Removing the Scaffolding: First Steps

C.W. Liew
Department of Computer Science
Lafayette College
Easton, PA 18042
1+(610)330.5537
liew@cs.lafayette.edu

Joel A. Shapiro
Department of Physics and Astronomy
Rutgers University
Piscataway, NJ 08854-8019
1+(732)445.3886

D.E. Smith
Department of Computer Science
Rutgers University
Piscataway, NJ 08854-8019
1+(732)445.3628

Topic: Intelligent Tutoring and scaffolding
Subtopic: Special Application Fields - Physics (College Level)

Abstract.

This paper describes our continuing work on developing tutoring systems where the scaffolding may be relaxed. This requires the tutoring system to determine more of the context of answers. In particular, we examine issues of identification of the meaning of variables in systems of equations that solve college level introductory physics problems. In earlier work we developed techniques that worked for a small set of problems and evaluated them on a small number of students. The work described here covers the extension to and evaluation of a much larger class of problems and a larger number of students. The results show that our technique uniquely determines the dimensions of all the variables in 83% of the sets of equations. By asking the student for dimension information about one variable, an additional 5% of the sets can be determined. Thus a physics tutoring system can use this technique to reason about a student's answers even when the scaffolding and context are removed.

1 Introduction

In teaching problem solving, Intelligent Tutoring Systems (ITS) often employ a rigid and explicit framework to guide the student along a definite sequence of steps. This mechanism called "scaffolding" is pedagogically sound and is beneficial to beginning students in the subject, because it helps them go through each step in detail. After some experience, students internalize and combine some of these steps, and a human tutor would not require the student to explicitly demonstrate the most basic steps. At some point, the scaffolding should be removed from the tutoring system.

Removing the scaffolding puts a greater burden on both the student and the tutoring system. The student must do more on their own without guidance from the tutor and the system must now interpret answers that may be in a different sequence or may have incorporated some basic assumptions.

This paper describes our continuing work on developing tutoring systems where the scaffolding is relaxed. In particular, we examine issues of identification of the meaning of variables in equation sets that solve college level introductory physics problems. In earlier work we developed techniques that worked for a small set of problems and evaluated them on a small number of students. The work described here covers the extension to and evaluation of a much larger class of problems and a larger number of students. The results show that the technique uniquely determines the dimensions of all the variables in 83% of the sets of equations. By asking for dimension information about one variable, an additional 5% of the sets can be determined.

2 Algebraic Physics Problems

Physics uses sets of algebraic equations to specify the interactions of a system of objects. One of the main differences between *generic* algebraic equations and algebraic equations describing a relationship in physics is that the latter must be dimensionally consistent. Two algebraic equations in physics are shown below.

$$T - m_1 * g = m_1 * a \tag{1}$$

$$a_1 = -a_2 \tag{2}$$

Algebraically speaking, these equations could be added to one another to form a new equation. However in physics, each of the variables, constants, terms, expressions, and even equations must have dimensions. Further they can only be combined using dimensionally consistent operations. Equation 1 is likely to have the dimensions of force ($\mathrm{kg} \cdot \mathrm{m/s^2}$) while equation 2 would have dimensions of acceleration ($\mathrm{m/s^2}$). It would thus be incorrect to add these equations since that operation would violate dimensional consistency. Physically speaking, the variables represent physical properties of an object or a system of objects and the equations describe the constraints between these quantities given by the laws of physics.

2.1 Issues in Removing the Scaffolding

Removing the scaffolding imposes an additional computational requirement on tutoring systems. We illustrate this with an example problem based on Atwood's machine, a pulley with two masses, m_1 and m_2 hanging at either end, as shown in Figure 1.

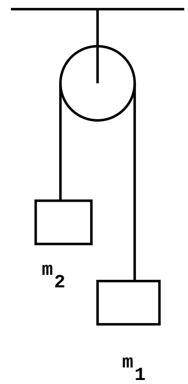

Figure 1: Atwoods Machine

A common problem based on Atwood's machine asks the student for the equation(s) that would determine the acceleration of the mass m_1, assuming that m_1 and m_2 are not equal. Equations 3 through 6 represent a solution to the problem.

$$T_1 - m_1 * g = m_1 * a_1 \quad (3)$$
$$T_2 - m_2 * g = m_2 * a_2 \quad (4)$$
$$T_1 = T_2 \quad (5)$$
$$a_1 = -a_2 \quad (6)$$

From a pedagogical standpoint, physics instructors teach beginning students that the steps involved in solving problems of this type are:

1. *variable definition:* Each variable is defined as to which object(s) and property it refers to. In some cases, the time period when this variable is applicable is also defined.

