SCIENCE AND ENGINEERING RESEARCH COUNCIL
RUTHERFORD APPLETON LABORATORY

INFORMATICS DEPARTMENT

SYSTEMS ENGINEERING DIVISION NOTE 43

SE Project Status Reports for Engineering Board Review

issued by
D A Duce
14 June 1989

DISTRIBUTION:
R W Witty
G A Ringland
B Ritchie
J C Bicarregui
J R Kalmus
B M Matthews
S K Robinson
C Reade

This is a revised version of SED Note 31 incorporating corrections of fact to the IPSE 2.5 report and correcting typographical errors in the other two reports.

(see next page)

IPSE 2.5

J. Bicarregui, D.A. Duce and B. Ritchie

Systems Engineering Division, Rutherford Appleton Laboratory

1. Background

The IPSE 2.5 project is being carried out under an Alvey contract awarded to a consortium comprising the following organizations:

STC Technology Limited
International Computers Limited
University of Manchester
Dowty Defence and Air Systems Limited
Rutherford Appleton Laboratory
Plessey Research Roke Manor Ltd
British Gas plc

The project began in October 1985 with three initial collaborators (ICL, STC and the University of Manchester), and had a successful first review with the Alvey Directorate in July 1986. RAL applied to join the project from April 1986 and received an allocation in July 1986. Plessey, Dowty and British Gas, subsequently joined the consortium. IPSE 2.5 is the largest project in the Alvey Software Engineering Programme with a total budget of around £10M. RAL receives 2.5MY per year from the project included in a total budget of £400K.

The IPSE 2.5 project [9] aims to demonstrate that computer based technology in the form of an integrated support environment can provide significant assistance towards addressing some of the major issues in system development. The project is concerned with IPSEs which integrate the management and control of the development *process* with the development activities themselves and particularly with development activities based on the use of *formal methods*. The name of the project derives from the fact that the IPSE being constructed lies somewhere between a second generation IPSE (using database technology) and a third generation IPSE (using knowledge base technology).

The project is concerned with a number of major issues. A strategy of separating major concerns is adopted in the project, giving rise to a small number of relatively separate threads or ''Themes''. The Themes are as follows:

Theme A	Non formal methods of systems development
Theme B	Formal methods of system development
Theme C	Advanced aspects of formal reasoning
Theme D	Concerns of management
Theme E	Integration

The project works to a set of sub-objectives, each associated with a particular Theme.

Theme C is the responsibility of the University of Manchester (Professor C.B. Jones), and RAL. This is the only theme to which these partners contribute. The project is managed by STC Technology Ltd. Ultimate authority in the project resides with the Project Review Board, on which Dr

D.A. Duce represents RAL. Technical work on the project is undertaken by J.C. Bicarregui and Dr B. Ritchie.

2. Objectives

The objective of the IPSE 2.5 project is:

> "To produce an IPSE and evaluate its effect on the productivity and quality of systems development as measured by the costs of production and maintenance by providing:
>
> (1) support for rigorously defined development processes which integrate management activities and development activities;
>
> (2) support for formal methods;
>
> (3) support for information reuse;
>
> (4) support for the transition in methods from the non-formal to the formal."

It is intended that IPSE 2.5 should be a generator of IPSEs that can be "instantiated" to provide application or method specific support.

Theme C is primarily concerned with research in the area of computer support for those activities of a formal reasoning nature which arise in the use of formal methods. The objective of theme C is:

> "To produce a formal reasoning system developed on Smalltalk and to provide a minimal proof obligation generator to provide tasks for proof. The formal reasoning system should be usable by software engineers trained in appropriate formal methods. The generic system will be tested by a VDM instantiation."

The work of theme C falls under the following headings:

(1) Investigate and evaluate existing theorem proving systems to determine the requirements for a prototype Formal Reasoning IPSE (μral).

(2) Build an experimental system to evaluate the use interaction with a proof system in order to refine the requirements for the μral.

(3) Build a formal reasoning system formed from a Proof Obligation Generator, a theorem prover, and facilities to support animation of specifications.

The formal reasoning system being constructed by Theme C, will not be integrated with the system being produced by the remainder of the project. Integration will be undertaken by the partners in Theme E, beyond the timescale of the present project.

The work at RAL has concentrated on the requirements for μral in the theory store area and the design and implementation of the proof obligation generator.

3. Progress to Date

3.1. Requirements

Technical work on the project commenced at RAL in July 1986 under Dr C.P. Wadsworth. J.C. Bicarregui and Dr B. Ritchie joined the project in October 1986. Dr Wadsworth joined the Transputer Initiative in January 1987 at which point Dr D.A. Duce took over managerial responsibility for the project, but, unlike Dr Wadsworth, had no technical role in the project. RAL and the University of Manchester operate as a single team in the project. Technical leadership is provided by Professor C.B. Jones.

