

---

## A COMPUTER-BASED STRATEGY FOR PRACTICAL SKILLS TEACHING

Polina Atanasova, Irina Zheliazkova, Avram Levi

*Abstract:* An adaptive learning technology embedded in e-learning environments ensures choice of the structure, content, and activities for each individual learner according to the teaching team's domain and didactic knowledge and skills. In this paper a computer-based scenario for application of an adaptive navigation technology is proposed and demonstrated on an example course topic.

*Keywords:* adaptive technologies, navigation graph, problem solving, programming skills, teacher, learner

*ACM Classification Keywords:* Computer and Information Science Education, Knowledge Representations of Formalisms and Methods

*Conference:* The paper is selected from Third International Conference "Modern (e-) Learning" MeL 2008, Varna, Bulgaria, June-July 2008

---

### Introduction

Achieve success in teaching is more likely if the approach of active learning is applied that means to engage the learner (*L*) in the learning process. In learner-centered education, learners should be engaged in active exploration, be intrinsically motivated, and develop an understanding of a domain through challenging and enjoyable problem-solving activities. By means of advanced *ICT* it is possible to expand experience, take a power interaction, create dialectic between experience and dialogue support, investigate and develop active learning itself. Nowadays the growing emphasis on active learning has yielded substantial changes in educational theory and practice revolving around constructivism, which emphasizes the *L*'s knowledge construction process [Tuparov G., 2007]. There are four modes of learning when dialogue with self and with others is coupled with experience of observing and doing. Each of these modes has its own value, and just using more of them should add variety and thereby be more interesting for the learners. When properly connected, the various learning activities can have an impact that is more than additive or cumulative; they can interactive and multiply the educational impact.

In an earlier paper [Zheliazkova et al., 2005] we proposed a computer-based scenario of a test like practical exercise consisting of a given cognitive type of tasks. The strategy embedding in educational environments leads to a combination of the learning modes. Besides concentration of the *L*'s attention in order to achieve success such kind of exercise allows an intelligent analysis that can provide the *L* with extensive error feedback and update the course materials, e. g. background material (BM), tests, exercises, and so on [Brusilosky P., 1999].

In the present paper presents an improvement of this scenario adding a new higher cognitive level, namely skills for programming tools and demonstrate its practical application on an example topic of the course "Discrete Structures and Modelling" taught the second year students specialty "Computer Systems and Technologies" at Rousse University. The rest of the paper is organized as follows. In the next two sections the adaptive navigation graphs for the course teaching team (*TT*) and *L* are presented. The rest three sections are devoted to their realization in the course topic "Modelling Boolean method for patterns recognition", namely: teaching the background knowledge, solving the practical problems, and programming skills development. The conclusion outlines the paper contributions and authors' plans for a near future.

---

### Exercise Planning by the Teaching Team

A *TT* consists of an author (*A*) leader of the team and responsible for the preparing of topic BM, test questions and exercise tasks covering it. The *T* is a member/members of the team responsible for planning, monitoring, and assessing the test and exercise of a group of learners. On fig. 1 the *TT*'s navigation graph for planning a test like

exercise, consisting of a given cognitive type of tasks is presented. Here subgraphs  $SG1$  и  $SG2$  correspond to the topic  $BM$  and pretest preparing. The way that the topic material is performed by the  $A$  or an educational tool does not matter. However it is supposed that each topic material entity (module, topic, subtopic, and so on) has a several positions code identificator, consisting of the numbers of the module, topic, and subtopic. The pretest has to guarantee the coverage of the topic material at the level of reproduction and understanding. The task identifier is received adding the current number to the topic identifier and links to the relevant pages with the topic material. The state  $OT$  on fig. 1 corresponds to the standard dialog window "Open/New" for reading the subprogram describing the  $A$ 's domain knowledge for problem solving in a specialized script language. The next state  $FT$  contains the task formulation with its computed parameters values. Depending on the task type the  $A$ 's domain knowledge can be extracted by means of one or more dialog forms ( $DF_1, \dots, DF_k$ ). The end of the  $A$ 's work is depicted with the state  $ST$ , corresponding to the standard dialog window "Save/Save As" for storing the subprogram file describing the  $T$ 's didactic knowledge about exercise planning in a exercise independent script language. Two directed arrows reflects the flexible navigation between the above mentioned states for editing the current file. The feedback from  $ST$  to  $OT$  reflects the preparing a set of exercise tasks. The states involved in the second cycle have the following meanings:  $OE$  - "Open/New" - reading the program file describing the  $T$ 's didactic knowledge foe individualized exercise planning;  $AD$  - editing the exercise administrative data;  $LD$  - editing a list with key directives for virtual  $T$ 's intervention during the exercise performance by the  $L$ ;  $CT$  - editing the list of tasks included in the individualized exercise;  $CA$  - editing the criteria for assessment of the exercise tasks;  $SE$  - "Save/Save As" - storing the program exercise file. The feedback from  $SE$  to  $OE$  here has a meaning of preparing an individual exercise for each  $L$ .

