-624{9421, 9454, 9105}, Fax: 34-91-6249430.
{aiglesia, pmf, ffernand}@inf.uc3m.es

Abstract. The paper shows the architecture of the RLATES system, an Adaptive and Intelligent Educational System that uses the Reinforcement Learning model (RL) in order to learn to teach each student individually, being adapted to their learning needs in each moment of the interaction. This paper is focused on the interface module of RLATES, describing how the student could navigate through the system interface and how this interface adjusts the page contents according to the user learning needs. For this adaptation, the system changes the links appearance of the page and the presentation of the system knowledge.

1 Introduction

Web-based education (WBE) is currently a hot research and development area. They have two useful benefits: classroom independence and platform independence. Traditional web-based courses usually are static hypertext pages without student adaptability, providing the same page content and the same set of links to all users. However, since last ninetieths, several research teams have been implementing different kinds of adaptive and intelligent systems for WBE [1].

Web-based Adaptive and Intelligent Educational Systems provide intelligence and student adaptability, inheriting properties from *Intelligent Tutoring Systems (ITS)* and *Adaptive Hypermedia Systems (AHS)*. These systems are the most complete up to now, because the domain knowledge, the student model and the pedagogical strategies used to individually adapt to the student are studied by Intelligent Tutoring Systems, and the application of different forms of student models in order to adapt the hypermedia pages content and links showed to the students (adaptive presentation and adaptive navigation support) are studied by Adaptive Hypermedia Systems. On the one hand, "Intelligent Tutoring Systems (ITSs) are computer-aided instructional systems with models of instructional content that specify *what* to teach, and teaching strategies that specify *how* to teach" [17]; and on the other hand, Adaptive Hypermedia Systems

adapt the content of a hypertext page to the user's goals, knowledge, preferences and other user's information for each individual user interacting with the system [1].

The first Web-based AIES appeared in 1995 and they arose from Intelligent Tutoring Systems. Some examples of these systems are ELMART [3], InterBook [2], CALAT [12] or WITS [13]. Subsequently new intelligent and adaptive techniques have been introduced in these systems, as the adaptability of the pages content adjusting them to the user needs in each moment of the interaction (Adaptation Hipermedia) [6][13] and collaborative adaptation in web [11], among others.

The RLATES system (*Reinforcement Learning in Adaptive and intelligent Educational System*) is a Spanish DataBase Educational System in Web that uses the Reinforcement Learning model in order to provide the student an "optimal" curriculum sequence according to his learning characteristics in each moment of the interaction. At the same time, the system uses adaptive techniques in order to adjust the presentation of the content to the student and to help him to navigate through the application interface. The RLATES system forms part of the PANDORA project¹ [5], whose main goal is to define methods and techniques for database development implemented in a CASE tool, useful for students and practitioners.

The paper is organized as follows: first, the proposed architecture of the AIES is briefly defined in section 2. In section 3, the navigation through the RLATES interface is described and finally, the main conclusions and further research of this work are given.

2 System Architecture

A typical structure of an ITS, and hence, of an AIES, is composed of four well differentiated modules [4] (see Figure 1).

The *student module* contains all important information about the student in the learning process: goals, student background knowledge, personal characteristics, historical behavior, etc. It is necessary, for the effectiveness of RLATES, to construct a good *student model* and to classify learners according to their critical characteristics in learning the system knowledge. This classification is very important for the pedagogical model, as we will see in following paragraphs.

The *domain module* contains all characteristics of the knowledge to teach. The traditional knowledge structure (hierarchical structure) could be an advantage in the pedagogical strategy in AIES, where each topic has been divided into sub-topics, and these in others sub-topics, and so on. This structure implies "is-parent" relationships between the topics. An other relationship could be applied to the knowledge, the "pre-requisite" relationship, where the *A* topic is pre-requisite of the *B* topic if the tutor can not show the *B* topic when the *A* topic has not yet been shown. At the same time, each node of the tree contains tasks (sets of definitions, examples, problems, exercises, etc.) in several formats (image, text, video, etc.).

¹ CASE Platform for Database development and Learning via Internet. Spanish CICYT project (TIC99-0215).

