Journal of Computer Science 1 (2): 232-243, 2005
ISSN 1549-3636

A Neural Network for Generating Adaptive Lessons

Hassina Seridi-Bouchelaghem, Toufik Sari and Mokhtar Sellami
Research Laboratory of Information Technology, Badji Mokhtar University
Annaba, B.P 12 23000 Algeria

Abstract: Traditional sequencing technology developed in the field of intelligent tutoring systems have not find an immediate place in large-scale Web-based education. This study investigates the use of computational intelligence for adaptive lesson generation in a distance learning environment over the Web. An approach for adaptive pedagogical hypermedia document generation is proposed and implemented in a prototype called KnowledgeClass. This approach is based on a specialized artificial neural network model. The system allows automatic generation of individualised courses according to the learner's goal and previous knowledge and can dynamically adapt the course according to the learner's success in acquiring knowledge. Several experiments showed the effectiveness of the proposed method.

Key words: Adaptive Courseware, Personalisation, Course Sequencing, Artificial Neural Network, Domain Ontology

INTRODUCTION

Web-based educational systems are becoming increasingly popular and are being used by heterogeneous user groups. Users of educational Web-based systems differ in their goals, background, interests and knowledge. In order to be adaptable to user's behavior, educational systems exploit increasingly more sophisticated techniques.

These techniques were introduced and evaluated since the 1950's in the area of adaptive instructions and learning psychology. Adaptive instructional methods adapt the content of instructions [1], the sequencing of learning units [2], their difficulties [3] and other parameters to students' knowledge.

Course sequencing is now the most popular technology in research-level web-based intelligent tutoring system [4]. The idea of course sequencing is to generate an individualised course for each learner by dynamically selecting the most optimal teaching operations. Several approaches were suggested [5-7] and offered limited ways in the dynamic sequencing, i.e. to change the learner's path through the material on the basis of his/her performances. Furthermore, the static structure contradicts with the traditional sequencing approach that did not accept any predefined structure and builds the sequences on the fly. Adaptive course sequencing ideas have not been used yet in the context of large-scale web-based education [8].

Some "non-symbolic" approaches in modern AI were used to expand traditional "symbolic" adaptive hypermedia in several directions [9]. There are a few promising examples of using various non-symbolic methods in adaptive hypermedia systems [10-12].

ELM-ART system [13] and its successors ELM-ART II [14], InterBook [13], NetCoach system [15], KBS hyper book system [16] are adaptive educational hypermedia systems that have been realized, updated and extended with a number of new techniques. In these systems, documents are annotated with a sets of documents having specific roles. This implies that the current state of the document space influences the metadata annotation. Whenever new documents are added, modified or deleted, the metadata annotation may become invalid because new prerequisites or outcomes can be found in the document space. Thus, it is only valid with a closed document space.

AHAM [17] is a free Web-course about hypermedia structures and systems. It uses a domain model, a user model and a teaching model which consists of some pedagogical rules to build adaptive hypermedia courses. These approaches intend to find ways for adapting pre-existent hypermedia. They do not aim to construct new links and their narrative organizations in response to user needs. Comparing with this approach, a difference in the domain description was noticed. Several models such as conceptual, navigational, adaptive, teacher and user models were defined; while this approach exploit the domain ontology for describing the concepts and their inter-relations, a neural network model, a learner model and a pedagogical component. By the use of a neural network model, a classifier of learning material as a function of concepts to be learned upon the past "experiences" is build.

The goal of this study was to present the approach behind the system GAPEDU that make possible to use the benfits of AI techniques for adaptive sequencing in web-based courses delivered.

The Educational Approach: A more progressive way of developing web-based courses, one that is growing in popularity, supports courseware reusability [18]. The coming generation of course management systems provides support for courseware reuse enabling authors to produce new courses from existing materials. One of the major goals of courseware reuse is to produce several versions of the same course from a set of learning materials and targeted to different audiences. In this study, the courseware reusability is intended automatically by the system. Adaptive and dynamic courseware generation was born and several approaches were suggested and adapted to the web [19]. The goal of this later is to generate an individualized course taking into account a specific learning goal and the initial level of the learner's knowledge with a dynamic re-planning of a course if it's necessary. Therefore the idea is to propose a system which is able to construct automatically several versions of a course from a set of learning material and targeted to several learners profiles.

In this study, the courseware reuse domain for the dynamic generation of adaptive pedagogical documents is investigated and an approach that applies as much adaptivity to an individual learner as possible with a formalization of the teacher's expertise is proposed. The presented approach is an evolution from the currently static web-based courses to the more flexible and adaptive web courseware.

The proposed approach is implemented in the GAPEDU system (Generation of Adaptive PEedagogical DocUments over the web) which is typically used in distance learning scenarios, where a learner uses the information on his/her own. Thus, teaching strategies used in this study encourages a learner to learn actively and not just to read passively the informations. So that, the style of teaching which is allowed by the GAPEDU system is in-line with the constructivist tendencies in learning environment design.

The learners are free to define their own learning goals and their own learning sequence. It's real that the generator produces a goal oriented didactic plans which are presented to the learners and gave them the freedom to visit any teaching material.

The dynamic generation of adaptive and educational documents (GAPEDU) system uses a set of pedagogical rules and an Artificial Neural Network (ANN) model to generate a plan of the course composed by a set of documents adapted to the learners' needs, knowledge and capabilities. Given a certain learning goal, i.e. that the learner wants to acquire and a learner model containing the already known concepts (initialised with a pre-test), the generator searches for a route that connects the concepts known by the learner with the chosen goal.

After that, a sequence of teaching materials related to each concept from the plan are identified, regrouped, sequenced and presented to the learner. At every point the learner can be tested on his knowledge on the current concept. If the learner is not able to achieve the concepts that are needed to proceed further towards the goal, a new plan composed of a set of further learning materials which are judged to be more adapted to his current state is constructed. The new plan intends to simplify the difficult concepts.

