Annales UMCS Informatica AI 7 (2007) 127-141

Annales UMCS
Informatica
Lublin-Polonia
Sectio AI

http://www.annales.umcs.lublin.pl/

Model of rule engines applied in the intelligent systems

Barbara Gocłowska*, Monika Puchacz

Institute of Computer Science, Maria-Curie Sklodowska University,
pl. M. Curie-Sklodowskiej 1, 20-031 Lublin, Poland

Motto: "A really interactive tutor should act not only in an interactive but also active mode. He should not sleep between two successive answers in reply to the questions but should be able to monitor the work of students and react to their errors continuously" [1].

Abstract

The aim of the present paper is the analysis of rule systems used in the implemented Intelligent Tutorial Systems using various technologies. The ready – made shells are used for majority of the systems being discussed herein; the CTAT shell is the most popular type. The rule programming languages i.e. Prolog, SmallTack and others have been used as the basis in the remaining cases. Those tools are necessary to organize the control – the decision whose production will be used as the next one must be made by the system in every stage of the learning process.

Using the dynamic planning, the Agent controlling the individual student path may decide when and how to teach in the next step. Therefore the choice of inference strategy and the selection of the tools to be used for rules control are important, which has been described in the present paper.

1. Expert systems in tutorial applications

The term „expert system" is applied for the program which enables the drawing of conclusions and decision making on the basis of detailed knowledge. The functioning of that system is similar to the human thinking process. The decisions to be made in the scope of Intelligent Tutorial Systems may be associated with the learning methods as well as with the choice of content.

The generated results are used in the expert systems [2], as well as the criticizing systems [3] and those making the decisions without human control [4]. The latter will be the main focus of the present paper. The knowledge acquisition module is one of the most important modules in an expert system. The purpose of the knowledge acquisition module is to ensure the system development as a result of the inflow of new information and the modification

*Corresponding author: *e-mail address*: gocbar@gmail.com

and supplementation of existing knowledge. The latter is obtained from the system user or from an expert maintaining its correctness. At least three knowledge bases are required for the Intelligent Tutorial Systems i.e. the base containing the subject knowledge to be introduced by the course author (other solutions are also possible, e.g. downloading of knowledge from websites [5]), the base containing the knowledge about the system user progress and the rules base.

The meta – knowledge in a typical Intelligent Tutorial System is used in the Planner module consisting of three modules: Curriculum Planner, Discourse Planner and Turn Planner. The meta – knowledge performs the functions of the component determining the further behaviours of the system including the planning of its reactions to the statements of student.

The Planner generates a series of sequences required to form the successive answers of the system in reply to the student's statements. Such a solution is equivalent to organization of the theory associated with individual case in the form of series of sentences to be used in precisely determined cases. The decision tables are used by the Planner to make the decision regarding the choice of successive sequences and their order in the learning process. In the case of large sized decision table [6], the designing of rules set is required to provide the basis for generation of successive sequences to present the content in the order conforming with that generated in the course of problem solving by the Problem Solver.

The plan of learning process for every problem is created after the arrangement of individual sequences in proper order after determination of their choice mechanism. Such plan consists of single teaching policy or set of policies [7]. A policy is allocated to each principal plan category [8]. The confirmation of the answer correctness may occur in various forms owing to complexity of correct answer. The answer correctness depends on how many partial correct answers are required for the question to be recognized as credited. For instance, in the case of the question asked by the authors [9] i.e. „What are the determinants Xa?", the answer may be partially correct when any of the specified determinants is correct and another is wrong or when the both determinants have been specified correctly (then positive confirmation is generated) or completely wrong when the both determinants have been specified incorrectly (then negative confirmation is generated).

Therefore the dialog is arranged in accordance with cyclically repeated actions:

1. Ask the question.
2. Evaluate the answer of the student.
3. Confirm the answer of the student.

In the case of an incorrect answer, generate a prompt or answer the question correctly [10].

