THE MECHANICS PROBLEM SOLVER:
A problem and goal driven inferencing system **

by

George F Luger
Research Fellow

Department of Artificial Intelligence
University of Edinburgh

abstract

The MECHO* project is discussed by one of the researchers who helped design it. The goal of this problem solver is the representation and solution by computer of problems in applied mathematics especially in the domain of mechanics. A forward or problem driven and a backward or goal driven inferencing system is employed. A pulley problem is presented and used to exemplify the approach of the MECHO system.

* For MECHanics Oracle, a Science Research Council Project, funded through the Department of Artificial Intelligence, University of Edinburgh -- Dr. A Bundy, grantholder, Dr. G Luger and Ms. M Palmer assisting.

** Published in Topics in Cybernetics & Systems (Rose, ed) Berlin: Springer Verlag, 1979.

THE MECHANICS PROBLEM SOLVER:
A problem and goal driven inferencing system

1. INTRODUCTION

The MECHO project consists of writing a computer program to solve problems in applied mathematics. The scope of the project is broad: to take the English statement of a mechanics problem, give it to a computer, and receive in return answers to the questions asked in the problem. Three problem domains within the general area of mechanics have so far been considered: (1) acceleration, velocity, distance problems such as might arise with trains travelling between two stations (Bundy, Luger, Stone and Welham, 1976), (2) the motion of particles over complex paths, such as the "roller coaster" problems tackled by de Kleer (Bundy, 1977) and (3) the domain of pulley systems (Luger, 1977).

The mechanics world offers a problem solving domain rich in semantic information. The MECHO project has designed a computer program to represent concepts, inferences, and search strategies within this domain. The thrust of the MECHO project research is pragmatic however, in that its primary goal is to design a computer program that can solve a wide class of problems. A further, but very important, goal of the project for the author is the study of the running computer program as a model of human problem solving activity. The trace of the program can be compared with the data of human protocols. The MECHO group has found this comparison fruitful both as a source of new ideas that may be incorporated in the computer program itself, as well as to clarify important differences between the human and mechanical problem solving systems.

One of the important insights gained from studying human problem solving protocols (Marples, 1974; Luger, 1977) has been to design the MECHO system as a forward or problem driven and a backward or goal driven problem solving system. The remainder of this paper will be spent clarifying this approach to problem solving and describing its implementation in the MECHO system.

2. FORWARD INFERENCING

Bhaskar and Simon (1976) refer to a problem solving domain such as ours as a "semantically rich" domain. Our world of mechanics (like their world of engineering thermodynamics) demands of the problem solver large amounts of prior semantic knowledge and task related information. For example, smooth pulleys and inextensible strings are objects with a fairly well defined list of properties and relationships to other objects in their worlds. As Bhaskar and Simon point out, no such information or previous learning is relevant to the usual problem solving studies of cryptarithmetic, missionaries and cannibals and the tower of hanoi problems.

In the search for an ideal representation of objects and inferences in the mechanics domain we have considered several sources. First, we have taken protocols from subjects solving mechanics problems. The subjects (each having passed a university course in mechanics) were asked to solve four related problems in the world of pulley systems. These were designed to "tease out" the subjects framework for conceptual knowledge in the mechanics domain (smooth surfaces, frictionless pulleys, and so on) as well as to outline their search and solution strategies. Subjects retrespective accounts were occasionally requested when the researchers felt sections of a protocol needed further clarification.

The second source of information was the study Learning to Solve Problems in Engineering conducted at Cambridge University by D Marples (1976). This study took numerous protocols of problem solving by engineering students as well as of regular tutorial sessions at the university in an attempt to characterize and improve the processes used to educate the engineer.

The study found two major sources of mistakes in the problem solving subjects:

1. the problem solver was uncertain about the nature of engineering concepts, especially their generalizations and relationships to other concepts;

2. the problem solver tended to remember concept relationships in a generalized (algebraic) form and the conditions of implementation were often forgotten or ignored.

