A connectionist approach for adaptive lesson

Hassina Seridi-Bouchelaghem, Toufik Sari, Mokhtar Sellami

A connectionist approach for adaptive lesson

Hassina SERIDI-BOUCHELAGHEM[1], Toufik SARI[1], Mokhtar SELLAMI[1]

[1] Laboratoire de Recherche informatique (LRI) ; Badji Mokhtar University ; B.P 12 Sidi Amar. Annaba
seridi@yahoo.com, tou_sari@yahoo.fr, sellami@lri-annaba.net

Abstract

This paper investigates the use of computational intelligence for adaptive lesson generation in a distance learning environment. A method for adaptive pedagogical hypermedia document generation is proposed and implemented in a prototype called *KnowledgeClass*. This method is based on specialized connectionist architecture. The domain model is represented in a connectionist-based system which provides an optimal didactic plan composed of a set of basic units. The generated didactic plan is adapted to the learner's goals, abilities and preferences.

Keywords: Adaptive learning, learner model, domain ontology, neural networks.

Résumé

Cet article étudie l'utilisation d'une technique de l'intelligence artificielle pour la génération adaptative de cours dans un environnement d'enseignement à distance. Une méthode pour la génération pédagogique adaptative de document hypermédia est proposée et implémentée dans un prototype appelé *KnowledgeClass.* Cette méthode est basée sur une architecture connexionniste. Le modèle de domaine est représenté par un réseau de neurones classique qui fournit un plan didactique optimal composé d'un ensemble d'unités de base. Le plan didactique produit est adapté aux buts, aux niveaux et aux préférences de l'apprenant.

Mots-clés : apprentissage adaptative, modèle de l'apprenant, ontologie du domaine, réseau de neurones.

Introduction

Intelligence in a Web-based learning environment (LE) is ensured with tools enabling efficient management of the available information. One of the important keys is the system capability to provide the learner with personalized, adaptive and effective teaching [1]. Thus, our research is focused on the adaptive systems to accommodate diversity in the learner needs, abilities and preferred learning style [2]. Towards this direction, this paper investigates the use of a connectionist method to support the lesson generation process.

For this personalized learning, two models should be built [3] : the domain and the learner model. A connectionist-based algorithm for selecting the appropriate documents is proposed and the interactions between the connectionist architecture and the learner model are described. In what follows, section 2 reviews the lesson generation process and learner modeling methods, section 3 explains the context of the project, and section 4 describes the interaction's layers in the generation process. Sections 5, present the domain and learner models respectively and 6 the generation process methodology and 7 experiments.

State of the art

Generating adaptive pedagogical documents using composition of documents, or documents fragments, must exploit an existing representation of the learning.

Generation of adaptive and dynamic courses

L.Sommaruya and the company Media Tech developed NURAXI, a system for the design, the production and the management of intelligent environments of remote training [4] making possible the creation of dynamic courses (ways in the teaching activities). The project KARINA developed at the school of Mines of Alès [5] had for purpose to provide a set of tools allowing teachers to integrate in their productions the course elements developed by other authors, joining to that, objectives of ARIADNE [6]. Our work is closer to these two projects. Therefore, the connectionist approach to a content-addressable database provides the learner/educator/system with an effective way to retrieve the data for different uses, such as the evaluation of the learner understanding level through the construction of several tests corresponding to several concepts and the difficulty level [7 ; 8] ; or the generation of lessons meeting the learner's needs according to his understanding level and preferences [9].

The learner model

The aim of the learner model is the guidance of the tutor in taking the teaching decisions better adapted to a learner [10]. The learner model must take into account the learner's current knowledge characteristics, the goal of the training session, etc. In this model, a key question is: what is to be represented? Overlay [11] and Buggy models [9 ; 12] are knowledge representation based approaches which determine the way for expressing the learner's knowledge. In Overlay models, the student knowledge is considered as a subset of the domain knowledge which should be incremented. This knowledge is set up and exploited by the evaluation module. However, Buggy models enable further modeling of faulty information in the system knowledge. The main concern of this paper is to generate lessons and try to help learner to see unassimilated concepts by giving a course specific to its current situation. The course generator tries helping

learners in re-examining the the concepts they don't learn.