If the prompt has been generated, the next question will be asked as the repetition of the last question. If the student's answer was correct or He/She received the correct answer in the form of system answer, Planner proceeds to the next question in the learning process. Having completed the whole plan associated with the learning process, Planner proceeds to the next phase of problem solving process. However, it is not easy in the case described above, because it is associated with fuzzy knowledge [11,12].

During the whole time of system operation, the inference engine directly operates on the knowledge base in order to find a solution. The content of knowledge base may be also modified in the form of new elements added by the inference engine. The structure of inference engine depends on the knowledge representation established for the specific expert system. Except for the facts, the set of inference rules necessary for problem solution, is the core of the knowledge base in an expert system.

2. Inference rules

Generally, the inference rules are presented in the following manner:

IF prerequisite THEN conclusion.

The prerequisite included in such expressions may encompass several statements connected by means of logic functors i.e. conjunction (AND) and alternative (OR). The prerequisites and conclusions are considered as the logical values; therefore if the prerequisites are true, the rule they create is also true. So called developed form of rules with an additional „branch" to be completed in case of non-fulfilment of a prerequisite, is used in many systems [e.g. ITS]. A/m form may be presented in the following manner:

IF prerequisite THEN conclusion 1 ELSE conclusion 2.

Among other,s the rule engine used in LISE application [13] consists of the rules in such form, for example:

```
IF (3 of 5 answers are wrong) THEN
   IF (critical question occurred) THEN
      IF (student's answer was correct) THEN go back to the begin of current
      chapter
   ELSE call up chatbot
   ELSE next.
```

Except for its developed form, the rule may include the embedded rules.

Assuming the following symbols:

a – 3 of 5 answers are wrong

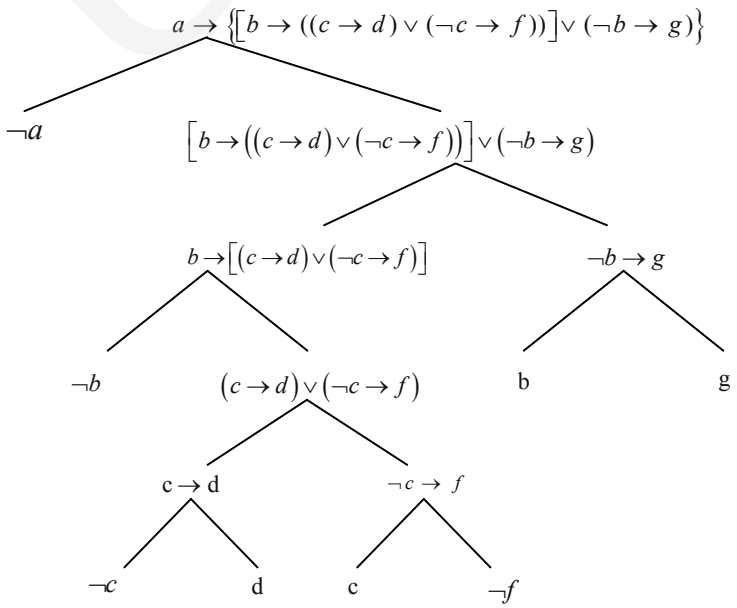
b – critical question occurred
c – student's answer was correct
d – go back to the begin of current chapter
e – student's answer was wrong = $\neg c$
f – call up chatbot
g – next

The following formula of the rule by means of Boolean operators is obtained:

$$a \to \left\{\left[b \to \left((c \to d) \vee (\neg c \to f)\right)\right] \vee (\neg b \to g)\right\}.$$

That expression is subject to cascade expansion when each successive condition is added. For instance, when except for a/m questions, any questions unmasking (or non- unmasking) the existence of preconception are encountered (or not encountered) by the student, NETS module for programmed learning should be called up. However, due to legibility of the semantic tree, the semantic tree presented below has been limited to the scope of the formula presented above.

In accordance with the set of realized rules presented as an example, a few or more than ten of events scenarios may be realised: a successive chapter of the handbook is made accessible by the system generating the dialog with chatbot, making the programmed questions accessible to enable the disclosure of natural knowledge [14] and to help in proper understanding of the content of learning process.



