Emulating Human Supervision in an Intelligent Tutoring System for Arithmetical Problem Solving

David Arnau, Miguel Arevalillo-Herráez, and José Antonio González-Calero

Abstract—This paper presents an intelligent tutoring system (ITS) for the learning of arithmetical problem solving. This is based on an analysis of a) the cognitive processes that take place during problem solving; and b) the usual tasks performed by a human when supervising a student in a one-to-one tutoring situation. The ITS is able to identify the solving strategy that the student is following and offer adaptive feedback that takes into account both the problem's constraints and the decisions previously made by the user. An observational study shows the ITS's accuracy at emulating expert human supervision, and a randomized experiment reveals that the ITS significantly improves students' learning in arithmetical problem solving.

Index Terms—Computer-assisted instruction, cognitive simulation, intelligent tutoring systems, knowledge modeling

1 Introduction

There are many tasks a teacher has to address when tutoring a student in solving story problems, e.g., check the validity of each solution step, identify the learner's weaknesses, provide explanations and feedback, generate learning sequences, manage time, etc. Although some of these tasks are independent of the rest, the determination of the validity of the student's actions is critical. This is so because the student's errors and difficulties are the basis to determine the student's profile, provide adaptive feedback and remediation, and generate an appropriate sequence of problems. This kind of feedback requires that the tutor be able to identify different ways of solving a given problem, when this is possible.

In fact, the capacity to support multiple solution paths has been claimed as necessary to yield a more effective learning in technological environments [1]. Thus, remediation in word problem solving could be improved by considering the solution scheme that the student is currently following (or has in mind), which can be worked out from his/her previous actions.

Some attempts have been made to build computer systems that facilitate the learning of arithmetical word problem solving (e.g., HERON [2], Story Problem Solver [3], WORDMATH [4], MathCAL [5] and AnimalWatch [6]). These systems emulate some of the tasks that a human tutor carries out to support the students' learning by, for example: 1) providing conceptual schemes that are required to interpret problem statements, (e.g., Story Problem Solver, WORDMATH); 2) supervising the validity of the problem solving process (e.g., HERON and MathCAL); or 3) building a student model to provide a problem sequence adapted to his/her characteristics (e.g., AnimalWatch).

The main technical difficulty when building interactive environments for the learning of word problem solving has been to combine the flexibility of supporting all potentially valid user actions with the ability to provide adequate remediation. Usually, system designers have preferred to provide supervision at the expense of restricting potentially valid solution paths. This has been commonly achieved by, e.g., 1) limiting the complexity of the problems; 2) breaking the problem down into simpler sequential steps that users necessarily have to follow; and/or 3) using hard coded help and error messages specifically designed for each problem.

By contrast, the design of most of these systems has been mainly focused on the user interface, in an attempt to promote some solver's cognitive processes. For example, HERON or MathCal force students to use tree diagrams to organize the problem solving process, and AnimalWatch attempts to engage students by using a teaching sequence based on narratives about endangered species. However, neither of the three support the possibility that a user starts several simultaneous solution paths.

From a technological and educational point of view, fundamental consequences of these approaches are that they impose important restrictions on the range of problems that can be used, significantly increase the difficulty of dynamically adding new problems, and make it harder to determine an accurate student model.

The necessity of analyzing the human tutoring in order to implement their behaviors in interactive learning environments has been highlighted in several studies [7], [8], [9]. In this paper, we present an intelligent tutoring system (ITS) that is able to supervise a student when solving an arithmetic word problem, without imposing any restriction on the solution path. The system is based on a careful analysis of the usual tasks performed by a human when supervising the student in a one-to-one tutoring situation. This ITS is

- *D. Arnau is with the Department of Didactics of Mathematics, University of Valencia, Avda. Tarongers, 4. Valencia 46022, Spain. E-mail: david.arnau@uv.es.*
- *M. Arevalillo-Herráez is with the Computing Department, University of Valencia, Avda. de la Universidad s/n. Burjassot., Valencia 46100, Spain. E-mail: miguel.arevalillo@uv.es.*
- *J. A. González-Calero is with the Department of Mathematics, University of Castilla-La Mancha, Plaza de la Universidad, 3. Albacete 02071, Spain. E-mail: jose.gonzalezcalero@uclm.es.*

Manuscript received 8 Apr. 2013; revised 29 Jan. 2014; accepted 4 Feb. 2014. Date of publication 19 Feb. 2014; date of current version 2 July 2014.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TLT.2014.2307306

able to identify the solution path that the solver is following and provide adaptive remediation by taking into account both the problem's constraints and the decisions previously made by the user while solving the problem. We also describe two experimental studies that support the following hypotheses: (a) expert and novice teachers provide different advices; (b) the aids provided by the ITS are consistent with those offered by expert teachers; (c) the system helps students solve problems that they had not been able to solve previously with paper and pencil; and (d) the ITS contributes to improving the students' capacity of solving word problems in an arithmetical way.

2 Theoretical Framework

2.1 Arithmetical Problem Solving

From a cognitive perspective, arithmetical problem solving involves an analytical-synthetic process [10]. The purpose of the analytical part is to reduce the problem statement to information of a mathematical kind, where quantities are identified and connected by proceeding from unknown to known ones. During synthesis, the structure generated is used to calculate unknown quantities from other known quantities.

In principle, an ideal arithmetical problem solving can be described in terms of two independent sequential processes: a complete analysis of the problem is done first and then a synthesis is carried out. However, the demand for a complete analysis previous to the synthesis "is difficult for pupils of nine or 10 since it artificially cuts the pupils off from concrete numbers, without giving them a chance to rely on numerical data" [11, p. 63], . Hence, in reality, students alternate between analysis and synthesis, rather than carrying out both processes in a sequential fashion.

