Knowledge Representation Requirements for Intelligent Tutoring Systems

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

University of Patras, School of Engineering
Department of Computer Engineering & Informatics, 26500 Patras, Greece
{prentzas, ihatz}@ceid.upatras.gr
Research Academic Computer Technology Institute, P.O. Box 1122, 26110 Patras, Greece
ihatz@cti.gr

Abstract. In this paper, we make a first effort to define requirements for knowledge representation (KR) in an ITS. The requirements concern all stages of an ITS'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 formalisms used (or that could be used) in ITSs 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 formalism that can satisfy the requirements of all of the modules of an ITS, but each one individually. So, a multi-paradigm representation environment could provide a solution to requirements satisfaction.

1 Introduction

Intelligent Tutoring Systems (ITSs), either Web-based or not, form an advanced generation of Computer Aided Instruction (CAI) systems. The key feature of ITSs is their ability to provide a user-adapted presentation of the teaching material. This is mainly accomplished by using Artificial Intelligence (AI) techniques.

A crucial aspect in the development of an ITS is how related knowledge is represented and how reasoning for problem solving is accomplished. Various single knowledge representation (KR) schemes have been used in ITSs such as, symbolic rules [10], fuzzy logic [7], Bayesian networks [9], case-based reasoning [3]. Also, hybrid representations such as, neuro-symbolic [5], [8] and neuro-fuzzy [6], have been recently used. Hybrid approaches integrate two or more single formalisms and are an emerging type of knowledge representation in ITSs in an effort to enhance the representational and reasoning capabilities of them.

An aspect that has not received much attention yet is defining requirements for knowledge representation in ITSs. The definition of such requirements is important, since it can assist in the selection of the KR formalism(s) to be employed by an ITS.

J.C. Lester et al. (Eds.): ITS 2004, LNCS 3220, pp. 87–97, 2004.

It is desirable that a knowledge representation formalism satisfy most, if not all, of them.

In this paper, we present a first effort to specify a number of requirements that a KR&R formalism, which is going to be used in an ITS, should meet in order to be adequate. The requirements refer to all stages of an ITS's life cycle (construction, operation and maintenance). They are also based on all types of users involved in those phases (experts, knowledge engineers, learners) as well as on the three basic modules of an ITS (domain knowledge, user model and pedagogical model). Based on them and a comparison of various KR formalisms, we argue that hybrid formalisms satisfy those requirements in a larger degree than single formalisms, because hybrid formalisms exhibit significant improvements compared to their component formalisms. Our final argument is that only a multi-paradigm environment would be adequate for the development of an ITS.

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

2 KR Requirements for ITSs

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

On the other hand, the system itself imposes a number of requirements to the KR formalism. An ITS consists of three main modules: (a) the *domain knowledge*, which contains the teaching content and 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.

2.1 Users Requirements

2.1.1 Domain Expert

The domain expert provides knowledge concerning the application domain. He/she is a person that has worked in the application field for an ample time period and knows in-depth the possible problems, the way of dealing with them as well as various practices obtained through his/her experience. In ITSs, the domain experts are mainly the tutors. Tutors are interested in testing teaching theories in practice to demonstrate their usability. They consider that the effectiveness of the theories in assisting students to learn the teaching subject is of extreme importance. Tutors are highly involved in the construction and maintenance stages. However, in most cases, their relation to AI is rather superficial. Sometimes even their experience in computers is low. This may potentially make them restrained in their interaction with the knowledge engineer. Furthermore, the teaching theories they want to incorporate within the system can be rather difficult to express.

So, it is evident that one main requirement that tutors impose on the knowledge representation formalism 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. The more natural the knowledge representation formalism, the better understanding of the existing knowledge and communication with the knowledge engineer.

Also, checking knowledge during the knowledge acquisition process is a tedious task. Capability of *providing explanations* is quite helpful for the expert. So, this is another requirement. On the other hand, if the knowledge base can be easily updated, then existing items of the acquired knowledge can be easily removed or updated and new items can be easily inserted. This demands *ease of update*.

