GENI real time workshop, Reston VA, 6,7 Feb 2006
Assurance, Security, Certification for GENI

John Rushby

Computer Science Laboratory
SRI International
Menlo Park CA USA
Certification

- Terminology differs across fields, but generally...

- **Certification** is a societal or institutional **judgment** that some system is safe or secure or... enough for some specific application in some specific context
  - Have to show you thought of everything
  - The challenge of "unbounded relevance"

- **Assurance** is the **technical** analysis in support of certification
  - Makes clear what you did think of
  - And how you dealt with it

- Another **good research topic**:
  - Move the boundary between these
  - In favor of more technical analysis
  - GENI could contribute to this
For Example

- **InterPeak** (Swedish company) are building a secure TCP/IP stack for EAL6+ evaluation
- First step is to identify the threat model
- Then construct the Protection Profile (PP)
  - And get agreement on that
- Then develop the stack following the processes of the PP
  - And provide the technical assurance specified in the PP
- Certifiers decide if they believe any of this
  - And if it’s good enough for their application
  - And environment
  - Maybe with restrictions (e.g., TS and S only)
State of the Art in Assurance

• Traditionally, lots of process stuff, lots of testing

• Increasingly it means formal methods

• Due to
  ○ More complex, higher risk systems (e.g., IMA)
  ○ Recent big advances in automated formal methods
  ○ And better integ’n with trad’l development practices
    ★ Move to model-based design (MBD)
    ★ FM extended to design exploration, debugging, testing

• Cost and practicality depend on type of system considered, nature of assumed environment, properties of interest, level of description (model vs. code), and scale of system
For Example: Safety Critical System Frameworks

- **System** is designed to be *synchronous* (deterministic)
  - Built on an integration framework such as TTA
  - **Guarantees** certain properties of systems built on it
    - Solves the hard problems once and for all
    - **Composability** (preservation of prior properties)
    - And **compositionality** (reason from parts to whole)
  - Without cooperation of components outside framework
- **Environment** may inject faults
- **Properties** are technical safety properties (mostly *invariants*)
  - Eventuality properties are bounded
  - May involve *real time*
- **Description** of the framework is at the level of algorithms and models (could go down to implementation)
- **Scale** is *modest* (tens of KLSOC)
SOA in Formal Methods

- Massive advances in power of automated reasoning methods
  - Use of **SAT solvers**, emergence of **SMT solvers**
  - **Abstract interpretation**

- Powerful methods for using these (**automated abstractions**)
  - Predicate abstraction, Craig interpolation, CEGAR
  - Infinite bounded model checking, k-induction

- Highly **customized automation** for special purposes
  - Static analysis, ESC, software model checkers, PCC

- And **integration methods** for putting things back together
  - Evidential tool bus
**Satisfiability Modulo Theories (SMT)**

- Individual decision procedures decide **conjunctions** of formulas in their decided theories

- **Combinations** of decision procedures (using, e.g., Nelson-Oppen or Shostak methods) decide conjunctions over the **combined theories** (e.g., arithmetic plus arrays)

- **SMT allows general propositional structure**
  - e.g., \((x \leq y \lor y = 5) \land (x < 0 \lor y \leq x) \land x \neq y\)
  - \ldots possibly continued for 1000s of terms

- Should exploit search strategies of modern SAT solvers

- So replace the **terms** by **propositional variables**
  - \((A \lor B) \land (C \lor D) \land E\)

- Get a **solution from a SAT solver** (if none, we are done)
  - e.g., \(A, D, E\)
Lemmas On Demand

- Restore the interpretation of variables and send the conjunction to the core decision procedure
  - e.g., \( x \leq y \land y \leq x \land x \neq y \)
- If satisfiable, we are done
- If not, ask SAT solver for a new assignment—but isn’t it expensive to keep doing this?
- Yes, so first, do a little bit of work to find fragments that explain the unsatisfiability, and send these back to the SAT solver as additional constraints (i.e., lemmas)
  - \( A \land D \supset \neg E \)
- Iterate to termination (e.g., \( B, D, E: y = 5, y < x; y = 5, x = 6 \))
- This is called “lemmas on demand” or “DPLL(T)"
- It works really well: yields effective SMT solvers

