

Knowledge Based Mechanisms for Tutoring Systems in Science and Engineering

C.W. Liew
Dept of Computer Science
Lafayette College,
Easton, PA 18042
liew@cs.lafayette.edu

Joel A. Shapiro
Dept of Physics & Astronomy
Rutgers University
Piscataway NJ 08854

D. E. Smith
Dept of Computer Science
Rutgers University
Piscataway NJ 08854

Abstract

In science and engineering courses, students are often presented a situation for which they are asked to identify the relevant principles and to instantiate them as a set of equations. For an ITS to determine the correctness and relevance of the student's answer and generate effective feedback, it must map the student variables and equations onto the physical properties and concepts that are relevant to the situation. The space of possible mappings of variables and equations is extremely large. Domain independent techniques by themselves are unable to overcome the complexity hurdles.

This paper describes how an ITS can use constraint propagation and algebraic techniques combined with domain and problem-specific knowledge to solve the mapping problem with systems of algebraic equations. The techniques described in this paper have been implemented in the PHYSICS_TUTOR tutoring system and evaluated on three data sets that contain submissions from students in several introductory Physics courses.

1 Introduction

An ITS designed to help students based on their submissions to an assignment requires a correct interpretation of the elements of their explication. In science and engineering courses, students are often asked to interpret a situation, identify the relevant scientific principles, and instantiate them as a set of equations in variables representing the relevant quantities. The ITS needs to determine the correctness, relevance and sufficiency of these equations, which requires mapping the student-chosen variable names and her equations into the known variables and equations of a known correct solution, or exemplar.

This paper describes our work on PHYSICS_TUTOR, a tutoring system designed to accept unconstrained student submissions, in the form of equations, to problems in an

introductory physics course. We describe how constraint propagation based on dimensional analysis and heuristic knowledge of common naming conventions can be used to determine the dimensionality of student variables. This inferred knowledge is then used to find possible maps from the student variables and equations into those of an exemplar. Since there are numerous correct solutions to each problem the correct solution cannot be represented by a single exemplar. Our system specifies a single complete and correct set of equations, called the *principal exemplar*, from which it can generate other sets of correct equations, exemplars, each a full description of the physical system. Domain and instance specific knowledge are used to guide the generation of correct solutions that are similar to the student's submission. The inferred information is used to identify the map and exemplar which best represents the student input. The remaining differences can then be used to point to incorrect equations and to identify possible conceptual errors that led to mistakes. The efficacy of the techniques has been evaluated on three data sets that contain submissions from students in several introductory Physics courses.

2 Tutoring for Science and Engineering

The domains of the physical sciences and engineering use sets of algebraic equations to specify the interrelations among physical quantities. One of the main differences between purely algebraic equations and algebraic equations describing a relationship in the physical sciences is that the variables and numeric quantities in the latter refer to physical properties that have dimensions and other semantics associated with them. For example, only quantities of the same dimensionality can be added to each other, while quantities such as masses cannot have negative values.

Thus an ITS for engineering and science should use a basic understanding of the physical concepts to parse, interpret, and determine the correctness of a student submission. Student submissions in these fields have some characteristics that set them apart from other problem domains. These

properties are:

- **restricted input language:** In most cases, students provide the answers either numerically (the most common case) or in the form of a system of algebraic equations. Equations are easier to parse than natural language, though there have been efforts to extend tutoring systems to accept natural language input[3].
- **disconnect between description and concept languages:** Students are asked to describe the interactions in a physical system through the specification and manipulation of algebraic equations. Answers are requested in one of several forms including the numeric value of the quantity requested, an algebraic expression for that quantity, or a set of algebraic equations that can be solved for that quantity. From the answer, a tutor must determine if the student has grasped the concepts underlying the system. If not, it will generate feedback, preferably in terms of the concepts and not the algebraic equations.
- **large solution space:** There are numerous correct ways of specifying the answer.

