FORMALWARE PROJECT REVIEW

Date of Review: 21 May 1998

Review Team Members

Dr. R. Nigel Horspool, University of Victoria (Chair)

Dr. Victoria Stavridou, SRI International

Mr. Michael Volker, Simon Fraser University

Dr. James Caldwell (Cornell University / NASA Ames Research Center)

Dr. Paul Sorenson (University of Alberta)

EXECUTIVE SUMMARY

FormalWARE is a project to investigate how formal methods can be used in the development of critical, software-intensive, projects.

The Canadian Automated Air Traffic System (CAATS) development project at Raytheon Systems (formerly Hughes Aircraft) inspired some of the directions which FormalWARE pursued. The formal methods, based on rigorous mathematical models and specification techniques, were applied to the analysis, verification and testing of software.

The review panel is very positive about the results achieved by the FormalWARE project. It recognizes how difficult it can be to persuade a large company like Raytheon or MacDonald Dettwiler (MDA) to even think about a more formal approach to software development. In this light, the project has achieved significant results by fostering continued interaction between industry and two universities over a two year period.

The project has placed graduate students inside the companies for lengthy periods; there has been a series of presentations and/or regular visits by university researchers to the companies; the project has developed software tools applicable to real problems faced by the companies; the project has used problems taken from the industrial setting as the basis for academic research.

All these interactions represent money well spent by the BC Advanced Systems Institute and the industry partners. We view the project as a success, particularly when the relatively modest funding level is taken into account.

Although the funded part of the project has now come to a conclusion, we expect the momentum generated by the project to lead to more collaborations in the future. We strongly urge Raytheon and MDA to continue to work with these researchers and other researchers in a mutually beneficial partnership, whether or not additional funds from government agencies can be obtained.

We fully expect some of the methods and software tools to be commercially useful. We recommend that opportunities for exploiting the technology be pursued both through the University-Industry Liaison Office and by more direct means.

1. INTRODUCTION

1.1 THE PROJECT HISTORY

The FormalWARE project came into being only through the continued and persistent efforts of Dr. Jeff Joyce. Dr. Joyce was formerly a faculty member in the Department of Computer Science at UBC, but left the University of British Columbia (UBC) to join what was then Hughes Aircraft in Richmond, BC.

Hughes Aircraft of Canada, which through a corporate takeover is now Raytheon Systems Canada Ltd., holds the Canadian contract to develop an air traffic control system. The Canadian Automated Air Traffic System (CAATS) represents a large, highly complex, combination of hardware and software. There are stringent requirements on the system for its reliability and its correctness. The software component of the system would appear to provide an ideal proving ground for software engineering techniques in support of developing correct software.

Dr. Joyce's goal has been to develop and try out *formal methods* that would be applicable to the production of critical software-intensive systems. To this end, he proposed a collaborative project between university researchers and two BC companies that develop critical software, the second BC company being MacDonald Dettwiler (or MDA for short). He applied to the BC Advanced Systems Institute (ASI) for funding for the project. After some delays due to problems concerned with intellectual property rights and attempts to find additional sources of finance, the project started on 1 April 1996 with a total budget of $150,000 per year. The budget was financed by contributions from Raytheon, MDA and ASI. The financing was for two years, which implies that the project has now officially come to an end, though some of the collaborations will continue for months or years to come.

1.2 PROJECT MEMBERS

The project has involved researchers at two universities, UBC and the University of Victoria (UVic). On the day of the review, the project team comprised the following principal members:

Dr. Jeff Joyce (Raytheon and Adjunct Professor at UBC)

Dr. Daniel Hoffman (Associate Professor, UVic)

Dr. Paul Gilmore (Professor Emeritus, UBC)

Dr. Bruce Kapron (Associate Professor, UVic)

Dr. Lee White (Professor, Case Western, on leave at UVic),

the following graduate students:

Kendra Cooper (PhD candidate, UBC)

Nancy Day (PhD candidate, UBC)

Michael Donat (PhD candidate, UBC)

Xuhui Chen (MSc candidate, UVic)

Jayakrishnan Nair (MSc candidate, UVic)

Ken Wong (MSc candidate, UBC)

and the following industry members:

Dr. Philip Gray (MDA)

Mr. Richard Yates (MDA).