2.1.2 Knowledge Engineer

The knowledge engineer manages the development of the ITS and directs its various phases. The main tasks of the knowledge engineer are to select the implementation tools, to acquire knowledge from the domain expert and/or other knowledge sources and to effectively represent the acquired knowledge. He/she is the one who decides on how expert knowledge is to be represented. He/she chooses or designs the knowledge representation formalism 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 formalism, the easier it will be for the knowledge engineer to translate expert knowledge. Furthermore, tutors, during construction, may frequently change part (small or big) of the knowledge imparted to the knowledge engineer. Also, even if the system's operation is satisfactory, changes and updates of the incorporated expert knowledge may be required.

Additionally, the KR formalism should facilitate the knowledge acquisition process. This can be achieved if the KR formalism allows acquiring knowledge from alternative (to experts) sources, such as databases of empirical data or past cases, in an automated or semi-automated way. In this way, more existing knowledge sources can be exploited and the knowledge acquisition process will not be hindered by the unavailability of a type of source (e.g. experts). So, *ease of knowledge acquisition* is another requirement.

Usually, in developing knowledge-based systems, a prototype is constructed before the final system. Testing the prototype can call for arduous efforts. As far as the KR formalism is concerned, two important factors are the inference engine performance and the capability of providing explanations. If the inference engine associated with the KR formalism is efficient, the time spent by the knowledge engineer is reduced. Also, the possibility of an explanation mechanism associated with the KR formalism is important, because explanations justifying how conclusions were reached can be produced. This feature can assist in the location of deficiencies in the knowledge base. Hence, two other requirements are *efficient inferences* and *explanation facility*.

2.1.3 End-User

An end-user (learner) is the one who uses the system in its operation stage. He/she imposes constraints regarding the user-interface and the time performance of the system. The basic requirement for KR, from the point of view of end-users, concerns time efficiency. ITSs are highly interactive knowledge-based systems requiring time-efficient responses to the users' actions. 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 significantly depends on the time-efficiency of the inference engine. In case of Web-based ITSs, time performance is even more crucial since the Web imposes additional time constraints. The server hosting the ITS may be accessed by a significant number of users. Some of them may even possess a low communication bandwidth. The server must respond as fast as possible. Besides efficiency, the inference engine should also be able to *reach conclusions from partially known inputs*. It is very common that, during a learning session, certain parameters may be unknown. However, the system should be able to make inferences and reach conclusion, no matter whether all or some of the inputs are known.

2.2 System Requirements

2.2.1 Domain Knowledge

The domain knowledge module contains knowledge regarding the subject to be taught as well as the actual teaching content. It usually consists of two parts: (a) knowledge concepts and (b) course units. Knowledge concepts refer to the basic entities/concepts that constitute the subject to be taught. Furthermore, various concepts are related among them, e.g. by the prerequisite relation, specialization relation etc. Finally, they are associated with course units. Course units constitute the teaching content.

Usually, concepts are organized in a type of structure. So, it is evident that a requirement that comes out of domain knowledge is the capability of the KR formalism to be able to naturally represent *structural and relational knowledge*.

2.2.2 User Model

The user 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, because it is much like the medical task of inferring a hidden physiological state from observable signs. 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 evaluation of learner's knowledge level. Other characteristics may be 'learning ability' and 'concentration'. Diagnosis in those cases means estimation of the learning ability and the concentration of the learner, based on his/her behavior while interacting with the system. Measurement and interpretation of such user behavior is quite uncertain.

There is not a clear process for evaluating learner's characteristics. Also, there is no a clear-cut between various levels (values) of the characteristics (e.g. between 'low' and 'medium' concentration). It is quite clear that a representation and reasoning formalism for the user model should be able to deal with *uncertain and vague knowledge*. Also, *heuristic (rule of thumb) knowledge* is required to make evaluations.

Table 1. Users Requirements

USERS REQUIREMENTS		
Expert	Engineer	Learner
• *naturalness* • *ease of update*	• *naturalness* • *ease of update* • *multi-source knowledge acquisition* • *explanation facility*	• *efficient inferences* • *partial input inferences*

Table 2. System Requirements

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

2.2.3 Pedagogical Model

The pedagogical model represents the teaching process. It provides the knowledge infrastructure in order to tailor the presentation of teaching the content according to the information recorded in the user model. The pedagogical model of a 'classical' ITS 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 a rule-type of knowledge or generally of heuristic knowledge. The above analysis of the requirements of knowledge representation for an ITS is depicted in Tables 1 and 2.

3 Knowledge Representation Formalisms

In this section, we investigate to what extent various well-known knowledge representation formalisms satisfy the requirements imposed by the developers, the users and the components of an ITS. We distinguish between single and hybrid KR formalisms.

