ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/230821299

Knowledge Representation in Intelligent Educational Systems

Chapter · October 2006

DOI: 10.4018/978-1-59140-729-4

CITATIONS
33

READS
1,092

2 authors:

Ioannis Hatzilygeroudis
University of Patras
229 PUBLICATIONS 2,327 CITATIONS

SEE PROFILE

Jim Prentzas
Democritus University of Thrace
80 PUBLICATIONS 1,243 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project Teacher Professional Develoment in Emerging Technologies View project

Project Information Processing and Intelligence View project

All content following this page was uploaded by Ioannis Hatzilygeroudis on 30 May 2014.

The user has requested enhancement of the downloaded file.

To be published in Z. Ma (Ed.), "Web-Based Intelligent e-Learning Systems: Technologies and Applications", Idea Group Inc., 2005.

Knowledge Representation Intelligent Educational Systems

Ioannis Hatzilygeroudis[1,2] and Jim Prentzas[2, 3]

(1) University of Patras, School of Engineering
Dept of Computer Engin. & Informatics
26500 Patras, Hellas (Greece)

(2) Research Academic Computer Technology Institute
P.O. Box 1122, 26110 Patras, Hellas (Greece)

(3) Technological Educational Institute of Lamia
Dept of Informatics & Computer Technology
35100 Lamia, Greece

Abstract

In this chapter, we deal with knowledge representation in intelligent educational systems (IESs). We make an effort to define requirements for knowledge representation (KR) in an IES. The requirements concern all stages of an IES's life cycle (construction, operation and maintenance), all types of users (experts, engineers, learners) and all its modules (domain knowledge, user model, pedagogical model). We also briefly present and compare various KR schemes as far as the specified KR requirements are concerned. It appears that various hybrid approaches to knowledge representation can satisfy the requirements in a greater degree than that of single representations. Another finding is that there is not a hybrid scheme that can satisfy the requirements of all the modules of an IES. So, multiple representations or a multi-paradigm representation environment could provide a solution to requirements satisfaction.

Keywords: Knowledge Representation, Requirements Specification, Knowledge Engineering, Knowledge-Based Educational Systems, Intelligent Educational Systems Development, Intelligent E-Learning Systems

INTRODUCTION

Recent developments in computer-based educational systems had as a result a new generation of them encompassing intelligence, to increase their effectiveness, called *Intelligent Educational Systems* (IESs). Intelligent Tutoring Systems (ITSs) constitute a popular type of IESs. ITSs take into account the user's knowledge level and skills and adapt presentation of the teaching material to the needs and abilities of individual users. This is achieved by using Artificial Intelligence techniques to represent pedagogical decisions as well as domain knowledge and information regarding each student. ITSs were usually developed as stand-alone systems. However, the emergence of the WWW gave rise to a number of Web-based ITSs (Brusilovsky, 1999), which is a type of *Web-Based Intelligent Educational Systems* (WBIESs) (Hatzilygeroudis, 2004).

Another type of educational systems is Adaptive Educational Hypermedia Systems (AEHSs) (Brusilovsky et al, 1998). This type of systems is specifically developed for hypertext environments such as the WWW. The main services offered to their users are adaptive presentation of the teaching content and adaptive navigation by adapting the page hyperlinks. Compared to ITSs, they offer a greater sense of freedom to the user, since they allow a guided navigation to the user-adapted educational pages. Furthermore, they dynamically construct or adapt the educational pages, in contrast to

ITSs, where the contents of pages are typically static. Enhancing AEHSs with aspects and techniques from ITSs creates another type of WBIESs.

A crucial aspect in IESs (hence WBIESs) is making decisions on the proper adaptation of the system to the user needs. This is mainly done by mimicking corresponding human decision making. So, a crucial aspect in the development of an IES, hence of a WBIES, is how related knowledge is represented and how reasoning for decision making is accomplished. Various knowledge representation (KR) schemes have been used in IESs. An aspect that has not received much attention yet is defining requirements for knowledge representation in IESs. The definition of such requirements is important, since it can assist in the selection of the suitable KR scheme(s).

In this chapter, we present an effort to specify a number of requirements that a KR scheme, which is going to be used in an IES, should meet in order to be adequate. Based on them and a comparison of various KR schemes, we argue that hybrid schemes satisfy those requirements in a larger degree than single schemes. Such a hybrid scheme, called *neurules*, is presented as an example. However, our final argument is that only multiple representations or a multi-paradigm environment would be adequate for the development of an IES. This paper is an extension of (Hatzilygeroudis & Prentzas, 2004b).

The chapter is organized as follows. Section 2 specifies the KR requirements. Section 3 presents a number of KR schemes and how they satisfy the requirements. Section 4 makes a comparison of the KR schemes and, finally, Section 5 concludes.

KR REQUIREMENTS

Introductory Aspects

Like in other knowledge-based systems, we distinguish three main phases in the life cycle of an IES, the *construction phase*, the *operation phase* and the *maintenance phase*. The main difference is that an IES requires a great deal of feedback from the users and iteration between phases. Three types of users are involved: *domain experts*, *knowledge engineers* and *learners*. Each type of user has different requirements for the KR scheme(s) to be used. We call them *user requirements*, since they mainly concern the needs of the users.

