A Web-Based Intelligent Tutoring System Using Hybrid Rules as Its Representational Basis

Jim Prentzas, Ioannis Hatzilygeroudis, and John Garofalakis

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

Abstract. In this paper, we present the architecture and describe the functionality of a Web-based Intelligent Tutoring System (ITS), which uses neurules for knowledge representation. Neurules are a type of hybrid rules integrating symbolic rules with neurocomputing. The use of neurules as the knowledge representation basis of the ITS results in a number of advantages. Part of the functionality of the ITS is controlled by a neurule-based inference engine. Apart from that, the system consists of four other 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, and the supervisor unit that controls the functionality of the whole system. The system focuses on teaching 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], [3], [13]. This is accomplished by using Artificial Intelligence methods to represent the pedagogical decisions and the information regarding each student. The emergence of the World Wide Web increased the usefulness of such systems [12].

Very significant for the development and operation of an ITS are the AI techniques it employs. The gradual advances in AI methods have been incorporated into ITSs resulting into more effective systems. During the past years, various AI formalisms have been developed for knowledge representation in knowledge-based systems such as symbolic rules, fuzzy logic, Bayesian networks, neural networks, case-based reasoning. Hybrid approaches (e.g. neuro-symbolic or neurofuzzy representations) integrating two or more formalisms have also been developed in an effort to create improved representations. A number of formalisms have been used for knowledge representation in ITSs [1], [7], [8], [9], [10]. Symbolic rules are perhaps the most prominent AI formalism used in ITSs. Till now, a few ITSs are based on hybrid

S.A. Cerri, G. Gouardères, and F. Paraguaçu (Eds.): ITS 2002, LNCS 2363, pp. 119–128, 2002.

formalisms (e.g. [7]). However, hybrid approaches can offer a number of benefits to ITSs not offered by single ones.

In this paper, we present the architecture and describe the functionality of a Web-based ITS, which uses a hybrid formalism for knowledge representation. The subject of the ITS is “Internet technologies”. Course units covering the needs of users with different knowledge levels and characteristics are offered. The system models the students’ knowledge state and skills and, based on this information, constructs lesson plans and selects the appropriate course units for teaching each individual user. The ITS uses neurules [4], a type of hybrid rules, to represent expert knowledge. Neurules offer a number of benefits to the ITS.

The paper is organized as follows. Section 2 presents an overview of the system’s architecture. Section 3 presents the knowledge representation formalism and 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 Architecture

Fig. 1 depicts the basic architecture of the ITS. It consists of the following components: (a) the *domain knowledge*, containing the structure of the domain and the educational content, (b) the *user modeling component*, which records information concerning the user, (c) the *pedagogical model*, which encompasses knowledge regarding the various pedagogical decisions and (d) the *supervisor unit*.

The ITS is based on an expert system aiming to control the teaching process. The expert system employs a hybrid knowledge representation formalism, called neurules [4]. According to their functionality, the neurules of the system are distributed into different neurule bases. 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 supervisor unit supervises the function of the ITS. It interacts with the other components of the ITS calling the inference engine of the expert system whenever it is necessary. Furthermore, it plays a user interface role. 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.

3 Knowledge Representation

The expert system has an inference engine in order to make decisions based on known facts and the rule bases contained in the user modeling component and the pedagogical model.

Symbolic rules constitute a popular knowledge representation scheme used in the development of expert systems. Rules exhibit a number of attractive features such as naturalness, modularity and ease of explanation. One of their major drawbacks is the difficulty in acquiring rules through the interaction with experts. Methods based on decision trees construct rules from training examples and deal with this problem. Another drawback is the inability to draw conclusions when the value of one or more conditions is unknown.

