International Journal of Artificial Intelligence in Education 22 (2013) 191–218

DOI 10.3233/JAI-130036
IOS Press

MATHESIS: An Intelligent Web-Based Algebra Tutoring School

Dimitrios Sklavakis, *Department of Applied Informatics, University of Macedonia, Egnatia, Thessaloniki, Greece*
dsklavakis@uom.gr

Ioannis Refanidis, *Department of Applied Informatics, University of Macedonia, Egnatia, Thessaloniki, Greece*
yrefanid@uom.gr

Abstract. This article describes an intelligent, integrated, web-based school for tutoring expansion and factoring of algebraic expressions. It provides full support for the management of the usual teaching tasks in a traditional school: Student and teacher registration, creation and management of classes and test papers, individualized assignment of exercises, intelligent step by step guidance in solving exercises, student interaction recording, skill mastery statistics and student assessment. The intelligence of the system lies in its Algebra Tutor, a model-tracing tutor developed within the MATHESIS project, that teaches a breadth of 16 top-level math skills (algebraic operations): monomial multiplication, division and power, monomial-polynomial and polynomial-polynomial multiplication, parentheses elimination, collect like terms, identities (square of sum and difference, product of sum by difference, cube of sum and difference), factoring (common factor, term grouping, identities, quadratic form). These skills are further decomposed in simpler ones giving a deep domain expertise model of 104 primitive skills. The tutor has two novel features: a) it exhibits *intelligent task recognition* by identifying all skills present in any expression through intelligent parsing, and b) for each identified skill, the tutor traces all the sub-skills, a feature we call *deep model tracing*. Furthermore, based on these features, the tutor achieves *broad knowledge monitoring* by recording student performance for all skills present in any expression. Forty teachers who evaluated the system in a 3-hours workshop appreciated the fine-grained step-by-step guidance of the student, the equally fine grained student model created by the tutor and its ability to tutor any exercise that contains the aforementioned math skills. The system was also used in a real junior high school classroom with 20 students for three months. Evaluation of the students' performance in the domain of factoring gave positive learning results.

Keywords. Intelligent tutoring systems, model-tracing tutors, web-based learning

INTRODUCTION

One-to-one tutoring has proven to be one of the most effective ways of teaching. It has been shown (Bloom 1984) that the performance of the average student under an expert tutor is about two standard deviations above the average performance of the conventional class (30 students to one teacher). That is, 50% of the tutored students scored higher than 98% of students in the conventional class. However, it is also known that one-to-one tutoring is a very expensive form of education. Due to this cost, we are still in the era of mass education, struggling to raise the teacher to student ratio. The problem of designing and implementing educational environments as effective as individual tutoring was termed by Bloom as "the two sigma problem", named after the mathematical symbol of standard deviation, σ.

The implementation of the one-to-one tutoring model by Intelligent Tutoring Systems (ITSs) has motivated researchers to aim to develop ITSs that provide the same tutoring quality as a human tutor

(VanLehn, 2006). Model Tracing Tutors (MTTs) (Anderson, Corbett, Koedinger, & Pelletier, 1995) have shown significant success in domains like mathematics (Koedinger & Corbett, 2006), computer programming (Corbett, 2001) and physics (VanLehn, Lynch, Schulze, Shapiro, & Shelby, 2005). These tutors are based on a *domain expertise model* that solves the problem under tutoring and produces the correct step(s). At each step, the *model-tracing algorithm* matches the solution(s) produced by the model to that provided by the student and gives positive or negative feedback, hints or/and help messages. However, the domain models of MTTs are hard to author (Aleven, McLaren, Sewall, & Koedinger, 2006). The main reason for this is the knowledge acquisition bottleneck: extracting the knowledge from the domain experts and encoding it into a MTT. Knowledge reuse has been proposed as a key factor to overcome this obstacle (Murray, 2003a; Mizoguchi & Bourdeau, 2000). Since expert knowledge and, particularly, tutoring knowledge is so hard to create, re-using it is of paramount importance. A good example of knowledge reuse is the Mass Production mechanism provided by Carnegie Mellon's Cognitive Tutors Authoring Tools (CTAT). This mechanism allows the creation of new tutors from existing ones for isomorphic problems, that is problems having nearly the same solution steps (Aleven, McLaren, & Sewall, 2009).

The main goal of the ongoing MATHESIS project is to develop authoring tools for model-tracing tutors in mathematics, with knowledge re-use as the primary characteristic of the authored tutors as well as for the authoring knowledge used by the tools. For this reason, in the first stage of the MATHESIS project, an Algebra Tutor was developed to be used as a prototype target tutor (Sklavakis & Refanidis, 2008). The purpose of developing the tutor was twofold: a) to investigate the design and implementation effort for developing an MTT having a domain expertise model with a breadth of 16 top level skills (algebraic operations) and – after elaborate cognitive task analysis – a greater depth and b) to provide the knowledge that would be represented in an ontology on top of which the authoring tools would be implemented (Sklavakis & Refanidis, 2010).

Concerning the former research goal, as the domain expertise model has been extended and deepened, the *scaling-up* problem was confronted: if a problem contains more than one task to be performed then a more complex task arises, i.e., identifying the tasks to perform! The solution to this tutoring problem was to equip the tutor with *intelligent task recognition* through sophisticated parsing of the algebraic expressions. Another, rather positive, consequence of adopting a broad and deep domain expertise model was the development of an equally detailed student model. Instead of simply keeping a percentage measure of the students' skill performance, the student model was extended to keep full records of the interactions between the interface and the student for each solution step.

This article describes the web-based intelligent tutoring school for expanding and factoring algebraic expressions. The school has been built around the MATHESIS algebra MTT and has been extended with a learning management system (LMS). The rest of the article is structured as follows: Section 2 describes the latest version of the MATHESIS algebra MTT with an extended domain model, a refined student model and a new interface integrating the tutor into the school. Section 3 describes the learning management system of the school, including an editor for teachers to create test papers with their own exercises and tools to inspect the student model. Section 4 presents related work. Section 5 presents an evaluation of the system while Section 6 concludes the article with a discussion of the research results and future directions of research.

THE MATHESIS ALGEBRA TUTOR[1]

The MATHESIS Model-Tracing Algebra Tutor was developed as a prototype target tutor for the MATHESIS project (Sklavakis & Refanidis, 2010). The ultimate goal of the project is the development of authoring

[1]http://users.sch.gr/dsklavakis/mathesis/en/MATHESIS_Main_Frameset.htm

tools for model-tracing tutors that will make extensive reuse of the valuable tutoring knowledge through ontological engineering. The MATHESIS tutor itself was designed with knowledge reuse as its main non-functional requirements. Consequently, the architecture of the system should be based on open, standardized and modular representations. Additionally, there were three more issues that determined the overall architecture:

a) The tutor interface should be web-based in order to be broadly accessible.
b) The model-tracing algorithm requires constant interaction between the cognitive model and the interface. Therefore they should lie at the same side, that is, the client side.
c) The programming language(s) that would implement the various tutor parts (interface, domain model) should be simple enough to be represented with an ontology. This ontology would be used by the authoring tools to guide non-expert authors in redeveloping the tutor.

The achievement of these requirements led us to implement the tutor using HTML for the user interface and JavaScript for the domain expertise and tutoring models. These two languages are the simplest ones for building dynamic, interactive web pages, they are open, non-proprietary and lend themselves to direct representation and manipulation from the developed MATHESIS authoring tools (Sklavakis & Refanidis, 2011). The user interface, shown in Fig. 1, has four main parts:

- The *messages area* (top), where the tutor displays information about the interface usage, as well as hints, help and feedback for correct and incorrect problem-solving steps.
- The *algebraic expression rewriting area* (a), where the algebraic expression under rewriting and its transformations are displayed.
- The *student's answering area* (b), where the student enters the answer for each problem-solving step.
- The *performed operation area* (c), where intermediate results are shown for multi-step algebraic operations.

The primary interface element is Design Science's WebEq (now MathFlow) Input Control applet, an editor for displaying and editing mathematical expressions in web pages (Design Science, 2011). There are three such Input Controls, i.e., the algebraic expression, the answering space and the performed operation Input Controls (Fig. 1). The WebEq Input Control is scriptable through JavaScript and represents algebraic expressions as MathML[2]. So, during the problem solving process, the problem-solving state as well as the student solution steps are represented via the open MathML standard and, therefore, they can be *interoperatable*, i.e. *inspectable*, *recordable* and *scriptable* (Murray, 2003b). As a result, the tutor can be used in the following ways:

a) The student can type directly in the algebraic expression area algebraic expressions using the math editing palette (Fig. 1, area (a)). Then, he/she can initialize the tutoring process by clicking the "Start Exercise" button.
b) The student can select an exercise from a test paper created by a teacher through the Learning Management System (Section 3) and then initialize the tutor.
c) The tutor can be initialized (opened) from any other e-learning program with the desired algebraic expression.
d) The tutor can recursively initialize (open) new instances of itself in order to break down more complex tutoring tasks.

This latter possibility is directly related to the issues of knowledge re-use and "scaling-up". The mathematical skill of *factoring by term grouping* is rather complex. In this factoring method (a) the terms of

[2]http://www.w3.org/Math/

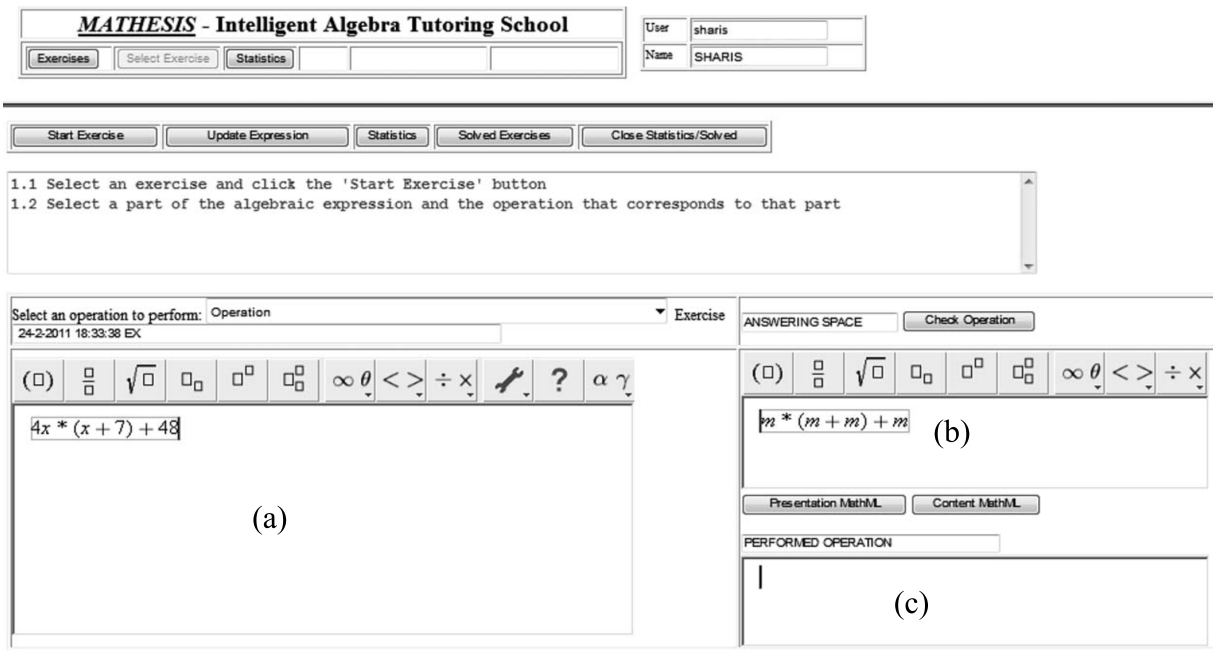

Fig. 1. The MATHESIS algebra tutor interface.

the expression must be separated into groups, (b) each group must be factored by some factoring method and (c) the resulting products must have a common factor. It is step (c) that makes step (a) and the whole method complex and raises the issues of knowledge re-use and "scaling-up". The intelligent task recognition of the MATHESIS tutor does not yet support guidance for the first step and therefore term grouping is not yet part of its domain model. However, provision has been made for steps (b) and (c). As an example, let's consider factoring the expression $x^4 - 1 + x^3 - x$ by grouping its terms: the first group, $x^4 - 1$, must be factored using the *identity* $a^2 - b^2 = (a + b)\ (a - b)$, yielding $(x^2 + 1)\ (x^2 - 1)$; the second group, $x^3 - x$, must be factored by *common factor*, yielding $x\ (x^2 - 1)$. To guide the student in applying different factoring methods, the tutor can open an instance of itself with the expression $x^4 - 1$ for the first group followed by an instance for expression $x^3 - x$. Each instance of the tutor can guide the student in factoring each group as separate problems and then return the factored expression to the parent tutor, thus yielding $(x^2 + 1)\ (x^2 - 1) + x\ (x^2 - 1)$. From this point, the parent tutor will guide the student in applying the common factor method, yielding $(x^2 - 1)\ (x^2 + 1 + x)$. Thus, the factoring methods supported by the tutor can be re-used in a completely new and complex factoring task, term grouping.

