On the Design of an Expert Help System for Computer Algebra Systems*†

R. P. dos Santos‡ and W. L. Roque§
Universität Karlsruhe
Institut für Algorithmen und Kognitive Systeme
Am Fasanengarten 5 – D7500 Karlsruhe 1 – FRG
Phone: (+49) 721 6084328
BitNet: kg07@dkauni2 and kg03@dkauni2

1 Introduction

REDUCE[2, 3], like many modern Computer Algebra Systems (CAS) (MACSYMA, MATHEMATICA, MAPLE, SCRATCHPAD to name a few), embodies a large amount of mathematical knowledge which is spread out through thousands of procedures in the source code of the system. Most of this knowledge is, however, in an *implicit* form almost inaccessible to the user, who would have to decipher the source files to recover it. Also, to profit from it one needs to know in addition the capabilities of these computing systems and how to operate them.

The process of learning how to use a CAS may be done by reading throughout the manual available for the system, a sort of user's guide, a book or in the practical *hands on* approach. However, soon the begginers get stuck and the most confortable and easiest way to solve the problem is consulting an *expert* on the system. They, in general, do not want to waste their time reading either a book or manual looking for the terminological structure of the system. Particularly, to its semantics, syntax and/or examples just to be able to start using the system to solve a simple problem: an integral, for instance. That is, perhaps, the reason why many the potential users with no previous experience with computers keep avoiding to use them.

These kinds of informations could be easily provided by an on-line help system (and some progress in this direction have been attained[4]) or even better, by an Intelligent Tutorial System[5]. The idea behind is that existing a stored knowledge base, consisting of a large number of information elements, structured in levels of specialization, an *expert help system* could generate by inference (not just by pure recovering) an information, which might not be explicity stored and that might be an adequate (in terms of abstraction, difficulty, detail, etc.) directive to the user. An intelligent help facility could reason about the user's query and then give a reasonable reply or else, suggest what has to be consulted or even what to do next.

It is our intention here only to discuss the nature, complexity and tools concerning the design of *Smart Help*, an expert help system with these features. Presently, the *Smart Help* domain knowledge base concerns REDUCE, however it may well be used to implement different CAS bases. Since the hybrid knowledge representation system (KRS) MANTRA[6] can be seen as a knowledge representation shell, one could make use of its (formal) reasoning facilities to build the *Smart Help* system. The ideas forwarded herein are in a prototypal basis, but we hope that they will raise the interest of the CAS community and evolve to reach at the end the full system implementation.

In section 2 we comment briefly on the computer algebra system REDUCE and propose a taxonomy for the knowledge embodied by this system. In section 3 we describe the knowledge representation system MANTRA. Section 4 and 5 are concerned with the *Smart Help* system design. Finally, in section 6, we give some conclusions and suggest further upgrades which would turn *Smart Help* into a truly *Intelligent Tutorial System*[5]. An example of the interaction is also given. Further details can be found in [1].

*This is a short, less technical version of [1].

†This research was supported by Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq.

‡On leave of Centro Brasileiro de Pesquisas Físicas, Rua Xavier Sigaud, 150, 22290 Rio de Janeiro, RJ, Brazil.

§On leave of Universidade de Brasília, Instituto de Ciências Exatas, 70910 Brasília, DF, Brazil.

2 The CAS REDUCE

REDUCE[2] is a fairly powerful system for carrying out a variety of mathematical calculations such as to manipulate polynomials in a variety of forms, simplify expressions, to differentiate and integrate algebraic expressions, do some modern differential geometry calculations, study the Lie-symmetries of systems of partial differential equations, and many others.

This system is made out of different knowledge domains (mathematical, terminological, computational, etc.). In order to represent in *Smart Help* the specific knowledge domain involved in operating REDUCE we propose to classify it in the following categories:

syntax: Informations about the number and type of the arguments, and requisites and prerequisites of a REDUCE command, declaration or operator (that is the only type of information given by many help systems).

terminology: Definitions of the various terms like *identifier*, *operator*, *kernel*, etc., employed in documents related to REDUCE and, also important, in error messages from REDUCE (this matter can be quite confusing to the initial user).

concepts: Informations like the fact that *integration* in REDUCE is represented by an operator called **INT**, or a reference for the algorithm which it implements.

procedures: General sequences of commands which should be given to perform a certain task like defining a new infix operator: one has to declare the operator infix through the INFIX declaration and then give him a precedence by means of the PRECEDENCE declaration.

heuristics: General rules and tips that simplify and improve approaches to problem-solving (this kind of information comes with experience and is one of the most frequent in consultations from beginners). For example, "If you want to compute a definite integration, you can try evaluating the indefinite integral, saving the resulting expression in a variable and then substituting locally the limits of integration in it and finally subtracting the results".

