2nd Hellenic Conf. on AI, SETN-2002, 11-12 April 2002, Thessaloniki, Greece,
Proceedings, Companion Volume, pp. 215-226

The Architecture of a Web-Based Intelligent Tutoring System Using Neurules as its Representational Basis

Jim Prentzas, Ioannis Hatzilygeroudis, Constantinos Koutsojannis

University of Patras, School of Engineering
Dept of Computer Engin. & Informatics, 26500 Patras, Hellas (Greece)
&
Computer Technology Institute, P.O. Box 1122, 26110 Patras, Hellas (Greece)
prentzas@ceid.upatras.gr, ihatz@cti.gr, ihatz@ceid.upatras.gr, ckoutsog@ceid.upatras.gr

Abstract. In this paper, we present the architecture of a Web-based Intelligent Tutoring System (ITS) in which knowledge representation is based on neurules, a type of hybrid rules integrating symbolic rules with neurocomputing. Neurules can be produced either from classical symbolic rules or from training data. The functionality of the ITS is controlled by a neurule-based inference engine. The system consists of three main parts: the domain knowledge containing the structure of the domain and the educational content, the user modeling component which records information concerning the user and the pedagogical model which encompasses knowledge regarding the various pedagogical decisions. The system's teaching subject focuses on Internet technologies.

1 Introduction

Intelligent Tutoring Systems (ITSs) form an advanced generation of Computer Aided Instruction (CAI) systems. Their key feature is their ability to provide a user-adapted presentation of the teaching material [1], [2]. This is accomplished by using Artificial Intelligence techniques to represent the pedagogical decisions and the information regarding each student.

ITSs have become extremely popular during the last years and have been shown to be quite effective at increasing their users' performance and motivation. The emergence of the World Wide Web increased the usefulness of such systems [3], [4], [5]. The Web's universality offers a versatile environment for testing the effectiveness of ITSs with numerous and diverse cases. Without doubt personalized environments prove to be an important asset in the field of e-learning [6].

In this paper, we describe the architecture of a Web-based ITS for teaching Internet technologies. It contains course units covering the needs of users with different knowledge levels and characteristics. The system models the students' knowledge state and skills. Based on this information, it constructs lesson plans and selects the appropriate course units for teaching each individual user. The functionality of the system is controlled by an expert system based on neurules, a type of hybrid rules integrating symbolic rules with neurocomputing [7].

The paper is organized as follows. Section 2 presents an overview of the system's architecture. Section 3 presents the knowledge representation formalism of the expert system and describes its advantages. Section 4 presents features of the domain knowledge. Section 5 describes the user modeling component. Section 6 presents the functionality of the pedagogical model. Finally, section 7 concludes.

2 System Overview

Fig. 1 depicts the basic architecture of the ITS. It consists of the following components:

- the domain knowledge, containing the structure of the domain and the educational content,
- the user modeling component, which records information concerning the user,
- the pedagogical model, which encompasses knowledge regarding the various pedagogical decisions,
- the user interface.

The ITS is based on an expert system aiming to control the teaching process. The expert system employs a hybrid knowledge representation formalism (i.e. neurules) [7]. According to their functionality, the neurules of the system are distributed into different neurule bases contained in the user modeling component and the pedagogical model. More specifically, there are four neurule bases, one in the user modeling component and three in the pedagogical model (in the teaching method selection module, course units' selection module, evaluation module).

The teaching subject (i.e. Internet technologies) of the ITS involves chapters such as the following:

- *Basic aspects of computer networks*
- *The Internet and its basic services*
- *The World Wide Web*
- *Email*

The following sections elaborate on the system's key aspects.