Most professors would say that the principal measure of student learning in an introductory course in science and engineering is the ability to take a verbal description of a physical situation and translate it into a set of variables representing the relevant physical quantities, together with a set of equations representing their knowledge of the principles that relate to the situation. Usually students are asked to solve numerically for a particular unknown quantity in terms of the given values of other physical quantities.

This is difficult for many students and is often simplified by providing variable names for the relevant quantities in the problem statement and walking the student through a succession of steps involving a single physics principle at each step. Many homework assignments and many computer-based homework systems [11, 8, 9] revolve around such an approach. Not surprisingly, students learn to effectively use this approach, but unfortunately this does not help the student gain a fuller understanding of the domain and the relevant physics principles. This approach, termed plug-and-chug, is often carried out by students as follows:

- List the given variables and the one asked for (the “sought”).
- Search for a formula that contains only, and probably all, of these variables.
- Plug the given values in the equation (“plug”).
- Solve the equation for the sought variable (“chug”).

Plug-and-chug problems are easy on both the student and any computer-based system that grades or tutors them, and they do provide exercise and familiarity with the canonical formulae. But they do not encourage, and may discourage, a better approach to understanding the domain, that requires a deeper conceptual analysis of a possibly complex situation, with multiple principles (and equations) required.

Why are plug-and-chug problems so common if they are not good at encouraging the attitude and learning that we would like to instill? Primarily because it is useless to assign more challenging problems if the majority of the students will get stuck early on and be unable to proceed further without help. If, however, help can be provided conveniently at every point, such problems can be assigned, and the student may learn to analyze complex situations and develop the analytical abilities we seek.

Thus the problem we address in creating an ITS for science and engineering is this: A problem is presented to the student. The tutor is presented with a solution in the form of an exemplar set of equations, expressed in canonical variables of known type, in particular having the appropriate dimensions (units). The exemplar set is expected to be as close to the basic physics principles as possible. The tutor is also presented a set of equations by the student, using her own set of variables. The set of equations and variables is likely to be smaller than the canonical set, because many of the canonical equations can be simply used to eliminate variables intuitively. An explicit example is presented in detail in [6]. The tutoring system should be able to judge the correctness of the student’s equations and the adequacy of the set of equations. It should also offer help in correcting incorrect equations or in finding necessary ones that are missing.

Thus the basic task for an ideal ITS for engineering and the physical sciences can be stated as follows:

Given:

- a description of a physical system,
- a student’s set of algebraic equations that describes the physical system,
- a complete and correct set of equations, the *principal exemplar* (the ITS is either provided the principal exemplar or a means to generate it), and a means of generating from it other sets of correct equations *exemplars* that describes the physical system,

Goal:

Analyze the student’s system of algebraic equations and determine if it is¹

¹In this analysis, we ignore redundant equations. The system will eliminate duplicate equations from consideration.

- **correct and complete:** all equations in the student's submission are correct and are sufficient to describe the interactions of the physical system.
- **correct and incomplete:** all equations in the student's submission are correct but are insufficient to describe the physical system (*i.e.*, some necessary equations are missing).
- **incorrect and complete:** one or more of the equations in the student's submission are incorrect. If the incorrect equations are corrected they are sufficient to describe the physical system.
- **incorrect and incomplete:** one or more the equations in the student's submission are incorrect. If the incorrect equations are corrected they are insufficient to describe the physical system.

If the submitted system of equations is either incorrect or incomplete, then the ITS must further determine:

- the equations that are missing and necessary to complete the description of the physical system.
- the equations that are incorrect and why they are incorrect.

3 Dimensional Constraints on Variables and Numeric Quantities

Until an ITS has identified the physical meaning of a student's variables, it can judge only trivial aspects of correctness, such as whether the expressions are algebraically well-formed. One of the main differences between purely algebraic equations and algebraic equations describing a relationship in a physical domain is that the latter must be dimensionally consistent.