The tutor's domain expertise model

The development of the domain expertise model was based on deep cognitive task analysis in the paradigm of Carnegie-Mellon's cognitive tutors (Anderson et al., 1995). The tutor can teach a breadth of 16 top-level cognitive math skills:

- Monomial multiplication
- Monomial division
- Powers of monomials
- Monomial-polynomial multiplication

- Polynomial multiplication
- Elimination of parentheses
- Collection of like terms
- Identities expansion: square of sum, square of difference, product of sum by difference, cube of sum and cube of difference
- Factoring: common factor, identities, quadratic form

Each one of these top-level math skills is further analyzed in more detailed sub-skills leading to a fine grained domain model of 104 primitive math skills. Part of this broad and deep domain model is given in the following list:

1. *Monomial multiplication:* $3\mathrm{x}^2\mathrm{y}\cdot(-4\mathrm{xz}^3) = -12\mathrm{x}^3\mathrm{yz}^3$
   1.1 Multiply coefficients: $3\cdot(-4) = -12$
   1.2 Multiply main parts:
      1.2.1 Add exponents of common variables: $x^2\cdot x = x^{2+1} = x^3$
      1.2.2 Copy exponents of single variables: $y\cdot z^3 = yz^3$
2. *Monomial division:* $\frac{-12\mathrm{x}^3\mathrm{yz}^3}{3\mathrm{x}^2\mathrm{y}} = -4\mathrm{xz}^3$
   2.1 Divide coefficients: $-12{:}3 = -4$
   2.2 Divide main parts:
      2.2.1 Subtract exponents of common variables: $x^3{:}x^2 = x^{3-2} = x$ *and* $y{:}y = y^{1-1} = y^0 = 1$
      2.2.2 Copy exponents of single variables: $z^3 = z^3$
3. *Collection of like terms:* $2xy - x^2 + 7xy + 6x^2 = 9xy + 5x^2$
   3.1 Find groups of identical terms: $2xy + 7xy$ *and* $-x^2 + 6x^2$
   3.2 Add the coefficients of each group: $2 + 7 = 9$ *and* $-1 + 6 = 5$
   3.3 Keep the main part of each group: $2xy + 7xy = 9xy$ *and* $-x^2 + 6x^2 = 5x^2$
4. *Monomial power:* $(-2x^2yz^3)^3 = -8x^6y^3z^9$
   4.1 Raise the coefficient to the power: $(-2)^3 = -8$
   4.2 Raise main part to the exponent:
      4.2.1 Multiply the exponents: $(\mathrm{x}^2\mathrm{yz}^3)^3 = \mathrm{x}^{2\cdot3}\mathrm{y}^{1\cdot3}\mathrm{z}^{3\cdot3} = \mathrm{x}^6\mathrm{y}^3\mathrm{z}^9$
5. *Monomial by polynomial multiplication:* $3x^3y\cdot(2x - y^2z) = 6x^4y - 3x^3y^3z$
   5.1 Identify the monomial terms of the polynomial: $2\mathrm{x}$ and $-\mathrm{y}^2\mathrm{z}$
   5.2 Multiply each one of them with the monomial: $3x^3y\cdot 2x = 6x^4y$ *and* $3x^3y\cdot(-y^2z) = -3x^3y^3z$
6. *Polynomial by polynomial multiplication*:
   $(3xy - 2x^2)\cdot(2x^2y^2 - 4xy) = 6x^3y^3 - 12x^2y^2 - 4x^4y^2 + 8x^3y$
   6.1 Identify the monomial terms of the first polynomial: $3\mathrm{xy}$ and $-2\mathrm{x}^2$
   6.2 Identify the monomial terms of the second polynomial: $2x^2y^2$ *and* $-4xy$
   6.3 Multiply each term of the first monomial with each term of the second monomial: $3xy\cdot 2x^2y^2 = 6x^3y^3$ *and* $3xy\cdot(-4xy) = -12x^2y^2$ *and* $-2x^2\cdot 2x^2y^2 = -4x^4y^2$ *and* $-2x^2\cdot(-4xy) = 8x^3y$
7. *Elimination of parentheses*:
   $(5x^2 - 8xy + 3) - (x^2 + 4xy - 5) = 5x^2 - 8xy + 3 - x^2 - 4xy + 5 = 4x^2 - 12xy + 8$
   7.1 Keep the sign of each parenthesized term if the sign in front of the parenthesis is a plus (+): $(5x^2 - 8\mathrm{xy} + 3) = 5\mathrm{x}^2 - 8\mathrm{xy} + 3$
   7.2 Change the sign of each parenthesized term if the sign in front of the parenthesis is a minus (−): $-(x^2 + 4xy - 5) = -x^2 - 4xy + 5$
   7.3 Collect like terms if there are any: $5x^2 - 8xy + 3 - x^2 - 4xy + 5 = 4x^2 - 12xy + 8$

Table 1
Expanding $-10\,(x-1)\,(x+1)$ in three different ways

Operation	Result
A1. Monomial-polynomial multiplication	$-10\,(x-1)\,(x+1) =$
A2. Polynomial multiplication	$(-10x+10)\,(x+1) =$
A3. Collection of like terms	$-10x^2 - 10x + 10x + 10 =$
	$-10x^2 + 10$
B1. Polynomial multiplication	$-10\,(x-1)\,(x+1) =$
B2. Monomial-polynomial multiplication	$-10(x^2 + x - x - 1) =$
B3. Collection of like terms	$-10x^2 - 10x + 10x + 10 =$
	$-10x^2 + 10$
C1. Identity $(a+b)\,(a-b) = a^2 - b^2$	$-10\,(x-1)\,(x+1)$
C2. Monomial-polynomial multiplication	$-10\,(x^2-1)$
	$-10x^2 + 10$

8. *Identity expansion*: $(2x+3)^2 = 4x^2 = 12x + 9$
   8.1 Recall the expanded form of the identity: $(\mathrm{a} + \mathrm{b})^2 = \mathrm{a}^2 + 2\mathrm{ab} + \mathrm{b}^2$
   8.2 Substitute a and b for the real terms: a = 2x and b = 3
   8.3 Take care for parenthesized terms: $(2\mathrm{x} + 3)^2 = (2\mathrm{x})^2 + 2 \cdot 2\mathrm{x} \cdot 3 + 3^2$
   8.4 Perform monomial multiplications and powers: $(2\mathrm{x})^2 + 2 \cdot 2\mathrm{x} \cdot 3 + 3^2 = 4\mathrm{x}^2 + 12\mathrm{x} + 9$
9. *Factoring by common factor*: $2x^3 - 4x^2 + 6x^2y = 2x^2(x - 2 + 3y)$
   9.1 Find the common factor: $2\mathrm{x}^2$
      9.1.1 Find the GCD of the coefficients: GCD (2, 4, 6) = 2
      9.1.2 Find the GCD of common variables: GCD $(\mathrm{x}^3,\ \mathrm{x}^2,\ \mathrm{x}^2) = \mathrm{x}^2$
   9.2 Divide terms by the common factor: $\frac{2\mathrm{x}^3}{2\mathrm{x}^2} = \mathrm{x}$ and $\frac{-4\mathrm{x}^2}{2\mathrm{x}^2} = -2$ and $\frac{6\mathrm{x}^2\mathrm{y}}{2\mathrm{x}^2} = 3\mathrm{y}$
10. *Factoring the quadratic form* $x^2 + Sx + P = (x+a)\,(x+b),\ a \cdot b = P$ *and* $a + b = S$:
    $x^2 + 5x + 6 = (x+2)\,(x+3)$
    10.1 Identify $P = a \cdot b$ *and* $S = a + b$: $P = 6$ *and* $S = 5$
    10.2 Find the pairs of integers a, b that have a product of $p = 6\ :\ 1 \cdot 6 = 6$ *or* $(-1) \cdot (-6) = 6$ *or* $2 \cdot 3 = 6$ *or* $(-2) \cdot (-3) = 6$
    10.3 Find the pair a, b that gives a sum of $S = 5\ :\ 2 + 3 = 5$
    10.4 Write the factored form: $(x+a)\,(x+b) = (x+2)\,(x+3)$

The main reason for developing such a broad and deep domain expertise model was the investigation and confrontation of the *scaling-up problem*: despite the success of model-tracing tutors, in the majority of implementations, the tutor teaches a very elementary (low level) cognitive skill in isolation (Aleven, McLaren, & Sewall, 2009). However, even in school textbooks, medium difficulty exercises demand the application of a multitude of composite (top-level) cognitive skills in combination with each other. Their solutions demand the application of more high-level skills, like the identification and decomposition of the top level skills that appear in the exercise.

To illustrate this situation, consider the algebraic expression $(x-3)^2 - 10\,(x-1)\,(x+1)$. In order to expand this expression, the student must first identify the operations that must be performed: a square of difference $(x-3)^2$, and a multiplication with three factors $-10(x-1)\,(x+1)$. Especially for the multiplication, the student can perform it in three different ways, described in Table 1.

So, it becomes clear that, even for a simple expansion exercise like the one in Table 1, a broad and deep domain expertise model containing all the potential skills is needed. In addition, intelligent recognition of the operations that are present in the expression is needed, whereas this is also a new cognitive skill that the tutor must be able to teach.

Intelligent task recognition

The key issue for tackling the scaling-up problem is the recognition by the tutor of the task(s) that must be performed, as well as of those entered by the student in order to match them, so as to provide guidance and feedback in each step of the tutoring process. In the MATHESIS tutor, these problems are tackled by parsing the MathML representation of the algebraic expressions and generating multiple internal representations. To illustrate how this is done, the algebraic expression $4x\ (x + 7) + 48$ will be used:

1. The tutor gets a tree representation of the expression's MathML presentation, analogous to the Document Object Model (DOM) of HTML. This is provided by the Input Control applet (Fig. 1) through JavaScript scripting. In this MathML DOM tree, every element of the algebraic expression is represented as a node.
2. This MathML DOM tree is parsed using special methods provided by the Input Control. Each element of the expression (node) is given a unique identification string (id), which is used in the internal representations of the expression to uniquely identify each element. At the same time, the "atomic" elements such as numbers, variables and operation symbols are grouped in mathematical objects like monomials and polynomials. These are represented using custom JavaScript objects, and they also get unique identification strings. For each monomial, its coefficient, variables and their exponents are kept along with their unique identification numbers. For each polynomial, its monomial terms are kept. In the case of expression $4x\ (x + 7) + 48$, four monomials are created, $4x$, x, 7 and 48, as well as a polynomial, $x+7$, having as its terms the monomials x and 7.
3. Identifying each operations' precedence is a key top-level skill for the expansion and factoring of algebraic expressions. As we will explain in the next subsection, the tutor teaches students the correct order of operations. Consequently, the intelligent parsing mechanism extracts this information from the algebraic expression and represents it appropriately.
4. Finally, using the precedence of operations, the expression is represented as a sum of products using JavaScript arrays. The expression $4x\ (x + 7) + 48$ is represented as a sum array of two product arrays, $4x\ (x + 7)$ and 48. The first product array has two factors, monomial $4x$ and polynomial $(x + 7)$, while the second product array has only one monomial, 48.

All this information is extracted and represented for the expression to be rewritten. When the student selects a part (or the whole) of the expression, this part is parsed again and the same information is extracted and represented by the tutor; however, now the parser does not assign identification strings to the elements of the selected expression but just gets the ones assigned by the original parsing of the expression. As a result, the tutor can identify exactly which part of the expression is selected, which operations are selected and whether they have the right precedence to be performed. Moreover, when the student suggests what kind of operation he/she has selected, the tutor can check whether this suggestion is correct. For example, in expression $4x\ (x + 7) + 48$, if the student selects $4x\ (x + 7)$ and proposes "Common Factor", the tutor checks its internal representation and sees that the selected (sub) expression is not a sum and therefore it can't be factored. If the student selects the whole expression, the tutor sees that the expression is a sum with two terms and only then tries to extract a common factor. If it finds one, it proceeds by asking the student to give the common factor. Otherwise, the student is given feedback that no common factor exists.

Moreover, the tutor checks that the student has selected the whole expression, since there is no point in getting a common factor of part of an expression.

This approach, with exhaustive and multiple representations of the algebraic expressions allows the tutor to handle even more subtle conditions like dealing with the commutative properties of addition ($x + y = y + x$) and multiplication ($x \cdot y = y \cdot x$). In practice, the commutative property means that in a sum or product, the order of the terms is not important. By representing the algebraic expressions as a sum of products, the MATHESIS tutor can easily check student answers that are sums or products. Thus, when expanding the expression $(x + y^2)^2$, the tutor can accept as a correct answer any of the expressions $x^2 + 2xy^2 + y^4$, $y^4 + 2xy^2 + x^2$, $x^2 + 2y^2x + y^4$and $x^2 + y^4 + 2y^2x$. Moreover, it can detect if a term is missing or is wrong and give the appropriate feedback. This performance is achieved by JavaScript functions that compare the sum and product arrays.

