

Explor@ Advisory Agent: Tracing the Student's Trail

Karin Lundgren-Cayrol
Gilbert Paquette
Alexis Miara
Frédéric Bergeron
Jacques Rivard
Ioan Rosca

LICEF, Research Center
4750, Henri Julien, Suite 100
Montréal, Québec, Canada
Email: klundgre@licef.teluq.uqubec.ca

Abstract: This paper presents research and development of an adaptive web-based system called Explor@ Advisory Agent capable of tailoring advice to the individual student's needs, actions and reactions towards pedagogical events as well as according to diagnosis of content acquisition. Explor@ Advisory Agent consists of two sub-systems, the Advice Editor and the Student Advisor. The Editor allows course designers to enter the instructional structure as well as the content structure in form of a hierarchical tree, tagging corresponding pages, as well as entering rules (conditions and operations) for how and when advice are to appear. The Student Advisor displays the student's progression through the two structures by indicating what is achieved or completed. The system traces the student's progression according to knowledge acquisition through questions, questionnaires and results on test, and according to how the student navigates through the instructional material. In this manner, the student modeling relies on both the overlay and diagnostic modeling techniques. Future research aims at introducing peer or team collaboration and assistance by allowing the agent to match learners' progress by displaying the progression bars, if the learner chooses to participate.

Keywords: Student Modeling; Adaptive Systems; Interaction and Feedback; Intelligent Assistance

Introduction and background

As the number of online courses and number of student registrations increase exponentially, the need for software tools capable of adapting courses to the individual student's needs becomes crucial. Student or user modeling is the basis for adaptive systems. Student modeling is essentially trying to take into account as many traces (data) as possible about the students behaviour in the system and his hers knowledge status, and then attempt to adapt the tutoring model and/or the learning environment accordingly (Wenger, 1987, Stauffer, 1996, Tsinakos & Mararitis, 2000; Stephanidis et al., 1998;). Obviously, this is not a simple task, especially not in an open learning environment. However, hypermedia and Internet technologies can not only trace the path taken by a learner through a set of hyperlinked pages, but also adapt according to the amount of content learnt by asking questions and then adapt or propose different paths. Most online courses can be seen as a set of hyperlinked documents of differing format (text, image, sound and video), where various types of navigation tools are offered in order to guide the student in his endeavors to acquire knowledge. Brusilovsky (2001) proposes a taxonomy of adaptive hypermedia technologies, which is helpful in contextualizing the type or types of assistance a web-based course should or could include in order to adapt to the learner's need.

One of the classical problems in distance education is the question of providing adequate and individualized feedback to students, either by machine or human assistance. A plethora of communications applications over the Internet (email, forum, videoconference) has to a certain degree solved the problem with human assistance, that is distance education students can now profit from team and group learning, collaborative learning strategies and one-to-one tutoring, both in a synchronous and asynchronous mode (Henri & Lundgren-Cayrol, 2001; Greer et al., 1998). For distance education, the possibility of providing not only interactivity but also tailored advice to individual learners or groups of learners has become a reality.

The main purpose of adaptive systems is to, on one hand, diminish tutors' workload by automating assistance, and on the other hand to provide adapted and immediate feedback or advice according to each learner's behavior and knowledge. To develop adaptive hypermedia systems, it is helpful to differentiate between knowledge acquisition and navigation assistance (Conati et al., 1997; Asnicar, 1997). As for knowledge acquisition, the Intelligent tutoring Systems has served as a model for many learner adaptive systems (Wenger, 1987; Johnson, 2000; Conati & Van Lehn, 2001, Virvou & Moundridou, 2000; Nakabayashi et al., 1997, Weber, 1997), essentially building on the idea that the "tutor" guides the student through some course material by either posing questions, correcting mistakes or giving explanations according to a preconceived scheme attempting to let the student find possible answers or solutions to a problem. The main concern is to keep the student cognitively active. These type of systems can be found in the ITS literature and goes back to the early 70ths (Wenger, 1987; Paquette & Bergeron, 1989; Marcos et al, 1990; Anderson et al, 1995).