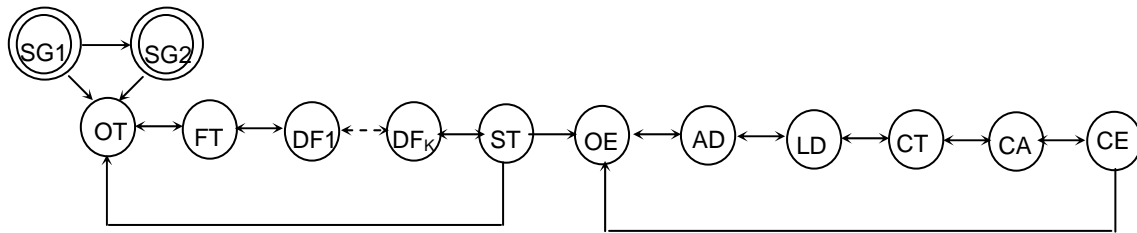


Fig. 1. Adaptive navigation graph of preparing the exercise by a course team

### Exercise Performing by the Learner

The way of teaching background knowledge (whether by a human  $A$  or by means of an educational system) does not matter that is presented on fig. 2 with the subgraph  $SG1$ .

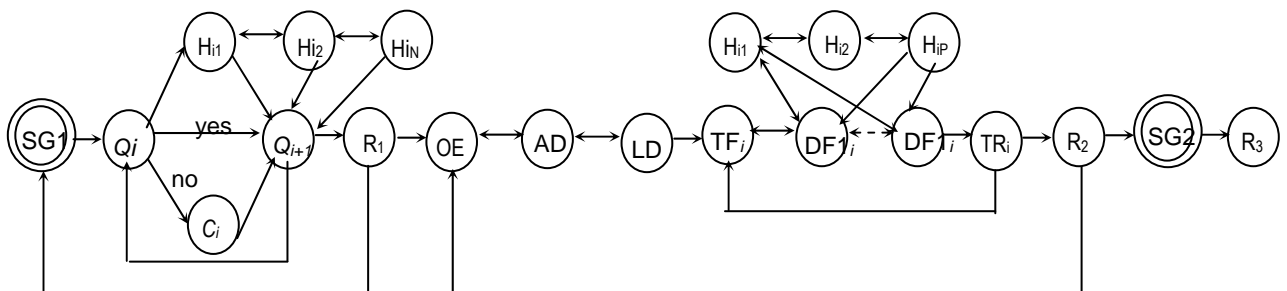


Fig. 2. Adaptive navigation graph for performing the exercise by the learner

The pretest performing by the  $L$  before the exercise is to assess his/her background knowledge and to fill the gaps in it or for remediation of the misconceptions. *t.e.* executes also a teaching function. That is why the state  $Q_i$  instead of two transitions has tree ones. The third the pages  $H_{ij}$  ( $j=1, \dots, N$ ) from the topic material, having links with the corresponding question. The feedback from  $Q_i$  to  $Q_{i+1}$  has a sense of presentation of the set of questions, and that one from  $R_1$  to  $SG_1$  when the test result  $R_1$  is lower than the desired threshold in other case the  $L$  is

allowed to the exercise. In the dialog window *OE* he/she has a possibility to read only the content of the dialog windows *AD*, *LD* and *CA* as well as the content of the task formulation ( $TF_i$ ) for the current task. In next dialog state ( $DF_j$ - $DF_M$ ) the domain knowledge for task solving are extracted by the L in a way similar to that of the A. The result of current task performance  $TR_i$  has a feedback to  $TF_i$  till all tasks are performed or the time planned is over. If directive "Help" had been chosen by the T (state *CT* from fig. 1), during the solution the L is accessible to the relevant topic material pages  $H_{ij}$  ( $j=1, \dots, P$ ). The exercise finishes with the exercise report  $R_2$  that can be viewed both the L and T. The feedback from  $R_2$  to *OE* is included in the case, when the L's mark is lower than the threshold setup by the T (state *CA* from fig. 1).

