Title: Results of a Formal Methods Demonstration I'reject

Authors: John C. Kelly, 1%.11., Richard G. Covington, Ph.D., and David Hamilton

Abstract

This paper describes the results of a cooperative study [D-1 1432] conducted by a team of researchers in formal methods (FM) at three NASA centers (The Jet Propulsion Laboratory (JPL), Johnson Space Center (JSC), and Langley Research Center (LaRC)) to demonstrate FM techniques and to tailor them to critical NASA software systems. FM is a set of techniques and tools based on formal logic and mathematics for the purpose of specifying and verifying computer systems and software. This pilot project applied FM to an existing critical software subsystem , the Shuttle's Jet Select subsystem (Phase 1 of an ongoing study). The Prototype Verification System (PVS) specification language and tool developed at SRI international was used for this study. This study shows that FM can be used successfully to uncover hidden issues in a highly critical and mature Functional Subsystem Software Requirements (FSSR) specification which arc very difficult to discover by traditional means.

Introduction

Recent studies of software subsystems in critical applications arc yielding data which expose a software requirements quality problem facing current and future projects. It appears that the early stages of the software life cycle are especially prone to errors which can have a lasting influence on the reliability, cost, and safety of a system. A sample of the conclusions from these studies points to requirements and design specifications as a high priority candidate for better software engineering techniques:

> Most hazardous software safety errors found during system integration and test of two NASA spacecraft were the result of requirements discrepancies or interface specifications [Lutz931
> The highest density of major defects found through the usc of software inspections was during the requirements phase. This was 7 times higher than the density of major defects found in code inspections [Kelly92]
> Requirements errors are bet ween 10 and 100 times more costly to fix at later phases of the software lifecycle than at the requirements phase itself [Basili84], [Boehm84], [Kelly92].

These problems indicate the need to advance the state-of-the-practice in the area of software requirements engineering. One reason that requirements engineering deserves special attention is its general lack of tools and effective procedures relative to later life cycle phases (e.g., detailed design and code), which are already supported by well-defined methods, languages, and automated tools. in contrast, current requirements engineering practice suffers from too much dependence on ad hoc methods and ambiguous natural language specifications, and from little or no automated support. Another motivation for moving beyond status quo techniques for requirements engineering is illustrated by the so-called "quality ceiling" which advanced organizations arc encountering in some of their high-end software development products. This occurs when the currently employed development and assurance techniques undergo so much optimization and fine-tuning that no additional major quality improvements can be expected. The demands of developing software systems in a high-criticality, high-reliability

application domain obliges these organizations to investigate new software engineering techniques which can potentially break through current quality ceilings.

Research into emerging Formal Methods (FM) techniques and tools has produced improved approaches to these problem areas. FM is a collection of disciplines and techniques based on discrete mathematics and formal logic, applied to the problems of specification and verification of systems. FM stands in contrast to traditional requirements techniques, which arc characterized by specifications written in English prose ("shahs") accompanied by various types of diagrams. FM offers significant potential for improvement over traditional techniques by augmenting thcm with specifications written in a rigorous formal language, which may then be subjected to automated formal deductive reasoning in order to prove important claims about the specification. The disciplines from mathematics and logic that give Formal Methods its foundation include (1) the predicate calculus (first-order logic) and higher-order logics, (2) rccu rsive function theory, (3) the lambda calculus, (4) programming language semant its, and (5) discrete mathematics (number theory, set theory, abstract algebra, etc.)

This paper illustrates some of the benefits of a FM approach by reporting on an application of FM techniques to a software function of the Space Shuttle On-Orbit Digital Auto Pilot called Jet Select. Shuttle software is an example of an ultra-high-quality system that is the product of a very extensive traditional development, quality assurance, and testing program. The function of Jet Select is to choose which thrusters (or jets) to fire, based on calculated desired changes to the Shuttle's direction of motion and attitude, from a total of 44 thrusters distributed over the Shuttle's surface. In making this choice, Jet Select must ignore certain classes of jets for safety or mission configuration considerations. The classes of jets to be ignored depend on jet availability, moding constraints and type of direction change. The application of FM to Jet Select allowed a ncw way of looking at the Jet Select system, and suggested ncw lines of reasoning in critiquing the requirements and the realization of those requirements

In the next section wc detail the steps of the technical approach used in applying FM techniques to Jet Select. In Section III, wc present some of the most important results from the study, Finally, in Section IV, wc discuss the significance of those results, especially with regard to the potential future role of FM in the development life cycle for high-criticality Systems.