The year 1996 is seen as the turning point for adaptive hypermedia systems because of the explosion of WEB applications, number of users, that goes far beyond previous multimedia productions, and also because the number of research projects and theses focused on new technologies providing a whole new area of research (Brusilovski, 2001). Further, he points out that these new research projects were in fact real world systems or "research systems developed for real world settings" (p. 89). Within his classification system the Explor@ Advisory System can be seen as an Adaptive Recommendation System in a closed corpus, where the recommendations or advice adapts to the learner's need by tracing user data (characteristics), usage data (path taken in the system) and environmental data (system data) (Kobsa et al. (1999).

This paper briefly describes the Explor@ Virtual Campus and the Explor@ Advisory System (Paquette et al, 1996), its architecture and components, followed by some examples of its integration into different types of learning environments.

The Course Environment – The Explor@ Virtual Campus

Since 1992, LICEF research centre has been researching and developing prototypes of the Explor@ Virtual Campus, where many of the features have been transposed to courses at the Télé-université (Paquette, 1995; Paquette et al., 1995; Paquette et al. 1996; Paquette, 1997; Paquette et al, 1997; Lundgren-Cayrol & De la Teja, 1998; Dufresne, A & Paquette, G, 2000). Evolving from an intranet prototype to a server residing system, Explor@ Virtual Campus now counts not only university applications, but also professional training courses.

Most WEB-based courses can be described as a set of hyper linked pages organized according to some principle coupled with some navigational assistance, but usually lacking any kind of individual adaptation capacity. The organizing principle could be the instructional structure, the content structure, a set of competencies or a set of resources needed to carry out some learning task. In the Explor@ Campus courses, via the Explor@ Resource Navigator, a student can access a course from multiple points of view, namely by navigating through the course using the course site, through the tree-structure and by accessing course resources directly from spaces in the Resource Navigator. Figure 1 shows a typical Explor@ learning environment, where the course site can be seen in the background, on the bottom left the Resource Navigator with its five spaces: self-management, information production, collaboration and assistance. The progression bar can be seen in the upper right corner, and below it, the advice window for this particular page.



Figure 1 The Explor@ Course Environment with Course site and the Explor@ Resource Navigator.

The Advisory Agent operates in three of the spaces, the self-management space by providing the learner with a progression bar, in the collaborative space by allowing a student to view other students' progression bars and to contact peers, in the assistance space by displaying context-specific advice.

The Advisory Agent The Architecture

The Explor@ Advisory Agent is one of the agents in the Explor@ Virtual Campus consisting of an Advice Editor and a Student Advisor available through the Internet. The editor allows course designers to insert both content and task specific advice into a web-based course. The underlying student model is based on the learner's progression through a set of instructional events (instructional structure) and the content (cognitive structure). The student model relies on traces captured by the system both according to the overlay and diagnostic modeling techniques.