Formal methods of software development are methods which use mathematical specification to capture the requirements of a design, and verification of design steps to yield implementations that are correct with respect to this specification. A development may involve several layers of

specification and the construction of formal proofs of various properties of them and of the relationships between the specifications is a vital part of the method. Each design decision - a choice of a particular representation of some abstract data type, or an implementation of an implicit function or operation - gives rise to certain *proof obligations* to justify that decision in terms of the original specification. There are also feasibility obligations upon the specifications themselves. Simple type checking can be performed mechanically, but when data type invariants are involved, proof obligations must be discharged in order to show that terms in the specification are properly typed. Doing proofs can also be useful in improving understanding of a problem by highlighting subtle consequences; sometimes the structure of proofs may suggest cleaner specifications of the problem.

The main thrust of the University of Manchester and RAL part of the project is to provide machine support for the formal reasoning component of an IPSE which will support formal methods of software development. The intention is to build tools which will enable a user (software engineer) to construct proofs at a workstation. It is believed that modern workstations with window and mouse interfaces such as the SUN 3, make it possible to design proof assistants which are much more usable than earlier tools developed around "glass teletype" interfaces. The project chose to explore proof assistants rather than fully automated theorem proving tools because the former allows the intuition of the human user, who has an insight into the problem and a feeling why the result is true, to guide the proof process along the right lines.

The initial stages of the RAL work were concerned with familiarization with the project and with formal methods in general. A number of concept papers were written by the project to establish the ground on which the project would build. Dr Wadsworth made a substantial contribution to the theorem proving concepts paper [1]. Dr Ritchie contributed to a review of existing theorem proving systems [2,3] in particular providing a review of the Edinburgh Interactive Proof Editor (IPE), which formed the subject of his Ph.D. thesis.

The University of Manchester constructed an experimental formal reasoning tool called Muffin. This was undertaken as an experiment in user interface design for a theorem prover, and embodied a very simple logic (propositional logic). RAL commented on the specification of Muffin. Muffin's successful implementation in Smalltalk demonstrated the usefulness of Smalltalk 80 for highly interactive formal reasoning systems and motivated its use for μral.

3.2. Logical Frame

One of the requirements for the formal reasoning component of the IPSE (μral) was that it should be capable of supporting a variety of formal methods and specification languages, for example VDM, Z and Hoare logics. Different methods and languages are, however, based on different logics: VDM uses Logic of Partial Functions (LPF) and Z uses classical set theory. μral must therefore be able to support a wide variety of logics, so that it can be instantiated to cater for a particular formalism. The approach taken in μral, and other projects, is to provide a *logical framework* upon which may be defined the notation, axioms and deduction rules of particular logics. Tools based upon the logical framework can then be used for theory development within any instantiated logic. The particular logical framework (FSIP) used in μral was designed by Dr P. Lindsay at Manchester. Mr Bicarregui provided input to this.

3.3. Theory Store

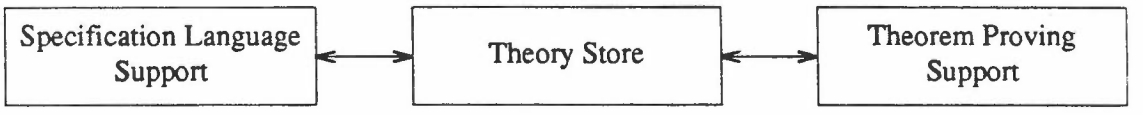
A theory of logic is a formal system of rules for the construction of valid forms of argument. The formal system consists of primitive assumptions and rules of inference in some formal language. A proof of a fact is a chain of deductions, each justified by other known facts which are either primitive (assumed *a priori*) or derived (proved from other facts). In this way we can build up a large number of theorems.

Not all results will be useful in all contexts. It may be necessary to postulate an axiom for use when reasoning about some particular objects which is not applicable in other cases, or a particular symbol may be used to represent different things in different contexts. There is thus a need to organize the "database" of results in such a way that those results which deal with the same symbols under the same interpretation are available together. The collection of results which are valid in a particular context is called a *theory*, and as the database is extended, a hierarchy of theories upon theories can be built. The construction of such a *theory store* allows reuse of results and hence raises the level of reasoning in proofs beyond the level which the primitive assumptions are stated, to the level of previously proven results.

RAL were responsible for determining the requirements for theory structuring for μral. The work proceeded through examination of usage scenarios for a hypothetical system supporting construction and refinements of specifications in VDM and other notations. The results are reported in [4].

