MONAP: Models, Methods and Applications

Ildar Kh. Galeev, Vadim I. Chepegin, Sergey A. Sosnovsky
Information Technologies and Technical Aids of Learning Department
Kazan State Technological University
monap@kstu.ru, chepegin@knet.ru

ABSTRACT

This paper describes MONAP authoring tools. The learning process is considered as a monitored and controlled process of learning tasks solving. The model of learning control describes in detail. Bayesian approach is used for student's knowledge identification. The adequate learning control is based on stabilization of subjective difficulty level of learning tasks. On the basis of student's answer analysis MONAP defines the learning task with optimal value of difficulty for each student. For a number of learning domains the tools providing full automation of ITS development.

Also MONAP-based intelligent tutoring system GRAD for German language grammar is described.

These authoring tools contain a modeling subsystem, which provides for convenient sequential didactic experimentation by the teacher to tune the values of parameters of the learning control subsystem.

Keywords

Authoring tools, Intelligent tutoring system, Bayesian approach, Didactic principles, Pedagogical modeling.

1. Introduction

Development of intelligent tutoring systems (ITS) is on the center of attraction for many researchers, for instance [Conati C. & VanLehn K., 1996; Kinshuk & Patel, 1997; Gertner A. et al., 1998; Yang & Akahory, 1999]. The learning process adaptive control is a key issue, which significantly influence the quality of such systems. In cases when control algorithms are based on the didactic principles, invariant for several domains, it is possible to create authoring tools for ITS design. Human teacher using such tools in specific domain have to tune parameters in accordance with specific domain and students.

The required formalization is based on the algorithmic approach to the organization of education. This approach is directed on the solving of the following basic problems:

- development of solving algorithms of concrete learning tasks and students' learning to apply this algorithms;
- development of learning adaptive control algorithms, which is realized in ITS.

Algorithms of learning tasks solving are developed by teachers on the basis of learning domain analysis and described by means of an aggregate of rules (operations): **IF** (condition), **THEN** (action).

The MONAP tools provide automation of ITS design, realizing algorithms of adaptive control of learning process in the chosen domain [Galeev I. et al., 1998; Galeev I., 1999]. Together with this basic function the MONAP provides automation of explanations subsystem design in ITS. Explanations subsystem forms an answer to the student's question "WHY?" in the form of that rule, in which the error took place, while the student was solving the learning task. Here the knowledge base is used, containing the aggregate of rules formulated by a teacher.

For different categories of students in the same domain it is expedient to use various environments, differing from each other by the didactic characteristics within the framework of the integrated learning model. In this case it is required to support the mechanism of properties inheritance, ensuring data sharing by different ITS. This requirement has been realized in the technology of formation of the ITS family as a net that essentially minimizes the difficult of forming new ITS belonging to the family.

In this paper also describes an approach to ITS designing, which proposes an alternative to ITS subject-dependent subsystems development for some domains. Such approach appreciably reduces the difficult of new ITS designing.

2. Basic notions

As a result of analyzing some learning theories, the following domain invariant principles of learning process organization have been chosen.

The learning process is considered as a monitored and controlled process of learning tasks solving. The determination of learning task properties and output of the support should be carried out on the basis of the student's knowledge identification at every learning step. The principle of passing from mastering elementary learning material to mastering the complicated one have be observed during the learning process. The transfer to the new learning material learning can be realized only when the previous material has been successfully mastered. During the learning process the subjective difficulty level for each student have be stabilized.

The chosen principles are formalized as follows. The teacher works out the algorithmic order, which describes the ways of solving learning tasks in the specified domain. The set of operations types, performed by the student while solving the named learning tasks and corresponding to the algorithmic order, is symbolized by $Y = [y_1, y_2, ... y_j, ..., y_J]$. The properties of a particular task are determined by an operation vector used to solve it: $[L_1, L_2, ..., L_j, ..., L_J]$, where L_j is the number of

y_j operations necessary for solving the task $(j = 1,2,...,J)$. All learning tasks of the learning domain may be grouped into R classes, each being characterized by a corresponding unique subset of operation types $Y_r \subset Y(\bigcup_r Y_r = Y; r = 1,2,...,R)$ used to solve learning tasks falling into the r-th class.

