THE METHOD OF COMPUTER TUTORING PROGRAM PEDAGOGICAL ACTIONS FORMATION FOR ITS INNER LOOP TASKS

Kulik, Anatoliy S.
National Aerospace University named after N.Ye. Zhukovsky "KhAI", Ukraine

Chukhray, Andriy G.
National Aerospace University named after N.Ye. Zhukovsky "KhAI", Ukraine

Pedan, Stanislav I.
National Aerospace University named after N.Ye. Zhukovsky "KhAI", Ukraine

Hall, Steven D.
Texas A&M University – Corpus Christi

ABSTRACT
The method of decision-making model construction for computer tutoring program pedagogical actions formation is described. Solving algorithm of many technical problems can be presented in the form of a program code. Stages of automatic model construction for such type problems are described. The example of the method work for a problem of linear equations system solving is presented. Testing of model is held and efficiency of its use in real tutoring process is proved.

INTRODUCTION
One of the purposes of intelligent computer tutoring programs (ICTP) creation is effective knowledge and skills transferring to the trainees due to adaptation to each of them, and also implementation of such important pedagogical principles, as activity, deliberateness and self-diagnostics (Komenski, Lock, Rousseau and Pestalozzi 1989). But obstacle in this way is inconsistent and incomplete evidences on tutoring process course, received by ICTP. The decision of the given problem can be found in creation of possibility of tutoring system to model learning process for receiving of the full and correct information on it and formation of correct pedagogical actions.

In the present research the problem of method development for ICTP pedagogical actions sequence formation is solved. In following section the question of approach choice to the developed method performance organization is considered.

CHOICE OF APPROACH TO THE TUTORING ACTIONS SEQUENCE FORMATION

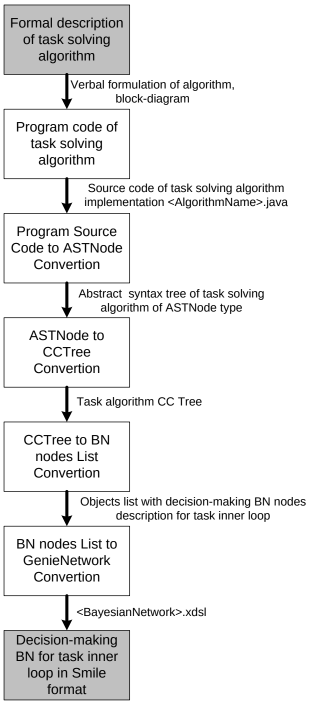
There is a set of approaches to formation of the program and principles of giving of a teaching material for each of trainees (VanLehn 2006). The simplest decision of the given problem is trainee's independent choice of the next task from their static list. One more widespread decision is formation of tutoring sequence by the teacher and providing to each trainee of a prepared tasks sequence. But an essential disadvantage of the described approaches is absence of adaptation of computer tutoring program to knowledge and skills of each separate trainee. Hereupon performance of deliberateness and activity pedagogical principles at studying of a new material can become impossible. For one trainee of high level competence the given tasks can seem too easy and uninteresting, for others to cause tearing away owing to high complexity. Formation of adaptive sequence of ICTP pedagogical actions taking into account different level of each trainee competence level can become an exit from a current situation. This purpose could be reached by implementation of ICTP inner and outer loops formation ideas proposed by American researcher Kurt Van Lehn in work (VanLehn 2006).

Outer loop is responsible for the decision-making, about what next task to propose to the trainee. ICTP should contain a set of tasks ordered automatically or manually by the teacher by its complexity level. Transitions in outer loop are carried out by corresponding decision-making model which chooses the next problem proceeding from “Vygotsky zone” condition (Vigotsky 2005), i.e. it should be a bit more difficult than the previous one. Inner loop is responsible for splitting of the task into sequence of steps and providing of timely feedback with the trainee at execution of the given steps. Inner loop execution means observance of rationality criterion at next tutoring iteration choice as the final goal of mastering by a set of task competence components (CC) should be reached for the minimum quantity of steps.

