McMaster University

Inspiring Innovation and Discovery

Research on Certification @ McMaster

Tom Maibaum

McMaster Centre for Software Certification

McSCert

UofYork – 17Jan2013

McMaster Centre for Software Certification

- My colleagues and I identified software certification as our research focus in 2004.
- Established Software Certification Consortium; 12 meetings since 2007 involving North American academics, companies, regulators.
- Obtained grant to work on the topic in 2009.
- My interest has been in the foundations of certification; my conclusion is that assurance cases are a confidence trick ☺: they have no semantics enabling a scientific evaluation to be made of any assurance case!

Software Certification Consortium

- Established as a forum for discussions related to software intensive systems in the context of a need for regulation in the software/systems domain
- US, Canada focus
- Regularly attended by academics, industry (some!), and several regulators/government agencies (USFDA, USNRC, CNSC, NIST, NASA)
- Safety/assurance cases have become an important focus, but scope is much wider

Agenda

MONDAY	
8:45 – 9:00	**Introduction**
9:00 – 9:30	**Rance Cleaveland,** University of Maryland / Fraunhofer USA
9:30 - 10:00	**Ramesh S.,** GM Global R&D
10:00 – 10:30	**Discussion**
10:45 – 12:30	**Breakout Session 1 & Report Back**
12:30 – 1:30	**Lunch**
1:30 – 2:00	**John Knight,** University of Virginia
2:00 – 2:30	**Daniel Jackson,** MIT
2:45 – 3:15	**Elizabeth Fong,** NIST
3:15 – 3:45	**Discussion**
3:45 – 5:30	**Breakout Session 2 & Report Back**
7:00	**Dinner**

Agenda

TUESDAY	
9:00-9:30	**Oleg Sokolsky & Insup Lee,** University of Pennsylvania
9:30-10:00	**Norbert Carte,** US Nuclear Regulatory Commission
10:00 - 10:30	**Discussion**
10:45 - 12:30	**Breakout Session 3 & Report Back**
12:30 - 1:30	**Lunch**
1:30 – 2:45	**Breakout Session 4**
3:00 – 4:30	**Report Back**
4:30 – 5:15	**Action Items & Wrapup**

Workshop Theme – "Evidence"

1. **Evidence that the system requirements are correct, complete and understandable**

2. **Evidence That the Implementation Satisfies the Requirements with the Appropriate Degree of Confidence for a Safety Critical Application**

3. **Evidence that the operational and maintenance requirements and constraints are identified correctly and satisfied**

4. **Planning for Future SCC Work**

A Certification Framework

- To understand the context of our research, we have defined an abstract framework for certification
  - What is the aim of certification?
  - What is the underlying *science* related to certification?
  - What is the epistemological status and underlying theory of argument based notations?
  - What is being certified?
  - …
  - Is a safety case part of the evidence of safety claimed for a system?

What Should Software Certification Really Certify?