3 The KRS MANTRA[1]

MANTRA[6] is a hybrid knowledge representation system which integrates three different knowledge representation methods:

Four-valued first-order logic language, used to express *assertional knowledge*, which is decidable but presents a weaker entailment mechanism which excludes the chaining of independent facts, thus ruling out *modus ponens* but allowing quantifiers.

Terminological language (a kind of *frame* method), extended to allow the definition of concepts and n-place relations over themselves.

Semantic network, which is a representation to define hierarchies with exceptions.

One could think that classical logic would be able to generate all knowledge entailed (i.e., implicity represented) by a given concept. Unfortunately, however, the systems based on the complete first-order logic, suffer from the inherent problem of *Combinatorial Explosion*. Also, the entailment problem is not decidable in first-order logic. In addition, the knowledge to be represented is often incomplete and incoherent.

Due to these facts, the knowledge representation system MANTRA has been designed associating the three knowledge representation methods above, based on a common four-valued semantic, which allows the modeling of the aspects of *ignorance* and *inconsistency*, useful for representing incomplete and/or incoherent knowledge.

4 The Conception of the *Smart Help* Expert System

The conception of *Smart Help* follows the tradition of help systems being passive. This means that the user learns how to use REDUCE playing freely with it without being interrupted by the *Smart Help*. To the user it looks like a normal REDUCE session but MANTRA is running behind and is accessible as an operator, by means of the interface MANTRA-REDUCE. When the user gets a confusing answer or a meaningless error message (and in fact REDUCE users know how often this happens), or even when he does not know what to do next to get his calculations done, he invokes the *Smart Help* to clarify the point.

The problem might have been caused by earlier mistakes (a forgotten variable definition long before, for example). An explanation facility[7] to trace the session history and find the exact place at which the misconception first had its effect was not included in *Smart Help*. It is then left to the user to do the right question to get the right answer. That is, progress can be

[1] This description of the MANTRA System is based on [6]

made only if, from the correct definition, usage, prerequisites, etc., of the queried topic, as returned by the *Smart Help*, the user can pinpoint the misconception which caused the problem.

5 The Architecture of *Smart Help*

Technically, *Smart Help* is a Production System on the top of a particular implementation of MANTRA which has REDUCE integrated as an additional knowledge representation module. Since the heuristic level of MANTRA has not yet been implemented, being presently represented by the Lisp language itself, *Smart Help* is coded in Lisp and resides in the same Lisp session of MANTRA.

Considering the taxonomy of the knowledge embodied by REDUCE, presented in section 2 above, and the knowledge representation methods available in MANTRA, as described previously in section 3, we had to find the best fit of both to guarantee efficiency in recovering knowledge from the base and in reasoning with it, according to the inherent structure of each knowledge category and to its adaptiveness to the specific representation method. The five categories of knowledge were implemented as *aspects* of the knowledge base *object*, making use of Corbit[8], an object-oriented extension of Common Lisp. Details of this can be found in [1].

A number of rules were implemented in the production system of *Smart Help*. We present them in the following. Presently, they are inserted in the code itself. We consider now to get them defined in a production rule base, what would give flexibility and clearness to our system.

When asked by the user about a topic, *Smart Help*

1. Assumes that the present level of familiarity of the user with REDUCE (student model) can be inferred from the level of specification of the queried topic, which is characterized by a numeric heuristical parameter, ranging from 1 to 3, associated to it. For example if someone queries about "integration" (very general – parameter = 1) it seems probable to be a very novice user but if one queries about "infix-operators" (more specialized – parameter = 2) it should be considered as an user with some familiarity.

2. Queries the domain knowledge base, to recover all informations related to the given topic and store them in the "answer" structure. This process consists in querying the *conceptual*, *terminological*, *syntactical*, *procedural* and *heuristical* aspects of the base.

3. Evaluates the level of generality of the terms recovered to see if they are in the same level of specialization (same value of the parameter). If a concept is too specific (much greater value of the parameter), *Smart Help* tries to redefine it in terms of more general concepts. If a concept is too general, however, it is simply deleted.

4. Formats the recovered knowledge in the form of a readable answer, defining the queried concept and its usage in terms of the recovered knowledge. Presently this process is quite crude as it is a peripherical point of the implementation. It will be improved latter on.

