Using of Intelligence Tutoring Systems For Knowledge Representation in Learning& Teaching Process

Dr.Abdual Satar Jabbar Nasir AL-Qusi

Assist .Pr

College of Managerial technical – Mosul

Abstract

Intelligent tutoring systems (ITS) are a new generation of computer systems for support and improvement of learning and teaching. The usual definition of an ITS characterizes it as a system based on some kind of knowledge which includes domain, teachers' and students' knowledge. In the research , we elaborate on the representation of knowledge in an intelligent authoring shell – which is an ITS generator system – Tutor-Expert System. Within TEx-Sys knowledge is represented through semantic networks with frames and production rules. Nodes are used for representation of domain knowledge objects, while links show relations among them. Besides, TEx-Sys supports properties and frames, as well as property inheritance and frames containing a conclusion-making mechanism.

In this research , we try to explain of practice ITS application in learning and teaching process. Because, Intelligent tutoring systems have been shown to be highly effective at increasing students' performance and motivation. For example, students using Software systems, an ITS for computers , performed equally well as students taking a theoretical & practice courses in computers, but required half as much time covering the material.

Introduction:

Intelligence is harder to define than knowledge. When researchers in the field of artificial intelligence talk about intelligence in technical systems, they usually use it to suggest that their software is more flexible, more readable, and easier to use than some other software. The structure of intelligent systems generally consists of the following components: user interface, inference engine and knowledge base with some subject matter see Figure (1), [1].

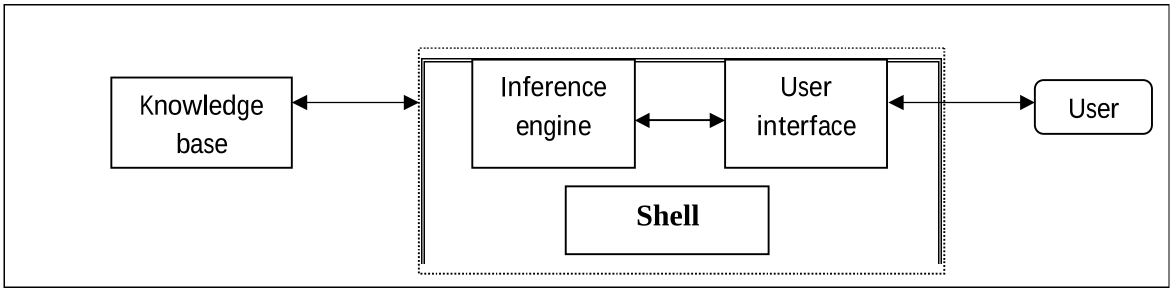

Figure [1] : Structure of Intelligence Tutoring Systems

In this research we focus on knowledge representation in intelligent tutoring shells, which are generators of particular intelligent tutoring systems (ITSs). The usual definition of an ITS characterizes it as a system based on some kind of knowledge. This "knowledge" includes:

1. Domain knowledge containing objects, relations among them, explanations, examples and exercises,
2. Teachers' knowledge as a strategy for the process of learning and teaching and
3. Students' knowledge as a model which is dynamically generated

as a result of overlaying it with teachers' knowledge [2].

Representation of all these kinds of knowledge in an ITS is usually separated from the inference and search engines that are contained in the system. Hence ITSs are knowledge based systems with the following structure:

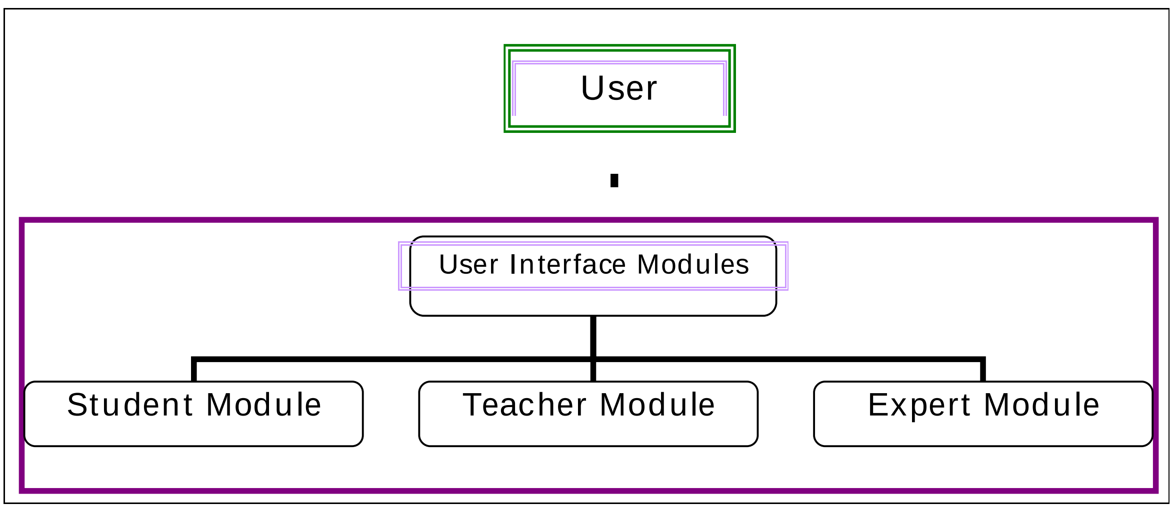

Figure [2]: Typical ITS architecture

⇒ The *domain module* is the repository for storing and structuring information; the domain base includes knowledge that a teacher wants a student to learn;

⇒ The *teacher module* resembles a human tutor; the module selects and sequences instruction and teaching styles and the learning scenario (e.g. guided free play, learning-by-doing, discovery learning, mixed-initiative dialogue);

⇒ The *student module* represents two major kinds of information: a student's personal data and predicted capability for that particular course and her/his current state of domain knowledge;

⇒ The *user interface module* to facilitate the interaction of both teachers and students with the system;

specifically it supports human teachers in domain base development, in specifying what, when and how to teach [3], and in monitoring the students' progress; obviously it should also provide a user friendly interface for students to learn the subject domain.

Types of ITSs

There are several ways of categorizing ITSs; we will concentrate on two dimensions: abstraction of the learning environment and the knowledge type of the instruction.

1- Abstraction of the learning environment

Many systems attempt to provide instruction by simulating a realistic working environment in which the student can learn

the task. There are many reasons for developing such systems, including the possible danger of training using the actual equipment and the lack of domain experts who can devote their expensive time to training novices. Therefore, a realistic simulated learning environment can reduce both the cost and the risks of training.

An example of a simulation-based ITS is the Advanced Cardiac Life Support (ACLS) Tutor [4] in which a student takes the role of team leader in providing emergency life support for patients who have had heart attacks. The system not only monitors student actions, but runs a realistic simulation of the patient's condition and maintains an environment that is reasonably faithful to the ``real life'' situation. Thus, the goal is not only to test the student's knowledge about the correct emergency procedures, but also to allow him to experience practicing those procedures in a more realistic manner than is possible in a traditional classroom.

Some systems take a less rigorous approach to representing the environment; the situations presented are similar to the real world scenarios in which the knowledge could be applied, but they are not exact simulations. Smithtown [5] takes this approach by providing a simulated setting for students to test hypotheses about economics. However, the underlying model of the environment is not an exact simulation of how the laws of economics would be applied in the real world. Another example of such a system is the Design for Manufacturing Tutor [6].At the extreme opposite of the simulation based tutors are those that teach knowledge in a de contextualized manner without attempting to simulate the real world. Many systems throughout the history of ITS research fall into this category [7]. These systems provide problems for the learner to solve

ITS that uses an analysis of expert behavior without trying to connect those problems to a real world situation and are designed to teach abstract knowledge that can be transferred to multiple problem solving situations.

