Experiences with the Application of Formal Methods to a High Criticality Space Application

Sally C. Johnson, NASA Langley Research Center
John C. Kelly, Jet Propulsion Laboratory
Ernie Fridge, NASA Johnson Space Center
Alice Robinson, NASA Headquarters

September 17, 1993

This report summarizes the results of a three- center (JPL, JSC, LaRC) Formal Methods Demonstration Project for Space Applications for the purpose of improving the quality of critical software subsystems. We have completed the first year of a two-year study to tailor formal methods to space applications, determine the benefits of formal methods, and lay the foundation for transitioning these techniques to select NASA projects.

1 Rationale for Formal Methods Study

Recent studies have shown the software quality problem is greatest during the early lifecycle phases of requirements and design. Formal methods is a set of widely researched techniques for assuring the quality of critical systems, software, and hardware, Formal methods techniques arc based on formal logical and mathematical reasoning and are appropriate for use in the verification of early lifecycle software engineering products.

The rationale for using formal methods on NASA applications includes the following points:

1. As space software evolves into more and more complex applications of increasing criticality, continual improvement of verification and validation techniques is vital to NASA'S high-integrity systems.

2. On select projects, NASA may have reached a quality ceiling at the natural limits of traditional proccxw-oriented assurance techniques, Formal methods offers the potential for significantly reducing risk beyond the norms attainable through traditional techniques.

3. Critical application such as nuclear reactor shutdown, automatic train protection, air traffic collision avoidance, and secure networks are already successfully using of formal methods techniques.

Formal methods can be adapted to a wide variety of software systems by scaling the amount of three factors: 1) degree of rigor, 2) coverage of software lifecycle phases, and 3) percent of the system which is analyzed by formal methods techniques. The degree of rigor is determined by what techniques are used: formal specifications, emulators, manual proofs, or mechanically verified proofs. Formal specifications are requirements expressed in special languages (PVS, EHDM, HOL, etc.) that have precisely defined syntax and semantics and provide the capability for mathematical examination of the specification. Emulators (or specification animators) are the literal translation of the formal specification into an executable high-level model of the system's externally observable behavior. A proof is a complete and convincing mathematical argument that demonstrate the truth of an assertion about the set of formal specifications. Mechanically verified proof is conceptually similar to manual proof but uses specification language tools to rigorously verify that the argument is indeed a valid proof. Formal specifications can also be used as a valid basis for "formal inspections" of software. Although formal inspection is not a rigorous proof of software correctness, it is a highly disciplined process for finding, correcting, and checking the correction of software defects, and plays an important supporting role in any process involving formal methods.

2 Technical Approach

The project began with a Feasibility Study to assess the maturity of formal methods and their applicability to NASA space software. The Feasibility study concluded with the recommendation of a 2-year Pilot Study to demonstrate and evaluate the application of formal methods to NASA software. We are currently at the end of the first year of the Pilot Study. Several candidate software systems were examined during the Feasibility study, including:

- Space Station Freedom Data Storage and Retrieval System
- Space Station Freedom Attitude Determination System
- Iii,gh-Level Model of Space Station Freedom Fiber Ring Architecture
- Cassini Spacecraft Attitude and Articulation Control Subsystem
- Space Shuttle Jet Select System

The Space Shuttle Jet Select application was chosen as an ideal first demonstration application for the Pilot Study, The Jet Select application contains the logic to select the

reaction-control jets to fire to perform a specific Space Shuttle maneuver. On the one hand, the Jet Select system is mature, so it represents an opportunity for application of formal methods in a controlled, stable environment. On the other hand, the application is currently being modified to respond to the changing requirements that arise out of Space Station Freedom docking constraints (Operational increment 24), Thus, the modifications to this application are currently going through a Requirements Analysis phase, which is the optimal phase for application of the formal methods techniques to be demonstrated.

The first step of the Pilot Study was for everyone involved in the project to attend tutorials on the Jet Select application and to learn how to use the Prototype Verification System (PVS) under development at SRI International, which was chosen to support the formal methods demonstration. We then proceeded to develop PVS specifications of Jet Select from the FSSR (Functional Subsystem Software Requirement) document by dividing the work into three independent parts, agreeing on interfaces and common data types, then integrating the resulting pieces. We identified several simple requirements of Jet Select that were called out in the FSSR and proved that our PVS specifications met those properties. Emulations of the Jet Select software were developed to aid in our understanding of what the software was supposed to be doing.

Realizing that the FSSR diagrams that are used in the Space Shuttle software development process are at too low a level of detail to be useful in performing effective Requirements Analysis, we then explored the development of a set of hierarchical specifications for a subset of Jet Select to provide descriptions of the system at higher levels of abstraction, Development of specification at the higher levels of abstraction was facilitated by the emulations plus discussions with Jet Select domain experts at IBM.

We have essentially completed our original goal of developing a formal specification of Jet Select and proving several properties about it. We arc currently in the process of completing two higher-level specifications of the Vernier/Alt mode portions of Jet Select in PVS and proving that each level is a correct implementation of the abstract level above. We have worked with domain experts at IBM to develop a paritla set of require.ments for the Vernier/Alt Modes and have proven that our highest level specification meets those specific criteria,

During Fiscal Year 1994, we are planning to participate in the Requirements Analysis process for the development of the Operational Increment 24 release of Jet Select by building on the work we completed in the previous year. Because PVS can save and rerun previous proofs, we can efficiently modify our specifications to meet the new requirements and complete our proof chain in much less time than the original specifications and proofs required.

3 Preliminary Observations and Findings

This project has piqued the interest of the Jet Select domain experts from the beginning, Print of our preference for selecting the Jet Select application for the demonstration was because of the apparent interest of the application project managers. Several software development research engineers at IBM have participated in this Pilot Study project under contract to JSC, developing and proving aspects of the formal specification and providing excellent, timely interaction with the IBM employees currently preforming Requirements Analysis for the Space Shuttle Jet Select modification. The feedback wc have obtained from the Jet Select Requirements Analysts has been positive. They recognize and appreciate the preciseness and structure that formal methods can bring to the process as well as the completeness and confidence.

We have uncovered a number of interesting issues while formally specifying the Jet Select application; however, they have all been relatively minor. In total, 46 questions and issues were raised during the forma] specification of the FSSR and were categorized in the following way:

- 1 Incorrect Logic (to be corrected by an approved Change Request)
- 1 Circular Reasoning (resolved correctly in the software)
- 1 Redundant Test
- 5 Confusing Notations of Logic

0 3 Type Mismatches

- 3 Typos (i.e., misspellings)

4 3 (Very) Confusing Notations

- 29 Clarifications (ambiguities in FSSR)

Several additional issues have been uncovered during the development and proofs of the higher levels of abstractions. The issues found at this level are of much greater importance than the problems found at the low level, because the correctness of the basic algorithms used are involved, For example, we uncovered a case where a failed jet might still be used by the algorithm during a maneuver. This issue is still being examined by the domain experts,

While we have successfully demonstrated that formal methods can add to the clarity and rigorousness of the Requirements Analysis process, much work remains to be done to create and advocate an effective method for incorporating formal methods into NASA's software development process for space applications.