The basic component of the student model is the vector $P(k) = [P_1(k), P_2(k),..., P_j(k),..., P_J(k)]$, where $P_j(k)$ is the probability of using the operation y_j correctly at the k-th step of learning. The complexity of the task $S(k)$, consisting of one or several learning tasks, output at the k-th learning step is defined as: $S(k) = \sum_{j=1}^{J} L_j(k)$. The task $T(k)$ complexity degree is input as an average proportion of mistakes expected when the task is carried out:

$$T(k) = \frac{MAT(k)}{S(k)}, \tag{1}$$

where $MAT(k)$ is the expectation of the mistakes number during the task execution (the task difficulty level), i.e. $MAT(k) = \sum_{j=1}^{J} q_j(k) \times L_j(k)$, where $q_j(k)$ is the probability of the operation y_j incorrect usage at the k-th learning step.

According to the principle of subjective difficulty level stabilizing it is desirable that the following inequality should be obeyed at every learning step: $\left| T_{opt} - T(k) \right| \le \Delta T$, where T_{opt} is the optimum complexity rate; ΔT – the interval rate.

In view of the introduced definitions, the learning objective can be formulated as:

$$Z: \begin{cases} P_j(k) \ge P_{jfin}; \\ S(k) = S_{fin}; \\ t \to \min, \end{cases} \tag{2}$$

where P_{jfin} is the required value of the operation proper application probability; S_{fin}–the required task complexity in the r-th class; t - learning time.

If the required being-trained level is achieved, the learning ends successfully. Provision is made for the learning emergency stopping in the case that the learning process is not effective, thus optimizing the expenses. The model makes it possible to identify the student’s knowledge and to decide on whether to continue learning or either finish it successfully or stop it in the case of a trouble at every learning step.

At present time we are working on model extension. This provides adaptive formation of theory materials for student.

3. Student's knowledge identification

The student's knowledge identification ($P_j(k)$ value determination) is carried out in the following manner. N hypotheses $H_i\,(i=1,2,...,N)$ corresponding to N being-learned states are input for every y_j operation. For every i-th being-learned state there is a conditional probability $P(A_j/H_i)$ of correct y_j operation application in each of its L_j applications, equal to $\frac{i}{N+1}$.

The hypotheses H_i forms a full group of antithetical events, i.e. $\sum_{i=1}^{N} P_{ij} = 1$ occurs, where P_{ij} is the probability of the hypotheses H_i for the operation y_j.

At every learning step the event $B_j(k)$ take place that consist in right using j-th operation $M_j(k)$ times from $L_j(k)$ defined.

This information serves to re-calculate the distribution of the hypotheses P_{ij} probabilities using the Bayesian formula.

Every k-th learning step is characterized by a priori and a posterior distribution of probabilities of the hypotheses for being-trained states $P^0_{ij}(k)$ and $P^1_{ij}(k)$ connected by the following relationship:

$$P^1_{ij}(k) = \frac{P^0_{ij}(k) \times P(B_j(k)/H_i)}{\sum_{s=1}^{N} P^0_{sj}(k) \times P(B_j(k)/H_s)}, \tag{3}$$

where $P(B_j(k)/H_i)$ is determine by the Bernoulli theorem, i.e.

$$P(B_j(k)/H_i) = C^{M_j(k)}_{L_j(k)} \times P(A_j/H_i)^{M_j(k)} \times (1 - P(A_j/H_i))^{L_j(k) - M_j(k)}, \tag{4}$$

where $C^{M_j(k)}_{L_j(k)}$ is the number of $L_j(k)$ combinations out of $M_j(k)$.