Additional current and former personnel include:

Christine Jensen (assistant project director)

Jamie Andrews (affiliate researcher)

Gunter Mussbacher (affiliate researcher)

Dilian Gurov (PhD candidate, UVic)

Georgi Kostadinov (MSc candidate, UBC)

David Baar (corporate affiliate)

Timothy Duty (corporate affiliate).

There were several individuals employed by the sponsoring companies who made significant contributions to the project as mentors and/or research collaborators. These included:

Gerry Pelletier (Raytheon)

Jim Ronback (Raytheon)

Steve Scroggins (Raytheon).

1.3 PROJECT DIRECTIONS

The project has focused on tool support to aid in the specification, verification and testing of critical software. The significant achievements include the following:

Automated Analysis of Tabular Specifications

A tabular-style notation for relating inputs and outputs for software was developed, and a software tool to check for completeness and consistency of the tabular specification implemented. [Refs: PhD research of N. Day]

Software Requirements Specification

A refinement of a *threads-style* specification method used within Raytheon is currently under development. The software requirements are to be specified using a parseable and precise English-style notation. The requirements can then be automatically checked for consistency and validity, and be used to generate test cases for the software. [Refs: PhD research of K. Cooper]

Generation of Test Procedures from Requirements

The software requirements, as written in an English-like notation, are automatically processed to generate "*test frames*", which encapsulate stimulus-response relationships. The test frames can subsequently be used by test engineers to produce test procedures. [Refs: PhD research of M. Donat]

Generation of Test Drivers

A test script notation has been developed. The Ada Package Exerciser (APE) translates the scripts into an Ada program which tests an Ada package for conformance to its specification. [Refs: research of D. Hoffman]

Derivation of Safety Verification Conditions

Hazard definitions, usually expressed as temporal relationships, are analyzed and translated into safety verification conditions. The conditions may be converted into run-time assertions or may be tested by applying verification techniques to the system's source code. [Refs: MSc research of K. Wong]

Specification and Analysis of State Transition Systems

The desired behaviour of future air traffic control systems is partially defined by documents known as "Aeronautical Telecommunications Network Standards and Recommended Practices" (SARPs). The SARPs use a combination of English and state transition tables. The SARPs have been translated into formal notation and analyzed by tools that perform type checking and model checking. [Refs: research of B. Kapron & D. Gurov; PhD research of N. Day]

S *Notation and the Fuss Tool*

An ASCII-based specification language, named S, was developed as a more usable alternative to Z notation. Z uses a rich set of mathematical symbols and requires special tools just for editing and printing specifications. S notation is supported by a tool named "*Fuss*", developed at UBC. Fuss has been used as a front-end to various tools used in the FormalWARE project. [Refs: research of J. Joyce, N. Day , M. Donat, J. Andrews]

Methodology and Foundations

> Project members have investigated methodologies for testing. [Refs: research of D. Hoffman, L. White, J. Nair, J. Andrews], have investigated the logical foundations of specification notations as used in practical industrial applications [Refs: research of P. Gilmore], and have investigated the logic foundations of model checking [Refs: research of B. Kapron, D. Gurov].

1.4 THE REVIEW PROCESS

On 21 May 1998, the review panel spent the day meeting the researchers and two representatives from industry. These industry representatives were Peter Holt from Raytheon and Philip Gray from MDA.

The researchers presented brief 10 to 20 minute overviews of their research projects and responded to questions, both after their presentations and in separate discussion sessions. There was active participation by the industry representatives and by Brent Sauder, the Executive Director of ASI, who attended during the afternoon.

The day concluded with a wrap-up session where the panel gave initial feedback on their impressions of the FormalWARE project.

2. Objectives and Scope of the Project

This section describes the project objectives and scope as understood by the reviewers and as they pertain to BC industry and the global market.

The general goal of the research project was to investigate the applicability of *formal methods* in the development of critical, software-intensive systems. This goal was realized through specific objectives focused on making improvements to systems/software engineering processes for requirements specification and validation, requirements-based testing, software component testing and software safety validation. Although formal methods have been proposed and studied for many years in academia, their application in industrial settings are not well understood. Therefore, from the review team's perspective, the research goal and objectives were interesting and innovative. The strong *buy in* from industry partners was particularly laudable.

Although the scope of the project was primarily focused on formal methods, it was apparent during the review and from the review materials that a broad research mandate had been selected, ranging from theoretical issues in logic formalisms to applied techniques in component testing. The research participants were happy with this broad scope. The industrial participants did not object to the broad scope; however, they felt it was more difficult to collaborate on aspects that were peripheral to the main thrust of the research program.

