Domain specific language approach to technology-enhanced learning.

Pozdniakov Sergei
e-mail: pozdnkov@gmail.com
Saint Petersburg State Electrotechnical University (ETU)
Saint Petersburg, Russia

Ilya Posov
e-mail: iposov@gmail.com
Saint Petersburg State University
Saint Petersburg, Russia

Abstract

Which language should a student use to interact with computer programs? The present work answers this question and explains the reasons for the answer. A student should interact with a computer using the language of the subject domain. The domain-specific language has a fixed context, that does not depend on a student, as well as a set of terms and operations, which can be formalized. That is why this language is both convenient for the user and good for computer implementation. The domain-specific language is also good as it does not require a special training for the teachers, because it is defined by the contents and tools of the subject field. We consider two types of domain-specific languages: the first type is needed for precise representation of problems and the second type serves for precise solution representation. "Two domain-specific languages" approach is implemented by us in the WiseTasks system and is presented in the report.

1. Introduction

The basic way to estimate the advantage or disadvantage of the computer software in teaching of mathematics is to understand its psychological role in intellectual progress of a student [7]. If we consider a computer as a tool [1] then the role of a computer tool can be estimated as:

- computer software is useful for teaching of mathematics if it fulfils tool functions, and a student forms her own internal intellectual mechanisms by means of it [2]
- computer software is harmful if it replaces intellectual operations and a student uses it instead of his or her own internal intellectual mechanisms.

A student masters his or her own intellectual mechanisms through the problems solving. In mathematical teaching the concept of a problem has both psychological and didactic meaning.

Here we will discuss possibilities to support students work with mathematical problems. The main goal of the article is to discuss some didactic ways to represent problems in systems that provide technology enhanced learning in mathematics. These ways of representations aim to improve the systems according to their main psychological goal that is the development of intellectual skills of the trainee.

Discussion of problems role in the process of learning mathematics is usually reduced to the discussion of solution features of various tasks. At the same time (Polya) [11] notes that when a student meets an unknown problem she should be able to formalize it herself, and while she is solving a hard problem, to break it up in a number of smaller or auxiliary problems and to simplify the initial problem statement. Thus, there is an important question of how to support a creation of well formalized mathematical problems statements by students.

Note that the most elaborate feature of the tools that support students in the work with problems is the ability to formally describe the process of solving a problem.

For example, dynamic geometry allows us to represent solutions of the geometry problems using the language of linear geometrical algorithms [9].

At the same time there is a lack of systems that allow formally describing the problems by themselves. To some extent this issue is addressed in the computer simulation systems that make user to describe a problem by means of a number of differential equations or some other complex objects. But such systems are not supposed to be used to describe problems that are usually stated for a student during her education. For example, if we consider combinatorial or geometrical problems, then we will face the fact that there are no existing systems that allow to describe a problem statement in such a way that it will be possible to automatically test a solution provided by a student. The absence of the systems with such properties leads us to the need of special knowledge assessment means such as the multiple-choice tests.

In this paper we present a way to extend capabilities of the existing systems such as the dynamic geometry systems so that they will provide students with formal problems statements that will be used to test formal solutions provided by students automatically.

Such approach is implemented by us in the system WiseTasksGeometry presented in the report.

We will call such an approach to creation of means that support problems solutions bilingual. This means that one should develop not only means to describe the problems solutions, but also means to describe problems statements. Both will use an object-oriented approach, i.e. special languages for description of these important aspects of problems. Thanks to the formalization of these components, the ability to test problems solutions over their statements arises.

The usage of two languages implements a basic principle: the student cannot use the tool to avoid problem solving because both languages (the language of problem description and the language of solution description) do not have any means to solve a problem and therefore all the tools can be made available to the student.

On the other hand, the fact that a student has a tool for statement description makes it possible for him or her to state their own research problems. Thus, the system gives new didactic possibilities in teaching of mathematics: the system supports students not only to solve problems but also to state them.

The presented approach is also applied in the other system presented in the report: WiseTasksCombinatorics [3] uses the language of the set theory to state problems and the language of mathematical formulas to solve stated problems.

The analysis presented in the report demonstrates the importance of domain specific languages for problems statements and solutions descriptions, and the importance of the usage of different languages for this purpose.