3.4. μral Prototype and VDM Support Environment

The user interface work with Muffin, the FSIP logical frame and the theory store requirements formed the starting point for the design of the μral prototype currently being constructed. The initial μral architecture was roughly:



μral was split into a left hand side (LHS) providing specification language support, and a right hand side (RHS) providing theorem proving support, communicating through the theory store. It was eventually decided that the theory store was too complex an interface and a simpler interface was designed and the theory store was incorporated into the RHS. The names LHS and RHS are, however, enshrined in μral vocabulary.

The theme structure of IPSE 2.5 envisaged that theme B would essentially provide LHS tools. However, due to circumstances external to the project, a reorganization of the consortium took place in the theme B area as a result of which theme C became responsible for providing a minimal LHS for use within the theme. Subsequently RAL took responsibility for this work.

The University of Manchester are responsible for the specification and construction of the RHS of μral. RAL were involved in reviewing the RHS specification. The LHS specification and construction are the responsibility of RAL and the University of Manchester have reviewed this. A prototype instantiation of μral to support specification and development in VDM is being built. The main aims in this are:

(1) to demonstrate that the generic μral can be instantiated to support a particular formal method;

(2) to generate "interesting" proof obligations upon which the theory store and proof assistant can be exercised.

An environment for formal software development would typically provide tools specific to the method:

(1) tools for building and storing specifications, programs and other method-specific constructs;

(2) means of recording the development process (for example, linking specification to implementation);

(3) means of constructing corresponding theories in the logical system;

(4) means of generating proof obligations and tracking their progress.

The project is not constructing such a fully-fledged environment; instead attention is being concentrated in the interface between VDM specific tools and the generic μral.

μral is being instantiated for a subset of the emerging BSI standard for VDM [5]. The design principles and proof obligations are taken from [6].

VDM provides a richly typed specification language, with constructors for set, sequence, map, and compound (record) types, together with subtyping (by giving invariants on a type). Data types suitable for the description of the intended system may be constructed with these type constructors. The language includes a notion of state: operations may affect the state and can be non-deterministic; functions are purely applicative and cannot alter the state. Both functions and operations may be specified implicitly, abstracting from any commitment to algorithmic implementation details by giving logical predicates to specify their pre- and post-conditions.

Development steps revolve around finding ''more-concrete'' representations for the data types of the specifications, (called *data reification*), and redefinition of the functions and operations to correspond to the new data types (*function/operation modelling*). The method requires the discharging of certain proof obligations arising from these steps, for example with each data reification it has to be shown that the concrete data type can be considered to be an ''full and faithful representation'' of the abstract. There is also a set of operation decomposition rules used to justify the (possibly partial) implementation of an operation as the composition of other operations. Composition includes a variety of constructs such as sequencing, if-then-else, and loops. A reification is considered to be formally verified when all of the associated proof obligations have been discharged.

The VDM support environment has now been designed and specified [7]. An implementation incorporating a subset of VDM was demonstrated to the Alvey Directorate at a management review of the IPSE 2.5 project on 17 January 1989. The environment provides the following facilities:

(1) BSI VDM specifications may be entered;

(2) function/operation refinement steps may be described;

(3) the system generates proof obligations corresponding to refinement steps and translates them to the syntax used by the RHS, and adds them to the theory store in the VDM instantiated RHS.

The method of handling operation decomposition is handled using Hoare-like annotations. This is described in [8].

3.5. Miscellaneous

Dr Ritchie and Mr Bicarregui have also contributed to the design and development of BSI VDM. Dr Ritchie is Document Editor for the protostandard.

Dr Wadsworth initially, and Dr Duce subsequently, have represented RAL on the Project Review Board, the top level management body in the project.

Dr Duce has contributed to the IPSE 2.5 Requirements Document.

4. Motivation for the Project

Informatics Department's first involvement in IPSE 2.5 was motivated by the need to keep abreast of advances in the methodology for constructing software. The advent of the Engineering Board's EASE (Engineering Applications Support Environment) Programme has provided further motivation for the work. EASE is the major facility and service operated by Informatics Department. EASE resources are divided roughly 50% to the provision of facilities for engineering

researchers and 50% to the promotion of awareness amongst eningeers of relevant developments in computing.

IPSE 2.5 is relevant in two ways to this programme. First IPSE 2.5 is about providing an environment to improve the ability to develop software and to manage the development of software. Second EASE itself can be seen as a form of IPSE which is tailored not just to software development but to provide environments appropriate to researchers in different engineering disciplines. This is an example of a form of genericity which IPSE 2.5 is seeking to provide

The project links Informatics Department with the UK software development and IPSE communities thus enabling the Department to keep abreast of developments and trends. The project also enables Systems Engineering Division to advise EASE in the short term on developments which should feature in EASE in the longer term.