3.1 Single Formalisms

Semantic nets and their descendants (*frames* or *schemata*) represent knowledge in the form of a graph (or a hierarchy). Nodes in the graph represent concepts and edges represent relations between concepts. Nodes in a hierarchy also represent concepts, but they have internal structure describing concepts via sets 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 knowledge updates are difficult.

Symbolic rules (of prepositional type) represent knowledge in the form of if-then rules. They satisfy a number of the requirements. Symbolic rules are natural since one can easily comprehend the encompassed knowledge and follow the inference steps. Due to their modularity, updates such as removing existing rules or inserting new rules are easy to make. Explanations of conclusions are straightforward and of various types. Heuristic knowledge representation is feasible and procedural knowledge can be represented in their conclusions too. The inference process may be not very efficient, when there is a large number of rules and multiple paths are to be followed. Knowledge acquisition is one of their major drawbacks. Also, conclusions cannot be reached if some of the inputs are unknown. Finally, they cannot represent uncertain knowledge and are not suitable for representing structural and relational knowledge.

Fuzzy logic is used to represent imprecise and fuzzy terms. Sets of *fuzzy rules* are used to infer conclusions based on input data. Fuzzy rules outperform symbolic rules and other formalisms in representing uncertainty. However, fuzzy rules are not as natural as symbolic rules, because the concepts contained in them are associated with membership functions. Furthermore, for the same reason, compared to symbolic rules, they have great difficulties in making updates, providing explanations and acquiring knowledge (e.g. for specifying membership functions). Inference is more complicated and less natural than symbolic rule-based reasoning, but its overall performance is not worse due, because a fuzzy rule can replace more than one symbolic rule. Explanations are feasible, but not all reasoning steps can be explained. Finally, fuzzy rules are much like symbolic rules as to structural, heuristic and relational knowledge as well as the ability to perform partial input inferences.

Case-based representations store a large set of previous cases with their solutions and use them whenever a similar new case has to be dealt with. Case-based representation satisfies several requirements. Cases are usually easy to obtain in most domains and unlike other formalisms case acquisition can also take place during the system's operation further enhancing the knowledge base. Cases are natural since their knowledge is quite comprehensible by humans. Explanations cannot be easily provided in most situations, due to the complicated numeric similarity functions. Conclusions can be reached even if some of the inputs are not known, through similarity to stored cases. Updates can be made easier compared to other formalisms, since no changes need to be made in preexisting knowledge. However, inference efficiency is not always the desirable when the case library becomes very large. Finally, cases are not suitable for representing structural, uncertain and heuristic knowledge.

Neural networks represent a totally different approach to AI, known as connectionism. Neural networks can easily obtain knowledge from training examples, which are usually available in abundance for most application domains. Neural networks are very efficient in producing conclusions and can reach conclusions based on partially known inputs due to their generalization ability. On the other hand, neural networks lack naturalness. The encompassed knowledge is in most cases incomprehensible and explanations for the reached conclusions cannot be provided. It is also difficult to make updates to specific parts of the network. The neural network is not decomposable and any changes affect the whole network. Neural networks do not possess inherent mechanisms for representing structural, relational and uncertain knowledge. Heuristic knowledge can be represented to some degree since it can be implicitly incorporated into a trained neural network.

Belief networks (or *probabilistic nets*) 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 (lead 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 a difficult task, which gives great problems to the knowledge acquisition process. 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, given that belief networks representation and reasoning are based on numerical computation, their naturalness is reduced.

3.2 Hybrid Formalisms

Hybrid formalisms are integrations of two or more single KR formalisms. In this section we focus on approaches belonging to the most popular categories of hybrid formalisms that is, symbolic-symbolic, neuro-symbolic, neuro-fuzzy and integrations of rule-based and case-based formalisms.

Connectionist expert systems [1] are neuro-symbolic integrations combining neural networks with expert systems. The knowledge base is a network whose nodes correspond to domain concepts. They 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. However, the additional (unknown) nodes inserted to deal with inseparability affect negatively the naturalness of the knowledge base and the provided explanations. In all other aspects, connectionist expert systems behave like neural networks.

There are various ways to integrate neural networks and fuzzy logic. We are interested in integrations that 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, that is 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 (usually knowledge is contained and applied by the connectionist part, but is described and presented by the fuzzy model). 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.