2. Languages for technology-enhanced learning

Let us consider the language aspects of working with problems. A teacher does not usually think whether the statement of the problem in the natural language that she offers to a student is completely formalized. At the same time all the attempts to formalize the presentation of mathematical ideas demonstrate that it is really hard to formalize all the aspects of the mathematical language. Generally a statement of a problem presupposes some conventions and agreements, or, in other words, the language context. Outside of this context a problem may even change its content. This reasoning demonstrates that it is almost impossible to use a natural language for formal description of problems that is aimed to be used in a computer system. The opposite way is the usage of some algorithmic language to represent statements. As Seymour Papert [2] demonstrated in his works, this approach is possible, but it is applicable mostly to the description of algorithms of problems solutions, and not the description of problems themselves.

The present work gives a reply to the raised question and explains the reasons for it. The student should interact with the computer using the language of the subject field. The subject field language has a fixed context, that does not depend on the student, as well as a set of terms and operations, which can be formalized. That is why this language is both convenient for the user and good for computer realization.
The analysis of the program tools supporting the teaching mathematics shows, that the environments based on the domain-specific languages are the most widespread. For example, in dynamic geometry systems, such language is the language of the geometric constructions, or, in the computer algebra systems, e.g. LeActiveMath [8], the language of the algebraic transformations is used.

The domain-specific language is also good as does not require a special training for the teachers, because it is defined by the contents and tools of the subject field.

3. Language tools to describe the mathematical problems statement and solutions

In the previous analysis we highlighted the two aspects of the representation of problems in the systems that support technology enhanced learning in mathematics. They are the representation of the problems statements and the representation of their solutions. While the latter can be considered as developed fairly well, the former needs a detalization.
The existing descriptive (language) means for the representation of problems statements are as follows:
The most spread languages for describing the problem statement are:

1. Verbal description of the problem. It is usually informal and cannot be used for the exact definition of the problem as it contains the context based on the implicit agreements between the teacher and the student, which are informal and can differ from one teacher to another.
2. The model when the professional modelling systems are used for the problem definition. This method is based on the phenomena description in the terms of different types of the equations systems (including differential ones) and it can also be represented by a graphic description of the relations between the problem objects.
3. Problem statement drawing, representing the problem data, but not the target to achieve when solving this problem. That is why the drawing is used together with the verbal description of the problem and makes it more exact.

The first and the last variants are difficult to be formalized. At the same time if we assume that it is possible to present a formal description of a problem, then it may be used to automatically generate its verbal and graphical representation.

Let us take a look on some software that allows describing mathematical objects.
«The Geometry Expressions» system is an interesting example of how the modelling is used to describe the geometrical problems through the relations between the geometric objects. This system uses a language describing the relations between the objects (for example, when building a perpendicular from the point O to the straight line l, a random point M on the line and the condition «OM is perpendicular to l» are used). In this system, every object of the drawing is described by its relations to the other objects, and the system solves the equations for building the defined object. This approach is totally different from the one of the dynamic geometry, where the object is built directly and is defined by a sequence of the elementary operations. By the way, «The Geometry Expressions» system is used as a dynamic geometry system - for describing the problems solutions, but in our opinion, this language (that is the language of the relations between objects) is better for describing the problem statement by the teacher, and the dynamic geometry language is better for solution description by the student. Such approach is realized in the WiseTasks_Geometry system.

The problems solutions description languages are more popular in developing computer tools for teaching mathematics:

1. The most popular method (which cannot be automatically analysed by now) is a combination of the verbal description of the solution along with building sketches, writing formulae and transformations sequences, logic conclusions.
2. The solution representation by filling in the templates, that suppose one or several standard solutions for the problem and "lead" the student to one of the solutions; this method has significant disadvantages, because limits the student's cognition [4] and does not contribute to productive thinking forming [5].
3. Solution representation as a sequence of actions (linear algorithm) in the subject field, not limiting the student in choosing the solution method. This approach is used in the dynamic geometry programs, and the popularity of using it is an evidence of its efficiency.