The simplest expert system resolving the existing problem, will try to analyse every version of the situation. First, the attempt will be made to check how many questions in the session have been answered by the user correctly. If the system finds at least three correct answers (conditions for that case described by the first rule), each of remaining conditions has to be checked by the system (i.e. if critical question occurred) in order to give correct advice, etc.

If there are no further conditions which are important in the existing situation, the system proceeds to find the answer to the next questions.

3. Knowledge processing in Intelligent Tutorial Systems

Although the major part of expert system has been designed on the basis of rule knowledge presentation and their functions are based upon the logic inference principle, there are also the systems functioning in accordance with the principle of information association, designed on the basis of neural networks. The processing of information contained in such network forming the structure of the neural connections is an alternative solution to the sequential data processed by means of actually existing computers.

Processing of knowledge consists in the determination of mechanisms and the sequence of operators transforming the initial state into the final state. The advantage of such mechanisms and searching methods consists in easy determination of the tasks to be performed in order to achieve expected results. Only the determination of the set of states for searching space for specific problem, the set of operators transforming those states as well as the initial state and the final state is required. However, the problem solution is determined by searching within the whole set of all possible states i.e. the searching space.

The searching space is defined as [15] a set containing all states possible which can be achieved by the process of searching for the solution of specific problem. A graph is used for the space formalization. The nodes of that graph represent the system states and its branches represent the productions changing the state of the system. The productions performed in the application environment enable the transformation from one state to another. A production transferring the initial condition of the edge into the state indicated by its another end exists between each pair of the states in the graphs which are connected together by means of the branch. However, the executive system is responsible for the process of transformation from the initial state into the final state. In every phase of tutorial process, the executive system must decide which of the productions fulfilling the preliminary conditions will be applied as the next one [5].

Owing to the diversified level of complexity of the graph representing the searching space, several strategies and procedures have been created to enable

its content searching in order to find the solution, according to the needs and system capabilities.

Two main solution search strategies i.e. top – down and bottom – up [2] strategy are mainly used in the expert systems. The bottom – up inference consists in the proceeding from the prerequisite to the conclusion. From the individual facts, the next facts at the successive levels of the graph are obtained until the conclusion is reached. This type of strategy is used in majority of Intelligent Tutorial Systems.

Any Intelligent Tutoring System dynamically decides what, when and how will be learned in the next step. The module responsible for creation of learning plans for a student, is also responsible for the monitoring of performance of generated plans and for their adaptation in the course of learning to the needs of individual student.

4. Representation of knowledge and mechanisms applied in ITS

In the majority of Intelligent Tutorial Systems, the interaction of the system with the user is commenced in the form of presentation of the content to be acquired by the student in the framework of single session or course. That content is presented in the form of semantic network with the nodes representing the basic units of the knowledge being presented (units, sessions or modules) and with branches representing the relations between them. If a/m relations are precisely determined, the inference algorithm is able to operate in accordance with the successive graph connections. Otherwise, the Boolean rules are applied. The system should be provided with the possibility to check the student's knowledge in each phase of learning process in order to search an optimal path for the presentation of knowledge adequate to His/ Her requirements and intellectual powers as well as in order to generate proper questions and prompts [16].

The operation of an Intelligent Tutoring System model is based on the question rules applied to the individual units of presented knowledge or on the rule engine. Its form can be diversified and is determined by the specific system operation features, structure of the content being presented in that system, the factors affecting the users evaluation and general conditions required for the achievement of satisfactory level of the knowledge acquired by the student. The rules determine the situations when the errors are identified and the prompts are generated by the system reversing the student to the preceding modules in order to improve the results and transferring the next sections of knowledge or additional modules [14].

Depending on ITS methodology, simple rules are used to control various system behaviours or neural networks in the expanded Intelligent Tutoring

Systems. The results obtained in the preceding phases of learning process i.e. output weights for one network layer determine the level of knowledge presented in the next modules; therefore they are the inputs for the next layer of the same network [17].