Some of the user requirements are related to the general requirements for a KR language, such as efficiency and naturalness. Efficiency mainly refers to how quickly conclusions are drawn, whereas naturalness refers to how easy is to construct and understand sentences of a KR language as well as inference steps (Reichgelt, 1991).

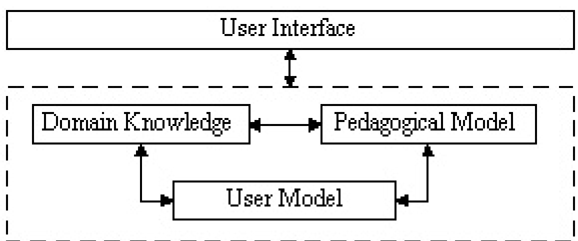

Figure 1. The Basic Structure of an Intelligent Educational System

On the other hand, the system itself imposes a number of KR requirements. An IES (as well as a WBIES) consists of three main modules (see Fig. 1): (a) the *domain knowledge*, which contains the teaching content and meta-information about the subject to be taught, (b) the *user model*, which records information concerning the user, and (c) the *pedagogical model*, which encompasses knowledge regarding various pedagogical decisions. Each component imposes different KR requirements. We call them *system requirements*, since they are related to the system components.

User Requirements

Domain Expert

The domain expert provides knowledge concerning the application domain. He/she is a person who knows in-depth the possible problems, the way of dealing with them as well as various practices obtained through his/her experience. In IESs, the domain experts are mainly the tutors. Tutors are highly involved in the construction and maintenance stages. However, in most cases, their relation to AI or even to computers is rather superficial. This may potentially make them restrained in their interaction with the knowledge engineer. Furthermore, the teaching theories to be incorporated in the system are rather difficult to express.

So, it is evident that one main requirement that tutors impose is *naturalness* of representation. Naturalness facilitates interaction with the knowledge engineer and helps the tutor in overcoming his/her possible restraints with AI and computers in general. In addition, it assists the tutor in proposing updates to the existing knowledge.

Also, checking validity of the represented knowledge is a tedious task, where the expert is involved. So, the capability of *providing explanations* is another requirement from the expert, which is of great help in checking represented knowledge.

Knowledge Engineer

The knowledge engineer manages the development of an IES and directs its various phases. The main tasks of the knowledge engineer are: acquire knowledge from the domain expert and/or other knowledge sources, select the implementation tools and effectively represent the acquired knowledge. He/she is the one who chooses (or designs) the KR scheme to be employed. Finally, he/she is who maintains the produced knowledge base.

Obviously, *naturalness* is again a basic requirement. The more natural the KR scheme, the easier it is for the knowledge engineer to transfer expert knowledge. Furthermore, tutors, during construction, may frequently change part (small or big) of the represented knowledge. Also, even if the system's operation is satisfactory, changes and updates of the incorporated expert knowledge may be required. This demands *ease of updates*.

Additionally, the KR scheme should facilitate the knowledge acquisition (KA) process. KA is usually a bottleneck in the development of a knowledge-based system. Facilitation can be achieved if the KR scheme allows acquiring knowledge from alternative (to experts) sources, such as databases of empirical data or past cases, in an automated or semi-automated way. So, *ease of knowledge acquisition* is another requirement.

Usually, in developing knowledge-based systems, a prototype is constructed before the final system. The prototype includes a small part of the whole knowledge. The rest of it is gradually added to the system. This is called *incremental development* of the system and it's a desirable feature. Furthermore, testing the continually incremented prototype can call for arduous efforts. In this context, two important factors are the inference engine performance and the capability of providing explanations. *Efficient inferences* reduce the time spent by the knowledge engineer. Also, *provision of explanations* is important, because it can assist in the location of deficiencies in the knowledge base.

End-User

An end-user (learner) is the one who uses the system in its operation stage. The basic requirement for KR, from the point of view of end-users, concerns time efficiency. IESs are highly interactive knowledge-based systems requiring time-efficient responses to the users' actions, which mainly depend on inference engine responses. In case of WBIESs, time performance is even more crucial, since the Web imposes additional time constraints, due to multiple users and the restricted communication bandwidth. Besides efficiency, the inference engine should also be able to reach *conclusions from partially known inputs*. During a learning session, the user may not be able or doesn't want to provide values for all parameters. However, the system should be able to make inferences without having all inputs known.

System Requirements

Types of Knowledge

System requirements refer to representation of the knowledge involved in the components of an IES. These requirements are mainly based on the required type(s) of involved knowledge, since different types of knowledge are more easily represented in different KR schemes (Reichgelt, 1991).