Because work of inner loop occurs in conditions of uncertainty about tutoring process and is aimed at formation of rational steps sequence of tasks execution for loop transition model implementation it is suitable to use probabilistic approach methods based on decision-making Bayesian networks (BN) (Murray, Lehn and Mostow). The given model allows to estimate probabilities of each tasks steps execution success and to choose the most useful of them from the pedagogical point of view. Principles of inner loop execution can be applied to solving of various technical problems which algorithm can be formalized and presented in the form of a program code. In following section the method of automatic formation of transition model for ICTP inner loop on the basis of source program code of problem solving algorithm is described.

DEVELOPMENT OF TUTORING ACTIONS FORMATION METHOD FOR ICTP INNER LOOP

The first stage of process of inner loop transition model automatic formation based on a program code of problem solving algorithm is its parsing. It is aimed at identification of CC, necessary for task performance. After that on the basis of task CC structure there is construction of the decision-making BN, forming rational sequence of task steps execution due to choice of the most useful alternative after receiving of the next evidences about tutoring process progress. In a picture 1 the scheme of decision-making BN automatic formation stages for ICTP inner loop is presented.



Picture 1 - The scheme of decision-making BN automatic formation stages for ICTP inner loop

The initial data for work of transition model automatic formation system is the formal description of problem solving algorithm in the form of the verbal information on sequence of its stages, block diagrams, formulas of mathematical calculations.

The result of system work is decision-making BN for tutoring program inner loop. BN is represented in SMILE format (GeNIe) (platform-independent library of classes for implementation of graphic probabilistic models, and also decision-making models), convenient for the decision of its analysis problems and solving of probabilistic inference problems.

For receiving of final decision-making BN it is necessary to execute a number of actions:

1) The initial problem algorithm is represented in the form of program code written in high level language, in our case in Java. Java provides wide tools for program code syntax analysis. The given function is carried out by the developer of the tutoring program or by available automated tools of problem solving algorithm graphic representation transformation in a corresponding program code. Result of this stage performance is the file with a Java-code of initial problem solving.

2) The construction of an abstract syntax tree (AST) of problem solving program code is made. The object of ASTParser class is created for this purpose. ASTParser is an element of the Eclipse JDT API-functions (Gamma, Helm, Johnson and Vlissides 1995) for program code AST construction, manipulations it, detection of its errors, compilation and execution. AST is the final, oriented tree in which internal tops are associated with operators of a programming language, and leaves with corresponding operands. Thus, leaves are empty operators and represent only variables and constants. After reference of program source code the object of ASTParser class will transform it to hierarchical structure of ASTNode type. Every Java language construction can be presented by node of corresponding type with various nesting degree concerning root node of a tree. For example, the

function containing the declaration of two variables, can be presented at top level by MethodDeclaration type node (ASTNode successor) and two nodes of VariableDeclarationStatement type nested in it. As a result the source program java-code will be transformed to corresponding hierarchical structure of abstract syntax tree nodes.

3) Algorithm AST will be transformed to a problem CC tree (CCT). The given tree represents the hierarchical structure consisting of terminal type nodes (operators and names of variables of a source code) and non-terminal, containing the information on the list of the nodes nested in them. CCT can be received from AST by application of idea of Composite pattern [6]. Thus for CCT construction AST nodes connected only with CC interested to us are analyzed, other nodes are ignored.

4) CCT will be transformed to objects array with the information on decision-making BN nodes for an inner loop of a problem. Thus, if CCT contained relations between the nodes, hierarchies going from the top nodes to the nodes nested in it the direction of BN relations changes from more nested to the top nodes of hierarchy. It is connected with features of decision-making BN probabilistic inference as large CC acquirement depends on results of performance of its components.

5) On an available array of the information on BN nodes the BN object of Network type which is a copy of API SMILE class is constructed. The given object can be saved in a file of Genie format (graphic environment for synthesis and the analysis of probabilistic models), and also is used for the internal program analysis and solving of BN probabilistic inference problems.

Let's consider process of decision-making BN automatic construction for a problem of tutoring to the solving of the linear equations systems.

Let the system of two linear equations in a general view looks as follows:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 = b_1, \\ a_{21}x_1 + a_{22}x_2 = b_2 \end{cases} \quad (1)$$

where $a_{11}, a_{12}, a_{21}, a_{22}, b_1, b_2$ - some known integers, $a_{11}, a_{12}, a_{21}, a_{22}, b_1, b_2 \in Z$. It is necessary to define values of roots $x_1, x_2 \in R$.