For example, Atwood's machine (*i.e.*, two masses suspended from a single massless, frictionless pulley), a commonly used source of problems in introductory physics, has a correct solution represented by the equations of Figure 1.

$$T_1 - m_1 * g = m_1 * a_1 \quad (1)$$

$$T_2 - m_2 * g = m_2 * a_2 \quad (2)$$

$$T_1 = T_2 \quad (3)$$

$$a_1 = -a_2 \quad (4)$$

Figure 1. Equation Set for Atwood's Machine

In this set of equations, each of the variables, constants, terms, expressions, and even equations must have specific

dimensions. Further they can only be combined using dimensionally consistent operations. Equations 1, 2 and 3 have the dimensions of force ($\text{kg} \cdot \text{m}/\text{s}^2$) while equation 4 has dimensions of acceleration (m/s^2). The rules of algebra would allow adding Eq. 3 and 4, but that would be incorrect physically, since that operation would violate dimensional consistency.

These dimensional constraints on algebraic operations are taught in introductory courses, and have even been incorporated in some programming languages[2, 1] as part of variable declarations. Tutoring systems that predefine the variables, and those [12] that require the students to explicitly identify their variables, use *a priori* knowledge to determine the variable dimensionality.

3.1 Dimension Inferencing Using Constraint Propagation

Students are allowed, and in fact should be encouraged, to use variables and equations of their own choosing. They can use more variables than found in the exemplar, fewer than found in the exemplar, and name their variables differently than those used in the exemplar. Similarly, students are not restricted to use the same set of equations as found in the exemplar.

Decoupling students solutions from the specific details of the exemplar encourages students to think about the problem in their own terms and discourages a plug and chug approach. But this decoupling comes with a cost. The student submission does not provide dimensionality information to the PHYSICS.TUTOR system and thus the system must infer what each equation and variable represents.

The system begins by assigning a set of likely dimensions to each variable based on typical naming conventions used in the field. It then constructs a constraint graph for the equation where variables and numeric quantities in the equation are instantiated as leaf nodes and operators (*e.g.*, $+$, $-$, $*$, $/$, $=$) and functions (*e.g.*, \cos , \sin , \tan) are instantiated as internal nodes. Finally it applies all constraints until a final assignment of dimensions is found or an inconsistency is detected. Constraints are instantiated for the following types of conditions:

- terms that are added, subtracted or set equal must have the same dimensions (*i.e.*, equality constraints)
- multiple instances of a variable must have the same dimensions (*i.e.*, identity constraints)
- the dimensions of the result of a multiplicative operator is the sum or difference of the dimensions of its operands.

The initial dimensions assigned to each variable are critical to the success of this method. The PHYSICS.TUTOR

system uses a dictionary that contains information about variable prefixes and the concepts to which they usually refer in Physics. For example, a variable beginning with T is likely to represent a time, a force (tension), an energy (kinetic), a temperature, or a distance (thickness), but very unlikely to represent a velocity or frequency or angle. These lists represent domain knowledge about physics, and can be further refined if the problem sub-domain(s) is known. For example, if a problem set involves only classical mechanics, we can exclude temperature (and probably thickness) from the list of possibilities.

3.2 Validation of Dimensionality Inference

The constraint propagation technique used in the PHYSICS_TUTOR system effectively determines the dimensions of the variables, numeric quantities and equations using a dictionary of commonly used prefixes. The effectiveness of the technique was evaluated on several data sets, two of which were obtained from the repository of ANDES data. The ANDES system [10] is a physics ITS currently in use at the United States Naval Academy. Student answers were extracted and used to evaluate the PHYSICS_TUTOR system. The three data sets were:

1. **ANDES** (Fall 2000 and Fall 2001): These two data sets were characterized by:

- **large number of student generated equation sets** — There were approximately 10,000 equation sets in each of the semesters. We only extracted correct equations and thereby limited our evaluation to equations sets that were correct but possibly incomplete and could contain irrelevant equations.
- **a large number of problem types** — The ANDES system has a repository of approximately one hundred different problems.
- **constrained input** — ANDES required students to define variables before they were used. This information was not used by the PHYSICS_TUTOR system but could have affected how students entered their answers.