As the relations among quantities are rarely offered in an explicit way, the recognition of the situation described in the problem is especially important. This allows the evocation of schemes associated with the particular scenario [3], [12], [13], [14]. For example, a problem in which the present and future ages of a person are mentioned makes the student evoke a scheme that relates these two quantities, e.g., the future age is equal to the current age plus the time elapsed. These schemes allow one to establish relationships among quantities. These relationships may involve quantities mentioned in the statement and others evoked while solving the problem. The identification and use of these schemes requires that they have been previously found in other situations. Moreover, the success at solving a problem depends on the schemes a person can refer to [3]. For this reason, it is usual to use graphical representations to describe situations when a student is initiating at arithmetic word problem solving. This intends to consolidate the relation between certain verbal propositions (or contextual clues) and a scheme. However, as the students make progress and the complexity of the problems increases, this analysis needs to be automated so that schemes can be identified straight away from the text in the statement.

2.2 Analytical Reading of a Word Problem

To represent the knowledge about the mathematical structure of a problem we use the idea of an analytical reading [15], [16], [17]. An analytical reading can be defined as the result of the interpretation of a problem statement by an individual, and includes sufficient declarative knowledge to be able to solve the problem. From an algorithmic point of view, an analytical reading is composed of a set of independent relations among quantities that satisfy the following three conditions: (1) the number of relations is the same as the number of unknown quantities; (2) the unknown quantities include the ones that the problem statement asks for; and (3) all the quantities are necessary to work out the solution to the problem. Different individuals, with different knowledge and/or intentions, may generate different analytical readings. Next, we exemplify the idea of an analytical reading by using the problem Oranges: "We have stored 180 oranges in two baskets of different sizes. In the big basket we have put 30 oranges more than in the other one. How many oranges do we have in each basket?".

A possible solution would start by mentally setting the 30 extra oranges apart; then make an equal sharing to calculate the number of oranges in the small basket; and finally determine the oranges in the big one. This particular solution can be explained by an analytical reading (R1) that reduces the problem to three known and three unknown quantities. Known quantities are: the number of oranges, N (180); the excess of oranges in the big basket, Mbs (30); and the number of baskets, B (2). Unknown quantities are: oranges if we remove the excess, Nre; oranges in the big basket, Bb; and oranges in the small basket, Bs. These quantities are related by: $N = Bb + Bs$, $N = Nre + Mbs$ and $Nre = Bs \cdot B$.

Although relations have been represented by using algebraic language, this does not imply that the solution be also algebraic. In fact, the above structure of relations among quantities can give raise to an arithmetical solution. To achieve this, it is necessary to find at least one relation with just an unknown quantity. Such relations are called entries. An entry allows the calculation of an unknown quantity by using the values of other already known ones. For example, the relation $N = Nre + Mbs$ in R1 is an entry, and allows the calculation of Nre. The use of this relation causes that $Nre = Bs \cdot B$ becomes an entry. By repeatedly applying this process, it is possible to determine the quantity (or quantities) asked for in the statement. An analytical reading is said to be arithmetical when all unknown quantities in the problem statement can be calculated by iteratively using entries (see [16], [17], [18]).

It is convenient to point out that this problem has several other analytical readings. In order to make the presentation clearer, we only take into account another arithmetical one. This different analytical reading (R2) for the problem above consists in dividing the oranges into two identical groups, and then mentally transferring half of the excess from one group to the other. This analytical reading implies using three known and four unknown quantities. The known quantities are the same as in R1. The unknown quantities in this case are: oranges in the big basket, Bb; oranges in the small basket, Bs; half of the oranges, Nh; and half of the excess of oranges, Eh. These quantities would be related by: $N = Nh \cdot B$, $Mbs = Eh \cdot B$, $Bb = Nh + Eh$ and $N = Bb + Bs$.

Analytical readings are not necessarily represented in this way at a mental level, but this is a convenient and

schematic representation that is useful to track the students' actions with the ITS, suggest a next suitable step, and provide adaptive remediation.

2.3 Human Tutoring of Word Problem Solving

To be able to supervise a student when solving a story problem, a human tutor needs to be able to identify alternative solution paths and determine which one the student is following. This is equivalent to identifying the analytical reading that the student is using and it allows the tutor to detect mistakes, spot unnecessary operations and provide adequate feedback and support. The information required for this task mainly comes from direct observation of the sequence of steps that the learner has already taken.

In principle, a human tutor would consider as incorrect any solution step that does not match a relation in any of the analytical readings he/she is able to generate. This means that, if a tutor is not able to solve a problem in a certain manner, he/she may interpret a right calculation as a wrong step. In the problem above, an inexperienced human tutor may consider the operation $180/2$ as incorrect, while it is a right step according to the analytical reading R2. Hence, the ability to identify all potential ways to solve a problem has a large impact on the tutoring quality.

Another important issue is related to the detection of unnecessary operations. An operation is unnecessary when, despite being correct, it does not match a relation in the solution path supposedly followed by the learner. However, apparently unnecessary operations may correspond to a legitimate deviation from the current reasoning. For example, a student may have realized of a fact that allows for a shorter solution, and hence switched to a different analytical reading. A human tutor would judge the convenience of the action by analyzing the advantages and disadvantages of changing reasoning.

Again, feedback quality heavily depends on the tutor's ability to determine the student's intentions, i.e., the particular solution scheme that the student is using. Suggestions that are not consistent with the analytical reading that the learner has in mind can potentially cause confusion, and lead to frustration or conceptual misunderstandings. Moreover, this form of supervision in which the tutor adapts feedback to the student's line of reasoning is a necessary skill. This is so because the initial determination of the student model has to be done with a minimum intrusive intervention, to let the student reveal his/her cognitive tendencies, strengths and weaknesses as a problem solver.

Still, providing aids which do not alter the solution path followed by the student may not always be the best teaching strategy. To make appropriate decisions, the tutor may combine multiple criteria, e.g., the current skills of the student, difficulties previously identified when observing other students' solutions, or the number of operations that would be left under the current and other analytical readings.

3 ITS Characteristics

3.1 ITS Architecture and Graphical User Interface (GUI) Features

According to [19], model-tracing tutors are recommended when the solutions contain low level information or when help and/or error management are required during problem solving. Consequently, the ITS follows a model-tracing architecture, and its functioning is supported by both procedural and declarative knowledge.