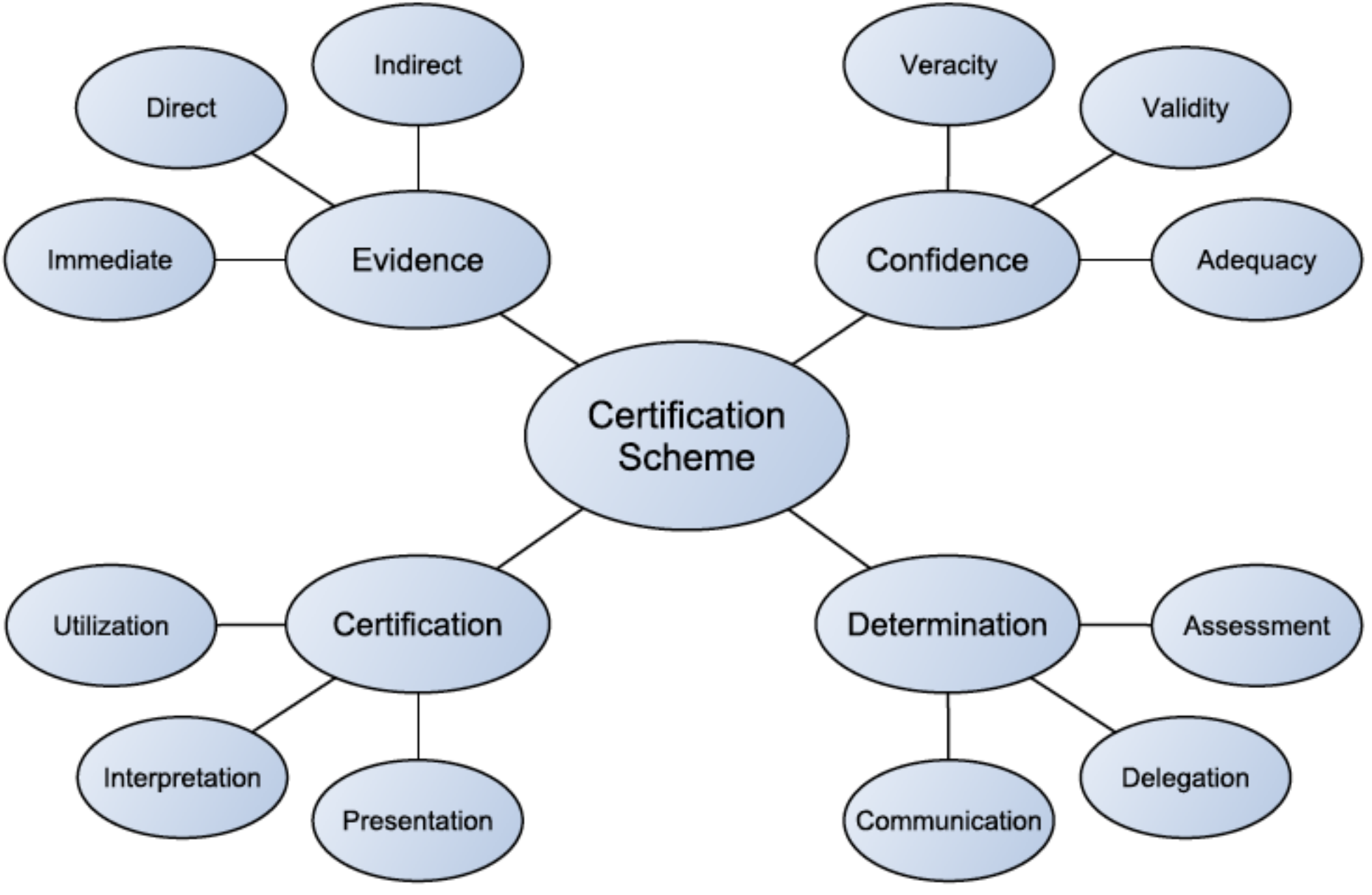
- Software certification should assess software artifacts, not just the code.
  - accepting that formal documentation is a part of the software artifact means that certification must take into account the software's environment
  - certifying the artifact should include checking the requirements, which means validating them against (a model of) the real world
- This includes epistemological questions that are (generally) not subject to proof, but only to scientific demonstration and argument.

A Certification **Framework**

- Once we accept that dependability does not rely only on simple yes/no questions, certification becomes quite an interesting and challenging problem.
- This is the motivation for our *certification framework*.
- Vaguely:
  - certification includes evaluation of engineering artifacts
  - furthermore, a successful evaluation results in a *certificate* being bestowed upon the artifact(s)

A Certification Framework

- *The goal of certification is to systematically determine, based on the principles of science, engineering and measurement theory, whether an artifact satisfies accepted, well defined and measurable criteria.*
- The challenge: develop a certification process that achieves this goal.
  - we must evaluate pieces of *evidence* about the artifact, which systematically increase our *confidence* (☺) that it is satisfactory, until the point where we make the *determination* that the artifact is acceptable, and assign a *certification*