There is a set of ways of the given linear equations system solving. Let’s use Kramer's method and find a system determinant, i.e. the determinant of the second order made of factors at unknowns:

$$V = \begin{vmatrix} a_{11} a_{12} \\ a_{21} a_{22} \end{vmatrix} \quad (2)$$

Let's make two more determinants from (1), having replaced in the first of them the first, and in the second - the second columns with the column made of free members b_1, b_2:

$$V_1 = \begin{vmatrix} b_1 a_{12} \\ b_2 a_{22} \end{vmatrix} \qquad V_2 = \begin{vmatrix} a_{11} b_1 \\ a_{21} b_2 \end{vmatrix} \quad (3)$$

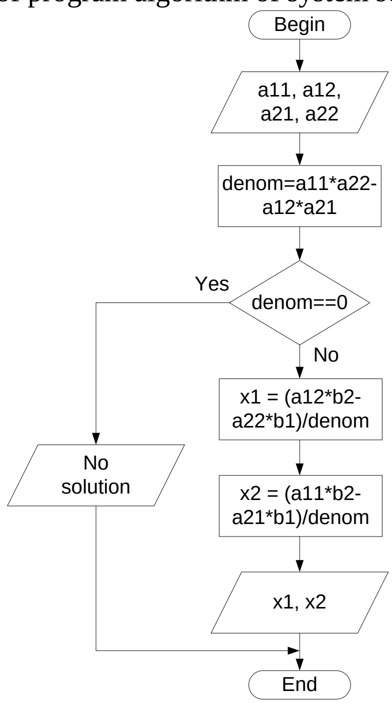
Then, according to Kramer's rule, if a system determinant $\Delta \neq 0$ the considered system has one and only one decision, and:

$$x_1 = \frac{V_1}{V} \qquad x_2 = \frac{V_2}{V} \quad (4)$$

Thus, for finding of linear equations system roots it is necessary:

1) To find value of system determinant according to the formula (2);
2) To check up a condition of system decision existence ($\Delta \neq 0$);
3) To construct 2 private system determinants according to the formula (3);
4) To find values of the first and second roots according to the formula (4).

Having the verbal description of solving algorithm of system from two linear equations, we can make the block diagram of program algorithm of system solving:

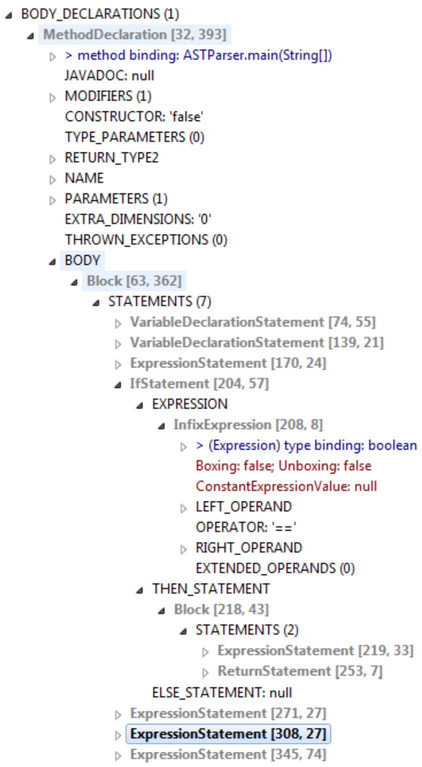


Picture 2 - Algorithm of two linear equations system solving

According to the algorithm made above it is possible to make its implementation in Java language. The program code by system roots finding (1) will have the following view:

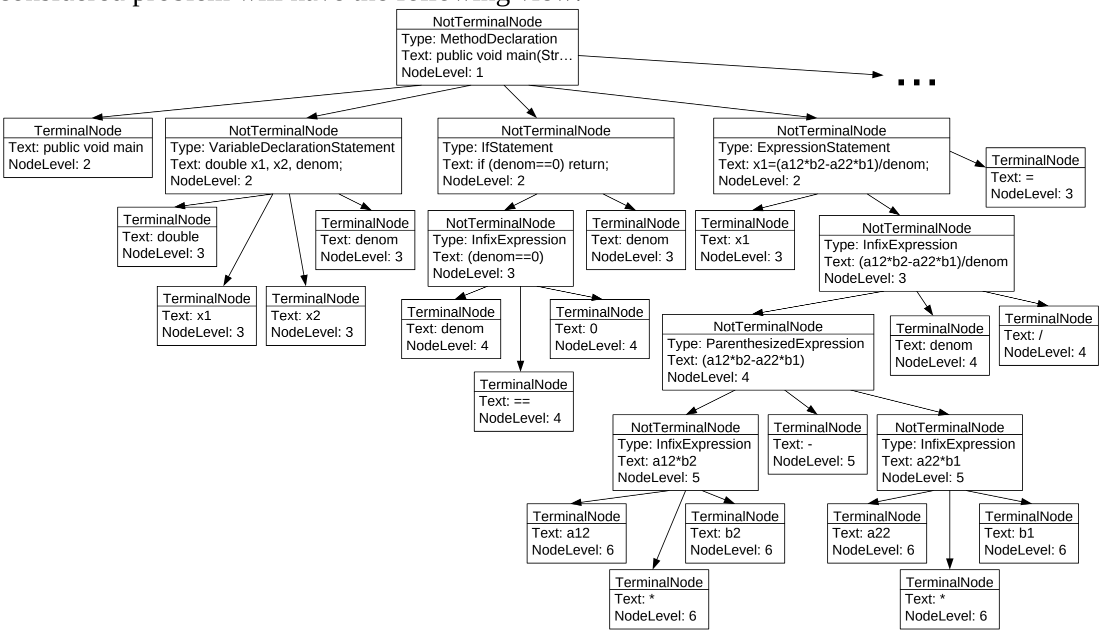
public class EquatSystemSolving {public void main(String[] args) {int a11 = 1, a12 = 2, a21 = 3, a22 = 4, b1 = 5, b2 = 6; double x1, x2, denom; denom = a11*a22-a12*a21; if (denom==0) {System.out.println("No solution"); return;} x1 = (a12*b2-a22*b1)/denom; x2 = (a11*b2-a21*b1)/denom; System.out.println ("x1="+ Double.toString(x1) +"; x2 = "+Double .toString (x2));}}

For creation of algorithm abstract syntax tree it is necessary to create object of ASTParser class and by means of a function setSource call to specify the java-code of system solving described above. By means of utility ASTView it is possible to receive visual AST representation.



Picture 3 - The fragment of AST structure of equations system solving algorithm

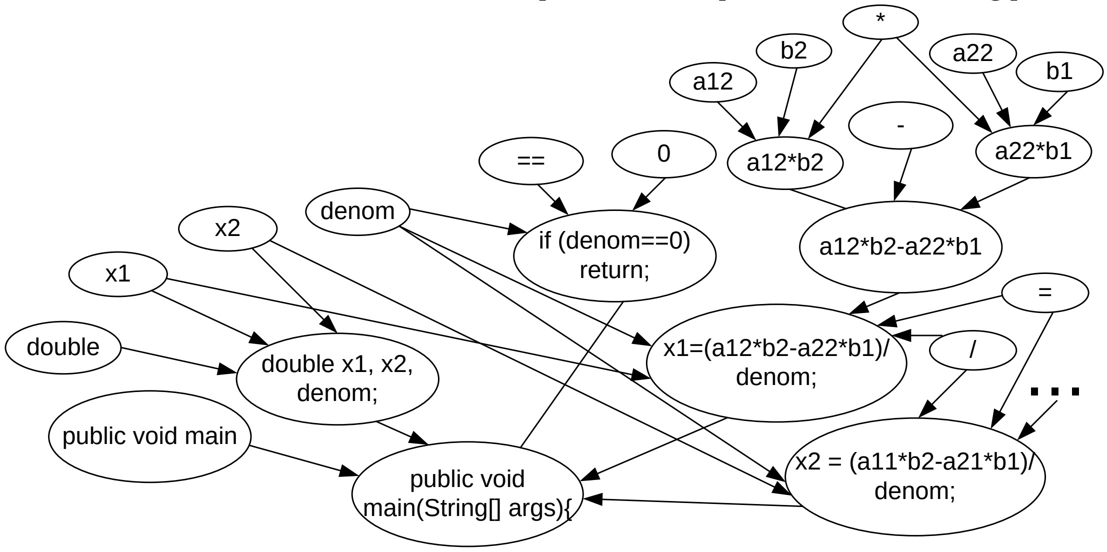
It is necessary to transform received AST algorithm in CCT which includes only interesting us AST nodes and their some characteristics. CCT represents hierarchical CC structure which the trainee should acquire for the linear equations system solving. Received CCT for a considered problem will have the following view:



Picture 4 - The fragment of problem CCT

CCT consists of two types nodes: terminal (not having descendants, private CC) and non-terminal (having descendants, compound CC). Nodes have an indicator of nesting level concerning root node. For example, the node with the text «a12» has the sixth nesting level concerning the uppermost node of a method main at which the given indicator is equal to 1. Each private CC has the corresponding weight factor defining degree of its importance in a problem solving concerning other components. The weight of compound CC is equal to the sum of weight factors of its components plus its own weight.

Apparently from the scheme represented in picture 4, CCT has some nodes duplicating each other located in different branches of a tree, for example terminal nodes “denom” or “*”. In BN made on the basis of CCT duplicated nodes unite in one node, thus a direction of connections between all nodes changes on opposite, i.e. from nodes with biggest nesting level to the smaller ones. Structure of a considered problem BN is presented in a following picture.



Pic. 5 – BN of problem solving

Described BN consists of chance nodes, each of which has two conditions: Satisfied (acquirement of CC) or Violated (absence of CC acquirement). Each node has the table of conditional probabilities connected with it (CPT). Values of the table for node A having n ancestors $a_1, a_2, ..., a_n$ are formed as follows:

P(A=Satisfied|$a_1,a_2,\dots,a_n$)=0.9, P(A=Violated|$a_1,a_2,\dots,a_n$)=0.1 at $a_1 \wedge a_2 \wedge ... \wedge a_n$=Satisfied;

P(A=Satisfied|$a_1,a_2,\dots,a_n$)=0, P(A=Violated|$a_1,a_2,\dots,a_n$)=1 at $a_1 \wedge a_2 \wedge ... \wedge a_n$=Violated.

Thus, on the basis of received statistics data, it is 90% of confidence that the trainee acquires compound CC if he acquires all its private CCs. In case the trainee does not acquire one of private CC it is obvious that he will not acquire also a compound component.

For example, for node «a12*b2», having three ancestors «a12», «*» and «b2», CPT will have the following view:

Table 1. CPT of «a12*b2» node

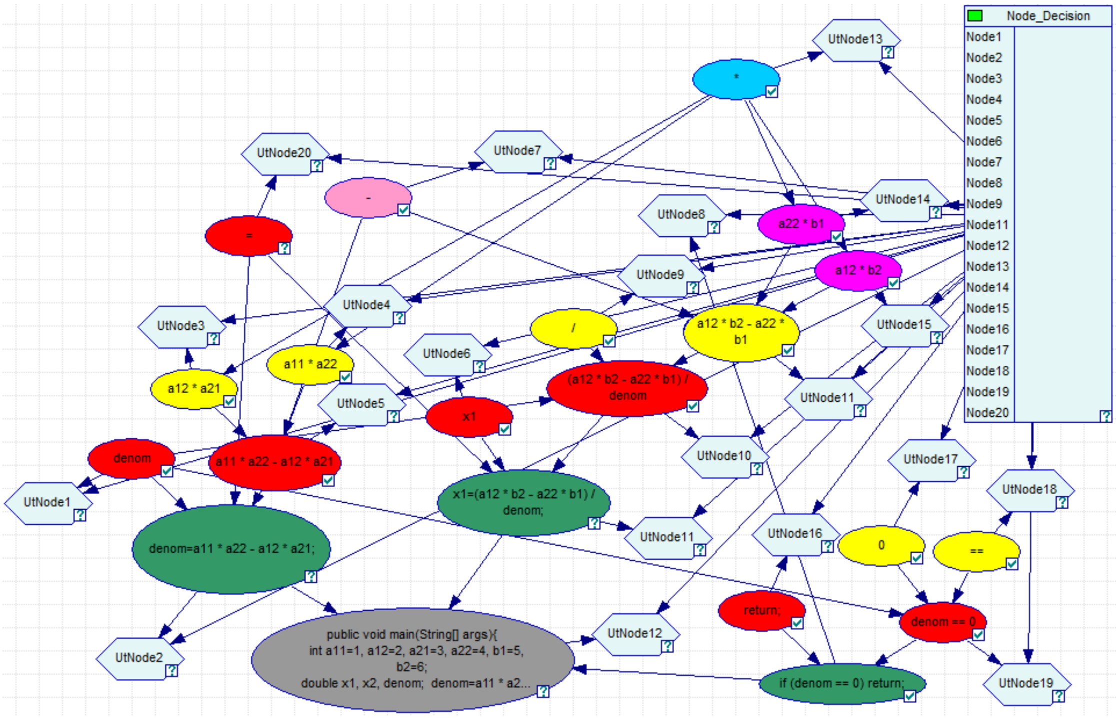
a12	Satisfied				Violated			
b2	Satisfied		Violated		Satisfied		Violated	
*	Satisf.	Violated	Satisf.	Violated	Satisf.	Violated	Satisf.	Violated
Satisfied	0.9	0	0	0	0	0	0	0
Violated	0.1	1	1	1	1	1	1	1