Declarative knowledge needs to be specified for each problem in XML format. The XML specification contains the quantities and relations that appear in all potential analytical readings for a problem, and a series of descriptions that allow the system to communicate with the student in natural language, e.g., when generating personalized feedback. Optionally, the XML can also include a list of incorrect relations among quantities to help respond to known typical mistakes. These are included as *ErroneousPath* elements in the XML description. Fig. 1 shows an XML specification for the problem Oranges described in Section 2.2. *KnownQuantity* and *UnknownQuantity* elements are used to specify the quantities that may appear in the different analytical readings associated with the problem. All quantities are given a name, which is used to refer to them when stating existing relations among quantities. Analytical readings are represented by using *Graph* elements. Each *Path* child element represents a relation among quantities. The attributes *type*, *result* and *nodes* are used to specify the relation. The type of a relation can be either *Addition* or *Multiplication*; the value of the attribute *result* corresponds to the name given to one of the *KnownQuantity* or *UnknownQuantity* elements; and the attribute *nodes* contains a comma separated list with the names of all other quantities that take part in the relation. For example, the first element in the analytical reading R1 represents the relation $N = Bb + Bs$.

The procedural knowledge, which is needed to supervise the student and decide on the most adequate tutoring actions, is embedded in the program. This independence between procedural and declarative knowledge allows one to add new problems to the system by just providing their corresponding XML specification. There are no limitations regarding the complexity of the arithmetical problems that the system is able to handle, other than the expert's ability to provide a complete XML specification.

When a user selects a problem, the ITS displays the statement and generates a calculator-like component. This component contains a button for each known quantity that appears in any of the available analytical readings. It also includes buttons for the four basic arithmetical operations. Users can only introduce arithmetical expressions by using this calculator. Each time that a new quantity is defined, a new button is created. This allows the user to employ the new quantity to define others. In addition, defined quantities are displayed on a table that includes both their values and descriptions. Descriptions may be introduced by users in the *Description* input field (see Fig. 2). If the user leaves this field empty, the program automatically extracts the description from the XML definition. The ITS considers that a problem has been solved when the learner has used all the relations, and hence determined all the unknown quantities in one of the analytical readings. When the ITS detects that the student has solved the problem, a final score is calculated and a report is shown. The final score is in the range from 0 to 10, and depends on several factors, such as the number of errors and hints requested or the number of unnecessary quantities that have been calculated. This

```
<?xml version="1.0" encoding="UTF-8"?>
<Wrapper>
<Exercise name="Oranges">

<Text>We have stored 180 oranges in two baskets of different sizes. In the big
basket we have put 30 oranges more than in the other one. How many oranges are
there in each one?</Text>

<KnownQuantity name="N" value="180">
        <Description>NUMBER OF ORANGES</Description>
</KnownQuantity>
<UnKnownQuantity name="Bb">
        <Description>ORANGES IN THE BIG BASKET</Description>
</UnKnownQuantity>
<UnKnownQuantity name="Bs">
        <Description>ORANGES IN THE SMALL BASKET</Description>
</UnKnownQuantity>
<KnownQuantity name="Mbs" value="30">
        <Description>EXCESS OF ORANGES IN THE BIG BASKET</Description>
</KnownQuantity>
<KnownQuantity name="B" value="2">
        <Description>NUMBER OF BASKETS</Description>
</KnownQuantity>
<UnKnownQuantity name="Nre">
        <Description>ORANGES IF WE REMOVE EXCESS</Description>
</UnKnownQuantity>
<UnKnownQuantity name="Nh">
        <Description>HALF OF THE ORANGES</Description>
</UnKnownQuantity>
<UnKnownQuantity name="Eh">
        <Description>HALF OF THE EXCESS OF ORANGES</Description>
</UnKnownQuantity>

<Graph name="R1">
        <Path type="Addition" result="N" nodes="Bb,Bs"></Path>
        <Path type="Multiplication" result="Nre" nodes="B,Bs"></Path>
        <Path type="Addition" result="N" nodes="Nre,Mbs"></Path>
</Graph>
<Graph name="R2">
        <Path type="Multiplication" result="N" nodes="Nh,B"></Path>
        <Path type="Multiplication" result="Mbs" nodes="Eh,B"></Path>
        <Path type="Addition" result="Bb" nodes="Nh,Eh"></Path>
        <Path type="Addition" result="N" nodes="Bb,Bs"></Path>
</Graph>

<ErroneousPath type="Addition" result="Bb" nodes="Nh,Mbs" description="As you have
divided the total of oranges in two parts, in each part there will be half of the
excess of oranges"></ErroneousPath>

</Exercise>
```

Fig. 1. XML for the problem Oranges presented in the text.

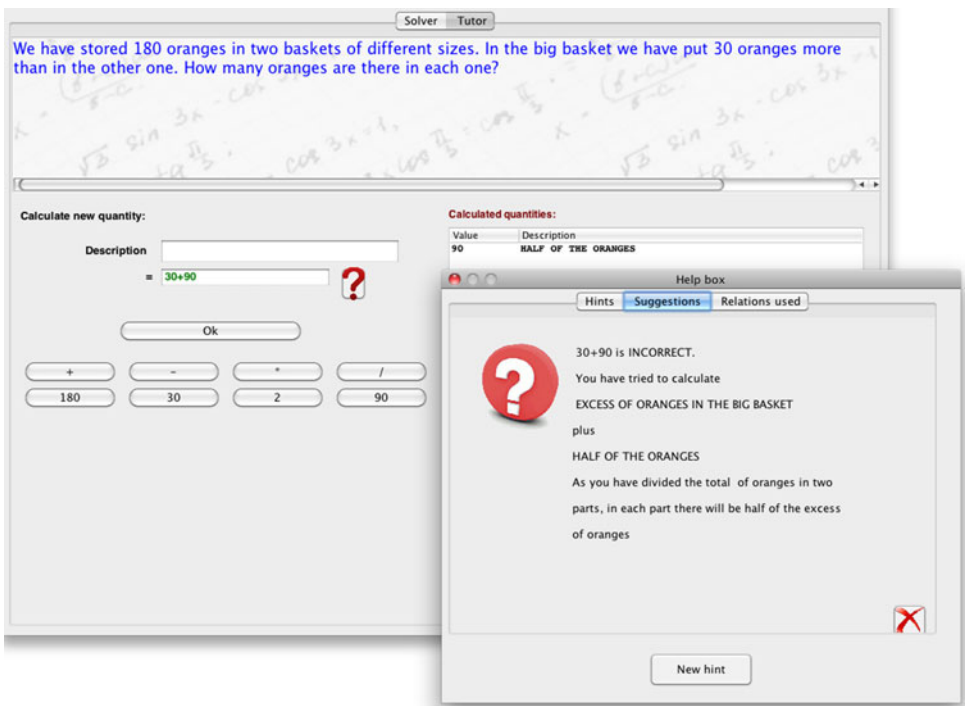

Fig. 2. A screenshot of a typical error dialog that includes a pre-defined help message.