11. Technical Approach

The goal of this study was to apply FM to an existing system, demonstrating how the approach could lead to a new understanding of the system and provide new lines of reasoning to guide the critique of intermediate development products (e.g., requirements, design, etc.). This goal was later augmented to demonstrate more explicitly the importance of levels of abstraction (i.e., different hierarchically arranged views of the system from requirements through design to implementation), and to demonstrate how FM helps to organize and link these levels. The FM tool used for the study was the Prototype Verification System (PVS), developed at SRI international [Shankar93]. PVS is a forma] specification language environment which allows specifications to be written in a predicate calculus notation. The PVS environment also supports automated parsing, type checking and theorem proving .

The FM team started by examining the existing requirements for Jet Select as documented in the FSSR (Functional Subsystem Software Requirements). The FSSR notation is a mixture of prose, tables, and "wiring diagrams" (digital logic functional boxes and data lines that represent the software requirements). The requirements were divided into three independent parts --

Primary Jets, Failure Reconfiguration, and Vernier/ALT Jets --, and common interfaces, data types, and formats were established. The requirements were then translated directly into PVS. The second step would normally be to state important properties of the system that should be satisfied by the formal definition of the system, Instead, since the existing FSSR was at an implementation level of detail, the team developed higher level "as-built" documentation in order to show how formal methods can be used at earlier lifecycle phases. This reconstruction of higher level structures was restricted to the Vernier/ALT Jets subsection of Jet Select. PVS specifications for three different views of Vernier/ALT Jets were then ultimately produced, corresponding to requirements, high-level design, and low-level design. The levels were established through a mix of activities, including emulation of FSSR descriptions, private conversation.s with Shuttle requirements analysts (RAs), and an examination of the HAL/S source code for Jet Select. At the requirements level, a list of 14 properties was stated and proved. At all levels, lists of issues were maintained and communicated to the software development team for evaluation.

111, Results

ISSUES AT THE DETAILED-DESIGN LEVEL (FSSR LEVEL SPECIFICATION)

At this level of abstraction, informal models (state and execution flow diagrams) were developed of the "as-built" Vernier/ALT Jets component of Jet Select, based on the code. These Iow-level informal models enabled the team to compare the as-built Vernier/ALT Jets software to the working description in the FSSR. An Ada emulator based on the FSSR and the PVS specification was also developed for part of the subsystem to investigate the dynamic aspect of the specification. The PVS formal specification at this level was based upon the FSSR specification, except in the case of a discrepancy between the FSSR and the informal code model. In these cases, the issue was documented against the FSSR for further investigation by a cognizant Shuttle requirements analyst (RA). Some of these issues were resolved as misunderstandings by the FM team and removed from the list. At the end of this portion of the demonstration study the list contained the 46 items summarized below:

- 1 Incorrect Logic in the FSSR document (but not in the code)
- 1 Circular Reasoning in the FSSR document (but not in the code)
- 1 Redundant Test (also appearing in the code)
- 5 Confusing Logic Notation in the FSSR document
- 3 Type Mismatches in the in the FSSR document
- 3 Typos (i.e., misspellings) in the FSSR document
- 3 Confusing Notation in the FSSR document
- 29 Clarifications needed for the FSSR document to be self explanatory

As a result of this part of the demonstration study, a Documentation Change Request was written and submitted to update the FSSR to more accurately match the current code implementation, The demonstration also contributed to establishing the credibility of the FM team member's approach to technical] y anal yzing the Jet Select function, 1 t should be noted that although significant changes to the FSSR document may result from this study, the Shuttle Jet Select software subsystem itself was not found to contain defects which would adversely affect its functionality. Given the thoroughness of the testing which is performed on each operational increment of the Shuttle's Flight software, this was not unexpected. However, the onc redundant internal test which the Jet Select software performs is indicative of the way in which a formal analysis approach can complement testing.