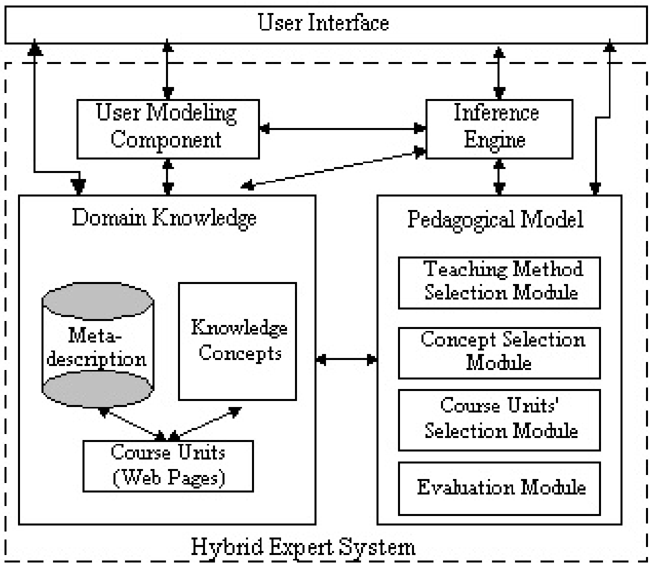

Fig. 1. Architecture of the ITS

3 Expert System

In this section, the semantics regarding the knowledge representation formalism of the expert system is briefly presented. Firstly, however, we present the desired features that should be encompassed by the knowledge representation formalism used in an ITS. In this way, the reasons why the specific knowledge representation formalism has been chosen as the basis of the ITS is clarified.

3.1 Prerequisites of an ITS's Knowledge Representation Formalism

The knowledge representation formalism used in an ITS should satisfy the following main criteria:

(a) *Efficiency*. Efficiency involves two primary aspects: time performance and storage space. ITSs are highly interactive knowledge-based systems requiring time-efficient responses to the users' actions. The Web imposes additional time constraints. The decisions an ITS makes during a training session are based on the conclusions reached by the inference engine associated with the knowledge representation formalism. The faster the conclusions can be reached, the faster will the system interact with the user. Therefore, the time performance of an ITS depends significantly on the time-efficiency of the inference engine. Furthermore, the storage space required by the knowledge representation formalism should not be excessive especially in case of Web-based ITSs.

(b) Ability to reach conclusions from *partially known inputs*. During a training session, certain parameters regarding the user may be unknown. The system should be able to operate effectively in such cases.

(c) *Easiness of updates*. There is always the possibility that the system's knowledge base may need to be changed. Most knowledge base updates usually take place during the construction stage when knowledge is acquired and the system prototype is implemented and tested. The operation of the system and the consequent feedback from users can also spark off changes to the knowledge base. The knowledge representation formalism should possess features that allow updates to be made easily.

(d) Ability to acquire knowledge from various knowledge sources. The knowledge representation formalism should provide mechanisms enabling the exploitation of various knowledge sources such as experts, available rule bases, databases containing training examples, etc.

As will be explained in the next section, neurules satisfy these criteria and for this reason were chosen as the representational basis of the ITS.

3.2 Neurules

The expert system has an inference engine in order to make decisions based on the known facts and the rule bases contained in the user modeling component and the pedagogical model. The expert system's knowledge representation formalism is based on neurules, a type of hybrid rules integrating symbolic rules with neurocomputing. The attractive feature of neurules is that they improve the performance of symbolic rules [7] and simultaneously retain their naturalness and modularity [8] in contrast to other hybrid approaches [9], [10].

The form of a neurule is depicted in Fig. 2a. Each condition C_i is assigned a number sf_i, called its *significance factor*. Moreover, each rule itself is assigned a number sf_0, called its *bias factor*. Internally, each neurule is considered as an adaline unit (Fig. 2b). The *inputs* C_i (i=1,...,n) of the unit are the *conditions* of the rule. The weights of the unit are the significance factors of the neurule and its bias is the bias factor of the neurule. Each input takes a value from the following set of discrete values: [1 (true), -1 (false), 0 (unknown)]. The *output D*, which represents the *conclusion* (decision) of the rule, is calculated via the formulas:

$$D = f(a), \quad a = sf_0 + \sum_{i=1}^{n} sf_i C_i$$

where a is the *activation value* and *f(x)* the *activation function*, which is a threshold function:

$$f(\mathbf{a}) = \begin{cases} 1 & \text{if } a \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

Hence, the output can take one of two values, ‘-1’ and ‘1’, representing failure and success of the rule respectively.

The general syntax of a condition C_i and the conclusion *D*:

<condition>::=<variable><l-predicate><value>

<conclusion>::=<variable><r-predicate><value>

where <variable> denotes a *variable,* that is a symbol representing a concept in the domain, e.g. ‘teaching-method’, ‘mark’ etc. <l-predicate> denotes a symbolic or a numeric predicate. The *symbolic predicates* are {is, isnot}, whereas the *numeric predicates* are {<, >, =}. <r-predicate> can only be a symbolic predicate. <value> denotes a value. It can be a *symbol* or a *number*.