Project description and context

The current project is a part of a global educational environment of our university allowing teachers the use of the Internet for diffusing their courses and to reuse existing bricks of courses developed by other teachers. To this end, a new approach of adaptive pedagogical documents composition is implemented in a system called "*KnowledgeClass*". A connectionist architecture, selecting pedagogical materials faciliting the selection and the filtering of courses' bricks is described. We choose the *'Computer Network'* as the matter to be learned in an intranet environment. A common methodology is used and described.

Methodology for adaptive documents generation

The teaching decisions are made at the end of the knowledge evaluation, the identification of the learner preferences and the domain model consultation. The generation of the adaptive didactic plan is made up by the composition of pedagogical documents adapted to the learner profile. The basic idea is to use the learner and the domain models to extract and organize the knowledge in order to satisfy the learning goal. The generation process is carried out in three stages:

- Selection of the learning goal by the learner.
- Planning the content: selection of the suitable concepts for the learning goal.
- Planning the presentation : selection of the hand-annotated basic units and their organization in didactic plan for delivering to the learner according to a defined teaching strategy.

Our approach supports the following adaptation mode: the learner selects a learning goal at the first stage (fig 1). Each learning goal relates to a subset of concepts of the field of knowledge. The whole of the concepts balanced and concerned are extracted from the domain model. The basic units which will constitute the course are selected, filtered and organized in a didactic plan to be presented. Several tests sessions will evaluate the concept's knowledge associated to the selected learning goal. During the interaction process *System-Learner*, the evaluation modules keep the trace on the performances of the learner and estimate its level of comprehension.

The results of the evaluation procedure influence the course generation process. Our main goal is to simulate tutor's methodology in selecting the appropriate course incorporated in basic units. The generated course must be adapted to the learner's abilities, prerequisites and preferences.

A common method to course generation can be divided in three steps:

Step 1: *pre-test*. In this phase, the initial learner knowledge is tested in order to identify his level and the concepts to be learned.

Step 2: Each concept of the course to be learned is associated to a set of basic units. To each basic unit, a pedagogical role is assigned. Two classes of roles are distinguished : basic roles (BR) and reinforcing roles (RR). In other words, basic units with (BR) are selected for the first stage of learning and are generated for learners having medium/high level. Basic units with (RR) are generated for special case learners, needing more illustrations and further explications.

Step 3: *post-test*. The level of the learner understanding is computed. The generator decides to pass on the learner or in contrast to reinforce his knowledge.

Non-understood concepts are presented to the learner using other more specific basic units.
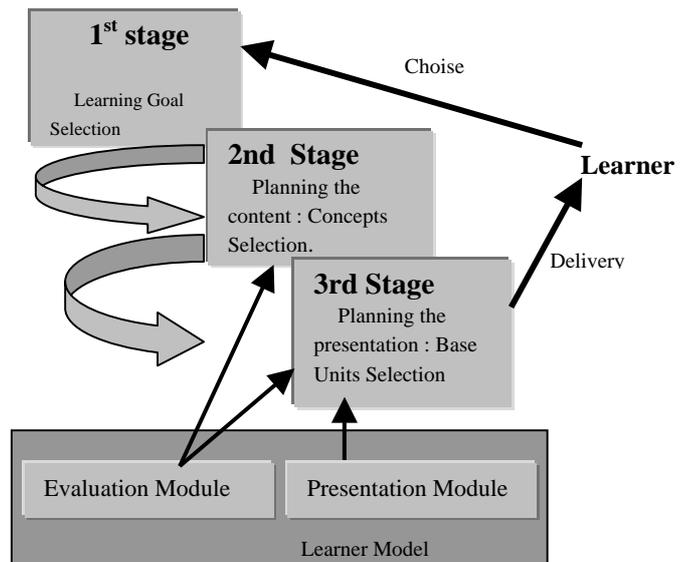