A first type of knowledge is called *structural knowledge*. Structural knowledge is concerned with types of entities (i.e. concepts, objects, etc) and how they are interrelated. It reflects the structure of the domain knowledge. Often, those relationships are hierarchical, i.e. they concern generalization/specialization relationships, e.g. "math is a form of an academic course, which itself is a form of a course". Another type of knowledge is *relational knowledge*. Relational knowledge concerns relations between entities of the domain. Those relations may be causal relations, e.g. "smoking causes cancer" or dependency relations, e.g. "mark depends on the number of attempts and the help asked".

From another point of view, there is *heuristic knowledge*. It is knowledge in the form of "rules of thumb", practical knowledge about how to solve problems based on experience. Sometimes, knowledge is not clear enough, but *uncertain* or *vague*. For example, values 'low' and 'medium', used to characterize the knowledge level of a student, are vague, since their boundaries are not clear. Also, knowledge may be not certain, but may have a degree of certainty.

Domain Knowledge

The domain knowledge module contains knowledge related to the subject to be taught as well as the actual teaching material. It usually consists of two parts: (a) knowledge model and (b) course units. Knowledge model refers to the basic concepts that constitute the subject to be taught and the types of relationships between them, e.g. the 'prerequisite', 'specialization', etc relationships. Finally, they are associated with course units, which constitute the teaching content.

Usually, concepts are organized in a type of structure. So, it is evident that a the KR scheme should be able to naturally represent *structural and relational knowledge*.

User Model

The user model (or student model) records information about the learner's knowledge state and traits. This information is vital for the system to be able to adapt to the user's needs. The process of inferring a user model from observable behavior is called 'diagnosis'. There are many possible user characteristics that can be recorded in the user model. One of them is the knowledge that he/she has learned. In this case, diagnosis refers to an estimation (or evaluation) of learner's knowledge level. Diagnosis of other characteristics such as, learning ability and concentration, means estimations based on learner behavior while interacting with the system.

Diagnosis of learner's characteristics is not a clear process. Also, there is not a clear-cut between various levels (values) of the characteristics. So, it is quite obvious that a representation scheme for the user model should be able to deal with *uncertain* and *vague knowledge*. Also, representation of *heuristic knowledge* is needed to make estimations about the values of the student characteristics.

Pedagogical Model

The pedagogical model represents the teaching process. It provides the knowledge infrastructure in order to tailor the presentation of teaching content according to the information recorded in the user model. The pedagogical model of a 'classical' IES mainly performs the following tasks: (a) course planning (or knowledge sequencing), (b) teaching method selection and (c) learning content selection. The main task in (a) is planning, that is selecting and appropriately ordering the concepts to be taught. The main task involved in (b) and (c) is also selection, e.g. how a teaching method is selected based on the learner's state and the learning goal. This is a reasoning process whose resulting conclusion depends on the logical combinations of the values of the user model characteristics, which reminds of *heuristic knowledge*. Furthermore, selection is not always clear, so *uncertain knowledge* representation may be required.

The above analysis of the requirements of knowledge representation for an IES is depicted in Tables 1 and 2.

KNOWLEDGE REPRESENTATION SCHEMES

In this section, we investigate satisfaction of the requirements specified above by various KR schemes. We distinguish between single and hybrid KR schemes.

Single Schemes

Structured Representations

Semantic nets and their descendants (*frames* or *schemas*) (Negnevitsky, 2002, ch. 5) represent knowledge in the form of a graph (or a hierarchy). Nodes in a semantic net graph represent concepts and the edges represent relations between the concepts. Nodes in a frame hierarchy also represent concepts, but they have internal structure that describes the corresponding concept via a set of attributes. They are very natural and well suited for representing structural and relational knowledge. They can also make efficient inferences for small to medium graphs (hierarchies). However, it is difficult to represent heuristic knowledge, uncertain knowledge and make inferences from partial inputs. Also explanations are not provided and knowledge updates are difficult. *Conceptual Graphs* are similar to semantic nets, whereas *ontologies* (Staab and Studer, 2004) refer to a representation scheme similar to frames, but are more restrictive.

In IESs, semantic networks have been used mainly for the representation of the domain knowledge structure.

Table 1. Users Requirements

USERS REQUIREMENTS		
Expert	Engineer	Learner
• *naturalness* • *explanations*	• *naturalness* • *ease of updates* • *incremental development* • *ease of knowledge acquisition* • *explanations*	• *efficient inferences* • *partial input inferences*

Table 2. System Requirements

SYSTEM REQUIREMENTS		
Domain Knowledge	User Model	Pedagogical Model
• *structural knowledge* • *relational knowledge*	• *vague knowledge* • *uncertain knowledge* • *heuristic knowledge*	• *heuristic knowledge* • *uncertain knowledge*

Symbolic Rules

Symbolic rules are one of the most popular KR methods (Negnevitsky, 2002, ch. 2). They represent general domain knowledge in the form of if-then rules: if <conditions> then <conclusion>, where the term <conditions> represents the conditions of a rule, whereas the term <conclusion> represents its conclusion. The conditions are connected with one or more logical operators such as ‘and’, ‘or’ and ‘not’. The conclusion of a rule is derived when the logical function connecting its conditions results to true. *Expert systems* constitute the most well known type of rule-based systems. The main parts of a typical expert system are: rule base, inference engine, working memory and explanation mechanism.

