UDC 004.8: 004.421.2
DOI: 10.20535/SRIT.2308-8893.2023.2.03

INFORMATION TECHNOLOGY FOR CREATING INTELLIGENT COMPUTER PROGRAMS FOR TRAINING IN ALGORITHMIC TASKS. PART 2: RESEARCH AND IMPLEMENTATION

A.S. KULIK, A.G. CHUKHRAY, O.V. HAVRYLENKO

Abstract. Information technologies, particularly artificial intelligence methods, involve more and more deeply into all spheres of human activity: science, technology, art, and education. Ukraine also has sufficient potential and needs to develop educational support, which is the subject of this paper. The work aims to demonstrate the components of information technology for the creation of Intelligent Tutoring Systems (ITS), which are involved in studying various engineering disciplines. The work includes methods of system analysis, mathematical and simulation modeling, technical diagnostics, and artificial intelligence. The proposed models and methods are implemented in ITS prototypes for teaching mathematics, programming, and the automatic control theory. The Intelligent Tutoring Systems were implemented in the educational process of KhAI University and other institutions in Ukraine, Great Britain, Austria, and China. Experimental studies have shown increased student learning success rates using ITS compared to traditional methods. Improved and adapted for computer training methods of technical diagnostics, Bayesian networks, and developed models of algorithmic tasks, the learning process and the learner are valuable from a scientific point of view. In a practical sense, the obtained results can be used to create new specialized ITSs and build an expandable common learning platform integrating the basic disciplines of a specific educational field.

Keywords: intelligent tutoring systems, experimental studies, results of implementation.

INTRODUCTION

The implementation of ITS is one of the highest priority directions of educational tools evolution. This fact is reasoned by numerous advantages of ITS usage over the classical approach: adoption for particular student, wide possibilities of virtual modeling of real objects and processes, decreasing of time and work efforts for completion and verification of tutoring courses, e-learning facilities, etc.

ITSs are characterized by supporting of inner and outer tutoring loops, minimal feedback (prompting hints and advices), nonlinear learning path, dynamic and customizable knowledge base, and self-learning support [1]. ITS usually include three main structural elements: a domain model, a student model and a pedagogical model, but researchers also often incorporate an interface model as the fourth element. Getting into account structure and functionality complexity of ITS, we can conclude that development process requires huge efforts and deep knowledge in programming and the subject area. One of the actual problems regards to providing of adaptive hints in the context of certain subject area. A hint must be generated according to the place and type of student's mistake and assumes possibility of alpha and beta errors. Moreover, it should provide numerous rules (constraints) for each type of a student's mistake. Program implementation of such as laborious process usually requires a high degree of programming skills. In well-structured domains where solution of the tutoring task can be modeled as a tree, it is possible to skip manual rule generation (constraints) and automatically perform pedagogical decisions.

The goal of the presented work was to develop ITSs for different domains on the basis of system approach [2; 3], models, methods and tools, which were described in part 1 of the paper [4] and to carry out their experimental studies.

EXPERIMENTAL STUDIES

Experimental studies of the proposed method [4] were performed using m-ary trees for the metrics of K. Tai, K. Zhang, D. Shasha – the minimum editing distance between m-ary trees. The performance of the developed NearestHash method was compared with the speed of the "naive" method and the method of Bustos, Navarro, Chavez (BNC).

Fig. 1 shows a comparison of the performance of the NearestHash method with the performance of the BNC method (basic and modified) by input conditions [4]. In the experiment for 10 and 100 trees, the first 10 trees from the original list of trees were searched, and in the experiment with 1000 – the first 50. As can be concluded from the figures, at the first search the developed method surpasses the known and modified known for the first experiment from 10 trees 1.71 and 1.67 times; for the second experiment out of 100 trees – 1.66 and 1.69 times and for the third experiment out of 1000 trees – 3.8 and 2.3 times.

Consider the formation of the set $SSim2$ [4] based on bigrams, i.e. $q=2$. Let it is given the strings $s0 = s0[1]\cdot s0[2]\cdot s0[3]\cdot\ldots\cdot s0[length(\mathrm{s0})-1]\cdot \cdot s0[length(\mathrm{s0})]$, and $snj=snj[1]\cdot snj[2]\cdot snj[3]\cdot\ldots\cdot snj[length(snj)-1]\cdot \cdot snj[length(snj)-1]$, where '$\cdot$' is a concatenation symbol.

The third subtask, which is the set $SSim3$ [4] formation, is solved using a method developed in collaboration with A.Yu. Zavgorodny [5]. The ITS data model and SQL-queries constructed within the method allow to choose adaptively the next task for the student [6].

In the training mode the calculation task process model includes the pedagogical action choice, which is based on the information obtained from the student's model and the current step taken by the student in performing the task j. The various conditions that precede the choice of pedagogical action are:

1. When performing the task j, the student made the wrong step i. In this case, there are two options:

a) showing a tip for the student on the basis of activated DM;

b) transition to a simpler task, which contains needed KSC.

2. The task is performed correctly by the student. Then there are two possible scenarios:

a) transition to the next task of the same class;

b) transition to the task of another class.