For implementation of function of a choice by system of the most useful next task execution step from the list of alternative variants, BN represented in picture 5, should be transformed in decision-making BN. For construction of the given network, and also possibility of solving of its various probabilistic inference problems, we will use the tools of platform-independent library SMILE. It is necessary to complement present BN with decision node which list of conditions (alternatives of a choice) will include all network chance nodes. Besides it, it is necessary to connect utility node corresponding to it with each chance node of a network. Utility values of a_i node if decision node has n alternatives a_1, a_2,…,a_n, are defined as follows:

Utility (a_i=Violated) = weight (a_i), utility values of other variants are equal 0,

where weight (a_i) - weight value of a_i node, equal to the sum weight of all its private nodes and initial weight value of a_i node.

Created through API SMILE interface decision-making BN can be seen by means of graphic editor Genie 2.0.



Picture 6 – The fragment of decision-making BN in Genie 2.0 editor

In picture 6 chance nodes of the ellipse form are shown by different colors depending on their nesting level in initial CCT. With each of such nodes the utility node is connected with names UtNode1 … UtNode20. The decision block "Decision Node" has 20 alternative conditions Node1 … Node20, which are corresponded by network nodes and CC associated with them.

For check of adequacy of the received model testing of it was performed in Genie 2.0 editor. Different variants of network nodes evidences sequences and the analysis of correctness of next step decision-making were modeled. One of such scenarios is presented in table 2.

Table 2. The description of the ICTP possible inner loop scenario execution

№	Current step	Result	Next step decision
1	public class EquatSystemSolving { public void main(String[] args){ int a11 = 1, a12 = 2, a21 = 3, a22 = 4, b1 = 5, b2 = 6; double x1, x2, denom; denom = a11*a22-a12*a21; if (denom==0) {System.out.println("No solution"); return;} x1 = (a12*b2-a22*b1)/denom; x2 = (a11*b2-a21*b1)/denom; System.out.println ("x1="+Double.to String (x1)+"; x2="+Double.toString(x2)); }}	Violated	x1=(a12*b2 - a22*b1) / denom; or x2=(a11*b2 - a21*b1) / denom;
2	x2=(a11*b2 - a21*b1) / denom;	Satisfied	x1=(a12*b2 - a22*b1) / denom;
3	x1=(a12*b2 - a22*b1) / denom;	Violated	(a12*b2 - a22*b1) / denom
4	(a12*b2 - a22*b1) / denom	Violated	(a12*b2 - a22*b1)
5	(a12*b2 - a22*b1)	Satisfied	denom=a11*a22 - a12*a21;
6	denom=a11*a22 - a12*a21;	Violated	a11*a22 - a12*a21
7	a11*a22 - a12*a21	Violated	a11*a22 or a12*a21
8	a11*a22	Satisfied	a12*a21