Emphasis of Instruction

There is a long history of classifying instructional goals according to the type of knowledge being taught. An important early attempt at this classification is Bloom's taxonomy [8] and much recent work in categorizing knowledge has been derived from this. In addition to classifying learning goals by knowledge type, one can also examine what the student will be able to do upon completion of the ITS's lesson. This can vary from the student being able to perform a set of skills in a manner similar to an expert to understanding abstract concepts such as Newton's third law.

For ease of development, systems tend to concentrate on teaching one type of knowledge. The most common type of ITS teaches procedural skills; the goal is for students to learn how to perform a particular task. There has been substantial research in cognitive psychology about human skill acquisition, so analyzing the domain knowledge in this framework can prove beneficial to instruction. Systems that are designed according to these principles are often called cognitive tutors. The most common result of this analysis is a set of rules that are part of a run able expert model. This set of expert rules often serves double duty as a knowledge of the domain and as the pedagogical module. If a student encounters difficulty, the specific remediation required can be determined from the expert model.

An example of a ``cognitive tutor'' is SHERLOCK, which has tutorial actions associated with each state in the ``effective problem space'' [9]. Another example of an is the LISP tutor [10], which encodes expert problem solvers' actions as production rules, and attempts to determine which rules the student is having difficulty applying.

Other ITSs concentrate on teaching concepts and ``mental models'' to students. These systems encounter two main difficulties. First, a more substantial domain knowledge is needed for instruction. Second, since learning concepts and frameworks is less well understood than learning procedures, there is less cognitive theory to guide knowledge representation and the pedagogical module. For these reasons, ITSs of this type require a larger domain knowledge base and are sometimes referred to as knowledge based tutors. As a result of not having a strong model of skill acquisition or expert performance, these systems are forced to use general teaching strategies. They also place more emphasis on the communication and presentation system in order to achieve learning gains. An example of such a system is the Pedagogical Explanation Generation (PEG) system [11] which has an explanation planning component that uses a substantial domain knowledge base to construct answers to student queries in the domain of electrical circuits.

These classifications are really points along a continuum, and serve as good rules of thumb rather than a definitive method of classifying intelligent tutors. A system that does not fall into either of these categories is Coach [12], which teaches how to use UNIX mail. This is a procedural skill, and hence cognitive in nature. However, the emphasis of this system is also knowledge based and involves generating explanations and using general pedagogical tactics for generating feedback.

Generally, tutors that teach procedural skills use a cognitive task analysis of expert behavior, while tutors that teach concepts and frameworks use a larger knowledge base and place more emphasis on communication to be effective during instruction. There are exceptions to these rules, but they serve as useful guidelines for classifying ITSs.

The Student Model

As noted previously, the student model is the component of an ITS that records information about the student. This information reflects the system's belief of the learner's current knowledge state. Since only overt student actions are visible, and the ITS only has a relatively narrow channel of communication with the user, there is difficulty in obtaining an accurate representation of the student's abilities. Therefore, the model of the student may not be perfectly accurate and steps must be taken to ensure that the system's actions on the basis of this inaccurate information are not inappropriate. For example, a tutor that interferes too much with a learner who is performing satisfactorily can obviously be detrimental.

After considering the above difficulties, an obvious question concerning student models is why to have one. Simply put, the student model is necessary in order to tailor instruction to a student's idiosyncrasies and learning needs. Without this knowledge, the pedagogical component of the tutor has no basis on which to make decisions, and is forced to treat all students similarly. This is analogous to earlier efforts in CBT and CAI which did not customize instruction for individual learners.

Representation of the student model

There are many methods for representing information about the student. Two commonly used techniques are overlay models and Bayesian networks.

The standard paradigm for representing a student model is the overlay model [13] in which the student's knowledge is considered to be a subset of the expert's knowledge (Figure 3a). With this representation, an ITS presents material to the student so that his knowledge will exactly match that of the expert. The knowledge types that can be represented within an overlay student model include `topics', which correspond to elements of the domain knowledge, and production rules [14].