Fig1 The interaction Levels

Architectural components

The general architecture of our system is given in fig 2. The processing is hold by the server which gives the access for the users to a dynamically generated interface over Internet. The architecture consists of a server, a navigator and other services. The access to the several structures is carried out by the course generator. The main components of the system architecture are presented bellow.
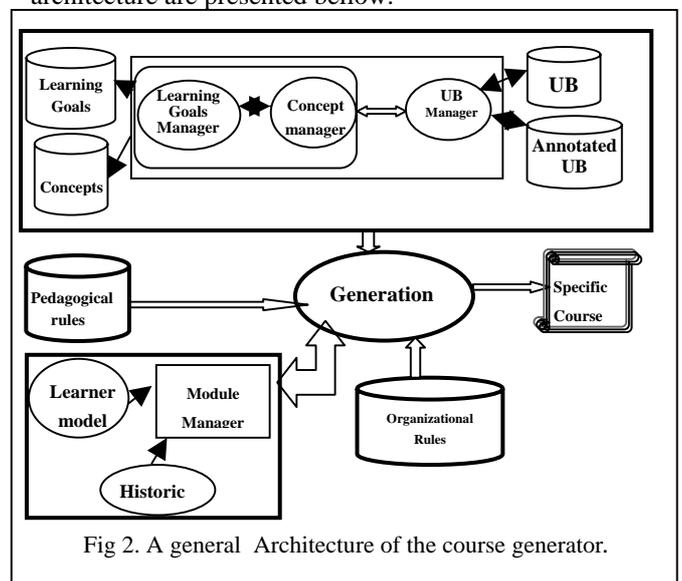

Fig 2. A general Architecture of the course generator.

The domain model

In a learning context, it is useful to partition the domain knowledge in order to ease the evaluation process [10]. A main issue in the development of an educational system able to support pedagogical decisions is the domain knowledge including multiple curricular view points [13]. To this end, the domain model is structured in three hierarchical levels of abstraction: learning goals, concepts and basic units.

The learning goal is a mental structure represented by a conceptual network which can be evaluated only with the behavior of the learner [14].

Concepts are evoked by the learning goals. A concept can concern more than one learning goal and the learner is evaluated on concepts of the chosen learning goal. The relationships between concepts determine the nature of links between them. Two relations are defined between the concepts : prerequisite and subconcept. The main concepts for a learning goal are identified and should be fully explained in the HTML pages using text, images, examples, exercises,… Prerequisite concepts are less important but essential for the learner to understand the main concepts of a goal. However, the main concept is composed of several subconcepts. Table 1, above, presents a learning goal with its associative concepts, referred on the course Computer Network teached in our university.

Table 1. Learning goal "Comprehension in the Computer Network domain, the concept of LAN topology". Each row contains a main concept followed by its prerequisite and Sub-concepts.

Main Concept	Prerequisite Concept	Sub-concept
LAN topology	Netwwork nodes, Types of connections, LAN	-
Types of connections	-	Bus, Star, Ring topology
Star topology	Point to point connection, Polling.	-
Bus topology	Multidropconnections.	-
Ring topology	Error rate	-

Learning goals, concepts and relations between concepts are defined in a domain ontology. While using a domain ontology, we try to adapt new techniques of knowledge representation to educational systems. The main interest is the modelling and the representation of knowledge based on the semantics. An ontology of the domain referred on the course "Computer Network" is conceived and modeled with the use of the standard RDF.

Basic units

A basic unit (UB) is a multi-media document having intrinsically a teaching quality, i.e. which it can be used within the framework of the knowledge transmission.

It appears convenient to share the basic units and to properly index them so that they can be found and re-used. The basic units can be created in our system by the author or imported from external sources and integrated in the system in the form of meta-data. [15].

Learning Technology Standardization Committee of the IEEE was based on work of ARIADNE [16] and of the IMS [17] to work out a proposal of a standard defining the structure of the pedagogical object metadata or LOM [18]. The description of our basic units is based on the ARIADNE recommendation. Three types of descriptors:
- general information {Identifier, Title, Authors, Date, Language,..},
- semantics of the (UB) {Learning goal, Main concept, Other concept,..},
- pedagogical attributes {Document type, Format, Level, Difficulty, Duration, Pedagogical rule…}.