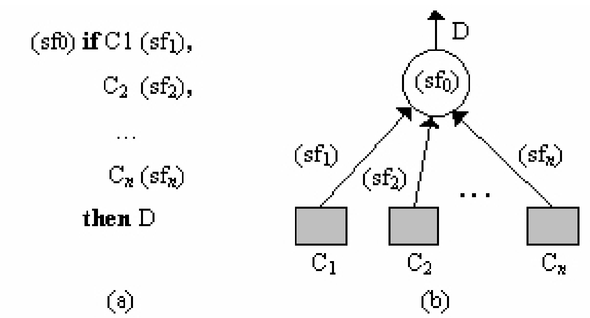

Fig. 2. (a) Form of a neurule (b) corresponding adaline unit

Neurules are constructed either from empirical data (training patterns) or symbolic rules. Each neurule is individually trained via the LMS algorithm. In case of inseparability in the training set, special techniques are used [7], [8]. In this way, the neurules contained in the neurule bases of the pedagogical model and the user modeling component are constructed. The inference mechanism is based on a hybrid rule-based inference engine [11]. It performs the task of classification: based on the values of the condition variables and the weighted sums of the conditions, conclusions are reached.

Neurules satisfy the criteria mentioned in the previous section and for this reason were chosen as the representational basis. More specifically:

(a) Neurules are efficient. On the one hand, neurules are time-efficient because they improve the performance of symbolic rules [7] and require fewer computations compared to other hybrid approaches in order to derive the inferences [11]. On the other hand, neurules are space-efficient since it has been proven that when neurules are constructed from symbolic rules, the number of rules contained in the rule bases is decreased reducing their required amount of space [7].

(b) In contrast to symbolic rules, neurule-based reasoning can derive conclusions from partially known inputs. This is due to the fact that neurules integrate a connectionist component (adaline).

(c) It is easy to update a neurule base because neurules retain the naturalness and modularity of symbolic rules enabling an incremental development of the neurule bases [7], [8]. Furthermore, the explanation mechanism produces natural explanations justifying how conclusions were reached [11]. This feature can assist in the location of deficiencies in the neurule base when the prototype system is tested.

(d) Neurules can be constructed either from symbolic rules [7] or empirical data [8] enabling the exploitation of various knowledge sources.

4 Domain Knowledge

The domain knowledge contains knowledge regarding the subject being taught as well as the actual teaching material. It consists of two parts: (a) the *knowledge concepts* and (b) the *course units*.

The knowledge concepts constitute the elementary pieces of knowledge for the given domain. Examples of concepts for the 'Internet technologies' teaching subject are the following: Web site, Web page, bookmark, file uploading, IP address, URL, hyperlink, discussion forums, mailing lists, search engine, etc. Every concept has a number of general attributes such as its name or its level of difficulty. Furthermore, it can have links to other concepts. These links denote its prerequisite concepts. In this way, one or more *concept networks* (Fig. 3) are formed representing the pedagogical structure of the domain being taught.

The concepts are organized into *concept groups.* A concept group contains closely related concepts based on the knowledge they refer to. Therefore, the domain space is dissected into subdomains. A concept group is associated with a teaching method bias. This parameter denotes preference to a specific teaching method (see Section 6) for teaching the concept group. Examples of subdomains in the 'Internet technologies' teaching subject are 'Computer Networks' and 'World Wide Web'.

Concept groups may contain a number of subgroups. 'World Wide Web' for instance contains subgroups such as 'Multimedia' which refers to the multimedia formats (e.g. static images, animations, sounds, videos) available on the Web.

The course units constitute the teaching material presented to the system users as Web pages. The teaching material involves a variety of courses starting from introductory topics and scaling up to more advanced ones. Each course unit is associated with a knowledge concept. The user is required to know this concept's prerequisite concepts in order to grasp the knowledge contained in the specific course unit. The distinct representation of the domain's pedagogical structure (concepts) and the actual teaching content (course units) facilitates the updates in the domain knowledge.

