

# Training the Teacher: A New Approach for Authoring ITSs for Teaching Decision-Making

Felisa Vázquez-Abad <sup>[1]</sup> (member *GERAD*), Arnoldo Rodríguez & Esma Aïmeur  
Department of Computer Science and Operational Research, University of Montreal  
P.O. Box. 6128, Downtown Station, Montreal, QC, H3C 3J7, CANADA  
*[vazquez, rodrigua, aimeur]@iro.umontreal.ca*

**Abstract:** In this article, we switch our focus from the student's to the teacher's problems in relation to authoring an Intelligent Tutoring System (ITS). In line with this shift, we will try to answer the question: how can a teacher be guided to teach decision-making using an ITS? Accordingly, in the examination which follows the teacher will be viewed as a trainee who needs to be instructed not only in the different pedagogical strategies involved in teaching decision-making skills, but in the usage of an ITS.

The implementation of our approach includes a Web-based ITS authoring tool to create learning sessions. In addition, the system uses an adaptive tutor to train instructors in teaching decision-making skills, while they are in the midst of authoring learning sessions.

Our solution implies a Web-based interface agent that can learn from the interaction between teachers and the system.

## Introduction

### The Problem

In this beginning of the 21<sup>st</sup> century, more and more universities are promoting the use of Computer-Based Teaching (CBT) and Web-Based Technology (WBT), and teachers are strongly encouraged and even sometimes required to modify their teaching methodologies to adapt to this new trend. Consequently, teachers must invest a lot of time and effort in preparing their courses using computer-based materials (Johnson, 01).

Although Intelligent Tutoring Systems (ITS) facilitate the development of supporting material for teaching, they are complex systems that may manipulate large amounts of information interconnected in highly complex ways, and most authoring tools used to create ITSs remain difficult to learn and use for most people (Frasson, 98).

Our research focuses on the teacher's problems in authoring an ITS that integrates the theory and practice of the decision-making process in higher education and training. Our pedagogical model complements traditional teaching with a series of Web-based activities that provide virtual practice and integration of concepts (Vázquez-Abad, 01). We consider the teacher to be a trainee requiring adaptive support to interact with an ITS. Our proposal seeks to create an intelligent adaptive guide and authoring tool that will enable any qualified teacher to create his or her own course-work with minimal programming effort. We use a novel three-dimensional model to represent the complexity of the help required by teachers. Our study analyses how a Web-based interface agent can improve the interaction between teachers and an ITS, and how that agent can learn from their interaction. A tutor that learns in this way is a novelty in the area of tele-learning and ITS.

### Previous Works

Past research has not focused on on-line intelligent and adaptive/hypermedia that helps instructors to teach using ITSs, and, in fact, few authors have mentioned the lack of research on the subject of instructor modelling and the advantages of using information about teachers to improve ITSs (Kinshuk 01). For Virvou and Moundridou (02), ITSs require intelligent help to assist teaching decision-making skills while users are authoring, because regular help is not enough. Most authoring tools do not provide interactive support to exploit their functionalities. There is some research that aims to guide workers while they are working, such as ADAPTS, an adaptive hypermedia ITS to guide trainers in maintaining complex equipment (Brusilovsky 99), but most authoring tools for creating ITSs do not provide interactive support to exploit their functionalities. WEAR and CACTUS are two

---

[1] Also Principal Investigator, DEEE, The University of Melbourne, Australia.

of very few systems that specifically address the issue of how to guide teachers using an authoring tool to create an intelligent tutoring system. CACTUS has an authoring tool to create ITSs as well as an agent that monitors users to guide them in using the tool (García 00). Using a teacher model, WEAR helps authors with teaching (strategy), and allows teachers to share exercises among themselves (Virvou 02).

## The Pedagogical Model

The pedagogical model behind this research comes from the Web-site BestNet (Vázquez-Abad, 01), which was conceived to teach how to improve the students' skills on network design. That model breaks down the study of the decision-making processes into a series of activities that incorporate game elements in order to help the student integrate theory and practice in a virtual environment of challenges and competition, oriented towards the training of engineers in the subject of Network Design.