2. **Lafayette**: This data set consists of approximately 350 answers to four similar problems from 88 students in an introductory physics course. There were no restrictions on the input. Students wrote their answers on paper that were later transcribed into electronic form.

The system was most effective when the dictionary was limited to knowledge of variables used in Newtonian mechanics rather than all of Physics. In 90% to 95% of all

submissions the PHYSICS_TUTOR was able to either (1) determine the dimensionality of all the variables and numeric quantities or (2) determine that the equations were dimensionally inconsistent. These results show that constraint propagation techniques augmented with domain knowledge can be effective at determining the dimensions of the components of systems of algebraic equations representing physical systems. Further details about this technique can be found in [4, 5, 6].

4 Mapping Equations and Variables

In the previous section, we described how constraint propagation along with domain knowledge is used to determine the dimensions of the variables in systems of algebraic equations. Determining the correctness of the submission, however, requires mapping the equations, variables and numeric quantities to the specific physical quantities and concepts of the problem, as represented by variables and equations in a known correct solution, an exemplar. The ITS needs to find the mappings between:

- variables in the student submission and the corresponding variables in the exemplar,
- numeric quantities in the student submission and either corresponding variables or numeric quantities from the exemplar,
- equations in the student submission and the corresponding equations in the exemplar.

The ITS can then use these maps to determine the correctness and relevance of the student's answer. If some part of the student's submission is incorrect, the maps can also be used to isolate the incorrect equations for further analysis and tutoring.

4.1 Computational Issues in Mapping

As previously noted there are numerous correct solutions to each problem. These are represented as a single principal exemplar together with a procedure for generating from it other exemplars, sets of correct equations, each of which describes the physical system. We assume that if the student's submission is correct there will be a 1-1 mapping of each variable and equation in the student's submission to one of the correct solutions generated by PHYSICS_TUTOR². But which one? Computationally we decompose the problem into two phases: a) identifying a correct solution to be

²It is possible a correct solution will not be mappable to any of the solutions generated by PHYSICS_TUTOR. This occurs infrequently; additional work is needed to handle such case

mapped to the student's submission and b) determining if a correct solution is mappable to the student's submission.

Identifying correct solutions that are likely to be mappable to the student's submission uses heuristics based on the number of equations and number of variables. Identifying possible maps between a correct solution set and the submission uses insights from graph isomorphism. The heuristics have proven to reduce what is seemingly an intractably large search space (*i.e.*, exponential in both the number of variables and number of equations) to one that can be searched effectively. These heuristics and examples of the reductions they provide are described in sections 4.1.1, 4.1.2, and 4.2.

The tutor's task must address the following issues to identify a map from the student submission to a correct solution. Further details about these issues can be found in [7].

1. **variable aliasing:** Students often use a single variable to represent either (1) groups of variables that have the same magnitude, *e.g.*, T to represent T_1 and T_2 if $T_1 = T_2$, or (2) a product of variables, *e.g.*, F where $F = m_1 * a_1$.
2. **substitution of numeric quantities for variables:** Most problems in introductory physics involve numeric quantities. Students may interchangeably use numeric quantities and variables, *e.g.*, the use of g or 9.8 m/s^2 to represent the acceleration due to gravity.
3. **algebraic simplification of numeric quantities:** For example, the use of 19.6 instead of $2 * 9.8$, or the removal of a common factor from equations.

4.1.1 Generating Mappable Correct Solutions

The dimensional consistency checks that were described earlier ensure that the ITS only need focus on reasoning about dimensionally consistent equations. The information inferred from the dimension checks includes the dimensions of the variables, numeric quantities and equations in the student's submission and in the exemplar (*i.e.*, problem and instance specific knowledge). The exemplar and the student's submission may differ in (1) the number of equations, (2) the number of variables, (3) the names of the variables and (4) the numeric quantities present. Generating all solution sets that are consistent with and equivalent to the exemplar would require using:

- all possible variable substitutions.
- all possible substitutions of numeric quantities for variables.
- all possible numeric simplifications.