The *pedagogical module* decides *what, how* and *when* to teach the domain module contents, taking the better pedagogical decisions according to the user needs. The definition of this problem as a Reinforcement Learning problem is described in previous papers [9][10], where the system receives the current student's knowledge by evaluations (*tests*) and choose the next action to be executed according to the current *action policy, B*. The *action* corresponds with showing set of leaves of the knowledge tree (*definition, exercise, problem, etc.*). When the action is executed, a state transitions it is supposed to occur, and the environment provide a *reinforcement signal* (positive or negative) to the system. The reward signal supplies a maximum value upon arriving to the goal of the tutor. This signal is used to update the system's action policy. The system behavior, *B*, should choose the actions that tend to maximize the long-run sum of values of the reinforcement signal, choosing in this way the optimal tutoring strategy (what, when, and how to teach; the best sequence of contents and how to teach them) to coach the current learner. The value-action function $Q(s,a)$

estimates the usefulness of executing one action (showing leaves of the knowledge tree to a student) when the system is in certain knowledge state. Therefore, the goal of the learning process is to find the policy that maximizes this function.

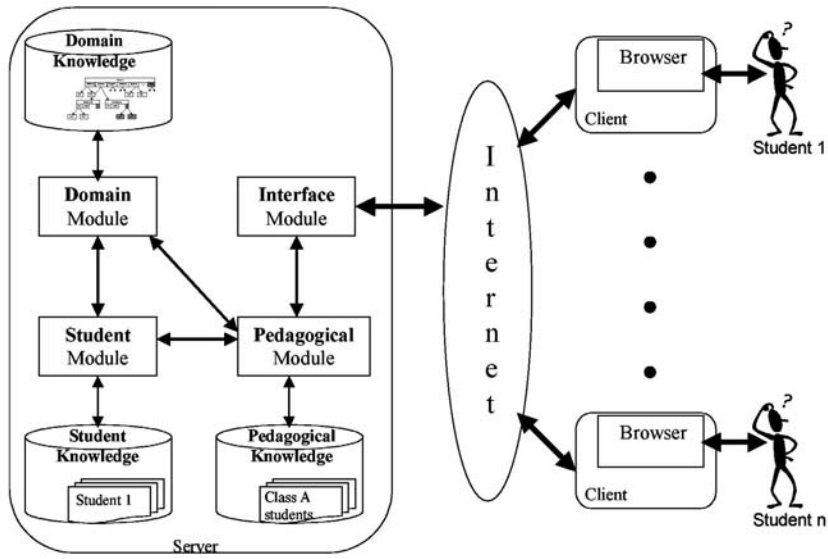


Fig. 1. RLATES Architecture

Updating the action policy can be performed by different algorithms. The algorithm implemented in RLATES is the Q-learning algorithm, where its value-action function is defined in the equation 1.

$$Q(s,a)=(1-\alpha) Q(s,a)+ \alpha\{r+ \gamma \max_a Q(s',a')\} \tag{1}$$

This equation requires the definition of the possible states, s , the actions that the agent can perform in the environment, a , and the rewards that it receives at any moment for the states it arrives to after applying each action, r . The γ parameter controls the relative importance of future actions rewards with respect to new ones, and α parameter is the learning rate, that indicates how quickly the system learns. In [9] how the Q-learning algorithm is adapted to the Adaptive and Intelligent Educational System domain is explained.

Finally, the *interface module* facilitates the communication between the AIES and the student. This module applies intelligent and adaptive techniques in order to adapt the content and the navigation to the students, leaning on the pedagogical module, that decides which is the next task to be showed to the student and in which format the knowledge is going to be taught.

This article focuses on the description of the interface module, since previous articles [10] have shown that the application of the reinforcement learning model in the pedagogical module permits that the system learns to adapt its pedagogical strategies according to the student needs in each moment, only based on information of the interaction with other students with similar learning characteristics. Besides, the theoretical model have been successfully applied to the DataBase Design domain [4], showing subsequently that the AIES can learn an optimal policy to teach students interacting with reasonably few students [5].

The intelligent and adaptive techniques used at the interface model of the RLATES system are described in next section, where we have distinguished between the adaptive navigation support and the adaptive presentation of the system contents.