A learner must be able to define learning goals on his/her own. To reach such a goal, the system should be able to find relevant didactic plan, i.e. reading sequences must be generated, related to the goal. Therefore, selection algorithms must be found. They should present the most suitable didactic plan to the learner which matches his current goal, considering his/her actual knowwledge and including necessary prerequisites he/she actually needs to know. The GAPEDU system is presented and the adaptation module will be explained in more detail because it's the heart of present study.

The GAPEDU System: The GAPEDU system is a tool for modeling, sequencing adaptive pedagogical documents. The documents are distributed and located anywhere in the web. The proposed architecture constructs a didactic plan, from a set of documents pedagogically sequenced in response to the learner's needs.

Conceptual Modeling for Adaptive Course Generation: This section describes the conceptual model, which models learning goals, concepts of the domain and different kinds of teaching materials. A modeling for the GAPEDU system using the Resource Description Framework (RDF) [20] is presented.

Modeling Learning Goal: A learning goal (LG) is the ability to do something effectively. It can generally be considered as a set of knowledge, know-how and attitudes, which is activated at the accomplishment of a given task. Particularly in our pedagogical context, learning goal is an abstract concept which can be reified through attributes or properties qualifying and quantifying the concerned ability.

A learning goal is defined as a competency to be acquired by a learner through a training process using existing pedagogical materials, i.e. the related contents [21]. A learner could access the pedagogical materials by only selecting his learning goal.

Learning goals are classified according to the domain area (e.g. Computer Sciences, Languages, Mathematics) and to Bloom's learning outcomes (Knowledge, Comprehension, Application, Analysis, Synthesis and Evaluation) [22].

In order to let a learner selecting suitable (LG), the system contains a (LG) library. Each (LG) is indexed by the concepts of the domain ontology that have to be understood in order to successfully accomplish the goal.

Modeling Concept: In a learning context, it is useful to partition the domain knowledge in order to ease the

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	-

evaluation process [23]. A main issue in the development of an educational system able to support pedagogical decisions is the domain knowledge including multiple curricular view points. To this end, the domain model is structured in concepts and relations between them.

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 relations between concepts determine the nature of links between them. Two relations are defined: prerequisite and subconcept. The main concepts for a learning goal are identified and should be fully explained in HTML pages using text, images, examples, exercises, etc. 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 presents a learning goal with its associated concepts referred on the Computer Network course teached in the university of Annaba.

The concepts and their inter-relations are defined in a domain ontology. By using domain ontology, we try to adapt new techniques of the knowledge representation to educational systems [24]. The main interest is the modelling and the representation of the knowledge based on semantics. The domain ontology is used to index the course content, i.e. to connect elements of teaching material called basic units with the domain knowledge. A domain ontology on the course "Computer Network" is conceived and modeled with the standard RDF. (Fig. 1).

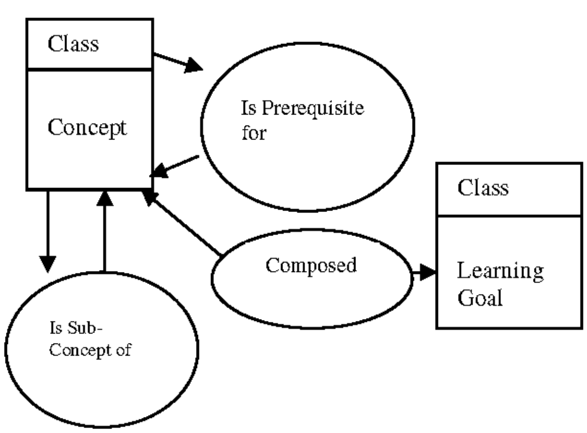


Fig. 1: Domain Ontology for a Learning Goal

RDF and RDFS [20] are used to annotate resources on the Web and provide the means by which computer can exchange and comprehend data. The resources in our system are identified by a unique identifier. The resources are stored on one server and other servers could hold further learning material. RDF triples <subject, property, value> represent specific annotations, where subject identifies the resource to describe, property specifies what property is used and value a specific value, expressed as a primitive data type or another URI.

Modeling the Basic Units: A Basic Unit (BU) is a multi-media document having intrinsically a teaching quality, i.e. which can be used within the framework of the knowledge transmission. Each course is semantically divided in several elementary fragments called Basic Units (BU).

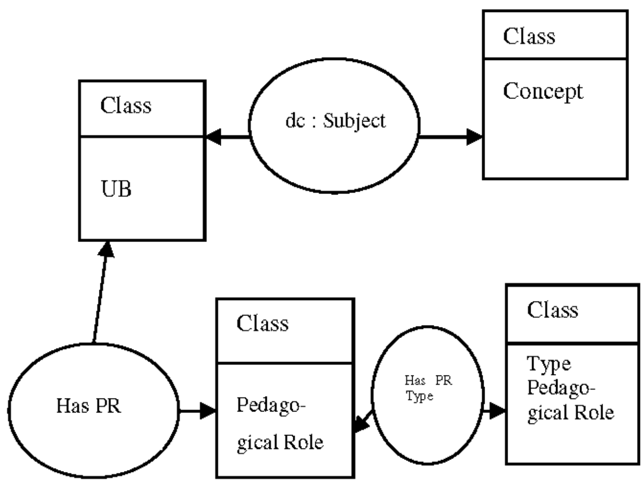
It appears convenient to share the basic units and to properly index them so that they can be easily found and re-used. The basic units can be created in our system by the author or imported from external sources and integrated as meta-data. RDF is used to annotate the BUs.

Indexing learning materials with meta data is becoming an important trend in practical web-based education. This trend has been fuelled by recent work on courseware reuse, learning pools, learning object libraries and meta data standards [18, 25, 26].