Another trend to hybrid knowledge representation are the *integrations of rule-based with case-based reasoning* [2]. We refer here to the approaches where rules dominate. Rules correspond to general knowledge, whereas cases correspond to specific knowledge. These hybrid approaches effectively combine the best features of rules and cases. 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 in the rule base (improved knowledge acquisition) and also enable partial input inferences. The existence of rules in these hybrid formalisms makes updates more difficult than 'pure' case-based representations. Also explanations can be provided but not as easily as in 'pure' rule-based reasoning because inference becomes more complicated, since similarity functions are still present.

Description Logics (DLs) can be also considered as hybrid KR formalisms, since they 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), which are called terminological knowledge, whereas ABox contains logical assertions about concepts and roles, which are 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, since they are based on logic, 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 are a type of hybrid rules integrating symbolic rules with neurocomputing, introduced by us [4]. The most attractive features of neurules are that they improve the performance of symbolic rules and simultaneously retain their modularity and, in a large degree, their naturalness, in contrast to other hybrid approaches. So, neurules offer a number of benefits for knowledge representation in an ITS. Apart from the above, updating a neurule base (add to or remove neurules from) is easy, due to the modularity of neurules [5]. The explanation mechanism produces natural explanations. Neurule-based inference is more efficient than symbolic rule-based reasoning and inference in other hybrid neuro-symbolic approaches. Neurules can be constructed either from symbolic rules or from empirical data enabling the exploitation of various knowledge sources [5]. In contrast to symbolic rules, neurule-based reasoning can derive conclusions from partially known inputs, due to its connectionist part.

Table 3. Comparison of KR formalisms

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

4 Discussion

Table 3 compares the KR formalisms discussed in Section 3, as far as satisfaction of KR requirements for ITSs 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 formalisms satisfies all the requirements for an ITS. However, some of them satisfy the requirements of the different modules of an ITS. Hybrid formalisms demonstrate improvements compared to most or all of their component formalisms. So, a solution to the representational problem of an ITS could be the use of different representation formalisms (single or hybrid) for the implementation for different ITS modules (i.e. domain knowledge, user model, pedagogical model). Then, the idea of a multi-paradigm development environment seems to be interesting. The next problem, though, is which KR paradigms should be included in such an environment.

5 Conclusions

In this paper, we make a first effort to define requirements for KR in an ITS. The requirements concern all stages of an ITS's life cycle (construction, operation and maintenance), all types of users (experts, engineers, kearners) and all its modules (domain knowledge, user model, pedagogical model). According to our knowledge, such requirements have not been defined yet in the ITS literature. However, we consider them of great importance as they can assist in choosing the KR formalisms for representing knowledge in the components of an ITS.

From our analysis, it appears that various hybrid approaches to knowledge representation can satisfy the requirements in a greater degree than that of single representations. So, we believe that use of hybrid KR approaches in ITSs can become a popular research trend, although, till now, only a few efforts exist. Another finding is that there is not a hybrid formalism that can satisfy the requirements of all of the modules of an ITS. So, a multi-paradigm representation could provide a solution.

We feel that our research needs to be further completed by getting more detailed and more specific to ITSs nature. What is further needed is a more in-depth analysis of the three modules of an ITS. Also, a more fine-grained comparison of the KR formalisms may be required. These are the main concerns of our future work.

Acknowledgements. This work was supported by the Research Committee of the University of Patras, Greece, Program "Karatheodoris", project No 2788.

References

1. Gallant, S.I.: Neural Network Learning and Expert Systems. MIT Press (1993).
2. Golding, A.R., Rosenbloom, P.S.: Improving accuracy by combining rule-based and case-based reasoning. Artificial Intelligence 87 (1996) 215-254.
3. Guin-Duclosson, N., Jean-Danbias, S., Norgy, S.: 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 (2002) 782-791.
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.: 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 (2004) 477-492.
6. 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.
7. Nkambou, R.: 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 (1999).
8. Prentzas, J., Hatzilygeroudis, I., Garofalakis J.: 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 (2002) 119-128.
9. Vanlehn, K., Zhendong, N.: Bayesian student modeling, user interfaces and feedback: a sensitivity analysis. International Journal of AI in Education 12 (2001) 155-184.
10. Vassileva, J.: Dynamic Course Generation on the WWW. British Journal of Educational Technologies 29 (1998) 5-14.