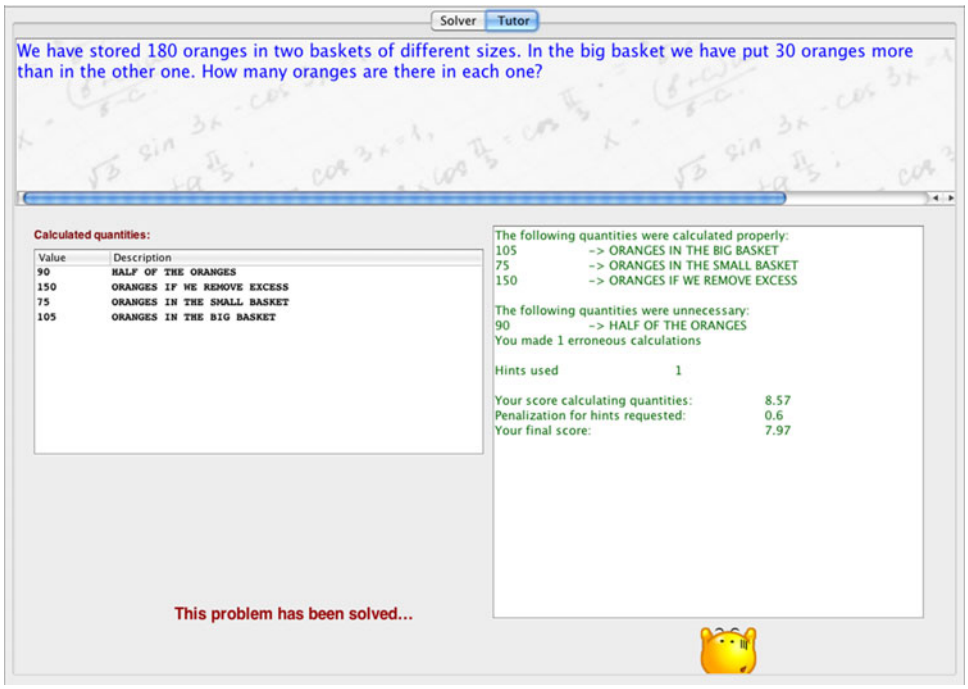

Fig. 3. An example of the final report displayed once a problem has been correctly solved, providing performance information.

information is all displayed on the report (see Fig. 3). The final score was introduced in response to the gaming behavior observed in previous experiments as a consequence of students being able to request hints at no expense.

The graphical user interface has been carefully designed to enforce a systematic approach to arithmetical word problem solving. First, it forces users to follow the same sequence of steps as they would do when solving the problem by using paper and pencil. Second, it allows solvers to start a new solution path while still keeping other previous solution lines active. Third, the ITS is powered by a domain-specific inference engine that is able to tutor algebraic problem solving [18], [20]. However, to avoid trial and error or algebraic approaches, the GUI does not offer the possibility to assign a numeric value or a letter to an unknown quantity.

The approach used to process the input allows the system to handle numerical distractors seamlessly. A distractor would cause the creation of a button that represents the given value, but the quantity would not appear as part of a relation in any of the analytical readings. Therefore, any attempt to use the distractor would result in an invalid expression.

3.2 ITS Emulation of Human Tutoring

To emulate the behavior of a human tutor, the fundamental actions identified in Section 2.3 need to be appropriately modeled and realized into procedural knowledge that can be incorporated into the system.

To process a user's input, the ITS needs to search for a matching entry in all the analytical readings associated with the problem. This is done in two stages. First, unused relations that contain the same operands as the user's input are identified. Then, the operations and the order of the operands are checked. If at least one matching instance is found, the ITS marks the relation as used. Then, the unknown quantity in the entry is marked as known and assigned the result of the calculation posed.

When a user's input does not match a relation in any of the analytical readings, the input is assumed incorrect. In this case, the system is able to generate automatic and consistent remediation (see Fig. 2). Feedback messages include a verbal description of the incorrect operation, which explains the lack of consistency of the calculation in the context of the problem. If the mistake matches an *ErroneousPath* element, the predefined text in the description field is added to the message. For example, the dialog window in Fig. 2 includes the text in the *description* field of the *ErroneousPath* element in Fig. 1.

The system is also able to provide hints on demand, according to the solution path that the student is hypothetically following. The criterion used to identify this path is to consider the analytical reading with the highest proportion of relations used. To this end, the system keeps track of which relations have already been used in each analytical reading. In the case of a draw, the analytical reading with the smallest number of unused relations is chosen. Fig. 4 shows an example of a hint message. This example assumes that the student had already calculated Nre, by using a relation that only appears in R1, and Nh, by using a relation that only appears in R2. The ITS uses the criterion above to determine that R1 is the most likely line of reasoning. The program then looks for an entry in R1 and generates a verbal hint message suggesting an operation using the values 150 (Nre) and 2 (B). This is done by filling a template with the descriptions and values of the known quantities, and the description of the unknown quantity that can be calculated by using the entry.