The Marples study also suggested an algorithm (discussed in Section 3) for generating a sufficient set of equations to solve the engineering problems. This algorithm starts from the "desired unknown" and creates a chain of equations going back to include the given relations of the problem. This algorithm, automated in the MECHO system, performs the main search in the solution procedure.

The paramount question in solving problems in applied maths seems to be not what information is available to the problem solver, but how this information is organized and stored. The Bhaskar and Simon study, the Marples research, and our own protocols suggest strongly that this is the key to successful problem solving. We have adopted a system of schemata to implement the representation of concepts and their interrelationships. Hinsley, Hayes and Simon, (1976) also talk about the notion of schemata in algebra word problems. They have designed several studies to determine the use of schemata in classifying and solving these word problems. Specifically, they found with algebra word problems, such as those found in secondary school textbooks:

1. People categorize problems into "types". Problems in a sense are not "unique" but may be classed generally into "age" or "river crossing" or "work" problems.

2. People categorize these problems even before they have assimilated all the information or formulated them for solution.

3. People have a body of information about each problem type which is potentially useful for formulating problems of that type for solution. This information helps focus an important facet of the problems, to direct attention to the proper equations, etc.

4. People use this category or problem type information in formulating problems for solution, not just for classifying them.

The MECHO system employs problem driven forward inferencing, such as that outlined above. This is accomplished by the creation and assertion of problem type schemata.

The word schema is used, following Bartlett and Piaget, to refer to a structuring of information, a loose confederation of relationships, that represent the capacity to perform some task or function. In the terminology of Hinsley, Hayes and Simon (1976) the problem type schema contains the semantic information present in a problem situation along with the ability to use this information for solving a particular problem.

A simple problem, in fact one of the first problems considered by the MECHO group, is:

I. A man of 12 stone and a weight of 10 stone are connected by a light rope passing over a pulley. Find the acceleration of the man.

The problem type schema is composed of three parts: the declaration of entities, the set of facts and inference rules describing the problem situation, and a set of default facts and inferences. The declarations for the pulley problem above were often explicitly stated in the protocols we took of expert problem solvers: "We'll treat the man and weight both as particles, point masses .... I'll put these two dots on the paper and join them by this rope looped over a pulley". The entities declared in this situation are two particles, a pulley, and a rope over the pulley joining the two particles.

The set of facts and inferences relating these entities are similar to the following:

(a) an angle is assigned to the string between the pulley point and the left end;

(b) an angle is assigned to the string between the pulley point and the right end;

(c) fixed contact of particle1 to the left end of the string;

(d) fixed contact of particle2 to the right end of the string;

(e) the tension in the left section of the string is the same as the tension in the right end if the pulley is smooth (frictionless);

(f) the acceleration of the system is constant if the particles are in fixed contact with the string.

(a), (b), (c) and (d) above are examples of facts, and (e) and (f) are inferences that represent part of the semantic content of the pulley system domain.

Finally, the pulley system schema contains a set of default values. These values are facts and inferences such as:

(i) if the pulley is underspecified assume it to be smooth;

(ii) a rope is assumed to have constant length unless specified as elastic;

(iii) a rope has fixed contact with objects at its end points unless the problem states otherwise;

(iv) the pulley itself is fixed unless specified as moveable.

Note that once the pulley system schema is invoked much of the "problem solving expertise" is represented in the inferences and the set of default values. For example, a TENSION will be assigned to the string and preserved throughout the string if the pulley is smooth (frictionless). Furthermore, an ACCELERATION will be assigned to a particle and preserved throughout the string and passed on to a second particle, if the string is not elastic, i.e., if it has constant length, and if the particles have a fixed contact to each end of the string.

There are many "house rules" (Marples, 1976) in the mechanics game. These are often represented in the default values. If nothing is said in the problem statement about the friction of the pulley, it must be assumed that the pulley is smooth. Furthermore, even though a rope is mentioned it is assumed to be a dimensionless entity that may preserve a tension, change the direction of an acceleration if placed over a pulley, and that keeps particles in contact. Finally, the mechanics domain is a model of a real world situation where gravity is constant, strings and attached weights hang vertically downwards, and so on. These facts play a part of the concept schema and are asserted as default values.