---

### Teaching the Background Knowledge

---

Modeling the patterns recognition is one of the module which purpose is to give general information concerning four patterns recognition methods, namely: systems of Boolean and systems of linear functions, probability and perception method [Too D., 1978]. In their classical and simple form these methods suppose that the classes of patterns  $\omega_i$  ( $i = 1, \dots, M$ ) are preliminary known. The focus of the BM [Zheliakova I., 1998] is on the formal and definition of the classification task as well as on the methodology of modeling the patterns recognition. If the whole set of the patterns is denoted with  $V$ , then the patterns classifications formally is reduced to seeking for a function  $f : V \rightarrow \prod(V)$  such that  $\bigcup_{i=1}^M \omega_i = V$  и  $\omega_i \cap \omega_j = \emptyset$  за  $i \neq j$ . The classification requires a

measure of similarity  $L(x_i, x_j)$  between the patterns  $X_i$  и  $X_j$ , meeting the following requirements: 1)  $L(X_i, X_j) \geq 0$ , e. g. to have non-negative values; 2)  $L(X_i, X_j) = L(X_j, X_i)$ , e. g. to poses the patterns symmetry; 3)  $L(X_i, X_j) = \max L(X_i, X_j)$ , e. g. the measure of similarity of a pattern with itself to be maximal in comparison with each other pattern; 4) to be a monotonous function of points in  $n$  – dimensional space. They are demonstrated on an example task using the Boolean functions method. In comparison with the other three methods the Boolean method is more simple and understandable. It assumes that the pattern's attributes have only logical values, e.g.  $1(x_i)$ , if the pattern has the corresponding attribute, and  $0(\overline{x_i})$ .

The choice of the measure depends on the used method, way of patterns coding, and concrete task for recognition. As in case of the Boolean functions method these requirements are satisfied automatically the demonstration is on the similarity between rectangle and ellipse with circle. The methodology of patterns recognition modeling is presented graphically as a flow chart shown on fig. 1. Bellow its main stages shortly are commented on an example of a real life application task having a good graphical visualization, and illustration of the human logical reasoning development.

Task formulation: Let's 5 classes of patterns are known, each one corresponding to one of the digit from 0 to 4. The example patterns are measured in a binary matrix with 5 x 5 elements, where  $x_i$  takes value 1, if the digit representation crosses the  $i$ -th matrix cell, and 0 in other case. The number of the classes will correspond to the number of the digits, namely: 0 ( $\omega_0$ ), 1 ( $\omega_1$ ), 2 ( $\omega_2$ ), 3 ( $\omega_3$ ), and 4 ( $\omega_4$ ). The considered task concerns only one position whereas the Bulgarian postal code consists of four positions.

Choosing the attributes: In case of the formulated task the total number of the attributes is 5 x 5 = 25. The measurement vector of 25 elements can be reduced to 5, if more informative attributes are chosen:  $x_1$  - 1 horizontal line;  $x_2$  - 2 horizontal lines;  $x_3$  - 3 horizontal lines;  $x_4$  - 1 vertical line;  $x_5$  - 2 vertical lines. A horizontal line is received when horizontally more than one neighbor attributes have value 1. A vertical line is received when vertically more than one neighbor attribute have value 1 (fig. 2).