This would be prohibitively expensive.

The PHYSICS_TUTOR takes the following steps to generate the "exemplar lattice" of most likely equivalent sets of equations:

1. transform each equation into a canonical sum of products
2. systematically reduce the number of variables (and equations) by introducing variable substitutions one at a time, generating a new exemplar with each substitution. Repeat this step for each set until the resultant set contains only a single equation. For computational reasons, the PHYSICS_TUTOR system only uses the most likely variable substitutions.

This procedure generates the *exemplar lattice*, the nodes of which are sets of equations derived from the principal exemplar through correctness-preserving algebraic transformations. Each node in the lattice (*exemplar*) differs from the others in one or more of the following ways: a) the number of equations and variables, b) the dimensions of the equations, c) the dimensions of the variables, or d) how the variables are used in the equations. Once the lattice is built, it is searched to find the exemplar that best matches the student submission. The ITS compares the student's submission with every node in the lattice that has the same numbers of variables and equations of each dimensionality.

4.1.2 The Mapping Algorithm

Once a candidate exemplar has been identified it must be mapped to the student's submission. Before the mapping is undertaken a pre-processing step that replaces numeric quantities in the student's equations with variables or a combination of variables is performed. This facilitates the mapping of submission into the exemplar and requires that the system be able to recognize common combinations of numeric quantities (*i.e.*, multiplication or division by 2) and replace them with the appropriate variables. For example, given the velocities of two objects when only the relative velocity is relevant, students frequently use the numeric value of the relative velocity. The pre-processing step determines that this numeric value is in fact $v_1 - v_2$ and replaces it with this expression.

As noted earlier, establishing a mapping between the submission and correct equation sets has great similarity to determining if two graphs are isomorphic. For each of the two equation sets for which a map is desired a bipartite graph can be constructed with edges between variables and equations if a variable is used in an equation. If the graphs are isomorphic there is a 1-1 mapping of equations to equations and variable to variables.

Our algorithm uses an adjacency matrix to represent the bipartite graph. The rows in this matrix correspond to the

equations, the columns to the variables, and the boolean entries to whether a given variable occurs in a given equation. For example, the exemplar equations given in Figure 1 for Atwood's machine is shown in Table 4.1.2.

Table 1. Example Matrix for Atwood's Machine

	T_1	T_2	m_1	m_2	a_1	a_2	g
Eq. 1:	1	0	1	0	1	0	1
Eq. 2:	0	1	0	1	0	1	1
Eq. 3:	1	1	0	0	0	0	0
Eq. 4:	0	0	0	0	1	1	0

If we look for a match between the graph of a student's set of N_e equations in N_v variables to an exemplar graph with the same number of each, the most general mapping of nodes could have $(N_e + N_v)!$ mappings. However, since we know the graph is bipartite (*i.e.*, variables can only map to variables and equations to equations) the number of possible mappings can be reduced to $N_e! \cdot N_v!$, corresponding to permutations of the rows of the matrix times permutations of the columns. The number of permutations that must be examined is next reduced by labeling each equation and variable with its dimensionality, and considering permutations only with variables (or equations) of the same label. The label of each node can then be extended by giving the number of nodes with each label to which this node is connected. This is a recursive procedure. For example, labeling in the Atwood's machine example first partitions nodes into equations and variables and then partitions equations and variables, based on dimensionality. It continues by separating g from the a_i 's based on the fact that g appears in two equations both with the dimensions of force but the a_i 's appear in one equation with a dimension of force and in another with a dimension of acceleration.