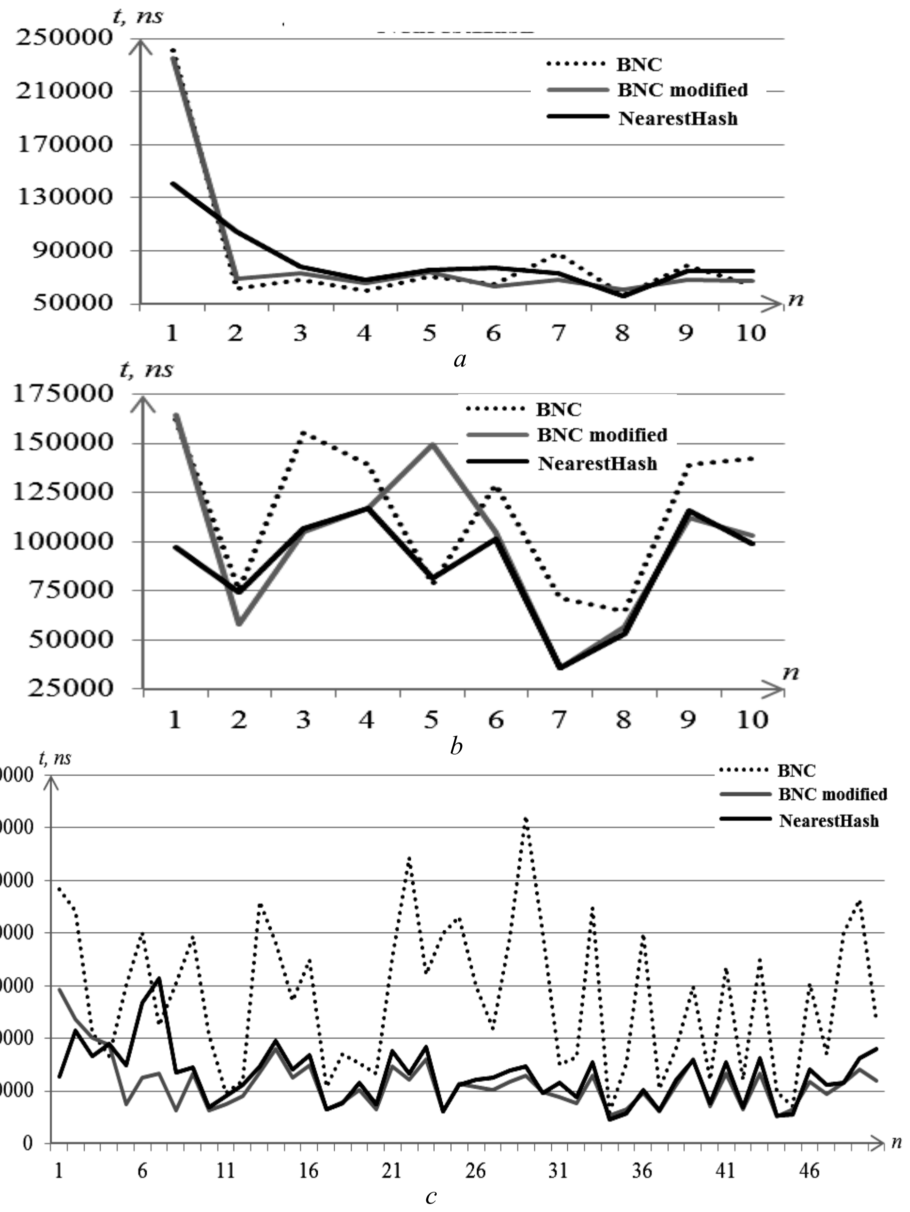


Fig. 1. Comparison of the three methods performance for set, consisting of *a* — 10, *b* — 100, *c* — 1000 randomly generated trees

All these transitions to the next task can be done in two ways:

a) a task is selected by the program;

b) a task is selected by the student.

In the case of student's incorrect step i while performing the task j hint on the basis of activated diagnostic models is shown. Then, in the model of the student after BN for the step i is inserted as many BN temporary layers, as incorrect steps have been made. In this case, if we represent the first layer on the left, such as the graph $G = (E, \Gamma)$, then all duplicate layers will be isomorphic to the original graph, and priori values of the probabilities of mastering KSC will be a posteriori values of probabilities for the previous layer.

An automatic transition from the current task with the several KSCs should be made to a simpler task that contains one of the KSCs deal with the appropriate SQL query. You can also jump to any task that contains fewer KSCs than the current task using a single SQL query. And the transition to tasks that contain lower number of KSC, than in the current task, can be done as a result of another SQL-query.

If a student performs the task correctly transition to the next task of the same class can be done using the method of generation and calculation of objects. As working with a class of tasks KSC are the same, then priori values of the mastering them probabilities will be a posteriori probabilities for the last completed task of this class. Thus, the values of the probabilities of mastering the relevant KSC will increase and reach a certain threshold of "mastery".

If a student solves the task correctly the new task of another class should be chosen from two variants:

a) a task with at least one KSC, which is new for the student;

b) a task with KSCs, which already occupied by the student, but is combined in another way.

The above transitions could be performed using SQL queries.

An alternative learning scenario is a scenario using task clustering. Tasks are grouped into clusters so that each cluster has similar tasks, and clusters are sorted by complexity – by the average number of KSCs in cluster tasks or by the average complexity of tasks, where the complexity of the task is defined as the total complexity of the KSC. In this case, the student begins training with the easiest cluster and does not leave it until all tasks in this cluster are completed. A tuple $(x_1, x_2, \ldots, x_n)$ is used as a representation of the task for clustering, where n is the total number of KSCs in the problem domain, $x_i \in \{0,1\}$ — a component of the tuple that indicates the presence or absence of the KSC with number i in the task. Hemming's metric are usually used as a metric for comparing different tuples. One of the known methods, for example *k*-means, is used for clustering of tasks. In addition, each task can be matched with a hierarchy of KSC, and each hierarchy – with its string representation. Then NearestHash is used to cluster tasks with a special data structure – a system of disjoint sets and metrics K. Tai, K. Zhang, D. Shasha – the minimum editing distance between trees [7; 8].

Since solution of the considered ATs requires to perform calculations using formulas and algorithms that cover not one specific task, but a whole class of tasks, it is necessary to separately organize training in the knowledge of these formulas and algorithms. In such way the student ability to perform any task from a given class could be improved. Let's draw the following analogy. Currently, neural networks are considered to be one of the most effective models of artificial intelligence. An artificial neural network is learned from examples, but there is no guarantee that it will work properly with any source data outside the training sample. Similarly, when teaching a person only by examples, there is no guarantee that he has mastered a general rule, algorithm or formula that covers all possible cases.

The program compiled by the student might be exactly the same as the reference program, for example in the case of small tasks. Then, if the texts of these two programs match, then correctness of the solution could be stated. If the texts do not match, then program testing is required. If the student's program does not match any of the reference programs, then after its testing there are two possible cases: a) the program is incorrect, b) the program is similar to the correct one. If the student's program has passed all the tests and the same or exceeds the existing reference programs in productivity, the ITS can make this program as a new optimal. As criteria of optimality the memory capacity and conciseness should be used as well. Thus, the automatic self-learning of ITS has been realized.

If the student's program does not pass at least one test, the next diagnostic task is solved – finding a place of discrepancy at the lexical level. For this purpose it is used modified Levenstein metric — the minimum editing distance between arrays of lexems of two programs — and modified Wagner–Fisher method to determine this distance as well as the trace of the matrix [8], i.e. the minimum conversion sequence from one program to another.

For the experiment computer Intel Core I5-3210, 2.5 GHz, RAM – 4 GB has been used. The comparison program was written in Java and run in the Eclipse SDK, Version: 3.5.1. As a result of comparison of two programs consisting of 28 bytes, it took 386295 ns. For comparison, when calculating unmodified Levenstein distances by the Wagner–Fisher method for files with a size of 28 bytes, the average calculation time is 443109 ns, for files with a size of 2895 bytes — 135536867 ns $\approx$ 0.1 s, for files of size 7251 bytes — 1250641880 ns $\approx$ 1.25 s, and for files of size 10701 bytes there was an error due to memory overflow.

To solve the problem of diagnosing at the structural level, two abstract syntax trees (AST) are obtained as a result of parsing the reference program and the program compiled by the student. The first way to compare AST is to use the width search method. The second way to compare the AST is to calculate the distance between the trees by the method of K. Zhang, D. Shasha and the editing trace, i.e. the minimum conversion sequence at the structural level. Thus nodes of trees are operands and operators.

If the program compiled by the student does not pass at least one test and the ITS stores more than one reference program for this task, the NearestHash method is used to find the reference program closest (at the lexical or syntactic level) to the student's program.

ALGORITHMS AND SOFTWARE IMPLEMENTATION

The method and models for diagnosing algorithms built by the student in a graphical environment was developed. Game models are given by two models: a model like "biathlon" and a model like "sailing regatta".

As for the first model, each student performs the calculated AT for a certain period of time $t_{finish} - t_{start}$. Since the task consists of n steps, in the terms of biathlon a step i could be considered as a shooting range i. In addition, for each wrong step a student should get penalty as additional Δt to the total time. Thus, a tutor is able:

1) to arrange competitions in the classroom with academic group;

2) to monitor students time at each shooting range;

3) to determine winners at the end of each championship;

4) to determine the absolute record holders after championships series.