The significance of the project objectives to BC industry and ASI (Advanced Systems Institute) are difficult to assess, primarily because the major paybacks for the proposed research is not short term. Industry is generally more interested in quick, rather than long-term, returns on their investment. The transfer of formal method expertise, tools and techniques to the participating companies was hoped for and the degree to which this was achieved is described later in the report. ASI is happy with their investment if the industry partners are generally satisfied with the results and highly qualified personnel are moving to the workforce. Certainly both of these were possible and were realized in this project.

On a global scale, the potential exists for the development of a recognized pocket of expertise in formal systems in the Vancouver area. This would make Vancouver an even more attractive location for companies developing *mission critical* software. One of the important, potential outcomes of this research is the *spinning off* of a new company that specializes in formal requirements specification and/or software component testing.

3. Scientific Merit and Overview of Results and Achievements

The technical work accomplished under the FormalWARE project addresses diverse topics. It links existing but more theoretical research in formal methods with existing software engineering practice in industry.

The research topics addressed by FormalWARE are best discovered by viewing the extensive and very informative project webpages (http://www.cs.ubc.ca/formalWARE/). Indeed, it would be impossible for a document of this nature to address all the topics as completely as they have been on the webpages. These webpages serve as an excellent educational resource for industry and university researchers interested in applications of formal methods to industrial problems.

In this technical review of the project we address the following topics.

- Formal Specification of Requirements
- Executing Formal Specifications
- Requirements Specification for Industrial Applications
- Automatic Generation of Test Frames from Requirements Specifications
- Automated Analysis of Tabular Specifications
- Automated Generation of Test Drivers for Ada Packages
- Derivation of Safety Verification Conditions from Hazard Definitions

Formal Specification of Requirements

The higher order specification language **S** developed by Joyce, Day, and Donat at UBC serves as the basis for much of the work in the FormalWARE project. In keeping with their goal of making formal methods palatable to their industrial partners, **S** provides a formal specification language that is somewhat more readable than traditional mathematical notations such as **Z**, **HOL** or **PVS**. Additionally, there is also a form of tabular input that can be translated into **S**. The Fuss type-checker facilitates type checking of **S** specifications. **S** appears to be an effective and readable formalism for the specification of computer software.

Certainly, writing down requirements in a formal language is a step that rarely occurs in current industrial practice. Under the umbrella of the FormalWARE project a number of formal specifications were constructed. **S** has been used to formally specify and analyze a number of standards documents of interest to engineers working on current and future air traffic control systems. The first is the specification of Air Traffic Separation Minima for the North Atlantic Region; another is a draft specification of the Aeronautical Telecommunications Network Standards and Recommended Practices developed by the International Civil Aviation Organization (ICAO); a third is a portion of the ICAO instructions for filling out flight plans. In all cases, the process of formalization uncovered previously unknown ambiguities in the specifications. Clearly, these are significant activities and directly address the goals of the FormalWARE project.

Executing Requirements

Given a formal specification, how do we know it captures the intended requirements? This is the validation problem. The validation of a formal specification at the requirements level is a difficult and well known problem. Requirements validation has been addressed in the FormalWARE project by a number of methods.

One approach was to translate higher order specifications into executable form. This allows for a testing based approach to requirements validation that industrial partners are both familiar and comfortable with based on their extensive experience with artifacts developed later in the software life-cycle. Andrew's work on translating **S** specifications into higher order logic programs in the language lambda-prolog is a scientifically sound approach. The choice of lambda-prolog as the executable language is a natural choice.

Requirements Specification Techniques for Industrial Applications

Formal notations have proven to be a daunting challenge to technology transfer of formal methods; although there have been some successes in transferring formal specification techniques into industrial settings, more often than not, they serve as impediments to technology transfer. The design of the **Q** specification language, which is semantically based on **S**, is an attempt make formal specification more accessible. This is accomplished by allowing natural language phrases to stand for un-interpreted constants in what appears to the user as a structured natural language specification.