The fixed point of this iterative labeling for Atwood's machine is indicated in Table 4.1.2 by the vertical and horizontal lines that partition the adjacency matrix into boxes. A valid mapping cannot permute elements from different boxes. Equation 3 cannot be permuted with any other equation, nor can g be permuted with a_i . For the example presented, the search space is reduced as follows:

Labeling	Permutation Bound	Value
Uninformed	$11!$	39,916,800
Bipartite	$(7!) \cdot (4!)$	120,960
Dimension	$(2! \cdot 2! \cdot 3!) \cdot (3! \cdot 1!)$	144
Fixed Point	$(2! \cdot 2! \cdot 2! \cdot 1!) \cdot (2! \cdot 1! \cdot 1!)$	16

This leaves $(2!)^4 = 16$ permutations to try on the student equation matrix to see if it matches the exemplar's. Such a

match may not be unique, however, because there may be a permutation which leaves the matrix invariant. In our case, there is one nontrivial permutation $T_1 \leftrightarrow T_2$, $m_1 \leftrightarrow m_2$, $a_1 \leftrightarrow a_2$, Eq. 1 \leftrightarrow Eq. 2 which does this, so there are two possible mappings of the student variables which need to be examined further.

The reduction in complexity provided by the algorithm is even more dramatic when the exemplar set is bigger. In fact, the four equations given to describe Atwood's machine is oversimplified if we are to anticipate all reasonable student variables. Students might well use variables describing the two weights, W_1 and W_2 , and the net forces on each object, F_1 and F_2 . This introduces four new equations and four new variables, all with dimensions of force. Of the 1.6 trillion possible permutations (*i.e.*, $8! \cdot 11!$) for this expanded bipartite graph, not permuting variables and equations with different dimensions reduces the possible permutations to 4.4 million (*i.e.*, $7! \cdot 1! \cdot 6! \cdot 2! \cdot 3!$). Recursively refining the labels (three times) as described previously reduces the complexity to $(2!)^3(1!)^2 = 8$ equation permutations and $(2!)^5 = 32$ variable permutations for a total of 256 possible mappings for the full set of equations.

This discussion addressed the isomorphism of two equation sets, but there are many correct ways a student can specify a system of equations that describes a physical situation. Thus, we must search our exemplar lattice for an exemplar with a graph that is isomorphic to the student's submission. Finding such a *dimensionally matching* exemplar does not mean that the student's submission is correct. First we need to check that variables occur correctly in each term of each expression. If we find such a *term-by-term* match, a further comparison is made to check (1) if there are sign mistakes, (2) the coefficients of each term and (3) the functions used in each term. A measure of difference is generated which is then used to judge which term-by-term matching exemplar provides the best match. Experimental evaluation (Section 4.4) has shown that application of the algorithm results in the correct match even when there are multiple nodes that match the student submission.

This algorithm does not find a map for all correct equations. For example, it is unable to find a map when the student simplifies the equation by removing common factors. The number of failures of this type are few and can be reduced further by generating equation sets with common factors removed and adding these as nodes to the lattice of complete and correct solutions. This approach does not allow all submissions with one or more common factors removed to be matched since students can remove some, and not all, common factors.

4.2 Mapping Against Incorrect Equations and Incomplete Equation Sets

The previous section described the process by which the PHYSICS_TUTOR system builds a map from the variables and equations of the student's submission to the corresponding entities in a known correct solution. This works very well when the student's submission is correct and complete. But when the student's submission is incomplete or no term-by-term match is found, the PHYSICS_TUTOR expands the student and exemplar lattice, greatly increasing the computational cost, in an attempt to identify as many correct equations as possible in the student's submission.

To identify these equations, the algorithm constructs, for each exemplar, another lattice of equation sets (called *sub-exemplars*) in which nodes at each level of the *sub-lattice* contain one less equation than its parent node. Therefore a sub-exemplar with five equations will lead to five sub-exemplars, each an incomplete set of four correct equations. We construct a similar lattice for the student's submission. The algorithm then uses the matching procedure described in the previous section to find the best term-by-term match from all the newly generated incomplete equation sets to a node in the student's lattice. Since the goal is to maximize the number of matching equations, the algorithm starts with the node at the top of the student's lattice that contains the entire student submission. If that fails, the algorithm then tries to find the best match for the nodes at the next level in the student's lattice. This procedure iterates, moving down the lattice, until either a match is found or all newly generated equation sets are visited.