Learning Technology Standardization Committee of the IEEE was based on the work of ARIADNE [27] and of the IMS [25] to work out a proposal of LOM which is a standard defining the structure of the pedagogical object metadata [26]. The description of our basic units is based on the ARIADNE recommendations. Three types of descriptors:

* General information {Identifier, Title, Author(s), Date, Language...},
* Semantic of the (BU) {Learning goal, Main concept(s), other concepts...},
* Pedagogical attributes {Document type, Format, Level, Difficulty, Duration, Pedagogical role…}.

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



The following example shows how the UB can be annoted :

LAN topology :
Computer_Network/concepts/LAN.html
Rdf : type→ doc: UB;
dc : subject → Comuter-Network : LAN;
dc : has PR → Definition;
…]
Computer_Network : LAN [
Rdf : type →cpt : concept;
Cpt : is prerequisitefor → cpt: Networknodes;
…].

Fig. 2: The Document Space Description

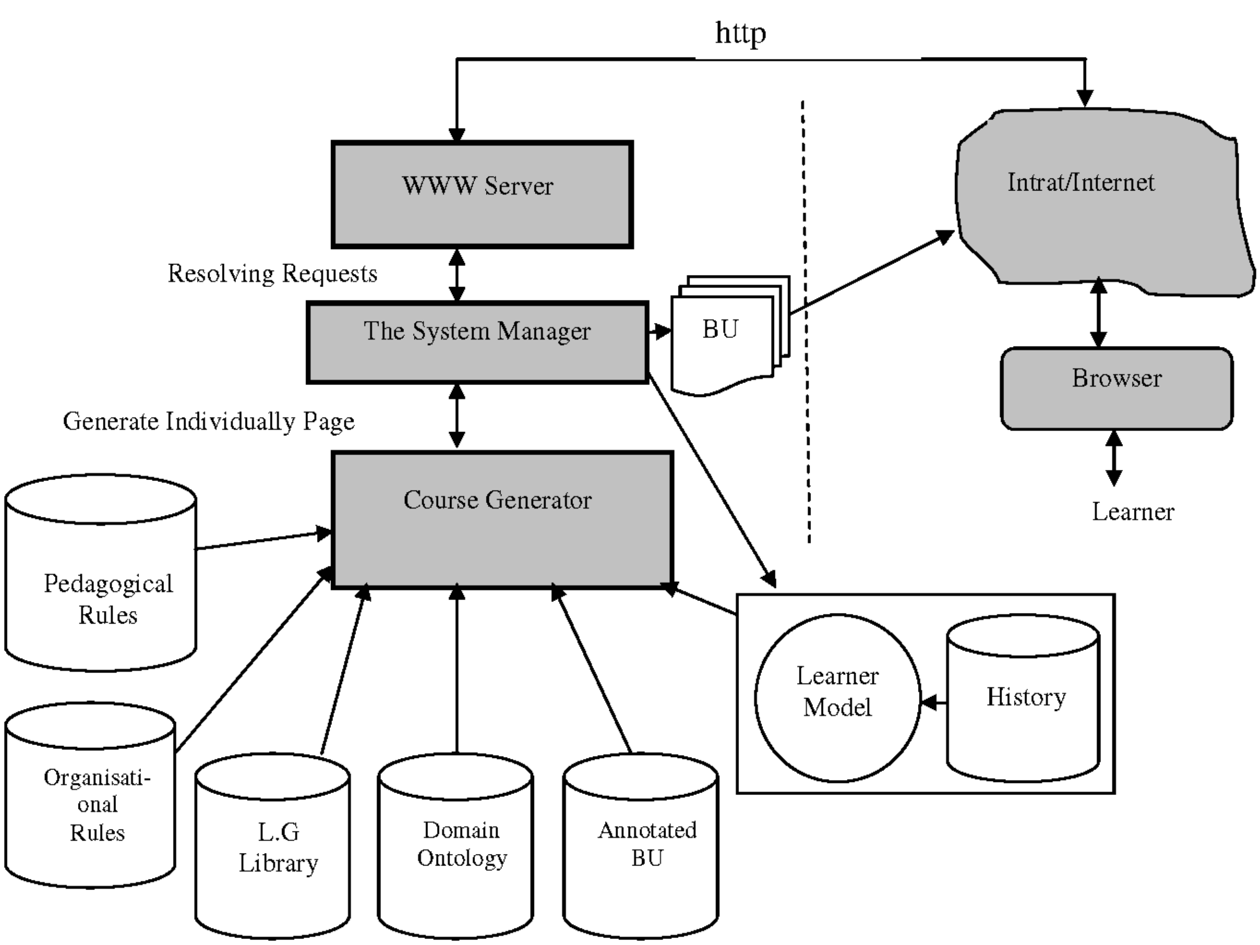


Fig. 3: A General Architecture of Adaptive Course Generator

Two categories of pedagogical roles are distinguished: basic roles and roles played by the (BU) in an activity becoming simplification or reinforcement of the non assimilated concepts (Fig. 2).
The generator is able to distinguish among several kinds of (BUs) (Basic, reinforcing, simplification). The type of a (BU) is a part of the index and the pedagogical roles allowed the course developper to specify more knowledge about the content and support more powerful algorithms.

The System Architecture: The general architecture of GAPEDU system is given in Fig. 3. The processing is held by the server which gives the access for the users to a dynamically generated interface over the Internet.
Figure 3 gives an overview over the system components and their interactions. When a learner logs on to the system, the browser connects to the Web server which functions as the bridge between the client's browser and the system. The requests from the user and responses from the system pass through it.
The Web server can fulfill some requests by itself, others are passed to the appropriate components. The web server contacts the session manager that sends the questionnaire via the web server to the browser. The information provided via the questionnaire is used to initialize and create a learner model.
When the learner has chosen a (LG), the session manager sends this request to the course generator. This later is responsible for choosing and arranging the content to be learned. The course generator contacts the domain ontology, in order to identify which concepts are required for understanding the goal, cheks the learner model in order to find out about the learner's prior knowledge and preferences and uses pedagogical rules and an (ANN) to select and arrange the content that is suitable for the learner. The functionality of the adaptation module will be described in detail in the following section.
The sequenced didactic plan is sent to the session manager. The learner's actions are analyzed by evaluators that calculate and update of the learner model. When the learner logs out, her/his newer learner model is stored.