Each basic unit will have a role to play at the time of the course organization. The roles are statement, solution of an exercise, explanation, exercise (QCM, Test True/False), conclusion, a proof, explanation, theorem, definition...

Two categories of pedagogical roles are distinguished: roles themselves and roles played by the UB in an activity becoming familiar, clearance or reinforcement of the non assimilated concepts.

At the time of the first activity, the UB whose roles are: simplification, comparison, remember, digression are presented to the learner.

The learner model

The learner model is the key element of our system since it intervenes in all levels of the learning process. It is composed of two parts:
- *Static part*: this information is static and changes rarely during a learning session and consists of: identification of the learner such as the name, surname, specialty, the diploma or the prepared certificate, the language, the learning style and the learning goal to acquire
- *Dynamic part:* this information changes with learner's evolution during the learning session, the way followed by the learner to accomplish activities in relation with the followed learning goal and the acquired competence for their concepts.

Every action of the learner is analysed and saved in his learner model. The later model must indicate at every step, the learner level of knowledge. The learner is evaluated on the concepts of the selected learning goal.

The static part of the learner model is initialized by the learner (questionnaire) and the dynamic one by the system. This task consists of initializing for each selected learning goal the different entities that better describe it.

Each time the learner visits a proposed basic unit of the didactic plan, the dynamic part in his model is updated, taking into account the learner's behavior. It is done for each concept of the learning goal on the basis of the existing information in the model.

Example: the evolution state of the learner for the concept must be done conforming to the form heuristics:

If (state=high) and (answer=debit) then state= medium.
The learner evolution is *high, low or medium*

Pedagogical rules

Teachers follow different strategies to instruct students some concepts. Our generator employes pedagogical rules in order to select the appropriate concept to be presented to the learner. An example of them is illustrated below:
Rule1: The concept is selected if all its prerequisite concepts are assimilated.
Rule2: The concept is selected if the learner level for the concept is "Medium" and will be presented with further (UB).

Organization rules

Several rules were built. Some of them are general but can always be applied, others are more specific to some learning styles. General rules are valid and implemented whatever the context. For example, " *an introduction to a given concept precedes all other instructions concerning the same concept*".

Rules are also relative to the (UB) organization, according to the constraints imposed by the concept. For example, " *when a concept is composed of sub-concepts, their corresponding UBs will be ranked before those of the main concept*".

Organizational rules, specific to a learning style are applied by the system. Logical and intuitive learning styles refer to a preferred organization of the basic units. A logical learner prefers clearly-structured courses, starting from A and logically building to Z, presenting theory before practice, values facts and details, dislikes ambiguity.

An intutive learner prefers flexible courses, starting from wherever he chooses, practice before theory, values creativity, and dislikes rigidity. For this purpose, some organizational rules are constructed taking into account the two learning styles.

The generation process

The generation of the adapted documents followes the Selection-Organization. Process. (Fig 3) Searching basic units consists in determining, according to a specified learning goal, a whole of conceptual descriptions corresponding to the part of the domain model to be treated.

The first phase is the production or the selection of resources that may be used or reused in the constuction of training course. These resources may have been produced either by a unique author or by different authors. In order to build courses, the generator needs to know the learner's profile. A connectionist approach is in charge of selecting among the available indexed basic units that can fulfill the conceptual description of the learner's goal.

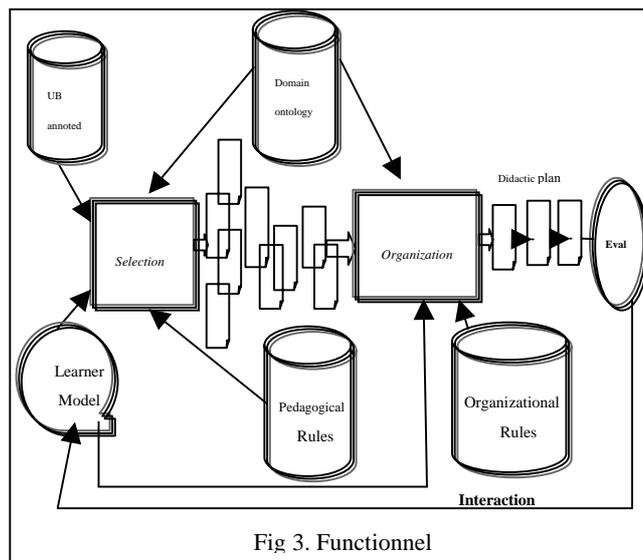The second stage is the organization of the selected basic units according the organizational rules.