If all newly generated equation sets are visited and no term-by-term match is found the algorithm has failed to map ("understand") any part of the student's submission. If a term-by-term match is found, the *matched* subset of student's equations is analyzed to determine if there are sign, coefficient, or function errors (see Section 4.1.2) and the *unmatched* subset of student equations is compared to the unmatched subset of exemplar equations. Heuristics are used to determine if the following types of mistakes could transform one of the unmatched exemplar equations to one of the unmatched student equations.

- **omission of a term:** where the student's equation has omitted one or more terms.
- **addition of an extraneous term:** where the student's equation contains extra terms.

4.3 Generating Feedback

The errors that the algorithm can detect are algebraic in nature due to the fact that both the submission and the so-

lution are sets of algebraic equations. The algebraic errors that are currently detected are:

1. **dimensional inconsistency:** The algorithm is frequently able to isolate the inconsistency to one or two terms in an equation.
2. **incorrect coefficients**
3. **incorrect operators:** The equation has passed the dimensional consistency check and the term-by-term matching check. But these will not detect an incorrect choice an addition operator, *i.e.*, a + for a - and vice versa.
4. **incorrect functions:** A dimensionally matched equation can also be wrong by using the incorrect function that has the same dimensional signature (dimensions of the input parameters and the result) as the correct function. This frequently happens with trigonometric functions.
5. **missing or extra terms:** These can be the result of either an arithmetic or a conceptual error.
6. **missing equations:** these are equations in the exemplar that cannot be mapped to any equations in the student's submission.
7. **extra equations:** these are equations in the student's submission that cannot be mapped to any equations in the solution.

¿From a tutoring standpoint, it is desirable to discuss the student's errors in terms of the conceptual errors that were made rather than the algebraic errors. This can be difficult to determine because different conceptual errors can result in the same algebraic error. For example, one common error that is found when analyzing an equation describing conservation of momentum is a missing term, *i.e.*, the student has neglected the contribution from one of the objects in the system. This can be due to an oversight on the student's part or that the student has not correctly grasped the concept of a system. In other problems, a missing term is frequently caused by arithmetic or algebraic errors. Domain knowledge and heuristics are necessary if the tutoring system is to generate feedback beyond the level of algebraic errors.

4.4 Validation of Variable and Equation Mappings

Experiments were run to evaluate the effectiveness of the mapping techniques. They were run against much smaller data sets because the PHYSICS_TUTOR system does not currently handle problems that use trigonometric functions.

The first data set consisted of answers to the Atwood's problem that were collected from 88 students in an introductory physics course. The students were not constrained in any way. The students' answers were evaluated against an exemplar with 8 equations. Eight of the submissions had dimension errors or were dimensionally ambiguous and were not considered. The PHYSICS_TUTOR system was able to correctly classify all but 11 of the rest of the submissions. In six submissions, the system correctly identified extraneous equations.

The second data set consisted of data extracted from the ANDES logs of Fall 2001. Three problems were selected that (1) had a large number of submissions and (2) did not require the use of trigonometric functions. The three problems (Exkt3a, Exkt4a and Exe5a) had 234, 260 and 145 submissions respectively. The results from the evaluation are shown in Table 2. The system failed to map some submissions either because (1) the system did not have the algorithmic transformations that would generate the equivalent set from the exemplar or (2) the student used some combination of numeric quantities that the system could not identify.