The criterion we have used tries to respect previous decisions made by the learner. Other criteria would lead

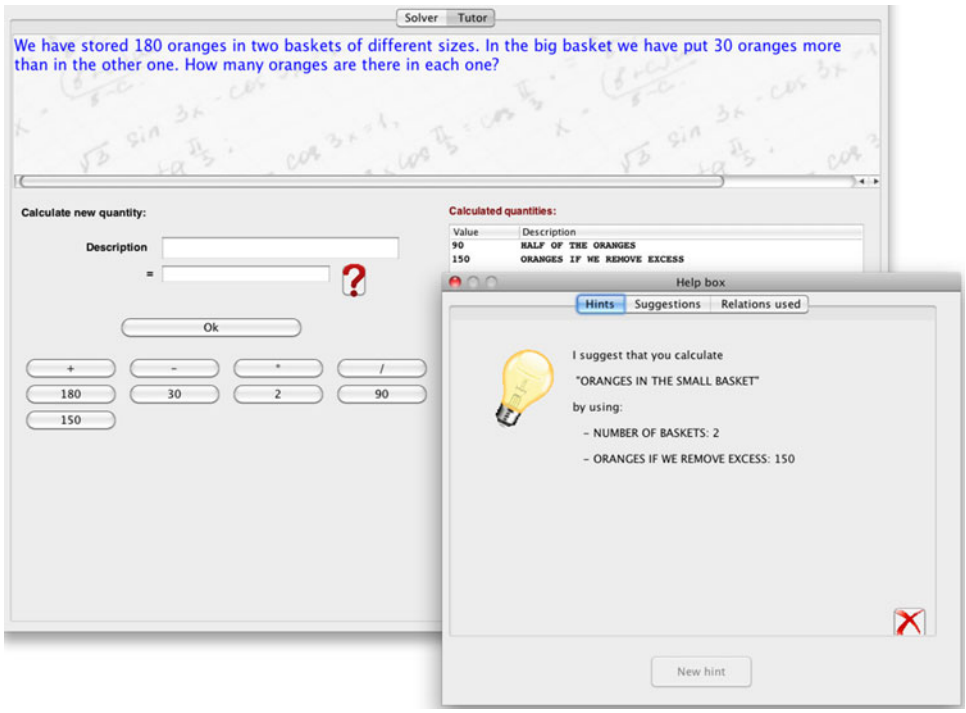

Fig. 4. A screenshot of a typical automatically generated hint provided on a student's request.

to offer more interventionist hints, e.g., conducting the solver to the analytical reading with the lowest number of relations.

4 Empirical Study 1: Emulating Human Tutoring

4.1 Participants, Material and Procedure

This first empirical study aims to test the hypotheses that there are differences between the hints provided by expert and novice teachers; and that the aids provided by the ITS are consistent with those offered by human experts.

A group of expert teachers and another composed of novice teachers participated in this observational study. Despite that there is no standard criterion to identify teachers with expertise in a mathematical task [21], it is quite common to use measures such as years of experience, personal academic background or educational outcomes of their students as expertise estimators [22]. In our experiment, the group of experts consisted of 19 university lecturers that teach aspects related to the learning of arithmetical problem solving. All of these teachers had from 10 to 30 years of teaching, combining both practical experience and a solid academic background. The group of novices consisted of 56 pre-service teachers, which were coursing a Bachelor degree in Education at a public university in Spain. All subjects in this group had successfully completed a course on teaching arithmetic and problem solving.

Both groups were administered a test composed of eight items. Each item consisted of a word problem statement and a solution fragment, representing an unfinished student solution for the problem at hand. Subjects were given instructions to provide a suggestion for the next operation, by taking into account the calculations already undertaken. They were informed that only arithmetic operations were allowed and also that suggestions had to be aimed at 10-11 years old students.

All problems could be solved by using several different analytical readings. Four solution fragments contained operations from a single analytical reading. These represent cases in which the student is following a consistent solution path. The remainder four solutions contained operations from at least two different analytical readings. These represent real situations in which the solver may have performed unnecessary steps or changed his/her strategy at some point. An illustrative example of each type of situation is provided in Table 1.

Hints provided by all subjects were classified as correct or incorrect. To this end, only answers that did not match a valid relation in any potential solution path were considered incorrect. For example, suggestions to use either $12 \cdot 400 = 4,800$ or $31 - 12 = 19$ as a next step in the first problem in Table 1 would both be considered correct, despite that the former expression is not consistent with the current line of reasoning. In addition, the correct aids provided by experts and novices were compared to the ones offered by the ITS and classified as an agreement when the suggestion was the same as the one that the ITS would have provided.

4.2 Analysis of Results

Table 2 shows that all aids provided by the expert group were correct. This is in contrast to the hints provided by novice teachers, which were incorrect in 17.86 percent of the cases. Moreover, correct suggestions made by the expert group were closer to the ones offered by the ITS. The average and standard deviation of the number of agreements were $M = 6.74$ $(SD = 0.87)$ in the expert group and $M = 4.34$ $(SD = 1.39)$ in the novice group. Note that a complete agreement with the ITS would have meant a score of eight points.

TABLE 1
An Example of Two of the Items Used in Experiment 1

Type of situation	Problem statement	Incomplete solution
Single analytical reading	**Car problem**. We have bought a car and already paid 12 installments of \$400 each. Knowing that the car costs \$12400, how much do we have left to pay?	$12400/400 = 31$
Multiple analytical reading	**Rabbits problem**. A farmer keeps 180 rabbits on two farms. One farm has 30 more rabbits than the other. How many rabbits are there in each farm?	$180/2 = 90$ $180 - 30 = 150$

The first problem includes a single operation from one analytical reading. The second contains operations from two different analytical readings.

To judge whether the difference in agreements between the expert and the novice group were significant, a Mann-Whitney's U test was performed. The selection of a non-parametric test was justified by the non-normal distribution of the variable number of agreements in the expert group. This was a consequence of the little variation of scores. The Mann-Whitney's U test showed significant differences ($U = 85.00$, $p < 0.001$, $r = 0.64$) between the mean ranks of concordances in the expert ($Mdn = 7.00$) and novice group ($Mdn = 4.00$).

In the particular case of the solution fragment $180/2 = 90$ and $180 - 30 = 150$ in the problem Rabbits, which is isomorphic to the problem Oranges, 84.21 percent of the experts provided the same suggestion as the ITS ($150/2$). Meanwhile, 15.79 percent gave another different but also valid help (namely, $30/2$). Within the novice group, 46.43 percent provided $150/2$; 23.21 percent, $30/2$; and 30.36 percent gave an incorrect hint (the most common mistakes were $90 - 30$ and $90 + 30$).