The overall result of this intelligent parsing is that the tutor can handle any algebraic expression that contains the math tasks (operations) described in the previous section. Therefore, the student can type any such expression and the MATHESIS tutor will parse it, detect which tasks are contained in it and guide the student appropriately. We call this feature *intelligent task recognition.* It is this feature combined with the broad and deep domain model that deals directly with the scaling-up problem: the MATHESIS tutor can handle *any* algebraic expression containing *any* combination of the math tasks described in the previous section. Thus, the MATHESIS tutor can guide a student in expanding expressions like $(2x + 3)^2 - 2\ (2x + 3)\ (2x - 3) + (2x - 3)^2$ or factor expressions like $4\ (x - 1)^2 - 9\ (x - 2)^2$.

The tutoring model: Deep model tracing with intelligent task recognition

Equipped with such a detailed cognitive model, the MATHESIS tutor is able to exhibit expert human-like performance. The tutor makes all the cognitive tasks explicit to the student through the structure of the interface. The whole process is described below using as an example a real student interaction with the tutor for factoring the algebraic expression $4x * (x + 7) + 48$:

1. The student enters the algebraic expression in one of the ways described in Section 2.
2. The student starts the tutor by clicking "Start Exercise", the tutor analyses the expression and *recognizes* the operations and their operands. As a result, the tutor displays an abstract representation of the algebraic expression, where each monomial in the expression has been substituted by an "*m*". Thus, the algebraic expression $4x * (x + 7) + 48$ is represented as *m * (m+m)+m* (Fig. 1, Student Answering area). The purpose of this *intelligent task recognition* feature is to help the student understand the operations present in the expression through a visual, simplified and compact representation of the algebraic expression. We realized that the use of letter "*m*" for representing a monomial could confuse the students, since this letter is normally used in mathematics to represent a variable. To avoid any such misconception, pen and paper exercises were given to the students, before using the system, where they had to transform algebraic expressions to the tutor's "*m*" letter representation (this is a common practice followed by human tutors). After a few exercices, all students, even the weakest ones, were able to correctly perform this transformation. On the other hand, alternative representations were considered. For example, one of them was to use empty squares instead of "*m*"; however, it was abandoned as an option because a square symbol was used by the MATHESIS tutor to provide templates that guide student input (see step 4, below). Using a tree representation of the algebraic expression was also considered. However, in pen and paper exercises, where students were asked to transform between natural and tree representation, significant cognitive load and confusion were observed.

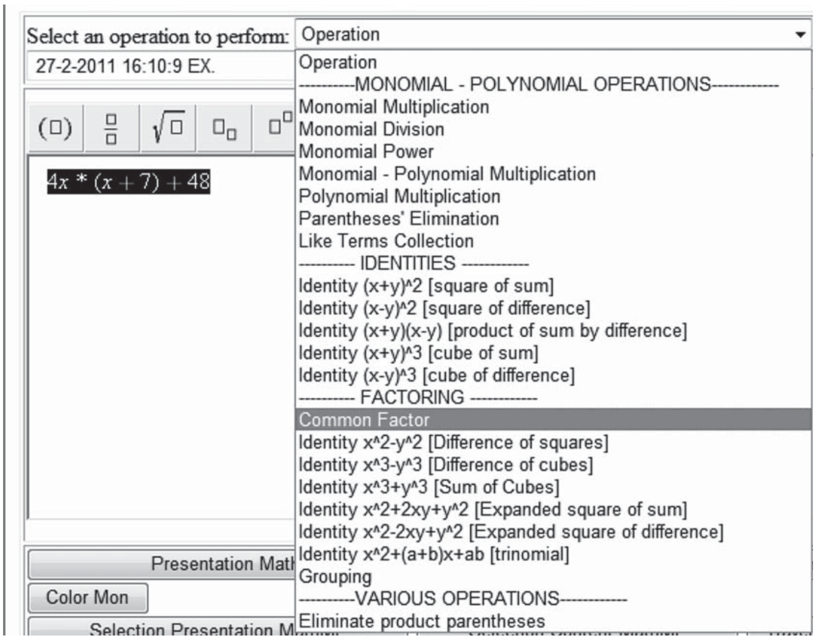

Fig. 2. The student proposes the operation "FACTORING-Common Factor" from the drop-down list of supported operations to be applied to the selected expression.

3. The student selects a part (or the whole) of the expression and then chooses from a drop-down list the operation that he/she believes corresponds to that part. In Fig. 2 the student selected the whole expression $4x * (x + 7) + 48$ (highlighted) and the operation "FACTORING – Common Factor" from the drop-down list. We must note that this tutoring step is not part of the "traditional" tutoring practice in the Greek educational system and, to the best of our knowledge, in many other educational systems. However, based on our personal tutoring experience, we believe that this step is crucial and constitutes what is known in expert systems as an *expert's blind spot*. Math teachers tend to believe that once students have been taught and practiced each operation separately, they are able to recognize and perform them when they appear in more complicated algebraic expressions. Our personal tutoring experience suggests that quite often students *don't know what to do* because they cannot recognize which operations are present and the human tutor has to guide them in analyzing the expression under consideration. It is this step, in combination with the abstract representation of the algebraic expression presented in the previous step, that makes the analysis of the algebraic expression explicit to the student.
4. The tutor, based on the results of the intelligent task recognition (step 2), confirms and continues or informs the student that the suggested operation is not correct. In Fig. 3, the suggested operation, "Common Factor", is correct; the tutor confirms that with an appropriate message and starts guiding the student to perform the operation in a step-by-step manner (Fig. 3, top, messages 2.1 and 2.2).

The tutor also knows that the common factor for the expression $4x * (x + 7) + 48$ is the greatest common divisor of 4 and 48, that is, 4. The authors' personal tutoring experience suggests that most students have considerable difficulties in finding the common factor. For this reason, the tutor displays in the student's answering area a visual scaffold of the common factor's form. Here, the common factor is only a number, denoted by a single square (Fig. 3, bottom right). The tutor also displays a message that explains the

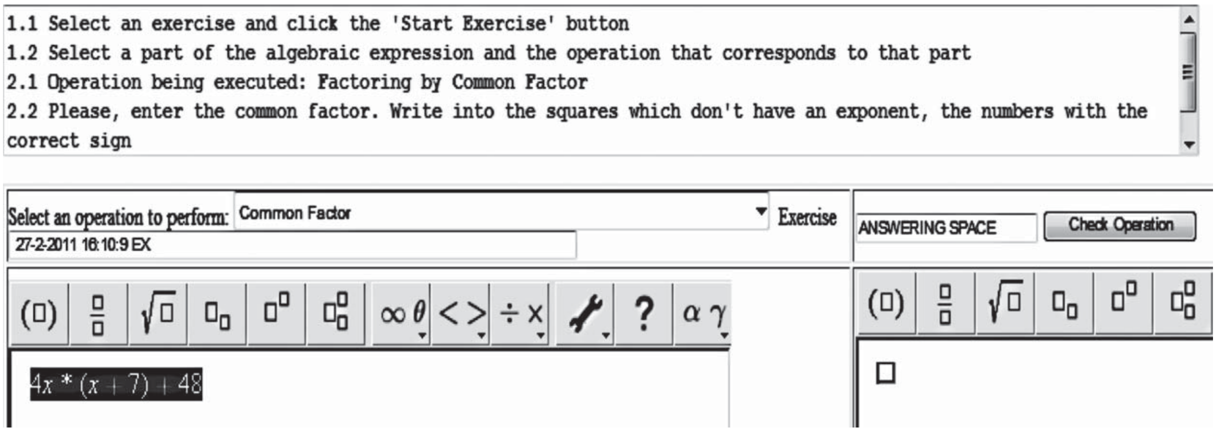

Fig. 3. The tutor checks and confirms the student's suggested operation "Common Factor" through messages 2.1 and 2.2 (top). The common factor under question here is 4, denoted by the empty square scaffold in the "ANSWERING SPACE" area (bottom right).

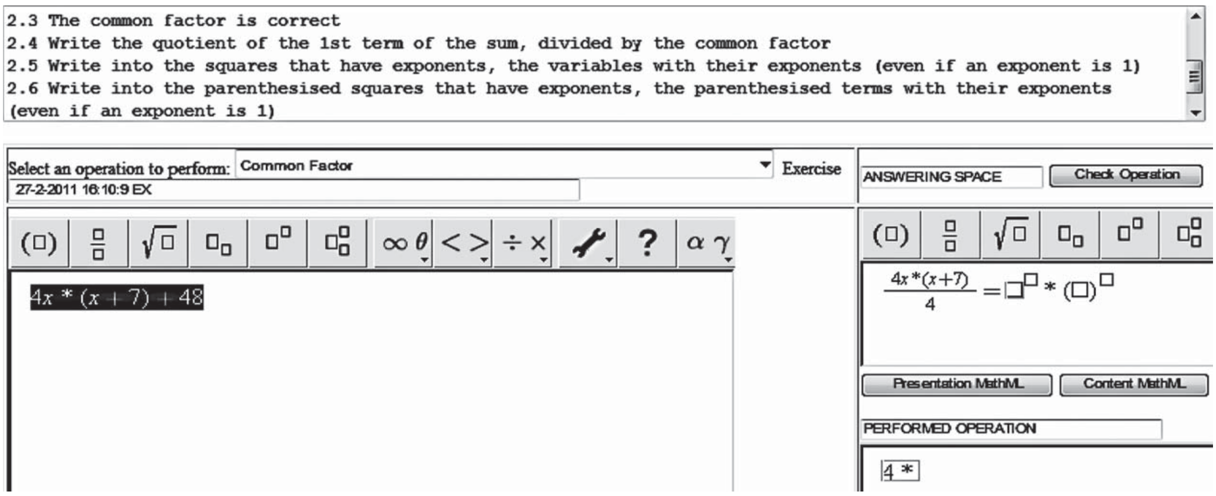

Fig. 4. The tutor confirms the entered common factor and asks for the first quotient by messages 2.3 and 2.4 (top). The quotient under question is $\frac{4x*(x+7)}{4} = x * (x + 7)$ denoted by the $\square^{\square} * (\square)^{\square}$ scaffold in the "ANSWERING SPACE" area (right).

meaning of the scaffold (Fig. 3, top, message 2.2). It must be noted that the tutor supports two other kinds of common factors: variables with exponents, denoted as $\square^{\square}$ and parentheses with exponents, denoted as $(\square)^{\square}$.

5. The student correctly enters 4 in the position indicated "ANSWERING SPACE" as the common factor and clicks the "Check Operation" button. The tutor performs *intelligent parsing* on the student's answer and confirms that it is correct (Fig. 4, top, message 2.3). The tutor also displays the common factor followed by a multiplication symbol, 4*, in the "PERFORMED OPERATION" area (Fig. 4, bottom right). The purpose of this area is to display the steps that have been performed in multi-step math skills. Now, the student must divide each one of the terms of the sum, i.e. $4x * (x + 7)$ and 48, by the

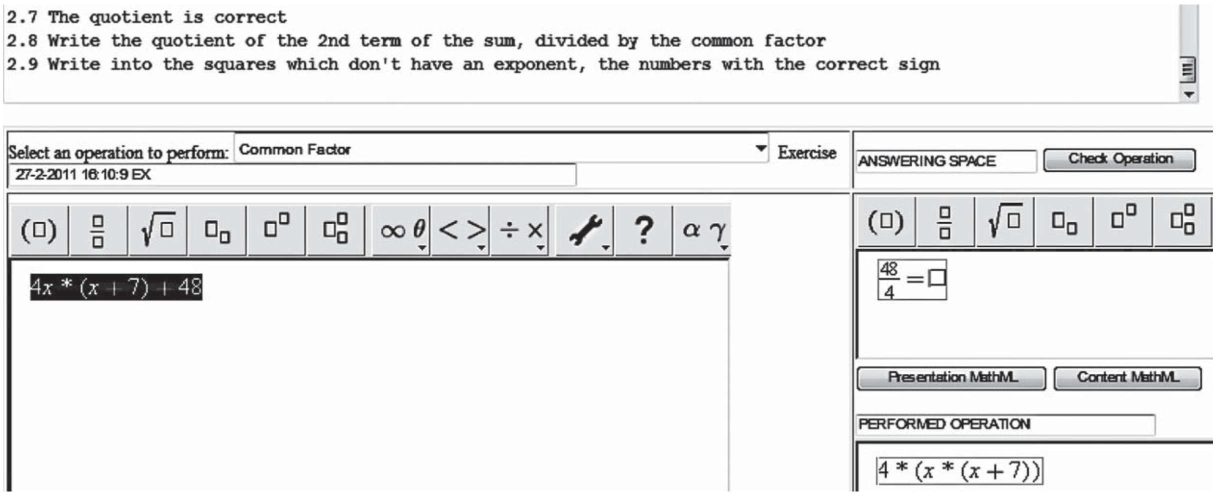

Fig. 5. The tutor confirms the first quotient and asks for the second quotient through messages 2.7 and 2.8 (top). The quotient under question is $\frac{48}{4} = 12$ denoted by the empty square scaffold in the "ANSWERING SPACE" area (right).

common factor. The first quotient that the student must calculate is $\frac{4x*(x+7)}{4} = x*(x+7)$. The tutor displays the quotient and a visual scaffold of the expected answer in the "ANSWERING SPACE" area (Fig. 4, right). The visual scaffold is $\square^{\square} *(\square)^{\square}$ denoting the expected answer $x^1 * (x+7)^1$.