Automatic Generation of Test Frames from Requirements Specifications

The application of formal methods to testing both requirements and components was emphasized in the project. Component testing of Ada packages is addressed below. Generating tests from formal requirements specifications is an extremely rich area of research that has seen surprisingly little attention from the mainstream formal methods research community. Donat's work on automatically generating test frames from **S** specifications (which may include quantifiers) appears to be unique. Aside from the practical aspects of automatic test-frame generation, the test frames provide a very interesting alternative to formal proofs for requirements validation. Engineers are able to examine the automatically generated test-frames to see if their specifications have the intended meanings.

Automated Analysis of Tabular Specifications

This work was based on, and motivated by, the analysis of an air traffic control separation standard for the North Atlantic region. The work involved developing a tabular specification notation with links to the **S** formal language. The use of tabular specifications is popular since tables convey information intuitively to engineers. A significant drawback is that structure and hierarchy inherent to the problem domain is often lost because of the *flat*ness of tabular specifications. FormalWARE rectifies the first of these deficiencies to some extent by preserving the problem structure in the tables -

each row is associated with a specific requirements specification notion. The work done is also interesting since it uses the notion of *lightweight* formal reasoning support which is much more likely to be used in practice. It will be interesting to see how well the work generalizes to other (non air traffic control) domains.

Automated Generation of Test Drivers for Ada Packages

The approach is aimed at the module as opposed at the system level. Its novelty lies in the use of sophisticated test scripts that allow direct embedding of Ada code in the test script. This is impressive technology and the results compare very favorably with those obtained from drivers written by experienced testers. These results indicate that the tool produced will be of considerable commercial interest. Although this work did not make overt use of formal methods, it is clearly related with the work on deriving test cases from requirements. It is hoped that future studies might pursue and engineer this very valuable link that would bring automation of the testing process much nearer to reality.

Derivation of Safety Verification Conditions from Hazard Definitions

This work strives to bridge the semantic gap between the safety requirements of a system and the code that ensures these requirements are met. This is a fruitful research area. Work so far has addressed the derivation of software verification conditions from hazard definitions, the aim being to discharge the resulting proof obligations in the code. This is interesting work that is well worth pursuing further. Areas that might be considered are the (partial) automation of the hazard refinement as well as building better links with the traditional safety analysis and management process. It would also be very worthwhile to adapt the technique so that it can be useful during design time as well as during V&V time – thus allowing the safety requirements to influence the design of the system as it is built.

4. POTENTIAL FOR INDUSTRIAL APPLICATION OF RESULTS

The whole area of formal methods as a methodology applied to software development and testing is not well understood by industry. In discussion with some of the industry participants, e.g. Philip Gray of MacDonald Dettwiler (MDA), one of their main reasons for participating in this project was to indeed learn more about the subject matter itself. Both industry participants in the project as well as some of the presenters (e.g. Daniel Hoffman, UVic) commented that industry, in general, does not have adequate tools for testing software code with respect to compliance with system specifications. Hence, the need exists within industry and the project results provided the industry participants with some insights and practical tools that could be applied in practice.

However, Philip Gray also commented that MDA should have contributed more in-kind effort, i.e. person-days to transfer project results and knowledge about formal methods from academe to industry. Although some tools were developed, these require adoption and regular use by industry in order to take these beyond the prototype stage to the industrial strength stage where they can be used by MDA and other firms.

James Caldwell of Cornell University, one of the reviewers, is involved in a NASA program which has been funding similar collaborative Formal Methods projects since 1988. Based on his experience at NASA, he is confident that the results of this project have wide applicability.

An unanswered question raised by Daniel Hoffman concerns component software testing versus system software testing. The trade-offs between testing at the component versus system levels are well understood by hardware designers but less so by software system developers. Dr. Hoffman should be encouraged to pursue this further and apparently he has approached the Software Productivity Centre in Vancouver in this regard.

For the project results to be adopted by industry – both the participants as well as other firms – on-going collaboration must be cultivated. It would be a pity if, just because the project itself has reached a logical conclusion, collaboration would end. For successful technology transfer, an on-going working relationship is essential.

5. Lessons Learned from Industry/University Collaboration

Effective collaboration requires a substantial commitment of human resources from both the industry and university participants in the project. Some of the students, e.g. Mike Donat and Ken Wong stated that they could have benefited from more industry mentorship.

A barrier to collaboration is the issue of intellectual property ownership. This factor was cited by many of the participants as well as by members of the review panel. At the early stages of the FormalWARE project, there were many delays, i.e. almost two years, just in getting the legal collaboration agreements finalized. One student, Kendra Cooper, felt that her PhD work progressed slower as a result of these agreements, e.g. she experienced some problems because of the publication ban clause in favour of industry. This resulted in her having to deal with additional, non-academic, obstacles in advancing her research.