The inference engine uses the knowledge in the rule base as well as facts about the problem at hand to draw conclusions. Typically, facts are provided by the user during inference. There are two main inference methods: backward chaining (guided by the goals) and forward chaining (guided by the data). The explanation mechanism provides explanations regarding the drawn conclusions.

Rules are natural (easy to comprehend) and rule-base updates (removing/inserting rules) can be easily made. Also, incremental development of a rule base is a quite natural process. In addition, heuristic knowledge is naturally represented by rules. However, a major drawback is the difficulty in acquiring them. KA may turn out to be a bottleneck. Furthermore, the acquired rules may be imperfect. Efficiency of the inference process depends on the length of the inference chains. Additionally, conclusions cannot be derived if some of the inputs is unknown. Finally, pure rules cannot represent uncertain or vague knowledge and are not suitable for representing structural and relational knowledge.

Symbolic rules have been used in IESs mainly to diagnose the learner's characteristics and to perform various pedagogical tasks (Vassileva, 1998; Simic & Devedzic, 2003). The system described in (Vassileva, 1998) uses heuristic knowledge in the form of rules (classified into groups with different functionality) to manage course generation based on learner's performance and the domain knowledge.

Case-Based Representations

Case-based representations (Leake, 1996) store a large set of past cases with their solutions in the *case base* and use them whenever a similar new case has to be dealt with. A case-based system performs inference in four phases: (i) retrieve, (ii) reuse, (iii) revise and (iv) retain. In the retrieval phase the most relevant stored case(s) to the new case is(are) retrieved. Similarity measures and indexing schemes are used in this context. In the reuse phase, the retrieved case is combined with the new case, to create a solution. The revise phase validates the correctness of the proposed solution. Finally, the retain phase decides on retention (or not) of the new case.

Cases are usually easy to obtain and, unlike other schemes, case acquisition can also take place during the system's operation. Cases are natural. Explanations cannot be provided as straightforward as in rule-based systems, due to the similarity functions. Conclusions can be reached even if some of the inputs are not known, through similarity to stored cases. Updates can be easily made. However, the efficiency of the inference process depends on the size of the case base. Finally, cases are not suitable for representing structural, uncertain and heuristic knowledge.

In IESs, case-based reasoning has been used in the user model to assess the learner's knowledge and in the pedagogical model to perform instructional tasks (Shiri et al, 1998; Gilbert, 2000; Guin-Duclosson, 2002). The approach described in (Guin-Duclosson, 2002) uses case-based reasoning to teach problem-solving methods. The system enables to model the knowledge observed in learners by explicitly defining a problem classification, the reformulation and the solution knowledge associated with it. According to that model, an expert in the teaching domain defines a hierarchy of problem classes and reformulation knowledge for the classification of a new problem based on discriminating attributes.

Neural Networks

Neural networks represent a totally different approach to AI, known as connectionism (Gallant, 1993). A neural network consists of many simple interconnected processing units called *neurons*. Each connection from neuron u_j to neuron u_i is associated with a numerical weight w_{ij} corresponding to the influence of u_j to u_i. The output of a neuron is based on its inputs and corresponding weights. Usually, neural networks are organized in three levels: input, intermediate (or hidden) and output level. The weights of a neural network are determined via a *training* process via empirical data. Input neurons are fed with the input values of the problem. These values are propagated through the network and produce the outputs by activating the corresponding neurons.

Neural networks are very efficient in producing conclusions, since inference is based on numerical calculations and can reach conclusions based on partially known inputs, due to their generalization ability. On the other hand, neural networks lack naturalness of representation, that is the encompassed knowledge is incomprehensible, and explanations for the reached conclusions cannot be provided. It is also difficult to make structural updates to specific parts of the network. Neural networks do not possess inherent mechanisms for representing structural, relational and uncertain knowledge. Heuristic knowledge can be represented to some degree via supervised training.

The system in (Tchetagui & Nkambou, 2002) employs a neural network to classify the learner into a knowledge level.

Belief Networks

Belief networks (or *probabilistic nets*) (Russell & Norvig, 2003, ch. 14) are graphs, where nodes represent statistical concepts and links represent mainly causal relations between them. Each link is assigned a probability, which represents how certain is that the concept where the link departs from causes (leads to) the concept where the link arrives at. Belief nets are good at representing causal relations between concepts. Also, they can represent heuristic knowledge to some extend. Furthermore, they can represent uncertain knowledge through the probabilities and make relatively efficient inferences (via computations of probabilities propagation). However, estimation of probabilities is difficult, making KA process a problem. For the same reason, it is difficult to make updates. Also, explanations are difficult to produce, since the inference steps cannot be easily followed by humans. Furthermore, their naturalness is reduced.

In IESs, belief networks have been used mainly in user modeling (Jameson, 1995; Vanlehn & Zhendong, 2001; Tchetagui & Nkambou, 2002). The system in (Tchetagui and Nkambou, 2002) uses Bayesian reasoning to aggregate performance values throughout the network of the domain knowledge structure.

