Adaptive Instructional Planning Using Neural Networks in Intelligent Learning Systems

Hassina Seridi, Toufik Sari, and Mokhtar Sellami
Laboratory of Research on Informatics (LRI), Badji Mokhtar University, Algeria

Abstract: *This paper investigates the use of computational intelligence for adaptive lesson sequencing in a distance-learning environment. A connectionist method for adaptive pedagogical hypermedia document generation is proposed and implemented in a prototype called AppSys. The proposed methodology based on the use of ontologies and learning object metadata. The generated didactic plan is adapted to the learner's goals, abilities and preferences. Several experiments have shown the effectiveness of the proposed method.*

Keywords: *Intelligent learning environment, web-based course, adaptive and automatic course sequencing, learner model, domain ontology, neural networks.*

Received July 132004; accepted April 18, 2005

1. Introduction

Distance learning through the Web offers an instructional delivery system that connects learners with educational resources. Its main features are the separation of instructor from learner in space and time, the use of educational media/technology unifying instructor to learner and transmitting the course content, and to realign the teaching-learning environment from *tutor-centered* to *learner-centered.* The design of a Web-based learning environment includes informed decisions about what comprises the educational content and how it is to be sequenced and synthesized, taught and learned. This process is essential in distance education, where the instructor and learners typically have minimal face-to-face contact.

Adaptive hypermedia instruction is relatively a newer research direction. The limitation of traditional "*static*" hypermedia applications is that they provide the same page content and the same set of links to all users [4]. Some "*non-symbolic*" approaches in modern AI were used to expand traditional "*symbolic*" adaptive hypermedia in several directions [9, 17]. There are few promising examples of using various non-symbolic methods in adaptive hypermedia systems [11, 12, 13, 24]. AHAM [6] is a free Web-course about hypermedia structures and systems. It uses a domain model, a user model and a teaching model that consists of some pedagogical rules to build adaptive hypermedia courses. These approaches attempted to find ways for adapting pre-existent hypermedia and did not target the construction of new links with narrative organizations responding to the user needs.

Adaptive Learning Environments (ALE) integrates Adaptive hypermedia instruction systems and Intelligent Tutoring Systems (ITS). This is in effect a combination of two opposed approaches to computer assisted learning systems: The more directive tutor centered style of traditional Artificial Intelligence (AI) based systems and the flexible learner-centered browsing approach of an Educational Hypermedia system.

The notion of *adaptation* is defined as the concept of making adjustments in the educational environment to accommodate diversity in the learner needs and abilities, in order to maintain the appropriate context for interaction [16, 28].

Intelligent Learning Environments (ILE) seeks to provide adaptive navigation and adaptive sequencing. Adaptive navigation tends to present the content of an on-line course in optimized order, where the optimization criteria takes into consideration the learner's background and performance, whereas adaptive sequencing is defined as the process for selection of learning objects from a digital repository and sequencing them in a way which is appropriate for the targeted learning community or individuals [14].

In most intelligent learning systems that incorporate course-sequencing techniques, the pedagogical module is responsible for setting the principles of content selection and instructional planning. The content selection is based on a set of teaching rules according to the cognitive style or learning preferences of the learners [20]. In spite of the fact that most of these rules are generic (i. e., domain independent), there are no well-defined and commonly accepted rules on how the content should be selected and how they should be sequenced to make "instructional sense" [14, 20]. Moreover, in order to design highly adaptive learning systems a huge set of rules is required, since dependencies between educational characteristics of learning objects and learners are rather complex.

In recent years, a number of research groups have put many efforts in developing suitable methodologies, approaches, tools, and practical systems to support Web-based education. However, there has been comparatively little research in applying the principles and achievements in computational intelligence to Web-based educational applications. To complement such a situation, this paper proposes the use of a method from computational intelligence, such as artificial neural networks to incorporate tutor's viewpoints into the educational environment and to perform lesson adaptation. To this end, a neural approach is proposed in order to adapt the content accessed by a particular learner to his/her current knowledge level, goals and characteristics.

