# Verification Challenges

Jim Woodcock

University of York

Newton Institute | Cambridge

24 September 2019

## Overview

- Tony Hoare's verification challenges
  - Construct a verifying compiler
  - Unify theories in computer science
- This talk: focus on industrial-scale pilot projects
- Outcomes and impact
  - Engineering solutions to challenges
  - Scientific advances in theories, tools, & techniques
  - Wider impact and sea-change since 2003
- Future pilot project in robotics

## History of UK Grand Challenges

- 2002: Programme Committee (chair: Tony Hoare)
  - Initial workshop in Edinburgh
- 109 submissions from the UK computing research community
- Seven themes emerged with one or two champions each
- Public email discussions, moderated, openly archived
- Principles
  1. No community submission rejected by the Committee
  2. No discussion about potential funding
  3. GC research not prioritised over theory or practice
- Unite research directions for long-term aspirations

- 2004: Conference on GCs for Computing Education
- 250 attendees
- 50 submissions all linked to an existing research challenge

## UKCRC Grand Challenges

| | | |
|---|---|---|
| GC1 | In Vivo, In Silico: virtual worm, weed, bug | Sleep |
| GC2 | Science for global ubiquitous computing | Kwiatkowska/Sassone |
| GC3 | Memories for life | Fitzgibbon/Reiter |
| GC4 | Scalable ubiquitous computing systems | Crowcroft |
| GC5 | The architecture of brain and mind | Sloman |
| **GC6** | **Dependable systems evolution** | Hoare/Woodcock |
| GC7 | Journeys in non-classical computation | Stepney |

## GC6: Dependable Systems Evolution

**CNN News** June 4th, 1996:

```
The Ariane 5 rocket was destroyed seconds after it
took off, a spokesman for Arianespace said today.
```

- ESA: 10 years, \$7bn, 6 tonne payload — what went wrong?
- Extensively tested software in Ariane 4
- Triggered simple arithmetic error in Ariane 5

- Millions of software faults hit users every day
- Each software fault offers an opening to a virus
- Code Red virus costs estimated at \$4bn world wide
- 2002: US department of Commerce faults costs \$60bn/year

- Dependability justified reliance on system behaviour
- Evidence and justification must be scientifically rigorous
- Very expensive and difficult to produce such evidence
- Exhaustive testing is impracticable
- More sophisticated approach to correctness: mathematics
- Sizewell B safety case: 100 person-years (c.£10M/£2.03B)

## Verification Grand Challenge (International GC6)

### Objectives

- Scientific foundation for justifiably dependable systems
  - Even in the face of the most extreme threats
- In the future...
- Inaccessible systems work for decades
- Very large-scale systems have controllable costs and risks
- Costs of rapid evolution reflect size of change
  - Not the scale of the system
- Scientific and technical advances trigger a radical change
  - In the practice of developing computer systems
- Sell software for safety, security, reliability...
- ...as well as for its functionality

Software will have warranties

## Verification Grand Challenge

- Three strands:
  1. Theory
  2. Tools
  3. Experiments

- Experimental Strand Pilot Projects

| | | |
|---|---|---|
| 1. Verified file store | Nasa | (US) |
| 2. FreeRTOS | Wittenstein HIS | (UK) |
| 3. Radio spectrum auctions | Smith Institute | (UK) |
| 4. Cardiac pacemaker | Boston Scientific | (US) |
| 5. Tokeneer ID station | Altran Praxis | (UK/US) |
| 6. **Mondex** | NatWest | (UK) |
| 7. **Hypervisor** | Microsoft | (US) |

## Mondex

- NatWest consortium Electronic purse hosted on a smart card
- 1996: High-assurance standard ITSEC Level E6
- Strong guarantees needed that transactions are secure
- Business case: electronic cash can't be counterfeited
- 400 pages of specification, design, and handwritten proofs
- Proof revealed bug in implementation of secondary protocol
- Convincing counterexample provided insight to correct it
- 3rd-party evaluators found an undischarged assumption

- First commercial product to achieve E6
- Sanitised Mondex documentation publicly available

"[In 1996,] mechanising such a large proof cost-effectively is beyond the state of the art."

Mondex challenge: investigate automation

## Mondex

### The Mondex players

- Alloy (MIT)
- *Circus* (York)
- CSP/FDR2 (Oxford/York)
- Event-B (Southampton)
- Isabelle/UTP (York)
- JavaCard (Augsburg)
- KIV/ASM (Augsburg)
- Perfect Developer (Escher)
- $\pi$-Calculus (Newcastle)
- Petri Nets (Florida)
- PVS/SAL (Macao/DTU)
- Raise (Macao/DTU)
- SAM (Florida)
- StaRVOOrS (Chalmers/Augsburg)
- UML/OCL (Bremen)
- UML/USE (York)
- VDM (Newcastle)
- Z & Z/Eves (York)

Summer Schools UK ($\times$3), Germany ($\times$2), SRI, China ($\times$3), Brazil ($\times$2), South Africa, . . .