References

1. C.B. Jones, P.A. Lindsay and C.P. Wadsworth, *IPSE 2.5 Theorem Proving Concepts Paper*, IPSE 2.5 document 060/00021, 1986.
2. P.A. Lindsay, R.C. Moore and B. Ritchie, *IPSE 2.5 Review of Existing Theorem Provers*, IPSE 2.5 document 060/00047, 1986.
3. P. Lindsay, Theorem Proving Review, Software Engineering Journal, 1988.
4. B. Ritchie and J.C. Bicarregui, *Theory Store Requirements Study*, IPSE 2.5 document 060/00063/1.4, 1987.
5. *VDM Specification Language Protostandard*, BSI IST/5/50 document 40, 1989.
6. C.B. Jones, *Systematic Software Development using VDM*, Prentice-Hall International, 1986.
7. B. Ritchie and J.C. Bicarregui, *The μral LHS Spec*, IPSE 2.5 document 060/00144/2.1, 1988.
8. B. Ritchie and J.C. Bicarregui, *Towards a Data Model for Operation Decomposition*, Rutherford Appleton Laboratory, 1988.
9. R.A. Snowdon, Scope of the IPSE 2.5 Project, document 060/00002/4.2, December 1988.
10. C.B. Jones and B. Ritchie, *IPSE 2.5 Interim Review: Choose Instantiation Language*, IPSE 2.5 document 060/00110, 1987.
11. B. Ritchie and J.C. Bicarregui, *IPSE 2.5 Theory Store Analysis*, IPSE 2.5 document 060/00111, 1987.
12. J.C. Bicarregui and B. Ritchie, *Providing Support for the Formal Development of Software*, in: Proceedings of the International Conference on Systems Development Environments and Factories, Berlin, 1989 (to appear).

Use of Knuth-Bendix Techniques for Theorem-Proving

B.M. Matthews, J.R. Kalmus and D.A. Duce

Systems Engineering Division, Rutherford Appleton Laboratory

1. Background

This project arose from an application made by Dr R.W. Witty, at that time Head of Software Engineering Group, to SERC's Computer Science Sub-Committee in April 1986. The proposal arose from the interests of Dr A.J.J. Dick, an Atlas Research Fellow working in the group. Dr Dick had worked on automated equational reasoning at Imperial College London under the supervision of J. Cunningham. Dr Dick started the development of ERIL, Equational Reasoning: an Interactive Laboratory, whilst at Imperial College and continued this work on joining RAL.

The project is of 3 year's duration and provides 1 MY of effort per year.

The research was initially carried out by Dr Dick alone. Subsequently Mr Kalmus joined the group and spent some of his time on the project. Mr Matthews became involved in the project in August 1988, and following the resignation of Dr Dick from the laboratory in November 1988, Mr Matthews has assumed responsibility for the technical work under the supervision of Dr Duce.

ERIL naturally complements the activities in the IPSE 2.5 project in that the latter is providing an environment for reasoning about specifications in so-called constructive notations, whilst ERIL provides a tool for reasoning about specifications in property-oriented or algebraic notations. The two approaches are also complementary in that IPSE 2.5 is providing a tool which is very much a proof assistant, relying on the human user for guidance in the proof process, whereas ERIL is closer to the fully automated style of theorem prover, whilst supporting some degree of interactive control.

2. Objectives

The general aim of the project was to compare the performance of the ERIL system with established resolution theorem proving techniques, and to assess the future of this approach to fully-automated theorem proving.

The research programme envisaged was:

1) Extending ERIL to meet further theoretical requirements. In particular, the following were stipulated.
   - i) Implementing an Associative/Commutative unification and completion algorithm.
   - ii) Implementing new termination orderings on rules.
2) Apply the ERIL theorem prover to case studies to assess the capabilities of the Knuth-Bendix approach. The main aims were:
   - i) Compare results with Resolution based theorem-proving methods.
   - ii) To gain experience over a wide area of problem domains in theoretical computer science, for example processing of algebraic program specifications, solution of domain equations etc, with a view to identifying design criteria for a new theorem proving tool.
3) To produce a feasibility report and design study for a fully-fledged theorem-proving tool.

3. Progress

3.1. Extending the ERIL Equational Reasoning System

The ERIL system [Dick87] is a highly configurable, order-sorted, equational reasoning system based on the completion technique first explored in. [KnuBen70] It is the configurable nature of ERIL, coupled with order-sorted typing that distinguishes ERIL from other tools for equational reasoning such as REVE.

The system has been extended gradually to reach the current version known as 1.6b. Some of the extra features such as the Unfailing Completion technique [HsiRus86], while not identified in the initial proposal for this project, nevertheless significantly enhance the power of ERIL and make it a more flexible and attractive tool for equational theorem proving.

3.1.1. Associative-Commutative Unification

A major drawback of the completion technique is its inability to directly handle commutative axioms. Several extensions to the original idea have been suggested in the literature, but it remains a technically very difficult area. However it is essential to be included within ERIL if it is going to be used as a practical theorem prover for the full first-order predicate calculus. This is because first-order logic is by its very nature commutative.