The following simplifications were chosen for the second model:

1) the yacht moves along a straight line without course change;

2) no currents;

3) the wind speed is stable;

4) the density of environment is constant.

The student's task is to synthesize the transfer function of the corrective device, which provides the best quality indicators when turning the sail and, as a consequence, when moving the yacht. To determine the effectiveness of the obtained student solution, a step-by-step calculation of the values of velocity over time, as well as the distance traveled. The regatta is visualized: students can compete with each other and with computer models.

The generalized scenario of the "internal cycle" of the ITS for the calculated AT is shown in Fig. 2. The scheme of the student's requests analysis method is shown in Fig. 3.

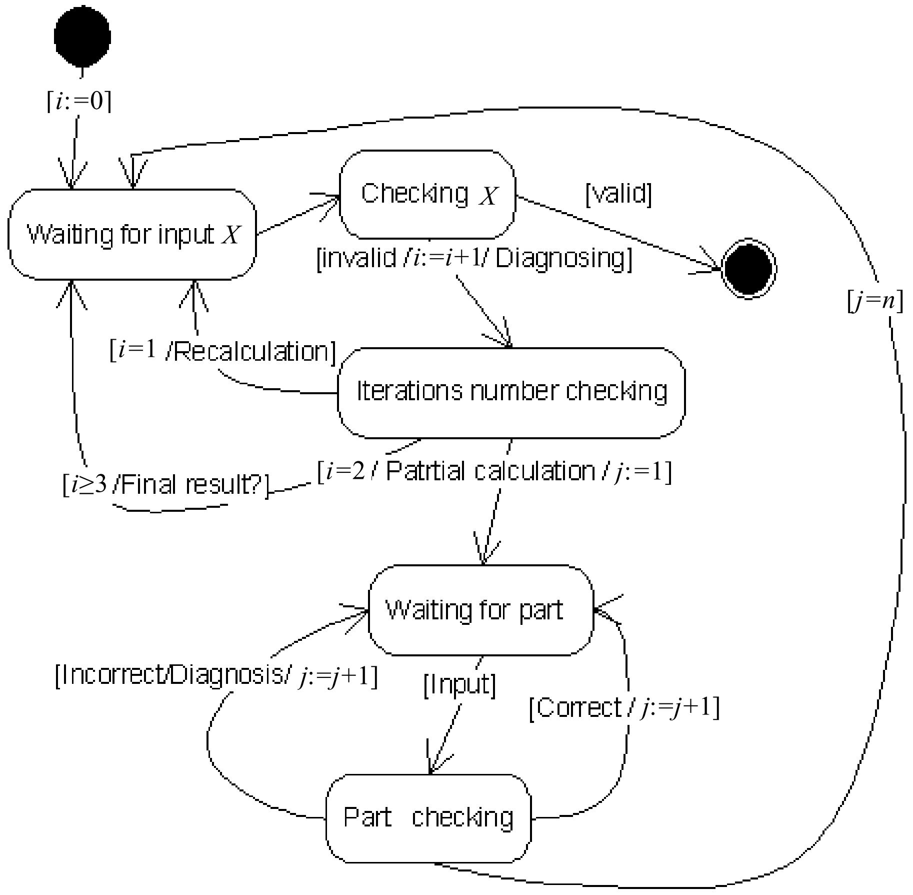


Fig. 2. Generalized diagram of the "internal cycle" of the ITS for the calculated AT

The methods of determining the similarity of SQL-queries are: $q1$ — the classical method of q-grams; $q2$ – the method of q-grams with the previous replacement of keywords in SQL; $q3$ – the method of q-grams derived from syntactic trees. The total similarity of the two SQL queries $qry1$ and $qry2$ is defined as $SIM_{qry1,qry2}^{total} = \sum_{i=1}^{3} SIM_{qry1,qry2}^{qi}$ [9].

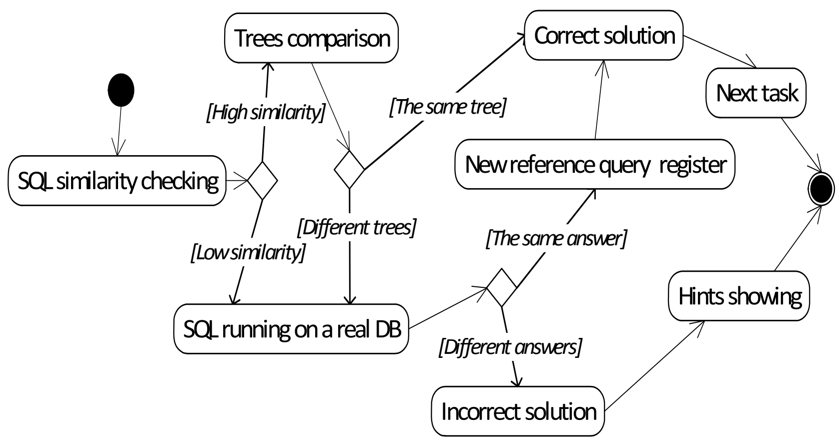


Fig. 3. Diagram of the student's SQL-queries analysis method

Fig. 4 shows the sequence of providing tips for a student. An indication of the error in the query is provided first. If the student is not able to correct the error on his own, then a more detailed indication of the error is provided, namely its location based on a comparison of trees. Next, there is a hint about the KSC that must be used while SQL query construction. Finally, a similar task is demonstrated with explanation of the necessary steps for its implementation.

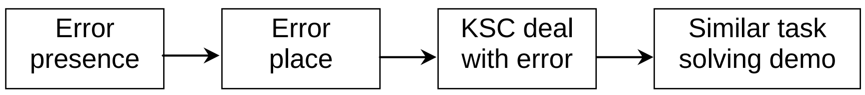
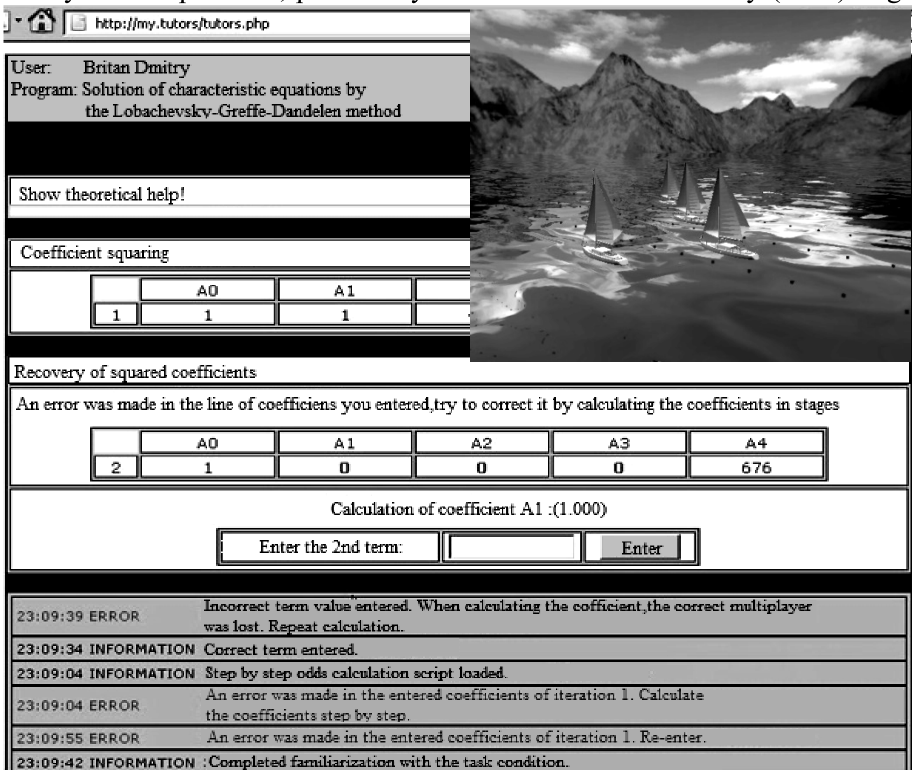