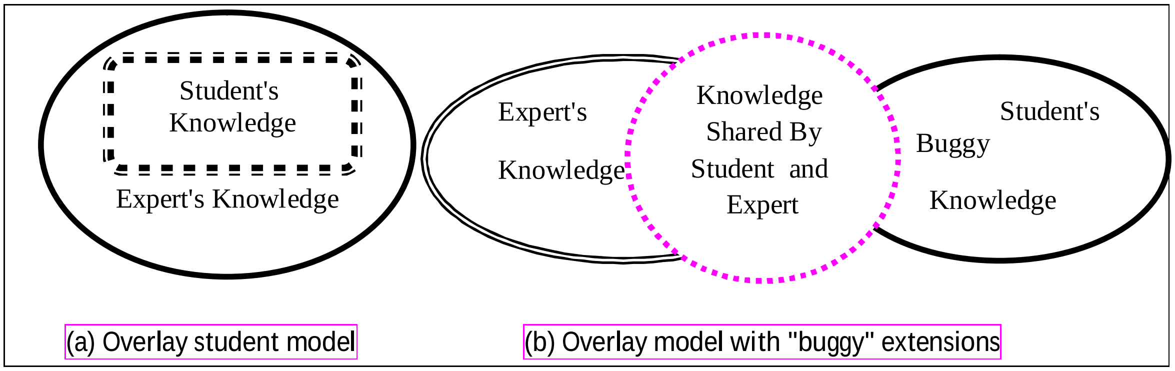

Figure [3]: Structure of student models

A drawback of this approach is that it does not acknowledge that students may have beliefs that are not part of the expert's knowledge base. For example, students frequently have misconceptions about a domain. Therefore an extension to the overlay model explicitly represents ``buggy'' knowledge that the student may have (Figure 3b) [15]. This extension allows for better remediation of student mistakes, since the fact that a student believes something that is incorrect is pedagogically significant.

Another mechanism for recording a student's knowledge is Bayesian networks [16]. These networks probabilistically reason about a student's knowledge state based on his interactions with the tutor. Each node in the network has a probability indicating the likelihood of the student ``knowing'' that piece of knowledge.

Components of Intelligent Tutoring Systems

Intelligent tutoring systems may outwardly appear to be monolithic systems, but for the purposes of conceptualization and design, it is often easier to think about them as consisting of several interdependent components. Previous research by Woolf [17] has identified four major components: the student model, the pedagogical module, the domain knowledge module, and the communication module. We have identified a fifth component, the expert model. Woolf includes this component as part of the domain knowledge, but we feel that it is a separate entity. Figure 1 provides a view of the interactions between the modules.

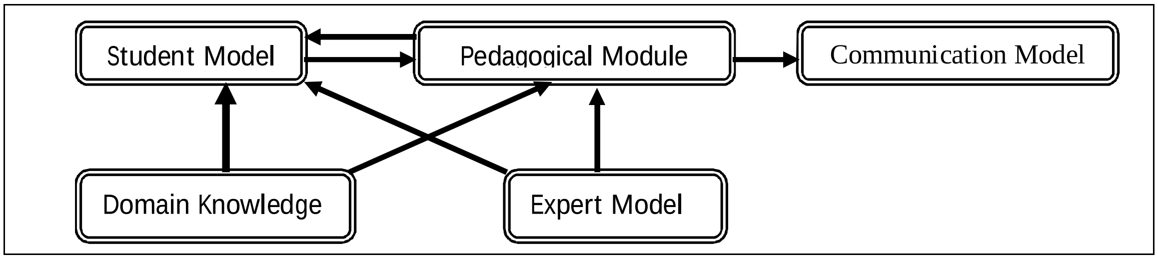

Figure [4]: Interactions of components in an intelligent tutoring system

1-Student Model

The student model stores information that is specific to each individual learner. At a minimum, such a model tracks how well a student is performing on the material being taught. A possible addition to this is to also record misconceptions. Since the purpose of the student model is to provide data for the pedagogical module of the system, all of the information gathered should be able to be used by the tutor.