Composing the Boolean functions: The system of Boolean functions  $f_i(x_1, x_2, \dots, x_n)$  ( $i = 1, \dots, M$ ) is composed analyzing the logical relationships among the attributes and/or classes. The simpler initial case is

when the number of the functions is equal to the number of the classes and the system is independent, e.g. practically for each combination of the attributes values only one function is equal to 1

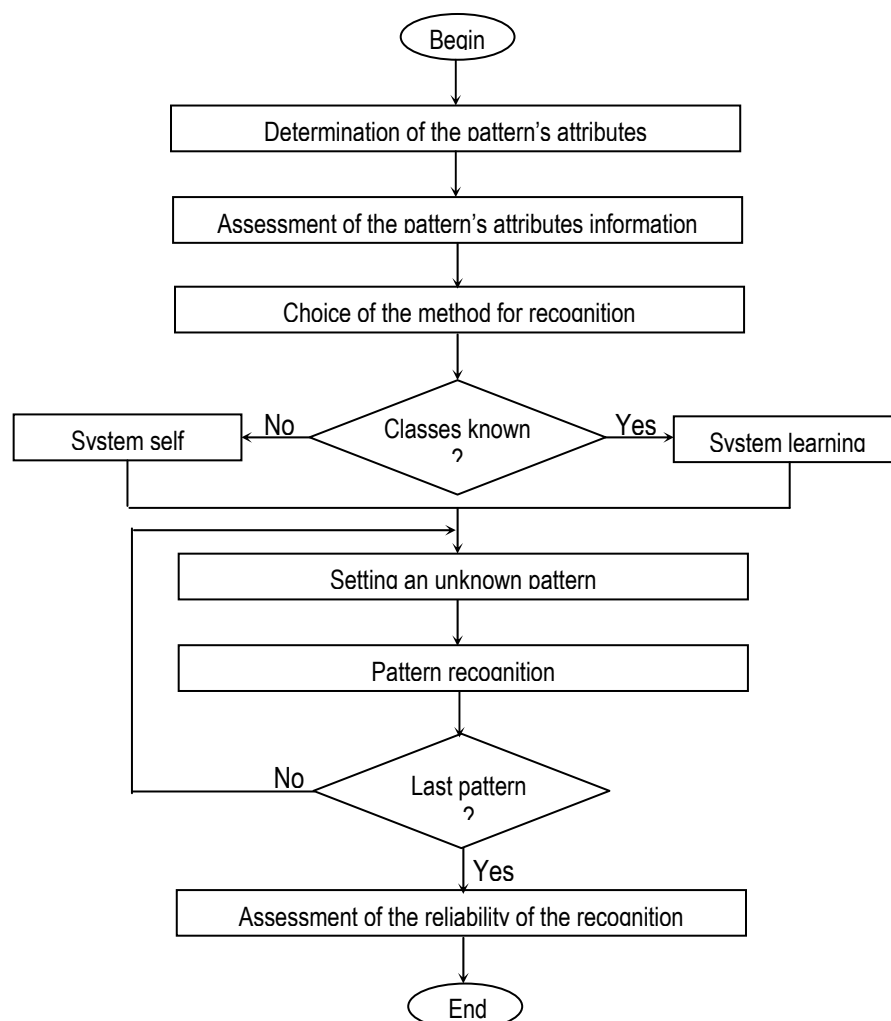


Figure 2. The measurement matrix Figure 1. The flow chart of the methodology of patterns recognition modeling

$$f_0 = x_2 \cap x_5; \quad f_1 = x_4 \cap \bar{x}_3; \quad f_2 = x_3 \cap x_5; \quad f_3 = x_3 \cap x_4; \quad f_4 = x_1 \cap x_5$$

Each function expresses a complex true assertion for the corresponding class by means of an appropriate combination of the well-known Boolean operations such as *and*, *or*, *not*, and *equivalence*. For the example task this system consists of 5 functions and includes 2 operations (*and*, *not*).

Recognition of unknown patterns: A simple decision rule is used for this purpose:  $x \in \omega_i$ , if  $f_i = 1$  and  $f_j = 0$  where  $j \neq i$ . Let's the unknown coded digit pattern is  $x = ((11111), (10001), (10001), (10001), (11111))$ . For the corresponding combination of the attributes' values, e.g.  $x_1 = 0; x_2 = 1; x_3 = 0; x_4 = 1; x_5 = 1$ ; the functions' values are computed (Table 1). As only  $f_0$  has value 1, e.g. the conclusion is that the unknown pattern belongs to the class  $\omega_3$ , e.g. the coded digit is 0. The correctness of the system of Boolean functions has to be checked for all other matrixes (fig. 2), e.g. for  $x = 01011; x = 10010; x = 00101; x = 00110; x = 00111$ .