Fig. 4. The sequence of providing tips to the student

The corresponding software was created for the different disciplines of aircraft control systems department, particularly for automatic control theory (ACT). Fig. 5


http://my.tutors/tutors.php

User: Britan Dmitry
Program: Solution of characteristic equations by the Lobachevsky-Greffe-Dandelen method

Show theoretical help!

Coefficient squaring

	A0	A1
1	1	1

Recovery of squared coefficients

An error was made in the line of coefficiens you entered,try to correct it by calculating the coefficients in stages

	A0	A1	A2	A3	A4
2	1	0	0	0	676

Calculation of coefficient A1 :(1.000)

Enter the 2nd term: Enter

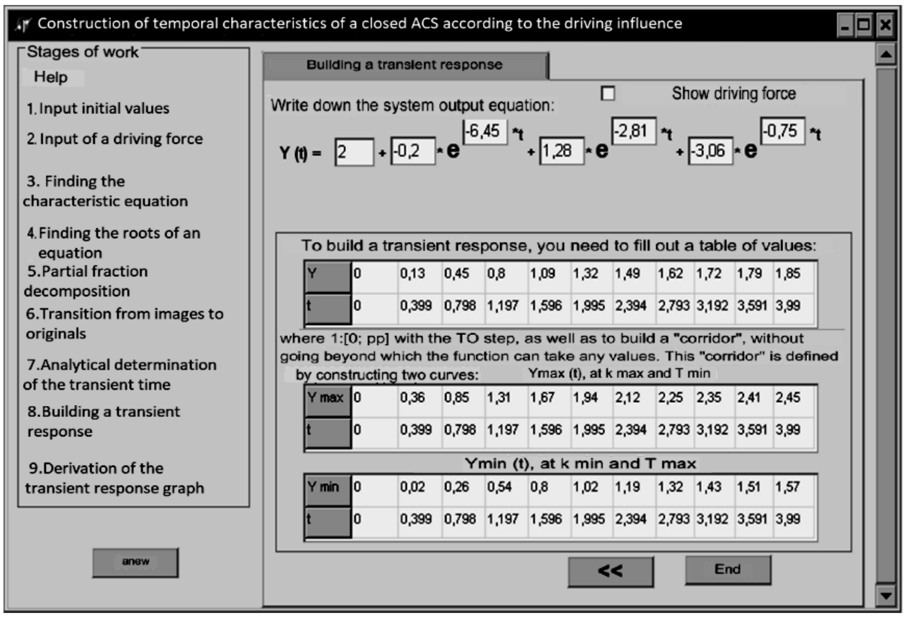
```
23:09:39 ERROR        Incorrect term value entered. When calculating the cofficient,the correct multiplayer
                      was lost. Repeat calculation.
23:09:34 INFORMATION  Correct term entered.
23:09:04 INFORMATION  Step by step odds calculation script loaded.
23:09:04 ERROR        An error was made in the entered coefficients of iteration 1. Calculate
                      the coefficients step by step.
23:09:55 ERROR        An error was made in the entered coefficients of iteration 1. Re-enter.
23:09:42 INFORMATION :Completed familiarization with the task condition.
```

Fig. 5. ITS for the LGD method and ITS "Sailing Regatta"

shows the screen forms of ITS for the method of Lobachevsky-Greffe-Dandelen (LGD) and ITS "Sailing Regatta".

The experiments showed that the minimum reduction in the time of manual completion of tasks by students due to training with ITS support for the LGD method was 23.3%, and the maximum – 50%. In this case, all students obtained proper solutions not only in the learning process with the help of ITS, but also manually during a post-test.

Figs. 6, 7 shows the screen forms of ITS for the construction of frequency characteristics of the of automatic control object and ITS for the construction of transient characteristics of the automatic control system.


Construction of temporal characteristics of a closed ACS according to the driving influence
Stages of work
Help
1. Input initial values
2. Input of a driving force
3. Finding the characteristic equation
4. Finding the roots of an equation
5. Partial fraction decomposition
6. Transition from images to originals
7. Analytical determination of the transient time
8. Building a transient response
9. Derivation of the transient response graph
anew
Building a transient response
Show driving force
<<
End

Write down the system output equation:

$$Y(t) = \boxed{2} + \boxed{-0{,}2} * e^{\boxed{-6{,}45} * t} + \boxed{1{,}28} * e^{\boxed{-2{,}81} * t} + \boxed{-3{,}06} * e^{\boxed{-0{,}75} * t}$$

To build a transient response, you need to fill out a table of values:

Y	0	0,13	0,45	0,8	1,09	1,32	1,49	1,62	1,72	1,79	1,85
t	0	0,399	0,798	1,197	1,596	1,995	2,394	2,793	3,192	3,591	3,99

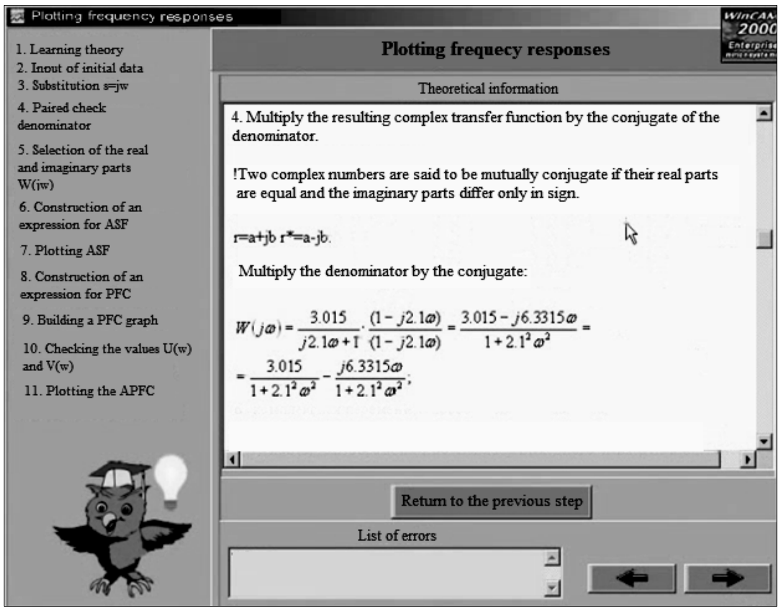
where 1:[0; pp] with the TO step, as well as to build a "corridor", without going beyond which the function can take any values. This "corridor" is defined by constructing two curves: Ymax (t), at k max and T min