However, irrespective of applied methodology, all possible solutions of a problem are represented by means of solutions graph. In the case of learning process, the first unit or module presenting the course content is the initial state of that graph and the exam is its target state. The intermediate states represent individual units or modules being the potential states of learning process. The tutor or agents controlling the path and the learning process in the case of an individual student is the executive system in the case of Intelligent Tutoring Systems.

5. Graphs model for Intelligent Tutorial System

The semantic network is used for the presentation of the sessions or courses structure. The network consists of the graph with the nodes representing the objects and the branches representing the relations between them. The nodes and the branches may be associated with certain parameters determining the rank of any relation or object in comparison with other relations and objects in the network or for example the probability of a relation between determined objects. The following parameters describing the relations can be used: the power of relation between the objects being considered or variability of the relation versus time or depending on other factors and conditions [2].

The basic unit is the node representing the single session of a course. Various relations and links occur between the nodes in the graph. Therefore it may happen that the review of any sessions represented by the node is impossible until all preceding nodes are credited. The nodes may be interrelated in diversified degrees: some nodes may be needed for other ones, very important for other ones, used as the help for other ones or only associated with them. Furthermore, the weights can be assigned to the arcs connecting the individual nodes in the graph. The weights determine certain semantic relations between the sessions being represented by those nodes.

Owing to various values of the weights, the system must meet various criteria between the nodes in the course of selection of the part of proper content resource for the student.

The sessions thematically connected together create the chapters being represented by the logical sets of interconnected nodes. Refer to the figure included below originating from the study [16] and illustrating a structure as an example.

The nodes provided with the numbers are the symbols of basic units contained in the course content, which may be arranged in groups. The arrows represent certain relations between individual units. For example, considering P4 (conception corresponding to an unit), node P1 must be credited by the student in order to be able to proceed to node Tk. It is also possible to reach that node using the path P3→P5 or another.

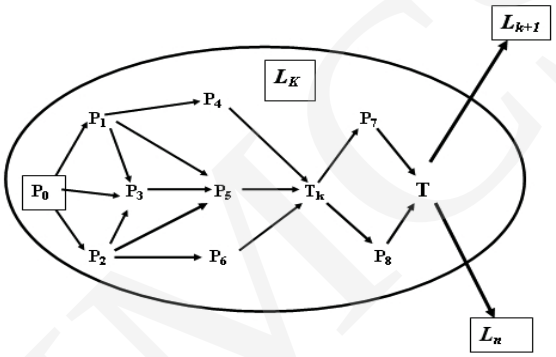


Fig. 2. Schematic structure of typical lesson module L_k

The Content Structure included in the Smart Tutor system is created according to the similar scheme. It is responsible mainly for the selection of its whole content associated with course map for a student and designed according to His/Her individual needs. The Question Bank is used as the component closely cooperating with the Content Structure.

The Question Bank is used as the questions storage to generate the tests and exams for corresponding courses. The generated tests and their results constitute the basis for the system enabling the control and evaluation of learning process for individual student and enabling the acquisition of information regarding His/Her achievements and advances in Student Model.

The structure of Question Bank is closely associated with Content Structure. In majority of the Intelligent Tutorial Systems, the nodes being the part of Content Structure are associated with the sets of questions referring to the materials represented by those nodes. The node may be also not provided by any of such relations. The connection of any node with the sets of questions or a lack of such connection, depends on its content and on specific features of the course and particularly on its parts which will be checked and evaluated, if the understanding of the whole problem is not affected by the degree of familiarization with those parts. It may also happen that the same question from Question Bank belongs to two different sets of questions associated with

different nodes. Therefore the degree of understanding of the content can be enforced for the both nodes.