Table 1

$x = 01011$	$x = 10010$	$x = 00101$	$x = 00110$	$x = 00111$
$f_0 = 1 \cap 1 = 1$	$f_0 = 1 \cap 1 = 1$	$f_0 = 1 \cap 1 = 1$	$f_0 = 1 \cap 1 = 1$	$f_0 = 1 \cap 1 = 1$
$f_1 = 0 \cap 0 = 0$	$f_1 = 0 \cap 0 = 0$	$f_1 = 0 \cap 0 = 0$	$f_1 = 0 \cap 0 = 0$	$f_1 = 0 \cap 0 = 0$
$f_2 = 0 \cap 1 = 0$	$f_2 = 0 \cap 1 = 0$	$f_2 = 0 \cap 1 = 0$	$f_2 = 0 \cap 1 = 0$	$f_2 = 0 \cap 1 = 0$
$f_3 = 0 \cap 1 = 0$	$f_3 = 0 \cap 1 = 0$	$f_3 = 0 \cap 1 = 0$	$f_3 = 0 \cap 1 = 0$	$f_3 = 0 \cap 1 = 0$
$f_4 = 0 \cap 1 = 0$	$f_4 = 0 \cap 1 = 0$	$f_4 = 0 \cap 1 = 0$	$f_4 = 0 \cap 1 = 0$	$f_4 = 0 \cap 1 = 0$

The computations can be distributed among several teams (in our case 5) in order to support their activity and interest and accelerate the performance of this stage.

Learning the system: In case of an unrecognized pattern, for example, corresponding to matrixes of the digit 5 or 6, two additional classes ( $\omega_5$  and  $\omega_6$ ) respectively functions  $f_5$  and  $f_6$  have to be introduced. Also the functions can be changed introducing new attributes and/or logical operations. In our case the attribute  $x_6$ , meaning the state (0 or 1) of the matrix cell 4,1 can be added.

Determination of the probability recognition: Let's the number of the incorrect recognized patterns is  $m$ . Then the method probability recognition will be computed as a percentage from the total number of the patterns  $n$ , e.g. as  $(m/n) \cdot 100$ . If the required value of the system probability has to be 100% the system of the Boolean functions could be totally changed. In our case, for example, introducing new logical operations the unknown patterns present digits 7,8, and 9 will be recognized. Respectively that means the final system will consist of 10 functions.

---

### Solving Problems in Practical Exercises

---

The practical exercises of the course are provided in a laboratory equipped with power computers and two specialized educational tools: a specialized intelligent multimedia environment for knowledge testing, and a WINDOWS-based tool for modeling the patterns recognition by means of the Boolean functions method. By means of this environment at the beginning of the exercise each performs an on-line pretest covering the topic material. The test checks precisely the student's knowledge at the level of remembering and understanding. The IMEKT helps the assistant in precise assessment of each test question and the test as a whole. On fig. 3 an example student's report is presented. In column "Answer" different types of question answers, e.g. such as ordered keywords, unordered couples, unordered keywords are shown. For each test question the student's answer, scores, and link to the corresponding lecture fragment, coefficient of proximity of the student's answer to the author's one are given. The total points and time spent also are displayed. On fig. 4 the names and faculty numbers in the student's subgroup intended to carry out the exercise together with their time results and the average mark.

The students assessed with the mark different from 2 begin to work in teams of 2-3 students. The assistant demonstrates the technology of using the specialized WINDOWS-based tool on a simple task for software company staff. In this case the classes correspond to 4 different kind of staff, namely:  $\omega_1$  - manager;  $\omega_2$  - analytic;  $\omega_3$  - developer;  $\omega_4$  - programmer and the attributes are 3, namely:  $x_1$  - makes decisions;  $x_2$  - has a boss;  $x_3$  - implements programs [Zheliakova I., 1998].

