Multi-Paradigm Programming through Graph Rewriting: GR/H/41300: Additional Information

Dr. J.R.W. Glauert
School of Information Systems, UEA, Norwich

The above grant application was considered recently by the Systems Architecture Committee. It was given a high alpha rating, but narrowly missed funding. I ask that the application be considered again in the forthcoming round.

The Secretary to the committee has passed on some detailed comments made by the committee. I enclose some additional information in response to these comments, with the aim of overcoming any remaining reservations about the merit of the proposal.

Relationship with ESPRIT BRA 3074: Semagraph

The applicant is a grant holder for the ESPRIT II Semagraph BRA. It is clear that the work proposed here is within the scope of Semagraph whose full title is "The Semantics and Pragmatics of Generalised Graph Rewriting". However, the scope of Semagraph is very broad, and in practice the substantial progress made by Semagraph has been achieved by focussing on the properties of a restricted form of graph rewriting known as Term Graph Rewriting (TGR).

In contrast, the present proposal is concerned with a class of generalised graph rewriting systems well outside the scope of TGR. Work on more extended notions of graph rewriting within Semagraph was undertaken primarily by Glauert and Papadopoulos and concerned graph rewriting models of logic programming. Some early studies of process notions have been undertaken recently, as outlined in the next section, providing a firm foundation for the present proposal.

The original Semagraph grant was extended from December 1991 to June 1992 with some limited additional funding and an application was made for a follow-on project under ESPRIT III. Funds will be granted for a Working Group to continue exploration of extended TGR systems. The focus will be more on theoretical aspects than on pragmatics due to the nature of a working group.

The added value from the current proposal arises because it builds on work started within Semagraph I which cannot be supported by Semagraph II. I will be involved in the Semagraph II Working Group which represents world-class expertise in the theory of graph rewriting, but with SERC funding I will be able to make a substantial attempt to explore the practical potential for using graph rewriting to integrate multiple language styles.

Recent Developments

The proposal outlined the emerging class of languages which integrate functional and process formalisms. Such languages provide a semantically rigourous framework for programming in a functional style while retaining the ability to program with non-determinism, state and input-output side-effects. Explicitly concurrent systems may be programmed.

I have developed a translation of CCS-based process formalisms, with both synchronous and asynchronous communication, into a much simplified graph rewriting notation. This notation seems a good candidate for the core rewriting model required by this proposal.

A paper "A New Process Model for Functions", Glauert, Leth, and Thomsen, describing the translation of the lambda calculus to such a notation was presented at the Semagraph'91 Symposium, in December (proceedings to be published by Wiley). I will present the paper "Asynchronous Mobile Processes and Graph Rewriting" at PARLE'92 (proceedings to appear in Springer LNCS). Through joint work at ECRC a translation for a full language integrating functional and process notations has been developed.

The graph rewriting model used is very simple and has some clear similarities with the continuation-passing style (CPS) which has proved successful in the implementation of Standard ML as described in "Compiling with Continuations", Appel, MIT Press. Communication and concurrency will require new techniques, but there is strong evidence that the bulk of the code generated for an integrated language can exploit these proven methods.

Scope of the Proposed Project

The committee expressed concern that the proposal was very ambitious and had some doubts about the feasibility of making substantial progress within the chosen timescale.

The project divides into two parts with the exploration of sequential and parallel implementation techniques. The previous section of these notes reports good progress towards the first project objective of identifying a core rewriting model. Combining this work with the state-of-the-art in sequential implementation techniques makes a serious sequential implementation attainable.

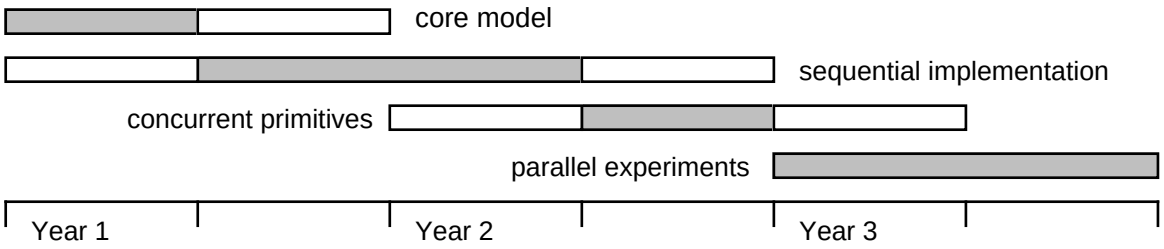
As discussed in the full proposal, the position is less clear when considering parallel implementations. There have been many recent developments in terms of announced hardware (CM5, KSR1, DEC Alpha) but little in the way of a consensus about implementation techniques for languages with dynamic patterns of parallelism. For this reason, the objectives of the proposed project are more modest in this area.

The decision to request a single RA post reflected advice on an earlier, unrelated application, which was rated in much the same way as this proposal. On balance, the ideal size of team would probably involve one RA(1A) to concentrate on language issues and techniques, and one RA(1B) to concentrate on implementation work. Experience with major ESPRIT projects makes me wary of the management load of larger teams.

The proposal requests one RA(1A) appointed at Point 4 on the scale. A good deal of work would be required from this RA in terms of both research and implementation work. If the committee believes that a single RA would be sufficient, it might be easier to attract someone of the required calibre if funds were provided for appointment at a higher scale point.

Project Plans

On reflection, it is clear that the proposal contained too little detail about how the project will be undertaken. The proposal suggests four work packages. The plan below suggests how they might be phased, with shaded periods indicating peak activity:



If two RAs were available, the RA(1B) would concentrate on the basic sequential implementation and parallel experimental work, under supervision of the proposer, with research on core model design and implementation techniques handled by the RA(1A).

It has been explained that the basis of the core model and sequential implementation work is in place. The initial focus for the sequential work will be CPS, using C as an intermediate language. SML will be used as implementation language. Facile will be the primary high-level language to be implemented. This is appropriate as Facile is realised as an extended form of SML.

The process models investigated are based on CCS and use communication extensively. Optimisation at the graph rewriting level eliminates most communication; the remaining communication is mainly point-to-point, and the intuition is that sending a message can be handled as a slightly complicated function call or coroutine switch.

For parallel experiments, work will start from ideas of Dally and Wills. Here again, the approach is to assume that communication local to a processing agent can be optimised, while messages sent to remote agents act as light-weight process descriptors on the receiving agent.

Clearly, in a research project it is not possible to be sure of how progress will be made. However, the basis for the work proposed is already in place, and indeed my continuing work while on study leave means that the project should be ready for a flying start once funded.

John Glauert
March 1992