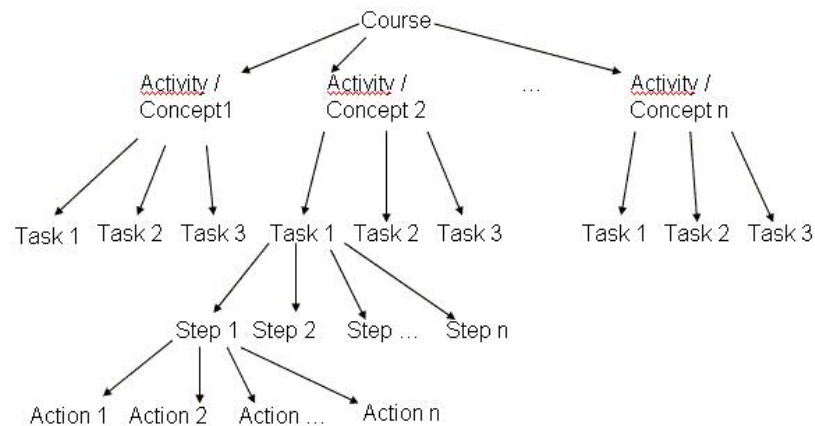
In this work we extend the pedagogical model of BestNet to support the teaching of complex decision-making. The extension that we propose incorporates a pedagogical model for the course material to encompass courses in diverse areas such as medicine, business, engineering, and commerce. In accordance with this model, we can consider a decision-making process as the transition (T) of a state ( $S_n$ ) to another ( $S_{n+1}$ ) provoked by an action (A):  $S_{n+1}=T(S_n, A)$ .

The syllabus of courses related to complex decision-making is usually composed of a series of subjects, each one dedicated to the description of how different sub-systems behave as a function of the actions. For instance, textbooks on network design contain a chapter on network reliability, which are related only to the network connectivity and links characteristics. Another chapter may deal with capacity allocation, which is a function of the demand. The decision-making problem related to each subsystem (such as network reliability) is stated as the problem of finding the best action for each state in order to optimize the corresponding performance criteria. Different subsystems focus on different performance criteria and may emphasize different transition functions. Mathematically, each performance  $F_k(S_n, A_n)$  is a function of the transformation of the state that will focus on a subset of possible actions. In the example of Network Design, the reliability (probability that the network is functional) is a performance criterion associated with a mathematical function of the chosen links to connect the nodes, independent of the traffic demand patterns. When allocating capacity, the actions represent the capacity of each of the (given) links and the performance will be expressed in terms of overall utilization (the proportion of time that resources are utilized in full). Therefore, each possible action has N components:  $A_n=(a_n(1), a_n(2), \dots, a_n(N))$ , and each of the performance criteria is expressed as a function of the states given a subset of all the possible decisions:  $F_k(S_n, a_n(k))$ .

The model described above considers a state as a complex system of elements and relationships, thus when an element or connection changes as a product of an action or an operation, all the other components and relations are modified, and the current state is turned into a new state.

Traditional textbooks dealing with the subject of reliability in a course on network design do not include elements such as topographical or political obstacles to the deployment of certain links in a particular urban area, for instance. Our ITS is intended to suggest to the teacher the elements outside traditional coursework that he or she might want to incorporate so that the students can learn how to manage and solve practical problems.

According to that pedagogical model, our ITS includes an authoring tool and an adaptive guide that support the construction of a sequence of *activities*, which allow students to learn how to apply concepts previously learned in class. For any course, we have a curriculum or plan with the main concepts used in the decision-making process that the students will learn in terms of actions and performances. Each concept will map to an activity, and each activity is broken down into three specific tasks or teaching strategies (of increasing degrees of difficulty) for the integration of knowledge of a subject. Each task is also broken into a sequence of steps and actions to be done. The general structure of our model is detailed in the next figure (Fig. 1).