Taking into consideration the fact that a priori distribution of the hypotheses probabilities at the k-th step coincides with a posteriori distribution at the $(k-1)$-th step, i.e. $P^0_{ij}(k) = P^1_{ij}(k-1)$ takes place, formula (3) may be re-written in the form which emphasizes its recursive character (the whole history of learning is allowed for), namely:

$$P^1_{ij}(k) = \frac{P^1_{ij}(k-1) \times P(B_j(k)/H_i)}{\sum_{s=1}^{N} P^1_{sj}(k-1) \times P(B_j(k)/H_s)}. \tag{5}$$

The probability of the operation y_j correct application at the k-th step is determined with the formula of complete probability:

$$P(A_j(k)) = \sum_{i=1}^{N} P_{ij}^{1}(k) \times P(A_j/H_i). \qquad (6)$$

The final evaluation $P_j(k)$ results from the reduction of the value calculated with formula (6) to the input being-learned states.

The mistake control and necessary explanation output at the k-th learning step provides a means for predicting the probability of the operation y_j correct application at the $(k+1)$-th learning step:

$$P_j(k+1/k) = V \times P_j(k), \text{ где } V = \frac{P_j(k)}{P_j(k-1)}. \qquad (7)$$

4. Adaptive control of learning process

The model determines the task, adequate to the student's knowledge for the next learning step, i.e. provides individual minimization of the learning period. With this objective the algorithm of stabilizing the learning task complexity rate is used. This algorithm may be represented as the following sequence of steps:

Step 1. The values of $q_j(k)$ for all j-s are determined relying on the k-th learning step results.

Step 2. The values of $q_j(k+1)$ are predicted for the $(k+1)$-th learning step: $q_j(k+1/k)$.

Step 3. For the task class under consideration, it is necessary to recalculate the task complexity values predicted for the $(k+1)$-th learning step, the learning tasks being of the same type $T_{rq_0}(k+1/k)$ as at the previous step. If the condition

$$T_{opt} - \Delta T \leq T_{rq_0}(k+1/k) \leq T_{opt} + \Delta T \qquad (8)$$

is met, the task of the indicated type include again into the learning task formed for the $(k+1)$-th learning step.

Step 4. If condition (8) is not met, for all task types of the class in question their complication value deviations from the optimum one are calculated:

$$\Delta T_{rq}(k+1/k) = \left| T_{rq}(k+1/k) - T_{opt} \right|; \; (q = \overline{1, Q_r}). \qquad (9)$$

Step5. If complexity should be decreased, i.e. $T_{rq_0}(k+1/k) > T_{opt} + \Delta T$, is the case, a special-type task search is carried out in the class being considered. The task complication $T_{rq_1}(k+1/k)$ here should have a minimum possible deviation from the optimum:

$$\Delta T_{rq_1}(k+1/k)=\min\{\Delta T_{rq}(k+1/k)\};\ (q=\overline{1,Q_r}). \quad (10)$$

In this case the difficulty level of the desired-type learning tasks have not increase, i.e. $MAT_{rq_1}(k+1/k) \leq MAT_{rq_0}(k+1/k)$. Furthermore, if different-type learning tasks have symmetric deviations of their complication values from the optimum:

$$\Delta T_{rq_1}(k+1/k)=\Delta T_{rq}(k+1/k);\ (q_1, q\in[1,Q_r], q_1 \neq q), \quad (11)$$

the task formulated at the $(k+1)$-th learning step have include a special-type task, its complication value being close to the value $T_{rq_0}(k+1/k)$, i.e. $T_{rq_1}(k+1/k) > T_{opt}$ is satisfied for it in addition in the case under consideration.

Otherwise, when it is necessary to increase complexity, i.e. $T_{rq_0}(k+1/k) < T_{opt} + \Delta T$ takes place, the same search criterion (10) is used. Opposite limitations have be obeyed in this case. The difficulty level of the desired-type learning tasks have not decrease, i.e.: $MAT_{rq_1}(k+1/k) \geq MAT_{rq_0}(k+1/k)$. If there is a symmetry in complication deviations (11),

$T_{rq_1}(k+1/k) < T_{opt}$

have be satisfied in addition.