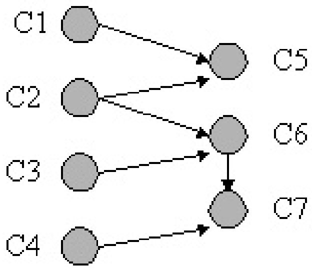

Fig. 3. Form of a concept network containing concepts C1, C2,...,C7.

A course unit may present theory, may be an example or an exercise. The examples assist the user in grasping the theory's key points. The exercises are based on the examples and are used to evaluate the user's knowledge level. When solving an exercise, the user can ask the system for help and view related examples.

The pedagogical model based on the user model selects and orders the course units presented to the user. In this way, a user-adapted presentation of the teaching material will be achieved. To this end, the explanation variant method implemented by the page variant technique [12] is used. More specifically, the system keeps variants of the same page (course unit) with different presentations.

The domain knowledge includes a *meta-description* of the course units containing their general attributes. Main such attributes for each course unit are its level of difficulty, its pedagogical type (theory, example, exercise), its multimedia type (e.g. text, images, animations, interactive simulations), the required Internet connection, etc. The meta-description of the course units follows the ARIADNE metadata recommendation (http://ariadne.unil.ch). The existence of the course units' meta-description provides the following benefits:

(a) it facilitates the selection and ordering of the course units by the pedagogical model,
(b) it assists the system administrator in managing the teaching material,
(c) it enhances the reusability of the course units and
(d) it facilitates the application of approaches such as the one described in [13] that enable the communication of the ITS with other intelligent educational systems.

5 User Modeling Component

The user modeling component is used to record information concerning the user which is vital for the system's user-adapted operation. It contains models of the system's users and mechanisms for creating these models (Fig. 4).

The user model consists of four types of items: (i) *personal data* (e.g. name, email), (ii) *interaction parameters*, (iii) *knowledge of the concepts* and (iv) *student characteristics*. The personal data concerns information necessary for the creation and management of the user's account. It is used for the identification of the user. The student characteristics and the knowledge of the concepts directly affect the teaching process whereas the interaction parameters indirectly.

The interaction parameters form the basis of the user model and constitute information recorded from the interaction with the system. They represent things like, the type and number of the course units accessed, the concepts and concept groups for which the user has accessed some of their course units, the type and the amount of help asked, the answers to the exercises, the marks obtained from the exercises, etc.

The student characteristics are mainly the following:

(a) Multimedia type preferences (e.g. text, images, or animations) regarding the presented course units.
(b) Knowledge level (novice, beginner, intermediate, advanced) of the subdomains and the whole domain.
(c) Learning ability level
(d) Concentration level
(e) Experience concerning the use of computers, hypermedia applications and the specific ITS.
(f) Available Internet connection.

The student characteristics are represented with the *stereotype model*, that is the user is assigned to predefined classes (stereotypes). The stereotypes denote typical users. Based on the way they acquire their values, the student characteristics are discerned into two groups. They can be either *directly obtainable* or *inferable*. The directly obtainable ones such as characteristics (a), (f) obtain their values directly from the user whereas the values of the inferable ones such as characteristics (b)-(e) are inferred by the system based on the interaction parameters and the knowledge of the concepts. The knowledge level of the whole domain is deduced from the knowledge levels of its subdomains. A neurule base containing *classification neurules* is used to derive the values of the inferable characteristics. The user models are dynamically updated during the teaching process.

The stereotype model cannot sufficiently represent the user's knowledge of the domain because the adaptation techniques of the pedagogical model require a more fine-grained model in order to be effective. For this reason the user's knowledge of the domain is represented as a combination of a stereotype and an *overlay model* [2]. The stereotype denotes the (sub)domain knowledge level. The overlay model is based on the concepts associated with the course learning units. More specifically, each concept is associated with a value denoting the user knowledge level of this concept.