Dynamic Generation of Adaptive Courses: The course generator decides which concepts will be taught and plans dynamically how to present the contents related to the current concept in an optimal for the learner way, i.e., what types of (BUs) will be selected and how to sequence them?
In order to implement an intelligent educational and adaptive system, several components are needed.

Pedagogical Component: This component contains the knowledge concerning the choosen concepts to be taught and the sequencing of the (BUs). The pedagogical components contains two parts: pedagogical and organisational rules.

Pedagogical Rules: Teachers follow different strategies to instruct students about some concepts [28]. The generator employs the pedagogical rules in order to select the appropriate concepts to be presented to the learner. An example of such rules are given below:

R1: The concept is selected if all its prerequisite concepts are assimilated.
R2: The concept is selected if the learner level for the concept is "Medium" and will be presented with further (BU).
R3: The concept is selected if the learner level for the concept is "Low" and will be presented with further (BU).
R4: The prerequisite concepts of assimilated concepts are not selected.
R5: A concept is selected if all its sub-concepts are assimilated.
R6: The prerequisite concept is selected if the learner level is "Low".
R7: The prerequisite concept is selected if the learner level is "Medium".

Organisational Rules: Several organisational rules were built. Some of them are general but can always be applied, others are more specific to some learning styles. General rules are always valid and can be 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 (BU) organization, according to the constraints imposed by the concept. For example, "when a concept is composed of sub-concepts, their corresponding BUs will be ranked before those of the main concept".
Several organizational rules, specific to some a learning styles are applied. Logical and intuitive learning styles refer to a preferred organization of the (BU). 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 intuitive 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 these two learning styles.
Some rules concern the chosen learning style. A rule used for the logical style mentions that if an example and an explanation refer to the same concept, the presentation of the explanation should precede the presentation of the example.

Learner Model: The aim of the learner model is to guide the tutor in taking the pedagogical decisions better adapted to a learner [29]. In this model, the first question to be answered is what is to be represented? Overlay models [30] and Buggy models [30] are knowledge representation approaches which determine how to express the learner's knowledge. In Overlay models, the student knowledge is considered as a subset of the domain knowledge which should be incremented. However, Buggy models enable further modeling of faulty information in the system knowledge. The main concern of this study is to generate lessons and try to help learner to see unassimilated concepts by giving a course specific to its current situation. So that, the Overlay model is more suited and set up by the evaluation module.

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 [31]:

* **Static Part:** This information is static and rarely changes 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 goals 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 competences for their concepts.

Every action of the learner is analysed and saved in his learner model. This later indicates at every step, the learner's knowledge level. 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 updating, 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.

The knowledge of a learner is modeled as a knowledge vector (LV). Each component of the vector is a conditional probability, describing the system's estimation that a learner L has knowledge about a concept C, on the basis of all observations E the system has about L.

Definition 1 (Learner vector): LV.

$$LV(L)= P(C1/E), P(C2/E), \ldots,P(Cn/E)$$

Where, C1,C2,...,Cn are the concepts of the application domain and E denotes the evidence the system monitors about L's grades of the course.

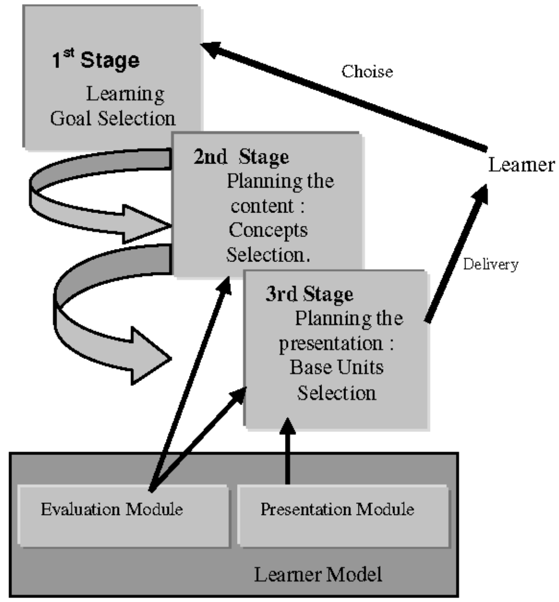


Fig. 4: The Interaction Levels

Observations about the learner's work with the course are stored for each concept. Each observation expresses the grade of knowledge the learner has on a concept. Three grades are used and a learner can have:

* "High knowledge" on a concept; rating H,
* "Medium knowledge" on a concept; rating M,
* "Low knowledge" on a concept; rating L.

Thus, the Cs are concepts describing the application domain of the course and are random variables with the three discrete values H, M, L coding knowledge grades. The knowledge vector gives the learner's current knowledge.

Using probabilities for giving estimation about a learner's knowledge is a very intuitive approach which have been used in [16] and relatively easy to estimate [32].

The Course Generation: The pedagogical decisions are made at the end of the knowledge evaluation and 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 a didactic plan for delivering to the learner according to a predefined teaching strategy.