It is important that those working on the project communicate the benefits of their work to industry. Peter Holt, of Raytheon, commented that this was an important skill for students to develop and that, with respect to this project, sometimes such communication was lacking. Perhaps more frequent hands-on in-situ demonstrations and tutorials would help facilitate this.

The greatly different agendas and motivations of researchers versus industry management remains a challenge. Publication and research imperatives often do not mesh well with short-term production pressures. What is important to a researcher may be insignificant to a corporate engineer. One person commented that one possible solution to this was for faculty to be more sensitive to industry needs as it may be easier for them to be flexible, especially on minor issues, because they act more autonomously than companies who act more as corporate bodies.

In addition to Jeff Joyce, who has Adjunct Professor status, the UBC faculty members significant involvements in the FormalWare Project were limited to Professors Ito and Gilmore.[1] Although the project participants strived to nurture collaborations with additional UBC faculty, this did not happen. There appear to be conflicts between the goals of industry and the goals of academic researchers, such as those caused by commercial deadlines, which are hard to reconcile.

[1] It should be noted that Drs. M. Greenstreet, A. Hu, N. Hutchinson, M. Ito and G. Murphy served on the PhD supervisory committees for some of the students involved in the FormalWARE project.

6. CONCLUSIONS AND RECOMMENDATIONS

The quality of the research, the diversity of topics, and the volume of research performed under the auspices of the FormalWARE project are extremely impressive. The project has successfully addressed important problems in the intersection of formal methods and software engineering. It has accomplished this in the context of existing industrial practice at Raytheon, and to a lesser extent at MacDonald Dettwiler. The pressures imposed by hard deadlines for industrial deliverables and cost control make the quality of the research accomplished under the project even more impressive. Another accomplishment that must not be forgotten is the training of highly qualified personnel – the students involved in the project have been exposed to methods used in industry and have interacted with industry on close to an equal basis. The training and experience that they have received will benefit their future employers immensely.

Our recommendations for future university/industry collaborations of a similar nature to this project are as follows.

- A project champion and mentor must be identified by the industry side as well as the academic participants. Some of the students lamented that, at times, they felt like orphans, i.e. the project work was too academic for industry and at the same time too practical to be of academic interest. Students are actually an ideal vehicle for bridging such a gap because typically there will be students on the project team who aspire to become academics as well as students who seek industry careers. However, they need encouragement from both camps.
- A significant amount of work was accomplished in the two year project, including a number of excellent publications. However, the reviewers believe the individual project researchers need to follow up by disseminating the results of the project more widely through additional conference and journal publications.
- Projects should not be delayed due to legal issues. Perhaps the University-Industry Liaison offices could lend their assistance in this regard?
- The results of the project could and should be transferred to industry. There are several companies that could be considered as possible future collaborators, and which should be made aware of the FormalWARE project. These include Architel Systems Corporation[1] in Toronto and Triant Technologies[2] in Vancouver, a fault tolerant systems developer. Another Vancouver systems consulting firm, Sierra Systems Consultants Inc.,[3] prides itself on its understanding of state of the art systems development techniques and methodologies and may be interested in using FormalWARE tools in its development activities.
- The project should not simply close down because the ASI funding has come to an end. The industry partners should recognize the value of this partnership and continue to col-

[1] http://www.architel.com; there is a local contact, Mr. Roy Trivett, a founder of the company, with e-mail address: roy@trivett.com.

[2] http://www.triant.com

[3] http://www.sierrasys.com

laborate with university faculty and students.

Dr. Jeff Joyce should be commended for initiating and leading a highly successful project. The scientific objectives of the FormalWARE project, as stated in the proposal submitted to the Advanced Systems Institute in November 1994, have been addressed by original and significant research. Commercial potential does exist for the methods and the software tools that have been developed as part of the FormalWARE project.

ACKNOWLEDGEMENTS

Thanks are due to Ms. Gayle Palas for expertly combining the disjoint textual fragments, in a variety of formats, provided by the review panel members. She corrected the spelling and the syntax, and is responsible for the look and feel of the document. Dr. Jeff Joyce supplied some minor factual corrections to the penultimate version of this document.