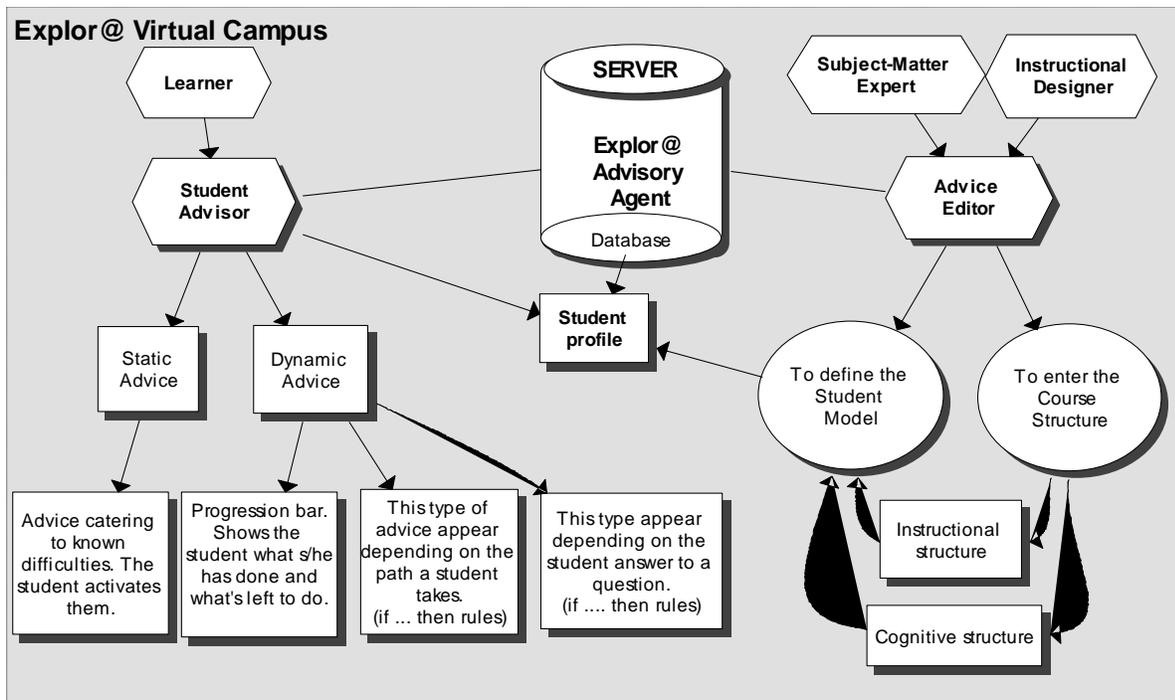


Figure 2. Architecture and components of the Explor@ Advisory Agent

According to the MISA Instructional Design Method (Paquette, in press) the instructional learning events and the content is modeled via a graphical tool called MOT. These models can be of four kinds: procedural, conceptual, prescriptive or hybrid (processes and methods). The models also include links that determine the relationship between two or more knowledge units or instructional events and its resources. The models are similar to conceptual maps. Using the Advisory Editor, these models can be imported and automatically translated into tree-structures with nodes, sub-nodes and leafs.

The progression bar reflects the learners progression in the two structures according to the weight of importance and the progression mode. The progression can be programmed according to four modes described in Table 1. Each node or sub-node has its corresponding progression bar and weight of importance. The weight of importance reflects the weight (%) that a node has on its main-node, a leaf has on its sub-node. For example, the course X has 4 modules, where modules 1, 3, and 4 is worth 90% of the course, and module 2 is worth 10%. The agent takes into account the weight of importance given to a node, sub-node or leaf when calculating the length of the bar. These two rules determine the progression state in the structures.

Type of progression	Rule	Example
Sequential mode	Sub-nodes or leafs have to be carried out in a sequential manner for the node to be considered completed.	The learner must complete Module 1 before Module 2 or activity 1.1 before activity 1.2.
Modular mode	Sub-nodes or leafs can be carried out in any order, but each must be fully completed before the bar shows the state of the progression according to weight.	Activity 1.1, 1.2 or 1.3 can be completed in any order. Activity 1.2 is worth 40% of Module 1, and Activity 1.3 30%. Activity 1.1 is worth 30%. If activity 1.2 is carried out in full, the bar moves 40%, if activity 1.1. it moves 30%.
Parallel mode	Sub-nodes or leafs can be carried out in any order, the agent calculates the average of completion and weight, and then adjusts the bar accordingly.	Activity 1.1 consists of 3 exercises. All exercises have to be completed. The agent calculates a mean completion rate according to their respective weight.
Optional mode	Only some of the sub-nodes or leafs must be completed for the bar at the node level to insert the completion state.	Activity 1.1 consists of 3 exercises, the learner only have to finish one for the activity to be completed.

Table 1. Type of progression mode and corresponding rule

To adapt the progression bar to an individual’s path through the course, the designer can put a time limit on a page. For example, a student browses through the course site, but the progression bar will not change because the designer has placed the condition that for the progression bar to indicate completion, the learner has to stay on the page at least 2 min (see Figure 6 below).

Domain expertise is commonly represented by a conceptual model which is fairly easy to transpose into an hierarchical tree structure, here called the **cognitive structure**. Nodes represent main knowledge units, sub-nodes its sub-concepts and leafs the attributes or facts that define a concept. All domain expertise can not be described by a conceptual model, and it is the designer’s task to represent it in a way that it can be transposed into a tree-structure. Each node or leaf can carry a static or dynamic advice. Since a leaf represents the smallest knowledge unit which can quite easily be diagnosed, this is where the diagnostic question feature is made available in the editor.

Pedagogical expertise is represented by a procedural model, called **the instructional structure**. In this model, the nodes represent main events, sub-nodes the activities within a learning event, and the leafs represents the tasks to carry out to complete an activity. It is organized according to what is perceived by the course designer to be the most effective and efficient way of learning some course material.