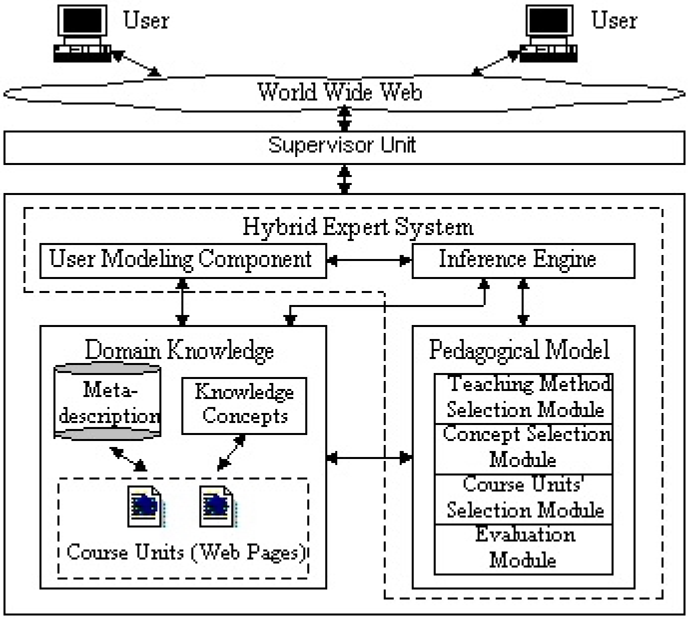

Fig. 1. The architecture of the ITS

During the last years, artificial neural networks are used quite often in the development of expert systems. Some of their advantages are the ability to obtain knowledge from training examples (with better generalization than the rules produced from decision trees), the high level of efficiency, the ability to reach conclusions based on partially known inputs and the ability to represent complex and imprecise knowledge. Their primary disadvantage is the fact that they lack the naturalness and modularity of symbolic rules. The knowledge encompassed in neural networks is in most cases incomprehensible.

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 [4] and simultaneously retain their naturalness and modularity [5] in contrast to other hybrid approaches.

3.1 Neurules

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:

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

where a is the *activation value* and f(a) the *activation function*, which is a threshold function returning '1' if a>=0 and '-1' otherwise. Hence, the output can take one of two values, '-1' and '1', representing failure and success of the rule respectively.

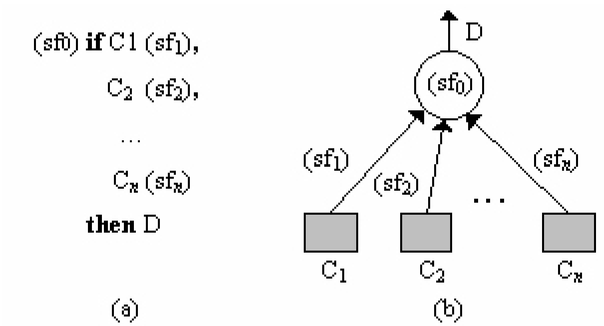

Fig. 2. (a) Form of a neurule (b) corresponding adaline unit

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', 'examination-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*. The significance factor of a condition represents the significance (weight) of the condition in drawing the conclusion(s). So, the semantics of significance factors are quite different from that of certainty factors or probabilities.

Neurules are constructed offline either from empirical data (training patterns) or symbolic rules using the methods described in [4], [5]. With these methods significance and bias factors are calculated; the user does not need to explicitly specify them. In this way, the neurules contained in the neurule bases of the ITS are constructed. The inference mechanism is based on a hybrid rule-based inference engine [6]. Conclusions are reached based on the values of the condition variables and the weighted sums of the conditions.

3.2 Benefits of Neurules

The use of neurules as the representational basis of the ITS results in a number of benefits, which enhance the construction stage as well as the real-time operation stage of the ITS. More specifically:

- Neurules are time-efficient because they improve the performance of symbolic rules [4] and require fewer computations compared to other hybrid approaches in order to derive the inferences [6]. This is very important since an ITS is a highly interactive knowledge-based system requiring time-efficient responses to users' actions. The Web imposes additional time constraints.
- 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 [4].
- 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). This feature is useful, because, during a training session, certain parameters related to the user may be unknown.
- 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 [4], [5]. One can easily add new neurules to or remove old neurules from a neurule base without making any changes to the knowledge base, since neurules are functionally independent units, given that they do not affect existing knowledge. This is difficult to do in other hybrid approaches. Ease of knowledge base updates is important, because there is always the possibility that the system's knowledge base should be changed.
- The explanation mechanism associated with neurules produces natural explanations justifying how conclusions were reached [6]. This feature can assist in the location of deficiencies in the neurule base when the prototype system is tested.
- Neurules can be constructed either from symbolic rules [4] or empirical data [5] enabling exploitation of alternative knowledge sources.

4 Domain Knowledge

Domain knowledge contains knowledge regarding the subject being taught as well as the actual teaching material. It consists of three parts: (a) *knowledge concepts*, (b) *concept (sub)groups* and (c) *course units*.

Knowledge concepts are elementary pieces of knowledge of the specific domain. Every concept has a number of general attributes such as name, level of difficulty, level of detail, lowest acceptable knowledge level. Furthermore, it can have links to other concepts. These links denote its prerequisite concepts. In this way, one or more *concept networks* are formed representing the pedagogical structure of the domain being taught.

