SIG on Formal Methods

NEWS LETTER VOLUME 4 , OCTOBER 2014

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. Yoganand Jeepu
11. Dr. Swatnalatha Rao

Planned Activities:

National Work Shop and Conference on Formal Methods for Software Engineering
Venue: M R C Auditorium, I I Sc , Bangalore, October 15-17, 2014

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

Amazon Web Services Uses Formal Methods

Compiled by Bhanumathi K S

Since 2011, engineers at Amazon Web Services (AWS) have been using formal specification and model checking to help solve difficult design problems in critical systems.

"To a first approximation, we can say that accidents are almost always the result of incorrect estimates of the likelihood of one or more things." - C. Michael Holloway, NASA

Benefits: Improved Understanding, Productivity and Innovation

We have found that writing a formal specification pays several dividends over the lifetime of the system. All production services at Amazon are under constant development, even those released years ago; we add new features that customers have requested, we re-design components to handle massive increases in scale, and we improve performance by removing bottlenecks. Many of these changes are complex, and they must be made to the running system with no downtime. Our first priority is always to avoid causing bugs in a production system, so we often need to answer the question, "is this change safe?" We have found that a major benefit of having a precise, testable model of the core system is that we can rapidly verify that even deep changes are safe, or learn that they are unsafe without doing any harm. In several cases we have prevented subtle, serious bugs from reaching production.

In other cases we have been able to make innovative performance optimizations – e.g. removing or narrowing locks, or weakening constraints on message ordering – which we would not have dared to do without having model-checked those changes. A precise, testable description of a system becomes a "what if …" tool for designs, analogous to how spread-sheets are a "what if …" tool for financial models. We have found that using such a tool to explore the behavior of the system can give the designer an improved understanding of the system.

****