Certification Scheme
Evidence
Direct
Indirect
Immediate
Confidence
Veracity
Validity
Adequacy
Certification
Utilization
Interpretation
Presentation
Determination
Assessment
Communication
Delegation

McSCert

Evidence

- Evidence embodies the *empirical* part of the certification effort. It consists of the "things" under consideration:
  - the real-world objects and documents that form the informational foundation of evaluation
  - examples of evidence specific to the software setting include source code, requirements, specifications, machine executables, models, test results, proofs of correctness, real-world trial results, etc
  - evidence can also include things like personnel qualifications/certifications and documented adherence to development processes

McSCert

Evidence

- Evidence in isolation from the other aspects is about how we identify, observe, measure, classify and organize items of interest:
  - not talking about their trustworthiness, accuracy, relevance, adequacy, …; these issues are epistemic and therefore fall into the confidence aspect
  - not talking about the performance of any observations or measurements: a pragmatic consideration which is a part of determination
  - do not treat certification artifacts — created during the certification activity — as evidence within the framework; from the "internal" point of view of certification, these are pragmatic side-effects and so we see them as part of the determination aspect (specifically, communication)

McSCert

Evidence

- Evidence is broken into subcategories based on how far removed it is from the product:
  1. *Immediate evidence*: evidence which is itself being evaluated; the parts of the candidate artifact: machine executables, source code, specifications, requirements documents
  2. *Direct evidence*: evidence which presents properties of the candidate; things that are directly about the immediate evidence: test results, proofs of correctness, static analysis results, hazards analyses, real-world trials, model checking results
  3. *Indirect evidence*: evidence which describes the circumstances relevant to the creation of the candidate; information about the development of the artifact: development processes, personnel qualifications, tool qualifications, content management systems
- But, we lack *metrics* that can serve as the direct evidence we need.

McSCert

Confidence

- Confidence is the *epistemic* aspect of certification: about our knowledge and judgements, how we value and reason about the evidence, consider and apply criteria, weigh the importance of pieces of information, and combine pieces of information
- Its focus is the rationale, ranging from qualitative and imprecise to quantitative and exact, about the evidence that is presented
- Examples of confidence specific considerations are: arguments, professional judgments, assumptions, uncertainty, weighting, inference (deductive or inductive), criteria, probability, trust and trustworthiness, soundness, relevance, tolerance

McSCert

Confidence

- We see confidence as divided into 3 sub-areas: veracity, validity and adequacy

1. *Veracity*: about the sources of our knowledge; deals with the level of trustworthiness or accuracy of pieces of evidence: measurement tolerance, precision, trust, reliability, reputability
2. *Validity*: about the interpretation of the evidence; encompasses the inferences and logical steps that we make: soundness, relevance, consistency, justifiability, defensibility, reasonableness
3. *Adequacy*: about the sufficiency of our knowledge; focuses on what is required of a system and the presented evidence to achieve certification: criteria satisfaction, completeness, comprehensiveness, sufficiency, conclusiveness, acceptability

McSCert

Determination

- Determination encompasses all of the *pragmatic* facets of the certification process.
- It denotes all of the "feet-on-the-ground" aspects of certification as an activity:
  - who does what, how measurement and evaluation takes place, how things are recorded and communicated, what is produced
- Having determination as a separate aspect allows us to focus on and talk about the real world issues involved in certification in isolation, thus drawing a clear line between theory and practice.

McSCert

Determination

- A tentative decomposition of determination:
  1. *Assessment*: about how evidence is examined, verified and evaluated, and how rationale is developed and scrutinized: inspection, re-running tests, checking sources, verifying authenticity, appraising rationale, identifying problems, auditing processes
  2. *Delegation*: about the human/social aspects of certification; the roles, responsibilities and functions of the parties involved: regulators, certifiers, independent evaluators, domain experts, professionals (engineers and otherwise)
  3. *Communication*: about expressing, relating, recording, tracking and archiving certification activities: document formats, mathematical knowledge management, media, security and encryption, clearance levels ("social information hiding")

Yogi Berra: "In theory there is no difference between theory and practice, but in practice there is."