2-Pedagogical Module

This component provides a model of the teaching process. For example, information about when to review, when to present a new topic, and which topic to present is controlled by the pedagogical module. As mentioned earlier, the student model is used as input to this component, so the pedagogical decisions reflect the differing needs of each student.

3-Domain Knowledge

This component contains information the tutor is teaching, and is the most important since without it, there would be nothing to teach the student. Generally, it requires significant knowledge engineering to represent a domain so that other parts of the tutor can access it. One related research issue is how to represent knowledge so that it easily scales up to larger domains. Another open question is how to represent domain knowledge other than facts and procedures, such as concepts and mental models.

4-Communications Module

Interactions with the learner, including the dialogue and the screen layouts, are controlled by this component. How should the material be presented to the student in the most effective way? This component has not been researched as much as the others, but there has been some promising work in this area [12].

5-Expert Model

The expert model is similar to the domain knowledge in that it must contain the information being taught to the learner. However, it is more than just a representation of the data; it is a model of how someone skilled in a particular domain represents the knowledge. Most commonly, this takes the form of a runnable expert model, i.e. one that is capable of solving problems in the domain [18]. By using an expert model, the tutor can compare the learner's solution to the expert's solution, pinpointing the places where the learner had difficulties.

II. STUDENT KNOWLEDGE AND SKILLS ACQUISITION PROCESS IN TEX-SYS

TEx-Sys is structured according to the cybernetic model of systems, hence interpreting the education process as a feedback system [19]. Within the model student knowledge and skills acquisition is a guided process, with a referent value defined through goals and tasks pertaining to subject matter to be learned. We define the model of a "good student" which is based on certain

evaluation criteria according to a specified student knowledge level [2]. The control function in TEx-Sys is based on:

1-measurement and diagnostics of student knowledge,

2-determination of differences between actual student knowledge and the referent model one, and ,

- ✓ and "dialogues of divided initiatives" [21];

3-evaluation of student knowledge with recommendations for future work. TEx-Sys is structured**.**

- ✓ *Login*: legalization of work on the system;
- ✓ *T-Expert*: building the base of freely chosen domain knowledge (for teachers, and in particular cases for students, too);
- ✓ *Learning and Teaching* of freely chosen domain knowledge (for students);
- ✓ *Exploring*: access to knowledge in the knowledge base; effectively this is a subsystem with a limited set of predefined sentences (nine questions and two statements) which the user is not allowed to freely form; there also exists a dictionary containing object names and properties as well as object attribute

  names and values;
- ✓ *Examination*: evaluation of a student's knowledge within a teaching scenario, according to Piaget's theory of "guided free play" [20] and combinations of scenarios of teaching by "articulated experts"
- ✓ Evaluation: access to the achieved results of learning and teaching (for teachers);
- ✓ Courseware: installation of lessons or even complete curricula of a subject matter (for students).into the following modules, see Fig.5:

Figure5:Interaction of TEx-Sys module in student knowledge and skills acquisition process

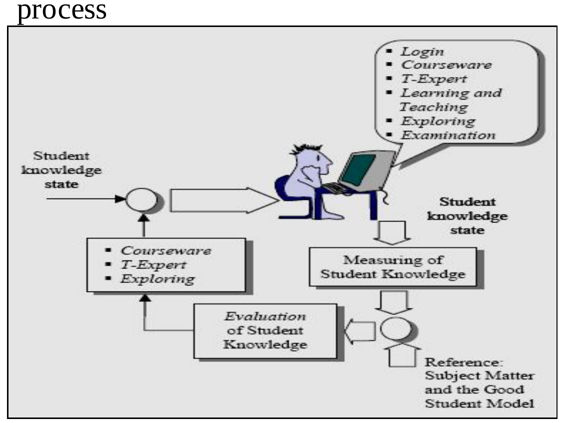

Figure [5] :Interaction of TEx-Sys module in student knowledge and skills acquisition process

III.KNOWLEDGE REPRESENTATION IN TEX-SYS

Within TEx-Sys knowledge is represented by semantic networks with frames (specifically in T-Expert, as well as in knowledge learning and teaching, exploring, examining and courseware modules) and production rules (in the examining module). The basic components of TEx-Sys semantic networks are nodes and links (see Fig. 6).