Another angle of the student modeling technique available in the Explor@ Advisory Agent is the **self-monitoring feature**. The learner manages his own learning progress by deciding whether the agent’s diagnosis is accurate by modifying the progression bar to the perceived level of performance in the two structures. The agent takes this information into account and adapts advice accordingly. Advice are appearing according to a rule based “if ... then” system explained in Table 1. In distance education, this feature is essential in order to encourage learners to become self-directed and to take on responsibility for their own learning (Bull, 1997; Ruelland, 2000).

Briefly, the student model takes into account the following traces (data):

- The learner profile (group, email, name, entering and leaving time)
- The learner’s navigation path in the structures (instructional and cognitive)
- The resources used (templates, applications and documents of all formats)
- Length of time spent on a page

- Response to questions
- Modification of the progression bar during learning sessions

Table 2 . The type of advice, student modeling method, the designer’s task and the learner view.

Type of advice	Method	Designer’s task	Learner view
Dynamic <ul style="list-style-type: none"> • If condition X (action or set of actions) then action Y 	Overlay	For each sub-node or leaf in the structures: <ul style="list-style-type: none"> • Identify location (URL) • Identify conditions • Compose advice message 	Pop-up advice
	Diagnostic	For each leaf, where diagnosis is desired: <ul style="list-style-type: none"> • Identify location (URL) • Compose questions • Compose answer categories • Compose the message of the advice according to each answer For each answer category: <ul style="list-style-type: none"> • Identify conditions • Compose answer 	Questions followed by tailored advice
	Self-monitoring	<ul style="list-style-type: none"> • Identify type of progression • Identify weight of importance • Define type of manipulation (tutor/learner) 	Progression bar
Static <ul style="list-style-type: none"> • If on page X then set of advice X 	Assistance space in the Resource Navigator	<ul style="list-style-type: none"> • Identify which pages (URLs) the advice should appear on • Compose context specific advice. 	Advice on demand

The Advice Editor

The Editor has gone from being a rather complicated programming interface to an easy “fill-in form” (see Figure 3 and 4). The course designer is asked to enter the instructional tree structure consisting of nodes (e.g., course modules/principal knowledge unit), sub-nodes (e.g., activity level) and leafs (e.g., task level). Nodes and leafs can be added, deleted or edited at any point.

For each node the course designer enters the title of the node, the abbreviation, the URL, the weight of importance (%) and the type of progression, as explained in Table 1. The designer then decides whether s/he wants to insert a contextual advice, which will appear in the Student Advisor. At the leaf level, the designer enters the above parameters plus decides whether or not to include the self-monitoring feature, that is whether the student will be permitted to manipulate the bar or not, and for how long a student has to stay on the page in order to display pop-up advice or diagnostic questions.

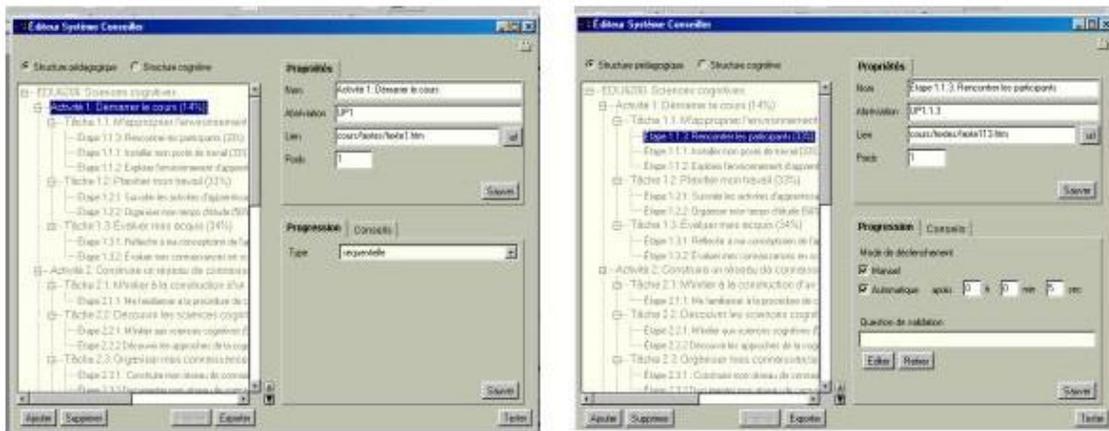


Figure 3 The node editor is shown on the left, and on the right the editor of leaf parameters.

Figure 4 shows the diagnostic question editor, where the designer can enter the questions, answer categories, responses and conditions.