The limitation of model range of application is a claim to organize operational-based control of learning process.

The analysis of the developed learning model has shown that it is in accord with key requirements imposed upon mathematical models (adequacy, convergence property, universality, effectivity) and can serve as a basis for developing authoring tools of designing the learning model subsystem in ITS.

5. Authoring tools for ITS design

MONAP is a kernel of authoring tools for ITS learning control subsystem design. The learning control subsystem designed by means of MONAP on the basis of step-by-step operational checking of student's answers calculates the student states of mastering for every operation using Bayesian approach, which allows us to consider the prehistory of learning. On the basis of student's answer analysis the MONAP defines the learning task with optimal value of difficulty for student and dispatch this information to the learning task formation subsystem. This subsystem generates or chooses a task from the knowledge base for the following learning step. Thus ITS being designed by means of MONAP organizes an adaptive control of learning process, that is provide complete automation of intelligent functions:

- student's knowledge identification;
- decision making on learning continuation, learning objective achievement, emergency termination;
- determination of properties of a learning task adequate to the student's knowledge for the next learning step.

The learning control subsystem is built up on the basis of the following models:

- learning domain model;
- student model;
- learning process control model.

Two major components may be indicated in MONAP:
- service authoring tools used for knowledge base creation and maintenance;
- functional tools used for the learning dialogue and control.

Knowledge base is an aggregate of learning environments. Figure 1 illustrates the structure of knowledge base.

Each learning environment included into the knowledge base contains following knowledge:
- knowledge about the learning task properties (what type of and how many operations it is necessary to perform for learning task solving);
- knowledge about the student (name, current learning step, properties of the learning task for the next learning step, probabilities of correct performance operations, probabilities of the hypotheses about the mastering levels at the current step);
- knowledge about the learning process control definite by the appropriate model parameter values (the number of hypotheses on being-learned states, optimum task difficulty level, minimum required mastering rate, "threshold of stress" etc.).