Inner loop execution begins with setting of the main network node evidence reflecting acquirement of whole equations system solving CC (method main), equal Violated. The system defines that the most possible reason of absence of this CC acquirement consists in absence of acquirement by definition of roots x1 or x2 values (nodes «x1 = (a12 * b2 - a22 * b1) / denom» and «x2 = (a11 * b2 - a21 * b1) / denom» accordingly). Further we model a situation when check of CC «x2 = (a11 * b2 - a21 * b1) / denom» acquirement shows positive result. The system in this case unequivocally defines that the possible lack of knowledge is covered in CC «x1 = (a12*b2-a22*b1)/denom». After given CC check, having negative result, the decision on error search in the right part of the given equality «(a12*b2 - a22*b1)/denom» is made. Further, after incorrect performance of the task for check of the last CC, the system defines numerator of fraction as the most probable place of the competence lack. The situation of correct performance of the task for check of numerator calculation is simulated. Search moves to CC of system determinant «denom=a11*a22 - a12*a21» calculations which further check leads to conclusion that a lack of knowledge is in nodes «a11*a22» and «a12*a21». In case of acknowledgement of «a11*a22» CC acquirement error search passes to «a12*a21» node. Thus, similar iterative search of the most probable place of trainee knowledge and skills lack with a choice of next step also forms dynamic sequence of TP task inner loop which performance is capable to produce to independent revealing of the reason and place of the made error, and also its correction. Hereupon due to realization of self-diagnostics principle the lack of required task CC acquirement is compensated.

MODEL TESTING IN REAL CONDITIONS OF EDUCATION

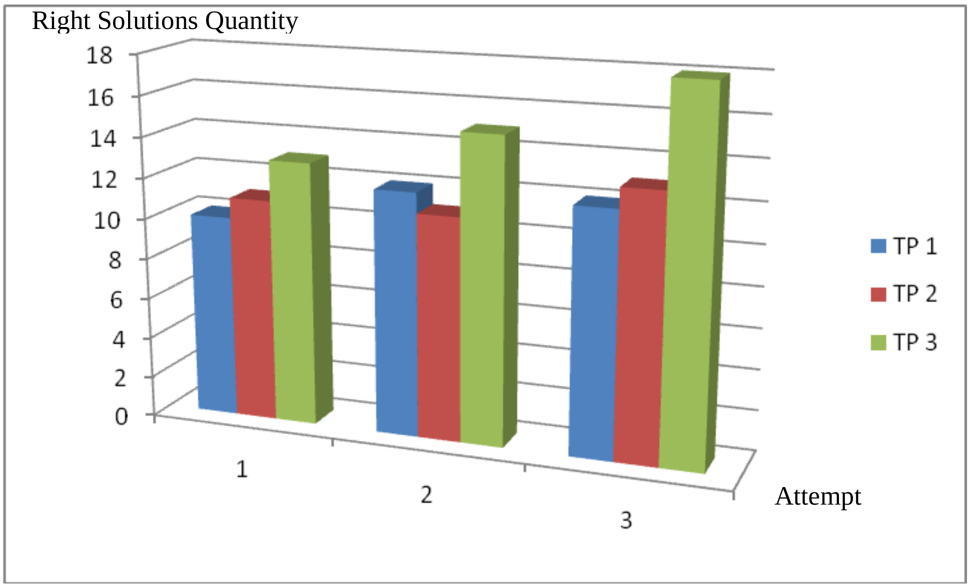
For check of efficiency of developed inner loop execution organization method was tested in real conditions of education. Three groups of students, 20 persons in each group were selected. Execution of TP task consisted in construction of linear equations system solving algorithm and performing of numerical calculations of its roots according to the given values of its factors. Task execution correctness was checked by comparison of results of constructed student's algorithm execution with results of work of etalon algorithm on some test set of the input data. Each of groups of students were given for performance the following linear equations systems solving TP:

1) TP task requires one attempt of its solution, thus after check of correctness of task execution it is informed the general result of its performance: correctly or incorrectly;

2) TP task supposes execution of three attempts of its solving, thus after each wrong attempt the trainee is notified on the fact of presence of an error in task execution;

3) TP task realizes the idea of inner loop described in the present article. The message of wrong task execution can be received only in case of wrong performance of all steps of task execution.

TP contained three task copies with different values of linear equations system factors. Sets of factors values in different TP kinds were identical. As a result of the performed testing following results of task execution success for each of TP kinds, represented on picture 7 have been received.