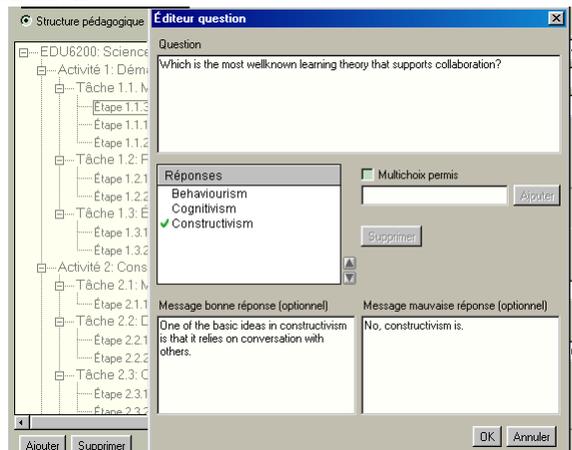


Figure 4 The Question editor

When the designer has finished entering the static and dynamic advice, the editor can be put in a validation mode, which is the simulation of what the learner will experience once the course is online. The user can switch from validation to editing mode in order to modify and verify the advisory system, until satisfied. This feature has proven very helpful to course designers to determine whether advice really are inserted at the right place and whether there are enough or too many advice, before the course is online.

The Student Advisor

As mentioned beforehand, the student advisor is actually present in three ways :

1. By displaying diagnostic questions and pop-up advice at appropriate while navigating in the course site
2. By making available contextual advice in the Assistance space in the Resource Navigator.
3. By displaying the progress bar for the two structures in the Self-management space

The following pictures illustrate the different types of advice provided by the Student Advisor. They are taken from different learning environments, where the Explor@ Advisory Agent has been implanted. Figure 5 shows how diagnostic questions are displayed.

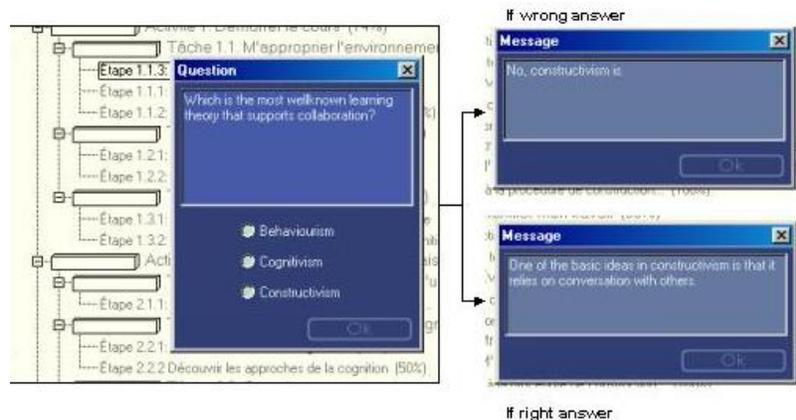


Figure 5 Question with multiple choice with two possible answer categories (right or wrong)

Figure 6 displays the progression bar of the instructional structure in an academic course. The checkmark indicates whether the student agrees with the Advisor's evaluation the progress. By highlighting a node or a leaf and then clicking the right mouse button the student can go directly to the corresponding URL to validate its correctness or ask for corresponding advice.

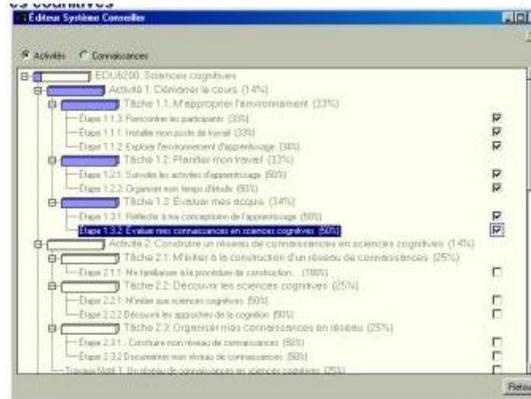


Figure 6 Progression bar in an academic course.

The picture below shows 3 contextual advice for a specific page in a course. Since the third advice is highlighted its message is displayed in the window to the right. On screen, the student can also see the page, but because of lack of space it is not included here.