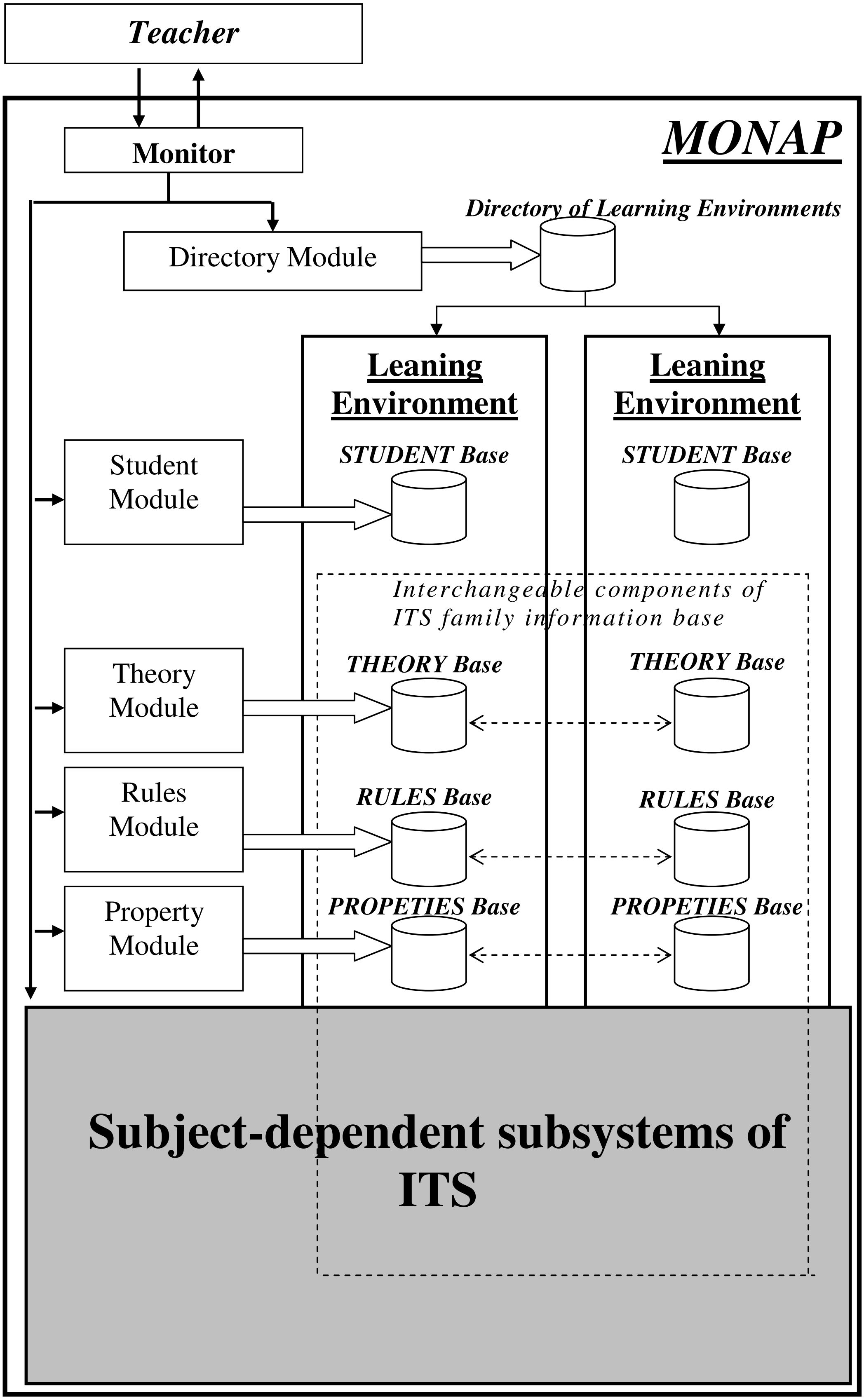

Fig. 1 Structure of knowledge base

6. Learning process modeling

There are no universal rules or exact recommendations for parametrical tuning of learning control model in some domain. So there is a need to include a new subsystem in the authoring tools architecture, which responses for such tuning. Accordingly, it is necessary to extend the architecture of authoring tools by including new subsystem for learning process modeling.

When particular ITS is under construction, teacher has to set values of the learning control subsystem parameters (Fig. 2).

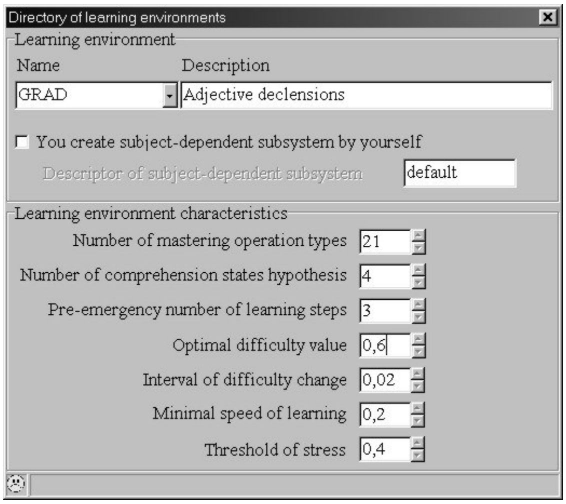

Fig. 2. Parameters of learning control subsystem

The values of some parameters can significantly influence selection of the next task for the student according to his/her mastering level. Experience of MONAP use for ITS GRAD design [Galeev I. et al., 1998] shows that the choice of the values for these parameters is not easy for a teacher regardless of the fact that help system contains recommendations for their evaluation.

