A THREE-LEVEL LEARNER'S MODEL FOR THE NEEDS OF AN INTEGRATED ENVIRONMENT FOR INDIVIDUAL PLANNED TEACHING

Irina Zheliazkova, Rumen Kolev

Abstract: *The learner's knowledge, skills, psychological characteristics, and preferred style of learning have been used for individualization of teaching in different generations of computer-based systems. A survey indicates that the problem of the learner modeling in the intelligent distributed teaching and learning environments is still opened. In the present paper a three-level learner's model for the needs of an integrated environment for individual planned teaching is discussed.*

Key words: *Learner's Model, Integrated Environment, Individual Planned Teaching.*

1. INTRODUCTION

Our survey [13] on learner modeling has shown that the learner model (LM) was developed with different parameters and accuracy in different Computer-Based Teaching (CBT) systems. As much the LM is rich so the corresponding system is more adaptive to the learner's knowledge and behavior. In comparison with the other generations of CBT systems the Intelligent Tutoring Systems (ITSs) spread in the nineteen's [9] supported more detailed and precise LM. The learner's knowledge, skills, psychological characteristics, preferred style of learning has been used for individualization of teaching in these systems. Such a rich model is able to present different categories of learners, e.g. novices, beginners, advanced, and so on. In such a way the learner step by step gathers knowledge coming up to the teacher's one. For this reason most of the CBT systems use similar dialog scenario or language for representation of teacher and learner's knowledge in a given domain. So the corresponding system is close to the ideal teaching goal: maximum knowledge for minimum time for each learner.

The survey also indicates that the problem of the learner modeling in the distributed teaching and learning environments spread in now days is still opened. Although their LM is domain-independent, the number and type of its parameters vary greatly from one environment to another. It is interesting to note, that the environments supporting two-levels LM [4,11] are potentially more adaptive to both teachers and learners.

In [8] the IEEE learning technology systems architecture (LTSA) is described in five successive refinement layers from highest to lowest and two of which are:

- *Learner and Environment Interactions*: Concerns the learner's acquisition, transfer, exchange, formulation, discovery, etc. of knowledge and/or information through interaction with the environment.
- *Learner-Related Design Features*: Concerns the effect learners have on the design of learning technology systems.

The LTSA system components identify the critical interoperability interfaces for learning technology systems. The LTSA does not identify all interoperability interfaces for particular learning technology systems. Briefly, the overall operation has the following form:

(1) The learning styles, strategies, methods, etc. are negotiated among the learner and other stakeholders and are communicated as learning preferences;
(2) The learner is observed and evaluated in the content of multimedia interactions;
(3) The evaluation produces assessments and/or learner information;
(4) The learner information is stored in the learner history database;
(5) The coach reviews the learner's assessment and learner information, such as preferences, past performance history, and, possibly, future learning objectives;
(6) The coach searches the learning resources, via query and catalogue info, for appropriate learning content;

(7) The coach extracts the locators from the available catalogue info and passes the locators to the delivery process, e.g. lesson plan;

(8) The delivery process extracts the learning content from the learning resources, based on locators, and transforms the learning content to an interactive multimedia presentation to the learner.

This paper deals with a three-level LM for the needs of an integrated environment for individual planned teaching [6] and is organized as follows. In the next section the complete relational model of the learner shortly is commented. Its three components exhaustively are discussed in the separated sections. The last section outlines the authors' conclusions.

2. THE COMPLETE RELATIONAL MODEL

The idea of the relational LM is not new [3,10]. In 1990 Hawkes L.W. et al. proposed an individualized fuzzy temporal relational model. The following arguments in favor of the relational LM in comparison with the other existing ones are listed in [12]:

- Table knowledge representation as a set of classes, objects, attributes and logical relationships between them is comprehensible for each user;
- The relationships between the objects have an universal character, e.g. the combinational, hierarchical and network relationships are the particular cases of the relations;
- Object attributes can be different types (e.g. Boolean, time, range, numeric value) with non complete and inaccurate information;
- The relational model does not limit the number of the domain structure levels and the number of elements for each level;
- Model processing can be formalized using the relational algebra or directed graphs with marked nodes and arcs;
- Direct transition to the program implementation is possible if a visual object-oriented language is chosen;