6. The student enters in the squares of the visual scaffold $\square^{\square} * (\square)^{\square}$ the correct answer, $x^1 * (x+7)^1$ and clicks the "Check Operation" button. Once again, the tutor performs *intelligent parsing* on the student's answer and confirms that it is correct (Fig. 5, top, message 2.7). The tutor also displays the expression $4*(x*(x+7))$ in the "PERFORMED OPERATION" area (Fig. 5, bottom right) to denote the progress of the factoring process. The second quotient that the student must calculate is $\frac{48}{4} = 12$. The tutor displays the quotient and a visual scaffold of the expected answer in the "ANSWERING SPACE" area (Fig. 5, right). The visual scaffold is $\square$ denoting the expected answer 12.
7. As soon as the student correctly enters the second quotient, the tutor displays a confirmation message (Fig. 6, top, messages 2.10 and 2.11), rewrites the expression $4*(x*(x+7)+12)$, parses the rewritten expression, displays its abstract representation and prompts the student to perform the next operation, as shown in Fig. 6.
8. The student now selects $x*(x+7)$ and performs monomial–polynomial multiplication. Once more the tutor exhibits its *deep model tracing* behavior and guides the student step-by-step to perform the two monomial multiplications, $x*x$ and $x*7$ yielding $x^2 + 7x$. The result of this operation is shown in Fig. 7.
9. The student selects $x^2 + 7x + 12$ and performs factoring of the quadratic form $x^2 + Sx + P$ (trinomial). In order to achieve this, the student must find two integers a and b, such that $a \cdot b = P = +12$ and $a + b = S = +7$. The tutor, tracing its deep math domain model, guides the student in detail. First, the tutor prompts the student to identify $a \cdot b$ *and* $a + b$ (Fig. 8, top, message 6.2) and displays the corresponding scaffold in the "ANSWERING SPACE" (Fig. 8, right). The student correctly enters 12 and 7 for $a \cdot b$ *and* $a + b$ correspondingly (not shown in Fig. 8).
10. The student now has to discover that $a = 3$ and $b = 4$. The student enters the incorrect answer $a = 2$ and $b = 6$ (this step is not shown). The tutor displays an error message and suggests the possible pairs of values for a and b (Fig. 9, top, message 6.4), asking again for the values of a and b (Fig. 9, right)).

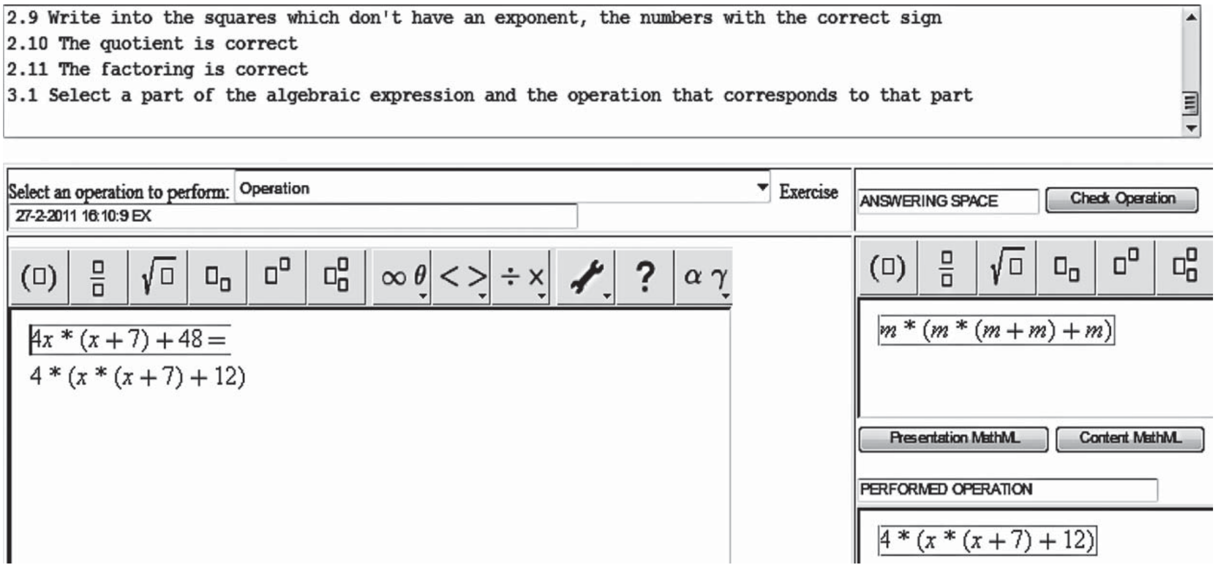

Fig. 6. Successful completion of the common factor method in expression $4x * (x + 7) + 48$.

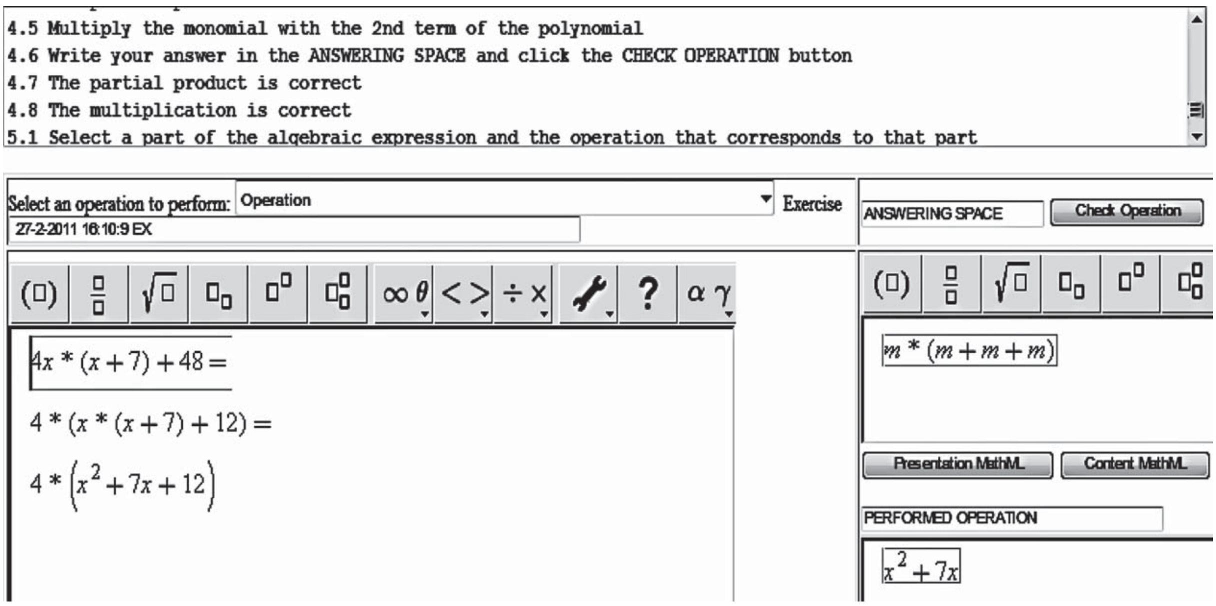

Fig. 7. Successful completion of the monomial-polynomial multiplication $x * (x + 7)$.

It must be noted that, for each one of the supported elementary skills, the model contains possible mistakes that the student might make. Each mistake is associated with error messages of varying depth, ranging from general suggestions down to the correct answer for the subtask. The depth and order of these messages are preset.

11. The student now enters the correct answer, $a = 3$ and $b = 4$ (not shown). The tutor checks the answer, confirms and rewrites the expression, yielding $4 * ((x + 3) * (x + 4))$. The factoring of $4x * (x + 7) + 48$ is now successfully completed (Fig. 10).

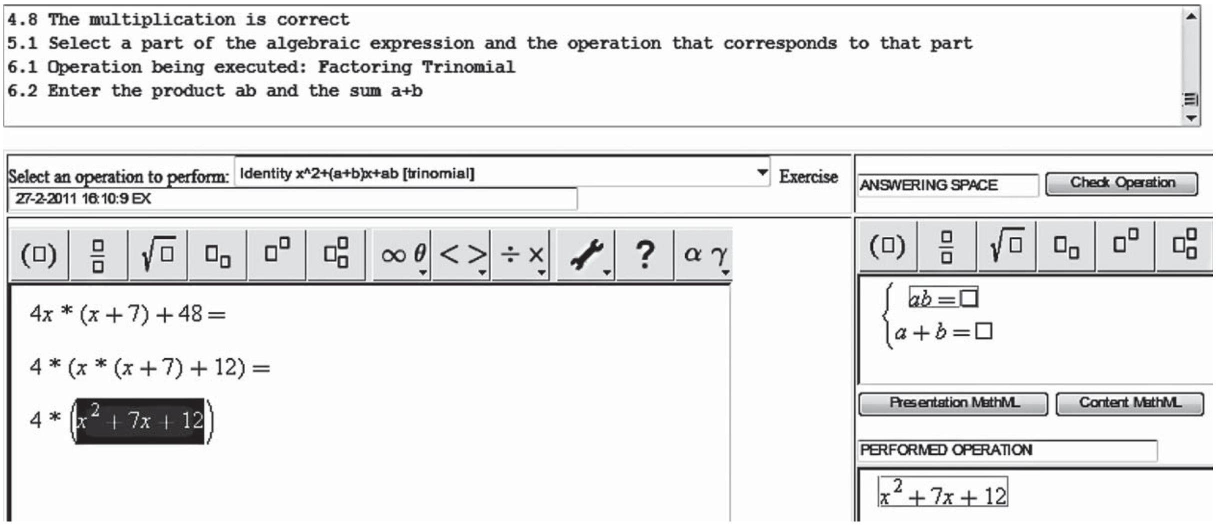

Fig. 8. First step of factoring $x^2 + 7x + 12$. The student must identify $a \cdot b = P = +12$ and $a + b = S = +7$.

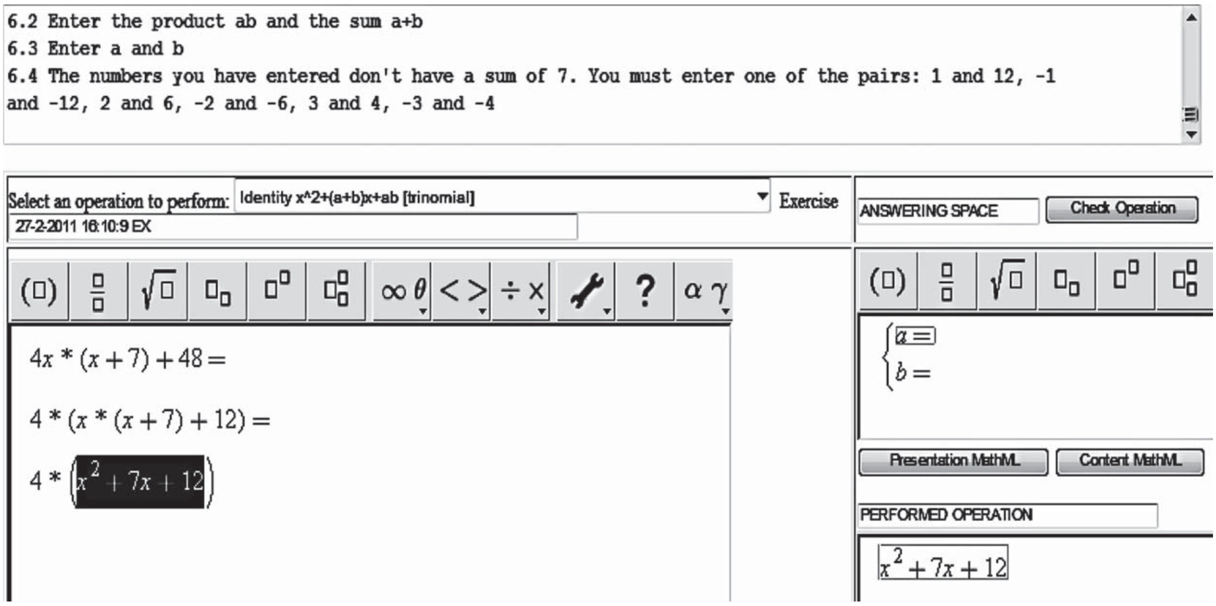

Fig. 9. Responding to a student error. The tutor displays an error message, gives help (top, message 6.4) and asks for the correct answer (right).