All these results allow us to state that there are significant differences between the help provided by expert and novice teachers; and also that the ITS behaves similar to human experts when providing suggestions.

5 Empirical Study 2: Evaluation of the ITS as a Learning Tool

5.1 Participants, Material and Procedure

This second experimental study aims to test the hypotheses that the system helps students solve problems that they had not been able to solve previously with paper and pencil; and that the ITS contributes to improving the students' capacity of solving word problems in an arithmetical way.

Participants were two groups of students in their fifth year of primary education (ages from 10 to 11) at a Spanish state school. Randomly, a group was identified as the control group (22 students) and the other one as the experimental group (20 students). Individuals who had previously been diagnosed a mathematics level two years behind the standard, and others who did not attend some of the sessions, were excluded from the analysis.

TABLE 2
Experts versus Novices

	Incorrect	Correct	
		Disagreements	Agreements
Expert Group ($n = 19$)	0.00%	15.79%	84.21%
Novice Group ($n = 56$)	17.86%	27.90%	54.24%

Second column shows the percentage of incorrect suggestions. Third and fourth columns show the percentage of disagreements and agreements with the aids provided by the ITS.

This finally resulted in 19 students in the control group and 17 in the experimental group.

In order to ensure a controlled environment, the experiment was carried out in the school premises during six 1-hour sessions, in the last term of the academic year. During the first session, a pre-test composed of 10 word problems was handed out to the students. The problems had a similar level to the ones in the Maths textbook used in class. The students had to solve them by using paper and pencil. Each participant was assigned a score in the pre-test according to the number of problems that they had been able to solve correctly. To dissociate problem solving abilities from calculation skills, students were always allowed to use a calculator.

The four central sessions (second to fifth) were organized as a differentiating training stage, to determine the potential of the ITS as a solving tool. During the second and third sessions, students in both groups had to solve a collection of 22 arithmetical word problems (11 per session) by only using paper, pencil and a calculator. The problems were extracted from other textbooks at the same educational level. For example, the problems in Table 1 were part of this collection. No teaching assistance was given to any of the students during this stage, recreating a similar environment as when students do their tasks at home and have no human help. Each student was assigned a score according to the number of problems that he/she solved correctly (first attempt score). In this way, the problems that a student was not able to solve were identified. At the beginning of the fourth session, two problems which had not been used so far were employed to demonstrate the operation of the ITS. This demonstration did not involve any student interaction with the system. To avoid the potential bias due to the students in the experimental group being trained with two more problems than the students in the control group, this was done in the presence of the students in both groups. Next, the students were handed a copy of each problem that they were not able to solve correctly during the second and third sessions. These copies included the incorrect solutions that they had previously submitted, and the correct answers expressed as a numerical value. In this way, the support offered was similar to that of a textbook with an answer key. Students in the control group were asked to re-attempt the problems by using paper, pencil and a calculator. Students in the experimental group retried the problems by using the ITS. To this end, the collection of problems employed in the previous two sessions was loaded on the ITS and the students selected the problems that they had solved incorrectly. Apart from processing the student's input and providing hints on demand in the way described in Section 3.2, the ITS gathered information about the problems selected, and whether or not they were solved

TABLE 3
Differences in Performance Obtained by Using the ITS in the Treatment Stage

	1st attempt $M(SD)$	2nd attempt $M(SD)$	Accumulated $M(SD)$
Control Group ($n = 19$)	11.37 (5.58)	1.53 (1.35)	12.89 (6.23)
Experimental Group ($n = 17$)	10.41 (5.66)	7.65 (3.86)	18.06 (3.15)

correctly. All students, both in the control and the experimental group, were assigned a score according to the number of problems that they were able to solve correctly during the fourth and fifth sessions (second attempt score).

At the sixth (last) session, a post-test composed of ten problems was handed out to the students. One to one, the problems in the pre- and post-tests had the same mathematical structure: i.e., the same number of isomorphic analytical readings. The instructions and conditions were the same as in the pre-test. Each participant was assigned a score in the post-test reflecting the number of problems solved correctly.

5.2 Analysis of Results

5.2.1 Potential of the Aids Offered by the ITS

To test the first hypothesis above and determine whether the ITS allows the students to solve problems that they were not able to solve on their own by just using paper and pencil, we compare the number of problems that the students solved in the four central sessions during the training stage.

As the value of the second attempt score could be a consequence of the number of problems that a student was not able to solve in his/her first attempt, the variable accumulated score was used instead. This was defined as the total number of problems that the student was able to solve after his/her first and second attempts. First attempt scores were previous to the training stage, and play the role of the pre-test in the previous analysis. Table 3 shows that the average score for the first attempt score was higher in the control group. On the other hand, the average accumulated score was substantially higher in the experimental group. A Kolmogorov-Smirnov test with a significance level of 0.05 did not ensure normality for the variables. Hence, non-parametric tests were used. A Mann-Whitney's U test did not yield significant differences ($U = 136.50$, $p = 0.426$) between the mean ranks of first attempt scores in the control ($Mdn = 12.00$) and the experimental ($Mdn = 11.00$) groups. On the contrary, the application of this test on the accumulated score showed significant differences ($U = 76.00$, $p = 0.007$, $r = 0.453$) between the mean ranks of scores in the control ($Mdn = 13.00$) and the experimental ($Mdn = 18.00$) groups. A global consideration of the data shows that students in the control group were able to solve 14.36 percent of the problems that they had not been able to solve in their first attempt. This percentage goes up to 65.99 percent in the experimental group. These results support the hypothesis that the ITS helps students solve problems that they had not been able to solve previously with paper and pencil.