McSCert

Certification

- A certification is the result of a successful evaluation activity
  - it is a "marker" or designator that results from the bid by a developer to certify or have certified their candidate product
- We see it as a signifier of that success; as such, certification in our framework embodies the *semiotic* aspect of the certification scheme.
- We see certificates as concrete entities: certifications are abstract
  - represented by different kinds of certificate

McSCert

Certification

- Semiotics has 3 aspects: syntactics (presentation), semantics (interpretation) and pragmatics (utilisation).

1. Presentation: how a certification is presented, realized or actualized; the real-world representation of the certification: certificate, database entry, proofs of certification, naming/numbering (e.g. IEC 61508)
2. Interpretation: about the meaning of a certification to various parties; the denotations and connotations of certification: records of certification, engineering log books, list of products certified, reputability of a certification
3. Utilization: the implications, restrictions and limitations on the use of a certification or standard; contexts of use and applicability: legal ramifications, liabilities, issues relating to international use/interpretation

McSCert

Certification

- The interpretation of a certification is loosely divided into its vindicative and indicative readings:
  - The *vindicative* interpretation consists of (a presentation of) all of the evidence, evaluation activities, involved parties, rationale, etc. that took place while obtaining the certification—the denotations of the certification, representing all of the actual facts that stand behind it
  - The *indicative* reading is what the certification entails to the "outside world"—the connotations of the certification, including any interpretation that goes beyond the established facts of the certification activities involved

McSCert

Conclusions

1. Software is complex. How can certification scale?

- we believe that the problem can be confronted gradually
- not all software systems need to be certified with the same level of rigor; not all development activities must be treated in the same way
- we can progressively tighten the certification scheme as appropriate methods become available, while still maintaining an approach that can essentially be applied to any system containing software

2. We lack metrics for software dependability:

- until more mathematical metrics become available we can still make use of engineering judgment, best practices and other qualitative forms of evidence
- this reflects the nature of software engineering: as a young branch of engineering, it is what Vincenti calls radical engineering, but as time passes it will become normal

Comment on Semantics of argument based frameworks

- Any box and arrow notation in SE requires a *semantics*, or else it is ultimately unreliable as an engineering tool! ☺
- Something to do with *argument*, as presented by Toulmin?
- But are these ideas precise enough for a semantics?
- Well, they are actually at a pre-scientific stage … ☹

McSCert

Comment on Semantics of argument based frameworks

- Toulmin argument schemes have levels of uncertainty associated with elements
- Many have observed that confidence is an important aspect of this kind of reasoning
- Some philosophers and epistemologists have observed that a Pascalian (frequency of occurrence, propensity) based explanation is impossible
- Some (e.g., Jon Cohen) have put forward a Baconian notion of probability, based on probative value: *confidence in the demonstration that some property is true*

McSCert

Comment on Semantics of argument based frameworks

- These ideas explain how legal reasoning works, like the meaning of *preponderance of evidence / balance of probabilities* and *beyond reasonable doubt*
- They support the understanding of the notion of confidence in scientific theories and how this changes with new evidence
- The notion of probability is not reducible to Pascalian axioms; it does not satisfy the usual laws of conjunction or negation
- A Bayesian basis for confidence is not possible

Comment on Semantics of argument based frameworks

- If the probability of some property (or event) E is X, then the probability of notE is in general not 1-X!
- If we have evidence for E, this does not imply anything about what evidence we have for notE. It could be 0!
  - Of course, the sum of the two must be ≤ 1
- If we have 2 bits of evidence for E and their probabilities are X and Y, then the joint probability is ≥ max {X, Y}, not ≤ min {X, Y}!
  - Think cumulative evidence in court cases!

McSCert

Finally

Toulmin Scheme and Rule of Inference