Y max	0	0,36	0,85	1,31	1,67	1,94	2,12	2,25	2,35	2,41	2,45
t	0	0,399	0,798	1,197	1,596	1,995	2,394	2,793	3,192	3,591	3,99

Ymin (t), at k min and T max

Y min	0	0,02	0,26	0,54	0,8	1,02	1,19	1,32	1,43	1,51	1,57
t	0	0,399	0,798	1,197	1,596	1,995	2,394	2,793	3,192	3,591	3,99

Fig. 6. ITS for object frequency characteristics construction


Plotting frequency responses
1. Learning theory
2. Inout of initial data
3. Substitution s=jw
4. Paired check denominator
5. Selection of the real and imaginary parts W(jw)
6. Construction of an expression for ASF
7. Plotting ASF
8. Construction of an expression for PFC
9. Building a PFC graph
10. Checking the values U(w) and V(w)
11. Plotting the APFC
WinCAM 2000 Enterprise
Plotting frequecy responses
Theoretical information
Return to the previous step
List of errors

4. Multiply the resulting complex transfer function by the conjugate of the denominator.

!Two complex numbers are said to be mutually conjugate if their real parts are equal and the imaginary parts differ only in sign.

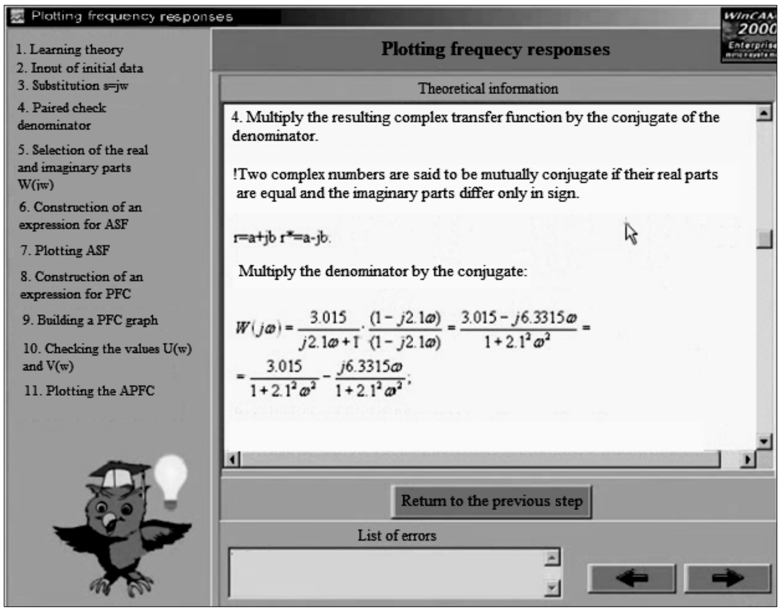
r=a+jb r*=a-jb.

Multiply the denominator by the conjugate:

$$W(j\omega) = \frac{3.015}{j2.1\omega + 1} \cdot \frac{(1 - j2.1\omega)}{(1 - j2.1\omega)} = \frac{3.015 - j6.3315\omega}{1 + 2.1^2\omega^2} =$$
$$= \frac{3.015}{1 + 2.1^2\omega^2} - \frac{j6.3315\omega}{1 + 2.1^2\omega^2};$$

Fig. 7. ITS for the transient characteristics construction

As a result of experimental researches it is established that due to ITS for construction of frequency characteristics of OAU reduction of time of performance of a task by students made from 25 to 95.4%. As the conducted experiments showed, thanks to ITS for construction of transitional characteristics of ACS the minimum reduction of time at manual performance of a task by students made 15%, and the maximum – more than 36.6%.


Plotting frequency responses
WinCAM 2000 Enterprise
1. Learning theory
2. Inout of initial data
3. Substitution s=jw
4. Paired check denominator
5. Selection of the real and imaginary parts W(iw)
6. Construction of an expression for ASF
7. Plotting ASF
8. Construction of an expression for PFC
9. Building a PFC graph
10. Checking the values U(w) and V(w)
11. Plotting the APFC
Plotting frequecy responses
Theoretical information
Return to the previous step
List of errors

4. Multiply the resulting complex transfer function by the conjugate of the denominator.

!Two complex numbers are said to be mutually conjugate if their real parts are equal and the imaginary parts differ only in sign.

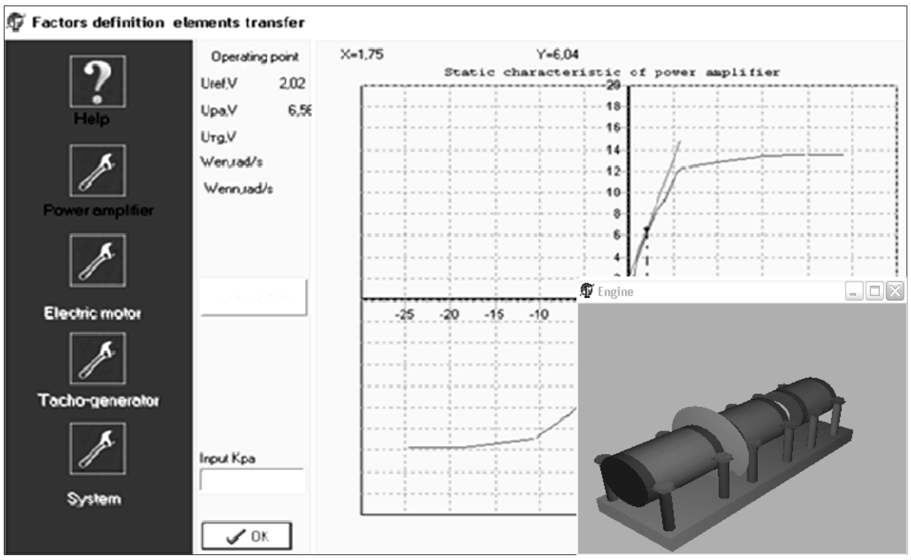
r=a+jb r*=a-jb.

Multiply the denominator by the conjugate:

$$W(j\omega)=\frac{3.015}{j2.1\omega+1}\cdot\frac{(1-j2.1\omega)}{(1-j2.1\omega)}=\frac{3.015-j6.3315\omega}{1+2.1^2\omega^2}=$$
$$=\frac{3.015}{1+2.1^2\omega^2}-\frac{j6.3315\omega}{1+2.1^2\omega^2};$$

Fig. 7. ITS for the transient characteristics construction

Fig. 8 shows the screen forms of virtual laboratory work (VLW) “Experimental determination of the parameters of the transfer functions of the objects of automatic stabilization”.