PhD & MSc theses

## Hypervisor

Verification target: Microsoft Hyper-V kernel

- 100kloc concurrent C, 5kloc x64 assembly code
- Runs on bare metal: no dependencies on libraries
- Runs on x64 processor with virtualisation features
- Relies on formal specification of x64 processor
- Concurrent C code
  - Course-grained (lock) + fine-grained (lock-free) concurrency
- Production code optimised for performance, not verification
- Top-level correctness theorem
  - Virtualisation simulates real processor + memory
- Verification challenge
  - Multi-level address translation
  - Lock-free concurrent translation lookaside buffers
  - High-speed caches to translate virtual to physical addresses

## Hypervisor

Verification tool: VCC

- Functional verification of C
- First-order predicate logic specification
- Function modular & thread-modular
  - No code-inlining/unrolling
- Annotations (residing in code)
  - Data structure invariants
  - Function contracts (heap-frame, pre- & post-conditions)
  - Correctness assertions
  - Ghost data structures + ghost code
- Verification condition generator
- Prover backend: SMT solver Z3
- Fully automatic, no proof language, no interactive proofs
- Verification guidance through code annotations only

- VCC and Z3 are (now) open-source on `github`

## Z3 SMT Solver

Bjørner & De Moura: 2019 Herbrand Award at CADE-27

"In recognition of their numerous and important contributions to SMT solving, including its theory, implementation, and application to a wide range of academic and industrial needs."

- Z3: 5,000 citations since 2008
- General keywords
  - Symbolic execution, program verification, model checking, . . . , industry 4.0, quantum, flash memory, distributed ledgers
- Specific keywords

| | | | | | |
|---|---|---|---|---|---|
| SMT | 593 | abstraction | 121 | implementation | 85 |
| software | 384 | Java | 105 | testing | 84 |
| solver | 222 | architecture | 96 | debugging | 78 |
| scalability | 204 | decidability | 89 | scheduling | 68 |
| ATP | 163 | boolean sat | 87 | probabilism | 62 |

## Z3 SMT Solver

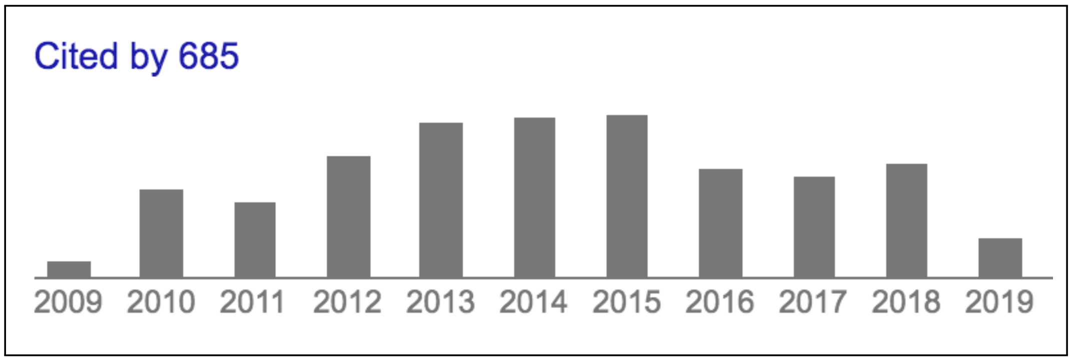
Ignited entire research disciplines and businesses

Examples

- Microsoft Security Risk Detection
  - `fuzz` testing service for finding security critical bugs in software
  - "Security Risk Detection is Microsoft's unique `fuzz` testing service for finding security critical bugs in software. Security Risk Detection helps customers quickly adopt practices and technology battle-tested over the last 15 years at Microsoft."
- Azure reliability
- Verified cryptographic libraries and protocols
- Verified compiler optimisations
- Product line configurations
- Real-time scheduling
  - E.g., retransmission-free time-sensitive network architectures

## Fun with Figures: Integers for Impact!

- 3 = Microsoft Verified Software Milestone Awards (3×£5k)
  - Tokeneer, CompCert, Intel Core i7
- 8 = fellowships
  - $\{\, FRS = 2 \wedge FREng = 1 \,\}\ VSI\ \{\, FRS = 4 \wedge FREng = 7 \,\}$
- 11 = VSTTE working conferences = 190k paper downloads
  - *VSTTE = Verified Software: Theories, Tools, & Techniques*
- 1,183 = *ACM Computing Surveys* VSI special issue citations
  - "Formal methods: Practice & experience"