**Figure 1:** General structure of our authoring model.

In the case of network design, our approach proposes four main concepts to be applied in the decision-making process of designing a network: *Reliability* (defined as the probability that all branches are connected by operational links), the *Concentrator Location*, the *Capacity Allocation*, and the *Design of a local area network*. The three first activities integrate knowledge from different sources and the last activity allows the integration of the previous ones. For example, it is common to teach mathematical formulations of the reliability independent of the demand of the network, and the problem of capacity allocation is formulated assuming no link failures. In real life, however, traffic patterns may affect the link failure probabilities and link failures may affect the utilization of the network.

For any course, the traditional curriculum can be divided into the main concepts that the students will learn in terms of actions and performances. Each concept will map to an activity, and each activity will allow three specific experiences or tasks to be done. The different tasks will be accomplished by following a sequence of steps and actions.

Each activity (except the last one) entails the following pedagogical strategies:

- *Analysis* of cases: in the case example, this activity prompts the student to compare various networks on the basis of reliability, cost, traffic balance, failure scenarios, etc.
- *Modification* of cases: following the same type of example, it is possible to select actions (delete or add a link) that cause a given change in the reliability of a case.
- *Competition*: an example of this is when a teacher asks students to search for a network topology that meets a specified reliability level. The "winner" is the person who achieves this goal with minimal cost.

The final activity integrates the concepts studied in previous activities. It is conceived as a virtual competition where an underlying simulation of IF-THEN cases is presented to the students to prompt them to integrate all the concepts according to the teacher's pedagogical objectives.

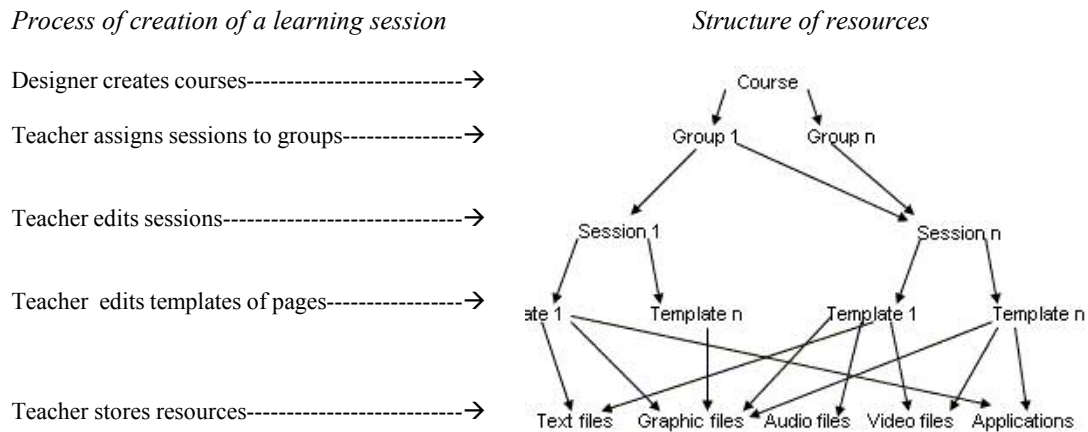
## Methodology

### A Basic Structure

Our approach entails having a curriculum created by a designer and the course includes resources, pedagogical strategies, specific rules, guidance for teachers, a case base and a learning session base. A *learning session* is an instance of specific sequences of activities, tasks, steps and actions that uses certain resources to teach one or more groups of students. Teachers can create one or more sessions for each of their groups or they could link sessions of different authors to their groups. The next figure (Fig. 2.) shows the organization of resources into templates for sessions of a course.