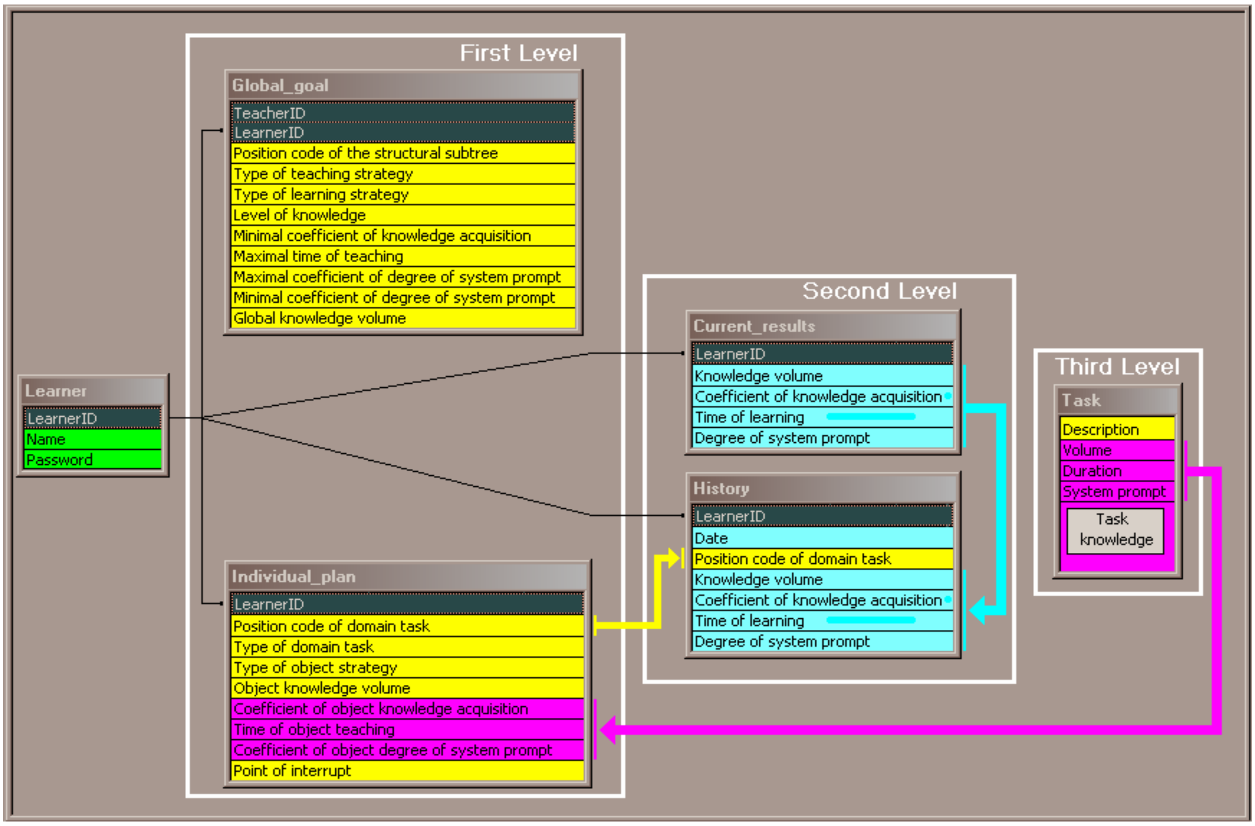

Fig. 1. The complete relational model

- The model support is reduced to standard procedures for creating, updating and deleting of a file, record, and field;
- The organization of friendly and flexible user interface is reduced to the classical procedures for reading a file, record, and field.

The graphical interpretation of the proposed relational LM for an intelligent environment for individual planned teaching is given on fig.1. The used notation differs from the classical one in presence of data flows (pictured as arrows) for the LM support.

3. THE FIRST LEVEL MODEL

This model that is common for the different task-oriented environments is domain-, object-, and task-independent. It can be classified also as long-term with low frequency of the learner-computer interactions. Using this model the most appropriate plan can be executed for the individual learner. The environment for individual planned teaching has to refresh this model performing a given task concerning a given learning object.