Once again, the scaling-up problem appears. The student could have followed a completely different solution path for factoring $4x * (x + 7) + 48$. The MATHESIS Algebra Tutor, based on its broad and deep expertise model as well as on the intelligent task recognition feature, is able to recognize this path and guide the student along.

Table 2 presents an alternative path in the solution space tree, involving only the top-level math skills (algebraic operations) the student could have followed and not the actual interaction with the tutor. As shown before, each one of these operations is a complex task that must be performed in a series of steps. The

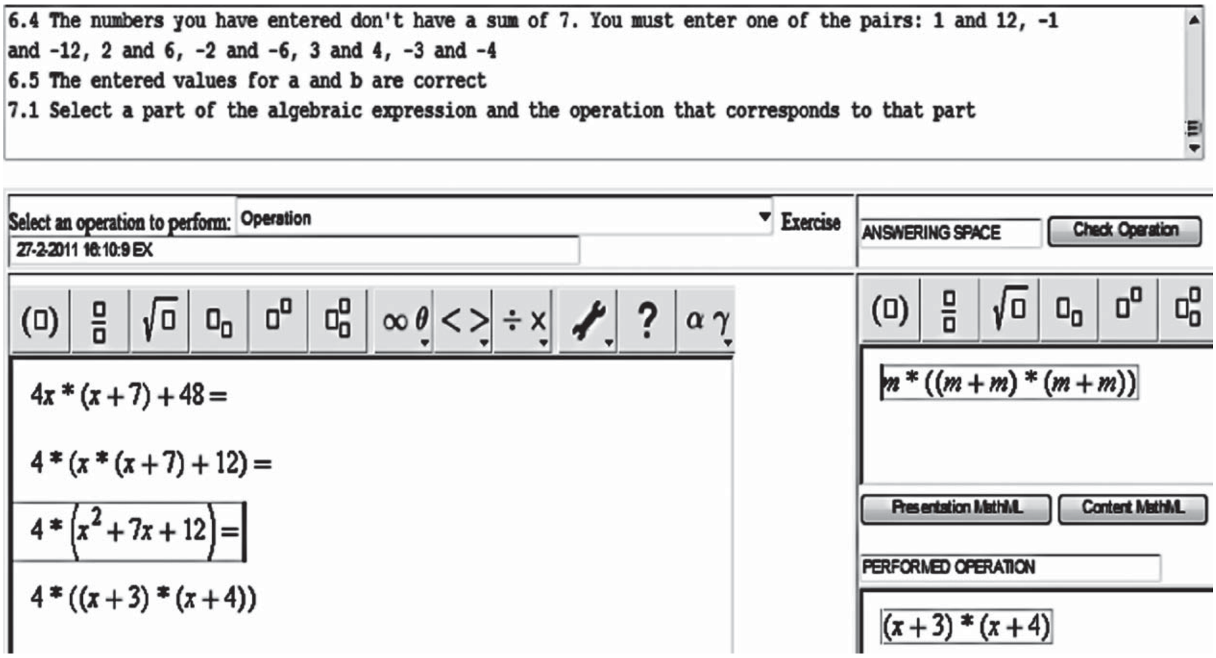

Fig. 10. Successful completion of factoring $4x * (x + 7) + 48$.

Table 2
Alternative path for factoring $4x * (x + 7) + 48$

Operation	Result
Initial expression	$4x * (x + 7) + 48 =$
1. Monomial-polynomial multiplication	$4x^2 + 28x + 48 =$
2. Common Factor	$4\,(x^2 + 7x + 12) =$
3. Factor $x^2 + Sx + P$	$4\,(x + 3)\,(x + 7)$

calculation of the quotient $\frac{4x*(x+7)}{4} = x * (x + 7)$ presented in step 5 (Fig. 4) demanded the development of a model for calculating quotients of arbitrary complexity, like, e.g., $\frac{8x^2y^3(x+2)^2(2x-1)^3}{2xy^2(x+2)\,(2x-1)^2}$. Equally complex is the task of finding two integers with a given product and sum, like the task presented in steps 9–11. As a consequence, if someone tried to draw the solution space tree for the factoring of expression $4x * (x + 7) + 48$ it would end up with a tree of considerable breadth and depth. The *fine-grained modelling* of each top level math skill (algebraic operation) and its sub-skills in conjunction with the *intelligent task recognition* described in the previous section, allows the MATHESIS Algebra tutor to guide the student throughout this broad and deep solution space. Thus, we call this feature *deep model tracing*.

The student model

Based on the breadth and depth of its math domain expertise model, the tutor creates and maintains in a database a deep and broad student model. For every step of the student's attempted solution, the tutor records the following information:

Table 3
The fine-grained student model: Solution steps

Skill	Expression	Answer	Correct
Automatic expression rewriting	$4x * (x + 7) + 48$	(*this step is performed by the tutor*)	1
Recognise the existence of a common factor	$4x * (x + 7) + 48$	Common factor	1
Calculate common factor	$4x * (x + 7) + 48$	4	1
Calculate the quotient of a term over the common factor	$\frac{4x * (x+7)}{4}$	$x * \ (x + 7)$	1
Calculate the quotient of a term over the common factor	$\frac{48}{4}$	12	1
Automatic expression rewriting	$4x * \ (x + 7) + 48 = 4 * \ (x * \ (x + 7) + 12)$	*(this step is performed by the tutor)*	1
Recognise a monomial by polynomial multiplication	$x * (x + 7)$	monomial by polynomial multiplication	1
Monomial multiplication	$x * x$	x^2	1
Monomial multiplication	$x * 7$	$7x$	1
Monomial by polynomial multiplication	$x * \ (x + 7)$	$x^2 + 7x$	1
Automatic expression rewriting	$4 * \ (x * \ (x + 7) + 12) = 4 * (x^2 + 7x + 12)$	*(this step is performed by the tutor)*	1
Recognise trinomial	$x^2 + 7x + 12$	Trinomial	1
Identify a and b	$a \cdot b = 12, \ a + b = 7$	$\begin{cases} a = 2 \\ b = 6 \end{cases}$	−1
Identify a and b	$a \cdot b = 12, \ a + b = 7$	$\begin{cases} a = 3 \\ b = 4 \end{cases}$	1
Automatic expression rewriting	$4 * \ (x^2 + 7x + 12) = 4 * ((x + 3) * \ (x + 4))$	*(this step is performed by the tutor)*	1

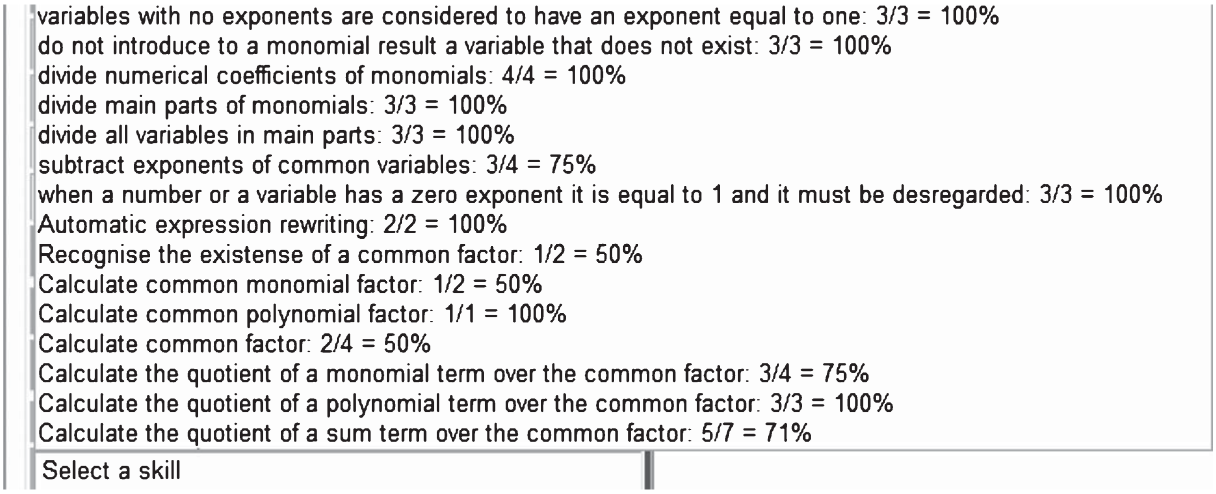

Fig. 11. The student model: Skill performance statistics.

Table 4
Performance of skill "Calculate common factor". The percentage is 2/4 = 50%

Operation	Expression	Answer	Correct	Date
Calculate common factor	$4x * (x + 7) + 48$	4	1	27-02-2011 16:55:33
Calculate common factor	$4x * (x + 7) + 48$	4	1	22-02-2011 18:26:07
Calculate common factor	$4x * (x + 7) + 48$	$4x$	-1	22-02-2011 18:26:02
Calculate common factor	$4x * (x + 7) + 48$	$4x$	-1	22-02-2011 18:19:53

- *Skill:* The algebraic operation that the student tried to perform in the specific step, e.g., "common factor calculation".
- *Expression:* The algebraic expression on which the algebraic operation was performed, like $4x * (x + 7) + 48$.
- *Answer:* The answer given by the student, for example $4x$.
- *Correct:* It signifies whether the answer was right (1) or wrong (-1).
- *Timestamp:* The date and time the step was performed.

This information is presented in a table, with one row for each solution step. The table for factoring the expression $4x * (x + 7) + 48$ is shown in Table 3. Rows with dark background emphasize incorrect steps. Both students and their teachers can see this tabular representation of the student's solution steps.

In addition, the tutor can display statistics over a selected period of time about a specific cognitive skill, as shown in Fig. 11. When a specific skill is selected, a table presenting the performance of the skill is displayed (Table 4).

It becomes obvious that such a detailed and time-stamped student model creates a digital timeline of the student's math skill mastery over time, with a number of possible uses: long term progress assessment, recent mastery status, automatic selection of exercises based on the student's weaknesses. The latter is not yet implemented in the system.

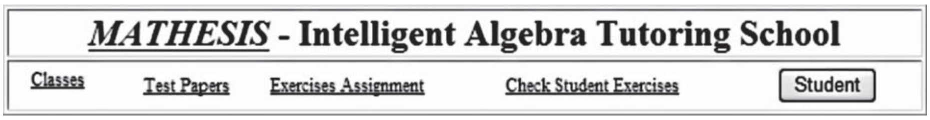

Fig. 12. The teachers' menu.

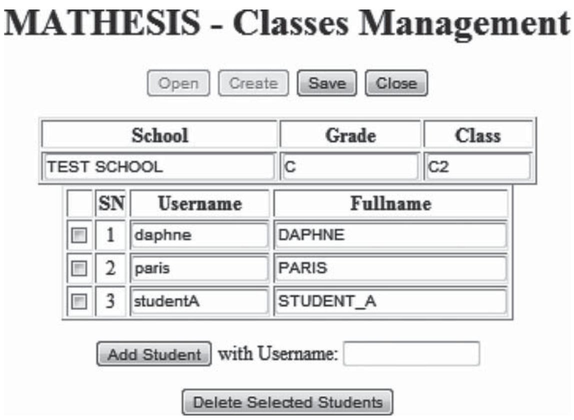

Fig. 13. The classes management page.

THE LEARNING MANAGEMENT SYSTEM

The MATHESIS Intelligent Algebra Tutoring School is accessible through a web interface[3]. Each user gets a unique Username and Password. Users can register either as teachers or students. Students are guided to the MATHESIS Algebra Tutor interface (Fig. 1), where they solve their assigned exercises as it was described in Section 2.2. Teachers are taken to the Teacher Menu (Fig. 12), which provides links for the following managerial tasks:

- *Classes:* Teachers can create classes. For each class the teacher enters the real school, grade and name of the class. Students are registered to the class by their Usernames. That means that the students must be already registered in the system. Students can also be deleted from a class. (Fig. 13).
- *Test Papers:* The system provides an online HTML editor for the creation and editing of test papers (Fig. 14). For each test paper the teacher enters the type of school, grade, book, chapter and section of a textbook that the contained exercises correspond to. Each test paper is also characterized as *public* or *private* (Fig. 14a). Public test papers can be accessed and used (but not modified) from any teacher registered in the system, while private ones can be used and edited only by their creator. Test papers are used for the assignment of exercises to students. Currently, the system provides five public test papers that contain exercises from the official textbook that is taught in the 3rd grade of Gymnasium (junior high school) in secondary education in Greece. Each test paper is an HTML page. Conceptually, each paper is organized as a set of exercises containing one or more questions. For each exercise, its questions are laid out in rows and columns using HTML tables. The author inserts new exercises by defining how many questions they contain and in how many rows and columns they will be arranged, using the "Insert Exercise" button and the corresponding fields (Fig. 14, left, below the editor). The system creates the appropriate HTML code for the table and displays it in the editing area.