To avoid these troubles, a teacher can use modeling mode. In modeling mode (Fig. 3), a teacher can simulate solutions of the learning tasks by some student and check how these results will affect the sequencing of the learning if current values of parameters will be used. The teacher can see the results of knowledge identification process (Bayesian-based approach) in digital and graphic forms. In addition he/she is informed about a type of the next task, which will be proposed for the student according to his/her mastering level and values of learning model parameters. Modifying results of learning task solution and values of the learning model parameters the teacher is able to assort the appropriate values of the parameters for the optimal learning process control.

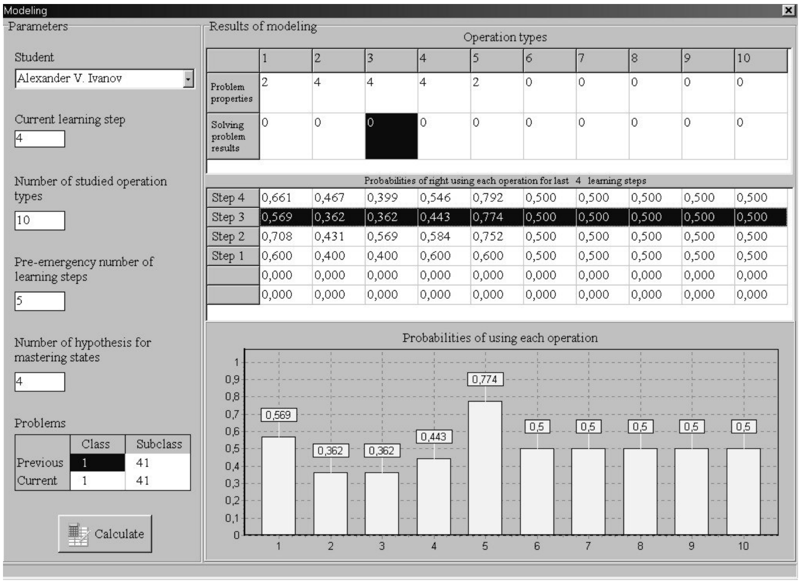

Fig. 3. Modeling learning process in MONAP

In the modeling mode, teacher has access to two different graphical forms to present the student's knowledge identification results. The numbers used for graphic representations are shown in the table. The first type of the graphic (Fig. 3) presents probabilities of the correct use of each operation by the student over the defined number of steps, i.e. presents the data from a chosen row of the table in a graphical way. The second type of the graphic presents a history of change of the mastering level for a particular operation (the graphical representation of a chosen column of the table). To switch between these presentation modes, a teacher has to click on the caption of the appropriate row or column accordingly.

In connection with multifactorial, dynamic and feebly formalized character of learning process there is a need to have tools, which provide for the teacher possibilities to make sequential didactic experimentation. The aim of such experiments is a tuning of learning control model, which provides to allow particular conditions of designed ITS's using. Modeling subsystem realized in MONAP meets the indicated requirements.

7. MONAP-based intelligent tutoring system

Many studies show that the CALL system is more effective than traditional human teacher [Yang & Akahory, 1999]. The analysis of learning domain (the German language grammar in the part of adjective declension) has resulted in developing the structure of a subject-oriented subsystem of a corresponding intelligent tutoring system (ITS GRAD) [Galeev I. et al., 1999]. The learning model subsystem has been designed using MONAP authoring tools. In accordance with the previously elaborated architecture of ITSs being designed, the composition of the subject-oriented expansion of the learning environment in ITS GRAD has been determined. Software has also been worked out, which provides the creation and maintenance of the expansion, including the possibility to build-up an ITS family in domain under consideration. Such ITSs have network architecture, i.e. use common components of information bases. The developed ITS GRAD, ensuring an adequate process of mastering the German language grammar in the part of adjective declension, represents an open-type flexible system able to expand and change in accordance with the teacher's demand.