5.2.2 Effect on the Arithmetical Problem Solving Competence of the Students

The comparison of the pre- and post-test scores obtained by the control and the experimental groups in Experiment 2 aims to determine the effect of using the ITS in the variation of the competence at solving problems in an arithmetical way. Table 4 shows the means and standard deviations of the pre- and post-test scores for both the experimental and control groups. An ANCOVA on the post-test scores with the pre-test as covariate was carried out. A significance level of 0.05 was adopted. We checked normality and homoscedasticity. A preliminary analysis evaluating the homogeneity-of-regression assumption indicated that the relationship between pre-test and post-test scores did not differ significantly as a function of the group, $F(1, 32) = 1.062$, $p = 0.310$. The ANCOVA revealed a significant difference in post-test scores between both groups after controlling the effect of pre-test scores, $F(1, 33) = 4.373$, $p = 0.044$, $\eta^2 = 0.117$. The finding that the experimental condition shows significantly higher means at the post-test after controlling for pretest is in line with the hypothesis that the ITS has a positive effect in the competence in solving word problems when the students come back to paper and pencil.

6 Conclusions and Further Work

In this paper, the idea of an analytical reading has been exploited to build an ITS that supports multiple solution paths and is able to provide adaptive help messages. The ITS has been designed around the main actions that an expert human tutor would do when supervising a student at solving an arithmetical problem. In this case, the system attempts to simulate the behavior of a teacher that offers remediation according to the user's intentions, i.e., a non-interventionist teacher that tries to support the solution scheme that the student has in mind despite that it may not be the optimum. This is done by inferring the current reasoning that the student is following, and generating help and feedback messages in consequence. The system is able to communicate by using natural language. The messages generated try to trigger the analytical process that leads to a feasible next action. To do this, they include verbal descriptions of the quantities, in an attempt to activate particular conceptual schemes.

TABLE 4
Differences between Groups in the Scores Before and After the Treatment Stage

	Pre-test $M(SD)$	Post-test $M(SD)$
Control Group ($n = 19$)	4.79 (2.27)	5.32 (2.67)
Experimental Group ($n = 17$)	5.24 (2.22)	6.65 (2.26)

The ability of the system at providing human-like expert support has been assessed through an experimental study. In addition, learning benefits of the ITS in the students' competence at solving problems in an arithmetical way have been empirically validated in a real educational setting. Results of this study suggest a significant improvement on students who have used the ITS, and also show that the ITS allows students to solve problems that they were not able to solve on their own by just using paper and pencil.

Effective help seeking is associated with better learning in educational technologies [23]. The ITS is able to build a user model by using the previous learner's interactions with the system (see [20] for implementation details). However, the ITS is not able to adapt the supervision style (intrusive or non-intrusive) to the student's characteristics. At present, we are working on strategies that support more flexible adaptive tutoring decisions, by taking into account the student model. We are also trying to expand the type of feedback and hints provided, incorporating messages with metacognitive purposes. A potential improvement of the ITS would consist in processing the student model in a better manner to determine whether giving a hint at a particular moment is appropriate. For example, the ITS could determine that a student does not need detailed help regarding a particular conceptual scheme if he/she commonly applies it in a correct way.

Another system limitation that is worth mentioning is related to the type of data used to build the user model. One-to-one interactions provide other sources of information that human tutors generally use at convenience to improve learning. For example, a human tutor may identify frustration situations by observing facial expressions or body language. Recent advances in the field of affective computing suggest that it is possible to detect some affective and/or mental states by using non-intrusive sources of information, e.g., video [24]. This kind of data could also be modeled and used by the recommendation engine to adapt help messages to specific user needs.

Currently, potential solution schemes for a given word problem need to be specified by the expert. This is a key issue because the system would only able to provide guidance according to the analytical readings that the expert has provided. Missing entries may result in misleading hints or misinterpretations of correct learner's steps as invalid actions. We are working on a system that computes all potential solution schemes from the problem quantities and the existing relations among them. Such a system would facilitate the production of problem collections, and help reduce development time further.

Acknowledgments

To the memory of Dr. Fernando Cerdán Pérez. He may have found in these lines an answer to some of his questions, but he would also certainly have found a lot of his ideas. The authors acknowledge the useful comments and suggestions received from Dr. Marta Molina, University of Granada, Spain. This work was partly supported by the Spanish Ministry of Economy and Competitiveness through projects EDU2012-35638 and TIN2011-29221-C03-02; by the Vicerrectorado de Convergencia Europea y Calidad of the University of Valencia, through projects DocenTIC UV-SFPIE_-DOCE13-147430 and Finestra Oberta UV-SFPIE FO12-80215; and by the Vicerrectorado de Investigación of the University of Valencia through project UV-INV-PRE-COMP12-80109.

References

[1] M. Waalkens, V. Aleven, and N. Taatgen, "Does supporting multiple student strategies lead to greater learning and motivation? Investigating a source of complexity in the architecture of intelligent tutoring systems," *Comput. Educ.*, vol. 60, no. 1, pp. 159–171, 2013.

[2] K. Reusser, "Tutoring systems and pedagogical theory: Representational tools for understanding, planning, and reflection in problem solving," in *Computers as Cognitive Tools*, S. P. Lajoie and S. J. Derry, eds., Hillsdale, NJ, USA: Lawrence Erlbaum, 1993 pp. 143–177.

[3] S. P. Marshall, *Schemas in Problem Solving*. New York, NY, USA: Cambridge Univ. Press, 1995.

[4] C.-K. Looi and B. Tan, "WORDMATH: A computer-based environment for learning word problem solving," in *Proc. 3rd Int. Conf. Comput. Aided Learn. Instruction Sci. Eng.*, 1996, vol. 1108, pp. 78–86.

[5] K. E. Chang, Y. T. Sung, and S. F. Lin, "Computer-assisted learning for mathematical problem solving," *Comput. Educ.*, vol. 46, no. 2, pp. 140–151, 2006.

[6] C. Beal, I. Arroyo, P. Cohen, and B. Woolf, "Evaluation of AnimalWatch: An intelligent tutoring system for arithmetic and fractions," *J. Interactive Online Learn.*, vol. 9, no. 1, pp. 64–77, 2010.

[7] R. Murray and K. VanLehn, "DT tutor: A decision-theoretic, dynamic approach for optimal selection of tutorial actions," in *Proc. 5th Int. Conf. Intell. Tutoring Syst.*, 2000, vol. 1839, pp. 153–162.