[3]http://users.sch.gr/dsklavakis

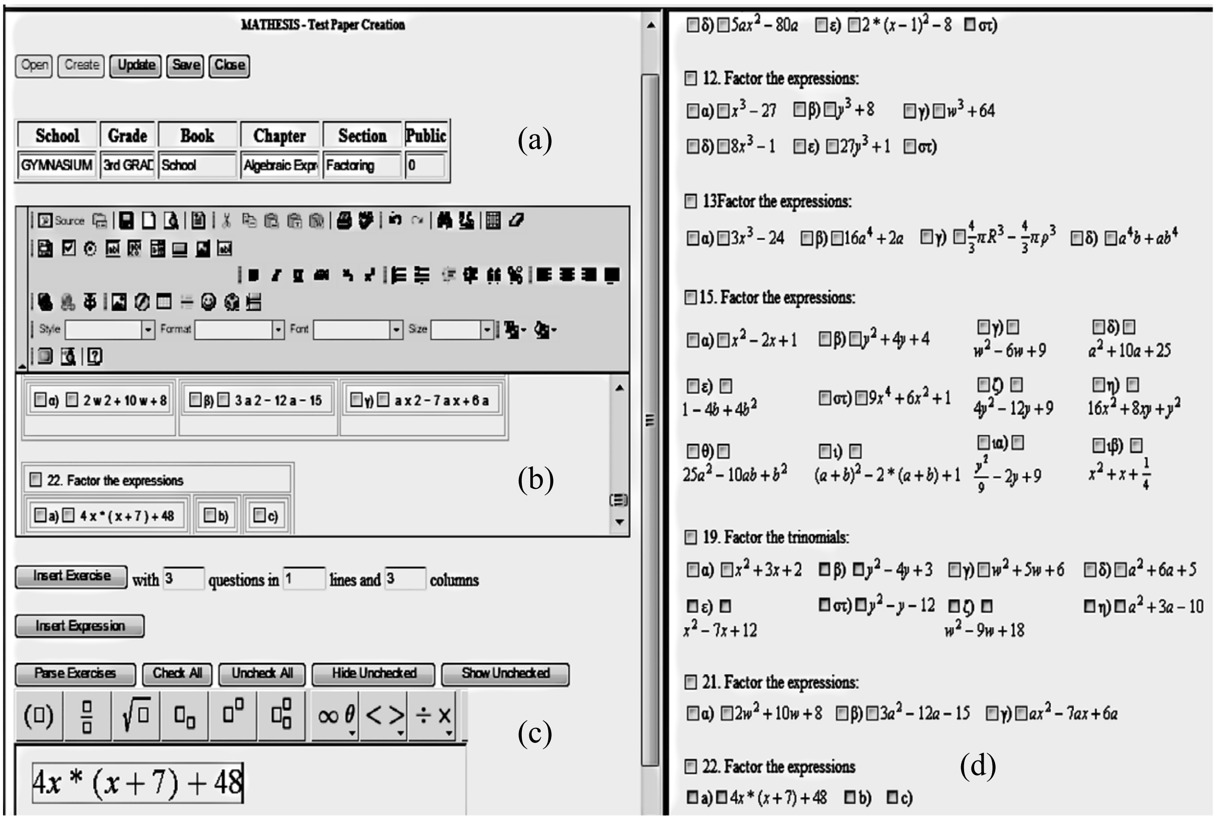

Fig. 14. Test paper editing. The author has just created exercise no. 22 using the HTML editor (b) and inserted expression $4x * (x + 7) + 48$ for the first question using the math editor (c). The paper is shown on the right with the newly added exercise at the bottom (d).

It also generates check boxes with unique identification strings in front of the exercise and each of its questions (Fig. 14b). These check boxes are used later for selecting and assigning exercises (Fig. 15). The author adds any text for describing the exercise and its questions. In Fig. 14, exercise 22 has just been added, containing 3 questions, arranged in one row and three columns, labeled by the author as 'a)', 'b)' and 'c)' (Fig. 14b). Finally, for each question, the author enters the algebraic expression using a WebEq Input Control. In Fig. 14, the author has just entered the expression $4x * (x + 7) + 48$ in question (a) of exercise 22 (Fig. 14c). The system displays on the right side of the editor the test paper as an HTML page, using the MathML viewer MathPlayer to display properly the mathematical expressions (Fig. 14d). The HTML code of each test paper is saved in a database, together with the papers' information, and can be recalled and edited any time by changing, adding or deleting exercises. It must be noted that, due to the intelligent task recognition feature of the tutor, the authors do not have to annotate or describe any solution steps for the questions.

- *Exercise assignment:* The system provides tools for individualized assignment of exercises. The teacher can assign different exercises to different students, according to their performance. The assignment process is simple: The teacher selects a class and any student(s) from this class as well as a test paper and any exercise(s) from it. By checking the appropriate boxes, the selected exercise(s) are assigned to the selected student(s) (Fig. 15).

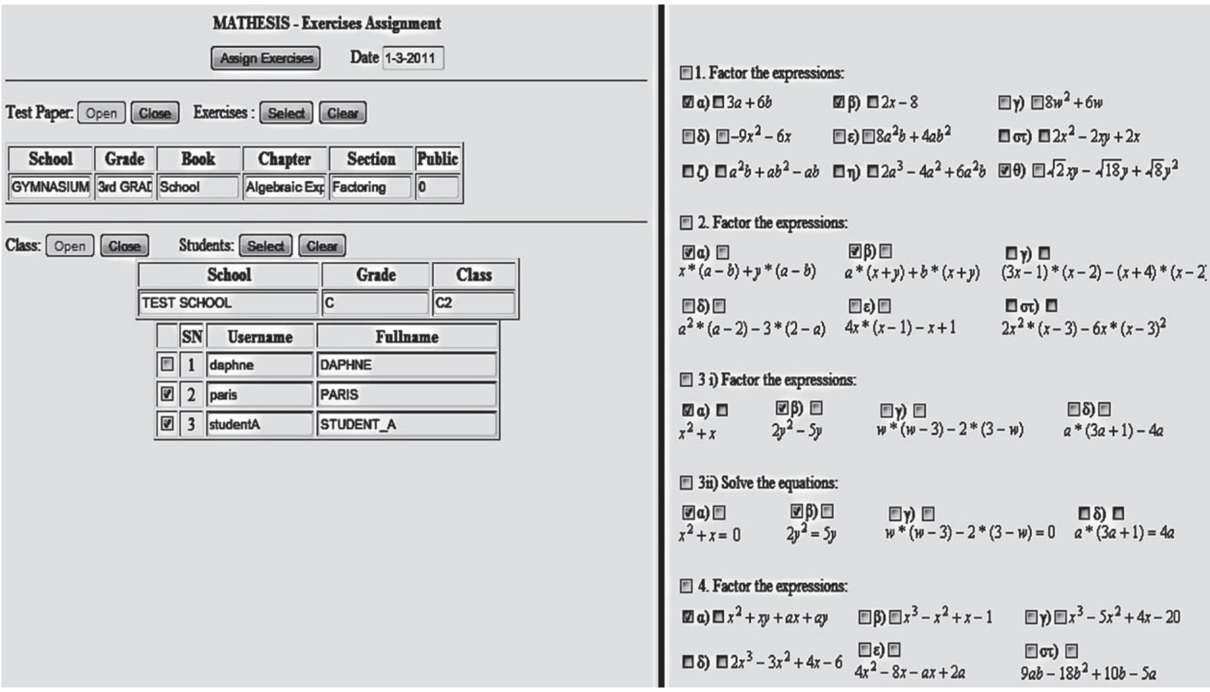

Fig. 15. Individualized assignment of exercises to students.

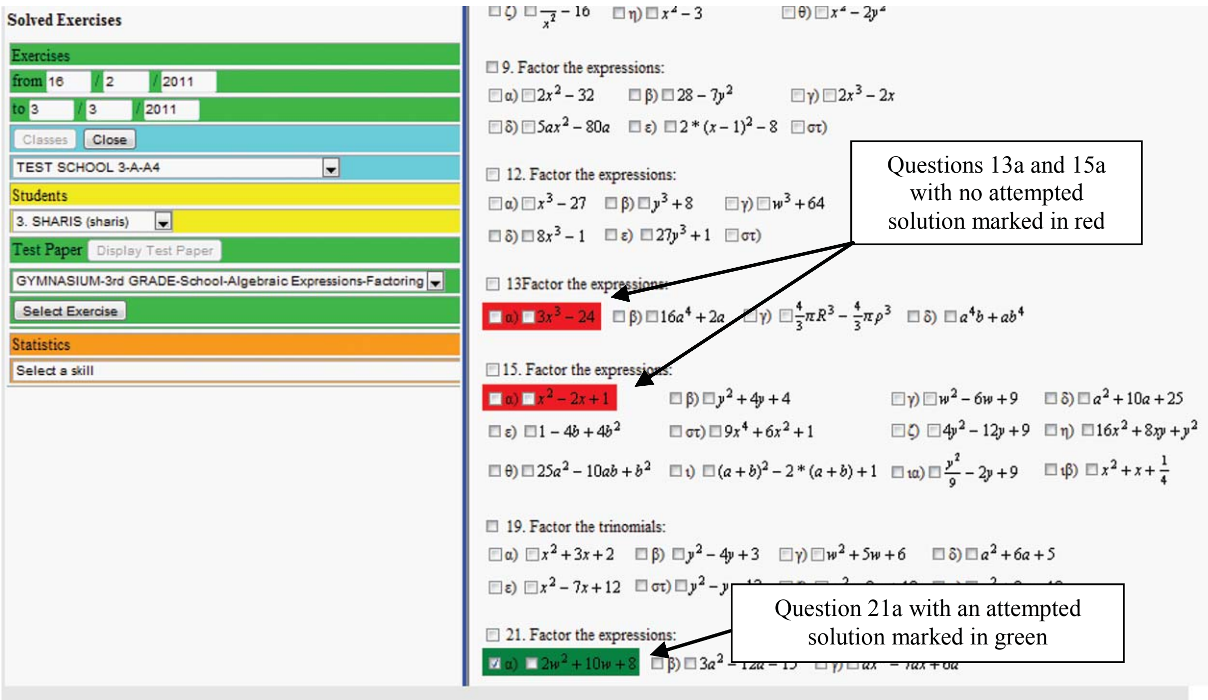

Fig. 16. Student assessment: Selecting a solved exercise.

- *Student assessment:* The solution steps taken by a student are recorded in the database and statistics are computed about the correct/incorrect performance of operations. These steps and statistics can be retrieved and viewed by the teacher. On the left side of Fig. 16, the teacher selects the time interval for which he/she wants to assess the student(s). He/She opens a classroom and selects a student. The system displays in a drop down list all the test papers containing exercises that were assigned to the student during the selected time period. The teacher selects a test paper and its contents are displayed (Fig. 16, right). Assigned exercises for which no solution was attempted by the student are marked in a red background. In Fig. 16, these are questions 13a) $3x^3 - 24$ and 15a) $x^2 - 2x + 1$ located in the middle of the test paper (red color appears as dark grey in grayscale). Those with at least one attempted solution, either correct or wrong, are marked with green background. In Fig. 16 this is question 21a) $2w^2 + 10w + 8$ (green color appears as light grey in grayscale).

By selecting an exercise and clicking the "Select Exercise" button, the attempted solution steps are displayed as shown in Table 3. The teacher can also select a specific math skill from the drop-down list on the lower left part of Fig. 16. As mentioned in Section 2.3, the list displays all skills performed by the student with their corresponding percentage of correct performances during the selected time period, as shown in Fig. 11. By selecting a specific skill, a table of the skill performances taken into account is displayed (Table 4).

RELATED WORK

The development of the domain expertise and the implementation of the model-tracing tutoring model in model-tracing tutors are so demanding in time and human resources (Aleven et al., 2006) that these tutors are currently developed by specialised research teams, they are usually experimental prototypes and they are used in strictly controlled and supervised educational settings, mainly in universities (VanLehn, 2006). The most successful and widely used math MTTs are Cognitive Tutors developed by Carnegie Learning[4], based on more than twenty years of cognitive science research at CMU (Koedinger, & Corbett 2006). Cognitive Tutors are now an integral part of complete curricula used in hundreds of middle and high schools throughout the United States. However, despite their innovative nature and practical success, Cognitive Tutors are commercial products that have to adapt to very strict guidelines and educational goals of the US educational system. They have to follow the textbook by teaching specific exercises that train the students in specific cognitive skills. In the case of algebraic expressions' operations, they teach each operation separately and not in combinations with each other. They also teach a fixed set of exercises where all the anticipated solution steps are *pre-computed* by solving the problem in all acceptable ways by running a rule-based problem-solver (Van Lehn 2006). Therefore, these tutors do not tackle the problem of parsing an *arbitrary* algebraic expression, identifying the existence of any possible combination of operations and their precedence and following the student in any possible correct path of the solution space tree. In other words, they are not designed to deal with the scaling-up problem (Aleven, McLaren, & Sewall, 2009).