2. *identification of physics laws:* Each applicable physics law, e.g., force balance or conservation of momentum, must be identified and the objects that they apply to must be specified.

3. *instantiation of physics laws:* The general physics laws are stated as equations with general variables. Each specific variable specified from the first step is substituted as appropriate. The result is an equation or system of equations relating the interactions between objects or systems of objects in the current problem.

As students become accustomed to the vocabulary of the domain, they start using problem solving "shortcuts". Instead of defining each variable explicitly, the students select from a dictionary of well-known physics variables to represent the properties that they desire. For example in force balance problems, variables beginning with m typically represent masses while variables beginning with an a usually represent accelaration. Thus the naming of a variable implicitly specifies the dimensions or properties. The judicious and consistent selection of subscripts with each variable specifies the object(s) that the variable refers to. For example, m_1 and a_1 would refer to the mass and acceleration of the same object while $p_{1,t1}$ might refer to the momentum of object 1 at time $t1$.

When the scaffolding is removed, the tutoring system must be able to determine the context of the system of equations. For example, the student might choose to use a single variable a to represent acceleration. The resulting system of equations is shown below.

$$\begin{aligned} T_1 - m_1 * g &= m_1 * a && (7)\\ T_2 - m_2 * g &= -m_2 * a && (8)\\ T_1 &= T_2 && (9) \end{aligned}$$

The tutor must determine that (1) the variable a has the dimensions of acceleration ($\mathrm{kg} \cdot \mathrm{m/s^2}$), (2) the single variable is mapped to the acceleration of object 1 and that (3) the accelerations of the other objects are replaced by an algebraic substitution using Eq. 6. In this paper, we focus on the first step, that of determining the dimensions of each variable. Our preliminary work in addressing the second step, that of mapping the variables to objects is described in [6].

3 Prior Work

Checking for dimensional consistency is a important first step for a physics tutoring system as it can then focus on reasoning about dimensionally correct equations only. Existing systems, e.g. ANDES [2] and PHYSICS-TUTOR [4], require that the dimensions of each variable and constant be known *a priori* either through a knowledge base of variables and constants or by having the student define them. Once these dimensions are known, it is a fairly simple step to determine if the equation is dimensionally consistent by using some form of "dimension mathematics".

There are many systems that use constraint propagation to ensure consistency of values of variables. Examples of such systems include VEXED [9], OPIS [8]. Their use of constraint propagation is similar except that they are propagating values and not dimensions.

There has also been some work done on adding dimension specifications to programming languages to support compile-time [7, 3] and run-time [1] detection of dimension errors. These systems are more like strongly typed programming languages where every variable has to be defined and has a type. Our system is analogous to a weakly typed language where variables are partially defined on first use and their types are inferred from the context.

4 Determination of Dimensions

In an earlier paper [5], we described an approach for determining the dimensions of every variable in an algebraic equation. The earlier version of the technique combined the use of a knowledge base of commonly used physics variables and constants with constraint propagation.

A constraint graph is built where variables in the equation are instantiated as leaf nodes and internal nodes represent operators, e.g., $+, -, *, /, =$, and functions, e.g., cos, sin, tan. The value at each node represents the set of possible dimensions for that node. The knowledge base is used to determine the probable dimensions of each variable. Each entry in the knowledge base consists of a name (a string) and values for each dimension. The initial value for each variable node is determined by matching the names in the knowledge base with the variable. If the name in the knowledge base is a prefix of the variable, then it is considered a match. There may be more than one possible combination for a variable as it may match several names in the knowledge base or a name may have multiple possible values. Constraint propagation is used to propagate dimension information to other terms and literals to (1) infer dimension information and (2) determine dimensional consistency. The algorithm can use partial information about the dimensions of a variable and combine that with knowledge of operators and functions (which are just operators) to completely determine dimensions. In essence knowledge, even incomplete knowledge, propagates from one part of the equation to another. This permits the algorithm to reason about dimensional consistency when the variables are not explicitly defined.

This algorithm was evaluated on roughly 350 answers to four physics problems from 88 different students in an introductory physics course for engineers and science majors. Only 5% of the submitted answers (two to three answers for each problem) were ambiguous and required additional information from the student. The technique was subsequently evaluated on equation sets extracted from the log files of the ANDES system [2].

4.1 The ANDES data

The ANDES system is also a tutoring system for introductory college level physics. It has a much large database of problem types and is in current use at the United States Naval Academy. Logs of student answers and tutor responses have been maintained since the initial introduction of the ANDES system. We extracted the student answers from one semester (Fall 2000) and used it to evaluate our system. The key features of this data set (and of the ANDES system) are:

- *large database of problems and problem types*: The ANDES system has a repository of approximately one hundred problems. These problems are much more diverse than the ones previously analysed.