5. Prints the answer returning control to REDUCE.

For better understanding, suppose that an user with no experience with REDUCE wants to do some calculations, for instance, integrate an expression in terms of a certain variable. Having access to an initialized session of REDUCE (in which the heading shows how to invoke *Smart Help*), the interaction would consist in typing `shelp integration` , and *Smart Help* would promptly reply:

```
"SHelp - Version 2.0:  23 Aug 1990"
INDEFINITE-INTEGRATION is the default
   for INTEGRATION.
INTEGRATION is represented in Reduce
   by INT.
Its syntax is:
INT(scalar-expression,variable)
Ex.:  INT(LOG(X),X);
INT is implemented through the
SIMPINT-PROCEDURE-IN-INT-SOURCE-FILE.
For DEFINITE INTEGRATION, one may try
to DO INDEFINITE-INTEGRATION,
to SAVE-RESULT-IN VARIABLE,
to LOCALLY-SUBSTITUTE VALUES.
References:  REDUCE MANUAL SECTION
   7.4.
See also:  DERIVATION
```

6 Conclusions

We have presented and proposed in this paper a fairly general design of an expert help facility for aiding users of Computer Algebra Systems. Although the expert help system presented here has been particularly oriented to REDUCE (as a consequence of our former experience with this system), we point out that the concept of *Smart Help* can be extended to other Computer Algebra Systems.

The reasons for introducing *Smart Help* facility include:

- It will provide an on-line help for the system, aiding the users to find specific informations about the system terminology, structure, syntax, etc.
- It will allow the potential user to access the whole capabilities of the system.
- It can contribute to the development of Intelligent Computer Algebra Systems[9], which more than simply being able to do calculations, could interact with the user and free him of many details concerning the specification of his problem.

The *Smart Help* has no intention to *teach* the user how to program efficiently in REDUCE or behave like a tutoring system. However, it can be used in the learning process as a complimentary teaching tool.

Following the ideas addressed in this paper, we intend afterwards to develop an *Intelligent Tutorial System*[5] (ITS) for REDUCE. The ITS should inherit all the compatible facilities already available in the *Smart Help*. In addition, many other facilities would become available, such as, a deeper understanding of REDUCE's semantics, keeping track of history, explanations[7], a dynamical reasonning on the student's model[5], etc.

A prototype of *Smart Help* is now running on a SUN work-station on an experimental basis. The full implementation of *Smart Help* as a final product was not our main concern here. This task will certainly need few more people working in a close colaboration to build up a satisfactory knowledge base to reach at the end the principal objective that is helping a CAS user.

7 Acknowledgement

We would like to thank Prof. J. Calmet for enlightening discussions and for the warm hospitality provided by him and by all members of his group and the Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq for the financial support.

References

[1] dos SANTOS, Renato P., and ROQUE, Waldir L., An Expert Help System for Computer Algebra Systems, internal report, Institut für Algorithmen und Kognitive Systeme, Universität Karlsruhe, 1990.

[2] HEARN, Anthony C., REDUCE User's Manual: Version 3.3, RAND Publication CP78, The Rand Corporation, Santa Barbara, Calif., 4/1987.

[3] MacCALLUM, M.A.H., and WRIGHT, Francis, REDUCE Lecture Notes, *in* REBOUÇAS, M.J. (ed.), proceedings of the I Brazilian School on Computer Algebra, vol. I, Oxford University Press (forthcoming book), 1990.

[4] HARPER, David, Reduce Forum, 23/8/1989; SCHOEPF, Rainer M., Reduce Forum, 24/8/1989; LAMBE, Larry A., Reduce Forum, 24/8/1989; AGER, Tryg, Reduce Forum, 24/8/1989; DEWAR, Mike, Reduce Forum, 24/8/1989; MARTI, Jed, Reduce Forum, 2/8/1990; WRIGHT, Francis, Reduce Forum, 2/8/1990; COPELAND, Gary, Reduce Forum, 2/8/1990.

[5] SLEEMAN, D. and BROWN, J.S., Intelligent Tutoring Systems, Academic Press, London, 1982.

[6] BITTENCOURT, Guilherme, The MANTRA Reference Manual, Interner Bericht 2/90, Univ. Karslruhe, Fak.f.Informatik, Karlsruhe, West Germany, 1/1990; BITTENCOURT, Guilherme, An Architecture for Hybrid Knowledge Representation, PhD Thesis, Univ. Karslruhe, Fak.f.Informatik, Karlsruhe, West Germany, 31/1/1990.

[7] MARTI, Jed, The Role of Explanation in Symbolic Computation, *in* INADA, N. and SOMA, T., The Second RIKEN International Symposium on Symbolic and Algebraic Computation by Computers, World Scientific, Philadelphia, PA, 1984.

[8] De SMEDT, Koenrad, Object-Oriented Programming in FLAVORS and CommonORBIT, *in* HAWLEY, R., (ed.), Artificial Intelligence Programming Environments: 157–176, Ellis Horwood Limited, 1987.

[9] CALMET, Jacques, Intelligent Computer Algebra System: Myth, Fancy or Reality? *in* JANSSEN, R., (ed.), Trends in Computer Algebra (Lecture Notes in Computer Science 296):3–11, Springer-Verlag, 1987; CALMET, Jacques, TJANDRA, I.A., and BITTENCOURT, G., An Environment for Mathematical Knowledge Representation, SIGSAM Bull., 24(3):47–48, 7/90.