As a consequence of the technical difficulty of AC-Unification, a complete implementation is not yet included in ERIL. A restricted form of completion modulo a set of equations as outlined in [Hue80], was implemented, however, this method is not general enough to perform first-order logic. A variant of the Knuth-Bendix algorithm due to Jieh Hsiang (Stony Brook, New York) was implemented which overcomes some of the cases where the original algorithm failed because an axiom could not be oriented as a rule. Experiments suggested that this method was not satisfactory for handling permutative axioms.

Dr Mike Lai at Royal Holloway and Bedford New College has been studying the literature in this area with a view to developing a new technique for AC-unification, and has come up with some significant results [Lai89]. This technique should significantly decrease the amount of computation required to produce a canonical set of equations modulo the Associative and Commutative axioms. It is intended to implement this work.

3.1.2. Orderings

Several more termination orderings have now been successfully implemented. These include a version of the Knuth-Bendix Ordering that allows the user to assign weights to function symbols (userKBO) three versions of the Recursive Path Ordering, [Ders82] one which requires the user to assign the function precedence in advance (userRPO), another which automatically determines the appropriate precedence (RPO) and another which also determines appropriate function status (RPOS). An automated version of the userKBO algorithm was developed during the project by Dr Dick, Mr Kalmus, and Dr Martin (Royal Holloway and Bedford New College). A paper has been submitted for publication [DicKalMar88].

3.2. Case Studies in Theorem Proving

The user manual for the ERIL system [DicKal88] contains several examples of the use of the ERIL system in a variety of problem domains. These include Completion Modulo a set of equations using Huet's method; Unfailing Completion [HsiRus86]; Theorem Proving in Horn-Clause Logic using Paul's method [Paul85b]; Equational Narrowing for solving equations in an canonical equational theory [Hul80]. These examples show that the ERIL system is a suitable and adaptable vehicle for a wide range of established techniques in equational reasoning.

In addition Mr Matthews has carried out a study of the suitability of using the existing ERIL system for more general theorem proving [Matt88]. In this work, the flexibilty of the ERIL system was demonstrated by allowing the user to configure the system as he or she wishes, in order to make the theorem proving process more flexible and efficient, and under greater user-control. This was largely carried out using the Horn-Clause method of Paul mentioned above. Also in this dissertation, there is an attempt to use Hsiang's method for First-Order Predicate Logic [Hsi85a]. Hsiang's method is unsuitable in ERIL as it stands as it requires A/C unification. However, it may be possible to simulate it using the Unfailing Completion

method. This attempt was not at the time successful due to the very large search space generated by the unfailing method. Some suggestions were made as to possible modifications to the ERIL system which may improve its configurability in a new version. Some of these suggestions are being taken up into a new design.

3.3. A New Implementation of the ERIL System

The existing ERIL system has been developed over a number of years. It has many good features and the interface to the user has been widely praised. However, the system remains slow and the practical use has revealed many deficiencies in its design: it was only ever designed to be a prototype. As a consequence of this, Dr Dick decided to embark on a reimplementation of the ERIL system. This would encompass many of the changes and extensions suggested in the work mentioned above, and also the theoretical work being developed by Mike Lai (AC unification) and Phil Watson (order-sorted rewriting) at Royal Holloway and Bedford New College. However, this new version of ERIL would not be of 'production' standard: rather it was to be considered as an extended runnable specification. Clarity is the top priority in this version, rather than efficiency. It is written in Prolog like the first version of ERIL for the ease of programming and introducing new features, not in C as was first suggested for the extra efficiency it would bring. This new implementation would comprise the aim of a detailed feasability study for a new efficient implementation as set out in the initial grant proposal as an aim of the project.

Large parts of the design are complete, including an extensible user interface, the central module manager, input and storage modules and many components of the inference engine.

When this part of the project is completed, it should be possible to carry out first-order theorem proving effectively. For example, it is suggested that the Schubert Streamroller problem [Sti86] should be tackled with the ERIL system. This is a classic problem for automated theorem provers. It's solution using resolution methods often leads to a computational explosion (thus its nickname of 'Steamroller'). However, in a order-sorted system such as ERIL, its solution should be more straight-forward.

3.4. ERIL Distribution

The ERIL system has been distributed to some 30 sites worldwide. Courses on ERIL have been given at the Polytechnical University of Catalonia, Barcelona (Spain) and the University of Minhio, Braga (Portugal). Plans are in hand to hold an ERIL course at RAL in October/November 1989, in conjunction with the BCS Formal Aspects of Computer Science (FACS) specialist group.