Detailed information referring to individual questions is stored in special records. Except for connections with the content of corresponding nodes, such fields as "weight", "difficulty level", "critical question", "preconceptions and misconceptions" and "time" which determine the parameters of any question which are crucial for its final evaluation by the system, are also included. The corresponding values for those fields are established by the instructors of the courses by means of CES, defining the weight of the question, the degree of its difficulty etc. In case of any question referring to several nodes and included in several sets, the individual results are stored in the record representing that question. Owing to such a method of the information acquisition, it is possible to create the learning process history for the individual student, its storage in Student Model as well as to perform corresponding analyses and to make corresponding decisions by the system.

The individual learning program (occurring in some courses, particularly those implemented in the remote learning platforms [13,14]), includes the content of the courses which may be or have to be acquired by the student in order to complete the course with positive results. Further to the established representation of the content by means of the graph, the individual learning program encompasses the set of nodes to be credited by the student. They are arranged in the schedule in accordance with the relations defined between them. Generally, the schedule of individual learning program is associated with the linear representation in the form of required nodes to be passed and corresponding results obtained from tests.

6. Model of engines

The content flow control in majority of intelligent systems [17] was associated with individual behaviours of the user. The Test Generator module has been used by the designers of the SmartTutor [Soul] system in order to generate the control rules. Its role consists mainly in checking of the number of questions included in the test, adaptation of their difficulty level, time to be used for their solution and percentage distribution of the questions referring to different parts of materials in the whole text for the student being evaluated. A/m values are selected by the Test Generator mainly on the basis of the profile of the student, His/Her knowledge and the marks obtained from the previous tests, the aim of testing, considering the fact if it is only the test used for knowledge checking or an exam carried out by the instructor and having the influence on the final mark.

The set of rules constituting the basis for the operation of that module is created depending on the needs and specific features of the test. An example of such set defining the principles required to create the test to be used for checking the knowledge to be acquired from individual session, chapter or the whole course, has been presented below.

Creating the test to be used for checking the knowledge to be acquired from the whole course, Test Generator must define what percentage of the total number of the questions in the test will be represented by the questions checking the knowledge to be acquired from individual chapters. When the purpose of the test being generated is to check the familiarization with individual chapters, it is necessary to determine the percentage of the questions checking the knowledge to be acquired from individual sessions included in a specific chapter.

Using the rule base with the structure described above, the progress of individual student may be checked, His/Her errors may be analysed and the next tests may designed by the Test Generator in the manner ensuring the student's concentration on the units where His/Her knowledge is the poorest. Therefore the attention of the user is attracted to the gaps in His/Her knowledge without unnecessary waste of time to repeat the satisfactorily acquired knowledge. Obviously all intermediate results are recorded throughout the learning process history in the corresponding Student Model records referring to the individual student.

Similarly, the progress and results achieved by the individual student can be also analysed by the Advisor using His/Her learning process history in order to create (instead of tests) the individual instructions, advice and task proposals intended to improve the results achieved by the student. The main principle of the Advisor is "if the test encompasses the questions from various nodes included in Content Structure, it will result in a kind of solutions structure including all nodes being analysed". In the case of any nodes included in that structure and marked as "not credited", the student will be informed by the Advisor about the subjects which mostly contributed to the negative result and which should be particularly considered in the course of repeated analysis.

Several rules have been presented below in order to define the functioning of the Advisor when using individual representations of errors and corresponding descriptions in the records included in Student Model.

IF the structure is a single node THEN the Advisor reports to emphasize the node.

IF the structure is linear THEN the Advisor reports to emphasize all nodes from the list simultaneously.

IF the structure is flat THEN the Advisor reports to concentrate on the node – root first and to emphasize remaining nodes thereafter.

IF the structure is a tree THEN the Advisor reports to analyse the source node first and the next nodes thereafter.

Owing to such representation of the information, the Advisor will be able to organize the knowledge on student's errors in a more efficient manner and to improve the functioning of the whole Expert Model.