The model consists of three relations called on fig.1 ***Learner***, ***Global goal,*** and ***Individual plan*** respectively. Let *i* stands for the position code of the structural sub-tree planned to be taught, *j* for the number of the current task, $Q^{gl}(i)$ for the knowledge volume, $Q^{cr}(i)$ for the volume knowledge acquired after the current task performance, $Q^{tk}(i,j)$ for the volume knowledge of the *j*-th task. Analogically, $T^{gl}(i)$, $T^{cr}(i)$, and $T^{tk}(i)$ have the same meanings but for the time of teaching. Now, if the flag for interruption of the session after the *j*-th task $I^{tk}(i,j)=false$ and flag for finishing the teaching $I^{gl}(i)=false$ then the learner-computer dialogue continues with teaching the next planned task *j+1*. If $I^{tk}(i,j)$ becomes *true* and $I^{gl}(i)$ remains *false* the achieved values of $Q^{cr}(i)$ and $T^{cr}(i)$ are computed and shown to the learner together with the corresponding $Q^{gl}(i)$ and $T^{gl}(i)$ settled by the teacher. Then the learning rate for the current session is computed and compared with the predicted rate for the rest of the planned time T_{rt}:

$$\frac{Q^{gl}(i)-Q^{cr}(i,j)}{T^{gl}(i)-T^{cr}(i,j)} \le \frac{\sum_{j=J1}^{J2} \frac{Q_{rt}^{tk}(i,j)}{T_{rt}^{tk}(i,j)}}{J2-J1}$$

Here *J1* and *J2* are respectively the numbers of the initial and final task of the current session. Satisfaction of this inequality predicts success of the learner's individual plan. In this plan the next session will continue with the task number *J2+1*. If the inequality is not satisfied the human teacher should correct the rest of the plan choosing among the following ways: increase the planned time of teaching, decrease the domain breadth and/or depth, and finally increase the coefficient of degree of system prompt. If $I^{gl}(i)=true$ the dialogue moves from current state of the learner–computer dialog to the final one.

4. THE SECOND LEVEL MODEL

The LM on this level includes two relations called on fig.1 respectively **CURRENT RESULTS** and **HISTORY**. Similar to the first level model it is also domain-, object- and task-independent and is refreshed after the learner finishes a given task. The environment can use this model mainly for improving the tasks pool. In the relation **CURRENT RESULTS** four learner's parameters are stored, namely: the level of the task difficulty *L,* coefficient of learner's assessment relatively to the teacher's one C_a^{tk}, time of learning T_a^{tk}, and degree of system prompt C_p^{tk}. The last parameter is a coefficient between 0 and 1.00 and shows what part of the teacher's knowledge is available during the task performance. The first parameter depends on the underlying graph model's type representing learner's task knowledge and skills. If the graph is isolated, then *L=1*, if linear – *L=2*, if a tree – *L=3*, if a cycled graph – *L=4*. The sum of the powers of the nodes set N_1 and of the connections set L_1 of the teacher's program tree can serve as an objective and precise measure for the knowledge volume, e.g. $Q_1=|N_1|+|L_1|\approx 2|N_1|$. Let *A* and *B* mean respectively the number of the nodes and arcs that are missing in T_1, but presenting in T_2, and *C* and *D* – the number of the nodes and arcs, that present in T_1, but miss in T_2. According to the graph theory it can be said that A+B is a measure for the missing knowledge atoms, and *C+D* – for the incorrect knowledge atoms of the learner. Then the following simple formula is close to the intuitive expectations about the learner's knowledge relatively the teacher's one: $C_a^{tk*} = (Q_1-(A+B+C+D)) / Q_1$. On fig.2 the graphical interpretation of the learner's assessment relatively the teacher's one is given.

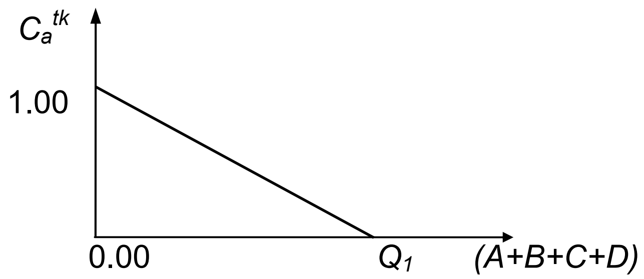

Fig.2. The learner's assessment relatively the teacher's one

The temporal correction C_a^{tk*} of this assessment takes into account also the time for learner's performance t_2 relatively to the time of the teacher's one t_1. If we assume that the learner performs the task *c* times slower or faster than the teacher, e.g. $t_1/c \le t_2 \le c.t_1$ then