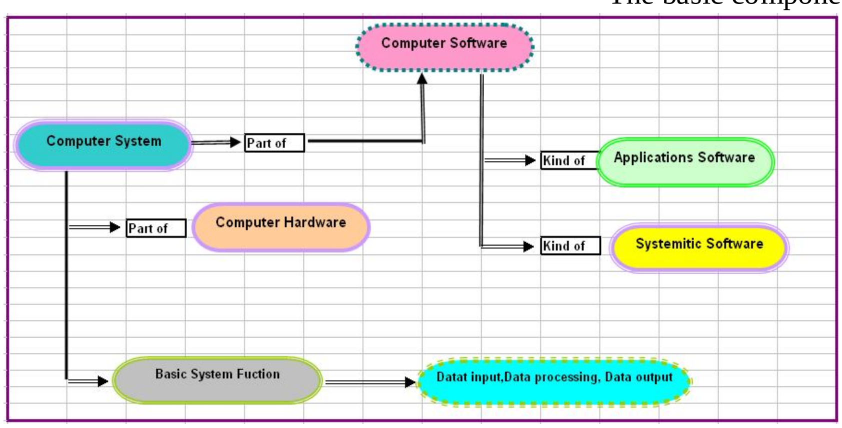

Figure [6]: A simple semantic network with frames in the TEx-Sys

Nodes are used for presentation of domain knowledge objects, while links show relations between pairs of objects. Beside nodes and links, the system supports properties and frames (attributes and respective values), along with property inheritance. The system relies heavily on modern supporting technologies, such as multimedia, with the following structure attributes: picture, animation, slides, URL addresses and hyper textual descriptions.

In the following we especially consider knowledge representation for :

1. domain knowledge and
2. student knowledge and
3. skills acquisition.

A. *Domain Knowledge Representation*

The domain knowledge base is implemented according to the entity-relationship model for databases and is built upon the following five objects, see Fig.7

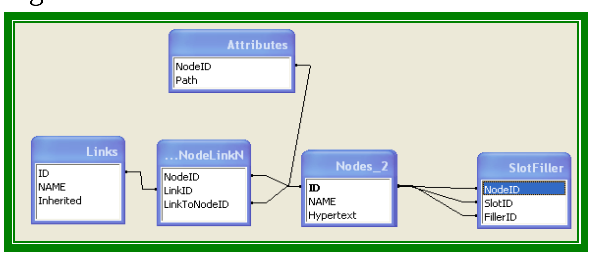

Figure [7]: Entity relationship model of domain knowledge base

B. *Student Knowledge Representation*

The formalization of student knowledge in TEx-Sys is based on the same syntax and semantics for nodes and links, and harmonized with knowledge representation using semantic networks with frames. Student knowledge is developed by overlaying it with the teacher one,

including misconceptions and missing conceptions [2],using the following three knowledge bases, see Fig.8

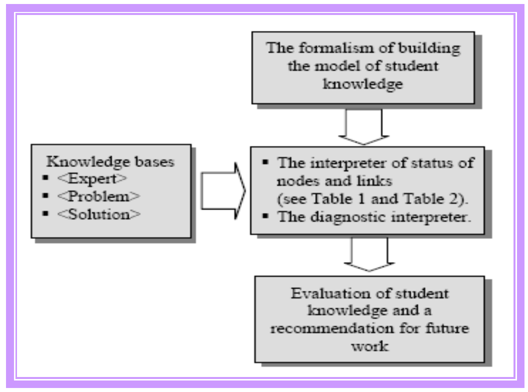

Figure [8]: Process of student knowledge evaluation

- ⇒ □□Expert knowledge base for chosen domain knowledge, which is denoted with <Expert>,
- ⇒ □□Problem knowledge base, denoted with <Problem>,
- ⇒ □□Solution knowledge base, denoted with <Solution>.