3 Hypermedia Adaptation in RLATES

The Adaptive Hypermedia Systems (AHS) provide a not lineal progression through the material of the AIES, where a *hyperdocument* consists of a set of pages connected by links. In previous hypermedia systems this property implied a problem: the student used to loose himself in the hyperspace, with no idea of what to study next and how to return to the previous page learned [15]. The new hypermedia systems resolve this problem adding an intelligent component to be adapted to the users and to help them. On the one hand, these systems could adapt the navigation of the students (*Adaptive Navigation Support*) helping them to find their paths in hyperspace by adapting link presentation and functionality of the pages according to the goals, knowledge and learning characteristics of an individual user.

On the other hand, the system could adapt the content of a hypermedia page (*Adaptive Presentation*) according to the goals, knowledge and other characteristics of the student.

In this section how the RLATES system adapts to user is described, analyzing the two major technologies in a adaptive hypermedia: a adaptive navigation support and adaptive presentation.

3.1 Adaptive Navigation Support

“The goal of the *Adaptive Navigation Support (ANS)* technology is to support the student in hyperspace orientation and navigation by changing the appearance of visible links” [1]. This technique shares the same goal that the technique of *Curriculum Sequencing* in Intelligent Tutoring Systems: to provide the student a sequence of knowledge items (knowledge units) and tasks (examples, questions, problems, etc.) to learn adapting this sequence to the learning characteristics of each individual student. That is to say, the system helps students to find an “optimal path” through the AIES learning material. But, at the same time, the ANS technique adds some advantages of the hypermedia context: it can guide the students directly (like curriculum sequencing), but also indirectly (by the presentation of the links in the page).

Next, the ways the RLATES system adapts the links are described, illustrated by several figures:

On the one hand, the system can **guide** the student **indirectly** through the AIES content by changing the appearance of the links in the page, changing the color or the order of the links in the page. The RLATES system includes a table of the AIES contents (in a tree format) always visible in the left frame of the application that provides the indirect guide to the users. This tree of contents has one link for each one unit of knowledge in the domain module of the AIES. If the student click one of them, the system will show at the right frame tasks of the item of knowledge. In Figure 2 we can observe how the system shows some tasks of the “Entidades” (*Entities*) item, chosen by the student.

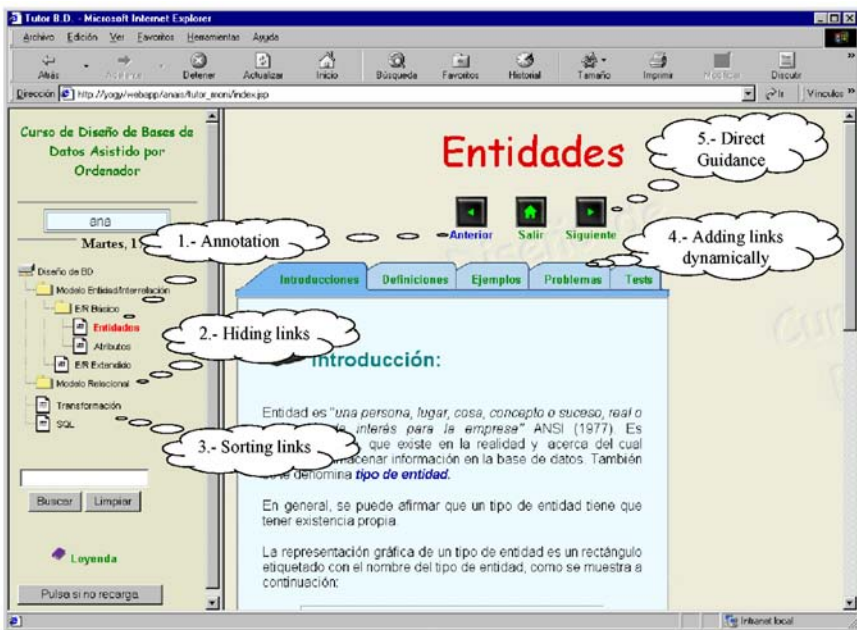


Fig. 2. Techniques of link adaptation in RLATES system

The RLATES system uses the following indirect techniques of guidance that we can see in an example in Figure 2:

1. **Annotation:** Is the most popular form of ANS. With this technique, the system augments the links with some comments about the pages behind the annotated links. The RLATES system changes the color of the links of the tree of contents according to the student historical behavior (see the Figure 3); the name of the link is bolded and the color is red if the topic is the actual topic (we can show tasks of this topic in the right frame); the color of the topic is black if the topic have not been shown yet; the color is green if the student has passed the last test about this topic; and finally, the color is orange if the student has not passed the last test about this topic.
2. **Hiding links.** This technique hides the topics that at a certain moment are not relevant for the student. For instance, when the student is studying the “Entidades” (*Entities*) topic of the “Modelo Entidad/Interrelación” (*Entity/Relationship Model*) topic, the units of knowledge of the “Modelo Relacional” (*Relational Model*) topic are not relevant for the student at this moment.
3. **Sorting links.** This technique tries to physically order in the page the links according to the model of the user (normally based on the student knowledge or in the student behavior). The RLATES system order the topics in the knowledge tree according to the “pre-requisites” and “is-parent” relationships of the units of knowledge described in the AIES domain module.
4. **Adding/Removing links dynamically.** This technique adds or removes dynamically links to the page according to the learning needs of the student. It tries to prevent the user from following links that are not relevant for him/her at a certain moment. RLATES adds or removes links of the right frame of the application according to the tasks stored at the domain model about this particular topic. For instance, if the “Entidades” (*Entities*) topic has not definition tasks, the link “Definitions” (*Definitions*) will be removed from the page.

On the other hand, the system can guide the student through the material by **Direct Guidance** when the student chose the “next” button to continue the learning process (see the Figure 2). When this situation happens, the system drives the student to the “best” page in the hyper space according to their learning characteristics (current knowledge, goal, etc.). The task of choose the next page to be shown (the next action to execute) concerns to the pedagogical module of the AIES. This module uses the reinforcement learning model in order to adopt a decision. A great variety of exploration/exploitation strategies could be used in reinforcement learning problems in order to choose the next action to execute [184]. For example, the *e-greedy* strategy tries to select the action with greater value of the Q function when the system is at the s state ($Q(s,a)$) with a probability of $(1-e)$. That is to say, the e parameter shows how greedy is the system when it chooses a new action to execute: when $e=1$, the system randomly selects the next action to be executed and when $e=0$ the system chooses the action with a greater value of Q . Another very common exploration/exploitation strategy is the Boltzmann exploration policy, that estimates de probability of choosing the

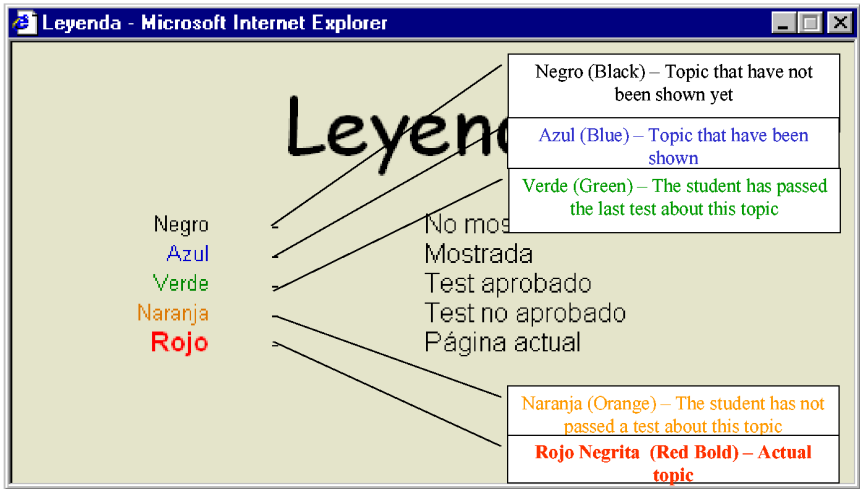


Fig. 3. Link annotation in RLATES system

action a according to the next function, where τ is a positive parameter called the *temperature* and $Q_t(x)$ is the value of the Q function at the t time when the system is in a certain state s and the action x is executed:

$$P(a) = \frac{e^{-\frac{Q_t(a)}{\tau}}}{\sum_{b=1}^n e^{-\frac{Q_t(b)}{\tau}}} \quad (2)$$

If the temperature is high, all the probability of the actions have almost similar values and if the temperature is low, it causes a great difference in selection probability for actions.