Task№	Answer	Scores	Help_information	d
1	данны>предсказуем>различен	2.00	1.1	0.33
2	2>r;1>6;4>в;3>a	12.00	1.2	1.00
3	a>r>в>б	2.00	1.3	0.50
4	a>b>d>	0.00	1.4	0.00
5	private;public;protected	3.00	1.5	1.00
Total Points: 19.00				
Mark: 4				
Start_Time: 21:13:33				
End_Time: 21:39:05				
Total Time: 26				

Figure 3. The student's report

Under the assistant monitoring the students' teams in an arbitrary order perform the rest more complex exercise tasks. As a rule one of the most interesting task is that for the triangles recognition based on their sides lengths ( $a, b, c$ ) due to introducing arithmetical operations *equivalence* and *non-equivalence*. The decision is simplified by substitution e.g.  $x_1 = (a = b)$ ;  $x_2 = (b \neq c)$ ;  $x_3 = (a = c)$ . Then the Boolean functions corresponding to the patterns classes ( $\omega_1$ -equilateral,  $\omega_2$ - isosceles и  $\omega_3$ - scalene) are written in a general form given below on the right.

$$\begin{array}{l}
 f_1 = x_1 \cap \overline{x_2} \cup x_3 \cap \overline{x_2} \cup x_1 \cap x_3 \\
 f_2 = x_1 \cap \overline{x_3} \cup \overline{x_2} \cap \overline{x_1} \cup x_3 \cap \overline{x_1} \\
 f_3 = \overline{x_1} \cap x_2 \cap \overline{x_3}
 \end{array}
 \quad \left| \quad
 \begin{array}{l}
 f_1 = 0 \cap 0 \cup 0 \cap 0 \cup 0 \cap 0 = 0 \\
 f_2 = 0 \cap 1 \cup 0 \cap 1 \cup 0 \cap 1 = 0 \\
 f_3 = 1 \cap 1 \cap 1 = 1
 \end{array}$$

In order to identify a given triangle with sides' length  $a = 3, b = 4, c = 5$  these data are replaced in the above-listed attributes and then the received Boolean values are replaced in the system of functions above on the right. As the value of the function  $f$  is equal to 1 the conclusion is that, the unknown pattern belongs to the class  $\omega_3$  - scalene triangles. This task then is modified according to a binary tree classification. In this case 3 additional attributes are added:  $x_4$  - has three acute angles;  $x_5$  - has an orthogonal angle;  $x_6$  - has an obtuse angle. The number of the functions increased to 12, corresponding to a triangle like orthogonal isosceles, acute scalene, and so on.

In an order different for different teams all classes are checked with at least one pattern. This step can be performed at once or class-by-class and in a random order by the different teams. The solution with minimal number of the attributes and/or operations is optimal. The time spent on each tasks is registered in the student's team exercise *WORD* document prepared in a highly interactive way [3].

## Programming Skills Development

The goal of the student's course work that is individual is to raise her/his skills to design, implementation, and documentation of an original WINDOWS-based application for a given patterns recognition method. The student has freedom in choice of the programming language and environment, and the concrete problem to be solved. The list of the requirements for the tool includes but not limited to: text and graphic mode of the monitor; sets and arrays for data representation in the computer memory, text and/or record file for data storing into the disk, Boolean operations and operations on sets for data processing.

Student's name	Fak No	Points	Mark	Questions' time, min	Total time, min
Ivan Ivanov	22222	23.00	2	1.33	2.00
Гаял Иванова	63344	66.00	5	9.03	14.00
Average mark: 3.5					