One of the desires of the team was to demonstrate the use of automatically steered theorem provers. Since the properties which the RA's were interested in having proved or disproved were at a significantly higher level abstraction, it was decided that it would be fruitful to reconstruct higher levels of informal and formal specifications to achieve this goal. This would enable the example to guide projects which are currently under development, and do not have the Shuttle's detailed level of FSSR specifications.

ISSUES AT THE HIGH-LEVEL DESIGN LEVEL

The next level of abstraction was the specification of the high-level design level. At this level, the FSSR wiring diagrams were coalesced into a half-dozen essential functional boxes, and individual data interfaces were aggregated into more structured data that was passed between functional boxes. The basic intent and operation of the system could be more readily inferred visually from the high-level design, This step required the FM team, with the help of the Jet Select domain experts, to probe the intent of the system designers in order to reveal the true underlying requirements. State diagrams were created to generalize the concrete design details and reveal underlying structure. At this level the intent of the branching decisions in selecting the appropriate jets were clarified and more key data t ypes were identified.

The following is a list of issues uncovered by the high-level specification exercise.

- 1 Incompletely documented variable in the FSSR document,
- 1 Inconstant use of notation in the FSSR document
- 1 Apparent no-op in the FSSR document and the code

ISSUES AT THE REQUIREMENTS LEVEL

Motivated by the products of the FM exercise at the high-level design level and by a list of essential Jet Select requirements compiled by Shuttle requirements analysts, key features of Vernier/ALT Jets were captured an additional layer of PVS specification at the requirements level. Lemmas stating the required behavior of the system were proven. Several issues that were raised during the requirements analysis in this step have been docu mented and brought to the attention of the Shuttle's domain specialist in this area.

Evaluation of the issues raised during the creation of the requirements level specification is somewhat problematic since there does not exist a working Shuttle document to use for comparison purposes. The FSSR maps more directly to the detailed design level, The issues from the requirements level are generally involve more global questions, which have been resolved in ways that were undocumented (or recorded in places unknown to the FM team) during the original development. The requirements issues listed below have not led to any known functional errors in the operational code. However, these high-level issues have motivated renewed efforts to understand some of the original decisions facing the designers of Jet Select subsystem.

- Possible attempts to use failed non-downfiring vernier jets.
- Non-positive dot product between desired direction vector and jet d irection vector
- Cannot decide to fire a second jet until decision to fire first jet is completed
- Downmoding to free drift is not done by Jet Select
- The terms "failed" and "unavailable" arc both used in the requirements, but only the term "unavailable" is defined

- Pulsing of primary jets inconsistent in the case of failing to find a jet to fire

1 V. Discussion

This task has demonstrated the thoroughness of a FM analysis by finding issues in a mature Jet Select FSSR document and implemented code, uncovering hidden requirements and assumptions, and developing a requirements level formal specification of Jet Select that yields a deeper understanding of the overall subsystem functionality. The three components of formal specifications, computer-aided proofs, and emulators all contributed to the analysis of specifications, from software requirements through detailed design. Although these techniques require additional cost, the reasonable increase in expenditure is warranted for highly critical systems where human lives and NASA spacecraft arc at risk.

Formal Methods dots not replace empirical testing. Especially in the domain of high-criticality systems, testing is always an essential step to demonstrate that the implemented system complies with its requirements. However, Formal Methods can assist the assurance process by uncovering problems earlier in the software development lifecycle than testing and adds considerabl y more rigor to existing assu rance techniques for early lifecycle engineering products.