Problem generation in TEx-Sys allows the evaluation of student knowledge of a chosen domain. The examination module envisages three problem kinds:

- ⇒ □□*Problem1*: all links are deleted and the student is asked to add them to the knowledge base <Problem>;
- ⇒ □□*Problem 2*: the student is asked to complete the knowledge base <Problem>, which is randomly generated; the number of generated nodes is limited to the range between 30% and 70% of the total

<Problem>; the student is asked to indicate and delete them, and fill in thbase

During problem solving the student can perform the following operations on the nodes: delete_node, add_missing_node and add_new_node (not already present in the base). The following operations can be performed on links: add_new_link (with newly entered nodes), delete_correct_link, delete_incorrect_link, add_correct_link, add_incorrect_link and add_missing_link.

By overlaying knowledge in the bases <Expert>, <Problem> and <Solution> the level of student

number of nodes;

Problem 3: beside (no less than 50%) missing nodes incorrect links are introduced in the base

knowledge (knowledge state in Fig. 2) is determined.

This process also determines the status of nodes and links in the base <Solution> (see Table 1. and Table 2.).

Subsequently, the diagnostic interpreter evaluates the student's knowledge, see Fig. 5. Both interpreters form the base for the overall student's evaluation.

Table [1]: Overlay Nodes Status

Status of Nodes	<Expert>	<Problem>	<Solution>	Overlay
Added Node	1	0	1	$(E \cap S) \setminus P$
Missing Node	1	0	0	$E \setminus (P \cup S)$
Deleted Node	1	1	0	$(E \cap P) \setminus S$
Node Without Change	1	1	1	$E \cap P \cap S$
New Node	0	0	1	$S \setminus (E \cup P)$

Table [2]: Overlay Links Status

Status of Links	<Expert>	<Problem>	<Solution>	Overlay
Deleted Incorrect Link	0	1	0	P\(EUS)
Add Correct Link	1	0	1	(E∩S)\P
Correct Given Link	1	1	1	E∩P∩S
Missing Link	1	0	0	E\(PUS)
Incorrect Link Given	0	1	1	(P∩S)\E
Correct Link Deleted	1	1	0	(E∩P)\S
Incorrect Link Added	0	0	1	S\(EUP)
New Link	0	0	1	S\(EUP)

Evaluation of a student's knowledge is enabled by particularly devised point criteria and the production rules based expert system MARK, containing 170 production rules, for generation of her/his knowledge grade.

The point criteria provide quantitative and qualitative descriptions of student activity in the problem solving process. MARK eventually offers the students a description of their success, explanations and recommendations for future work.

IV. CONCLUSION

This research describes domain and student knowledge representation in intelligent tutoring systems generated by the intelligent authoring shell TEx-Sys. TEx-Sys is accommodated to both students and teachers with the purpose of improving the process of learning and teaching in a freely chosen domain knowledge. The knowledge is represented through semantic networks with frames and production rules. Nodes in the semantic network express knowledge on subject matter objects (i.e. facts and terms) in the chosen knowledge base. Links express the process of thinking about relations among nodes of the base. The semantic network is implemented according to the entity-relationship model for databases.

TEx-Sys has successfully been used for some time in the educational process at the Faculty of Natural Sciences, Mathematics and Education in Split, where a number of knowledge bases for different undergraduate courses have been developed. The system turned out to be an appropriate tool for CAI (Computer Aided Instruction), such that even students in the education curriculum, as non-expert users, use it in their course building assignments

References

[1] F. Chabris 1987. Artificial Intelligence & Turbo PASCAL, Multiscience Press, Inc. 1987.