The paper is organized as follows. In section 2, we review the approaches to implement adaptivity in educational hypermedia. Section 3 explains the proposed methodology for automatic course sequencing. Section 4 describes the domain knowledge model. Section 5 presents the leaner model components. Section 6 proposes an approach to instructional design that exploits the neural network for the instructional planning. Section 7 presents the implementation. The paper ends in section 8 with experiments and discussion.

2. Approaches to Implement Adaptivity in Educational Intelligent Learning Systems

In intelligent learning systems, the structure of the knowledge domain is usually represented as a semantic network of domain concepts, or generally elementary pieces of knowledge for the given domain, related with different kinds of links (see [3] for a review on these systems). Learner's knowledge is often represented by an *overlay model* based on the structural model of the subject matter. The idea of the overlay model is to represent an individual learner's knowledge of the subject as an "overlay" of the domain knowledge. The pedagogical knowledge incorporated in the system affects its adaptivity and effectiveness. This type of knowledge supports didactic decisions and implements the tutoring strategy of the system, which is responsible for deciding how to sequence knowledge in order to achieve instructional goals and for selecting a particular activity relevant in the current context.

To this end, the selection, sequencing, and synthesis of the educational material of a Web-based course must be based on understanding the context of learning, the nature of the content, or task that is to be taught, the instructional objectives, the learners' characteristics, preferences and educational needs, the processes of learning and the constraints of the medium.

In literature, two main approaches in automatic course sequencing have been identified [5]:

- *Adaptive Courseware Generation*, where the main idea is to generate a course suited to the needs of the learners. Instead of generating a course incrementally, as in a traditional sequencing context, the entire course is adaptively generated before presenting it to the learner.
- *Dynamic Courseware Generation*, where as in the previous approach, the goal of dynamic courseware generation is to generate an individualized course taking into account specific learning goals, as well as, the initial level of the student's knowledge. The difference here is that the system with dynamic generation observes and adapts to students' progress during his interaction with the generated course. If the student's performance does not meet the expectations, the course is dynamically re-planned.

The benefit of this approach is that it applies as much adaptivity to an individual student as possible. Through dynamic regeneration, each student is able to get a highly personalized course for his/her needs.

In this paper, we address the problem of learning object sequencing in intelligent learning systems by proposing a new methodology that instead of forcing an instructional designer to produce a decision model that represent the way the designer decides, based on the designer's reaction over a small-scale sequencing problem. In the next section, we will explain the common methodology for capturing expert decisions used in the automatic course sequencing.

3. Methodology for Automatic Course Sequencing

The pedagogical decisions come out as a result 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. This approach aims to organize and structuring the content, supplement the theory with a variety of practical tasks and activities, and finally, to provide learners with self-assessments and assessments to test their knowledge. The greatest challenge in the presented work is to build an environment in which the learners are motivated to assess their personal knowledge goals and objectives, and to become active participants in the overall learning process.

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 chosen 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.

The presented approach supports the following adaptation mode: The learner selects a learning goal at the first stage. 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 educational material, 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 educational materials. 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 educational materials. To each educational material, a pedagogical role is assigned. Two classes of roles are distinguished: Basic roles (BR) and reinforcing roles (RR). In other words, educational materials with (BR) are selected for the first stage of learning and are generated for learners having medium/high level. Educational materials 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 educational material.

4. Modeling the Domain Knowledge

A key point in producing a system that meets the individual educational needs and objectives of each particular learner is to structure the domain knowledge in such a way that it will be possible to do adaptations.

The structure of the domain knowledge is based on symbolic methods and is usually represented as a semantic network of domain concepts, or generally elementary pieces of knowledge for the given domain, related with different kinds of links. Alternatively, the use of a concept level hierarchy, or a graph of concepts has been suggested.

4.1. Knowledge Structuring

Ontology is a shared conceptual representation of domain knowledge that provides a common understanding of a domain. Ontologism was originally developed in Artificial Intelligence to facilitate knowledge sharing and reuse [8]. Later, ontologies have been used for intelligent knowledge retrieval in the WWW as an instrument to model semantic information (metadata) used to annotate web documents (see e. g., [7]). In our approach, ontologies are used to provide maximum flexibility in the representation of domain knowledge. Moreover, they are an essential element in achieving the separation of presentation and contents. The main component of the knowledge-based approach to developing adaptive educational system is a structured domain model that is composed of learning goals and concepts.