- *large number of equation sets:* The ANDES data analyzed contained 9,865 equation sets in 6,000 logs. These logs were created by many students each who worked on many problems. The system recorded answers, including partial answers, making the number of equation sets larger than the number of logs. Many of these equation sets contain incomplete answers, i.e., the student has not entered all the equations. Our analysis does not group equations sets by either student or problem but rather treats all 9,865 equations sets as a single corpus.

- *variables are explicitly defined before use.* The ANDES framework requires that the students define all variables before they can be used in equations and provides a graphical interface to help them with this step.

- *use of numeric values:* The questions in ANDES are given in terms of explicit numerical quantities and require numeric answers. While students were strongly encouraged to generate complete algebraic solutions before substituting numeric values to arrive at the answer, students frequently use numeric values in place of variables at earlier stages.

The data from the ANDES logs provides a good evaluation of our technique in several ways that our original experiments did not. These are:

- how general is our technique? how well will it perform on a more diverse set of problems?
- how well will the technique perform on incomplete sets of equations?

4.2 Initial Results

The knowledge base was greatly expanded to handle the larger class of problems. Entries were created for all uses of variable names in physics. Possible dimension values for each variable were determined by matching the beginning of variable names against a single list of well-established prefixes. The initial results showed that the dimensions could be completely determined for a small set of equations (less than 50%). There were many sets of equations that were categorized as consistent but ambiguous, i.e., at least one variable had more than one valid value for dimensons was applicable.

4.3 Analysis and Extensions

Analysis of the results showed problems that were not revealed with the earlier smaller data set. Most equation sets had more than one set of possible dimension assignments for the set of variables. We observed that because we were using possible concepts from all of physics, including electricity and magnetism and modern physics (which were not covered in the Andes problems) the range of choices of dimensionality were often very large and the constraints are often insufficient to uniquely determine the correct choice.

This problem was fixed by (1) splitting the knowledge base into broad subfields of relevance and (2) adding a more powerful matching capability to the knowledge base. The knowledge base was split up into disjoint categories, e.g., Newtonian mechanics, electricity and magnetism, and modern physics, and the ANDES problems were annotated to specify that they were problems in Newtonian mechanics. In addition, instead of just searching for a matching prefix, the knowledge base now supports three types of matches. Each entry into the knowledge base consists of (1) a string, (2) a set of dimensions, (3) category and (4) type of match. The three types of matches are:

- *prefix match:* Any variable name whose beginning matches the string of an entry in the knowledge base will have the associated set of dimensions as a possibility. The variable alp will prefix match with the entry a and will have dimensions associated with acceleration as one of the valid possibilities.
- *pre-emptive match:* Any variable name whose prefix matches the string of an entry in the knowledge base will pre-empt any prefix matches. The variable $alpha1$ will pre-emptively match with $alpha$ and have radians as one of the possible dimensions. This match will also remove acceleration (and any other prefix matches) from the list of possibilities.
- *exact match:* Any variable name that matches exactly with the string of an entry in the knowlege base will have the associated set of dimensions. This match overrides and excludes all other matches. The variable G will have the dimensions of the univeral gravitational constant and the match will remove all prefix or pre-emptive matches with G. The variable $G1$ however will not be an exact match.

The improved matching capability provided us with ways to specify preferences amongst the different possible matches for a variable.

Another problem made it more difficult to generate good feedback when the equations are not dimensionally consistent. Depending on the order in which constraints are checked, information from an inconsistent equation may be propagated to other equations before an inconsistency is discovered. At that point, it is difficult to determine the origin of the problem, i.e., which equation was inconsistent. In addition, a variable can be used in multiple equations and more than once in an equation. The constraint graph only maintains one copy of each variable since all occurrences have the same set of dimensions. This makes it difficult to determine which instance of the variable was used incorrectly when an inconsistency is discovered. These problems were solved with the following changes to the constraint graph and associated procedures.

- *create a leaf node for each occurence of a variable:* Instead of having only one node for each variable, a node is created for each occurrence of a variable. When a variable is determined to be inconsistent, the specific instance that is at fault can then be pinpointed. To maintain consistency within the system of equations, a new type of constraint is added, an **identity** constraint. The constraint connects all nodes that are instances of the same variable and restricts the nodes to have the same set of dimensions.

- *delay propagation across terms and equations:* Essentially, this heuristic favors propagation of information to (1) nodes in the same local region and then to (2) nodes "further away". This is a means of making it easier to detect inconsistencies in the regions where the fault lies.

The goal of these changes is to delay information propagation across terms and equations and maximize the inference of dimension values. This heuristic is intended to discover inconsistencies as locally as possible before incorrect dimension information can be propagated to other terms or equations.