It is important to discuss the representation method for the geometrical proof problems used in the little-known educational software «Planimetry 7-9» [6]. This program proposes the building relations between geometric objects both for problems and solutions descriptions. The relations are searched by the student and are checked by the tools that are usually used in dynamic geometry. The problem statement is represented by an unordered set of relations, and the solution - proof - by an ordered sequence of relations between the existing and constructed during the solution geometric objects (additional constructions). The proof in this case cannot be checked automatically, because the logical links in it are not fixed as well the theorems that are the base for making conclusions. Given the analysis represented in the article by the key works, we came to the following conclusions:

1. To make an adequate representation of the mathematical problem statement and to describe its solution in the way allowing the automatic check, the problem statement and solutions description languages are needed.
2. The language for problem description and the language for the solution description are different. The first one is used to avoid uncertainty in the statement formulation and contains the description of the properties for the solution to comply with and the set of operations used to build the solution.
3. The solutions description language should provide the representation of any solution of the problem from the given set using the subject tools. It also should allow the verification of the solution created using this language based on the problem description.

Let's examine the WiseTasks system created within the present work and including the problem description language, the solutions description language and also the solutions verification tools. The WiseTasks system allows automatic verification of the solution created using the solutions description language, on the set of predicates of the problem statement written using the problem description language.

4. WiseTasks system

The present analysis gave two directions to develop the domain-specific languages to support teaching mathematics:

1) The development of the languages to describe the problems in different subject fields
2) The development of the languages to describe the problem solutions within the environments modelling the corresponding subject fields

These languages are, on the one hand, independent, on the other hand - the y should work with the same data representations (in the teacher's area the problem is created, in the student's area it is solved, then the solution is checked again in the first area to comply with the problem statement described formally).

Let's examine a realization of the system for teaching geometry. For it, a special language to describe the statement of the problems that are usually solved by means of the dynamic geometry has been created. The GeoGebra system was chosen as the realization of the solutions description language. This system is open for integration with other applications.

5. Domain-specific languages for geometry problems statement and solutions editors

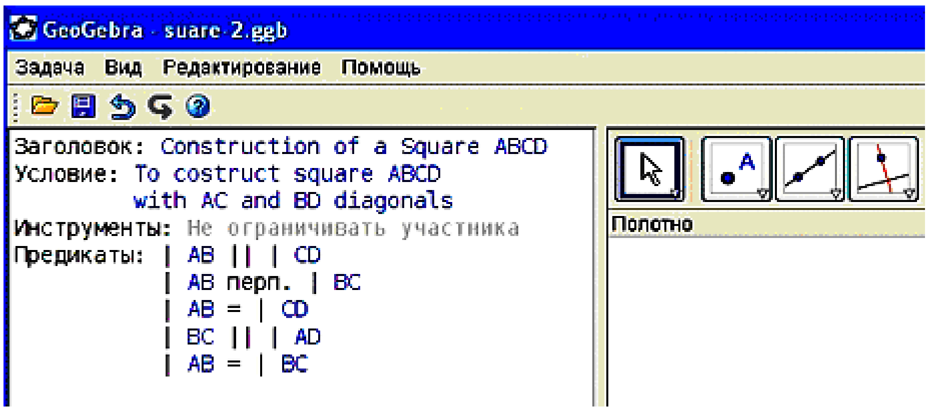
To develop the WiseTasks domain-specific language editor, the MPS (Meta Programming System) from JetBrains [10] was taken as a base. It allows creating domain-specific languages and editors for them. This system is complex enough and does not have an export feature for the created editor to other products, so we use an MPS-like editor in our system. This editor looks like the text one, but uses the grammar constructions context and limits the user in language sentences forming; therefore, there is no possibility to enter the text with incorrect syntax.

The problem in a structured editor consists of cells. There are cells without editing (hints for the users) and cells with editing (content part of the problem).

There are several types of the cells with editing.

1. Simple cells in which a number, a string, a boolean value (yes/no) is entered or a value is selected from the list. Example: the task heading field, the tool selection field (an item from the list of the tools). By default, such cells are empty. Available operations: empty the contents, select from the list (in case it exists).
2. Cells-structures, which consist of the sequences of cells. By default, these cells are empty (for example, an element of the predicates list). The expanding list (like the simple cell with value selection) allows to select the structure type. Available operations: empty the contents, select the contents type.
3. Cells-lists, which consist of the sets of cells. By default, such set can be empty (for example, the "do not limit the participant" in the tools list) or contain one empty element (for example, the list of predicates).