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, LG is an abstract concept that can be reified through attributes or properties qualifying and quantifying the concerned ability. An LG is defined as a competency to be acquired by a learner through a training process using existing pedagogical materials, i. e., the related contents [23]. A learner could access the pedagogical materials by only selecting his LG.

LGs 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) [2].

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 *sub concept*. 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 sub concepts. Table 1, presents a learning goal with its associated concepts referred on the computer network course taught in the University of Annaba.

The concepts and their inter-relations are defined in domain ontology. By using domain ontology, we try to adapt new techniques of the knowledge representation to educational systems [4]. The main interest is the modeling 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. Domain ontology on the course “computer network” is conceived and modeled with the standard RDF [27].

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 subconcepts.

Main Concept	Prerequisite Concept	Sub-concept
LAN Topology	Network Nodes, Types of Connections, LAN	-
Types of Connections	-	Bus, Star, Ring Topology
Star Topology	Point to Point Connection, Polling	-
Bus Topology	Multidrop Connections	-
Ring Topology	Error Rate	-

4.2. Basic Unit Structuring

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 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. PDF 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 [1, 10, 26].The description of our basic units is based on the LOOM recommendations [15].

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... 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.

The generator is able to distinguish among several kinds ofBus (Basic, reinforcing). The type of a BU is a part of the index and the pedagogical roles allowed the course developer to specify more knowledge about the content and support algorithms that are more powerful.

5. Learner Model Components

The aim of the learner model is to guide the tutor in taking the pedagogical decisions better adapted to a learner [21]. In this model, the first question to be answered is what is to be represented? Overlay models [18] and Buggy models [18] 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 that should be incremented. 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. 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:

- *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 analyzed 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, by a 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 based on the existing information in the model.

5.1. Evaluating Learner’s Knowledge

A Performance Estimation (PE) component estimates the learner’s performance level and updates the learner model (see [22] for a detail on this component). A qualitative model classifies learner’s knowledge level to one of the three levels of proficiency. Three levels are used and a learner can have high, medium or low knowledge on a concept.

Following the termination of the pre-testing, the PE estimates the prior knowledge level of the learner on each concept. The initial knowledge level of the learners was assumed as {Medium}. Additionally, we estimated the learners’ knowledge level based on the percentage of correct answers, according to heuristic rules i. e., if the percentage of correct answers is between 0-49% or between 50-75% or over 75% then the level is estimated as {Low}, {Medium} or {High} correspondingly.

6. Instructional Planning

The Course Generator (CG) generates the course, carries out the interaction with the learner and maintains the learner model. The pedagogical component and an Artificial Neural Net (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, and 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.

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 as shown in Figure 1.

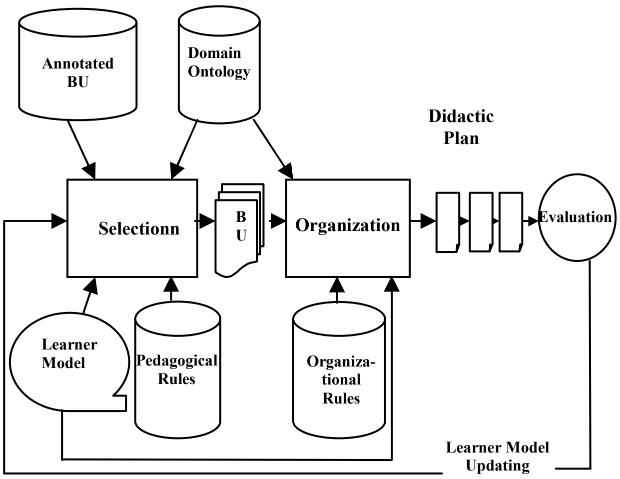

Figure 1. Functional model of the developed system.

A set of pedagogical rules manages the selection of content according to the learner's profile. An example of such rules is illustrated 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".

6.1. Neural Network Approach for Adaptive 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.

ANN models have particular properties such as ability to adapt, to learn or to cluster data [19]. 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*" as shown in Figure 2. 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*'.