The importance of mastering the grammar of the German language in the part of adjective declensions is conditioned by the fact that the knowledge about adjectives is used both for the synthesis, and for the analysis of a sentence in German. When a sentence in German is synthesized, the main task at studying an adjective is to determine its ending correctly. In case of analyzing (understanding) a sentence in German, the knowledge about adjectives may play the most important role when a large variety of analysis learning tasks are being solved:

- the subject recognition;
- the number distinction;
- the extended sentence identification, etc.

So, solving the synthesis learning tasks, the student acquires some sort of knowledge, which can be also used when analyzing a sentence in German. In this connection it is worthwhile to design ITS GRAD (GR- Grammatik, AD - Adjektive) in which a learning task represents a set of sentences in German with the roots of adjectives. The student has to put the necessary endings in a correct way. A fragment of a learning task can appear as:

Das klein… Kind trinkt warm… Milch.

On the basis of the adjective morphology analysis and taking into consideration the results of analyzing the determined noun and articles morphology, a general approach to algorithmic directive formation has been formulated. The algorithmic directives in question are aimed at the determination of adjective endings, are organized as a set of rules (operations) like “IF (condition), THEN (action)” and realize the external form of representing the expert–teacher’s knowledge about the German language grammar in the part of adjective declension.

It is obvious that a whole class of such algorithmic directives can be formed in NL. In this case the algorithmic directives of the mentioned class may differ considerably from each other. The following basic differences are possible:

- according to the types of the assigned operations included into the algorithmic directive;
- according to the detail degree of the assigned operations included into the algorithmic directive;
- according to the sequence of the assigned operations set forth by the teacher;
- according to the NL form of representing the algorithmic directive stemming from both the NL ambiguity, and its synonymy.

As an example illustrating a potential diversity of algorithmic directives in the learning domain under consideration, a number of operations describing this or that subset of adjective declension rules in a weak form has been offered:

IF there is a definite article *der* or one of the *pronounse*, *dieser*, *jener*, *solcher*, *jeder*, *welcher*, before the adjective and the noun, and the noun being attributed is of the masculine gender, THEN the adjective ending is -e;

IF there is a definite article *der* before the adjective and the noun, and the noun being attributed is of the masculine gender, THEN the adjective ending is –e.

Within the framework of the formulated general approach, a particular algorithmic directive has been worked out, the file “RULES” being used for storing it.

The teacher is able to work out his own algorithmic directive which differs from the one proposed and adequately (both in the NL representation form, and in the content) reflects his knowledge scope in the learning domain being analyzed. ITS GRAD software supports such an opportunity.

The following structures of subject-oriented ITS GRAD subsystems have been defined. The subsystem of tutoring task formation is aggregate of the learning tasks and tools for it’s creation and maintenance. Also this subsystem includes program for task representing and properties, which define by the learning process model in accordance with the student’s current knowledge.

The learning model subsystem has been realized via authoring tools MONAP [Galeev I. et al., 1996]. There is no task–solver subsystem in connection with the decision not to construct the task generator.

The diagnostics subsystem provides:

- student’s answer input;
- student’s answer analysis;
- diagnostic message output;
- assistance representation to the student;

- identification of the type and number of the errors made;
- feedback formation and control return.

The explanation subsystem is defined as the diagnostics subsystem expansion due to including the file RULES into it's structure. The file is used to form the answers to the student's questions “WHY?”. The answer is formulated in a natural language way as “IF..., THEN...”.

Software has been developed, which ensure the creation and maintenance of the subject-oriented expansion of the educational environment ITS GRAD. The indicated software supports the ITS openness level preassigned by the MONAP authoring tools. A subject-oriented component of the tutoring dialogue has been worked out. This component is carried out by the corresponding subject-oriented subsystems ITS GRAD performing their basic functions. Moreover, the process of the dialogue contemplates analyzing the logical integrity (completeness and consistency) of the ITS GRAD information base. If the integrity is broken, the teacher receives emergency message.