Basically, a learning session is a sequence of templates of different pages. The system has some standard templates for the different steps of the tasks, and, when a teacher finishes the actions to be done on a page, a new record is stored in the database, except if there is a similar one that can be shared for two sessions. Some instances of templates are associated with specific pages or steps of a task in a session. Each template arranges one or more

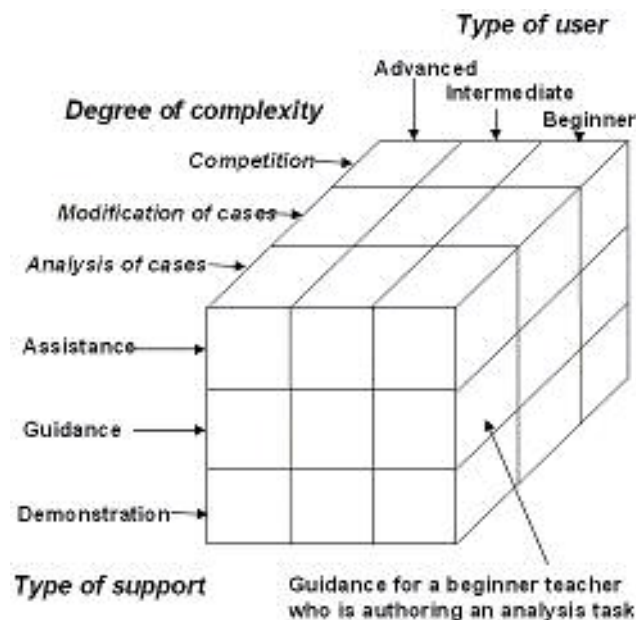
fields with resources (texts, text boxes, graphics, labels, links, and buttons) that are static or editable, passive or active. Different instances of a type of template can show different pictures depending on the file selected by each author, but they can share the same text from a file. A program on the server side generates the code of the template that integrates the required resources for an instance, and a record of the instance is stored with a specific automatic identification number.



**Figure 2:** Process of creation of a learning session and general structure of resources in learning sessions.

### The Dimensions of the Model

We propose a three-dimensional structure to adapt the support provided to the teacher according to the characteristics of the instructor. One dimension represents (1) the *type of user*, in our case the profile of each teacher; the second is related to (2) the *degree of complexity* of the design task (analysis, transformation of cases, competition); and the final dimension is (3) the type of content and media (*type of support*) that will be provided to support the teacher according to his user profile and the complexity of the task. (Fig. 3.) shows a three-dimensional model of the three types of factors mentioned above.



**Figure 3:** A 3-D model of tasks and guidance.

We can classify teachers in three basic categories or levels according to their attributes and experience using the system: *beginner*, *intermediate* and *advanced* users. The support for users at these three levels could include:

*demonstration, guidance, and assistance* in free explorations. An interface agent can apply those three types of support in an explanation of the design of the learning sessions, or to demonstrate the system functions. Demonstrations explain specific tasks. Guidance implies step-by-step tutorials, and it is more suitable for beginner and intermediate users. The assistance with full explanations for free explorations is for advanced users.

The system adapts to the teacher as he progresses from beginner to advanced user by using a user model (overlay model) that stores the teacher's levels of experience for the different tasks. The user can work with the authoring tool, access help and information, and the system will record tracks of that interaction. The level of experience using the tool is measured by the number of times the teacher worked with a task, the frequency of usage for each task, and the type and frequency of access to each topic of help related to this task. According to the teacher's history of interaction, the system will provide him less support as his experience increases.

A specific adaptation arises when the system makes decisions about which help strategy (demonstration, guidance, and assistance) would suit better for a teacher authoring a certain task. If the user has been working frequently on a task, and he has been helped previously with demonstrations about this task, during a next access the system only will have to provide minimal guidance for the same task, because the degree of complexity of this task becomes relatively lower. According to the stored information, a teacher with experience in the analysis phase of creating learning sessions will receive less basic guidance, but he or she will receive more intensive help to create a competition the first time he tries to perform this task. The system will provide different support for each task.