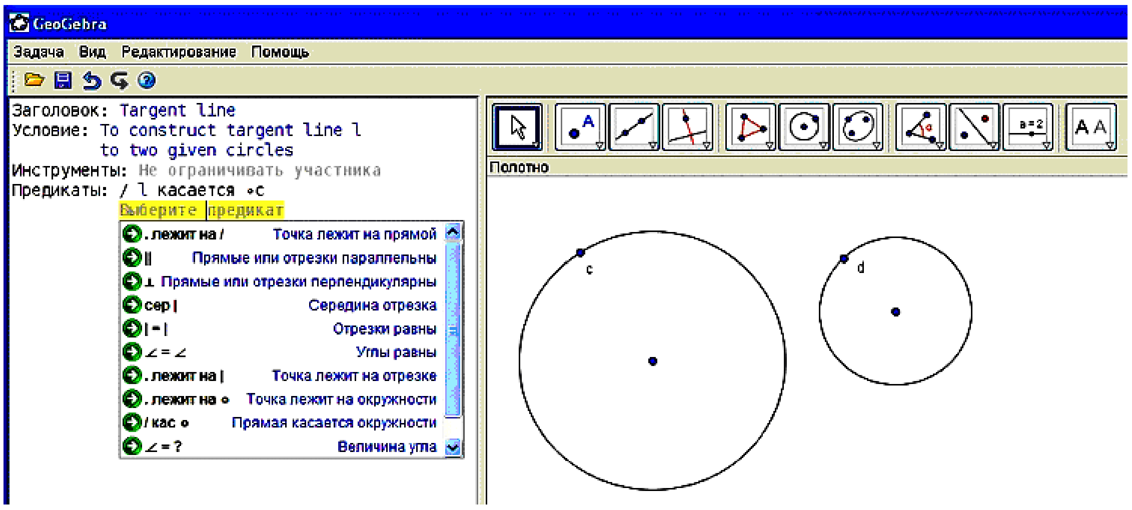
The screen-shots on Figures 1-3 represent the operation of the WiseTasks_Geometry program that is based on combination of the developed problem description and solutions verification language with the GeoGebra environment used to build constructions by the student. The Figure 1 shows the statement enter window, which consists of problem heading, informal heading description (in the combinatorics problems support system described below, it is filled in automatically to avoid the mismatching with the formal problem description), tools selection menu, that are available for the student (by default, all the tools are available) and the editor for consecutive entering of the predicates, describing the relations between the defined elements and the ones to find, that is, in other words, the formal description of the problem statement.



```
Заголовок: Construction of a Square ABCD
Условие: To costruct square ABCD
         with AC and BD diagonals
Инструменты: Не ограничивать участника
Предикаты: | AB || | CD
           | AB перп. | BC
           | AB = | CD
           | BC || | AD
           | AB = | BC
```

Figure 1: *the formal description of the problem statement*

Let's note that the data can be defined by the drawing or entered from the drawing (Figure 2).



```
Заголовок: Targent line
Условие: To construct targent line l
         to two given circles
Инструменты: Не ограничивать участника
Предикаты: / l касается ∘c
           Выберите предикат
```