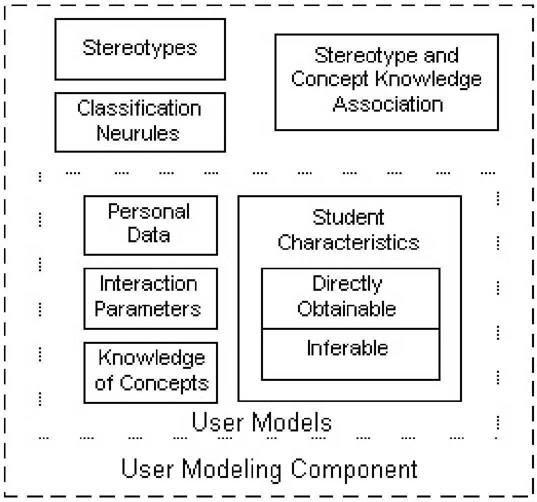

Fig. 4. The User Modeling Component

The combination of stereotype and overlay modeling for the representation of a user's domain knowledge has given good results [12]. The overlay model has the problem of initialization since for a new user it is hard to set the knowledge values for all the concepts. This difficulty can be overcome by associating a fixed set of pairs concept-value with each stereotype [12]. This association will be based on the concepts' level of difficulty. For instance, concepts with medium level of difficulty may be considered to be unknown to novice users or beginners and known to intermediate users or experts.

6 Pedagogical Model

The pedagogical model guides the teaching process. It provides the knowledge infrastructure in order to tailor the presentation of the teaching material according to the information contained in the user model. As shown in Fig. 1, the pedagogical model consists of four main components: (a) *teaching method selection module,* (b) *concept selection module*, (c) *course units' selection module* and (d) *evaluation module*. Each of these components but the concept selection module contains a neurule base.

In a learning session, the user is taught in a specific concept group in case it does not have concept subgroups. In case the concept group contains subgroups, a learning session will be required for each subgroup. In a specific learning session, the pedagogical model must sequentially perform the following tasks:
(i) Select a concept (sub)group to teach. This selection is based on the user's knowledge of the domain.
(ii) Select a teaching method for a specific concept (sub)group.
(iii) Select the concepts to be taught.
(iv) Select the course units to be presented.
(v) Evaluate the user's performance. The evaluation of the user's performance updates the inferable student characteristics and may create a feedback for tasks (ii) and (iv).

The teaching method selection module selects the appropriate teaching method for a specific concept (sub)group. It is important for an intelligent tutoring system to offer more than one tutoring strategy because this will entail into a richness of tutorial actions. If the system offers limited tutoring strategies it will be restricted in its pedagogical scope. A neurule base is used to select the teaching method based on parameters concerning the user model and the specific concept (sub)group. User parameters considered include the user's learning ability level, concentration level, knowledge level as well as the percentage of accessed course units within the specific concept (sub)group. In addition, the concept group's teaching method bias is taken into account. These parameters appear in the conditions of the neurules used to select the teaching method. There are totally six teaching methods. For instance, according to one such method in order to teach the user a specific concept (sub)group, course units containing theory, examples and exercises should be presented. Another method states that the most appropriate way of teaching would be to present only examples and exercises.

The task of the concept selection module is to construct a user-adapted lesson plan by selecting and ordering the appropriate concepts. This is based on the user's knowledge of the concepts, the user's (sub)domain knowledge level, the concepts' level of difficulty and the links connecting the concepts.

According to the plan constructed by the concept selection module, the course units' selection module selects and orders the course units that are suitable for presentation. A neurule base performs the selection and ordering task. For this purpose, the student characteristics of the user model as well as the meta-description of the course units are taken into account. These parameters appear in the conditions of the neurules.

The evaluation module evaluates the user's performance based on the user's interaction with the system and updates accordingly the user model. More specifically, based on the interaction parameters, it assigns knowledge values to the concepts and updates the inferable student characteristics by using the classification neurules of the user modeling component. The evaluation module contains *evaluation neurules* for assigning marks to the presented exercises. For each presented exercise, the user obtains a mark ranging from bad to excellent. The mark is given based on the number of times he/she asked for assistance, the number of related examples seen by the user, the number of answering attempts made by the user and if the answer was finally provided by the system or not. The conditions of the neurules contain these parameters. A similar approach for the evaluation of users is used in [1].

Based on the acquired marks, the knowledge values of the concepts as well as the knowledge levels of the concept subgroups and concept groups are derived. If the user gains an acceptable knowledge level of the concepts belonging in the initial lesson plan (based on the marks obtained from the exercises), another concept (sub)group will be selected and a new learning session will ensue. The system records if a teaching method has been used successfully for teaching a specific concept (sub)group. If the user performs badly in the exercises, remedial teaching involving tasks (ii) and (iv) will be necessary causing reselection of the teaching method and/or course units. Remedial action in the level of course units may involve selection of variants of the previously selected course units.