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. Examples of subdomains in the 'Internet technologies' teaching subject are ‘Computer Networks’ and ‘World Wide Web’. Concept groups may contain a number of subgroups.

A concept (sub)group is associated with a teaching method bias denoting preference to a specific teaching method (see Section 6) for teaching the concept (sub)group. Another important attribute is the detail level of a concept (sub)group which can be compared with the user’s desired detail level of the presented educational content (see Section 5) in order to decide whether contents of the concept (sub)group will be presented or not. Furthermore, concept (sub)groups may be interconnected with precedence links used for the selection of the concept (sub)group to be taught. Some concept (sub)groups may be independent from the others meaning that their selection for teaching does not need to be preceded by the teaching of other (sub)groups.

The course units constitute the teaching material presented to the system users as Web pages. 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 course units present theory, examples or exercises.

The system keeps variants of the same page (course unit) with different presentations using the explanation variant method implemented by the page variant technique [2]. Domain knowledge includes a *meta-description* of the course units containing their general attributes such as the level of difficulty, the pedagogical type (theory, example, exercise), the multimedia type (e.g. text, images, animations, interactive simulations), the required Internet connection, the detail level. The meta-description of the course units is based on the ARIADNE recommendation.

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.

The user model consists of four types of items: (i) *personal data*, (ii) *interaction parameters*, (iii) *knowledge of concepts* and (iv) *student characteristics*. The personal data concerns information necessary for the creation and management of the user’s account (e.g. name, email). It is used for the identification of the user. The student characteristics and the knowledge of the concepts directly affect the teaching process whereas most of 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 course units accessed, the concepts and concept groups for which the user has accessed some of their course units, the type and amount of help asked, the correct and wrong answers to exercises, the marks obtained from 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) concentration level, (d) experience concerning the use of the

ITS, (e) available Internet connection, (f) desired detail level of the presented educational content.

Student characteristics are represented with the *stereotype model* that is, the user is assigned to predefined classes (stereotypes). Based on the way they acquire their values, the student characteristics are discerned into two groups: *directly obtainable* or *inferable*. The directly obtainable ones such as characteristics (a), (e), (f) obtain their values directly from the user whereas the values of the inferable ones such as characteristics (b)-(d) are inferred by the system based on the interaction parameters and knowledge of concepts. 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.

A neurule base containing *classification neurules* is used to derive the values of the inferable characteristics. The variables of the classification neurules' conclusions correspond to inferable characteristics. The variables of the conditions correspond to the parameters the inferable characteristics are based on. More specifically, the knowledge level of the subdomains is inferred based on the user's knowledge of the concepts belonging in the subdomains. The knowledge level of the whole domain is deduced from the knowledge level of the subdomains. The concentration level depends on the marks obtained from the exercises, the type and amount of help asked and the percentage of wrong answers. Experience is deduced from the knowledge level of the whole domain and the percentage of accessed course units.

6 Pedagogical Model

The pedagogical model provides the knowledge infrastructure in order to tailor presentation of the teaching material according to the user model. 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 specific learning session, the pedagogical model must perform the following tasks: (i) Select a concept (sub)group to teach, (ii) select-order the concepts to be taught, (iii) select a teaching method, (iv) select the course units to be presented, (v) evaluate the user's performance.

Selection of a concept (sub)group is based on the user's knowledge of the domain, links between concept (sub)groups, correspondence between concept (sub)groups' detail level and user's desired detail level of the presented educational content. Evaluation of the user's performance updates the inferable student characteristics and may create a feedback for tasks (iii) and (iv). In the following, the last four tasks are briefly described.

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 user's desired detail level, the concepts' detail level and the links connecting the concepts. More specifically, for the specific subdomain, the concepts for which the user's knowledge level is unsatisfactory are identified. These concepts are candidates for being selected in the construction of the lesson plan. Concepts whose detail level is incompatible with the user's desired detail level are eliminated from the candidate set. The lesson plan is formed based on the remaining set of concepts. Ordering of the selected concepts is performed based on the links connecting the concepts.

The teaching method selection module selects the appropriate teaching method using a neurule base. Selection is based on parameters concerning the user model and the specific concept (sub)group. User parameters considered include concentration level, knowledge level and 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 only examples and exercises should be presented. Table 1 (left column) presents an example neurule for selecting the teaching method.