ITS GRAD, ensuring an adequate process of mastering the German language grammar in the part of adjective declension, represents an open-type flexible system able to expand and change in accordance with the teacher's demand. Using the software at his disposal the teacher is able:

- to alter the existing (supplied) ITS;
- to build up a set of independent and essentially different ITS in the considered learning domain for different students' categories;
- to build up an ITS family in the considered learning domain, having a network architecture, i.e. using common components of information bases.

8. Future Extension

At present development of intelligent CALL system for Russian language have begun. The approbation of proposed approach in another domains is planned.

The benefits of Web-based ITS are well known. A number of existing Web-based adaptive educational systems, for instance, ELM-ART, CALAT, WITS and Belvedere, were developed on the basis of earlier standalone ITS [Brusilovsky P., 1998]. As we know Web-based ITS authoring tools for courses design are available only just on the system developer's server. This greatly restricts a range of teachers, having possibility of making their own author's courses in these systems, what entails an amount restriction of created courses. So, one of the primary goal of new MONAP versions is realization of authoring tools for the learning environment designing, which help teacher to create their own learning environment from the remote computer, connected to the Internet. As a result, a great deal of teachers gets a possibility to build an ensemble of learning environment with different didactic features in one domain. Thereby, the system turns into the suitable instrument for stating of the scale pedagogical experiments and collection a lot of useful statistical information that increase the system's means in the end enlarge value for students.

9. References

[Conati C. & VanLehn K., 1996] Conati, C. & VanLehn, K., POLA: A student modeling framework for probabilistic on-line assessment of problem solving performance. In Proceedings of UM-96, Fifth International Conference on User Modeling. Kailua-Kona, HI: User Modeling Inc.

[Galeev I. et al., 1996] Galeev, I., Ivanov, V., Akhmadullin, M. A Learning Model in MONAP // Human-Computer Interaction. The 6th International Conference. EWHCI'96. Moscow, Russia, August 12-16, 1996. - P.320-323.

[Kinshuk & Patel, 1997] Kinshuk, Patel, A. “A Conceptual Framework for Internet based Intelligent Tutoring Systems” Knowledge Transfer, volume II, Ed. A. Behrooz, pAce, London, 1997, pp. 117-124

[Galeev I. et al., 1998] Galeev, I., Ivanov, V., Akhmadullin, M. The experience of development of intelligent tutoring systems, in proceeding of 27th International symposium "Ingenieurpädagogik'98", Volume 2 "Padägogische Probleme in der Ingenieurausbildung" - pp. 255-258, Moscow, 1998.
[Brusilovsky P., 1998] Brusilovsky, P. Adaptive Educational Systems on the World-Wide-Web: A Review of Available Technologies, in proceedings of 4th International Conference on Intelligent Tutoring Systems (ITS'98), San Antonio, 1998.
[Gertner A. et al., 1998] Gertner, A., Conati, C., and VanLehn, K. Procedural help in Andes: Generating hints using a Bayesian network student model. In: Proceedings of the Fifteenth National Conference on Artificial Intelligence AAAI-98. Cambridge, MA: The MIT Press. pp.106-111
[Yang & Akahory, 1999] Yang, J. Ch., Akahory, K. An Evaluation of Japanese CALL Systems on the WWW Comparing a Freely Input Approach with Multiple Selection // Computer Assisted Languages Learning, 1999, Vol. 12, No. 1, pp. 59-79
[Galeev I., 1999] Galeev, I. "Automation of the ITS Design" Educational Technology journal, V. XXXIX, No. 5, September-October 1999, pp. 11-15.
[Galeev I. et al., 1999] Galeev, I., Tararina, L., Sosnovsky, S. "The structure and functions of ITS GRAD", Proceedings of 8th International conference on Human-Computer Interaction (HCI'99) Volume 2 Munich, Germany, Lawrence Erlbaum Associate, Publishers, London, 22 – 26 of August, 1999, pp.682-685.