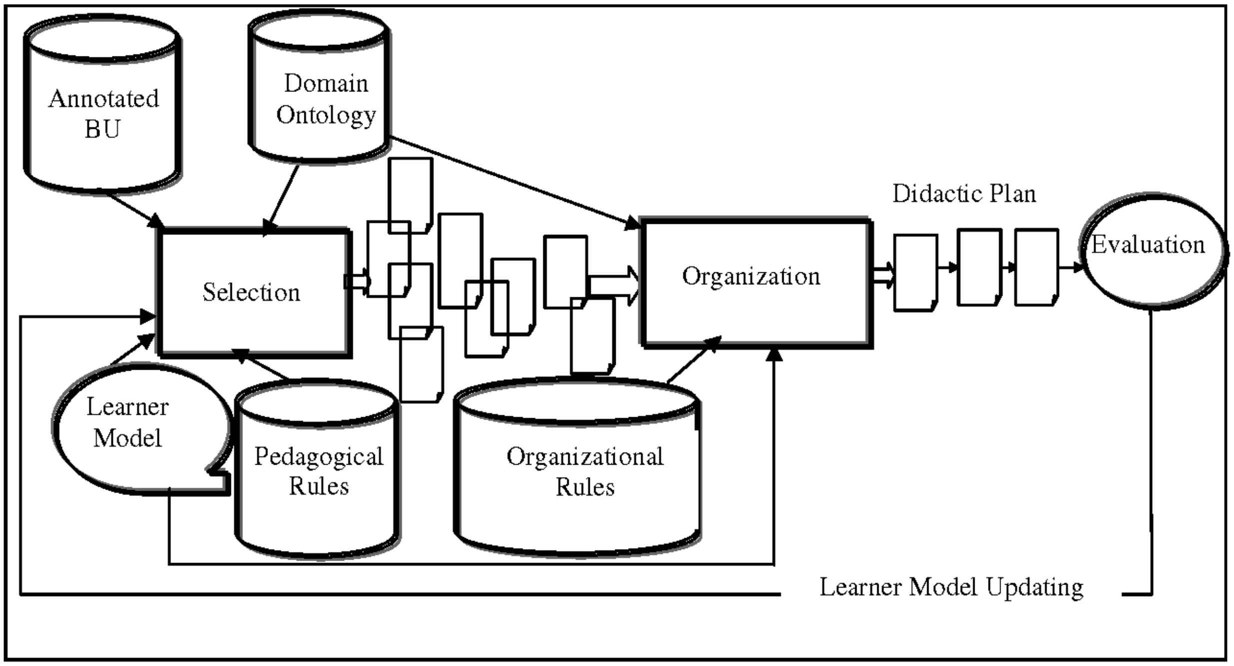


Fig. 5: Functional Model of the Developed System

The presented approach supports the following adaptation mode: the learner selects a learning goal at the first stage (Fig. 4). Each learning goal relates to a subset of concepts of the knowledge field. 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 test sessions will evaluate the concept's knowledge associated to the selected learning goal.

During the interaction process System-Learner, the evaluation module keeps track on the learner performances and estimates his/her level of comprehension. The results of the evaluation procedure influence the course generation process.

The 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 strategy 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: Learning. 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 learner level 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.

The Course Generator (CG) generates the course, carries out the interaction with the learner and maintains the learner model. The pedagogical component and an ANN model are needed to decide dynamically how to sequence the presentation of (BUs) for the concepts of the plan.

According to the pedagogical rules, the generator selects the suitable concepts to be taught, then the generator consults the (ANN) model to classify and selects the appropriate (BUs) on an appropriate type of media, taking into account the learner's level and preferences.

The selected (BUs) are presented after their sequencement. When the learner occurred an (BU) involving a test or an exercise, the learner model knowledge is updated according to the test item's conditional probabilities. If the learner fails to acquire the concept (insufficient knowledge probability of the concept in the (LM), the (CG) is able to find a new content plan (DP).

The presented system provides one type of re-planning, it tries to find an alternative way to present unassimilated concepts.

A set of pedagogical rules manages the selection of content according to the learner's profile. The generator first decides which concepts will be taught, i.e. dynamically creates a content of the course. The representation of the teacher's expertise with use of a neural network model allow the system to plan dynamically how to present the contents related to the current concepts in a way suited to the learner. Finally, a set of organisational rules are used by the generator to assign an order between the regrouped (BUs) (Fig. 5).

The Neural Network Approach for Basic Unit Selection: Two Multi-layer Perceptrons (MLP), with one hidden layer are constructed to process the selection task and to make the decision upon learner's understanding.

Artificial Neural Net (ANN) models have particular properties such as ability to adapt, to learn or to cluster data [33]. (ANN) was intensively employed in multiple fields related to the classification tasks such as pattern and speech recognition, non-linear systems identification and control. They are able to discover the hidden relations between data.

The problem of adaptive course generation upon learners' profiles can be viewed as a classification problem, since the purpose of this process is to find the appropriate set of basic units associated to the set of parameters computed from learner behaviour.

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" (Fig. 6). (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 the 'BackPropagation' and 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.

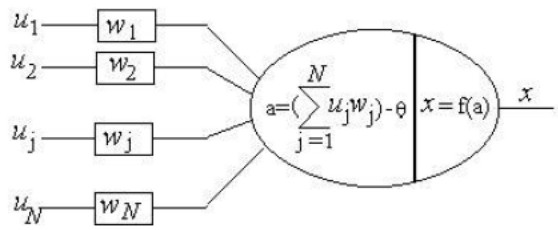


Fig. 6: Artificial Neuron

Such as:

$$f(a)=\frac{1}{1+e^{-a}} \quad (1)$$

ui: The response of the neuron i from the previous layer
wi: The link weight
θ: The threshold of the neuron activation
f: The activation function
x: The current neuron response
N: The number of neurons of the previous layer

The first neural network is used to select the appropriate (BU) for the learner in the first stage of learning. The input layer represents the concepts of the course, one neuron per concept. The input vector (VI) is a set of values belonging to the set {0, 0.5, 1} where, the values $VI_i=1$ indicates that the corresponding concept (c_i) is important to the learner and the values $VI_i = 0$ 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 BUs with Basic Roles (BR) (Fig. 7).

The second neural network intervenes when the learner do not succeed the post-test of the concepts. This later generates a vector of marks related to the concepts and called reinforcing vector (VR). Three measures are used: 0 for low, 0.5 for medium and 1 for high learner levels of the concept understanding. The (VR) is used as input values for the second neural network, the output layer for selecting the BUs having reinforcing roles.

