Elementary School Word Problem Solution Environment TEKSTER

Rein Prank, Evari Koppel, Joosep Kibal, Katrin Valdson, Joosep Norma

University of Tartu (Estonia)

Word problems are a topic that **caus**es **difficulties** for many pupils in basic school.

Therefore it is desirable to try also create computer software for supporting solution of word problems.

Two levels of word problems

The **solution methods** of word problems (and correspondingly the tasks themselves) **can be divided into two levels**.

Level 1. The students
choose at each solution step a quantity and find its value.
In elementary grades each solution step consists of one arithmetical operation. Further (Geometry, Physics, Chemistry) the formulas for finding the next quantity can contain more operations.

Level 2. Tasks do not have reasonable sequence of sensible quantities. The students
1) denote the quantities by variables,
2) build from the known and unknown quantities
an **equation or equation system**,
3) find the unknown quantities by solving the equation/system.

- TEKSTER is an **environment for Level 1**,
  by successive calculation of quantities.

In our paper we consider

- The solution dialog in student program,
- Attributes of tasks entered by the teacher
- What is necessary for further development

Our programs use

- computer algebra system Maxima for checking of equivalence of algebraic expressions and
- a prover-like Solver for checking the solvability of the tasks, for generation of demo solutions and hints.

Solution dialog : Steps

In TEKSTER, the student solves the word problem **step by step**, finding at each step a quantity (that has some comprehensible name).

At each step he/she performs the following substeps:

1) Composes a question/sentence that tells what quantity will be calculated at this step,
2) Forms a one-operation arithmetic expression that finds the desired quantity,
3) Calculates the result of the operation.

The program

- checks each substep and
  requires correction in case of mistakes/misplanning,
- Provides hints for the next step

Demo

Task composition.
Stage 1: Text and variables

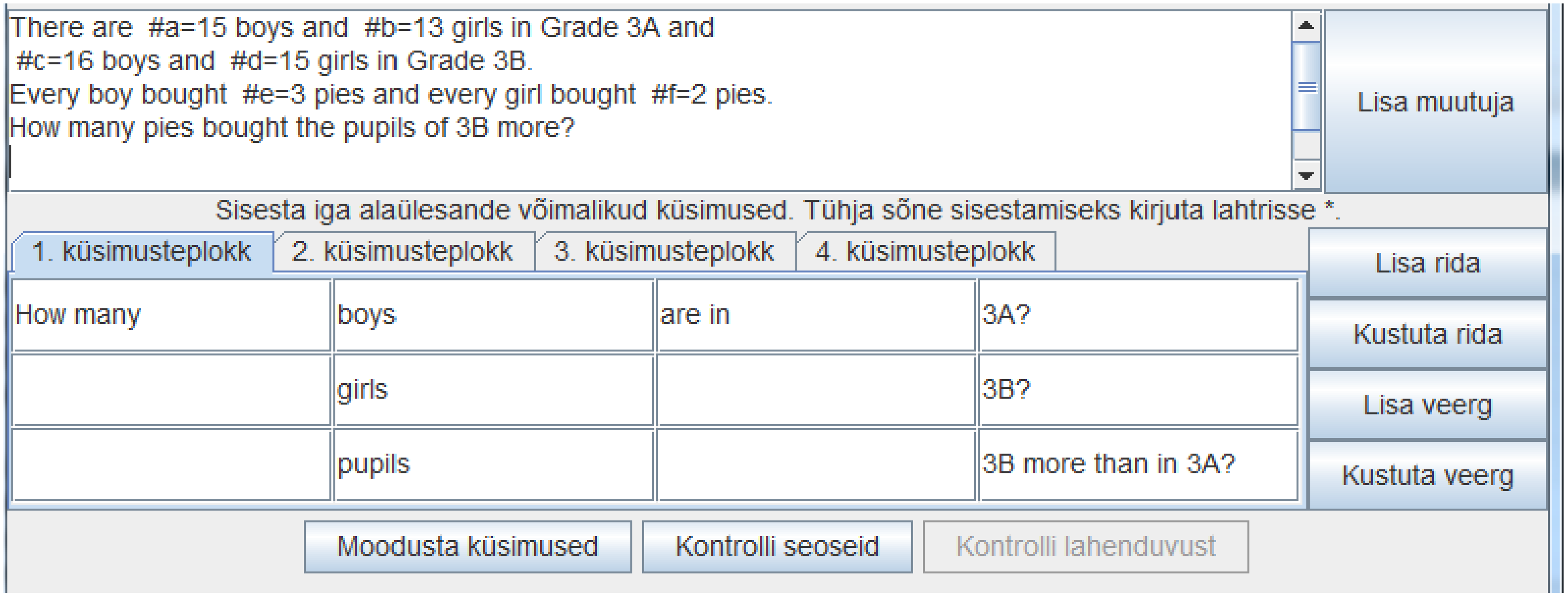
The **tasks are composed** in teacher program.