The authors of Intelligent Tutoring Systems arranged the conditional statements in groups in the manner enabling the use of commonly repeated conditions. For instance the designers of Andes system (Andes – Authoring Module) have used the knowledge basis generating all equations required to describe any physical problem presented by the system. That basis encompasses 600 rules classified into the two types i.e.: goal rules (used by the system to compile the knowledge path) as well as physics-knowledge rules (used by the system to supply the physical knowledge). Also the graphs containing all possible solution paths are produced by the knowledge basis. Those solution paths in combination with the equations are used to generate the prompts, to support the student's work and to generate feedback opinion of the system. The solution graph is used for the next step planning. As the solution is generated by the knowledge basis for all possible paths, the student is not enforced to proceed according to any particular path to achieve His/Her solution.

The Andes – Expertise module contains the knowledge entered by the teacher and to be used for the presentation to the students by means of semantic networks, procedural representation and production systems. The script-frames enabling rules writing in a declarative manner are also applied.

In the case of simple methodology (constant number of test questions, lack of any diversification in the degree of difficulty and importance or the condition of correct answers to all quiz questions), conditional statements are not complex and their generality degree is sufficient to enable their common use for the content flow control within various chapters, courses etc.

The situation becomes more complex in the result of weights introduction resulting in the three or four times increased number of rules. The type of condition:

$$\text{IF } \sum m > \tfrac{1}{2} \sum n \ \wedge \ Ts < Tm \wedge S1 \text{ THEN } Ch(n+1)$$

has been replaced by the condition considering the weights which can be expressed in the following manner:

$$\text{IF } \sum wn > k \sum wm \ \wedge \ Ts < Tm \wedge S1 \text{ THEN } Ch(n+1),$$

where – wn –means the weights for individual conceptions associated with the corresponding questions and the correct answers to those questions,

$\sum wm$ – sum of the products of weights and the number of questions included in the quiz

Ts – time required for quiz solution by the user,

T2 – time of quiz solution recorded in the StudentModel,

S1 – the quiz is resolved by the user the first time,
Sn – the quiz is resolved by the user the next time,
Ch(n+1) – the next successive chapter of the course.

On the basis of analysis of the papers written by many authors of Intelligent Tutoring Systems, such mathematical model should be expected on the basis of applied strategy. The differences are associated with the use of various factors of proportionality with the values of 0.5 up to 1. However, the lack of standards is problematic when determining the weight of the question, in case of simple executive systems limited to the control for the chapters, units and tests. The execution of graph and matrix analysis seems to be the simplest solution [previous didactic publications] for a specific course, the calculation of average functionality of conceptions and proper selection of an adequate scale of weights. E.g. average functionality of conceptions for the physical conceptions matrix presented below amounts 3.75. In the case of functionality scatter between 0 and 14, as in the mentioned case, the linear establishment of weights in the scale form -1 to 1 would give the result presented in Table 1. The form of the condition presented above could be maintained.

Table 1. Assignment of weights to conceptions (and to questions indirectly associated with them) with determined weight

Weight	- 1	- 0.75	- 0.5	- 0.25	0	0.25	0.5	0.75	1
Functionality scope	<0-2)	<2-3>	(3-5)	<5-6>	(6-8)	<8-9>	(9-11)	<11-12>	(12-14>

A different approach towards the determination of weights is presented by Kukla [8] assigning the weight to teaching strategies. Therefore it is possible to determine the status of the learning system which may result in generation of rules adequate to the status.

The status of learning system determines the model of domain (DM) and student model (SM): $E=DM\times SM$. At the time "t" of didactic process, the domain knowledge is determined by the ith fragment of knowledge $C_{i,t+1}$, i.e. its content to be transferred to the student as the next one, type of this fragment of knowledge T_i ={declarative, procedural} and the manner of its presentation P_i ={multimedia, text }. Thus $WD_t=C_{i,t+1}\times T_i\times P_i$. The model for the kth student is defined by His/Her intellectual predispositions I_k, advance degree $Z_{k,t}$ = {beginner, advanced}, results of checking tests $S_{k,t}$ and individual preferences IP_k, thus $MUk,t = I_k \times Z_{k,t} \times S_{k,t} \times IP_k$.. Simultaneously it has been assumed that the intellectual predispositions and preferred strategies are the individual features of the student and are not changed in course of knowledge transfer. The value of the transfer function $W(s_i,e_{k,t})$ is equal to the strategy s_j, being

characterized by the maximal weight determined on the basis of the learning system analysis, according to the following formula:

$$W(s_i, e_{k,t}) = s_j <=> w(s_j, e_{k,t}) > w(s_l, e_{k,t}) \text{ dla } l = 1, 2, ..., n,$$

where "n" determines the number of learning strategies accessible in the system.