3.5. Other Activities

A very profitable collaboration has developed between RAL, Dr Martin's group at Royal Holloway and Bedford New College (RHBNC) and Dr Thomas' group at the University of Glasgow. Theoretical work undertaken at RHBNC and Glasgow is now being put to practical use in ERIL. The RAL implementation work is a good test of the practical usefulness of the theoretical results.

A UK Term Rewriting Group has been established by Dr Dick and Dr Martin, with support from SERC's Logic Initiative. Several meetings have been held.

Mr Kalmus organized a highly successful BCS FACS meeting on term rewriting held at Bristol University during 1988.

4. Motivation for the Project

It has already been noted that the ERIL project is complementary to the IPSE 2.5 project in that ERIL provides a tool for manipulating specifications given in an axiomatic style, whilst IPSE 2.5 will provide tools for handling specifications in the constructive style. Both approaches to specification are important. The Department needs to be aware of work in both areas in order to produce sensible advice on what should be provided in EASE in the longer term.

Equational reasoning is a field in which there has, until fairly recently, been little UK activity. Much of the work in the field stems from research groups in the USA, France and Germany. The ERIL project has done a certain amount to redress the balance, through the creation of an active UK Term Rewriting Group which has members from Royal Holloway and Bedford New College, the University of Glasgow, Hatfield

Polytechnic and other sites. Dr Dick was instrumental in forming the group and Mr Kalmus has provided invaluable administrative support. SERC's Logic Initiative provided a small amount of financial support to the group.

Within the ISO standards arena, LOTOS and ACT ONE are being standardized as formal description techniques, primarily for describing communications protocols. ACT ONE is an algebraic language, for which ERIL might provide a valuable formal reasoning tool. There is industrial interest in this area, especially from British Telecom.

References

Ders82. Nachum Dershowitz, "Orderings for Term Rewriting Systems," *J of Theoretical Computer Science* **17** pp. 279-301 (1982).

Dick87. A J J Dick, "Order-Sorted Equational Reasoning and Rewrite Systems," PhD Thesis, Imperial College, University of London (1987).

DicKalMar88. A J J Dick, J R Kalmus, and U H Martin, "Automating the Knuth-Bendix Ordering," *submitted to Acta Informatica*, (1988).

DicKal88. A J J Dick and J R Kalmus, "ERIL (Equational Reasoning: an Interactive Laboratory) User's Manual," Rutherford Appleton Laboratory Report, RAL-88-055. (1988).

Hsi85a. Jieh Hsiang, "Refutational Theorem Proving using Term-Rewriting Systems," *Artificial Intelligence* **25** pp. 255-300 (1985).

HsiRus86. Jieh Hsiang and Michael Rusinowitch, "On Word Problems in Equational Theories," Draft Report, Dept of Computer Science, State University of New York (1986).

Hul80. Jean-Marie Hullot, "Canonical Forms and Unification," pp. 318-334 in *5th Conf. on Automated Deduction, Lecture Notes in Computer Science, 87.*, Springer-Verlag (1980).

KnuBen70. Donald E. Knuth and Peter B. Bendix, "Simple Word Problems in Universal Algebras," pp. 263 - 297 in *Computational Problems in Abstract Algebra*, ed. Leech J, Pergamon Press (1970).

Lai89. Mike Lai, "A Peak Reduction Lemma in Rewritings of Term Algebras Defined by Some Associative and Commutative Equations," Submitted to 3rd Conf on Rewriting Techniques and Applications. (1989).

Sti86. Mark.E.Stickel, "Schubert's Steamroller Problem: Formulations and Solutions.," *J. of Automated Reasoning*, (2) pp. 89-101 (1986).

Matt88. Brian M Matthews, "Strategies for Theorem Proving in an Equational Reasoning System," MSc Dissertation, Imperial College, University of London (1988).

Paul85b. Etienne Paul, "On Solving the Equality Problem in Theories Defined by Horn Clause," in *Proc. of EUROCAL 85, Linz Austria, Lecture Notes in Computer Science 203.*, Springer-Verlag (1985).

Theoretical Studies of Emerging Graphics Standards

D.A. Duce

Systems Engineering Division, Rutherford Appleton Laboratory

1. Background

This project started in 1985 and ends in 1989. The resources provided were 48mm of staff effort and funds for travel and subsistence. The project was a sequel to a project funded by the Alvey Programme entitled Specification of the Graphical Kernel System (GKS) [1]. GKS was the first ISO standard for computer graphics. F.R.A. Hopgood, D.A. Duce, J.R. Gallop and D.C. Sutcliffe at RAL made substantial contributions to the development of GKS, Duce, Gallop and Sutcliffe being ISO Editors of the GKS document.

This present project is concerned primarily with the problems of specifying the emerging standards GKS-3D and PHIGS.