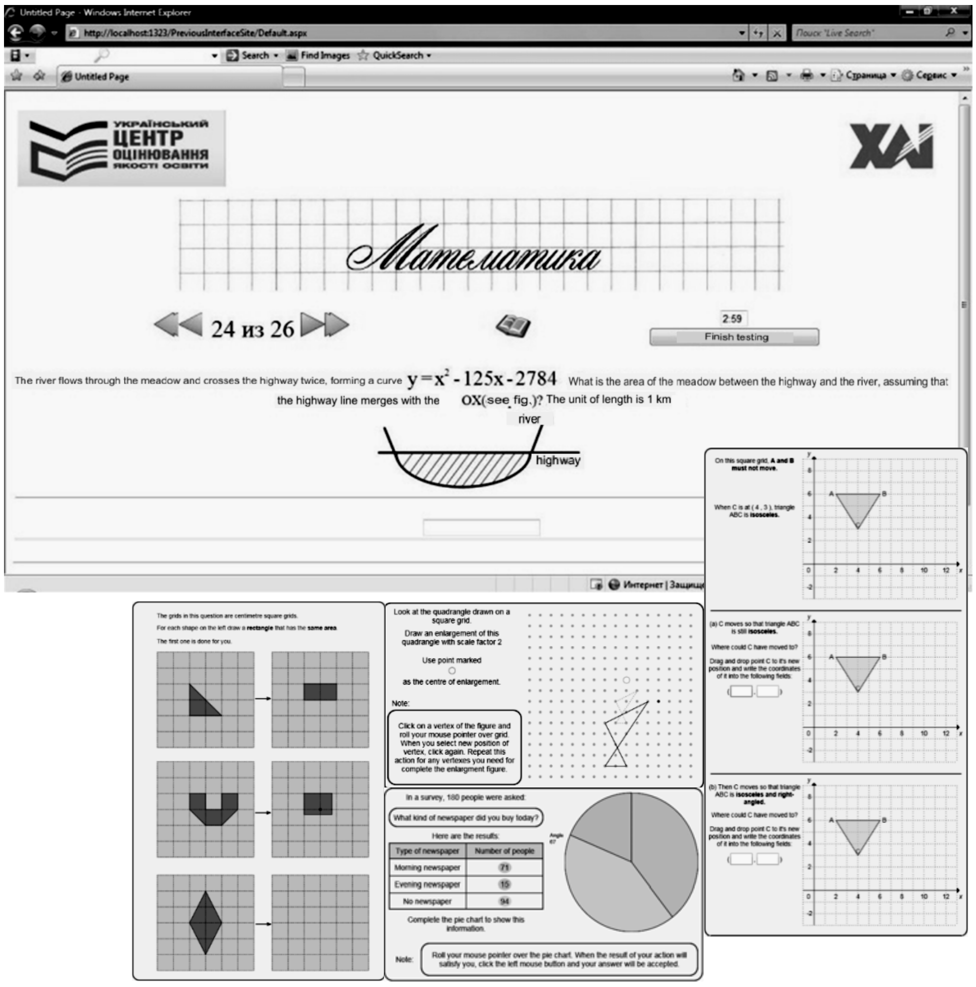

Factors definition elements transfer
Help
Power amplifier
Electric motor
Tacho-generator
System
Static characteristic of power amplifier
Engine
OK

Operating point
Uref,V 2,02
Upa,V 6,5€
Urg,V
Wen,rad/s
Wenn,rad/s
Input Kpa
X=1,75 Y=6,04

Fig. 8. Screen forms examples of VLW from automatic control theory

The minimum reduction in the time students performed this laboratory work due to ITS was 33.3%, and the maximum was 43.3%.

Fig. 9 shows screen forms of software for testing knowledge and skills in mathematics, created by order of Kharkiv Regional Centre of Education Quality Estimation, as well as Plymouth Inoovation Centre in Mathematics Teaching (Great Britain).


Untitled Page - Windows Internet Explorer
http://localhost:1323/PreviousInterfaceSite/Default.aspx
Search
Find Images
QuickSearch
Untitled Page
Страница
Сервис
УКРАЇНСЬКИЙ ЦЕНТР ОЦІНЮВАННЯ ЯКОСТІ ОСВІТИ
ХАІ
Математика
24 из 26
2:59
Finish testing
river
highway
Интернет | Защищ

The river flows through the meadow and crosses the highway twice, forming a curve $y = x^2 - 125x - 2784$ What is the area of the meadow between the highway and the river, assuming that the highway line merges with the OX(see fig.)? The unit of length is 1 km

The grids in this question are centimetre square grids.
For each shape on the left draw a **rectangle** that has the **same area**.
The first one is done for you.

Look at the quadrangle drawn on a square grid.
Draw an enlargement of this quadrangle with scale factor 2
Use point marked ○ as the centre of enlargement.
Note:
Click on a vertex of the figure and roll your mouse pointer over grid. When you select new position of vertex, click again. Repeat this action for any vertexes you need for complete the enlargment figure.

In a survey, 180 people were asked:
What kind of newspaper did you buy today?
Here are the results:

Type of newspaper	Number of people
Morning newspaper	71
Evening newspaper	15
No newspaper	94

Complete the pie chart to show this information.
Note: Roll your mouse pointer over the pie chart. When the result of your action will satisfy you, click the left mouse button and your answer will be accepted.

On this square grid, **A and B must not move.**
When C is at (4, 3), triangle ABC is **isosceles.**

(a) C moves so that triangle ABC is still **isosceles.**
Where could C have moved to?
Drag and drop point C to it's new position and write the coordinates of it into the following fields:

(b) Then C moves so that triangle ABC is **isosceles and right-angled.**
Where could C have moved to?
Drag and drop point C to it's new position and write the coordinates of it into the following fields:

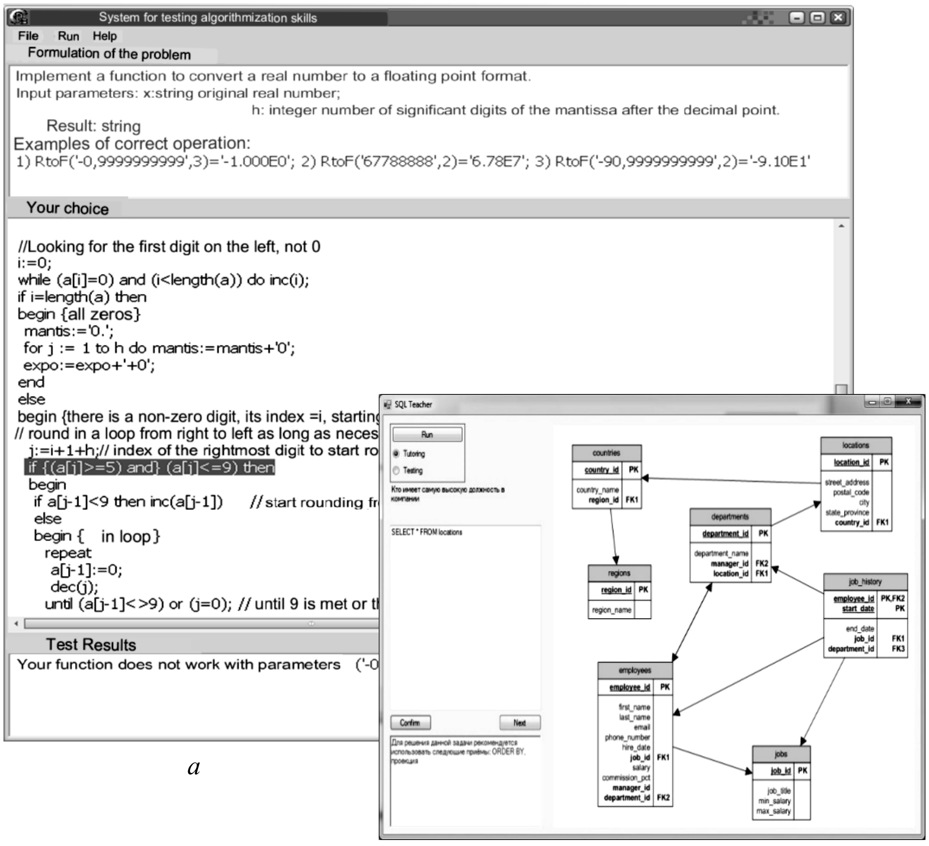
Fig. 9. Software for testing knowledge and skills in mathematics

The minimum time reduction for the development of the site for Centre of Education Quality Estimation was 553 hours, and the minimum savings – 8188 pounds, which is confirmed by the relevant act.