The schema for pulley problems is "hierarchical" in the sense that more information may be assumed and/or asserted depending on the requirements of each problem. The determination of the needs of each problem is a complex task and will not be exhaustively discussed here. The following example should indicate the nature of this task:

II. A smooth pulley is at the peak of a wedge-shaped block. Two particles of masses M1 and M2 are on the smooth faces of this block and are connected by a string over the pulley. If the block's faces have inclination α with the horizontal, what is the acceleration of the system?

In this problem (II) c, d, e and f of the facts and inferences above must be supplied by the schema while a and b may be determined from the problem statement. In problem (I) above, the angles of inclination of the string had to be assigned as vertically downwards. The schema system that satisfied both problems was "hierarchical" in that all the facts and inferences for problem II were needed for problem I and further specification was then needed for problem I.

The MECHO system's data base is ordered so that a problem type schema, when it is invoked, is able to create new entities and assert new facts and inferences at the "top" of the data base and assert the default values at the "bottom". Thus, when a call is made to the data base the facts and inferences about entities are checked first, then finally, after every other check is made, the default values are assumed. In the pulley problem above, when a resolution of forces formula is attempting to assign a tension to the string, it will need to know whether the pulley is smooth. (If it is, a uniform tension will be assigned to the entire string). When no information about the friction of the pulley can be found, as is the case in this problem, the default value of a pulley without friction will be asserted. Similarly, when the angle of the string is sought, the default value of the string with the weight hanging vertically downwards will be asserted.

So it is that the problem type schema, representing the semantic information of the problem situation, is asserted. This represents the forward or problem driven aspect of the MECHO problem solver. The goal driven aspect is represented by the "Marples" algorithm for equation extraction.

3. GOAL DRIVEN REASONING: THE MARPLES ALGORITHM

Marples (1974), from his analysis of engineering students protocols, has described a method for solving problems in the mechanics domain. This method, very much like Polya's (1945), "take the problem as solved" heuristic, was automated and forms the basis of the MECHO algorithm for extracting equations.

A set of possible equations, and the situations in which they may be called, is asserted. For example, for the pulley system problem above one of the set of possible equations would be the equation for resolving forces at the contact point of a particle and a string. It is asserted if there is a particle and a string and a period of time, and if the particle has a MASS, if the string has a TENSION, and if the particle and string have an ACCELERATION. (It should be noted that the actual equation for resolving forces in MECHO is more complex than this in that it is able to resolve arbitrary forces at the contact point and so able to handle inclined planes, table tops that are rough, and other situations).

The Marples algorithm is a procedure which starts from the desired unknown of the problem and works "backward", attempting to instantiate equations until a set of simultaneous equations sufficient to solve the problem is determined. The MECHO system has a focusing technique that "forces" the Marples algorithm to consider equations appropriate to the problem type, rather than to thrash about through lists of all possible equations. The focusing technique in the pulley system domain forces the Marples algorithm to consider first the general resolution of forces equations at the contact points of the string and weights.

Furthermore, the Marples algorithm is able to create "intermediate unknowns" in the process of solving the desired unknowns of the problems. In the pulley problem, for example, the desired unknown is the acceleration of the man. But it is impossible to determine the man's acceleration (using the resolution of forces) from the givens of the problem. Thus the Marples algorithm creates an intermediate unknown, the TENSION in the string.

The Marples algorithm starts its search with a list of givens and a list of sought unknowns. For problem I these would be:

Givens list: M1, M2, G (mass of man, mass of weight, acceleration of gravity)

Unknowns list: A1 (acceleration of man)

The algorithm first attempts to find an equation for the first unknown as a function of the givens. If this fails, as it will in this instance, the algorithm finds an equation for the first unknown as a function of the givens and one or more intermediate unknowns. The original unknown (A1) is then added to the list of givens, the intermediate unknown (T1) added to the list of unknowns, and the equation asserted in a list of equations. In the example:

Givens list	M1, M2, G, A1
Unknowns list	T1 (tension in the string)
Equation list	M1*G - T1 = M1*A1

The Marples algorithm makes equations recursively until the unknowns list is empty. (Only a bit more sophisticated control is necessary to guard against looping and to assure a list of linearly independent equations). In the example the resolution of forces routine is called once more and the algorithm halts with:

Givens list	M1, M2, G, A1, T1
Unknowns list	nil
Equation list	M1*G - T1 = M1*A1, M2*G - T1 = -M2*A1

The equation list is then solved (Bundy, 1975) with the given values substituted for M1 and M2, and the solution A1 = $^{G}/11$ is asserted.