4.4 Final Evaluation

The changes described in the previous subsection were implemented and the resulting module was re-tested on the data from the ANDES logs. The results are shown in Table 1 where the column labelled *Unique* is the number of sets of equations where the dimensions of all variables and constants are uniquely determined, the column labelled *Ambiguous* is the number of sets of equations where there is at least one variable or constant that has more than one possible valid set of dimensions. The first row (Full KB,full match) shows the results from using a knowledge base with (1) all three matching techniques and (2) information about the use of variables throughout all of Physics. The full knowledge base contains 171 entries. The second row (Newton KB, full match) shows the results from using a knowledge base with (1) all three matching techniques and (2) information about Newtonian mechanics only. This knowledge base contains 109 entries.

As described earlier, the ANDES system permits the students to use numeric values in place of variables, e.g., 9.8 instead of g for gravity. Consequently, constants can sometimes have unstated dimensions and the system has to treat each constant initially as having all dimension possibilities. In the evaluation, we found that there were many equation sets where

Set	Unique	Ambiguous	Inconsistent	%Unique
Full KB, full match	4921	4396	548	50%
Newton KB, full match	7470	1983	412	76%
Newton KB, full match, constants	8195	1258	412	83%
Newton KB, full match, constants, 1 variable	8677	776	412	88%

Table 1: Evaluation on the ANDES data

the dimensions of all the variables were determined but the dimensions of some of the constants were ambiguous. Further examination revealed that the constants should be treated as dimensionless but the initial assumption prevented this. The third row (Newton KB, full match, constants) shows the result of treating these equation sets where only constants were ambiguous, as unique (an accounting shift). The results show that without any special information about ANDES, e.g., variable naming conventions, our technique can determine the dimensions of all the variables in 83% of the sets of equations even when most of the sets are incomplete. The last row (Newton KB, full match, constants, 1 variable) shows the increase in the number of unique sets if the system could ask the student for the dimensions of one variable. Four hundred of the ambiguous sets of equations (or 4.89% of the total) from the second row have only one variable that is not uniquely defined. Thus, by asking at most one question of the student, the technique can uniquely determine the dimensions of all the variables in 88% of the sets of equations.

5 Conclusion

This paper has shown how domain knowledge combined with heuristic constraint propagation is used to determine the context and implicit information contained in student answers, specifically the dimensions of variables in systems of equations. This approach has been tested and evaluated on answers from students at two colleges. The results show that the technique uniquely determined the dimensions of all the variables in 83% of the sets of equations. By asking for dimension information about one variable, an additional 5% of the sets can be determined.

Scaffolding is a technique that is useful and helpful to beginning students. After some experience, students would benefit from having the scaffolding removed. The experiments validate the hypothesis that our technique allows us to remove the scaffolding from a physics tutoring system and still be able to determine the dimensions of the variables used in the equations.

6 Acknowledgements

We are grateful to Kurt Van Lehn and the Andes group for making the ANDES logs available to us, and to Anders Weinstein for answering questions about the ANDES implementation. Jim Appenzeller and Dave Santin provided invaluable help in the implementation and evaluation of the system.

References

[1] R. Cunis, A package for handling units of measure in lisp. *ACM Lisp Pointers 5*, 2 (1992).

[2] A. S. Gertner, Providing feedback to equation entries in an intelligent tutoring system for physics. In *Proceedings of the 4th International Conference on Intelligent Tutoring Systems* (1998).

[3] P. N. Hilfinger, An Ada package for dimensional analysis. *ACM Transactions on Programming Languages and Systems 10*, 2 (1988), 189–203.

[4] C. W. Liew, Joel A. Shapiro and D. E. Smith, Reasoning about algebraic answers in physics. In *Proceedings of Twelfth International Florida AI Research Society Conference* (1999), pp. 167–171.

[5] C. W. Liew and D. E. Smith, Checking for dimensional correctness in physics equations. In *Proceedings of Fourteenth International Florida AI Research Society Conference* (2002).

[6] C. W. Liew and D. E. Smith, Reasoning about systems of physics equations. In *Lecture Notes on Computer Science (LNCS 2363) - Intelligent Tutoring Systems 2002* (2002), Springer Verlag.

[7] G. S. Novak, Conversion of units of measurement. *IEEE Transactions on Software Engineering 21*, 8 (August 1995), 651–661.

[8] P. S. Ow and S. F. Smith, Towards an opportunistic scheduling system. In *Proceedings of 19th Hawaii International Conference on System Sciences* (1986).

[9] L. Steinberg, Design as refinement plus constraint propagation: The VEXED experience. In *Proceedings of AAAI-87* (1987).