[8] K. VanLehn, S. Siler, C. Murray, T. Yamauchi, and W. B. Baggett, "Why do only some events cause learning during human tutoring?" *Cognition Instruction*, vol. 21, pp. 209–249, 2003.

[9] D. Fossati, B. Di Eugenio, C. Brown, S. Ohlsson, D. Cosejo, and L. Chen, "Supporting computer science curriculum: Exploring and learning linked lists with iList," *IEEE Trans. Learn. Technol.*, vol. 2, no. 2, pp. 107–120, Apr.–Jun. 2009.

[10] Z. I. Kalmykova, "Processes of analysis and synthesis in the solution of arithmetic problems," in *Soviet Studies in the Psychology of Learning and Teaching Mathematics, Vol. XI: Analysis and Synthesis as Problem Solving Methods*, J. Kilpatrick, I. Wirszup, E. G. Begle, J. W. Wilson, and M. G. Kantowski, eds., Stanford, CA, USA: School Math. Study Group Stanford Univ. Survey Recent East Eur. Math. Literature, 1975 pp. 1–171.

[11] A. N. Bogolyubov, "A combined analytic and synthetic method of solving arithmetic problems in elementary school," in *Soviet Studies in the Psychology of Learning and Teaching Mathematics, Vol. VI: Instruction in Problem Solving*, J. Kilpatrick and I. Wirszup, eds., Stanford, CA, School Math. Study Group Stanford Univ. Survey Recent East Eur. Math. Literature, 1972 pp. 61–94.

[12] M. S. Riley, J. G. Greeno, and J. L. Heller, "Development of children's problem-solving ability in arithmetic," in *The Development of Mathematical Thinking*, H. P. Ginsburg, ed., New York, NY, USA: Academic, 1983 pp. 153–196.

[13] M. J. Nathan, W. Kintsch, and E. Young, "A theory of algebra-word-problem comprehension and its implications for the design of learning environments," *Cognition Instruction*, vol. 9, no. 4, pp. 329–389, 1992.

[14] K. R. Koedinger and M. J. Nathan, "The real story behind story problems: Effects of representations on quantitative reasoning," *J. Learn. Sci.*, vol. 13, no. 2, pp. 129–164, 2004.

[15] L. Puig, "Researching (algebraic) problem solving from the perspective of local theoretical models," *Procedia - Soc. Behav. Sci.*, vol. 8, pp. 3–16, 2010.

[16] F. Cerdán, *Estudios sobre la Familia de Problemas Aritmético-Algebraicos*. Valencia, Spain: Servicio de Publicaciones de la Universidad de Valencia, 2008.

[17] E. Filloy, T. Rojano, and L. Puig, *Educational Algebra: A Theoretical and Empirical Approach*. New York, NY, USA: Springer, 2008.

[18] D. Arnau, M. Arevalillo-Herráez, L. Puig, and J. A. González-Calero, "Fundamentals of the design and the operation of an intelligent tutoring system for the learning of the arithmetical and algebraic way of solving word problems," *Comput. Educ.*, vol. 63, pp. 119–130, 2013.
[19] V. Kodaganallur, R. R. Weitz, and D. Rosenthal, "A comparison of model-tracing and constraint-based intelligent tutoring paradigms," *Int. J. Artif. Intell. Educ.*, vol. 15, no. 2, pp. 117–144, 2005.
[20] M. Arevalillo-Herráez, D. Arnau, and L. Marco-Giménez, "Domain-specific knowledge representation and inference engine for an intelligent tutoring system," *Knowledge-Based Syst.*, vol. 49, pp. 97–105, 2013.
[21] D. C. Berliner, "Learning about and learning from expert teachers," *Int. J. Educ. Res.*, vol. 35, no. 5, pp. 463–482, 2001.
[22] Y. Li and G. Kaiser, "Expertise in mathematics instruction: Advancing research and practice from an international perspective," in *Expertise in Mathematics Instruction*, Y. Li and G. Kaiser, eds., New York, NY, USA: Springer, 2011 pp. 3–15.
[23] V. Aleven, B. McLaren, I. Roll, and K. Koedinger, "Toward metacognitive tutoring: A model of help seeking with a cognitive tutor," *Int. J. Artif. Intell. Educ.*, vol. 16, no. 2, pp. 101–128, 2006.
[24] R. El Kaliouby and P. Robinson, "Mind reading machines: Automated inference of cognitive mental states from video," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, 2004, vol. 1, pp. 682–688.

David Arnau received the degree in physics from the University of Valencia, Spain. In 2010, he received the PhD degree in educational mathematics from the University of Valencia. He taught mathematics and physics at secondary level for 13 years. He is currently a lecturer in the Department of Didactics of Mathematics at the University of Valencia. He is a member of the board of the Spanish Society of Research in Mathematics Education (SEIEM). His research interest include educational algebra and the development of interactive learning environments for the teaching and learning of word problem solving.

Miguel Arevalillo-Herráez received the first degree in computing from the Technical University of Valencia, Spain, in 1993, the BSc degree in computing in 1994, the PgCert in teaching and learning in higher education, and the PhD degree in 1997, all from Liverpool John Moores University, United Kingdom. On the same year, he gained accreditation as a teacher in HE by the Staff and Educational Development Association (SEDA) and became a senior lecturer at Liverpool John Moores University. In 1999, he left to work for private industry for a one year period, and came back to academy in 2000. He was the program leader for the computing and business degrees at the Mediterranean University of Science and Technology until 2006. He is currently working as a lecturer in the University of Valencia, Spain. His research interests include applied artificial intelligence and pattern recognition.

José Antonio González-Calero received the degree in industrial engineering from the University of Castilla-La Mancha, in 2005. He received the master's degree in secondary school teacher from the University of Castilla-La Mancha and also the master's degree in research in didactic of mathematics from the University of Valencia in 2010 and 2011, respectively. Since 2010, he has been a lecturer in the Department of Mathematics at the University of Castilla-La Mancha. He is currently working toward the PhD degree in educational mathematics at the University of Valencia. He worked in developing accounting systems for financial institutions until 2009. His research interests include educational algebra, word problem solving, and interactive learning environments. He is a member of the Spanish Society of Research in Mathematics Education (SEIEM).