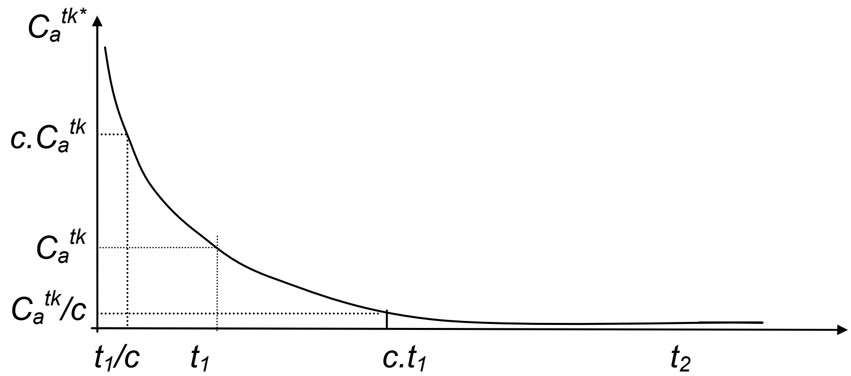

Fig.3. Graphical interpretation of the temporal correction

the following function for temporal correction could be used $C_a^{tk*} = C_a^{tk}(1+c.t_1/t_2)$. It is interesting to note here that both models (C_a^{tk} and C_a^{tk*}) satisfy the requirements formulated by Samojlov et al., 1989 [7] for assessing the learner's training task: *1) to be simple and comprehensible*, 2) *based on simple and clear information principles,* 3) *normalized in a given range, and 4) to increase after a successful training.*

In the relation **HISTORY** the accumulated values of the current learner's parameters are stored, namely: *coefficient of knowledge acquisition* (C_a^{cr}), *degree of system prompt* (C_p^{cr}), *knowledge volume* (Q_a^{cr}), *time of teaching* (T_a^{cr}). If the task number $(j+1)(j=1,N)$ for a given learning object with an index code *i* from a given course these parameters are refreshed using the following simple formulas:

$$C_a^{cr}(i, j+1) = [C_a^{cr}(i, j) + C_a^{tk}(i, j+1)] / 2;$$
$$C_p^{cr}(i, j+1) = [C_p^{cr}(i, j) + C_p^{tk}(i, j+1)] / 2;$$
$$T_a^{cr}(i, j+1) = T_a^{cr}(i, j) + T_a^{tk}(i, j+1);$$
$$Q_a^{cr}(i, j+1) = Q_a^{cr}(i, j) + Q_a^{tk}(i, j+1);$$

Of course, at the beginning of the teaching process the initial values of the first two parameters must be equal to 0.5, and of the last two parameters equal to 0.

5. THE THIRD LEVEL MODEL

On this level each task-oriented environment included in an integrated environment for planned teaching supports its own task-dependent LM [1]. In comparison with the previous ones this model exists for a shorter time and is refreshed with higher frequency. The main purpose of the model is for diagnosis and remediation the learner's misconceptions and missing knowledge. The model on this level is in the form of a program representing the learner's knowledge and skills about the given task. This program is generated during the task performing by the learner. A common program structure of the tasks for training in dynamic systems is given in table 1.

Table 1

<learner's program> ::= ***SYSTEM*** *[<string>]* ***DESCRIPTION*** *<free text>* ***DURATION*** *<integer>* ***VOLUME*** *<integer>* *{<description of a parameter>}* *{<description of a dependency >}* *[**COLOURS** <integer><integer>]* *[**DISCR_STEP** = <real>]* *[{<description of an event >}]* *[**TRACE** <list of parameters >**END**]* *[**SPEED** = <integer>]* *[**TIMER*** *@|<integer><integer><integer>]* *[{<procedural operator >}]* *[{<operation>}]* ***END***	*< description of a parameter > ::=* ***VAR*** *<string>* *[**X**= <integer>]* *[**Y**= <integer>]* *[**WIDTH** <integer>]* *[**COLOURS_ON**]* *[**INVISIBLE**]* *[**UNITS** <string>]* *[**VALUE** = <real>]* *[**STEP** = <real>]* *[**LIMITS** <real> | **@ :** <real> | **@**]* *[**NORMAL** <real> | **@ :** <real> | **@**]* ***END*** *<description of a dependency> ::=* ***DEPENDS*** *<string>=<expression>* *<description of an event> ::=* ***IF*** *<expression>* ***THEN*** *{<action>}* ***ELSE*** *{<action> }* ***END***

This kind of learner's programs present a sequence of declarative and procedural blocks of knowledge, also require operational skills necessary for a given task performance. In the terms of dynamic systems declarative knowledge are general facts about the system, i.e. its purpose, components, relationships among them, and system functions. Procedural knowledge consists of standard procedures and activities that skilled operators use to manage the system. Operational skills represent the way to use declarative and procedural knowledge to cope with abnormal situations in real time. These skills are needed to perform successfully an investigative, control or diagnostic task.