2. Objectives

The main objectives of the project were:

(1) to apply formal specification techniques to the GKS-3D and PHIGS proposed standards for computer graphics software;

(2) to assess a range of specification techniques for this type of software;

(3) to characterize the relationship between these proposals and GKS.

The grant application described the following work programme:

(1) **PHIGS** The main problems to be tackled were to give a specification of the PHIGS database and traversal mechanism for displaying structures stored in the database and to discover a structure for the specification which highlighted the links with GKS.

(2) **GKS-3D** The main problems to be tackled here were to produce an outline specification which highlighted the connection between GKS-3D and GKS and to explore the claim that programs written for GKS would run without modification on a GKS-3D implementation.

Changes have been made to this work programme in the light of experience and in response to changes of direction within the ISO projects in computer graphics.

3. Progress to Date

Dr D.A. Duce and Dr M.S. Parsons prepared three papers [2,3,4] for the Eurographics GKS Review Meeting in September 1987. One covered proposals for improved input facilities in GKS. The second was a specification of the polyline, polymarker and fill area output primitives of GKS in the framework repored in the Arnold, Duce and Reynolds paper at Eurographics '87. The third paper recorded the lessons learnt from the GKS specification work so far and areas in which the GKS definition could be improved.

The GKS Review Workshop produced many interesting ideas for directions in which GKS could be simplified and at the same time provide richer facilities. A good example of this was the idea to replace the GKS segment store with a primitive store from which primitives could be selected

using a name set/ filter mechanism.

Many suggestions were made concerning output primitives: the current set of primitives in GKS is unsatisfactory for several reasons. Dr Parsons subsequently looked at more fundamental and structured ways of defining primitives [5]. The underlying philosophy is that the set of output primitives will never be fixed, people will always want to add new primitives to do a specific job, or to exploit particular hardware. To allow for this extensibility, a framework approach for graphical primitives is proposed: as long as new primitives can be shown to fit the framework model, they may be considered to be primitives. The case of poly primitives has been looked at in some detail and a working paper using a functional specification style has been produced [5]. This specification has been partially implemented using Miranda (a functional language developed by Turner at the University of Kent) and PostScript.

The relationship between GKS and PHIGS has received considerable attention. This work culminated in [6]. Essentially PHIGS can be described as a database on top of GKS or GKS-3D. Traversal of this database generates graphical output, which can be described in terms of GKS functions. This simple model does not completely define the relationship beween GKS and PHIGS, there are complications arising from differences between coordinate systems and clipping in the two systems. The control of operator attributes, visibility, highlighting and detectability, is also different in the two standards, which limits the analogy. Reference [15] describes an extension to GKS-3D which incorporated namesets and filters.

Following the PHIGS review meeting in May 1987, a short paper was written which analyzed the PHIGS name set concepts and showed some of its limitations [7]. The results influenced the functionality provided in the incremental spatial search functionality incorporated into PHIGS at that time.

The results of this work have been fed into the development of GKS-3D and PHIGS through BSI as a part of the UK comments accompanying the votes on Draft Proposals and Draft International Standards. Although the UK desire to see a clean model and relationship has not been totally achieved, the RAL research has had a positive impact on the presentation of PHIGS, making the document easier to understand.

The relationship between GKS-3D and PHIGS is one aspect of a broader issue, namely, a Reference Model for Computer Graphics. Insights from the specification work formed the basis of an outline reference model [8] submitted to BSI in February 1988. This was transmitted in revised form to a special advisory group meeting of ISO/IEC JTC1/SC24 (the ISO/IEC committee responsible for standardization of computer graphics) on future policy in April 1988. This meeting came out with a recommendation that the BSI reference model should be developed further, together with a recommendation that a Components/ Frameworks model should be explored as a basis for the development and management of future graphics standards. The idea was that a graphics standard could be constructed from a collection of components set in a framework. Components would include datatypes and operations. A framework is the "glue" which joins components together to form a system and performs management concerned with display and control. The model was seen as providing harmonization of standards through the use of common components and frameworks. The relationship between the PHIGS standard and the PHIGS BR proposal illustrates this idea in that PHIGS and PHIGS BR share a common framework (centralized structure store, traversal of which generates graphical output on workstations), but differ in their choice of output primitive component and attribute component. PHIGS BR uses a richer set of components which take illumination into account.

At a subsequent meeting of the BSI Reference Model Group, it became clear that the BSI Reference Model could be integrated with a component/framework model, essentially by recognizing a connection between abstract datatypes and components. Dr Duce was one of the authors of a revised paper which formed part of the BSI input to the SC24/WG1 meeting in July 1988 [9].

Dr Parsons did some joint reference model work with Dr C.L.N. Ruggles at Leicester University. This started from the recognition that in a graphics system there are two dimensions in which graphics data can be reified (made more concrete). The first is a data reification, the second a graphical reification. A working paper describing their ideas went through several iterations.