Hybrid Schemes

Hybrid schemes are integrations of two or more single KR schemes. In this section we focus on the most popular ones.

Fuzzy Rules

Fuzzy logic is good at representing imprecise and fuzzy terms, like ‘low’ and ‘high’. Fuzzy logic extends traditional logic and set membership by defining membership functions over the range [0.0, 1.0], where 0.0 denotes absolute falseness and 1.0

absolute truth. *Fuzzy expert systems* constitute the most popular application of fuzzy logic. In such systems, sets of *fuzzy rules* (Dubois et al, 1993) are used to infer conclusions based on input data. Fuzzy rules include fuzzy variables. Inference process includes three phases: fuzzification of inputs (via membership functions), application of fuzzy rules and defuzzification (to produce the output).

Given the above, fuzzy rules are good at representing vagueness. However, fuzzy rules are not as natural as symbolic rules (due to membership functions), fact that complicates the KA process and the updates to the rule base. It is difficult to specify membership functions. Inference is more complicated and less natural than in simple rule-based reasoning, although its overall performance is not worse (because a fuzzy rule corresponds to more than one simple rule). Provision of explanations is feasible, but not all reasoning steps can be explained.

Fuzzy rules have proven quite helpful in the user modeling component of various ITSs (Hwang, 1998; Nkambou 1999). The Web-Based ITS described in (Hwang, 1998) employs a fuzzy expert system to assess learner characteristics and guide the learning process. The user model records fuzzy characteristics (like knowledge level, concentration, etc) and non-fuzzy characteristics (like total session time, effective learning time, etc.). The non-fuzzy characteristics are used to determine the values of the fuzzy ones. Fuzzy rules are used for subject material selection.

Connectionist Rule-Based Representations

A number of neuro-symbolic approaches have been developed, but we concentrate here on *connectionist expert systems*, because they satisfy more requirements. *Connectionist expert systems* (Gallant, 1993) combine neural networks with rule-based expert systems. The knowledge base is a network whose nodes correspond to domain concepts. Dependency information regarding the concepts is used to create links among nodes. The network's weights are calculated through a training process using a set of training patterns. Besides the knowledge base, connectionist expert systems also consist of an inference engine and an explanation mechanism. Compared to neural networks, they offer more natural representation and can provide some type of explanation. Naturalness is enhanced due to the fact that most of the nodes correspond to domain concepts.

Neurofuzzy Representations

There are various ways to integrate neural networks and fuzzy logic (Nauck et al, 1997). We are interested in integrations where the two component representations are indistinguishable. Such integrations are the *fuzzy neural networks* and the *hybrid neuro-fuzzy representations*. Fuzzy neural networks are fuzzified neural networks: they retain the basic properties and architectures of neural networks and "fuzzify" some of their elements (i.e., input values, weights, activations, outputs). In a hybrid neuro-fuzzy system both, fuzzy techniques and neural networks, play a key role. Each does its own job in serving different functions in the system. Hybrid neuro-fuzzy systems seem to satisfy KR requirements in a greater degree than fuzzy neural networks. They combine more and in a more satisfactory way the benefits of their component representations.

The system described in (Magoulas et al, 2001) is an Adaptive Educational Hypermedia System, which uses neural and fuzzy modules to accomplish its tasks. Neural and fuzzy modules are used in the domain knowledge, the learner evaluation and the pedagogical model. This hybrid approach enables the representation of incomplete, imprecise and vague information about the learner and also exploits the generalization capability of neural networks.

Integrations of Rules and Cases

Another trend to hybrid knowledge representation is the *integrations of rule-based reasoning with case-based reasoning* (Golding and Rosenbloom, 1996). We refer here to approaches where one method (either rules or cases) dominates and not to balanced approaches, because reasoning in them is more complicated. In such systems, naturalness of the underlying components is retained. Compared to 'pure' case-based reasoning, their key advantage is the improvement in the performance of the inference engine and the ability to represent heuristic and relational knowledge. Furthermore, the synergism of rules and cases can cover up deficiencies of rules (improved knowledge acquisition) and also enable partial input inferences. The existence of rules in such hybrid schemes makes updates more difficult than 'pure' case-based representations. Also explanations can be provided but not as easily as in pure rule-based representations, given that similarity functions are still present.

Description Logics

Description Logics (DLs) (Baader et al, 2002) combine aspects from frames, semantic nets and logic. They consist of two main components, the Tbox and the Abox. Tbox contains definitions of concepts and roles (i.e. their attributes), called *terminological knowledge*, whereas ABox contains logical assertions about concepts and roles, called *assertional knowledge*. DLs offer clear semantics and sound inferences. They are usually used for building and maintaining ontologies as well as for classification tasks related to ontologies. Also, DLs can be built on existing Semantic Web standards (XML, RDF, RDFS). So, they are quite suitable for representing structural and relational knowledge. Also, as logic-based, they can represent heuristic knowledge. Furthermore, their Tboxes can be formally updated. Their representation is natural, but not as much as that of symbolic rules. Inferences in DLs may have efficiency problems. Explanations cannot be easily provided.

