SIG on Formal Methods

NEWS LETTER VOLUME 5 , MAY 2015

SIG on Formal Methods:

Formal Methods (FM) has been in existence since 1940's, when Alan Turing proved the logical analysis of sequential programs using the properties of program states; and Floyd, Hoare and Naur used axiomatic techniques to prove program correctness against the specifications in 1960's.

These initial successes helped inculcate an interest in applying FM to the field of computer science.

Academia has been instrumental in bringing this field to the forefront, through continued research and development. The use of Formal Methods requires an expert skill-set expertise and therefore, its use is limited to those trained in the field.

Mission Statement:

Computer Society of India wants the field of Formal Methods to have a wider audience and more people to benefit from the application of these methods to all spheres of life. There is a need to use effective, correct and reliable approaches to design, develop and qualify complex, high assurance system software with the rigid schedules and budget. For this we need advanced tools, techniques and methods. Industry standards like RTCA DO-178C (Civil Aerospace), ISO 26262 (Automotive), IEC 61513(Nuclear), EN50126 (Railways) have recommended the usage of formal –method based approach to be used in the various phases of engineering process to achieve the required levels of safety and security.

Today there are proven techniques and tools that can be used in specification, design and verification & validation phases to assure correct requirement-capture, implementation, software functionality and security. This helps in developing high assurance software for applications such as cyber-physical systems, net-centric warfare systems, autonomous robots and Next Generation Air Transportation.

- **One day workshop on FM was conducted during April 2013**

Objectives of the Special Interest Group (SIG) are:

- To bring together scientists, academicians active in the field of formal methods and willing to exchange their experience in the industrial usage of formal methods
- To coordinate efforts in the transfer of formal methods technology and knowledge to industry
- To promote research and development for the improvement of formal methods and tools with respect to their usage in industry.
- To bring out practical engineering methods where formal methods will be integrated with current engineering methods

Some of the known applications of formal methods are:

- Formal verification, including theorem proving, model checking, and static analysis
- Techniques and algorithms for scaling formal methods
- Use of formal methods in automated software engineering and testing
- Model-based formal development
- Formal program synthesis
- Formal approaches to fault tolerance
- Use of formal methods in safety cases
- Use of formal methods in human-machine interaction analysis
- Use of formal methods in compiler validation and object code verification

3. Committee Members

1. Ms. Bhanumathi K S, Convener
2. Mr.Chander Mannar
3. Prof.Anirban basu
4. Mr. Suman Kumar
5. Prof. Shyam Sundar
6. Ms. Manju Nanda
7. Ms. J. Jayanthi
8. Ms. Saroja Devi
9. Prof. Meenakshi D'Souza
10. Dr. Yoganand Jeepu
11. Dr. Swatnalatha Rao
12. Dr. Aditya Kanade
13. Dr. A. Indira
14. Mr. Dhinakaran Pillai

Convener:

Bhanumathi K S
“Ganadhakshya” #406, 8 C Main,
H R B R First Block
Kalyan Nagar
Bangalore 560043
Email:bhanushekar@gmail.com
Mobile:+91 95350 92589

Formal Methods for Software Safety Assessment

Compiled by Dr. Manju Nanda, CSIR-NAL

Safety is not the sole responsibility of the System Safety engineer. Creating a safe system is a team effort and safety is everyone's responsibility. Software is a vital part of most systems. It controls hardware and provides mission-critical data. Software must be safe.

There are many types of safety analyses that can be completed during software development. Every phase of the lifecycle can be affected by increased analysis as a result of safety considerations. The analyses can range from Requirements Criticality Analysis to Software Fault Tree Analysis of the design to Formal Methods

In the production of safety-critical systems or systems that require high assurance, FM provides a methodology that gives the highest degree of assurance for a trustworthy software system. FM has been used with success on NASA, military, and commercial systems that were considered safety-critical applications. The benefits from the application of the methodology accrue to both safety and non-safety areas. FM does not guarantee a precise quantifiable level of reliability. At present, FM is only acknowledged as producing systems that provide a high level of assurance.

FM is used in several ways:

a. As a way to assure the software after-the-fact

b. As a way to assure the software in parallel.

c. As a way to develop the software

"After the fact" software verification can increase the confidence in a safety-critical system. When the regular software development is completed, then the formal specification and verification begin. The Software Assurance, Safety, or IV&V engineer converts the "human readable" requirements into a formal specification and proves properties about the specification. The code that implements the system may also be formally verified to be consistent with the formal specification. With this approach, two separate development activities occur, increasing cost and schedule. In addition, problems found at this late stage are costly to fix.

"In parallel" software verification still uses two separate teams (software development and FM verification), but they operate in parallel during the whole process. The development team uses the regular practices of good software development. At the same time the FM team writes formal specifications for the system and verifies them. While still costly, this method of assuring the software allows for quicker development. Software errors are found earlier in the development cycle when they are less expensive to correct. However, communication between the two teams is vital for this approach to work.

Formal model-based safety analysis is primarily an extension of existing safety analysis. Using formal methods for model-based safety analysis can support the safety assurance complex systems. The basis is the construction of a common formal system model which is shared between the developer and the safety engineer. Such a model generally consists of an abstract model of the system, a model of the physical behavior of the environment and a model of the possible faults and failure modes. A model expressed in a language with formal semantics allows for the analysis using automatic verification tools and can therefore support the safety analysis process of complex systems. Compared to more traditional approaches the advantages are firstly that using a common system model requires less effort in case of design changes and secondly in the increased automation which make more efficient safety analysis possible. Formal analysis can expose interesting failure scenarios.

Fault Tolerance Verification Using Model-Checkers **(**SMV**)**
Fault Tree Generation using PVS

Some of the tools and techniques specifically proposed for automating safety analysis.

FSAP/NuSMV-SA: FSAP/NuSMV-SA is a tool for automating the generation of fault trees.

Galileo – Dynamic Fault Tree Analysis Tool

HiP-HOPS: HiP-HOPS (Hierarchically Performed Hazard Origin and Propagation Studies) is a method for safety analysis originating from a number of classical techniques such as Functional Failure Analysis (FFA), Failure Mode and Effects Analysis and Fault Tree Analysis.

Altarica – Language with support for Fault Modeling: The Altarica language was designed to formally specify the behavior of systems when faults occur. An Altarica model can be assessed of means of complementary tools such as a fault tree generator and a model-checker.

References:

Model-Based Safety Analysis Final Report, Anjali Joshi, Mike Whalen, Mats P.E. Heimdahl, Available at: http://shemesh.larc.nasa.gov/fm/papers/Model-BasedSafetyAnalysis.pdf

Formal methods for Safety Assessment of Critical Software at RATP, Department -- RATP/ING/STF/QS/AQL Evguenia DMITRIEVA / Oct 16 2014, Available at: http://projects.laas.fr/IFSE/FMF/J4/slides/P07_Evguenia_Dmitrieva.pdf

NASA Software Safety Guidebook, NASA-GB-8719.13, March 2004, Available at: http://www.hq.nasa.gov/office/codeq/doctree/871913.pdf

Formal Methods and the Certification of Critical Systems, John Rushby, Available at: http://www.csl.sri.com/users/rushby/papers/csl-93-7.pdf

****