Picture 7 - Comparative results of linear equations system solving TPs execution

Apparently from the diagram, TP with inner loop (TP 3) has shown the greatest learning efficiency because as a result of third problem copy execution we have achieved that 90 % of students have solved a problem correctly. Thus the first and second TP kinds as a result of third problem copy execution have shown 60 and 65 % of success accordingly. The developed inner loop execution organization method allows providing engineering problems learning efficiency increase, such as linear equations system solving, practically on 30 % in comparison with traditional methods of TP task execution organization.

CONCLUSIONS

In the given research choice of approach to formation of ICTP pedagogical actions sequence, and also development of a method of automatic construction of decision-making model for performance of its inner loop was performed. The principle of developed method has been investigated on an example of two linear equations system solving. According to the made algorithm of system solving its program implementation in java-code form has been created. The program code has formed a basis for algorithm AST formation. AST has been transformed in CCT by identification of nodes in it, associated with necessary CC. On the basis of last tree BN of CC has been constructed. The final stage was inner loop decision-making BN construction for considered problem. Testing of created decision-making BN in graphic Genie 2.0 editor at different scenarios of task steps execution has confirmed adequacy of the received model and its ability to compensate the insufficient trainee competence due to the correct choice of next task execution step. Compensation is reached due to independent revealing by the trainee of error reason and character, and also its correction. Thus, such important pedagogical principles of education as activity, deliberateness and the self-diagnostics, allowing increasing efficiency of computer tutoring are implemented.

REFERENCES

Y.A. Komenski, D. Lock, J.-J. Rousseau and J.H. Pestalozzi (1989). "Pedagogical heritage." *Moscow: Pedagogics*

VanLehn K (2006). "The Behavior of Tutoring Systems." *International journal of Artifical intelligence in education*, Volume 16, Issue 3, 227-265.

Vigotsky L.S (2005). "Psychology of the human development." *Moscow: Publishing house "Smisl", Publishing house "Eksmo".*

GeNIe. "Decision Systems Laboratory." University of Pittsburgh. http://genie.sis.pitt.edu/The Eclipse Foundation. http://www.eclipse.org/

E. Gamma, R. Helm, R. Johnson and J. Vlissides (1995). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.

Russell, Stuart J. and Norvig, Peter (2003). "Artificial Intelligence: A Modern Approach (2nd ed.)." *Upper Saddle River, New Jersey: Prentice Hall, ISBN 0-13-790395-2*

R. Charles Murray, Kurt Van Lehn and Jack Mostow. "Looking Ahead to Select Tutorial Actions: A Decision-Theoretic Approach." *International Journal of Artificial Intelligence in Education*, 14(3, 4), 235-278.

K. R. McNaught, A. Zagorecki. "Using Dynamic Bayesian Networks for Prognostic Modelling to Inform Maintenance Decision Making." Department of Engineering Systems and Management, Cranfield University, Defence Academy of the UK, Shrivenham, Swindon, UK.

Mitrovic, A. (2003), "An intelligent SQL tutor on the web", *International Journal of Artificial Intelligence in Education*, 13, 171–195.

Murray, R.C. and Van Lehn, K. (2000). "DT Tutor: A dynamic, decision-theoretic approach for optimal selection of tutorial actions." *G. Gauthier, C. Frasson and K. Van Lehn, Intelligent Tutoring Systems, Fifth International Conference*, ITS 2000, 153-162.

Kulik A., Chukhray A. and Chukhray M (2006). "Diagnostic models of intelligent tutor system for teaching skills to solve algebraic equations." *Proceedings of the Interactive Computer Aided Learning Conference*, Villach, Austria, Sept, 26-28, *International Journal of Emerging Technologies in Learning.*(2007), Vol. 2, No.1, http://www.i-jet.org

Rakov S., Kulik A., Chukhray A. and Pedan S (2008). "Models and software for intelligent web-based testing system in mathematics*"* *Proceedings of the East-West Fuzzy Colloquium*, Zittau, Germany.

Kulik A., Chukhray A., Pedan S. and Anzenberger P (2008). "Development of the automated laboratory practical work at the course «Modeling of systems»" *Proceedings of the Interactive Computer Aided Learning Conference*, Villach, Austria, Sept. 24-26.

Chukhray A.G., Pedan S.I. and Vagin Ie.S (2010). "Modeling of engineering skills acquirement process via dynamic Bayesian nets" *Proceedings of the East-West Fuzzy Colloquium*, Germany, Zittau, IPM, 257-263.