Neurules

Syntax and Semantics

Neurules are a type of hybrid rules integrating symbolic rules with neurocomputing (Hatzilygeroudis and Prentzas 2000 & 2001a). In contrast to other hybrid approaches, the constructed knowledge base retains the modularity of rules, since it consists of autonomous units (neurules), and also retains their naturalness in a great degree, since neurules look much like symbolic rules.

The form of a neurule is depicted in Figure 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), 0 (false), 0.5 (unknown)]. The output D represents the conclusion of the rule. The output can take one of two values (‘-1’, ‘1’) representing failure and success of the rule respectively.

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

<condition/conclusion>::= <variable> <predicate> <value>

where <variable> denotes a *variable*, e.g. ‘mark-level’, ‘solution-attempts’, etc. <predicate> denotes a symbolic (is, isnot) or a numeric (<, >, =) predicate (not used in conclusions). <value> denotes a value (a symbol or a number). The significance factor of a condition represents the significance (weight) of the condition in drawing the conclusion. Table 3 presents an example neurule, used in assigning examination marks to a student.

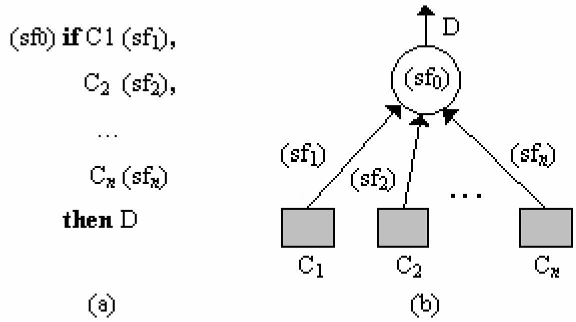

Figure 2. (a) Form of a neurule (b) corresponding adaline unit

Neurules can be constructed either from symbolic rules (Hatzilygeroudis and Prentzas 2000), thus exploiting existing symbolic rule bases, or empirical data (Hatzilygeroudis and Prentzas 2001a). Each adaline unit is individually trained via the Least Mean Square (LMS) algorithm.

Table 3. A neurule for assigning examination marks

(-9.7) **if** assistance-times is 1 (4.7), assistance-times is 0 (4.6), solution-attempts is 2 (4.6), requested-examples is >1 (3.2), requested-examples is 1 (1.4) **then** mark is average

A neurule-based system consists of the same basic components as a rule-based system. The neurule-based inference engine is based on a backward chaining strategy and uses neurules and facts (typically acquired from the user) to draw conclusions. Evaluation of a neurule is based on special neurocomputing measures (Hatzilygeroudis and Prentzas 2001b). A neurule fires if the output of the corresponding adaline unit is computed to be ‘1’ after evaluation of its conditions. A neurule is said to be ‘blocked’ if the output of the corresponding adaline unit is computed to be ‘-1’ after evaluation of its conditions.

Experiments have shown that the neurule-based inference process does better not only than simple rules but also than other similar systems, like MACIE (Gallant, 1993). Neurules are also associated with an explanation mechanism, capable of providing explanations of various types in the form of if-then rules. Experiments have shown that neurules explanation mechanism produces more natural explanations with less rules (Hatzilygeroudis & Prentzas 2001b).

Using neurules in an ITS

We constructed an intelligent tutoring system using neurules as its main knowledge representation scheme (Prentzas et al, 2002; Hatzilygeroudis & Prentzas 2004a). Neurules were used for representing knowledge in the user modeling unit and the pedagogical unit. In the user modeling unit, neurules were used for user classification in some stereotype and for student evaluation. In the pedagogical unit, they were used for three tasks: method selection, concept selection and unit selection. There is a neurule-based expert system, which make pedagogical decisions during the learning process, with a neurule-based inference engine and a neurule base consisting of five partial neurule bases, distributed between the user modeling and the pedagogical unit.

An important characteristic of the ITS is the existence of a special unit, called knowledge management unit (KMU). KMU has facilities for (a) acquiring knowledge from various sources (experts, existing symbolic rule bases, empirical data), (b) updating the knowledge stored in the neurule bases.

The use of neurules in the development of the ITS revealed a number of benefits:

- Neurules can be acquired in a semi-automated way from various sources, such as symbolic rules, empirical data or an expert. This is very important for IESs, given that KA is harder than other systems, due to the existence of more than one knowledge-based module.
- Neurules support incremental development of the neurule bases. One can easily add new neurules to or remove old neurules from a neurule base. This is difficult for other hybrid approaches.
- Neurules are space-efficient: produce quite smaller knowledge bases compared to simple rules. The size reduction in the ITS was 35-40%.
- Neurules can make robust inferences. In contrast to simple rules, neurules can derive conclusions from partially known inputs. This feature is useful, because, during a learning session, values of some parameters may be unknown.
- Neurules provide a more time-efficient inference engine than simple rules. This is very important, since an IES is a highly interactive knowledge-based system.
- Neurule bases can be efficiently updated, i.e. without thorough reconstruction of them. This is quite helpful during the construction and maintenance stage, where many updates are required. Knowledge base updates constitute a bottleneck for other hybrid approaches.