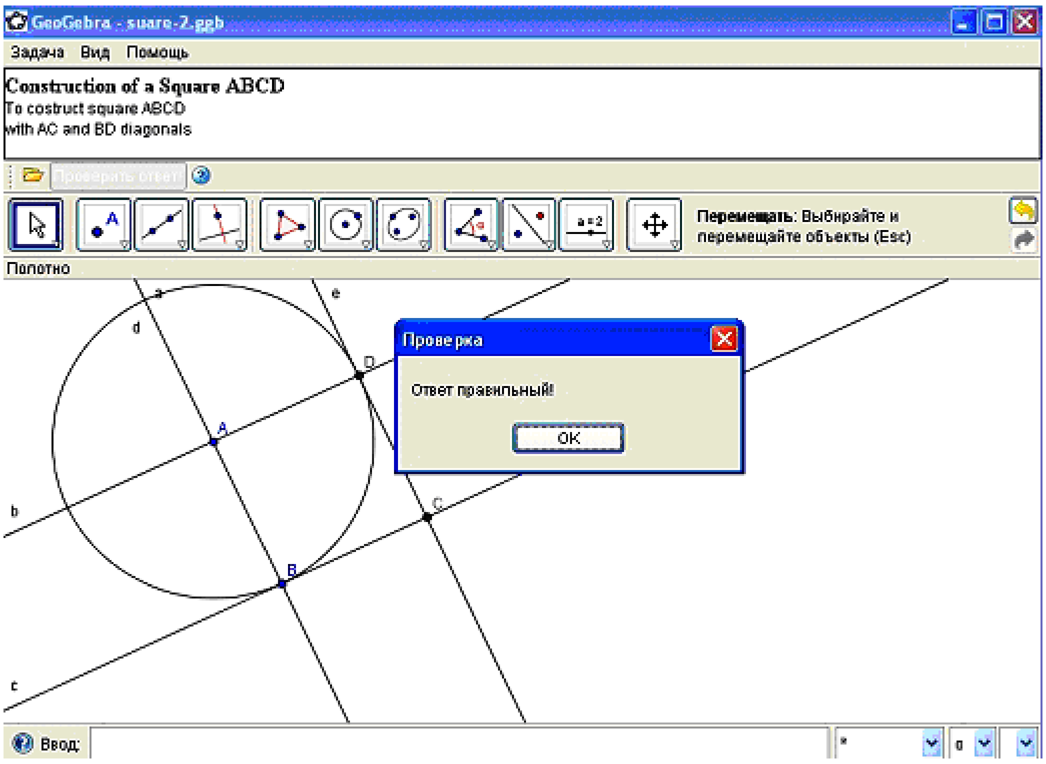
. лежит на /	Точка лежит на прямой
‖	Прямые или отрезки параллельны
⊥	Прямые или отрезки перпендикулярны
сер |	Середина отрезка
| = |	Отрезки равны
∠ = ∠	Углы равны
. лежит на |	Точка лежит на отрезке
. лежит на ∘	Точка лежит на окружности
/ кас ∘	Прямая касается окружности
∠ = ?	Величина угла

Figure 2: *data can be defined by the drawing or entered from the drawing*

The problem is saved in the xml-file with an additional file in the GeoGebra format, which includes the data on the constructions on the drawing. In the student's framework, the problem statement is returned; the tool bar contains only the allowed tools.

The student can perform any actions to solve the problem. The created dynamic drawing can be verified by executing the command "Check solution" in the area of the formal description of the problem.

Let's note, that procedurally the system allows entering wider class of the verified problems without setting up the parameters values. For example, the problem shown on the figures 1 and 3 includes the construction of a square by means of any tools of the dynamic geometry.



Construction of a Square ABCD
To costruct square ABCD
with AC and BD diagonals

Проверка

Ответ правильный!

OK

Figure 3: *construction of a square by means of any tools of the dynamic geometry*

Meanwhile, the task does not include neither the length of the side nor the square position that makes the problem more natural in terms of its mathematical contents.
Another feature of the WiseTasks system is the possibility for the student to set up a problem by himself, without knowing its solution. Figure 2 shows more difficult problem of a tangent line construction to two circumferences. This problem is easy to set, but difficult to solve. But the system will check the solution of this problem proposed by the student, though the solution has not been entered into the system!

6. Languages for combinatorial problems description

The idea of the combinatorial problems description, solution and verification differs from the geometrical application described above. But at the same time the general approach of problem representation may be the same: to represent the task as xml-file including the exact problem description and the verification method based on the comparison of the provided answer with the problem statement.

Let's examine one class of combinatorial problems related to the finding the formula for calculating the number of the objects complying with a certain criteria.
The criteria in this case may be a formal set definition. The operations on sets such as Cartesian product which allows constructing the objects with a variety of features, or projection operation, which allows marking those features and finding relationships between them, are included in the problem description language. The language can be extended by including other operations on sets or even with the new objects. For example, the possibility to describe the groups of symmetry in a shape was included specifically for the problem of Platonic solids colourings (Burnside's lemma).
For the solution description language the language of arithmetical expressions is used, with the addition of general combinatorial functions such as number of permutations (factorial), number of combinations.