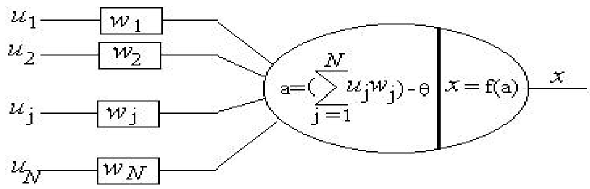

Figure 2. Artificial neuron.

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.

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.

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) as shown in Figure 3.

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.

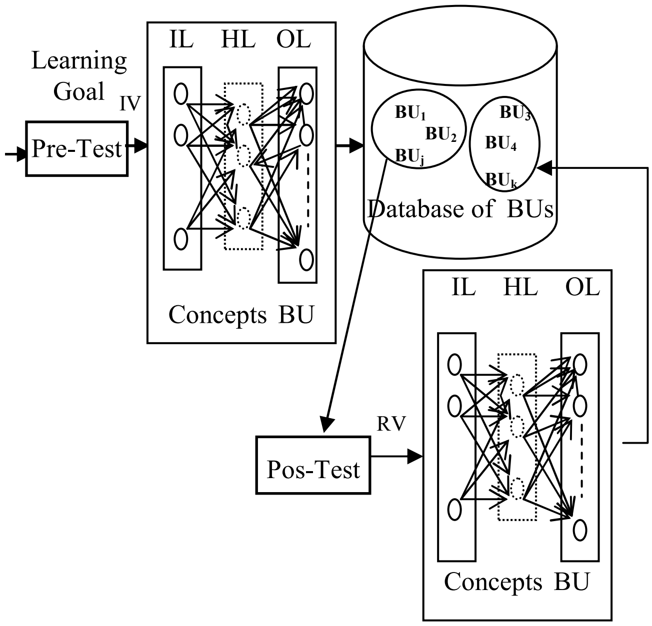

Figure.3. The developed connectionist based architecture, where: IL: Input Layer, HL: Hidden Layer, OL: Output Layer.

6.2. 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.

Several organizational 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 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 [25]. 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*.

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 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.

7. Implementation

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

Figure 4 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 it; 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 an 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, cheeks 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 in a way that is suitable for the learner.

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.

The system is implemented entirely in Java. A serve let 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 through the course by activating links of the presented didactic plan. As shown in Figure 4.

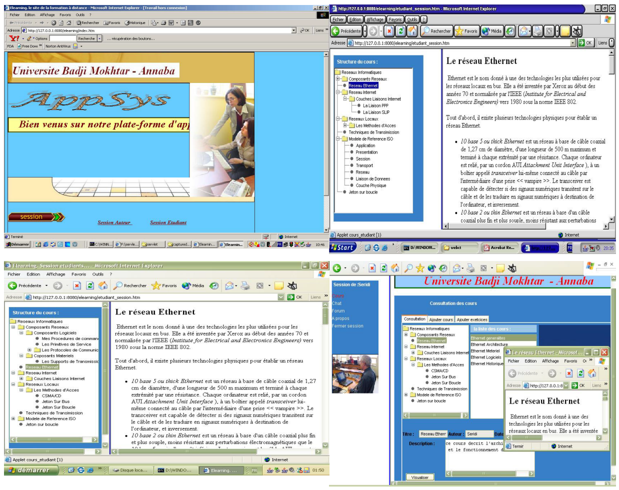

Figure 4. Learner and author interface.

8. Experiments

The presented architecture 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. 33 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.

Ninety undergraduate students' profiles and their respective didactic plans were computed manually by three different teachers of the course "computer networks". 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. In order to evaluate the total efficiency of the proposed methodology, an evaluation criterion have been designed and defined by: Success (%) = 100 * (Correct (BUs) selected/n), where n is the number of the desired BUs for each LG thatwill act as input to the instructional planner.

8.1. First ANN Evaluation and Testing

First, 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. The different scores mean that the NNs were able to generate the expected BUs.

Table 2. Experimental results on the first ANN training.

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%

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.

8.2. Second ANN Evaluation

The second NN was evaluated on the data from student who did not succeed the first stage. 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 3 shows the obtained results.

The different NNs approximate well on their respective data but have poor generalization 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.