Figure 4. The student's group report

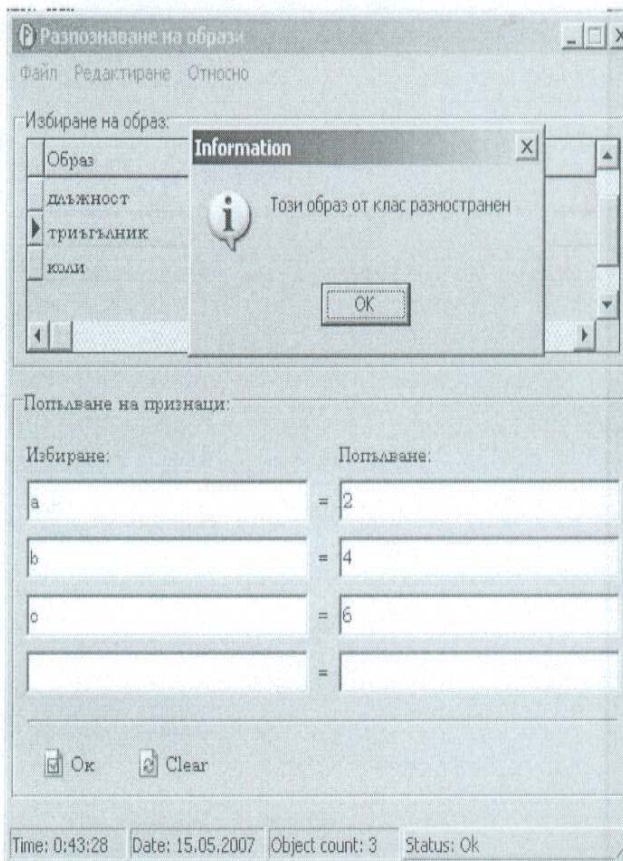


Figure 6. Screenshot with a successful pattern recognition

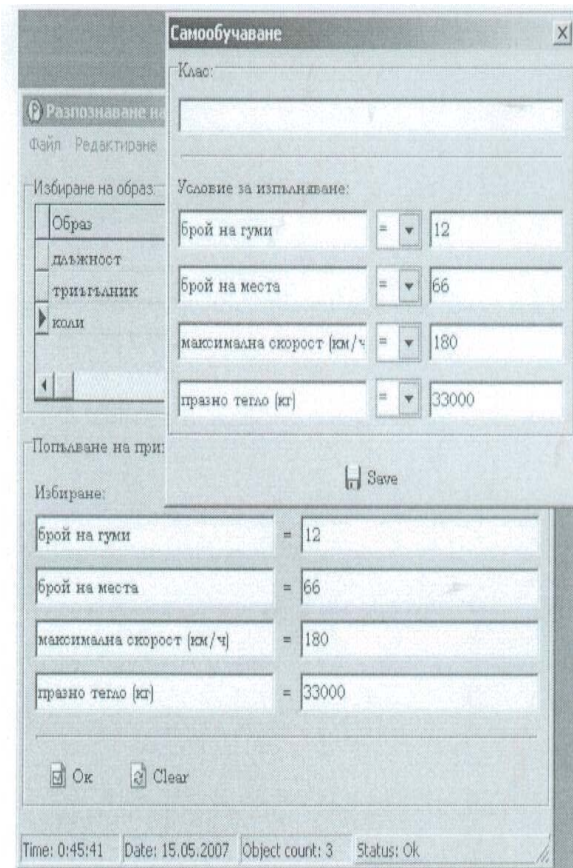


Figure 7. Screenshot with a new class

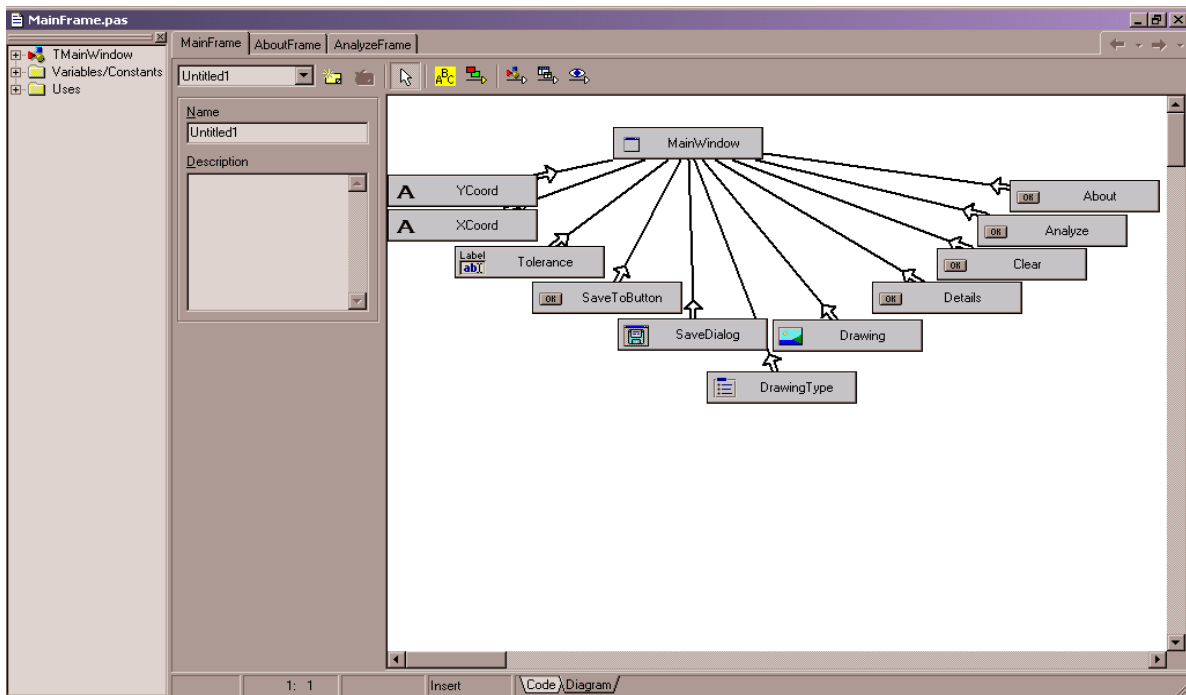


Figure 8. The generated class diagram of a student's project in DELPHI 7.0

The report has to contain: title page, contents, problem formulation, program project, modules description, instructions for installing, running, and using the tool, as well as the source code. Again the accumulated experience has shown that in comparison with the other three methods taught the Boolean method is preferred due mainly to its simplicity, understandability, and a lot of practical applications.

Among the visual programming environments like DELPHI, VISUAL BASIC, and VISUAL C++ the first one is preferred due to its power component library, intuitive visual programming, support of different databases, generation of the class diagrams, and so on. Fig. 8 present a generated class diagram of a student's project implemented in DELPHI programming environment and evaluated with 10.

The student's course work is assessed by the lecturer depending on the degree of fulfillment of the above-listed requirements. Each unsatisfied requirement decreases the mark with a score. In case of other features such as support of database, usage of object-oriented programming, visual programming, and so on the evaluation of the course work can be increased to 10. The final six-scale mark is computed as an average of the test, exercise, and coursework mark.

---

### Acknowledgments

---

The authors are grateful to academic year 2006/2007 students' specialty "Computer Systems and Technologies" at Rouse University for their motivation, interest, and creativity as the first participants in the application of the described scenario in the topic "Modeling for patterns recognition by Boolean method". Without their invaluable help this paper could be impossible.

