AN AUTOMATED CONSULTANT FOR MACSYMA

by Michael R. Genesereth

Center for Research in Computing Technology
Harvard University
Cambridge, Massachusetts

Laboratory for Computer Science
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

Consider a person trying to solve a problem with a computer system he does not fully understand. And assume that, although he has encountered a difficulty due to his lack of knowledge, he is unwilling to learn more about the system than is necessary to solve the problem. The simplest way for him to acquire just the information he needs and no more is to consult an expert. Consultation is a method widely used in computer centers as well as in domains like business, law, and medicine. Unfortunately, human consultants are a scarce resource and quite expensive. The purpose of this paper *is* to propose as an alternative an automated consultant and, in doing so, to outline its requirements and publicize a current research effort* directed toward constructing such a consultant for the algebraic manipulation system MACSYMA. Such a program should be able to converse with its user in English about a difficulty he has encountered and provide advice tailored to his need. As currently conceived, the MACSYMA Advisor would be a program distinct from MACSYMA with its own separate data base and expertise.

One of MACSYMA's strongest user aids is its staff of human consultants, available on-line to help users with resource knowledge difficulties. During the last three years, the author has served as a MACSYMA consultant and recorded many of these consultation sessions. During the same three years, MIT. has offered a course on "knowledge-based systems" in which one of the requirements is the solution of a MACSYMA problem and an analysis of the resulting protocol. The analyses were supposed to indicate which information sources were consulted and why. The author also had the opportunity to read many of these analyses. An examination of the data obtained from such consultations and protocol analyses reveals that in using MACSYMA, people perceive the need for five general classes of information.

(1) The user needs to know the name of a command or technique to do some task. If he were to phrase his need as a question, he would ask "How do I do ...?" This is called a HOWDO need.

(2) He needs to know a command's prerequisites, arguments, postrequisites, etc. He would ask "What are the ... of ...?". A WHAT need.

(S) He needs to check his beliefs about MACSYMA. He would ask "Is it the case that...?". An IS need.

(4) He needs a procedural explanation of how a command works or a result was obtained. He would ask "How did MACSYMA do ...?". A HOW need.

(5) MACSYMA has returned an unexpected result, and he can find nothing wrong with his derivation. He needs sufficient information to pinpoint and correct the misconception underlying his erroneous expectation. He would ask "Why is It that...?" A WHY need.

Of these, the questions requiring the most sophisticated treatment are WHY and HOW. WHAT, HOWDO, and IS questions can be answered directly, with no consideration of the the user's purpose or his state of knowledge. A WHY or HOW question calls for different answers to different people in different situations.

A WHY question arises when a contradiction between MACSYMA and the user's model (the misconception) becomes manifest in a violated expectation. However, the underlying misconception need not be immediately apparent. The violated expectation may depend on earlier mistakes that were not observed at the time, i.e. the point at which the user's misconception first had its effect in the user's plan (the locus) may be remote from the observed contradiction (the manifestation). A WHY question is a request by the user for enough information to correct the misconception underlying the manifestation. This can be supplied either by identifying the manifestation and its locus (model debugging) and correcting it directly or by explaining the correct result in detail (explanation) and letting the user do the debugging himself.

The key to the model debugging process is the user's "plan". A plan is essentially a goal-subgoal tree with enough annotation to explain the user's expectations. The Advisor constructs an explicit representation of the user's plan (plan finding) and then tries to find a bug. It may be able to recognize some pattern that indicates a "standard" mistake, one that many users make (using plan recognition). Or it may be able to recognize an almost correct plan (by partial recognition). In either case, having discovered an issue, the Advisor then tries to confirm that it is the locus by engaging in a question and answer session (plan taking) in which the user verifies the Advisor's deductions and supplies further information. Once the issue is confirmed, corrective action is taken.

In asking a HOW question the user is seeking a procedural explanation for a result or act or fact in order to remedy a deficiency in his model of MACSYMA. However, no violated expectation is provided, and so the Advisor cannot pinpoint this deficiency. Whenever a deficiency cannot be discovered, as with a HOW question or when model debugging fails, the Advisor resorts to explanation. An explanation is a partial plan. For acts, only the skeleton of the plan is given; for facts, the Advisor also includes enough annotation to prove the result. Explanation plans can be generated by a problem solving proof procedure with the aid of a meta-evaluator. Using an explicit model of the user, the plan can then be pruned by an enthvmeme deletion process to eliminate general statements the user already knows and that would therefore be useless.

Although the various parts of the Advisor have all been implemented, as of this writing they have not yet been combined into a working system. Also, the present data base is at best meager. The current timetable calls for its release to the MACSYMA user community this fall, where if successful it will find heavy use and provide valuable data for further improvements.

*Genesereth,M.R.: Automated Consultation for Complex Computer Systems. Ph.D. Thesis, Harvard Univ., Nov. 1977.