Fig. 10 shows the screen forms of ITS for algorithms [10] and ITS for SQL [11]. In the case of ITS for algorithms, each student out of twenty in the academic group should perform two tasks during the experiments. As a result of the experiment:

– in the first task: 10 from 20 persons obtained at once a conditional solution and 10 obtained at once a full solution. From those who obtained a conditional solution, 8 persons managed to obtain a complete solution due to ITS support later;

– in the second task: 7 from 18 persons obtained at once a conditional solution and 6 – a full solution. From those who obtained a conditional solution, 3 persons obtained a full solution with ITS support later.



Implement a function to convert a real number to a floating point format.
Input parameters: x:string original real number;
h: integer number of significant digits of the mantissa after the decimal point.
Result: string
Examples of correct operation:
1) RtoF('-0,9999999999',3)='-1.000E0'; 2) RtoF('67788888',2)='6.78E7'; 3) RtoF('-90,9999999999',2)='-9.10E1'

```
//Looking for the first digit on the left, not 0
i:=0;
while (a[i]=0) and (i<length(a)) do inc(i);
if i=length(a) then
begin {all zeros}
 mantis:='0.';
 for j := 1 to h do mantis:=mantis+'0';
 expo:=expo+'+0';
end
else
begin {there is a non-zero digit, its index =i, startin
// round in a loop from right to left as long as neces
  j:=i+1+h;// index of the rightmost digit to start ro
 if {(a[j]>=5) and} (a[j]<=9) then
  begin
   if a[j-1]<9 then inc(a[j-1])     //start rounding fr
   else
   begin {   in loop}
     repeat
      a[j-1]:=0;
      dec(j);
     until (a[j-1]<>9) or (j=0); // until 9 is met or t
```

Your function does not work with parameters ('-0

Кто имеет самую высокую должность в компании

```
SELECT * FROM locations
```

Для решения данной задачи рекомендуется использовать следующие приёмы: ORDER BY, проекция

b

Fig. 10. ITS screen forms: *a* — for algorithms; *b* — for SQL

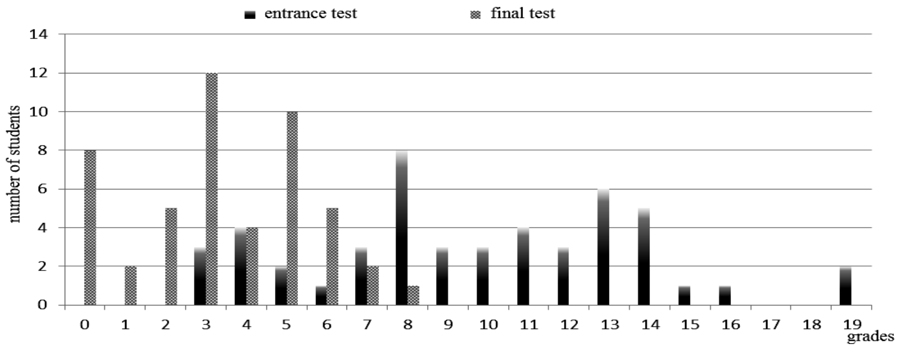
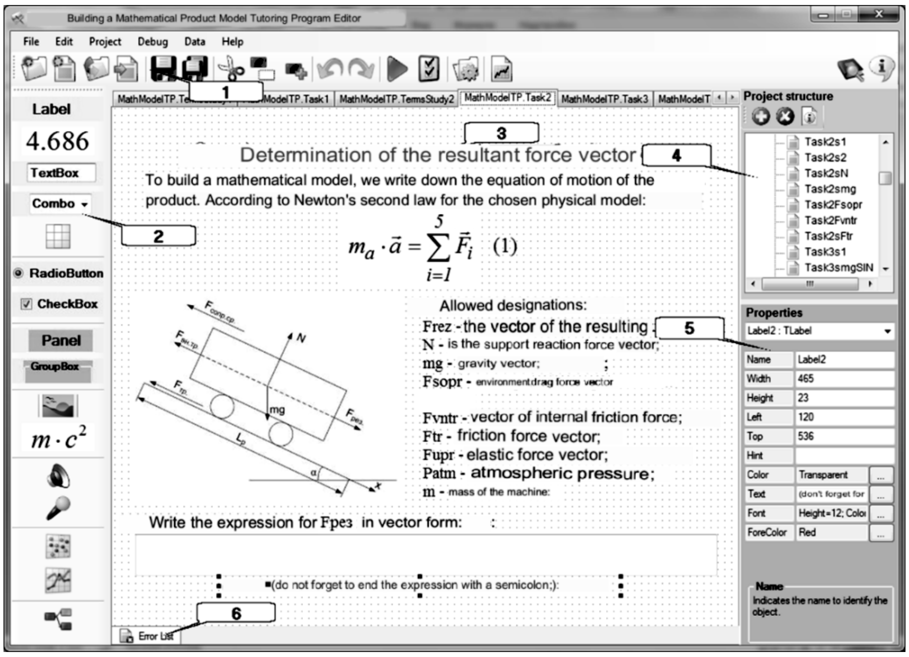


Fig. 11. Histogram for entrance and final students testing in ITS for SQL

Regarding the ITS for the SQL language, the experimental study of this ITS was conducted on 49 KhAI students. At the first stage, students performed 5 tasks of entrance testing, for which they were given a grades. Then students were solving 10 tasks in the studying mode and after that they passed a final test, which consists of the same tasks as the entrance one. As a result, for each student the score for the final test exceeds the score for the entrance test. The combined histogram for entrance and final assessments is shown in Fig. 11.

Experimental studies of the software for the ITS automated creation and run universal environment were also carried out [12]. The screen form of the environment and the histogram with results of students training by means of the created ITS for the discipline "Basics of systems modeling" are shown in Figs. 12, 13.



Determination of the resultant force vector

To build a mathematical model, we write down the equation of motion of the product. According to Newton's second law for the chosen physical model:

$$m_a \cdot \vec{a} = \sum_{i=1}^{5} \vec{F}_i \quad (1)$$

Allowed designations:
Frez - the vector of the resulting
N - is the support reaction force vector;
mg - gravity vector; ;
Fsopr - environment drag force vector

Fvntr - vector of internal friction force;
Ftr - friction force vector;
Fupr - elastic force vector;
Patm - atmospheric pressure;
m - mass of the machine:

Write the expression for Fpe3 in vector form: :

(do not forget to end the expression with a semicolon;):

Name	Label2
Width	465
Height	23
Left	120
Top	536
Hint	
Color	Transparent
Text	(don't forget for
Font	Height=12; Colo
ForeColor	Red

Name
Indicates the name to identify the object.