Fig 3. Functionnel

The connectionist approach for the basic units selection

Two Multi-layer Perceptrons (MLP) with one hidden layer are constructed to model the selection phase of used information and the decision to make upon learner's understanding.

Artificial neural net (ANN) models have particular properties such as ability to adapt, to learn or to cluster data [13]. These models are inspired by our understanding of the biological neural system and are made up with a total interconnection of simple computational elements corresponding to the biological neurons. Each connection is characterized by a variable weight that is adjusted during the "*training stage*". MLP for multi-layer Perceptrons are ANNs that try to build a correspondence between input vectors and outputs ones. These latter are known as '*desired outputs*'.

An artificial neuron calculates a function of all incoming values corresponding to the neurons outputs of the previous layer multiplied by the link's weights. In this neural net each output neuron in the output layer is assigned to a *basic unit*, while input neurons in the input layer represent the *concepts* related to the learning goal of the course. The hidden layer is one that does the most computations. In the conception phase, the number of hidden neurons is heuristically initialized and will be manually modified during the training stage. The used algorithm for training the MLP is called '*BackPropagation*' and it works by calculating the difference between the neural net responses upon input vectors and the desired outputs. If this difference is greater than a predefined threshold a back return is done in order to adjust the link weights. Only links exciting '*bad neurons*' are modified. Bad neurons are those having an important error against desired output.

This algorithm is executed for each input-output vector and repeated several times until the convergence [19].
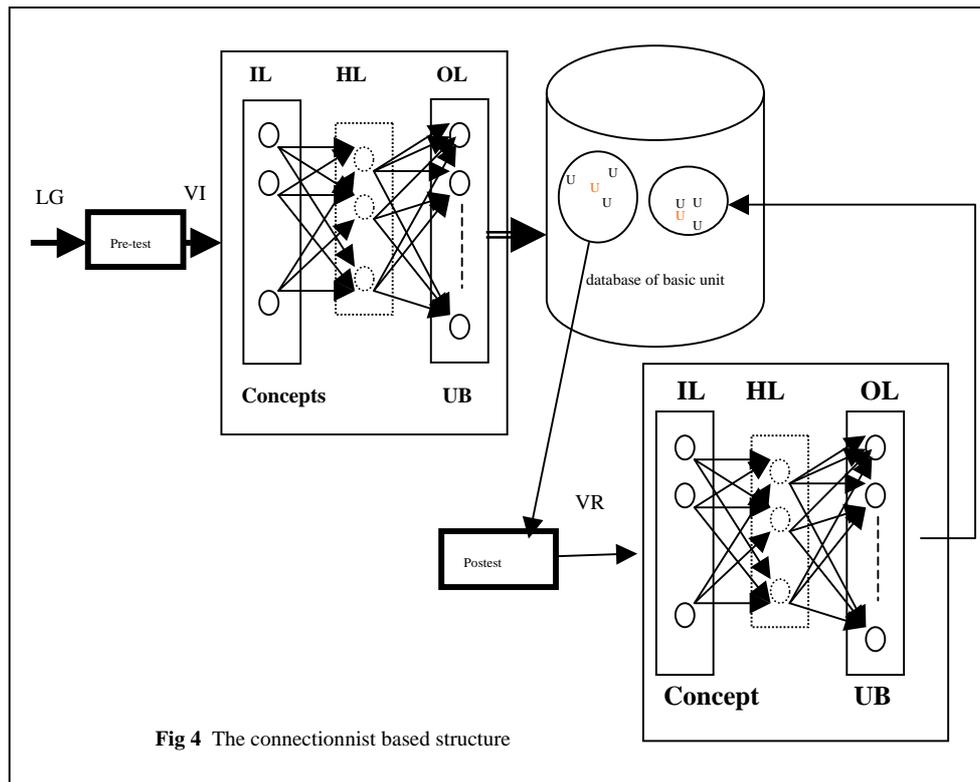
Fig 4 The connectionnist based structure