Despite the above benefits, we experienced some difficulties too. First, we could not use neurules to represent domain knowledge, due to its structural nature. So, we had to rely on degenerate (hence weak) representation methods, like relational tables. Another difficulty was that we could not represent vague knowledge. So, we had to use clear cuts among various classes of a test mark level or the knowledge level of a student (low, average, high, etc).

COMPARISON OF KR SCHEMES

Table 4 compares the KR schemes discussed in the previous sections, as far as satisfaction of KR requirements for IESs are concerned. Symbol ‘-‘ means ‘unsatisfactory’, ‘√-’ average, ‘√’ ‘good’ and ‘√+’ ‘very good’.

A conclusion that can be drawn from the table is that none of the single or hybrid schemes satisfies all the requirements for an IES. However, some of them satisfy the requirements of one or two modules of an IES. So, taking into account only the learner's and system requirements, one can say that semantic nets, frames, description logics and belief networks are more suitable for representing knowledge in the domain model. Also, fuzzy rules, belief networks and neuro-fuzzy representations are more suitable for the student modeling module. Finally, symbolic rules and neurules are more suitable for the pedagogical model. Hybrid schemes in general demonstrate improvements compared to most or all of their component schemes and therefore are preferable. However, a solution to the representational problem of an IES could be the use of different representation schemes (single or hybrid) for the implementation of different IES modules. Hence, the idea of a multi-paradigm development environment seems to be interesting.

Table 4. Comparison of KR schemes

	USERS REQUIREMENTS						SYSTEM REQUIREMENTS				
	Naturalness	*Ease of Updat*	*Efficient Inference*	*Explanations*	*Knowledge Acquisition*	*Partial input inferences*	*Structural knowledge*	*Relational knowledge*	*Uncertain knowledge*	*Vague knowledge*	*Heuristic knowledge*
Semantic nets/frames	√+	√-	√+	-	√	-	√+	√+	-	-	-
Symbolic rules	√+	√+	√	√+	√-	-	-	√-	-	-	√+
Case-based representations	√+	√+	√	√	√+	√	-	√	-	-	-
Belief networks	√-	-	√+	-	√-	-	√	√+	√+	√-	√-
Neural networks	-	-	√+	-	√+	√+	-	√-	-	-	√-
Fuzzy rules	√	-	√	-	√-	-	-	√-	√-	√+	√+
Connectionist expert systems	√-	√-	√+	√-	√+	√+	-	√-	-	-	√-
Neurofuzzy representations	√-	-	√	-	√	√-	-	√-	√-	√+	√
Cases and rules	√+	√	√	√	√	√	-	√	-	-	√
Description logics	√	√-	√-	√-	√	-	√+	√+	-	-	√
Neurules	√	√	√+	√+	√+	√+	-	√-	-	-	√+

CONCLUSIONS

In this paper, we make an effort to define requirements for knowledge representation in an IES. This work was motivated by the fact that we found symbolic rules inadequate in an effort to construct an ITS. The requirements concern all stages of an IES's life cycle (construction, operation and maintenance), all types of users (experts, engineers, learners) and all its modules (domain knowledge, user model, pedagogical model). According to our knowledge, such requirements have not been defined yet in the IES literature. However, we consider them of great importance as they can assist in choosing the KR schemes for representing knowledge in the components of an IES. To this end, we briefly present and compare various KR schemes. Our decision about the satisfaction level of a requirement by a KR scheme is based on advanced basic research results from the literature.

It appears that various hybrid approaches to KR can satisfy the requirements in a greater degree than that of single representations. The use of hybrid approaches to knowledge representation in IESs can become a popular research trend, although, till now, few IESs employ hybrid KR schemes. Another finding is that there is not a hybrid scheme that can satisfy the requirements of all of the modules of an IES, but each one individually. So, multiple representations or a multi-paradigm representation environment could provide a solution.

REFERENCES

Baader, F., Calvanese, D., McGuiness, D., Nardi, D., & Patel-Schneider, P.F. (2002). The Description Logic Handbook: Theory, Implementation and Applications. Cambridge University Press.

Brusilovski P. (1999). Adaptive and Intelligent Technologies for Web-based Education. In C. Rollinger and C. Peylo (Eds), Kustliche Intelligenz, Special Issue on Intelligent Systems and Teleteaching, 4, 19-25.

Brusilovsky P., Kobsa A., & Vassileva J. (Eds.) (1998). Adaptive Hypertext and Hypermedia. Kluwer Academic Publishers.

Dubois, D., Pride, H., & Yager, R. R. (1993). Fuzzy rules in knowledge-based systems, Reading in Fuzzy Sets for Intelligent Systems. Morgan Kaufmann.

Gallant, S.I. (1993). Neural Network Learning and Expert Systems. MIT Press.