4. SUMMARY AND CONCLUSION

The MECHO problem solver is not the creation of this author. In fact the SRC grantholder, Dr Alan Bundy, is the chief architect and implementer of the ideas contained in the system. The research assistants, including the author, have been a continuing source for many of the ideas that went to make up the program. Fortunately, the demands of the program were large enough to challenge us all.

Even an overview and characterization of a program, such as this is, need not express the outlook of all those working on it. The grant holder, perhaps as a hold over from earlier "theorem proving" days, describes the MECHO as a backwards directed inferencing system (Marples algorithm) with schemes used to fill the "gaps" between what is given in the problem statement and what is needed to solve the problem. The author's bias, perhaps coming from analysis of human problem solving such as the Hinsley, Hayes and Simon (1976) study cited above, and protocols taken from subjects, sees the MECHO problem solver as forward or problem driven inferencing using the schema system coupled with backwards inferencing driven by the goals of the problem. Such is the approach presented in this paper.

In fact either interpretation is valid as the MECHO program is implemented in PROLOG (Warren, 1977) and has many of the properties of a condition driven and action driven production system (Waterman, 1977) including a total modularity and ordering property of clauses. Modularity allows the addition or deletion of clauses (such as is done by the schema system) and allows several researchers to work on different parts of the program without it coming completely "apart". Priority ordering of clauses allows different clauses to be called depending on the needs of the solver. This, for example, allows default values to be asserted at the "bottom" of the list of clauses and be called only after every other clause has been called and fails.

The MECHO project has been described in greater detail and from other viewpoints in Bundy et al (1976), Bundy (1977), Luger et al (1977) and Luger (1977).

ACKNOWLEDGEMENTS

The author would like to thank the other members of the research group for many hours of challenging and creative explorations together.

REFERENCES

Bundy, A., Luger, G.F., Stone, M. & Welham, R. MECHO: Year One, *D A I Research Report No. 22*, Department of Artificial Intelligence, University of Edinburgh, 1976.

Bundy, A. Will it reach the top? Prediction in the mechanics world. *D A I Research Report No. 31*, Department of Artificial Intelligence, University of Edinburgh, 1977.

Bundy, A. Analysing mathematical proofs (or reading between the lines). *Proceedings of International Joint Conference on Artificial Intelligence, 4*. Winston (ed), Georgia, Cambridge, MIT-AI Press, 1975.

Hinsley, D.A., Hayes, J.R. & Simon, H.A. From words to equations: meaning and representation in algebra word problems, *C.I.P. 331*, Department of Psychology, Carnegie-Mellon University, 1976.

Luger, G.F., Bundy, A. & Palmer, M. Problem solving in applied mathematics: An Artificial Intelligence approach . In *American Society of Engineering Education monograph* Lubkin (ed), 1978.

Luger, G.F. The use of protocols and the representation of semantic information in pulley problems. *D A I Research Report No. 36*, Department of Artificial Intelligence, University of Edinburgh, 1977.

Marples, D.L. Final Report to U.K. Social Science Research Council of Cambridge Project, 1976.

Marples, D.L. Argument and technique in the solution of problems in mechanics and electricity. *CUED/C - Educ/TRI*, Department of Engineering Memo, University of Cambridge, 1974.

Polya, G. *How to solve it*. Princeton, Princeton University Press, 1945.

Warren, D. Logic programming and compiler writing. *D A I Research Report No. 44*, Department of Artificial Intelligence, University of Edinburgh, 1977.

Waterman, D. An introduction to production systems. *AISB Newsletter, Issue 25*, Department of Artificial Intelligence, University of Edinburgh, 1977.