The first neural network is used to select the appropriate (UB) for the learner in the first stage of learning. The input layer presented the concepts of the course, one neuron per concept. The input vector (VI) is a {-1,1} set where the values $VI_i=1$ indicates that the corresponding concept (c_i) is important to the learner and the values $VI_i = -1$ means that (c_i) is not. The (VI) values are set by the evaluation module for the pre-test phase. The output layer is assigned to the basic units with basic roles (BR). (see Fig 4). The second neural network intervenes when the learner don't succeed the post-test of the concepts. This later generates a vector of marks related to concepts called reinforcing vector (VR). Three measure are used : 0 for low, 0.5 for medium and 1 for high learner level of the concept understanding. The (VR) is used as input values for the second neural network, the output layer for selecting (UB) having reinforcing roles. Note that, when the learner's knowledge with respect to a concept is characterized as non acquired, a value of approximately 1 is assigned to the corresponding input neuron in the second neural network, which means that the learner has to study this concept with further (UB), more simplified. On the other hand, a value of approximately 0 is assigned when the learner's knowledge on a concept is evaluated as acquired.

Organization of the course

In the organization phase, the system assigns an order between basic units to allow the system building a didactic plan. The final structure of the course is dictated by the learner learning style.

To every learning style declared is associated a set of organizational rules that describe the document structure. The rules determines the order and the structure of the wished educational document. The learner can prefer an example before the explanation of the concept. Another can want the explanation of the concept before the example.

Implementation

This new approach support learning in open corpus educational courseware that we are currently investigating in our university. The mecanism behind this approach is presented and a system called "KnowledgeClass" that implements this approach is introduced. Some results of a classroom study of this system are presented. The learner interface part of the "KnowledgeClass" system is implemented as a set of dynamic HTML pages with embedded Javascript programing. The generator system is implemented with use of Java programming and it use both Netscape Communicator and Internet Explorer. It runs locally under Windows 95/98/NT/2000/XP. A version of the system is currently in programming under Linux with use of graphical library QT. Our approach was tested in the Logic course in which 9 concepts are identified (Propositional, JMP, CNF, NEG, AND, OR…). 35 Basic units are designed in 15 basic pedagogical roles (Introduction, Description, Example Exercise, Explanation, Theorem,…) and 10 reinforcing roles (Simplification, Comparison, Reformulating, Discussion,...). The first Neural network is composed of 9 neurons in the input layer, 15 neurons in the hidden layer and 20 neurons in the output layer. It was trained on 40 learner profiles and tested on 10 other

unknown profiles. The obtained results with human generated (UB) are very encouraging : 15 learners didn't succeed the post-test and so the results of 10 of them were used to train the second neural network and the other for the test. The second neural network is constructed with 9 neurons in the input layer, 10 neurons in the hidden layer and 15 neurons in the output layer.

Conclusion

This paper has described some important parts of an adaptive learning environment, how they are designed and interact. The constitution of adaptative training plan on Internet is an important centre of research in the teaching at distance. We have introduced an approach that permits to realize this adaptative capacity from a mode of conceptual navigation. The approach presented, accommodate the goal of improving the learner's learning process by matching the lesson to their level of understanding and needs. The results of our preliminary study shows that our approach is promising for building dynamic adaptive learning.

References

Books

Wenger, E. 1987. Artificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to the Communication and Knowledge », Los Altos (CA), Morgan Kaufmann Publishers, p. 486, ISBN 0-934613-26-5.

Journal Articles

[10] Nkambo, R. ; Gauthier, G. ; Frasson, C. 1997. Un modèle de Représentation de Connaissances relatives au contenu dans un système Tutoriel intelligent *Sciences et Techniques Educatives*. Vol. 4,N°3, pp. 299-330, 1997.