[2] S. Stankov, V. Glavinic, M. Rosic " On Knowledge Representation in an Intelligent Tutoring System, project , Ministry for Science and Technology of the Republic of Croatia,2003.

[3] M. Urban-Lurain .2000:" Intelligent Tutoring Systems": An Historic Review in the Context of the Development ofArtificial Intelligence and Educational Psychology, http://www.cse.msu.edu/~urban/ITS.htm.

[4] Eliot, C. and Woolf, B. 1995. "An Adaptive Student Centered Curriculum for an Intelligent Training System". User Modeling and User-Adapted Interaction, 5, 1, pp. 67-86.

[5] Shute, V., R. Glaser, and K. Raghaven. 1989. "Inference and Discovery in an Exploratory Laboratory". Learning and Individual Differences, Ackerman, P., R. Sterberg, and R. Glaser, eds., pp. 279-326.

[6] Haugsjaa, E. and Woolf, B. 1996. "3D Visualization Tools in a Design for Manufacturing Tutor". In Proceedings of Educational Multimedia and Hypermedia, Boston, Mass.

[7] Shute, V. 1995. Smart: Student Modeling Approach for Responsive Tutoring. User Modeling and User-Adapted Interaction, 5, 1, pp. 1-44.

[14] Anderson, J. 1993." Rules of the Mind". New Jersey, Lawrence Erlbaum Associates

Report, AI Lab Memo 406, MIT.

[8] Bloom, B. (ed.) 1956. "Taxonomy of Educational Objectives". New York, NY, Mackay Publishing.

[9] Lajoie, S.P. 1993. "Computer Environments as Cognitive Tools for Enhancing Learning. Computers as Cognitive Tools," Lajoie, S. and Derry, S., eds., Lawrence Erlbaum Associates, NJ, pp. 261-288.

[10] Anderson, J. and Reiser, B. 1985." The LISP Tutor". *Byte*, 10, 4, pp. 159-175.

[11] Suthers, D. 1992." Answering Student Queries": Functionality and Mechanisms. In Proceedings of the 2nd International Conference on ITS (ITS-92), Montreal, June.

[12] Winkels, R., J. Sandberg, and J. Breuker. 1990. "The Coach. In Eurohelp": Developing Intelligent Help Systems, Breuker, J., ed., EC: Copenhagen, pp. 119-146.

[13] Carr, B. and Goldstein, I. 1977. "Overlays: a Theory of Modeling for Computer-aided Instruction", Technical

http://www.ce.gatech.edu/research/projects/computer/Epitome/epitome.html

[15] Holt, P., Dubs, S., Jones, M., and Greer, J. 1993. "The State of Student Modeling". In Student Modeling: The Key to Individualized Knowledge-Based Instruction, Greer, J. and McCalla, G., eds., Springer-Verlag, New York.

[16] Martin, J. & VanLehn, K. (in press)," Student assessment using Bayesian nets". International Journal of Human-Computer Studies, Vol. 42.

[17] Woolf, B. 1992. AI in Education. "Encyclopedia of Artificial Intelligence", Shapiro, S., ed., John Wiley & Sons, Inc., New York, pp. 434-444.

[18] Clancey, W. and Letsinger, R. 1981." Tutoring Rules for Guiding a Case Method Dialog". In Proceedings of the Sixth IJCAI, Vancouver, B.C., Morgan-Kaufmann, San Mateo, California, pp. 829-835.

[19] Principia Cybernetica Web, http://pespmc1.vub.ac.be/

[20] R. Sugerman: "'What's new, teacher?' Ask the computer", IEEE Spectrum, September, 1978, pp. 44-49

[21] N. C. Baker: Engineering Platform for Intelligent Tutors and Multimedia Experiences (Project EPITOME), School of Civil Engineering, Georgia Institute of Technology, 1999,