First the teacher enters the text of the task in the upper box. Thereby he/she marks the quantities that can be used as data in the solution and the program assigns them the variables.

Demo

Task composition
Stage 2: Question composition blocks

In the middle part of the window the teacher builds question composition blocks (up to four) that enable to compose necessary questions.



There are #a=15 boys and #b=13 girls in Grade 3A and
#c=16 boys and #d=15 girls in Grade 3B.
Every boy bought #e=3 pies and every girl bought #f=2 pies.
How many pies bought the pupils of 3B more?

Sisesta iga alaülesande võimalikud küsimused. Tühja sõne sisestamiseks kirjuta lahtrisse *.

How many	boys	are in	3A?
	girls		3B?
	pupils		3B more than in 3A?

Form questions Check expressions Check solvability

Task composition
Stage 3: Assigning expressions to the questions

- When the teacher pushes the button
  ‘Moodusta küsimused’ (‘Form questions’)
  the program writes in the left side of the third part of the window all the sentences that can be built from the question blocks.
- The teacher enters now
  the expressions that express the asked quantities through the variables from the text panel. One of the questions should be marked as corresponding to the final answer.
- The program checks syntax of the expressions
  when the user pushes
  ‘Kontrolli seoseid’ (‘Check expressions’).

Demo

Task composition
Unacceptable questions

If the expression box after a question remains empty then selection of this question by the student ends with message that this question is not reasonable.

The teacher can leave a box empty by different reasons:

- Some combinations of phrases can give meaningless sentences
- Some questions can point to quantities that cannot be calculated from input variables
- Some quantities are not usable for any reasonable solution (in our example - the numbers of pupils in 3A and 3B)
- Some quantities can be not desirable because the negative numbers are not yet known to the pupils
- Also, the teacher can confine the solution space by making some (less desirable) solutions impossible

On the other hand, the box can remain empty because the teacher did not notice some solution possibility

Task composition .
Stage 4:Checking solvability

If the result of syntax check is positive, it makes active the button for the last step of task composition – 'Check solvability'.
The program uses its **Automated Solver** and
builds the solution or
gives the message that it is impossible to find the final answer, counting at each step the value of one of the expressions that are in the right part of the lowermost panel.

In case of positive result the task is ready for saving in the file.

Options and limitations

- TEKSTER enables to use integers and decimal fractions.
- Instead of concrete values the task can contain input variables with randomly generated values.

- The student program records in solution file the choices and mistakes made by the student.
- The teacher program enables to review the solution files.

More advanced mathematical tasks

TEKSTER can be used also for some more „adult“ tasks:

Get the value 15 starting with value 1 and performing minimal number of operations.

In TEKSTER:

The value of a is #a=1 . Get the value 15 applying minimal number of additions and subtractions.

2   4 8 16 15

Further ideas

Next versions of could enable solution of
more complex tasks and
tasks from **Geometry, Physics** and **Chemistry**.

Necessary additions:

- Steps with more than one operation
- Using constants that are not included in the text of the task
- Garbage collection
- Obvious/minimal solutions