Another kind of pseudo-MTTs is the example-tracing tutors (Aleven, McLaren, Sewall, & Koedinger, 2009) under development at Carnegie Mellon University. There are two websites that provide example-tracing tutors for middle-school mathematics: the Mathtutor[5] website (Aleven, McLaren, & Sewall, 2009) and the Assistments[6] website (Razzaq, Feng, Nuzzo-Jones, Heffernan, Koedinger, Junker et al., 2005). Example-tracing tutors have a very narrow and shallow, exercise-specific, domain expertise model. They

[4] www.carnegielearning.com
[5] https://mathtutor.web.cmu.edu/
[6] www.assistments.org

offer considerable reduction in development time but are even further away from dealing with the scaling-up issue.

ActiveMath[7] is another web-based intelligent tutoring system for mathematics (Mellis, Andrès, Büdenbender, Frischauf, Goguadze, Libbrecht et al. 2001). The systems aims mainly for adaptive guidance and presentation of mathematical content based on ontological representation of mathematical concepts, learning goals and acquired knowledge. However, when it comes to problem-solving skills, ActiveMath offers mainly multiple choice questions and some more interactive exercises. In these, the system does not guide the student along a solution path. It uses the external Computer Algebra Systems (CAS) to simply check the correctness of the student's solution. Therefore, the system completely avoids the hard problems of model tracing, that is, generating the correct solution(s) at each step, comparing the students' input, recognising errors and providing feedback.

Aplusix[8] is an Algebra Learning Assistant. After several years of research (Nicaud, Bouhineau, & Chaachoua, 2004), it is now a commercial product. It covers the domains of arithmetic calculations, expansion, simplification and factoring of algebraic expressions, solution of polynomial and rational equalities and inequalities. The system combines features of microworlds and Computer Algebra Systems. The student can type an algebraic expression, suggest its domain (calculation, expansion-simplification, factoring, solution) and enter the solution steps. At each step, the system checks the student's input for *equivalence* using encoded transformation rules. As a result of this type of checking, the system only suggests if the expression entered by the student is correct or incorrect, without any further feedback about the error committed. However, the student can ask for suggestions about the possible operations that he/she can perform and can also ask the system to perform them. We could say that the resulting tutoring model is almost equivalent with that of the MATHESIS tutor though less fine-grained. In unusual situations, this can lead the Aplusix system to "miss" intermediate student errors. For example, the expression $2x - (6 - x) - (6 + x)$ is correctly expanded and simplified by changing the signs of the parenthesized terms as in $2x - 6 + x - 6 - x = 2x - 12$. However, a student can arrive at the correct result by making the same mistake twice, that is, not changing the signs of $-x$ in the first parenthesis and of $+x$ in the second one, as in $2x - 6 - x - 6 + x = 2x - 12$! Moreover, the Aplusix system has considerable limitations to the kind of expressions that it can factor: polynomial expressions in one variable and degree no higher than 4, or in two variables and degree at most 2. It cannot handle expressions like $3x^2y^2z^3 - 6xyz^2$, $x^4 - y^4$, $(x + y)^2 - z^2$ or $4x\ (x + 7) + 48$.

As far as we know, the MATHESIS Algebra Tutor is unique with regard to the combined breadth and depth of its domain expertise model as well as the intelligent task recognition feature.

EVALUATION OF THE MATHESIS SYSTEM

The MATHESIS Algebra Tutor is a research prototype, performance-oriented, domain expert system with emphasis on the scaling up problem. The tutor is part of the MATHESIS project, which aims at the development of authoring tools for real world model-tracing math tutors. Therefore, the MATHESIS Algebra Tutor and the MATHESIS tutoring school built around it were designed to become part of real educational settings. For this reason, the following factors were taken into consideration:

1. *Teaching performance:* In order for an intelligent system to be used by teachers and students, it should contribute to observable positive learning outcomes. Besides any kind of scientific evaluation, teachers and students must feel and see that using the system helps students learn more effectively. It has been

[7] www.activemath.org
[8] www.aplusix.com

shown that model-tracing tutors do produce considerable learning outcomes, mainly because of their domain expertise models (Corbett 2001; Ritter, Kulikowich, Lei, McGuire, & Morgan 2007). In this work we adopted a holistic approach: developing a deep model of a sufficiently broad domain in mathematics with intelligent task recognition and deep model-tracing.

2. *Usability:* This factor is multidimensional, with the most important dimensions being:
    a. Easy to learn and use interface. We have tried to keep the user interface as simple as possible – given the complex task of teaching that this interface must perform – and as close as possible to the "traditional" way of doing things. For the teachers, this means following the day-to-day workflow of selecting, assigning and assessing exercises. For the students, we tried to keep the problem-solving procedure as close as possible to the pen and paper paradigm without losing the benefits of a digital environment.
    b. Easy access to the system. The MATHESIS system is web based and therefore accessible anytime from anywhere, provided there is an internet connection. In addition, it has minimal requirements in hardware and connection speed.
3. *Scalability:* The set of exercises that the tutor is able to teach has to be of considerable breadth and depth. Limiting the set of supported exercises is a major factor of system rejection by the teachers. Teachers must be given the flexibility to choose exercises of different complexity and difficulty levels in order to accommodate the varying levels of competence of their students. The systems' deep and broad domain expertise model in conjunction with the intelligent task recognition system covers a considerable set of exercises.

Evaluation by teachers

The system has been demonstrated to real math teachers, both through on-site live presentations and through invitations to use it online. The most extensive evaluation of the system was held in a three hours' workshop at the 2nd PanHellenic Conference on Digital and Web Applications in Education, held in Naoussa in April of 2010 (http://hmathia10.ekped.gr/) The purpose of the workshop was to teach math teachers the use of the system and investigate their attitude towards adopting the system in their everyday teaching. More specifically, we wanted to investigate their opinions regarding the following system features, which we consider the most decisive for the adoption of the system by a broad group of math teachers:

- The usability of the system.
- The ability to create their own exercises and assign them to individual students.
- The teaching performance of the system, particularly the depth and granularity of the domain model.
- The value of the fine-grained student model for their assessment tasks.

Forty (40) math teachers in secondary education participated in the workshop. Most of them were young, around 30 years old, self-motivated and positive in using computer programs for math teaching.

First, the teachers used the LMS to sign up, create students and enroll them to classes. Then, they used the existing test papers to assign exercises to their students. They have actually assigned one exercise for each one of the 16 top-level skills covered by the tutor as well as a few exercises with combinations of these skills. The teachers spent most of their time solving the assigned exercises as if they were students. They were also instructed to make deliberate mistakes to test the system's responses. They were also instructed to inspect the student model between the solutions of the exercises to see how this model was dynamically updated by their performance as students.

After using the system, the teachers filled in a short questionnaire. The questions and the teachers' answers are shown in Table 5. These questions are in direct correspondence with the aforementioned system features we wanted to evaluate.

Table 5
Evaluation results given by forty (40) math teachers after a three-hour hands-on workshop (questions are translated from Greek)

Questions	Answers			
1. You find the overall use of the system...	Easy 31/40 (77.5%)	Fairly Easy 4/40 (10.0%)	Fairly Hard 3/40 (7.5%)	Hard 2/40 (5.0%)
2. How well does the Learning Management System fits your day-to-day teaching tasks?	Very much 19/40 (47.5%)	Much 13/40 (32,5%)	Quite well 8/40 (20.0%)	Not at all 0/40 (0.0%)
3. You find the ability to create your own exercises as...	Very Important 18/40 (45.0%)	Important 10/40 (25.0%)	Indifferent 12/40 (30.0%)	Useless 0/40 (0.0%)
4. You find the ability to assign different exercises to different students as...	Very Important 18/40 (45.0%)	Important 10/40 (25.0%)	Indifferent 12/40 (30.0%)	Useless 0/40 (0.0%)
5. Do you think that the level of analysis for the solution steps proposed for each operation is...	Excessive 8/40 (20.0%)	Normal 32/40 (80.0%)	Inadequate 0/40 (0.0%)	
6. How would you characterize the step-by-step guidance of the student?	Very Important 40/40 (100.0%)	Important 0/40 (0.0%)	Indifferent 0/40 (0.0%)	Useless 0/40 (0.0%)
7. How would you characterize the ability to see the students' solution steps regarding his/her assessment?	Very Important 40/40 (100.0%)	Important 0/40 (0.0%)	Indifferent 0/40 (0.0%)	Useless 0/40 (0.0%)

Thirty five teachers (87.5%) found the system easy or fairly easy to use (Question 1). Thirty two teachers (80%) agreed that it naturally follows the short- and long-term tutoring tasks workflow (Question 2). Twenty eight teachers (70%) appreciated the freedom provided by the system to create their own work papers with their own exercises, as well as the ability for *individualized* assignment of exercises (Questions 3 and 4). Thirty two teachers (80%) found the fine grained student model unique and decisive when it came to assessment. However, five teachers (12.5%) considered that it might be too fine-grained for well-performing students. Three teachers (7.5%) complained that this step-by-step guidance of the model-tracing algorithm could be too authoritative and restrictive in the development of the students' self-confidence (Question 5). All forty (40) teachers were impressed by the human-like step-by-step guidance given to the student by the system and the ability to see the students' solution steps (Questions 6 and 7).

Evaluation in a real classroom

In late 2011 the system was also used and evaluated for three months in a third grade class (ages 14-15) of 20 students in a junior high school at the town of Drama, in northern Greece. The purpose of this evaluation was to integrate the use of the system in the normal, daily, official educational practice and investigate the following features:

Table 6
Evaluation results given by twenty (20) students after a three-month period (questions are translated from Greek)

Questions	Answers			
1. You find the overall use of the system...	Easy 17/20 (85.0%)	Fairly Easy 5/20 (15.0%)	Fairly Hard 0/20 (0.0%)	Hard 0/20 (0.0%)
2. How would you characterize the step-by-step guidance of the tutor?	Too detailed 5/20 (25.0%)	Natural 15/20 (75.0%)	Inadequate 0/20 (0.0%)	
3. You find that your frustration when you solve an exercise with the tutor is...	Bigger 2/20 (10.0%)	Equal 1/20 (5.0%)	Lower 17/20 (85.0%)	
4. Which do you think are the most important advantages for you when using the tutor? (multiple answers)	Adequate time to think 15/20 (75.0%)	Freedom to make mistakes 18/20 (90.0%)	Step-by-step guidance 13/20 (65.0%)	Ability to try possible solutions 16/20 (65.0%)

- The usability of the system.
- The students' attitude towards the tutoring performance of the system, particularly the fine-grained, step-by-step guidance provided by the system.
- The affective impact of the system to the students, particularly the impact on frustration and fear during the solution of exercises.
- The potential raise of student performance.

Mathematics in this grade is taught four hours a week using the textbook, blackboard lessons and worksheet practice both in classroom and at home. In our evaluation three hours were taught in the traditional way using blackboard lessons and worksheet practice. The fourth hour was taught in the school's computer laboratory, where students used the MATHESIS system. Some of the students also used the system from their homes for extra practice. The system was evaluated by the students for its usability and tutoring behaviour using short questionnaires (Table 6). The results of the students' evaluation are:

Usability

85% of the students found the system easy to learn and use, while the rest 15% found it fairly easy to learn (Question 1). In practice, the first group of students (85%) needed one or two 45-minute sessions with the system to get fully acquainted while the second group (15%) needed three or four sessions.

Tutoring performance

75% of the students said that the guidance and assistance they got from the system was similar to the human tutor's teaching. The rest 25% found the help and guidance of the system too detailed and fine grained (Question 2). These students were the best performing ones and they proposed that the system should allow the student to skip some "trivial" problem solving steps.

Affective impact

85% of the students replied that the use of the system helped them to overcome the most common emotional problems they face with mathematics, that is, frustration and disappointment (Question 3). The reasons are that they have the time they need to think (75%), they get step-by-step guidance (65%), they have the freedom to try the solution steps they think correct (65%) and make mistakes (90%) (Question 4).