7. Problem statement and solution description domain-specific languages editors in combinatorics

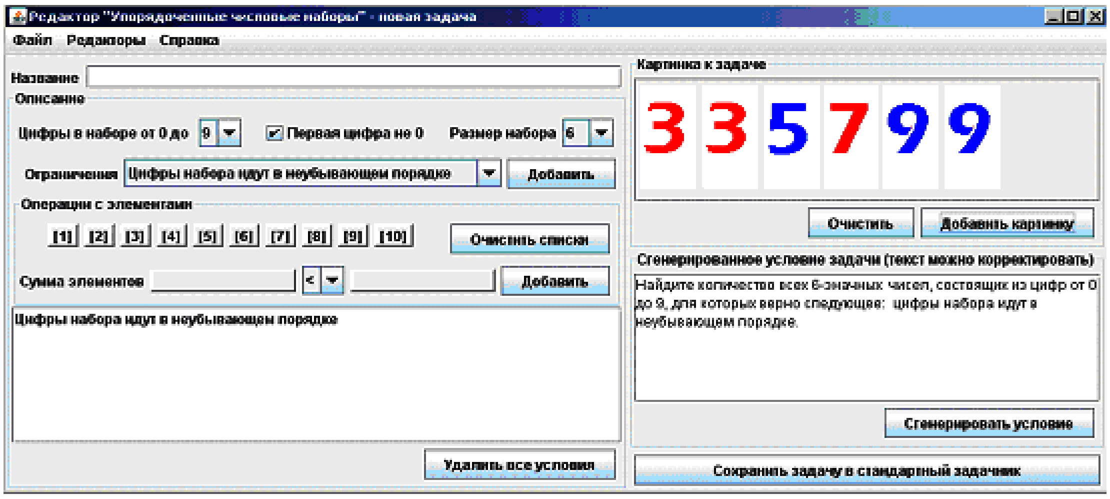
The following two approaches are used to create an easy-to-use problem statement editor:

1. Use of a general editor explaining the entered object at the time of input (by selecting the possible option), just like the editor of the geometrical problems described above
2. Creating the simple editors for each class of the combinatorics problems. In such editors, the problem is constructed by combining the parameters which define the set of the problems of the same class. The resulting statement will be automatically translated into general editor language.

The Figure 4 shows an example of the editor based on the problem description using the variety of the ordered numeric character sets. The parameters are: the range, the count and the position numbers of the digits on which the restriction is imposed. The restrictions on the digits of the set and the sum of the digits standing on the specific positions are defined using the equality and inequality.

For example a popular Russian problem about the "lucky tram tickets" can be easily formulated using this language: how many six-digit decimal sets with the sum of the first three digits being equal to the sum of the rest exist?

After the parameters are chosen and the restrictions are imposed, one should click the Generate button, so the formal problem description is automatically translated into verbal description (that guarantee the equivalence of the textual and formal statement). The editor also allows modifying the automatically generated textual description in order to make it more literary. Moreover, a picture may be added to the problem by combining the default picture objects (digits in the given example). A correct combination may be used as a picture for the problem.