John Rushby, SRI

Assurance, Security, Certification: 8
SMT Solvers

- SMT solvers are being honed by competition
- Various divisions (depending on the theories considered)
  - Equality and uninterpreted functions
  - Difference logic \((x - y < c)\)
  - Full linear arithmetic
  - \ldots for integers as well as reals
  - Arrays
- Next competition at FLoC (Seattle, Summer 2006)
- SMT solvers enable infinite bounded model checking, and powerful backends to interactive theorem provers
Example: Real Time

- Traditionally hard for automated analysis because continuous time excludes finite state methods.
- Timed automata methods handle continuous time.
  - But defeated by the case explosion when (discrete) faults are considered.
- SMT solvers can handle both dimensions.
  - Timeout automata, k-induction, disjunctive invariants.
- E.g., Biphase Mark Protocol for asynchronous communication.
  - Clocks at either end have different skew, rates, jitter.
  - So have to encode a clock in the data stream.
  - Used in CDs, Ethernet.
  - Verify parameter values for reliable transmission.
Real Time: Biphase Mark (ctd)

- First verified by human-guided proof in \texttt{ACL2} by J Moore
- \textbf{Three different verifications} used \texttt{PVS}
  - One by Groote and Vaandrager used \texttt{PVS + UPPAAL}
  - Required 37 invariants, 4,000 proof steps, \textit{hours} of prover time to check
- Brown and Pike recently did it with \texttt{sal-inf-bmc}
  - \textbf{Three} lemmas proved \textit{automatically} with 1-induction,
  - Statement of theorem discovered \textit{systematically} using \textit{disjunctive invariants} (7 disjuncts)
  - Theorem proved \textit{automatically} using 5-induction
  - Verification takes \textit{seconds} to check
- \textbf{Adapted} verification to 8-N-1 protocol (used in UARTs)
  - Revealed a \textit{bug} in published application note

John Rushby, SRI
Assurance, Security, Certification: 11
Analysis of Security Properties/Secure Systems

- **Topmost properties are slippery**
  - Noninterference is not a property
  - Does *not* compose or refine nicely

Usual to impose safety properties that are stronger than noninterference

- **New trend** (revival of an old one): **MILS**
  - Development and automated verification of commercial separation kernels is well under way
  - These are integration framework for security, just like TTA for safety in IMA

- **But the real challenge is a development and verification process for systems built on these**
  - Should exploit deconstruction opportunities of MILS
Analysis of Security Properties/Secure Systems (ctd)

- Security protocols
  - Authentication etc. are pretty well solved
  - Challenges are in subtle properties: anonymity, etc.

- Possible opportunity for GENI
  - Not just secure communications
  - But an integration framework for distributed secure systems
Analysis of Networking/Networked Systems

- Mostly focus on variants of the asynchronous model
  - Failure detectors
  - Partial and timed asynchrony of various kinds
- Harder to reason about than synchronous systems
  - And harder actually to achieve properties of interest
    Because one must deal with tricky eventuality arguments
- Modest progress; most verifications require human guidance
- Possible opportunity for GENI
  - An internet with synchronous guarantees
  - Cf. Verissimo’s timely computer base

Would allow simpler assurance arguments for properties of complex distributed systems
Other Areas

- **Protocols**
  - Model checkers inside J-Sim

- **Code level analysis**
  - Recent rapid advances by focusing on limited properties
  - Highly customized verifiers
  - Microsoft: SDV
  - Airbus: Caveat (INRIA), Astree (Cousot), AbsInt (Wilhelm)

- **Hybrid Systems**
  - This is the formal methods technology for analysis and synthesis of control systems
  - Big recent advances based on abstraction
  - And automated theorem proving
  - Successful application to biology
Summary

- Assurance, certification need a **compositional systems** view
- A focus for GENI could be as an integration framework
  - For **safely synchronous, secure, real time** systems
  - Deliver minimal compositional properties to clients that ease their assurance and certification tasks
  - In Helen’s terms: migrate edge concerns into the core
  - In Lui’s terms: reinterpret some QoS in terms of composable properties
  - Could help save us from consequences of accidental systems
- Formal analysis technology will be ready when you are
- Probably