[11] Murray, T. 1998. Authoring Knowledge Based Tutors: Tools for Content, Instructional Strategy, Student Model, and Interface Design. Journal of the Learning Sciences, Vol 7, N° 1, pp. 5-64, 1998. http://www.cs.umass.edu/~tmurray/papers/JLSEon/JLS98.html

[13] Michel, A.; Yen, Si. Y. G. 1991. Analysis and Synthesis of a class of Described on Hypercubes'. IEEE Tr. Neural Network, Vol.2.

Conference proceedings

[1] Brusilovsky, P.; Schwarzi, E.; Weber, G. 1996. A Tool For Developing Hypermedia-based ITS on WWW. In Proceedings of the Third International Conference on intelligent Tutoring System (ITS'96). Berlin.

[2] Seridi-Bouchelaghem, H.; Sari, T.; Sellami, M. 2003. Dynamic Generation of Adaptive Pedagogical Documents. ITHET(2003), Marrakech, Maroc.

[3] Seridi-Bouchelaghem, H. ; Sellami, M. 2001.Un Environnement d'Enseignement à Distance Sur le Web *Sciences et Techniques Educatives (STE)*. Vol. 8. Paris, France.

[4] Sommaruga, L.; Catenazzi, N.; Giroux, S.; Moulin, C. 2000. A distance learning environment architecture", Actes de TICE 2000. Troyes. France, 18-20 octobre 2000.

[5] Crampes, M.; Ranwez, S. 2000. Ontology-Supported and Ontology-Driven Conceptual Navigation on the World Wide Web ». Proceedings of HT'00, the 11 ACM Conference on Hypertext, San Antonio, Texas.

[6] Wentland, D. ; Fortem, M. ; Bitschnau, J.F ; Simillion, F. 1997. ARIADNE, un environnement pédagogique sur Internet. Expérimentation, Evaluation et Modifications , EIAO'97, Proceedings of the 5 th EIAO of Cachan, Hermès.

[7] Tchetagni, M. P; Nkambo, R. 2002. Hierarchical Representation and Evaluation of Student in an Intelligent tutoring System. International Conference, ITS 2002. Biarritz, France and San Sebastian, spain, June 2002. LNCS 2363, Springer Verlag Edition.

[8] Kyparisia, A.; Papanikolaou, George.; Magoulas, D.; Grigoriadou, M. 2000. A connectionist Approach for supporting Personalized learning in a Web learning environment" Adaptive Hypermedia and Adaptive Web-Based Systems. International Conference, AH 2000. Trento, Italy, August 28-30.

[9] McCalla, I.; Greer, E. 1994. Granularity-Based Reasoning and Belief: Revision in Student Models. In student Models, the key to Individualized Educational Systems, J.Greer and G.McCalla (eds). New york : Springer Verlag.

[12] Gouarderes, G.; Minko, A.; Richard, L. 2000. Traçabilité de l'usager avec un simulateur. In the proceedings of the international conference on system and ingeneering and NTIC, Nîmes TIC'2000.

[14] Seridi-Bouchelaghem, H.; Sellami, M. 2002. An adaptive Distance Learning Architecture. Poster dans la 6th International Conference, ITS 2002. Biarritz, France and San Sebastian, spain, June 2002. LNCS 2363, Springer Verlag Edition.

[19] Weber, G.; Specht, M. 1997. User modelling and adaptive navigation support in WWW-Based tutoring systems. Workshop on the adaptive systems and user modeler, Chialaguna, Sardainia.

Thesis

[15] Ranwez-Chabert, S. 2000. Composition Automatique de Documents Hypermédia Adaptatifs à partir d'Ontologies et de Requêtes Intentionnelles de l'Utilisateur. Thèse de doctorat en informatique, Université de Montpellier II, décembre 2000.

References on the WEB

[16] ARIADNE: Alliance of Remote Instructional Authoring and Distributions Networks for Europe. 2001. http://ariadne.unil.ch/Metadata

[17] IMS; Standard for Learning Objects. 2002 http://www.imsproject.org/matedata

[18] LOM: Draft Standard for Learning Object Metadat. http://Itsc.ieee.org/wg12/index.html