Another important aspect to take into account is forgetfulness. A teacher can forget partial or completely some concepts or procedures he had assimilated previously. Since we accept a teacher can forget some topics, our system is intended to deal with teacher's forgetfulness by helping him to remember aspects related to teaching strategies, knowledge of the domain and how to use the functions of the authoring tool. The system relies in the teacher's experience, so, as more experienced a user is, refreshing and remembering information could be easier for him. For instance, after a month absence, a teacher restarts teaching a task, and then the system tries to adjust the support according to the teacher's forgetfulness for the selected task. The system checks the stored information about the teacher's experience for the chosen task and starts a dialogue with him. If the teacher is not used to author this task frequently, the system can suggest refreshing some topics. The teacher could also ask explicitly to review a basic level of help about the task he is trying to perform.

## **The Architecture**

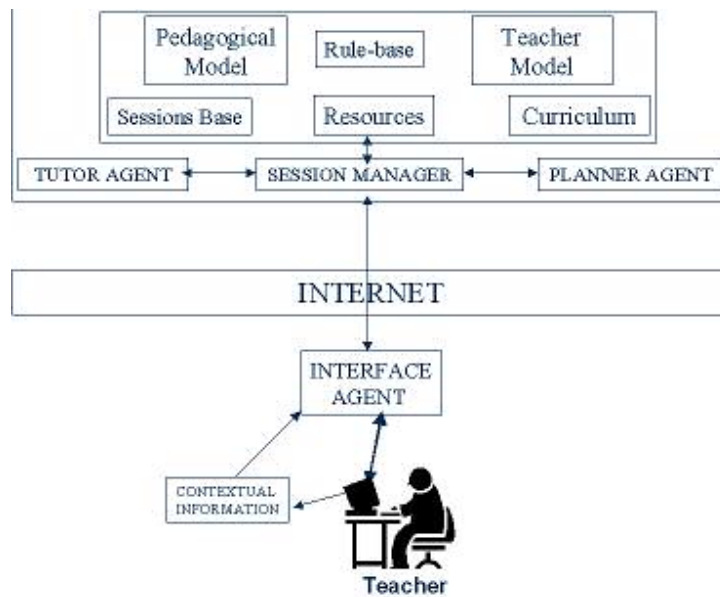
The model we have discussed previously has a client-server architecture, where the *server side* hosts a Web-based authoring tool with an adaptive ITS to help authors, and an intelligent adaptive interface guides teachers on the *client side*. Both sides communicate by Internet using a hypermedia structure, and the functionality on both sides is supported by agents that interact with other applications and models stored on the server side. On the server side, the system includes three agents (Session Manager, Planner Agent, Tutor Agent) and the models required for the ITS and the authoring tool: Curriculum, Teacher Model, Pedagogical Model, Rule-base, Learning Sessions Base and Resources. On the client side, the Interface Agent interacts with the user, monitoring his environment and sending information to the server. The main functionality of the system (authoring tool and ITS) is performed by the four agents (Shiri-Ahmadabadi, 99). The server hosts three agents with the following characteristics:

- 1) *The Session Manager* is an information agent that manages the flow of information into the society of agents and between the agents and other applications.
- 2) *The Planner Agent* adapts the sequence of activities and tasks according to each teacher's profile.
- 3) *The Tutor Agent* is in charge of selecting strategies, content and media to help or guide the author. This agent organizes the sequence, frequency and quantity of information sent to the client side.

On the client side there is an *Interface Agent* that interacts with the users, monitors the context and exchanges information with other agents and components of the system. The Interface Agent is in charge of delivering functionality and help adapted to the user.

The agents communicate among themselves and the models hosted in the server by means of the session manager. The other components have relations of mutual dependency. For instance, information about the teacher model

includes the strategies and resources most used by the teacher. The next figure shows the general architecture of the systems (Fig. 4).



**Figure 4:** A general architecture of the system includes agents and other components.

*Resources.* Since each page is generated dynamically according to the teacher profile, the level of complexity of the methodology and the functionality of the authoring tool, it is not possible to keep static pages. An alternative is to have a database of reusable resources: graphics, text files, sounds, video clips, specific applications, simulators. Each resource is associated with different contents and concepts to be taught, and every type of help maps to some resources. This method allows resources to be reused to build the pages.