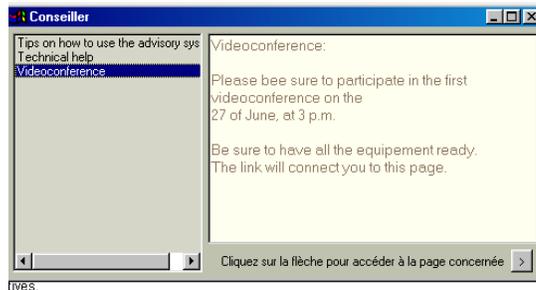


Figure 7 Contextual advice

Conclusions

Since 1998, the Advisory Agent has been refined and implemented in several on-line courses and training situations. The first working prototype was the Job Search Advisor (Paquette et al., 1998), which was implanted in several academic courses at the Télé-université. Some of these implementations have been described elsewhere. You can read about the CVAC (Virtual Center for Continuous Learning), an application that was developed for three professional orders (de la Teja et al, 2000; Damphousse, 2000). Another example is a course on the "Exploitation Code" for linesmen, where the Advisor Agent was implemented in the form of self-diagnostic tests (Damphousse, 2001).

The next step in the endeavours to refine the Explor@ Advisory Agent is to expand its collaborative capacities and to adapt it to all the actors (tutors, designers and managers) in the Explor@ Virtual Campus. The collaborative features aim matching students by tracing the structures and then to program the agent so that it can communicate whether and when help is needed.

Bibliography

- Anderson, J. R., Corbett, A. T., Koedinger, K. R., and Pelletier, R. (1995). Cognitive tutors: Lessons learned. *The Journal of the Learning Sciences* 4,167-207.
- Asnicar, F.A. & Tasso, C. (1997). IfWeb: a Prototype of User Model-Based Intelligent Agent for Document Filtering and Navigation in the World Wide Web. *Proceedings of the workshop "Adaptive Systems and User Modeling on the World Wide Web", Sixth International Conference on User Modeling, Chia Laguna, Sardinia, 2-5 June 1997.*