Table 3. Results on the second ANN training.

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%

9. Conclusion

The constitution of adaptive training plans on Internet is an important field of research in the distance teaching. This paper had described some important parts of an adaptive learning environment, how they are designed and interact. The approach presented accommodates the goal of improving the learner's learning process by matching the lesson to their level of understanding and needs.

Comparing with other approaches, 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 experiences is build.

Using artificial neural networks for generating adaptive lessons demonstrates 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. MLPs 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] ARIADNE, Alliance of Remote Instructional Authoring and Distributions Networks for Europe, available at: http://ariadne.unil.ch /Metadata, 2001.

[2] Bloom's Taxonomy, available at: http://www. wested.org/tie/dlrn/Blooms.html, 1999.

[3] Brusilovsky P., Schwarz E, and Weber G., "A Tool for Developing Adaptive Electronic Textbooks on WWW," *in Proceedings of WebNet'96*, 1996.

[4] Brusilovsky P., "Adaptive Hypermedia," *User Modeling and User Adapted Interaction*, vol. 11, no. 1/2, pp. 87-110, 2001.

[5] Brusilovsky P. and Vassileva J., "Course Sequencing Techniques for Large-Scale Web-Based Education," *International Journal of Continuing Engineering Education and Lifelong Learning*, vol. 13, no. 1/2, pp. 75-94, 2003.

[6] De Bra P., Houben G. L, and Wu H., "AHAM: A Dexter-Based Reference Model for Adaptive Hypermedia," *in Proceedings of UK Conference on Hypertext,* pp. 147-156, 1999.

[7] Fensel D., Angele J., Decker S., Erdmann M., Schnurr H. P., Staab S., Studer R., and Witt A., "On2broker: Semantic-based Access to Information Sources at the WWW," in *Proceedings of the World Conference on the WWW and Internet (WebNet'99)*, Honolulu, Hawaii, pp. 366-371, 1999.

[8] Gruber T. R., "Toward Principles for the Design of Ontologies Used for Knowledge Sharing," in Guarino N. and Poli R. (Eds), Kluwer Academic Press, Boston, 1993.

[9] Henze N. and Nejdl W., "Adaptation in Ppen Corpus Hypermedia," *International Journal of Artificial Intelligence in Education*, vol. 12, no. 4, pp. 325-350, 2001.

[10] IMS, Standard for Learning Objects, available at: http://www.imsproject.org/matedata, May 2005.

[11] Kayama M. and Okamoto T., "A Mechanism for Knowledge-Navigation in Hyperspace with Neural Networks to Support Exploring Activities," *in Proceedings of Workshop 'Current Trends and Applications of Artificial Intelligence In Education' at the 4^{th} World Congress on Expert Systems*, Mexico, pp. 41-48, 1998.

[12] Kayama M. and Okamoto T., "Hy-SOM: The Semantic Map Framework Applied on an Example Case of Navigation," *in Proceedings of ICCE'99*, *7^{th} International Conference on Computer in Education*, Amsterdam, IOS Press, vol. 2, pp. 252-259, 1999.

[13] Kettel L., Thomson J., and Greer J., "Generating Individualized Hypermedia Applications," in *Proceedings of ITS'2000 Workshop on Adaptive and Intelligent Web-Based Education Systems*, Montreal, Canada, pp. 37-49, 2000.

[14] Knolmayer G. F., "Decision Support Models for Composing and Navigating Through E-Learning Objects," *in Proceedings of the 36^{th} IEEE Hawaii International Conference on System Sciences (HICSS'03)*, Hawaii, 2003.

[15] LOM, Draft Standard for Learning Object Metadata, available at: http://Itsc.ieee.org/wg12/index.html, October 2005.

[16] Masthoff J., "Automatic Generation of a Navigation Structure for Adaptive Web-Based Instruction," *in Proceedings of the AH'2002 Workshop on Adaptive Systems for Web-based Education*, Malaga, Spain, 2002.

[17] Melis E., Andres E., Budenberder J., Frishaud A., Goguadse G., Libbrecht P., Pollet M., and Ullrich C., "Active Math: A Generic and Adaptive Web-Based Learning Environment," *International Journal of AI in Education*, vol. 12, no. 4, 2001.