*Curriculum.* A course includes the teaching content and its organization or sequence. On the server side, it would be possible to store different curricula about subjects such as business, engineering or medicine. The structure of the curriculum allows mapping contents, concepts and the activities and tasks of each course to specific steps that the teacher must accomplish. According to the characteristics of the teacher and his history of interaction with the systems, each step will be associated with a template that arranges and reuses the resources (text, images, links) and widgets required for that step. In our system, we have two Curricula: 1. the Curriculum to be taught to the students, and 2. the Curriculum to be offered to teachers.

*A Session-base* stores specific learning sessions. A learning session is a sequence of steps that belongs to a set of tasks related to the same learning session. Since a step is stored as a row of values in a table of a database, a session is the outcome of the authoring decision-making process to assign values to each step. On demand, the planner agent can propose a different sequence of steps from the session-base.

*A Rule-base* can guide the decision-making processes of the ITS and the authoring tool, according to different information. It is better to have a Rule-base that is possible to edit, so the system will be more flexible instead of having embedded specific rules in different sections of the code. The rules allow the system “to reason” what to do since a user is logged in until he or she receives Web pages.

*The Teacher model* profiles the characteristics of the teacher in categories related to identification, knowledge level, learning style, teaching style, preferences and history of previous activities. It is also possible to classify the characteristics according to: cognitive factors related to the knowledge; interests and skills of the users; as well as affective aspects that imply preferred media to received messages, availability to share and communicate knowledge or layout details such as the size of the letter to show running text on the screen. In the case of the knowledge level we have three sub-categories: Low, medium and high ability. To deal with uncertainty generated when a user does not provide all the explicit information required to create his/her profile, we use rules to get data from categories (stereotypes) of user’s profiles to supply missing information in a specific profile.

*The Pedagogical model* is the knowledge about how to teach. In this research we are implementing three strategies to teach students: comparison of examples of a case, modification of an example of a case, and

competition to find the best solution to a problem. In general, the objective is to apply instructional strategies, which allows teachers to design competitions. However, since we see each teacher as a trainee, our pedagogical model includes the strategies and techniques to guide teachers when they are authoring learning sessions. Those strategies include demonstrations, guidance (guided learning by doing) and assistance, using text based explanations, graphics, and multimedia resources.

## **An Example**

### **Decision-Making on Network Design**

To provide a more detailed description of the system, we have selected network design as a subject to explain how a teacher has to interact with the system to create a learning session. First, we imagine the case of a novice user who has to start using the site to create a learning session on network design. That user must begin with the analysis task of the activity “Reliability” to offer a series of network configurations the students must compare. Our beginner teacher has entered the environment, filled in the profile, but he or she does not know what to do first. The system has stored the information that he or she has never used the site, he or she prefers to receive text based support, and learns slowly. When he or she had to choose between creating a new session or reusing one from the session-base, the Interface Agent advised him or her to use a session from the session-base because following an example would be better for him or her. The Planner Agent uses the information from the database to trigger some rules and to create the part of the page related to the menu and the author tool. The agent checks the Teacher Model, the Curriculum to be taught and the Session-Base to know to which step of which learning session the user will go (in this case it is the first step of task 1). After that, the agent defines which resources are required and sends the information to the Session Manager. To generate the support to be provided to the teacher, the Tutor Agent deals with the information from the Teacher Model to apply rules for filtering data of the Pedagogical Model, the Curriculum to train instructors, and the Resources. This agent delivers its part of the support, then the Session Manager integrates the part of the Planner and the part of the Tutor to send the client-side a page with a navigation menu on the left side, a form to author the task, and, on the right side, two initial text boxes with explanations of the steps to do the task and how to perform them.

The user receives a page that is a template with a set of actions to generate an assignment in which students are required to write an essay comparing different configurations. The instructor can edit some parts of the templates by typing new text or selecting different criteria students will apply to compare the two cases provided by the retrieved session. When the author is working, the Interface Agent detects in which section the user is to adapt the content of the help. This agent also monitors if changes entail something new to learn about how to teach, and offers additional help options to learn how to improve the support system for that user.