The comparison of learner's procedural knowledge with the teacher's one is simpler than declarative knowledge comparison. The reason for this simplicity is the fixed sequence of the procedural operators in the teacher's program. The number and the sequence of their parameters are fixed too. After analysis of the results of the current procedural operator a multiple choice question could be presented by the training system to test whether the learner has made the right conclusion. In order not to disturb the learner's process of real-time decision making the comparison is provided when the time, allocated to this task in the teacher's program, is over.

When the current unit of the learner's declarative knowledge does not match the knowledge unit of the teacher's program the reason can be attributed to many causes. Through context-dependent instructions the training system can then rectify the most probable learner's misconceptions. The diagnostic hypotheses and the remedial instructions for knowledge and skills of dynamic systems can be found in [2].

6. CONCLUSIONS

The three-level learner model is proposed for the needs of an integrated environment for individualized planned teaching [14]. This model has the good diagnostics possibilities similar to ones of the overlay models in the classical ITSs. Additionally, the information in this model can be useful for improving the initial teaching material, teaching and learning strategies, as well as for implementation of adaptive algorithms for making decisions [5,7] and generation of a more realistic individual learner's plan.

References

[1] Georgiev G. T., I. Zheliazkova. Representation and Processing of Domain Knowledge for Simulation-Based Training in Complex Dynamic Systems. Int. Journal Intelligent Systems, 2000, Vol.10, No.3, pp.255-277.

[2] Georgiev G., I. Zheliazkova. Diagnosis and Remedial of Students' Knowledge about Complex Dynamic Systems. Proceeding of the 13th International Conference SAER, 18-19 September, Varna, Bulgaria, 1999, pp.224-228.

[3] Hawkes L. W., S. J. Derry, E. A. Rundensteiner. Individualized Tutoring Using an Intelligent Fuzzy Temporal Relational Database. Int. Journal of Man-Machine Studies, 1990, pp. 409-429.

[4] Jesshope Chris, E. Heinrich, D-r Kinshuk. Technology Integrated Learning Environments for Education at Distance. Massey University, Palmerton North, New Zealand. http://www10.org/cdrom/papers/207/WWW10-207.html.

[5] Paschin E. H., A. I. Mitin. Automated Teaching System EKSTERN. MSU, Moscow, 1985 (in Russian).

[6] Peachey D. R., G. I. McCalla. Using Planning Techniques in Intelligent Tutoring Systems. Int. J. of Man-Machine Studies, 1986, Vol.24, pp.77-98.

[7] Samojlov, V. D., V. P. Bereznikov, A. P. Pisarenko, S. I. Smetana. Automation of Coaches and Teaching Systems. Naukova dumka, 1989, Kiev (in Russian).

[8] Shoikova E., V. Denishev. eLearning Technology Standars Overview. Proceedings of the International Conference on Computer System and Technologies (e-Learning), CompSysTech, 2002, pp.P.1-1, P.1-13.

[9] Sleeman D., J. S. Brown. Intelligent Tutoring Systems. Academic Press, London, 1982.

[10] Stoyanova L. J., Hierarchical Model of the Database Organization in Computer-Based Systems. Theses of PhD Dissertation, Sofia, 1991 (in Bulgarian).

[11] Vadillo J. A., A. Diaz Ilarraza. The Role of the User Model for Explanation in Intelligent Tutoring Systems for Training. 3rd International Conference on Computer Aided Engineering Education, 13-15 September, 1995, Bratislava, Slovakia, pp.101-106.

[12] Zheliazkova I. et al. User Interface Modeling in an Intelligent Hypertext Environment for Support of Teaching. Learning and Training. Annual report of the Project No. I-511/95, NSRF, Rousse, 1998

[13] Zheliazkova I., R. Kolev. Learner Modeling in the Computer-Based Teaching Systems. Journal of Information Technologies and Control, 2004 (accepted).

[14] Zheliazkova I. Models of Teaching and Learning Environments. Proceedings of Conference on Computer Systems and Technologies, 2004 (accepted).

ABOUT THE AUTHORS

Assoc. Prof. Irina Zheliazkova, Ph.D., Department of Computer Systems and Technologies, University of Rousse, Phone: +359 82 888 744, e-mail: irina@ecs.ru.acad.bg

Ph.D. student Rumen Kolev, Department of Computer Systems and Technologies, University of Rousse, Phone: +359 82 888 744, e-mail: rkolev@ecs.ru.acad.bg