Dr Duce participated in the ISO Reference Model meeting held in January 1989, the outcome of which was a reference model [10] closer to the BSI paper [8] than the component/frameworks model, though components and frameworks could be seen as another dimension of structuring to the model.

A review of GKS is now starting within ISO, and the work of the project is being fed into this process. Papers have been written for the March 1989 meeting which describe a simpler structure for GKS and propose the removal of extraneous functionality [11]. Central to this proposal is the explicit idea of a picture in NDC space which the application is constructing, and the notion of a workstation viewing this picture through a filter mechanism based on the PHIGS name set mechanism. Some of the ideas in this originated in this research project. The description of GKS has also been simplified extensively by using more abstract datatypes and functionality. A formal specification of the simplified model has been given [12] which has a particularly simple structure.

A description of the GKS input model using Hoare's CSP notation, has been produced as a result of a collaboration with Drs P.J.W. ten Hagen and R. van Liere at CWI, Amsterdam [13]. The description shows clearly the differences between the different operating modes for GKS logical input devices and gave the authors some new insights into the input model.

It has long been felt that it should be possible to allow user configurable input devices and allow hierarchically structured devices. During a visit to CWI in February 1989, it was found that the CSP input description could be extended easily to describe hierarchically structured devices. Some examples were worked out and the first draft of a paper was produced [14].

The thrust of the project is now concerned with the input model description, transferring insights from the project to the Reference Model work and producing a tidy revision of GKS, including a formal description.

4. Difficulties

The project has experienced staffing difficulties following the resignations of Miss Fielding in December 1985 and Dr Parsons in June 1988. From May 1987 to May 1988 Dr Duce was acting Head of Software and Knowledge Engineering Group which made it difficult for him to devote adequate time to the project.

5. Relationships with Other Groups

The BSI and ISO/IEC computer graphics working groups have provided valuable input to the project and have been a major outlet for the results.

The link with CWI in Amsterdam has been particularly fruitful. The project has good links with the Universities of Leeds, Manchester, East Anglia and Leicester.

6. Motivation for the Project

Informatics Department has made substantial contributions to the development of standards over the last 10 years. SERC has strongly encouraged the use of standards wherever appropriate in its programmes, in order to make research more effective and to facilitate technology transfer from research programmes.

Computer graphics is a technique of major concern to engineers in the design process. Graphics standards are of key interest to EASE, especially as EASE comprises a heterogeneous equipment

environment.

Two questions facing standardization are how to define a standard in a precise unambiguous way and how to know if an implementation conforms to a standard. This project is of relevance to both these key issues.

References

1. D.A. Duce, 'SE024 - Specification of the Graphical Kernel System (GKS) - Final Report to the Alvey Directorate', April 1987.
2. D.A. Duce and M.S. Parsons, 'GKS - Some Lessons Learnt from Formal Specification', Eurographics GKS Review Workshop, 1987.
3. D.A. Duce and M.S. Parsons, 'A Specification of the GKS POLYLINE, POLYMARKER and FILL AREA SET Primitives', Eurographics GKS Review Workshop, 1987.
4. D.A. Duce, 'Extensions to the GKS Input Model', Eurographics GKS Review Workshop, 1987.
5. M.S. Parsons, 'A Functional Specification of Graphical Poly Primitives', Working Paper, Rutherford Appleton Laboratory, 1988.
6. D.A. Duce, 'GKS, Structures and Formal Specification', accepted for the Eurographics '89 Conference.
7. D.A. Duce, 'A Discussion Paper on PHIGS Namesets and Filters', BSI IST/21/2/3: 47, 1987.
8. D.A. Duce, 'Towards a Computer Graphics Reference Model', BSI, 1988.
9. UK Experts, 'Contributions on a Computer Graphics Reference Model', ISO/IEC JTC1/SC24 N4, 1988.
10. 'Information processing systems - Computer graphics - Reference model of computer graphics', document RM/20, ISO/IEC JTC1/SC24/WG1 - Reference Model Rapporteur Group, 1989.
11. K.W. Brodlie, F.R.A. Hopgood and D.A. Duce, 'GKS Review - The NDC Picture', GKS Review meeting, 1989.
12. D.A. Duce, 'Outline Specification of Revised GKS', GKS Review meeting, 1989.
13. D.A. Duce, P.J.W. ten Hagen and R. van Liere, 'Components, Frameworks and GKS Input', accepted for the Eurographics '89 Conference.
14. D.A. Duce, P.J.W. ten Hagen and R. van Liere, 'An Approach to Hierarchical Input Devices', Working Paper, 1989.
15. 'Compatibility between GKS-3D and PHIGS', IST/21/2/3: 15, BSI, 1986.