Редактор "Упорядоченные числовые наборы" - новая задача

Название

Описание

Цифры в наборе от 0 до: 9

[✓] Первая цифра не 0

Размер набора: 6

Ограничения: Цифры набора идут в неубывающем порядке

Сумма элементов <

Цифры набора идут в неубывающем порядке

Картинка к задаче

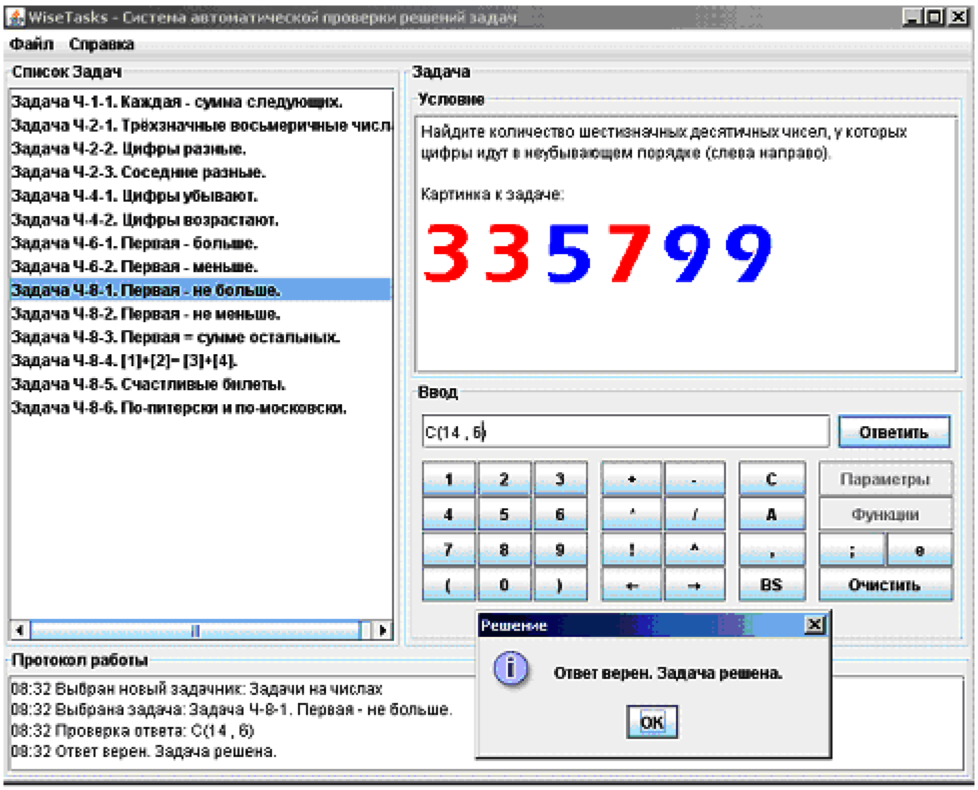
Сгенерированное условие задачи (текст можно корректировать)

Найдите количество всех 6-значных чисел, состоящих из цифр от 0 до 9, для которых верно следующее: цифры набора идут в неубывающем порядке.

Figure 4: *editor for problems of the class "ordered numeric sets"*

Using the student part of the program, one can select any problem from the predefined sets (standard book, problems proposed by the student, etc.)

A special formula editor with predefined combinatorial functions is provided to enter the answer to a problem. The answer is verified by comparing the number of the possible variants generated by the program based on the problem formal definition with the arithmetical expression entered by the user. The comparison is always accurate since the program is supposed to deal with the long integers. The student part of the program keeps a record on what the user entered and the received results of verification (Figure 5).


WiseTasks - Система автоматической проверки решений задач

Файл Справка

Список Задач

Задача Ч-1-1. Каждая - сумма следующих.
Задача Ч-2-1. Трёхзначные восьмеричные числ
Задача Ч-2-2. Цифры разные.
Задача Ч-2-3. Соседние разные.
Задача Ч-4-1. Цифры убывают.
Задача Ч-4-2. Цифры возрастают.
Задача Ч-6-1. Первая - больше.
Задача Ч-6-2. Первая - меньше.
Задача Ч-8-1. Первая - не больше.
Задача Ч-8-2. Первая - не меньше.
Задача Ч-8-3. Первая = сумме остальных.
Задача Ч-8-4. [1]+[2]=[3]+[4].
Задача Ч-8-5. Счастливые билеты.
Задача Ч-8-6. По-питерски и по-московски.

Задача

Условие

Найдите количество шестизначных десятичных чисел, у которых цифры идут в неубывающем порядке (слева направо).

Картинка к задаче:

335799

Ввод

C(14 , 6)

Протокол работы

```
08:32 Выбран новый задачник: Задачи на числах
08:32 Выбрана задача: Задача Ч-8-1. Первая - не больше.
08:32 Проверка ответа: C(14 , 6)
08:32 Ответ верен. Задача решена.
```

Решение

Ответ верен. Задача решена.

Figure 5: *students interface*

8. An experiment

To experimentally confirm the stated hypothesis about the importance of the bilingual approach in the creation of software tools the combinatorics educational materials were developed: ten sets of problems coverings all the topics of the combinatorics and some topics of the number theory. The sets of problems were implemented as web-applications and put on the Internet site. The students (16-17 years old) were assigned to: 1) Solve several problems from the problems sets 2) Devise a problem, formalize it and put it in some of the problems sets, solve it and publish on the site. 3) Solve several problems of other students.