---

### Conclusion

---

The interactive interface for extracting the L's domain knowledge for problem solving is close to the A's one and the last is separated from the T's didactic knowledge. The presented scenario ensures the learner's progress from the background knowledge trough solving the practical tasks to tools programming skills. The accumulated experience has shown that it contributes the development of the student's activity and motivation, teacher's skills for exercise planning and monitoring as well as for his/her teamwork with the course author.

---

### Bibliography

---

- [Brusilovsky P., 1999] Adaptive and Intelligent Technologies for Web-based Education. In: C. Rollinger and C. Peylo (eds.), Special Issue on Intelligent Systems and Teleteaching, Künstliche Intelligenz, 4, 19-25.
- [Too D., 1978] Gonzales P. Principles of patterns recognition, Mir, Moskow, 1978 (in Russian).
- [Tuparov G., 2007] D. Tuparova, Modeling of Adaptive learning Scenario in E-Learning Environments, In: Proceedings of E-Learning Conference'07, 27-28 August, Istanbul, 2007, 42-47.
- [Zheliaskova I., 1998], S. Stefanova, I. Valova, G. Georgiev, S. Kalinova, Manual for problems solving in discrete structures, part 1, Rouse University, 1998 (in Bulgarian).
- [Zheliaskova I., 2005] P. Valkova, A Strategy for Practical Teaching in Task-Oriented Design Environments, In: Proceedings of the Annual Scientific of the Rouse University, 44, Seria 6.1, 2005, 107-112 (in Bulgarian).
- 

### Authors' Information

---

*Polina Atanasova* – PhD student, University of Rouse, Studentska street 8, Rouse 7017, Bulgaria;  
e-mail: [valkova\\_99@yahoo.com](mailto:valkova_99@yahoo.com)

*Irina Zheliaskova* – Associate Professor; University of Rouse, Studentska street 8, Rouse 7017, Bulgaria;  
e-mail: [irina@ecs.ru.acad.bg](mailto:irina@ecs.ru.acad.bg)

*Avram Levi* – Associate Professor, University of Rouse, Studentska street 8, Rouse 7017, Bulgaria;  
e-mail: [ALevi@ecs.ru.cad.bg](mailto:ALevi@ecs.ru.cad.bg)