In distance educational system (like our system), to maintain the attention of the user is absolutely necessary, and to give the sensation that the student has the control of the interaction is very important. This is why RLATES has implemented the *direct guidance* by proposing the student more than one possibility for the next page (at almost 10 possibilities). Then, the student is in charge of choosing the next page to visit, bearing in mind the suggestion of RLATES that appears at the page in percentage format. In Figure 4 we can see the page that appears when the student clicks the “next” button in our application.

The RLATES system implements the Boltzmann exploration/exploitation policy, because it provides the probability of choose an action to be executed, not as the *e-greedy* exploration/exploitation policy. In [7], some experiments analyzing the differences between the Boltzmann and the *e-greedy* exploration/exploitation policies applied to Adaptive and Intelligent Educational Systems have been done.

3.2 Adaptive Presentation

“The goal of the *Adaptive Presentation* technology is to adapt the content of a hypermedia page to the user’s goals, knowledge and other information stored in the user model” [1]. RLATES implements this adaptation too by using the reinforcement learning model. The tasks of the topics are stored in the domain model, keeping information about the formats of their pages. When the student clicks the “next” button, the system will choose a task in a format that adjusts to the student learning needs.

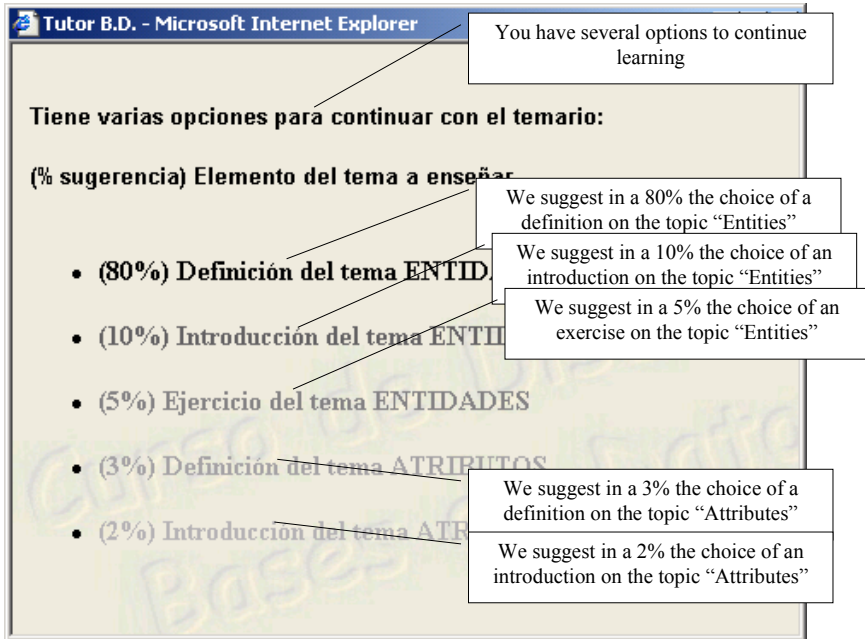


Fig. 4. Hypermedia page where the student can choose the next task of RLATES to be executed.

4 Concluding Remarks

This paper describes the architecture of the RLATES system, focusing on the interface module. How the RLATES system uses intelligent and adaptive techniques in order to adjust the teaching according to the current student in each moment of the interaction has been explained. The system adaptation has been divided into two problems: the *Adaptive Navigation Support*, where RLATES guides the user both, by *direct guidance* (by clicking the “next” button”) and by *indirect guidance*, changing the appearance of the links in the page (by a notation, sorting links, hiding links, adding and removing links, etc.) and the *Adaptive Presentation*, where the system is leaned on the pedagogical model of the system.

This system is being implemented as part of a PhD research. Some experiments with simulated students have been done in order to evaluate the convergence of the Q-learning algorithm in Adaptive and Intelligent Tutoring Systems [7][9]. Now we are doing some experiments in order to evaluate the system with real students.

Acknowledgements. We thank Ana Isabel Sebastián and Mónica Merino for helpings as at the implementation of the web pages of the RLATES system.