[18] McCalla I. and Greer E., "Granularity-Based Reasoning and Belief: Revision in Student Models," in Greer J. and McCalla G. (Eds), *Student Models, the Key to Individualized Educational Systems*, Springer-Verlag, Berlin, pp. 39-62, 1994.

[19] Michel A., Yen Si Y. G., "Analysis and Synthesis of a Class of Discrete-Time Neural Networks Described on Hypercubes," *IEEE Transactions Neural Network*, vol. 2, no. 1, pp. 32-46, 1991.

[20] Mohan P., Greer J., and McGalla G., "Instructional Planning with Learning Objects", *in Proceedings of the 18^{th} IJCIT Workshop on Knowledge Representation and Automated Reasoning for E-Learning Systems*, August 2003.

[21] Nkambo R., Gauthier G., and Frasson C., "Un Modèle de Représentation de Connaissances Relatives au Contenu Dans un Système Tutoriel Intelligent," *Sciences et Techniques Educatives,* Hermès, Paris, France, vol. 4, no. 3, pp. 299-330, 1997.

[22] Seridi H. and Sellami M., "An Adaptive Distance Learning Architecture," *in Proceedings of the 6^{th} International Conference on intelligent Tutoring Systems (ITS'2002)*, LNCS 2363, Springer-Verlag, Biarritz, France and San Sebastian, Spain, pp. 1010, June 2002.

[23] Seridi H. and Sellami M., "Un Environnement d'Enseignement à Distance Sur le Web," *Sciences et Techniques Educatives*; vol. 8, 2001.

[24] Seridi H., Sari T., and Sellami M., "A Connexionist Approach for Adaptive Lesson Generation," *in Proceedings of the International Conference IFIP of World Computer Congress (WCC)*, Toulouse, France, August 2004.

[25] Tennyson R. D. and Christensen D. L., "MAIS: An Intelligent Learning System," in Jonassen D. H. (Ed), *Instructional Designs for Microcomputer Courseware*, Hillsdale, N. J. Erlbaum, 1988.

[26] Verhoeven B., Cardinael K., Van Durn R., Duval E., and Olivié H., "Experiences with the ARIADNE Pedagogical Document Repository," *in Proceedings of Conference Educational Multimedia, Hypermedia & Telecommunications (ED-MEDIA'2001),* Tampere, Finland, AACE, pp. 1949-1954, 2001.

[27] W3C, Working Group, *Resource Description Framework*, available at: http://www.w3.org/RDF, 2002.

[28] Weber G., Kuhl H. C., and Weibelzahl S. "Developing Adaptive Internet Based Courses with the Authoring System Netcoach," *in Proceedings of 3^{th} Workshop on Adaptive Hypermedia*, AH'01, revised for LNCS, vol. 2266, pp. 226, Jan 2002.

Hassina Seridi received her master degree in computer science from the University of Badji Mokhtar, Algeria, in the specialty of artificial intelligence. She prepared her PhD in the educational technology field. Currently, she is an assistant professor at the Department of Information Science, University of Badji Mokhtar, Algeria, and a member of the Computer Research Laboratory at Annaba University. Her research interests are in the areas of adaptive hypermedia, adaptive web-based systems, learner modelling, and intelligent learning systems.

Toufik Sari received his BSc in computer science and his MSc from Badji Mokhtar University, Algeria, in 1996, 2000, respectively. Currently, he is a member of the RADAR team for pattern/speech recognition and document analysis at LRI laboratory, Algeria. He is pursuing a doctorate program in computer science, and is affiliated with Badji Mokhtar Annaba University as an associate professor. His interests are mainly in handwriting modelling and recognition, with special focus on hybrid systems. He is also involved in document analysis, medical image recognition, speech recognition, and neural networks.

Mokhtar Sellami received his PhD in computer sciences from the University of Grenoble, France, in the speciality of artificial intelligence and logic programming. He participates in many international research projects in Europe and Algeria. Currently, he is a director of Computer Research Laboratory at Annaba University and a senior lecturer in artificial intelligence and expert systems. His professional interests include pattern recognition applied to Arabic image processing, hybrid systems, and knowledge engineering.