Note that, when the learner's knowledge with respect to a concept is characterized as non understood, 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 BUs, 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.

For the first investigation, in the two neural networks, links between neurons are initialized by random values. The back propagation algorithm is used for training and the tan-sigmoid as activation function.

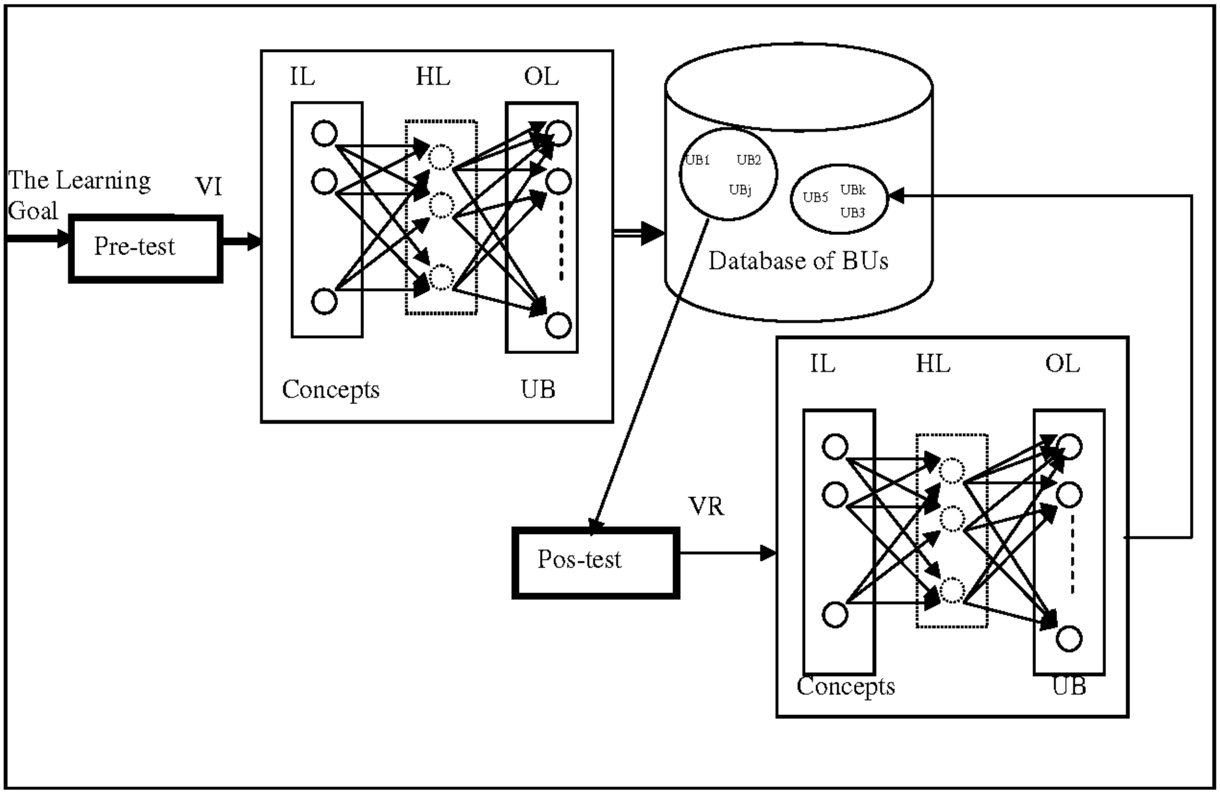


Fig. 7: The Developed Connectionist Based Architecture
Where, IL: Input Layer, HL: Hidden Layer and OL: Output Layer

The Course Organization: In the organization phase, the system assigns an order between basic units to allow the system building a didactic plan. Indeed, the system is able to impose precedence constraints between the BU according to their pedagogical roles.

The first set of constraints concerning the order of the BU is general and is expressed by the domain ontology. For example, the fact that a concept (C1) precedes another concept (C2), imposes that a BU-example of (C1) precedes a BU-example of (C2). The rules qualified as general and presented in the previous section are used.

The second set of constraints is specific to a type of learning style. For the logical learning style, declared by the learner, a BU-explanation of a given concept must precede the (BU) that refers to the *example* of the same concept, for example. When precedence constraints are assigned to all selected (BUs), the system is then able to build a didactic plan.

The final structure of the course is then dictated by the learner learning style and the domain model. To every declared learning style is associated a set of organizational rules that describe the document structure.

Dynamic Course Re-planning: During the presentation of the didactic plan, if the learner answers the test basic units correctly, he/she progresses along the course and no changes to the course are necessary.

However, if the learner fails to acquire knowledge of a concept, a re-planning of the course follows. Re-planning takes place first at the presentation level, i.e. an alternative sequence of (BUs) of the concepts is presented to the learner. If the learner fails again, the (CG) generates a new sequence of concepts leading to the goal, starting from the current state of learner knowledge as recorded in the learner model.

Implementation: This new approach supports the learning in an open corpus educational courseware that are currently investigated in the university of Badji Mokhtar Annaba. The mechanism behind this approach and its implementation in a system called "KnowledgeClass" in the "Computer Network" course is introduced. Some results of several classroom studies are outlined.

The system is implemented entirely in Java. A selvelet residing in the web server represents the whole system. The learner browses the course with an HTML browser capable of handling frames, which all necessary processing is done on the server side. The learner navigates the course by activating links of the prented didactic plan (Fig. 8).

Fig. 8a: Evaluation Interface

Fig. 8b: Author Interface

Experiments: The presented approach was tested in the *'Computer Networks: LAN Topology'* course in which 15 concepts are identified (Types of connections, Star topology, Bus topology, Ring topology…). 82 BUs are designed from which 50 with basic pedagogical roles (Introduction, Example, Exercise, Explanation,…) and 32 with reinforcing roles (Simplification, Comparison, Reformulating, Discussion...).