- £2,341,113,000 = value of 2013 UK 4G spectrum auction

## “Doing Formal”

### VSI zeitgeist

- 2002: Formal methods in industry
  - inmos, IBM, Praxis, GEC Alsthom, MATRA Transport, RATP, NatWest, Rockwell Collins, Airbus, . . .
- 2019: ARM, AdaCore, Airbus, Alacris, Altran, Amazon Web Services, Apple, BAE Systems, Bedrock Systems, Boeing, Bosch, British Energy, CERN, Centaur Technology, Cog Systems, Data61, Elastic Global, eSpark Learning, Ethereum, Facebook, FinProof, FireEye, Galois, Google, Grammatech, Green Hills Software, IBM, ISP RAS, InfoTecs, Intel, JetBrains, Kaspersky Lab, Kernkonzept, Kind Software, MUCT, Machine Zone, Microsoft, MongoDB, NASA, Oracle, Particular Software, PingCAP, Rockwell Collins, SiFive, Statebox, Sukhoi, Synopsis, T-Platforms, TrustInSoft, Trustworthy Systems, Zilliqa

## Some Lessons Learnt

1. Funding for academic research
   - Active champions very important: it has to be their day job!
2. Industrial participation essential
   - They have to know they want our help!
3. It takes time: 15 years so far
   - 2020 Newton Institute Workshop planning next 15 years
4. Primary and secondary impact
   - Hoare's leadership inspired others to innovate and apply
   - History of ideas:
     - "An axiomatic basis for computer programming": 7.5k cites
     - Program logic $\rightarrow$ VSI $\rightarrow$ industrial exploitation
   - See "Continuous Reasoning: Scaling the impact of formal methods" by O'Hearn
   - Facebook, Amazon, Microsoft, Google, Altran

## What’s Next?

What’s the major problem facing verification today?

- Cyber-physical systems, including robotics
- Inherently heterogenous system descriptions
  - Discrete/continuous, hardware, environment, control
- Interdisciplinary engineering tools & techniques
- Pairing virtual and physical worlds
  - Digital twins with run time monitoring of operational big data
  - Correctness wrto runtime assumptions
- What research advances are needed?
  1. Improvements in modelling physical elements
  2. Physical & cyber model integration
  3. Modelling & verification techniques
  4. Validation & simulation techniques

## Improvements in Modelling Physical Elements

### Foundations needed

- Theory: contracts, refinement, verification
- Modelling: interfaces + contracts + modularity
- Model transformation: debug, abstract, refactor, clone
- Static analysis: data and control dependencies
- Discretisation: real-life data and sampling frequency
- Model validation: tuning, precision, outliers, timebands

## Physical and Cyber Model Integration

- Semantics for hybrid models
  - Develop software engineering workflow
  - Verification of hybrid models
  - Derive verification conditions using semantics
  - Discharge using model checkers & tactical theorem provers
  - Patterns, abstractions, and architectures for CPS & DTs
- Efficient verification algorithms
  - Reactive programs
  - Continuous components
  - Hybrid models
- Unification
  - Apply to hybrid combinations: eg, Event-B + Modelica
  - Unification of different approaches

## Modelling & Verification Techniques

- Quantitative theories
  - Robustness, resilience, reliability, safety, and security
  - Design and problem space exploration: analysis + simulation
  - Scalability of probabilistic & statistical model checking
  - Proof theory for PRISM, Storm, etc
  - Efficient proof automation
  - Probabilistic model checking modulo theories
- Emergent behaviours
  - Efficiently predicting emergence
  - Engineering emergence
  - Emergence & refinement
  - Theories for run-time monitoring of emergence

## Validation & Simulation Techniques

- Contributions needed
  - Semantics of practical simulation tools
  - Deriving sound simulations from formal models
  - Practical techniques for model coupling
  - Further contributions to FMI
- Directions
  - Testing theories
  - Principled, regressive, oracular, automated, empirical
  - Proof theory for co-simulation
  - Simulation validation
  - General properties of master algorithms
  - Uncertainty in the environment

## Butler Lampson (2002): No more roads deaths

### Grand challenge: Verify autonomous vehicles

We should choose to do this, not because it is easy, but because it is hard; because that goal will serve to organise and measure the best of our energies and skills, because that challenge is one that we are willing to accept, one we are unwilling to postpone, and one we intend to win.