References

1. Brusilovsky, P. *Adaptive and Intelligent Technologies for Web-based Education*. Kunstliche Intelligenz. Special Issue on Intelligent Tutoring Systems and Teleteaching. Vol. 4, 1999.
2. Brusilovsky, P., Eklund, J., and Schwarz, E: *Web-Based Education of all: A tool for developing adaptive courseware*. Computer Networks and ISDN Systems. 30, 1–7, pp. 291–300, 1998.
3. Brusilovsky, Peter, Schwarz, Elmar and Weber, Gerhard. *ELM-ART: An Intelligent tutoring system on World Wide Web*. Intelligent Tutoring Systems. Springer Verlag. Ed. Claude Frasson and Gilles Gauthier and Alan Lesgold, pp. 261–269 (1086), 1996.
4. Burns, Hugh & Capps, Charles. *Foundations of Intelligent Tutoring Systems: An Introduction*. Foundations of Intelligent Tutoring Systems. Hillsdale, N.J: Lawrence Erlbaum Associates. 1–19, 1988.
5. Castro, E., Cuadra, D., Martínez, P., and Iglesias, A. Integrating Intelligent Methodological and Tutoring assistance in a CASE platform: the PANDORA experience. In Proceedings of the Informing Science & IT Education Conference. Cork, Ireland, 2002.
6. De Bra, P. and Calvi, L. *An open Adaptive Hypermedia Architecture*. The New Review of Hypermedia and Multimedia, pp. 115–139 (4), 1998.
7. Iglesias, A., Martínez, P., Aler, R. And Fernández, F. *Analising the Advantages of Using Exploration and Exploitation Strategies in an Adaptive and Intelligent Educational System*. 2nd International Conference on Multimedia and Information and Communication Technologies in Education (m-ICTE03). Badajoz (Spain). 2003
8. Iglesias, A., Martínez, P., Cuadra, D., Castro, E. and Fernández, F. *Learning to Teach Database Design by Trial and Error*. 4th International Conference on Enterprise Information Systems. Ciudad Real (España). Pp. 500–505, 2002.
9. Iglesias, Ana, Martínez, Paloma and Fernández, Fernando. *An experience applying Reinforcement Learning in a Web-based Adaptive and Intelligent Educational System*. In Informatics in Education International Journal. To appear.
10. Iglesias, Ana, Martínez, Paloma and Fernández, Fernando. *Applying Reinforcement Learning in Intelligent Tutoring Systems*. 4th International Conference on New Educational Environments. Lugano (Switzerland). Pp. 11–14, 2002.
11. Leigh, J., Johnson, A.E. and DeFanti, T.A. Issues in the Design of a Flexible Distributed Architecture for Supporting Persistence and Interoperability in Collaborative Virtual Environments. Proceedings of the 1997 ACM/IEEE conference on Supercomputing (CDROM). Conference on High Performance Networking and Computing, San Jose, CA, pp. 1–14, 1997.

12. Nakabayashi, K., Maruyama, M., Koike, Y., Touhei H. and Fukuhara, Y. *Architecture of an Intelligent Tutoring System on the WWW*. Proceedings of the 8th World Conference of the AIED Society, pp. 18–22, 1997.
13. Okazaki, Y., Watanabe, K., and Kondo, H.: *An Implementation of an Intelligent Tutoring System (ITS) on the World Wide Web (WWW)*. Educational Technology Research 19. 1, pp. 35–44, 1996.
14. Sison, R. *Framework for Intelligent Internet-Based Learning Support*. Proceedings of the Ninth International Conference on Computers in Education. pp. 926–931, 2001
15. Stern, Mia Keryn. *Using Adaptive Hypermedia and Machine Learning to Create Intelligent Web-based Courses*. PhD. Directed by: Professor Beverly Park Woolf. University of Massachusetts, Amherst. 2001.
16. Trhun, Sebastian B.. *The role of exploration in learning control*. In David A. White and Donald A. Sofge (Ed.), *Handbook of Intelligent Control: Neural, Fuzzy and Adaptive Approaches*. Van Nostrand Reinhold, New York, 1992.
17. Wenger, E . *Artificial Intelligence and Tutoring Systems*. Los Altos, CA: Morgan Kaufmann, 1987.