- Brusilovsky, P. (2001). Adaptive Hypermedia. Kluwer Academic Publishers: *User Modeling and User-Adaptive Interaction* (11). 87 – 110.
- Conati, C., Gertner, A., VanLehn, K. and Druzdzel M. (1997). On-Line Student Modeling for Coached Problem Solving Using Bayesian Networks, Jameson, A., Paris, C. and Tasso, C. (Eds.) *User Modeling: Proceedings of the Sixth International Conference*, UM97, 231-242. Vienna, New York: Springer Wien New York.
- Conati, C. & Van Lehn, K. (2001). [Providing adaptive support to the understanding of instructional material](http://www.pitt.edu/~vanlehn/distrib/titles.html). 2001. <http://www.pitt.edu/~vanlehn/distrib/titles.html>
- Damphousse, L. (2000). A Virtual Centre for Professional Training using Explora. *Proceedings from TL-NCE, Toronto*, 5-7 November, 2000.
- Damphousse, L. (2001). Explor@ un portail de diffusion et de gestion de téléformation. *Journées-réseau sur l'appropriation des TIC 2001*, Université de Québec à Hull. le 3 mai 2001
- Dufresne, A. & Paquette, G. (2000). ExploraGraph: A flexible and adaptive interface to support distance learning. *Actes de l'EdMedia 2000, Montréal*, pp. 269-274.
- De la Teja, I., Longpré, A. & Paquette, G. (2000). Designing Adaptable Learning Environments for the Web : A Case Study. *Proceedings from ED-Media 2000*, Montreal, Quebec, Canada.
- Girard, J., Paquette, G. & Giroux, S. (1997). Architecture de Système Coseiller multiagent sur la collaboration dans un Système d'apprentissage. *Research Report*, LICEF, Télé-Université, Montréal, Canada.
- Girard, J., Paquette, G., Miara, A., & Lundgren-Cayrol, K (1999). Intelligent Assistance for Web-based TeleLearning. *AI-ED'99 Conference Le Mans*, France, July 1999
- Greer, J., McCalla, G., Collins, J., Kumar, V. , Meagher, P., & Vassileva, J. (1998). Supporting Peer Help and Collaboration in Distributed Workplace Environments. *International Journal of Artificial Intelligence in Education* 9, 159-177.
- Johnson, W. L. (2000). Pedagogical Agents. http://www.isi.edu/isd/carte/ped_agents/pedagogical_agents.html, *Center for Advanced Research in Technology for Education (CARTE)*.
- Henri, F. & Lundgren-Cayrol, K. (2001). *Apprentissage collaboratif à distance. Comprendre et concevoir les environnements d'apprentissage virtuel*. Québec, Canada : Presses de l'Université du Québec.
- Kobsa, A. (2001). Generic User Modeling Systems. *User Modeling and User-Adapted Interaction* 11: 49-63.
- Lundgren-Cayrol, K. et De la Teja, I. (1998). The Virtual Campus Global Prototype: The Results. *Proceedings from ED-Media-98*, Freiburg, Germany.
- Marcos, B., Lundgren, K., Lapointe, J. Therien, L., Veillette, M. & Boily, A. (1990). Thermex: An educational expert-system for thermodynamic students. *Canadian Journal of Educational Communication*, 19(3), 217-235.
- Nakabayashi, K., Maruyama, M., Koike, Y., Kato, Y. Touhei, H. & Fukuhara, Y. (1997). Architecture of an Intelligent Tutoring system on the WWW. *Artificial Intelligence in Education B. du Boulay and R. Mizoguchi* (Eds.) IOS Press. (pp. 39-49)
- Paquette, G., Pachat, F., Giroux, S. & Girard, J. (1996) Generating Advisor Agents for Existing Information Systems. *Journal of Artificial Intelligence in Education*, (7) 3/4, pp. 349-379
- Paquette, G., Ricciardi-Rigault, C., Bourdeau, J., Paquin, C. & Liégeois S. (1995). Modeling a Virtual Campus. Environment for Interactive Distance Learning. *ED-Media International Conference*, Graatz, Austria, juin 1995.
- Paquette , G., Ricciardi-Rigault, C., Paquin, C., Liégeois, S. & E. Bleicher (1996). Developing the Virtual Campus Environment. *ED-Media International Conference*, Boston, USA, June 1996.
- Paquette, G. (1995). Modeling the Virtual Campus. In *"Innovating Adult Learning with Innovative Technologies"* (B. Collis and G. Davies Eds) Elsevier Science B.V., Amsterdam, 1995.
- Paquette, G. (1997). Virtual Learning Centers for XXIst Century Organizations. In *F. Verdejo and G. Davies (eds.), The Virtual Campus* (pp. 18-34). Champman & Hall, London,
- Paquette, G., C. Ricciardi-Rigault, C., de la Teja, I. & C. Paquin. (1997) Le Campus Virtuel : un réseau d'acteurs et de ressources. *Revue de l'Association canadienne d'éducation à distance*, volume XII, No1/2, pp.85-101
- Ruelland, D. (2000). Vers un modèle d'autogestion en situation de télé-apprentissage (Towards a Self-management Model for TeleLearning Setting. *Unpublished Thesis*, Faculté des études supérieures, the University of Montréal, Montréal, Canada.
- Stauffer, K. (1996). Student Modeling and Web-based Learning Systems. Arthabaska University: Canada. <http://ccism.pc.athabascau.ca/html/students/stupage/Project/initsm.htm>
- Stefani, A. & Strapparava, (1999) Exploiting NLP techniques to build user model for Web sites:the use of WordNet in SiteIF Project. *Proceedings of the 2nd Workshop on Adaptive Systems and User Modeling on the WWW*.

- Stephanidis, C., Paramythis, A., Akoumianakis, D., & Sfyarakis, M. (1998). Self-Adapting Web-based Systems: Towards Universal Accessibility. In C. Stephanidis & A. Waern (Eds.), *Proceedings of the 4th ERCIM Workshop on "User Interfaces for All"*, Stockholm, Sweden, 19-21 October
- Tsinakos A. A. and Margaritis K. G., (2000). "[Student Models: The transit to Distance Education](#)", *European Journal of Open and Distance Learning* (EURODL), ISSN 1027-5207, Vol. 11/2000.
- Virvou, M. and Moundridou, M. (2000) A Web-based Authoring Tool for Algebra-Related Intelligent Tutoring Systems. *Educational technology & Society*, 3(2). http://ifets.ieee.org/periodical/vol_2_2000/virvou.html.
- Weber, G. and Specht, M. (1997). User Modeling and Adaptive Navigation Support in WWW-based Tutoring Systems, <http://www.psychologie.uni-trier.de:8000/projects/ELM/Papers/UM97-WEBER.html#Heading7>