This is an excerpt from a refereeing process in 2003, shedding light on the decision, to build a prototype of the front-end for the existing math-engine and to postpone research and publication.

We present this excerpt in order to show the opinion\(^1\) of an expert about a proposal for re-engineering algebra systems to 'transparent systems'.

1 Referee C

1. Good points

(a) The grand scheme of this proposal is certainly a good one. At present a person wanting to solve a mathematical problem must translate the problem from the everyday language in which the problem arises to mathematical language. Every teacher knows that a student may be able to solve a mathematical problem such as "find the maximum of this function" but the same student will be unable to solve a problem which is stated in words and which leads to the same mathematical problem.

So one of the aims of this project is to provide a new layer of language that allows computers to assist in the translation of problems into mathematics. Then the traditional programs that solve mathematical problems can start their work.

(b) The breaking out of mathematical methods. At present the major computer algebra systems contain a lot of knowledge, and much of this knowledge was created by them and has never been published in any form. The knowledge is not available to the general public.

The proposal refers to reference [Buc00a] which is Buchberger’s article on computer algebra and mathematics, in which he argues that computers have created new and difficult mathematical problems. The reference is out of date, since the report has now been published elsewhere, suggesting to me that the proposal was written some time ago and not properly revised.

Thanks for the hint on the reference, we used several times without having updated !

The main point, however, is that the proposal points out that a vast amount of mathematical knowledge is hidden in computer algebra systems. This I agree with.

The proposal suggests that we should be braking up the computer algebra systems into sub-systems, in which mathematical knowledge is displayed.

(c) The use of mathematical systems that explain themselves. The proposal talks about systems that show users how they are "thinking" or in less emotional terms, they display the steps they follow.

---

\(^1\)The referee is cited in italic and indented, the answers of the applicant (ISAC) is roman.
... and which show users the knowledge (theories, problems and methods) underlying and guiding their steps. Thus the knowledge becomes accessible from the point of view of application, which may be helpful and complementary to the deductive point of view.

2. Weak points

(a) Although I agree that a system that encoded problems would be a great step forward, I do not see any evidence in the proposal that the authors know how to start developing such a system. On page 9, we see a sample of what would be the goal of the research, but I see little that convinces me that the proposer really has an idea of how to do this.

The according references given in the proposal were [Neu99a, Neu99b, Neu01b], i.e. technical reports and a doctoral thesis. These references are together several hundreds of pages not brought to the point required for the proposal — we understand that they have been disregarded for the referees report.²

ZSACs knowledge-interpreter, implemented to the extent as described in the proposal, was the basis for the practical part of two diploma theses cited as well in the proposal: [Kar02] implemented the algebraic operations on multivariate rational terms, integrated them into ZSACs rewriter, and implemented ’reverse rewriting’ to explain the steps of calculation (which cannot be done by rewriting). [Lan03] gave a proof of concept for ZSACs problem-handler by implementing a hierarchy of elementary equations, which features automated refinement to the appropriate type of equation (i.e. problem).

If the user had a “track record” of system development, then I might consider that it is worth the gamble of letting the principal investigator do his best. The proposer claims to have designed ZSAC, but where is the documentation that he has implemented such a system ?

(b) The specific proposals listed are much more traditional than the radical overall scheme. All the ”TheCx” proposals are traditional computer algebra topics. This gives them a higher chance of being completed, but where is the publishing record of the proposer ?

We are grateful for the confirmation that the subtasks ”TheCx” have some chance of being completed. Part of the concepts is presented in [Neu01a]. ³

(c) The idea that mathematical theorems and algorithms can be liberated from the systems opens up a Pandora’s box. First, there are now systems that are already working on these ideas. In the jargon of computer algebra systems, we are talking about ”single stepping”: There are already calculators on the market offering single stepping, and there are others in beta test (sorry I am not allowed to say more because of non-disclosure agreements).

² TODO: Now there are the following references in high-quality conference proceeding . . .

³ The overall scheme the subtasks ”TheCx” follow, however, had not been published in conference proceedings on computer-mathematics at the time of writing the proposal. TODO: Now there are the following references in high-quality conference proceeding . . .
Nevertheless, thanks for the hint. So far we only knew about considerations in the DERIVE group hinted at in [Ric02]. One of the systems gone public in the meanwhile is at http://www.calc101.com — it is not comparable in interactivity I$\SAC$ achieves by specific interpretation of the separated language layer for methods (not yet contained in the proposal).

\((d)\) The weakness of single stepping, or showing re-write rules, is that modern computer algebra systems use algorithms that cannot be described as re-writing. To factor an integer these days, one uses Pollard rho algorithm or elliptic curve methods. These cannot be explained to a high school student using a single stepping system. In integration, the integration of rational functions is done completely differently in a computer system. High school (university) students use partial fractions; computer systems use the Rothstein-Trager-Lazard-Rioboo algorithm, which uses resultants. Polynomials are factored using Hensel lifting; students know nothing of this.

We are aware of the issues of opening up this Pandora’s box (i.e. computer algebra functions): In C.2.a we already mentioned the cancellation of multivariate rational terms, which cannot be done by rewriting. Thus [Kar02] implemented that using ‘modular methods’ shifting the problem into finit fields following [Win96]. And a newly introduced ‘reverse rewriting’ generates stewise rewrites as explanations\(^4\). (The proposal documented this fact on p.10 in bold face together with the reference [Kar02].)

The algorithms mentioned above may be implemented by I$\SAC$ in a second go as alternative methods\(^5\) for advanced students: I$\SAC$s first interest is education and consequently I$\SAC$ will implement the elementary algorithms taught in elementary math courses first. For instance, I$\SAC$ will actually integrate rational functions using partial fractions, i.e. we are planning to decompose Risch’s algorithm into subproblems and to arrange them in a hierarchy analogously to the hierarchy of equations done in [Lan03].

3. Conclusions

\((a)\) The overall ambitions of this project are grand and useful.

\((b)\) The specific projects proposed are attainable, although I think the idea that diploma students can produce the desired results is risky, or optimistic.

So far we were lucky, as the two diploma theses [Kar02, Lan03] show. The subtasks defined are modularized such that a drop-out is no risk for the forthcoming project.

\((c)\) There is a sense in which the commercial systems are getting to similar ideas already. This does not mean that a new person cannot contribute, but the proposer must realize that the field is not a virgin one.

\(^4\)TODO Now this technique, which evoked interesting new questions in term rewriting, is documented in . . .

\(^5\)We do not know how to automatically generate explanations for steps of the Pollard rho algorithm or elliptic curve methods or the Rothstein-Trager-Lazard-Rioboo algorithm etc. But there are ideas (not yet published, again) how to do that for the ‘modular methods’ in cancelling multivariate fractions.
We are looking forward with great interest to what the commercial systems will show up with.

(d) One of the characteristics of computer algebra systems has been that they try to be all things to all mathematicians. Thus they try to incorporate the most advanced methods and theories from mathematics, so that they can get good reviews from "the experts" while at the same time, they try to sell to high school students. I think it may be that computer systems will have to choose between education and research. This means the proposers must also decide where they want to go.

This decision has been made at the beginning of ISAC as mentioned in our remark to 1(d) above.

It also means that the idea of breaking out knowledge may mean that the methods are not suitable for teaching.

We agree; there are already comments on this in C.2.a and C.2.d.

The proposer has a big advantage over most "developers" of computer algebra systems. He comes from a background of actually using the programs. This has the advantage of giving him the direction to work in. However, there is no documentation that the proposer has actually written any code for a computer algebra system.

References