Conclusions from this study include:

- Issues Uncovered. Many issues and questions were raised during this study. The issues list is itself compelling evidence that FM provides' a rigorous and very productive approach to validating specifications. In total, over 50 questions and issues were raised and categorized. Some of these issues were significant enough to be bundled together to form a ncw Change Request for future updates to the Jet Select FSSR document,

- Cost. This study did not focus on the detailed collection of cost metrics associated with FM. However, it dots appear that the costs arc within reason and arc roughly comparable to the current cost of requirements analysis. When considering highly critical subsystems the cost arc not prohibitively expensive.

- Potentially increased capability for future analysis of Jet Select. This case study has documented the Jet Select requirements in a formal notation that supports automated analysis for validation. Additionally, the augmented levels of specification provided by this project yield a basis for analyzing incremental changes to the existing system. Formalizing the requirements level specification has allowed the FM team to validate a set of system level properties not addressed elsewhere, and to document previously implicit information and underlying assumptions in an unambiguous language.

V. Acknowledgements

Contributors to this case study in addition to the authors include, from the Jet I'repulsion Laboratory: Mori Khorrami, Robyn Lutz, Ph. D.; from Johnson Space Center: Ernie Fridge, Doc

Shankar, Ph.D. (IBM), David Hamilton (IBM), Chris Hickey (IBM), Scott French (IBM); from Langley Research Center: Sally Johnson, Ben DiVito, Ph.D. (VIGYAN), John Rushby, Ph.D. (SRI), Judith Crow, Ph.D. (SRI), Sam Owre (SRI); and from NASA HQ Code Q: Alice Robinson. The NASA/ASEE Summer Faculty Fellowship Program funded Brent Auernheimer, Ph.D. (California State University) and Betty Cheng, Ph.D. (Michigan State University) to perform a cooperative study on the Shuttle's Phase Plane and contribute to the overall team.

The research described in this paper was carried out in part by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. Funding was provided by a POP from NASA Code Q for a multicenter investigation into Formal Methods at the Jet Propulsion Laboratory, Johnson Space Center, and Langley Research Center.

Reference herein to any specific commercial product, process, or service by trade, name, trademark, manufacturer, or otherwise, does not constitute *or* imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology.

VII. References

[Basili84] Basili, V.R. and Perricone, B.T. "Software Errors and Complexity: An Empirical Investigation", *Communications of the ACM, 21(1): 42-52,* January, 1984. -

Boehm84] Boehm, B.W. "Software Engineering Economics", *IEEE Transactions on Software Engineering,* SE-10(1): *4-21,* January, 1984.

D-11432] *"Formal Methods Demonstration Project for Space Applications - Phase I Case Study: Space Shuttle Orbit DAP Jet Select",* at JPL : Kelly, J. C., Covington, R., Khorrami, M., & Robyn Lutz; at JSC: Ernie Fridge, Doc Shankar, Ph.D. (IBM), David Hamilton (IBM), Chris Hickey (IBM), Scott French (IBM); at LaRC: Sally Johnson, Ben DiVito, Ph.D. (VIGYAN), John Rushby, Ph.D. (S1{1), Judith Crow, Ph.D. (S1<1), Sam Owre (SRI); at NASA HQ Code Q: Alice Robinson, Brent Auernheimer, Ph.D. (California State University) and Betty Cheng, Ph.D. (Michigan State University). NASA- JPL Report #D-11432, Jet Propulsion Laboratory, Pasadena, CA., Dec. 22,1993

[Kelly92] Kelly, J.C., Sherif, J, S., and Hops, J. "An Analysis of Defect Densities Found During Software Inspections", *Journal of Systems and Software,* VOI 17, 111-117, January, 1992,

[Lutz 93] Lutz, R.R. "Analyzing Soft ware Requirements Errors in Safety-Critical Embedded Systems", *Proceedings of the IEEE International Symposium on Requirements Engineering,* San Diego, CA, January, 1993.

[Shankar93] Shankar, N., Owre, S., and Rushby, J. M., "The PVS Specification Language (Beta Release)", SRI International, Menlo Park, CA, March 31, 1993.