Fig. 12. Universal environment for the ITSs automated creation and translation

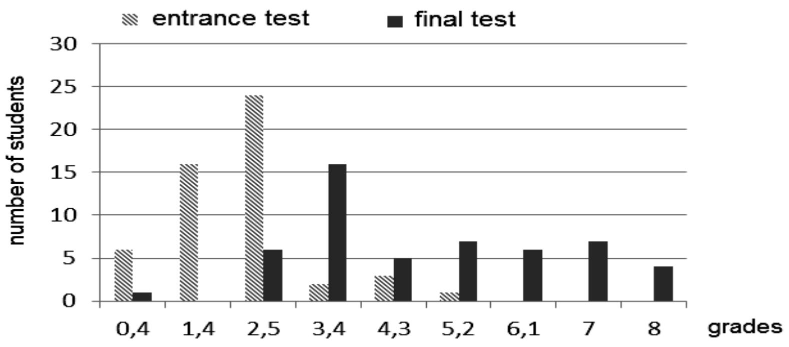


Fig. 13. Histogram of the students' learning results with ITS

The experiment participants were 52 students, who were asked to take an entrance test, training and final test from the discipline in the ITS environment. As a result, each student had increased his grades.

CONCLUSIONS

The second part of the paper presents practical results for the development and implementation of specific ITSs. There were developed different ITS for specific domains. The experimental studies allow stating following:

1. Every student who studies by means of ITS has occupied necessary knowledge, skills and competencies while performing tasks, despite possible mistakes made during solving, ignorance or incompetence of a student.

2. Task solving time has been reduced at least twice on average.

3. Students got skills not only in certain tasks solving, but also in operating with whole task classes.

4. The tutor routine load has been significantly reduced.

5. Motivation of students to master new competencies has been increased.

The scientific and practical results of the work were shown in one doctoral and three candidate dissertations. There are over 100 scientific publications, indexed in the Scopus, reported and approved at scientific conferences in Ukraine, USA, Germany, Austria, Spain, Poland.

The developed ITSs have been implemented in a number of organizations, educational institutions and enterprises in Ukraine, Austria, Great Britain and China.

REFERENCES

1. A. Chukhray, *Methodology for learning algorithms: monograph*. National Aerospace University "KhAI", 2017, 336 p.
2. M.Z. Zgurovsky and N.D. Pankratova, *System analysis: problems, methodology, applications*. Kyiv: Nauk. opinion, 2011.
3. N.D. Pankratova, "System analysis in the dynamics of diagnosing complex technical systems," *System Research & Information Technology*, no. 4, pp. 33–49, 2008.
4. A. Kulik, A. Chukhray, and O. Havrylenko, "Information technology for creating intelligent computer programs for training in algorithmic tasks. Part 1: Mathematical foundations," *System Research & Information Technologies*, no. 4, pp. 27–41, 2021.
5. A. Kulik, A. Chukhray, and A. Zavgorodniy, "Similar strings detecting methods," in *Proceedings of the East-West Fuzzy Colloquium*, *Zittau, Germany, IPM, 2005*, pp. 183−191.
6. J.P. Martínez Bastida, O. Havrylenko, and A. Chukhray, "Developing a self-regulation environment in an open learning model with higher fidelity assessment," *Communications in Computer and Information Science*, vol. 826, pp. 112–131, 2018.
7. K. Zhang and D. Shasha, "Simple fast algorithms for the editing distance between trees and related problems," S*ociety for Industrial and Applied Mathematics. Journal on Computing*, no. 18 (6), pp. 1245–1262, 1989.
8. A. Chukhray and O.Havrylenko, "Proximate Objects Probabilistic Searching Method," *Advances in Intelligent Systems and Computing, 1113 AISC, 2020*, pp. 219–227.
9. A. Chukhray and O. Havrylenko, "The method of student's query analysis while intelligent computer tutoring in SQL," *Radioelectronic and Computer Systems*, no. 2 (98), pp. 78–86, 2021. doi: 10.32620/re`ks.2021.2.07.
10. Ie. Vagin, O. Havrylenko, J.P. Martínez Bastida, and A. Chukhray, "Computer Intelligent Tutoring System "SQLTOR,"" ICTERI2019 *ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer*, pp. 541–546. Available: http://ceur-ws.org/Vol-2387/20190525.pdf
11. D. Gaydachuk, O. Havrylenko, J.P.Martínez Bastida, and A. Chukhray, "Structural diagnosis method for computer programs developed by trainees," *Proceedings of 15th International Conference on ICT in Education, Research, and Industrial Applications*, *2019, Kherson, Ukraine*, pp. 485–490.
12. A. Chukhray and O. Havrylenko, "The engineering skills training process modeling using dynamic Bayesian nets," *Radioelectronic and Computer Systems*, no 2 (98), pp. 87–96, 2021. doi: 10.32620/reks.2021.2.08.

Recieved 27.12.2022

INFORMATION ON THE ARTICLE

Anatoliy S. Kulik, ORCID: 0000-0001-8253-8784, National Aerospace University "Kharkiv Aviation Institute", Ukraine, e-mail: anatolykulik@gmail.com

Andrey G. Chukhray, ORCID: 0000-0002-8075-3664, National Aerospace University "Kharkiv Aviation Institute", Ukraine, e-mail: achukhray@gmail.com

Olena V. Havrylenko, ORCID: 0000-0001-5227-9742, National Aerospace University "Kharkiv Aviation Institute", Ukraine, e-mail: lm77191220@gmail.com

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ СТВОРЕННЯ ІНТЕЛЕКТУАЛЬНИХ КОМП'ЮТЕРНИХ ПРОГРАМ ДЛЯ НАВЧАННЯ АЛГОРИТМІЧНИМ ЗАВДАННЯМ. Частина 2: Дослідження та реалізація / А.С. Кулік, А.Г. Чухрай, О.В. Гавриленко

Анотація. Інформаційні технології і, зокрема, методи штучного інтелекту дедалі глибше проникають у всі сфери людської діяльності: науку, техніку, мистецтво та освіту. Україна також має достатній потенціал і надзвичайно потребує засобів підтримки навчання, розробці яких і присвячено цю статтю. Метою роботи є демонстрація компонентів інформаційної технології створення інтелектуальних системи навчання, які залучено до вивчення різних дисциплін технічного профілю. У роботі використано методи системного аналізу, математичного та імітаційного моделювання, технічного діагностування, штучного інтелекту. Запропоновані моделі та методи реалізовано в прототипах ITS для навчання математиці, програмуванню, теорії автоматичного управління. ITS було впроваджено в навчальному процесі університету ХАІ, інших установах України, Великобританії, Австрії, Китаю. Експериментальні дослідження показали підвищення показників успішності навчання студентів за допомогою ITS у порівнянні з традиційними методами. З наукової точки зору мають цінність покращені і адаптовані для комп'ютерного навчання методи технічної діагностики, Байєсових мереж та розроблені моделі алгоритмічних задач, процесу навчання та особи, що навчається. У практичному сенсі отримані результати можна використовувати для створення нових спеціалізованих ITS, а також для побудови єдиної розширюваної навчаючої платформи, яка об'єднує базові дисципліни певної галузі.

Ключові слова: інтелектуальні системи навчання, експериментальні дослідження, результати впровадження.