The situation is more complex in the case of systems performing an intelligent of student solutions and Interactive problem solving support described in the study elaborated by Brusilovski [1] or in the systems criticizing the final solution of the system user [5]. Therefore some additional factors must be considered. In that case, various solutions are proposed in literature. For instance, Leem Seop Shim created the method applied in the educational dialog systems [18]. In the framework of the method described by Leem Seop Shim, the use of Confidence Factors (CF) has been preferred to assess the probability that the problem is understood by the student. The history of answers is a historically ordered list (i.e. the last answer is included as the extreme answer on the RH end of the list) of "C" and "W" letters representing the correct and wrong answers correspondingly. In order to determine the Confidence Factors (CF) Shim [19] suggests the use of time function producing the real numbers between 0 and 1 as the evaluation values,

$$CF\left(R_1,...,R_n\right) = \frac{R_{n-k+1}W_{n-k+1} + ... + R_{n-1}W_{n-1} + R_nW_n}{W_{n-k+1} + ... + W_{n-1} + W_n}$$

where,

$CF(R_1,\ldots,R_n)$ – Confidence Factor (CF) after "n" answers,

k – answers accepted by the system i.e. in the case of more than "k" answers given by the student, the answers will be rejected by the system,

W_i – weight of the ith answer, while $W_i = 2(k + i – n) – 1$ for $n – k + 1 \le i \le n$,

R_n – the nth answers, while,

$R_n = 1.0$ if the nth answer is correct,

$R_n = 0.0$ if the nth answer is wrong,

If $n – k + 1 < 1$, then $R_{n–k+1} = 0.5$ and answer $R_{n–k+1}$ is unknown

The other authors focused on the weak points (in their opinion) of the model in order to improve and simplify the latter. Therefore the evaluation methods proposed by Shem have been modified by Gregory Hume [20] who introduced the set of rules determining the Confidence Factors (CF). At each "C" added to the list representing the answers history, the value of Confidence Factors (CF) is increased by one (except for the case when two "C" have already been included in the row). At each "C" added to the list representing the answers history, the value of Confidence Factors (CF) is reduced by one (except for the case when two "W" have already been included in the row) for each case of failure. The model has been simplified by Byung-in Cho also on the basis of the Confidence Factor Theory (CF). Byung-in Cho elaborated the methods of evaluation using

the results obtained by the predecessors to enable the evaluation of partially correct answers and to resolve the Confidence Factor distribution problem.

The precise and effective methods of student evaluation enable the development and improvement of individual approach towards the user, because they are the basis enabling the correct planning of the learning process in accordance with the intellectual powers and progress of individual users. The correctly prepared evaluation methods accurately reflect the knowledge of the student, directly affecting the functioning of the whole system being the tool simulating the private teacher repeating the issues which are not understandable for the student without considering those acquired perfectly. They may also determine the decisions regarding the change of learning plan for any student, made by the system when it appears that the actual system is improper for that student. It is possible by means of so called multiple tutoring protocols.

Conclusions

The majority of adaptive systems described in literature, are the experimental systems and their authors focus on one selected aspect, for instance on an assisting or criticizing chatbot, learning process accompanying emotions emulator, comprehensive glossary of terms, user's questions answering module etc. There are also some simple but efficient learning process aiding systems. Both groups seem to be necessary at the present level of IT development and users' demand. Therefore further examinations, trials and tests associated with the creation of ITS aiding agents will be continued. The systems using the rule engines enabling the investigation of the learning process for the user and generating the reports related to the methodology and essential errors in the scope of courses prepared by their authors are the central point of our interest.