Table 2. Results from ANDES Fall 2001 data

	Exkt3a	Exkt4a	Exe5a
Successful analysis	89.8%	96.2%	93.2%
a) Correct Solution	35%	45.4%	81.4%
b) Partial Solution	47%	38.1%	7.6%
c) No nontrivial equations	7.3%	12.7%	4.1%
Unsuccessful analysis	10.2%	3.8%	6.8%
d) Non-derived equations	2.5%	0%	2.7%
e) Unknown constants	7.7%	3.8%	4.1%

5 Conclusion

This paper has shown how domain and problem specific knowledge has been used to solve a complex problem, that of determining the correspondence between the variables, numeric quantities and equations in a student's submission and the physical quantities and concepts that apply to a problem. Without the domain knowledge, the constraint propagation techniques that were used would not be able to either (1) solve the problem or (2) solve it within a reasonable amount of time. Domain knowledge about the role of dimensions in Physics and the vocabulary by which variables are assigned dimensions (concepts) based on the first letter of the variable play a very important role. The domain knowledge helped the PHYSICS_TUTOR system to identify the dimensions of the variables, numeric quantities and equations in the students' submissions. The derived instance-specific knowledge was then used in conjunction

with constraint propagation to find the best fitting maps (correspondence) between the student's submissions and a correct solution. The efficacy of these techniques was evaluated on data sets of submissions obtained from students enrolled in several introductory Physics courses.

Acknowledgments

We are grateful to Kurt VanLehn and the Andes group for making the ANDES logs available to us, and to Anders Weinstein for answering questions about the ANDES implementation. Ryan McCall, Jim Appenzeller and Dave Santin provided invaluable help in the implementation.

References

- [1] R. Cunis. A package for handling units of measure in lisp. *ACM Lisp Pointers*, 5(2), 1992.
- [2] P. N. Hilfinger. An ada package for dimensional analysis. *ACM Transactions on Programming Languages and Systems*, 10(2):189–203, 1988.
- [3] R. Kumar, C. P. Rosé, V. Alevén, V. Iglesias, and A. Robinson. Evaluating the effectiveness of tutorial dialogue instruction in an exploratory learning context. In *Proceedings of Intelligent Tutoring Systems Conference*, 2006.
- [4] C. Liew, J. A. Shapiro, and D. Smith. Identification of variables in model tracing tutors. In *Proceedings of 11th International Conference on AI in Education*, pages 464–466. IOS Press, 2003.
- [5] C. Liew, J. A. Shapiro, and D. Smith. Inferring the context for evaluating physics algebraic equations when the scaffolding is removed. In *Proceedings of Seventeenth International Florida AI Research Society Conference*, 2004.
- [6] C. Liew, J. A. Shapiro, and D. Smith. Determining the dimensions of variables in physics algebraic equations. *International Journal of Artificial Intelligence Tools*, 14(1&2):25–42, 2005.
- [7] C. W. Liew and D. E. Smith. Reasoning about systems of physics equations. In Cerri, Gouarderes, and Paraguacu, editors, *Intelligent Tutoring Systems*, 2002.
- [8] <http://www.masteringphysics.com/>.
- [9] D. E. Pritchard and E.-S. Morote. Reliable assessment with cybertutor, a web-based homework tutor. In *World Conference on E-learning in Corporate, Government, HealthCare, and Higher Education*, Montreal, Canada, 2002. E-Learn.
- [10] K. Schulze, R. Shelby, D. Treacy, M. Wintersgill, K. VanLehn, and A. Gertner. Andes: An intelligent tutor for classical physics. *The Journal of Electronic Publishing*, 6(1), 2000.
- [11] A. P. Titus, L. W. Martin, and R. J. Beichner. Web-based testing in physics education: Methods and opportunities. *Computers in Physics*, 12, #2, 1998.
- [12] K. VanLehn, C. Lynch, L. Taylor, A. Weinstein, R. Shelby, K. Schulze, and M. Wintersgill. Minimally invasive tutoring of complex physics problem solving. In Cerri, Gouarderes, and Paraguacu, editors, *Intelligent Tutoring Systems*, pages 367–376, 2002.