METHODS

Keynote

Certification is ultimately about judgment that a system is adequately safe. We need a "science of certification".

- Favor explicit over implicit approaches
- Be wary of qualitative evidence

A proposed research agenda

- Science for certification
- Specification and verification of integration frameworks
- High performance automated verification for strong properties of model based designs
- Compositional certification
- Tool qualification
- Integrated methods and arguments

Presentations

- Lui R. Sha
    - Verification is as good as the assumptions you base it on, and assumptions change as technologies change. Testing is vital.
    - Architecture and Implementation Divergence.
    - Two most important issues are interaction complexity reduction and the co-stability of software and hardware.
- Allen Goldberg
    - Making reliable software systems from less reliable software subsystems
    - Idea is Software Fault Protection – model how the software behaves and what response should be made to fault sin software.
    - Software redundancy maybe an option but it is expensive.
    - Effective redundancy can be done at runtime and maybe better than doing it at verification time.
    - Our work extends ARINC 653.
    - Fault containment is essential to fault isolation.
- Peter Skaves
    -
- Eric Feron and Arnaud Venet
    - Static stability analysis for autocoded software for aviation systems.
    - Matlab should not be looked at as a programming language but instead should be looked at as a specification environment.
    - The autocode that is generated from Simulink/Matlab does not free you from verifying the code it has generated.
    - Certified autocode cannot be done from a mathematical perspective.
- Emilio Frazzoli
- Matthew B. Dwyer

    - All validation, verification, synthesis has limits, and industry, govt, and research need to understand these limits.
    - When techniques fail there is a lot of information that can help with the follow on analysis.
    - Use a mixture of techniques to validate, but what to do when we are at the point of deployment. How to detect and respond to errors at runtime.
- Natasha Neogi
    - Safety and security properties can lead to competing requirements.
    - What to do – build in safety and security from system inception.
    - Questions – how do you quantify and qualify safety and security, what are the effects of other qualities?
- Robin Bloomfield
- Reinhard Wilhelm
    - Concerned with real time guarantees for hard real time systems
    - A380 subsystems are being certified using aiT.
    - Over-provisioning will no longer work. Completely deterministic system will not perform.
    - A new research agenda – design for predictability, and design only what you can analyze.

General Discussion

- Eric's Slides
- Sha
    - We need to understand and document the limitations
    - How can we handle residual bugs in the system
- Storm (Lockhead)
    - Not one technique will address it all
    - Biased towards formal methods in this workshop however there are other elements
- Shohan (NASA AMES)
    - People underestimate error handling, underflow and overflow
    - What does it mean if my tool cannot search through the space
    - Need to know how to codify the science and put trust into the methods.
- Taft
    - What does the FAA do to certify the hardware components?
    - Lingberg (FAA) – FAA is not in the business of telling applicants how to do something, only what needs to be done.
    - Feron – should we also include the legal system
    - Lingberg – it is involved, that is where you get the formal standards
    - Feron – for example law professors or lawyers working with research teams
    - Lingberg – lawyers are always involved, within the standards as well as guidance.
    - (audience) – maybe they are helpful for clarification of law but not more than that.
    - Hansmen – bringing in extra people may not help with the understanding.
    - Stosch (Idaho) – lawyers do have experience in interpreting evidence, and that is where they can help.

- Susner Aircraft
    - Need to state our assumptions and how they impact on V&V
- Wheeler (North Carolina)
    - Timing predicitability is important, and need to look at average and worse case performances.
    - How you achieve it is interesting – we need to be more proactive and find new ways instead of just turning off architecture to see what the impact is.
    - Need special support in hardware
    - Feron – are you arguing that we should have more self-aware processes?
    - Need a rethinking of process design to look at aspects of worst case and average case.
- Sha
    - Can you enter onto your slides the point of software stability despite residual errors
    - Feron – what do you mean residual errors?
    - It means you cannot completely debug the system
    - Feron – what kind of errors?
    - Need to look at types of errors, as tackling specific errors can lead to more bugs.
- Winbuilt
    - Need to understand what it means by diversity? This is a difficult research theme and we need to address it.
    - One technique is to map out who is responsible for particular areas.
- Hansmen
    - One issue is to archive and document the assumptions – both implicit and explicit.
- Storm (Lockhead)
    - There are always issues with integrating someone else's stuff, so we cannot bring together affordable code. So specifying interfaces will not help all the time.
    - Pearce (FAA) – granted we need more rigorous interface, however maybe thinking of it as a contract (or at least formalizing) may help
- Feron
    - Control has given us a number of tools to look at composibility and aspects of interface design
- Hushby (SRI)
    - Strong notions of interfaces will help in composition and construction.
- Cormen (Boeing)
    - The real challenge is to build in safety and security into a complex air control system and keep the air control system working.
    - Feron – how about having a PGP like process, where every airplane publishes a public key.
    - Gabor – there are many aspects – implementing security affects performance. Bad thing about security is that you can't port onto an existing system easily.
    - Neogi – one of the biggest problems with security is the interlinking of mixed aircraft. Essentially you have to have teirs and different levels of security.
- Gabor
    - What needs to be put into an interface to identify it? – for example what processor was it compiled on?
    - Ziloskwy – one way is to wrap it into the component, or you can separate the processor out.

- Feron – is there a way to specify software as a block and hardware as a block in such a way that we can predict the hardware/software block.
- Sha – we need to formalise the components and the FAA should take that up.
- Feron – is there a way of doing this so that the whole community agrees
- Sha – no, which is why the FAA needs to enforce it.
- Taft - DO297 standard maybe a place to start to go to the next level of interface design.
- Ziloskwy – there are architectural approaches to defining this.
- Lingberg – FAA is in the process of updating the DO297.