References

[1] Brusilovsky P., *Adaptive and Intelligent Technologies for Web-based Education*. n: C. Rollinger and C. Peylo (eds.) Künstliche Intelligenz, Special Issue on Intelligent Systems and Teleteaching, 4 (1999) 19.

[2] Devedzic Vladan B., Key Issues In Next-Generation Web-Based Education. *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews*, 33(3) (2003).

[3] Schulmeister R., *Grundlagen hypermedialer Lernsysteme. Theorie-Didaktik-Design*. Oldenbourg: München, Wien 3. Aufl. 2002. Ursprüngliche erschienen bei Addison-Wesley: Bonn 1995, 2. Aufl. Oldenbourg, (1996).

[4] Kim J.H., Freedman R., Glass M., Evens M.W., *Annotation of Tutorial Dialogue Goals for Natural Language Generation*. Discourse Processes, 42(1) (2006) 37.

[5] Gorana S., Vladan D., *Building an intelligent system using modern Internet technologies*. Expert Systems with Applications, (2003) 1.

[6] Feng-Jen Yang, *Turn planning for a dialogue-based Intelligent Tutoring System*. Illinois Institute of Technology, Chicago, Illinois, (2001).

[7] Virvou M., Alepis E., *Mobile educational features in authoring tools for personalised tutoring*. Computers & Education, 44 (2005) 53, Shaaron Ainsworth, Piers Fleming, „Evaluating authoring tools for teachers as instructional designers", Computers in Human Behavior, 22 (2006) 131.
[8] Kukla E., *Zarys metodyki konstruowania strategii nauczania w multimedialnych Inteligentnych Systemach Edukacyjnych (MITS)*, in Polish.
[9] Elmi, Mohammad A., Evens M.W., *Spelling Correction Using Context*. Proceedings of COLING 98, Montreal, Canada, (1998) 360.
[10] Evens M.W., Chang R.Ch., Lee Y.H., Shim L.S., Woo Ch.W., Zbang Y., Michael J.A., Rovick A.A., *CIRCSIM-Tutor: An Intelligent Tutoring System Using Natural Language Dialogue*. (2006).
[11] Freedman R., Brandle S., Glass M., Kim J.H, Zhou Y., Evens M.W., *System demonstration Content Planning as the basis for an Intelligent Tutoring System*. Proceedings of the Ninth International Workshop on Natural Language Generation (INLG-9), Niagara-on-the-Lake, Ontario, (1998) 280.
[12] Rutkowski L., *Metody i techniki sztucznej inteligencji*. Wydawnictwo Naukowe PWN, Warszawa, (2006).
[13] Gocłowska B., Łojewski Z., *Inteligent Tutorial System LISE*. in press, Annales Informatica, IV (2007).
[14] Ford M., Billington D., *Strateges in Human Nonmonotonic Reasoning*. Computational Intelligence, 16(3) (2000).
[15] Mulawka J.J., *Systemy ekspertowe*. Wydawnictwa Naukowo-Techniczne, Warszawa, (1996) 235, in Polish.
[16] Stefanowicz B., *Sztuczna Inteligencja i systemy eksperckie*. Oficyna Wydawnicza Szkoły Głównej Handlowej, Warszawa, (2002) 68, in Polish.
[17] Gocłowska B., Łojewski Z., *Java Server Faces and Java Bean Technologies in Expert Application*. Annales Informatica, IV (2007).
[18] LeemSeop Shim, *Student Modelling for an Intelligent Tutoring System: Based on the Analysis of Human Tutoring Sessions*. Illinois Institute of Technology, (1991).
[19] Gertner Abigail S., *Critiquing: effective decision support in time-critical domains*. University of Pennsylvania, (1995) 167.
[20] Hume G., *Using Student Modelling to Determine When and How to Hint in an Intelligent Tutoring System*. Illinois Institute of Technology, (1995).