Proofs as Highest-Quality Evidence for Certification

Brian Larson

Kansas State University

May 4, 2015

One Question for Software-Based Medical Device Approval Applicants

What evidence shows the untested state-space is also safe and effective?

Must Distinguish Validation from Verification

- Verification says merely that software meets its specification.
- Validation says that the specification meets requirements, and that the requirements meet the design intent.

Validation, of course

Validation of requirements and specifications will always be inherently subjective.

Formal methods can help make requirements and specifications complete and unambiguous.

Will, intent, and need are inherently human.

Requirements: natural language, imprecise

Specification: formal, precise

Certifying Verification

Verification can be made precise.

Software certificates attest to conformance to specification.

Formal methods can give higher confidence than testing alone.

Verification is Objective; Validation is Subjective

Verification can be objective; validation will always be inherently subjective.

Validation is crucial, of course.

Validation $\equiv$ Systems Engineering

See INCOSE Handbook for state-of-practice systems engineering.

Proof ≡ Truth

> *Since proofs are incontrovertible mathematical truths, once a software component is certified, its trustworthiness (with respect to its specification) would presumably last for eternity.*[1]

Shao specifically limits software certification to verification.

Shao's certificates attest to conformance to specification (correctness) not fitness for purpose, safety, or efficacy.

Other formal methods can be helpful, but don't show conformance of software to specification.

[1] Zhong Shao, *Certified Software* in Communication of the ACM, December 2010 p.61

Medical Device Virtual Integration

Sometimes, formal methods (but not necessarily correctness proofs) are needed.

Can't test all combinations of interoperable medical devices, especially devices yet to be designed.

MD-VI verifies composition of classes of devices that meet a formal behavior specification.

Device conformance to specification may be informal (testing).

Correctness Proofs are Strongest Verification Evidence

Correctness proofs show programs conform to specifications for entire state space.

Issue: Semantic gap when programs and specifications need to be translated into a theorems prover's language (Why3, Coq, Isabelle/HOL, PVS, ACL2). What is actually proved?

Silver bullet? Programs, specifications, and independently-auditable correctness proofs written in same language.