Cognitive performance

We believe that the most important attribute of an intelligent tutoring system is its cognitive performance, that is, its ability to build deep, long-term and transferable knowledge within the student's minds. The cognitive performance of the MATHESIS Algebra Tutor was specifically tested in the domain of factoring, using the methods of *common factor and identities difference of squares* $x^2 - y^2 = (x + y)\ (x - y)$, *square of sum* $x^2 + 2xy + y^2 = (x + y)^2$ and *square of difference* $x^2 - 2xy + y^2 = (x - y)^2$. The students were initially taught this subject for six weeks without using the system at all. After this period, the students completed a test to assess mastery of the subject. Then, the students used the MATHESIS system for two weeks to solve all the relevant exercises provided by the system. Some of these exercises can be found in: Fig. 15, exercises 1, 2, 3 and 4; Fig. 14, exercise 15; and Fig. 16, exercise 9. Right after they had completed these exercises, they took a post-test with exercises similar to those of the pre-test. The results are shown in Table 7. There, the pre-test items are denoted by "Pre", while post-test items are denoted by "Post". In the left column four pairs of exercises are shown. For each pair the pre-test and the post-test exercises are shown. The next three columns show the elementary math skills needed to correctly perform each factoring method. For each skill, the percentages of students who performed it correctly are shown both for the pre-test and post-test exercises.

Exercise 1 is a common factor method. Exercises 2 and 3 correspond to the three different identities mentioned above. Although they seem to share some identical sub-skills, like the "Find squares" and "Apply identity", in practice the identity $x^2 \pm 2xy + y^2 = (x \pm y)^2$ is more demanding: the student has to verify that the third term is actually the double product of the two squares and take into account the sign of the double product. The similar success percentages in Exercises 2 and 3 do not reflect these subtle differences in the application of these identities. Exercise 4 is a more complex one. First, the term x^4 is a square of a square that is, $(x^2)^2$. Second, after the first application of the identity $x^2 - y^2 = (x + y)\ (x - y)$, the term $(4 - x^2)$, which is also a difference of squares, appears. These two difficulty factors significantly reduce the success percentages. In the pre-test only nine students (45%) recognized that $x^4 = (x^2)^2$ and of these students, only four (20%) factored the term $(4 - x^2)$. The corresponding results for the post-test (60% and 45% correspondingly) are considerably raised but still remain low.

This comparison in our opinion further supports our empirical observation that in mathematics there are non-intuitive practical differences in what are formally "identical tasks". It seems that the application of the same task (square recognition) in a more complicated expression, like x^4, demands the recall and application of "deeper" sub-skills like the one expressed by the formula $x^{2n} = (x^n)^2$. In turn, this fact supports the necessity for broader and deeper models in intelligent tutoring systems. In any case, the results in Table 7 show a considerable performance rise, given the limited time of two weeks that the students had in their disposal for using the MATHESIS system.

DISCUSSION AND FURTHER WORK

We believe that the MATHESIS system and especially the MATHESIS Algebra Tutor is a successful proof-of-concept system. The basic research hypothesis of the MATHESIS project is that, in order to build successful intelligent real-world tutoring systems, we must build powerful domain expertise models. The engineering of such broad and deep models has to overcome the common obstacle of all expert systems, the *knowledge acquisition bottleneck*: the extraction of the expertise from domain experts and its

Table 7
Students' performance rise by the MATHESIS algebra tutor

	Math skills					
Exercise 1 Pre: $6xy^2 + 3x^2y - 9xy = 3xy\,(2y + x - 3)$ Post: $8x^2y^3 - 4x^2y^2 - 12x^3y^4 = 4x^2y^2\,(2y - 1 - 3xy^2)$						
	Recognize common factor method		Calculate common factor		Calculate quotients inside the parenthesis	
	Pre	Post	Pre	Post	Pre	Post
	85%	90%	65%	85%	70%	80%
Exercise 2 Pre: $4y^2 - 81 = (2y)^2 - 9^2 = (2y - 9)\,(2y + 9)$ Post: $9 - 25x^2 = 3^2 - (5x)^2 = (3 - 5x)\,(3 + 5x)$						
	Recognize difference of squares method $x^2 - y^2 = (x + y)\,(x - y)$		Find the squares		Apply the identity	
	Pre	Post	Pre	Post	Pre	Post
	85%	95%	50%	65%	60%	80%
Exercise 3 Pre: $x^2 + 6xy + 9y^2 = x^2 + 2 \cdot x \cdot 3y + (3y)^2 = (x + 3y)^2$ Post: $25a^2 - 60ab + 36b^2 = (5a)^2 - 2 \cdot 5a \cdot 6b + (6b)^2 = (5a - 6b)^2$						
	Recognize square of sum method $x^2 + 2xy + y^2 = (x \pm y)^2$		Find the squares and the double product		Apply the identity	
	Pre	Post	Pre	Post	Pre	Post
	85%	95%	50%	65%	60%	80%
Exercise 4 Pre: $a^4 - 81 = (a^2)^2 - 9^2 = (a^2 + 9)\,(a^2 - 9) = (a^2 + 9)\,(a + 3)\,(a - 3)$ Post: $16 - x^4 = 4^2 - (x^2)^2 = (4 + x^2) \cdot (4 - x^2)$ $= (4 + x^2) \cdot (2^2 - x^2) = (4 + x^2) \cdot (2 + x) \cdot (2 - x)$						
	Recognize difference of squares method $x^2 - y^2 = (x + y)\,(x - y)$		Find the squares		Apply the identity	
	Pre	Post	Pre	Post	Pre	Post
	9/20	12/20	7/20	11/20	7/20	9/20
	45%	60%	35%	55%	35%	45%
	4/20	9/20	4/20	6/20	3/20	6/20
	20%	45%	20%	30%	15%	30%
	(4/9 44%)	(9/12 75%)	(4/7 57%)	(6/11 55%)	(3/7 43%)	(6/9 67%)

representation in efficient ways. In the domain of knowledge engineering, the most profitable solution up to now is *knowledge reuse*, which is achieved through open, modular, interchangeable, inspect-able, formal knowledge representations and system implementations (Aitken & Sklavakis, 1999). Equally important, the models must be deep and broad, having a wide basis of low level knowledge about simple task performance, on top of which is built the knowledge for performing higher level domain tasks. Otherwise, models are *brittle* (Lenat & Guha, 1990), performance is limited, scaling up is intractable and the systems fail to cope with real-world demands. We believe that the MATHESIS Algebra Tutor incorporates all these characteristics that make it a successful real-world intelligent tutoring system.

Of course, the system is an experimental prototype and more evaluation is needed. The teachers that took part in its evaluation were self-motivated and enthusiastic about the use of technology in education. Also, they did not use the system for a long period of time in their everyday teaching duties and they were under our direct supervision when they met any difficulties in using the system. Therefore, more evaluation is needed before the system is ready for widespread use by a broad group of teachers. As for the learning outcomes, we have not used a comparison group of students. The reason is that our system is designed as an additional learning aid and not as a self-contained teaching method. In addition, we evaluated the system only in the domain of factoring and not the whole domain that the system covers. Finally, a feature of the system that has not been adequately evaluated is its fine-grained student model and the possible benefits of the detailed information it provides to both students and teachers.

In order to further investigate the reusability and expandability of the system, we intend to extend its domain model to teach algebraic operations of rational algebraic expressions. To simplify rational expressions, a student should make full use of the operations already taught by the MATHESIS Algebra Tutor. Implementing such a demanding task will be the best test for the knowledge reusability and implementation extensibility of the MATHESIS system.

REFERENCES

Aitken, S. J., & Sklavakis, D. (1999). Integrating Problem Solving Methods into CYC. In T. Dean (Ed.) *Proceedings of the 16th International Joint Conference on Artificial Intelligence* (pp. 627-633). San Francisco, Morgan Kaufman.

Aleven, V., McLaren, B., Sewall, J., & Koedinger, K. R. (2006). The Cognitive Tutor Authoring Tools (CTAT): Preliminary Evaluation of Efficiency Gains. *Proceedings of the 8th International Conference on Intelligent Tutoring Systems 2006*. LNCS 4053 (pp 61-70). Berlin, Springer.

Aleven, V., McLaren, B. M., & Sewall, J. (2009). Scaling up programming by demonstration for intelligent tutoring systems development: An open-access web site for middle school mathematics learning. *IEEE Transactions on Learning Technologies*, *2*(2), 64-78.

Aleven, V., McLaren, B. M., Sewall, J., & Koedinger, K. R. (2009). A new paradigm for intelligent tutoring systems: Example-tracing tutors. *International Journal of Artificial Intelligence in Education*, *19*, 105-154.

Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). Cognitive tutors: Lessons learned. *The Journal of the Learning Sciences*, *4*(2), 167-207.

Bloom, B. S. (1984). The 2 sigma problem: The search of methods for group instruction as effective as one-to-one tutoring. *Educational Researcher*, *13*(6), 4-16.

Corbett, A. (2001). Cognitive Computer Tutors: Solving the Two-Sigma Problem. In M. Bauer, P. J. Gmytrasiewicz & J. Vassileva (Eds.): *Procceedings of the 8th International Conference UM2001*, LNAI 2109 (pp. 137-147). London, Springer

Design Science (2011), MathFlow, The MathML Toolbox, http://www.dessci.com/en/products/mathflow/

Koedinger, K., & Corbett, A. (2006). Cognitive Tutors: Technology Bringing Learning Science to the Classroom. In K. Sawyer (Ed.), *The Cambridge Handbook of the Learning Sciences* (pp. 61-78), University Press, Cambridge.

Lenat, D. B., & Guha, R. V. (1990). Building Large Knowledge-Based Systems, Representation and Inference in the Cyc Project. Reading, Massachusetts, Addison-Wesley.

Mellis, E., Andrès, J., Büdenbender, J., Frischauf, A., Goguadze, G., Libbrecht, P., Pollet, M., & Ullrich, C. (2001). A generic and adaptive web-based learning environment. *International Journal of Artificial Intelligence in Education*, *12*, 385-407.

Mizoguchi, R., & Bourdeau, J. (2000). Using ontological engineering to overcome common AI-ED problems. *International Journal of Artificial Intelligence in Education*, *11*, 107-121.

Murray, T. (2003a). Principles for Pedagogy-Oriented Knowledge Based Tutor Authoring Systems. In T. Murray, S. Ainsworth, & S. Blessing (Eds.) *Authoring Tools for Advanced Technology Learning Environments* (pp. 439-466). Netherlands, Kluwer Academic Publishers.

Murray, T. (2003b). An Overview of Intelligent Tutoring System Authoring Tools: Updated Analysis of the State of the Art. In T. Murray, S. Ainsworth and S. Blessing (Eds.). *Authoring Tools for Advanced Technology Learning Environments*, (pp. 491-544). Netherlands, Kluwer Academic Publishers.

Nicaud, J.-F., Bouhineau, D., & Chaachoua, H. (2004). Mixing microworld and CAS features in building computer systems that help students learn algebra. *International Journal of Computers for Mathematical Learning*, *9*(2), 169-211.

Razzaq, L., Feng, M., Nuzzo-Jones, G., Heffernan, N. T., Koedinger, K. R, Junker, B., Ritter, S., Knight, A., Aniszczyk, C., Choksey, S., Livak, T., Mercado, E., Turner, T. E., Upalekar, R., Walonoski, J. A., Macasek, M. A., & Rasmussen, K. P. (2005). The Assistment Project: Blending Assessment and Assisting. In C.K. Looi, G. McCalla, B. Bredeweg and J. Breuker.(Eds.). *Proceedings of the 12th International Conference on Artificial Intelligence in Education*, Frontiers in Artificial Intelligence and Applications 125 (pp. 555-562). Netherlands: IOS Press.

Ritter, S., Kulikowich, J., Lei, P., McGuire, C., & Morgan, P. (2007). *What evidence matters? A randomized field trial of Cognitive Tutor® Algebra I*, In T. Hirashima, H. U. Hoppe, & S. Shwu-Ching Young (Eds.), *Supporting learning flow through integrative technologies* (pp. 13-20) Netherlands: IOS Press.

Sklavakis, D., & Refanidis, I. (2008). An Individualized Web-Based Algebra Tutor Based on Dynamic Deep Model-Tracing. In *Proceedings of the 5th Hellenic Conference on Artificial Intelligence* (SETN '08), LNAI 5138 (pp. 389-394). Berlin: Springer.

Sklavakis, D., & Refanidis, I. (2010). Ontology-Based Authoring of Intelligent Model-Tracing Math Tutors. In *Proceedings of the 14th International Conference on Artificial Intelligence AIMSA 2010*, LNAI 6304 (pp. 201-210). Berlin: Springer.

Sklavakis, D., & Refanidis, I. (2011). The MATHESIS Semantic Authoring Framework: Ontology-Driven Knowledge Engineering for ITS Authoring. In *Proceedings of the 15th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems KES 2011*, LNAI 6882 (pp. 114-123). Berlin: Springer.

VanLehn, K., Lynch, C., Schulze, K., Shapiro, J., & Shelby, R. (2005). The Andes physics tutoring system: Lessons learned. *International Journal of Artificial Intelligence in Education*, *15*(3), 147-204.

VanLehn, K. (2006). The Behavior of Tutoring Systems. *International Journal of Artificial Intelligence in Education*, *16*(3), 227-265.