7 Conclusions

In this paper, we describe the design of a Web-based Intelligent Tutoring System (ITS) for teaching Internet technologies. The system tailors the presentation of the teaching material to the diverse needs of its users. The system's function is controlled by a hybrid expert system using neurules, a type of hybrid rules integrating symbolic rules with neurocomputing.

The use of neurules instead of symbolic rules or other hybrid approaches integrating symbolic rules with neurocomputing offers a number of advantages. Neurules are efficient, conclusions can be drawn from partially known inputs, neurule bases can be easily updated and various available knowledge sources can be exploited. Thus, neurules encompass the features desired by the knowledge representation formalism of an ITS.

Our future work will involve two aspects. The first aspect will concern the use of the described ITS architecture in order to implement an ITS for distant education of nursing students. The teaching subject will concern fundamental issues of the most common medical equipment. We believe that an ITS with this teaching subject is necessary because the introduction of new technologies in health care raises the issue of having trained personnel and imposes new demands for tools and methods for their overall management. The implementation of this ITS will offer the opportunity to test the effectiveness of the presented ITS architecture in another teaching subject.

The second aspect of our future work will involve the use of distributed Artificial Intelligence methods to achieve the communication of the ITS with other intelligent educational systems (i.e. ITSs or Adaptive Educational Hypermedia Systems) teaching the same or closely related subjects. For this purpose, an agent-based approach dealing with learning from multiple collaborating intelligent tutors [13] will be used. Communication between the intelligent educational systems will take place when the trainee is not satisfied with the teaching material contained in the educational system he/she is currently interacting with.

References

1. Angelides, M., Garcia, I.: Towards an Intelligent Knowledge Based Tutoring System for Foreign Language Learning. Journal of Computing and Information Technology 1 15-28 (1993).
2. Polson, M.C., Richardson, J.J.: Foundations of Intelligent Tutoring Systems. Lawrence Erlbaum Associates (1988).
3. Brusilovsky, P.: Adaptive and Intelligent Technologies for Web-based Education. Kunstliche Intelligenz 4 (1999) 19-25.
4. Stern, M., Woolf, B.: Curriculum Sequencing in a Web-based Tutor. In: Goettl, B.P., Halff, H.M., Redfield, C.L., Shute, V.J. (eds.): Fourth International Conference on Intelligent Tutoring Systems. Lecture Notes in Computer Science, Vol. 1452. Springer-Verlag, Berlin (1998).
5. Papanikolaou, K.A., Magoulas, G.D., Grigoriadou, M.: Computational Intelligence in Adaptive Educational Hypermedia. Proceedings of the International Joint Conference on Neural Networks (2000).
6. Karagiannidis, C., Sampson, D., Cardinali, F.: An Architecture for Defining Re-usable Adaptive Educational Content. Proceedings of the IEEE International Conference on Advanced Learning Technologies (2001).
7. Hatzilygeroudis, I., Prentzas, J.: Neurules: Improving the Performance of Symbolic Rules. International Journal on Artificial Intelligence Tools 9 (2000) 113-130.
8. Hatzilygeroudis, I., Prentzas, J.: Constructing Modular Hybrid Knowledge Bases for Expert Systems. International Journal on Artificial Intelligence Tools 10 (2001) 87-105.
9. Gallant, S.I.: Neural Network Learning and Expert Systems. MIT Press (1993).
10. Ghalwash, A.Z.: A Recency Inference Engine for Connectionist Knowledge Bases. Applied Intelligence 9 201-215 (1998).
11. Hatzilygeroudis, I., Prentzas, J.: An Efficient Hybrid Rule Based Inference Engine with Explanation Capability. Proceedings of the 14th International FLAIRS Conference. AAAI Press (2001) 227-231.
12. Brusilovsky, P., Kobsa, A., Vassileva, J. (Eds.): Adaptive Hypertext and Hypermedia. Kluwer Academic Publishers, Dordrecht, Netherlands (1998).
13. Solomos, K., Avouris, N.: Learning from Multiple Collaborating Intelligent Tutors: An Agent-Based Approach. Journal of Interactive Learning Research 10 (1999) 243-262.