Gilbert, J.E. (2000). Case-Based Reasoning Applied to Instruction Method Selection for Intelligent Tutoring Systems. Proceedings of the ITS'2000 Workshop on Case-Based Reasoning in Intelligent Training Systems, 11-15.

Golding, A.R., & Rosenbloom, P.S. (1996). Improving accuracy by combining rule-based and case-based reasoning. Artificial Intelligence, 87, 215-254.

Guin-Duclosson, N., Jean-Danbias, S., & Norgy, S. (2002). The AMBRE ILE: How to Use Case-Based Reasoning to Teach Methods. In Cerri, S.A., Gouarderes, G., Paraguacu, F. (eds.): Sixth International Conference on Intelligent Tutoring Systems. Lecture Notes in Computer Science, Vol. 2363. Springer-Verlag, Berlin, 782-791.

Hatzilygeroudis, I., & Prentzas, J. (2000). Neurules: Improving the Performance of Symbolic Rules. International Journal on Artificial Intelligence Tools, 9, 113-130.

Hatzilygeroudis, I., & Prentzas, J. (2001a). Constructing Modular Hybrid Rule Bases for Expert Systems. International Journal on Artificial Intelligence Tools 10 87-105.

Hatzilygeroudis, I., & Prentzas, J. (2001b). An Efficient Hybrid Rule Based Inference Engine with Explanation Capability. Proceedings of the 14th International Florida Artificial Intelligence Research Society Conference. AAAI Press, 227-231.

Hatzilygeroudis I. (Guest Editor) (2004). Special Issue on AI Techniques in Web-Based Educational Systems. International Journal on AI Tools (IJAIT), 13(2).

Hatzilygeroudis, I., & Prentzas, J. (2004a). Using a Hybrid Rule-Based Approach in Developing an Intelligent Tutoring System with Knowledge Acquisition and Update Capabilities. Journal of Expert Systems with Applications, 26, 477-492.

Hatzilygeroudis, I., & Prentzas, J. (2004b). Knowledge Representation Requirements for Intelligent Tutoring Systems. In Lester, J. C., Vicari, R. M. and Paraguacu, F. (eds.): 7th International Conference on Intelligent Tutoring Systems (ITS 2004). Proceedings. Lecture Notes in Computer Science, Vol. 3220. Springer-Verlag, Berlin, 87-97.

Hwang, G.-J. (1998). A Tutoring Strategy Supporting System for Distance Learning on Computer Networks. IEEE Transactions on Education, 41(4), 1-19.

Leake, D.B. (ed.) (1996). Case-Based Reasoning: Experiences, Lessons & Future Directions. AAAI Press/MIT Press.

Magoulas, G.D., Papanikolaou, K.A., & Grigoriadou, M. (2001). Neuro-fuzzy Synergism for Planning the Content in a Web-based Course. Informatica, 25, 39-48.

Nauck, D., Klawonn, F., & Kruse, R. (1997). Foundations of Neuro-Fuzzy Systems. John Wiley & Sons.

Negnevitsky, M. (2002). Artificial Intelligence, A Guide to Intelligent Systems. Addison Wesley.

Nkambou, R. (1999). Managing Inference Process in Student Modeling for Intelligent Tutoring Systems. In Proceedings of the Eleventh IEEE International Conference on Tools with Artificial Intelligence. IEEE Computer Society Press, 19-23.

Prentzas, J., Hatzilygeroudis, I., & Garofalakis J. (2002). A Web-Based Intelligent Tutoring System Using Hybrid Rules as its Representational Basis. In Cerri, S.A., Gouarderes, G., Paraguacu, F. (eds.): Sixth International Conference on Intelligent Tutoring Systems. Lecture Notes in Computer Science, Vol. 2363. Springer-Verlag, Berlin, 119-128.

Reichgelt, H. (1991). Knowledge Representation, An AI perspective. Ablex.

Russel, S., & Norvig, P. (2003). Artificial Intelligence, A Modern Approach. Prentice Hall.

Shiri, M. E., Aimeur, E., & Frasson, C. (1998). 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, Vol. 1452. Springer-Verlag, Berlin, 394-404.

Simic, G., & Devedzic, V. (2003). Building an Intelligent System Using Modern Internet Technologies. Expert Systems with Applications, 25, 231-146.

Staab, S., & Studer, R. (Eds) (2004). Handbook on Ontologies. Springer-Verlag.

Tchetagui, J. M. P., & Nkambou R. (2002). Hierarchical Representation and Evaluation of the Students in an Intelligent Tutoring System. In Cerri, S.A., Gouarderes, G., & Paraguacu, F. (eds.): Sixth International Conference on Intelligent Tutoring Systems. Lecture Notes in Computer Science, Vol. 2363. Springer-Verlag, Berlin, 708-717.

Vanlehn, K., & Zhendong, N. (2001). Bayesian student modeling, user interfaces and feedback: a sensitivity analysis. International Journal of AI in Education, 12, 155-184.

Vassileva, J. (1998). Dynamic Course Generation on the WWW. British Journal of Educational Technologies, 29, 5-14.

View publication stats