The first neural network is composed of 15 neurons in the input layer, 35 neurons in the hidden layer and 50 neurons in the output layer. It was trained on 60 learner profiles and tested on 30 other unknown profiles. The obtained results compared with human generated (BU) are very encouraging. Thirty-three among 90 learners didn't succeed the post-test, so the results of 21 of them were used to train the second neural network and the other for the test.

The second neural network is constructed with 15 neurons in the input layer, 12 neurons in the hidden layer and 32 neurons in the output layer.

For validating the NN we began by performing some experiments in order to collect all necessary data.

All the experiments were performed on an IBM-PC with 3.2 GHz.

N.B. The number of hidden neurons was determined heuristically.

Collecting Data: The students' data used in this study have been obtained from two experiments performed in our laboratory.

Ninety undergraduate students' profiles and their respective didactic plans were computed manually by three different teachers of the course "Computer Networks".

Experiment One: "Isolated Tutoring" (IT Experiment): In the first experiment, each teacher was asked to provide the estimation of several concepts to be learned and their associated Basic Units concerning thirty different students. The generated courses were presented to each set of thirty learners in separate classrooms. Teachers evaluated their associated students and provided the test results (the concepts to re-teach or to re-enforce) and the associated Basic Units which were in the most cases different from those of the first stage.

Experiment Two: "Collaborative Tutoring" (CT Experiment) : In this second experience, the three teachers were asked to work together to do the same task as described in the IT experiment. They reported their results for other fifty two students.

First ANN Evaluation and Testing

First experiment data (IT-Data): Firstly, we evaluated the (ANN) on the data provided by each teacher independently. Twenty learners' data were used for the first (ANN) training and the remaining ten for testing. In the following sections T_i designates the set of the ten learners' data collected by the teacher i and NN_i the NN trained with the corresponding twenty learners' data. Table 2 shows the obtained results on IT data.

Table 2 : Experimental Results on IT-Data

Test data Neural networks	T_1	T_2	T_3
NN_1	96%	85%	86%
NN_2	88%	97%	89%
NN_3	85%	88%	95%

The different scores mean that the (NNs) were able to generate the expected (BUs).

It would be noted that the training was stopped when each ANN exceeded the 95% of good results. In other words when they were able to select 95% of the Basic Units generated by the teachers. The reason to choose this threshold is that performing of 100% of approximation took more time and the NN_3 had never exceeded 95%.

The obtained results are very promising. Each NN was able to approximate its associated data and generalized better on other unknown data.

Second Expriment Data (CT-Data): This experiment used the results of the collaborative work of the three teachers. The data related to forty students was used for training and twelve for testing the NN. We obtained a rate of approximately 98% of good generation. The NN was also tested on all the data collected in the first experiment (Table 3).

Table 3: Experimental Results on CT-Data

Data 1	Data 2	Data 3
95%	96%	95.5%

The NN trained with collaborative data is more reliable than the others. That can be explained by the fact that the collaborative work is more precise and in general involves individual works.

Second ANN Evaluation: The second NN was evaluated on the data from student who didn't succeed the first stage.

IT-Data: The data from 33 students' profiles is used, 9 from the teacher1, 11 from the teacher2 and 13 from the teacher3. The NNs were trained, as explained in 7.2a, on 2/3 of the data provided by each teacher independently and tested on the remaining data. Table 4 shows the obtained results.

Table 4: Results on IT-Data

Test data Neural networks	T_1	T_2	T_3
NN_1	96.5%	85%	81%
NN_2	82%	96%	83%
NN_3	81%	81%	95%

The different NNs approximate well on their respective data but have poor generalisation on other data sets. This problem is very known in ANN literature, it is due to not enough training. In our case, the training sets are very small; the NN cannot reach their global minima. But we consider the results as good.

CT-Data: 15 students didn't succeed the post-test. Their respective data from the second stage was used to train the NN. This later was able to predict 97.5% of the Basic Units. The same NN was also tested on the IT-Data (Table 5).

Table 5: Results on CT-Data

Data 1	Data 2	Data 3
98%	96%	95%

CONCLUSION

The constitution of adaptive training plans on Internet is an important field of research in the distance teaching. This study has described some important parts of an adaptive learning environment, how they are designed and interact. 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 use of the RDF standard for basic units annotation and the modeling by the domain ontology enhance the reuse and the exchange of these resources, giving distributed adaptive hypermedia systems access to the full content of the Web.

The using of artificiel neural networks for generating adaptive lessons demonstrate the usefulness of the techniques based on some training which is considered the main drawback of classical approaches. The problem of dynamic document composition has been rethought as a classification problem since selecting document components upon predefined constraints is well adequate for neural networks. MLP for Multi-Layer Perceptrons are known as universal classifiers, they can approximate any function. The results of our preliminary study show that our approach is promising for building dynamic adaptive learning.

In this current version of our system the concepts related to a learning goal are selected using some pedagogical rules. These later are not reliable and do not resolve completely the problem of selecting the effective concepts. So, another neural network is under consideration to handle this task.

REFERENCES

1. Tennyson, R.D. and W. Rothen, 1977. Pre-task and on-task adaptive design strategies for selecting number of instances in concept acquisition. J. Educational Psychol., pp: 69.
2. Weber, G. and M. Specht, 1997. User modelling and adaptive navigation support in WWW-based tutoring systems. In Workshop on the Adaptive Systems and User Modeler, Chialaguna, Sardainia.
3. Tennyson, R.D. and D.L. Christensen, 1988. MAIS: An Intelligent Learning System. In D.H. Jonassen, Ed. Instructional Designs for Microcomputer Courseware. Hillsdale: N.J. Erlbaum.
4. Brusilovsky, P., Adaptive and intelligent technologies for web-based education. http://www2.sis.pitt.edu/~peterb/papers/KI-review.html.
5. Kettel, L., J. Thomson and J. Greer, 2000. Generating individualized hypermedia applications. In: Proc. of Workshop on Adaptive and intelligent Web-based Education Systems at 5th International Conference on intelligent Tutoring Systems (ITS'2000), Montreal, Canada.
6. Masthoff, J., 2002. Automatic generation of a navigation structure for adaptive web-based instruction. In: Proc. of Workshop on Adative Systems for Web-based Education at the 2nd International Conference on Adaptive Hypermedia and Adaptive Web-based Systems (AH'2002). Malaga.