There were about 200 students participating in the experiment. The analysis shows that 1) the problem set implemented as an application can be effectively used in the distance learning 2) the existing teaching methodology does not facilitate originality in devising of problems, there were just a few original problems 3) about 10% of participants proposed interesting problems with unobvious solutions, that shows, that it is feasible to use the developed tools for support of the productive approach in the teaching of mathematics. 4) All participants managed to check the correctness of solutions of problems they devised without a teacher, so the teacher did not have to perform any routine activities, and were able to concentrate on the problems contents and the approach students offered to solve a problem (that was certainly correct because was ensured by the system).

During the second experiment the “mathematical battle” competition was organized for the students that were 12-15 years old. A participant should devise problems that were as hard as possible but still solvable by herself, give them to the opponents, get the problems prepared by the opponents and solve them. The analysis of the results demonstrated that the conclusions presented above still hold for the younger students. Also it demonstrated certain issues that arise with the usage of this technology. It is hard for students to foresee all the problems of automatic solutions verifications that would arise for their statements. The most active participants invented problems

that needed intensive computation and it was not possible for the system to complete them in reasonable time.
Needless to say, the last issue may be addressed by adding additional restrictions to the languages used to describe problems statements.

9. Conclusions

The introduced approach for the mathematical problems automatic preparation and verification is based on the different languages used for a problem description and solution. The formal description of a problem allows verifying the solution even if it is unknown. The usage of the problem description languages supposes the development of the difficult problems (not tests) representation formats and allows considering them apart from the methods of the solution verification. The differentiation between the problem statement description editors and problem solution description editors gives an opportunity to make a research in the application domain, which the problem description editors were created for. The students will be able to create themselves the new tasks and then solve them together or with the teachers. The proposed approach gives the new technological opportunities for the already established mathematical competitions like “mathematical tournament” when two teams exchange the problems. The tools for the problems creation support can bring the new feature in the competition: creation and preliminary solution of a problem.

10. References

1. Vygotsky L. *Thought and language*. Cambridge, MA: MIT Press, 1934.
2. Papert, Seymour. *Mindstorms: Children, Computers, and Powerful Ideas,* 1980, ISBN 0-465-04674-6.
3. Bogdanov M., Pozdnyakov S., Pukhov A. *Multiplicity of the knowledge representation forms as a base of using a computer for the studying of the discrete mathematics*. PEDAGOGIKA, v.96, 136-142, 2009.
4. Pozdnyakov S., Ivanov S. *Productive teaching of mathematics or how information technologies can support intellectual freedom of the learner*. Proc. 10-th International Congress on Mathematical Education (National presentation: Russia, Selected materials, 115-124). Copenhagen, Denmark, 2004.
5. Wertheimer, Max. *Productive thinking*. Harper & Row, enlarged edition, 1945.
6. Kobelsky V.L., Stepanova E.V. *Computer educational system “Planimetry 7-9”*. Computer Tools in Education Journal (rus), 2-2001, 59-67, 2001.
7. Viktor Freiman, Djordje Kadijevich, Gerard Kuntz, Sergey Pozdnyakov, Ingvill Stedoy. *Challenging mathematics beyond the classroom enhanced by technological environments (in Challenging Mathematics In and Beyond the Classroom: Chapter 3).* C Springer Science + Business Media, LLC 2009.
8. G. Goguadze, A. Gonz´alez Palomo, E.-Melis. *Interactivity of Exercises in Active-Math. Towards Sustainable and Scalable Educational Innovations Informed by the Learning Sciences - Sharing Good Practices of Research, Experimentation and Innovatio*. Edited by Chee-Kit Looi, David Jonassen, Mitsuru Ikeda, Frontiers in Artificial Intelligence and Applications, Volume 133, 109 – 115, 2005.
9. Ulrich Kortenkamp, Axel M. Blessing, Christian Dohrmann, Yves Kreis, Paul Libbrecht, Christian Mercat. *Interoperable interactive geometry for Europe – first technological and educational results and future challenges of the InterGeo project*. Proceedings of CERME 6, January 28th-February 1st 2009 (1150-1160). Lyon, France, 2010.
10. *Meta Programming System* http://www.jetbrains.com/mps/
11. Polya, G. *How to Solve it*. Princeton University Press, 1945.