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. For this purpose, the student characteristics of the user model, the selected teaching method as well as the meta-description of the course units are taken into account. Ordering of the course units is based firstly on their pedagogical type and secondly on their difficulty level. Ordering based on the pedagogical type is specified by the selected teaching method. A neurule base performs subsequent ordering based on the difficulty level. For instance, a specific ordering based on the difficulty level states that the presentation order of course units should start from course units with minor difficulty and proceed to more difficult ones. The variables of the neurules' conditions correspond to the inferable student characteristics and the teaching method.

Table 1. Example neurules for selecting the teaching method and assigning examination marks

TM-RULE (-2.4) **if** teach-meth-bias is examples-exercises (1.5) concentration-level is low (1.2), knowledge-level is low (1.0), percent-accessed-cunits < 0.30 (0.9), teach-meth-bias is theory-examples-exercises (0.9) **then** teaching-method is theory-examples-exercises	EVAL-RULE (-11.2) **if** attempts-solution>2 (10.2), number-requested-examples=0 (9.9), number-requested-examples=1 (6.4), times-asked-assistance=1 (6.3), times-asked-assistance=0 (3.3) **then** examination-mark is average

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 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 and the number of answering attempts made by the user. Table 1 (right column) presents an example evaluation neurule.

Based on the acquired marks, the knowledge values of the concepts as well as the knowledge levels of the concept (sub)groups and the whole domain are derived. The user's knowledge level of each concept belonging in the initial lesson plan should be greater than or equal to its lowest acceptable knowledge level. If this is the case, another concept (sub)group will be selected and a new learning session will ensue. Otherwise, tasks (iii) and (iv) will be re-executed causing reselection of the teaching method and/or course units since different inputs will be given to the corresponding neurules.

7 Conclusions and Future Work

In this paper, we present the architecture and describe the functionality of a Web-based ITS, which uses a hybrid formalism for knowledge representation. The system's function is controlled by an 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 neuro-symbolic approaches offers a number of advantages. Neurules encompass the features desired by the knowledge representation formalism of an ITS. The use of hybrid approaches in ITSs is likely to gain interest in the following years. Hybrid approaches are more efficient than their component representations. In fact, hybrid intelligent systems have been proven effective in solving difficult problems.

Our future work is directed to the use of Distributed AI methods (such as the one in [11]) to achieve communication of the ITS with other intelligent educational systems teaching the same or related subjects.

References

1. Angelides, M., Garcia, I.: Towards an Intelligent Knowledge Based Tutoring System for Foreign Language Learning. Journal of Computing and Information Technology 1 (1993) 15-28.
2. Brusilovsky, P., Kobsa, A., Vassileva, J. (Eds.): Adaptive Hypertext and Hypermedia. Kluwer Academic Publishers, Dordrecht, Netherlands (1998).
3. Georgouli, K.: Modelling a Versatile Mathematical Curriculum for Low-attainers. Proceedings of the 8th Panhellenic Conference in Informatics (2001) 463-472.
4. Hatzilygeroudis, I., Prentzas, J.: Neurules: Improving the Performance of Symbolic Rules. International Journal on Artificial Intelligence Tools 9 (2000) 113-130.
5. Hatzilygeroudis, I., Prentzas, J.: Constructing Modular Hybrid Knowledge Bases for Expert Systems. International Journal on Artificial Intelligence Tools 10 (2001) 87-105.
6. 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.
7. Magoulas, G.D., Papanikolaou, K.A., Grigoriadou, M.: Neuro-fuzzy Synergism for Planning the Content in a Web-based Course. Informatica 25 (2001) 39-48.
8. Martin, J., VanLehn, K.: Student assessment using Bayesian nets. International Journal of Human-Computer Studies 42 (1995) 575-591.
9. Nkambou, R.: Using Fuzzy Logic in ITS-Course Generation. Proceedings of the 9th IEEE International Conference on Tools With Artificial Intelligence. IEEE Computer Society Press (1997) 190-194.

10. Shiri, M. E., Aimeur, E., Frassen, C.: Student Modelling by Case-Based Reasoning. 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) 394-404.
11. Solomos, K., Avouris, N.: Learning from Multiple Collaborating Intelligent Tutors: An Agent-Based Approach. Journal of Interactive Learning Research 10 (1999) 243-262.
12. 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) 574-583.
13. Vassileva, J.: Dynamic Courseware Generation. Journal of Computing and Information Technology 5 (1997) 87-102.