## **Conclusion**

This research aims to improve the communication between teachers and ITSs. We are analysing the *communication model* and the *teacher model*, aspects that have not been studied to the same extent as the student model or the pedagogical model of intelligent tutoring systems.

Our system is intended to learn from its interactions with users to improve the ability to help teachers to create learning sessions by using an authoring tool. Since we regard a teacher as a trainee, we deal with both learning and *teaching styles* for the same user, as opposed to most of the current research that focuses only on students. In addition, saving learning sessions authored by teachers is a way to acquire their expert knowledge, and freeing the access to a Session-base is a good way to share knowledge among instructors. Learning in our system also implies the ability to recommend sessions and to generate new ones.

Using an ITS, the instructor may have a cognitive overload because he or she has to teach and, at the same time, learn how to teach using the ITS. Nowadays, that type of overload is a problem for many workers, so our research attempts to contribute to solving the difficulties of on-the-job-training.

Our research attempts to break the bottleneck that blocks the access of professors to the new information technology applied to education. This research could help to include more teachers in the current socio-economic and pedagogical trend of increased use of the Internet for tele-learning.

## References

- (Brusilovsky, 99) Brusilovsky, P. and Cooper, D. (1999). *ADAPTS: Adaptive hypermedia for a Web-based performance support system*. <http://www.wis.win.tue.nl/asum99/brusilovsky/brusilovsky.html>
- (Frasson, 98) Frasson, C. (1998). Using cognitive Agents for Building Pedagogical Strategies in a Multistrategic Intelligent Tutoring System. In *Deuxième journée Acteurs, Agents et Apprentissage*, Bayonne, September. <http://www.iro.umontreal.ca/~frasson/FrassonPub/Bayonne-Agents cognitifs-sept98.doc>
- (García, 00) García, F. (2000). CACTUS: automated tutorial course generation for software applications. 113-120 International Conference on Intelligent User Interfaces. *Proceedings of the 5th international conference on Intelligent user interfaces*. United States. <http://doi.acm.org/10.1145/325737.325798>
- (Johnson, 01) Johnson, W. L. (2001). *Research in animated pedagogical agents: progress and prospects for training*. Based on a report prepared for Naval Air Systems Command, Orlando FL. United States.
- (Kinshuk 01) Kinshuk, Hong, H. & Patel, A. (2001), 'Extended ITS Framework with Human Teacher Model'. In *Proceedings of the International Conference on Computers in Education. Korea*. <<http://www.icce2001.org/cd/pdf/p11/NZ005.pdf>>
- (Shiri-Ahmadabadi 99) Shiri-Ahmadabadi, M. (1999). Étude et modélisation des connaissances et raisonnement de l'apprenant dans un STI . Thèse présentée à la Faculté des études supérieures en vue de l'obtention du grade de Philosophie Doctor (Ph.D.) en informatique. Université de Montréal. Canada.
- (Vázquez-Abad, 01) Vázquez-Abad, F., Rodríguez, A., Ng, A., Zukerman, M., and Warfield, R. (2001). Bridging theory and practice in network design using web-based simulation. In *Proceedings of the 12th Annual Conference of the Australasian Association for Engineering Education (AaeE 2001)*, Queensland University of Technology. 26-28 September 2001. Australia. pp. 40-45.
- (Virvou, 02) Virvou, M. & Moundridou, M. (2002). Adding an instructor modelling component to the architecture of ITS authoring tools. In *International Journal of Artificial Intelligence in Education*. 12, Pp. 185-211.

## Acknowledgements

The author Arnaldo Rodríguez is funded by the Universidad de Costa Rica, the National Council for Scientific and Technological Research of Costa Rica and the Ministry of Sciences and Technology of Costa Rica.

This work was partially supported by NSERC and FCAR Grants of the Governments of Canada and Québec.