7. Melis, E., E. Andres, J. Budenberder, A. Frishaud, G. Goguadse, P. Libbrecht, M. Pollet and C. Ullrich, 2001. Active math: A generic and adaptive web-based learning environment. Intl. J. Artificial Intelligence in Education, 12: 4.
8. Brusilovsky, P. and J. Vassileva, 2003. Course sequencing techniques for large-scale web-based education. Intl. J. Cont. Engineering Education and Lifelong Learning, 13: 1/2.
9. Brusilovsky, P., 2001. Adaptive Hypermedia, User Modeling and User Adapted Interaction 11. Kluwer Academic Publishers. Printed in the Netherlands.
10. Kayama, M. and T. Okamoto, 1998. A mechanism for knowledge-navigation in hyperspace with neural networks to support exploring activities. Proceedings of Workshop 'Current Trends and Applications of Artificial Intelligence in Education' at the 4th World Congress on Expert Systems, Mexico city, Mexico.
11. Kayama, M. and T. Okamoto, 1999. Hy-SOM: The semantic map framework applied on an example case of navigation. In: Proc. of ICEE'99, 7th International Conference on Computers in Education. Frontiers ub Artificial Intelligence and Applications, Amsterdam: IOS Press, 2: 252-259.
12. Seridi-Bouchelaghem, H., T. Sari and M. Sellami, 2004. A connexionist approach for Adaptive lesson generation. In the Kluwer Proocedings of the international Conference, WCC (2004), Toulouse, France, August.
13. Brusilovsky, P., E. Schwarz and G. Weber, 1996. A tool for developing adaptive electronic textbooks on WWW. In the Prooceedings of WebNet'96.
14. Weber, G. and M. Specht, 1997. User modeling and adaptive navigation support in WWW-based tutoring systems. In: Proc. of the Sixth International Conference on User Modeling, UM 97, Sardinia, Italy.
15. Weber, G., H.C. Kuhl and S. Weibelzahl, 2001. Developing adaptive internet based courses with the authoring system Netcoach. In: Proc. of the Third Workshop on Adaptive Hypermedia, AH 2001.
16. Henze, N. and W. Nejdl, 2001. Adaptation in open corpus hypermedia. Intl. J. Artificiel Intelligence in Education.
17. De Bra, P., G.L. Houben and H. Wu, 1999. AHAM: A dexter-based reference model for adaptive hypermedia. In UK Conference on Hypertext.
18. Verhoeven, B., K. Cardinael, R. Van Durn, E. Duval and H. Olivié, 2001. Experiences with the ARIADNE pedagogical document repository. In: Proc. of ED-MEDIA'2001-World Conference on Educational Multimedia, Hypermedia and Telecommunication, Tampere, Finland AACE.
19. Seridi-Bouchelaghem, H. and M. Sellami, 2002. An adaptive distance learning architecture. In the 6th International Conference on Intelligent Tutoring System, ITS 2002. Biarritz, France and San Sebastian, Spain. LNCS 2363, Springer Verlag Edition.
20. W3C. Working Group, 2002. W3C, resource description framework. http;//wwww.w3.org/RDF/.
21. Seridi-Bouchelaghem, H., M. Sellami, 2001. Un Environnement d'Enseignement à Distance Sur le Web. Sciences et Techniques Educatives (STE), Vol. 8. Paris, France.
22. Bloom's Taxonomy, 1999. http://www.wested.org/tie/dlrn/Blooms.html
23. Murray, T., 1998. Authoring knowledge based tutors: Tools for content, instructional strategy, student model and interface design. J. Learning Sci., 7:1. http://www.cs.umass.edu/~tmurray/papers/JLSEon/JLS98.html
24. Crampes, M. and S. Ranwez, 2000. Ontology-supported and ontology-driven conceptual navigation on the world wide web. Proceedings of HT'00, 11th ACM Conference on Hypertext, San Antonio, Texas.
25. IMS; Standard for Learning Objects. http://www.imsproject.org/matedata
26. LOM: Draft Standard for Learning Object Metadata. http://ltsc.ieee.org/wg12/index.html
27. ARIADNE., 2001. Alliance of Remote Instructional Authoring and Distributions Networks for Europe. http://ariadne.unil.ch/Metadata
28. Vassileva, J., 1994. A practical architecture for user modeling. In a hypermedia based information system. In the 4th International Conference on User Modeling. Hyannis: MITRE.
29. Nkambo, R., G. Gauthier and C. Frasson, 1997. Un modèle de Représentation de Connaissances relatives au contenu dans un système Tutoriel intelligent. Sciences et Techniques Educatives, 4: 3.
30. McCalla, I. and E. Greer, 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.
31. Tchetagni, M.P. and R. Nkambo, 2002. Hierarchical representation and evaluation of student in an intelligent tutoring system. In the International Conference, ITS 2002. Biarritz, France and San Sebastian, Spain. LNCS 2363, Springer Verlag Edition.
32. Pearl, J., 1990. Bayesian Decision Methods. In Shafer, G. and J. Pearl (Eds.). Readings in Uncertain Reasoning. Morgan Kaufmann.
33. Michel, A. and Si.Y.G. Yen, 1991. Analysis and synthesis of a class of described on hypercubes. In IEEE Transactions. Neural Network, Vol. 2.