Assurance for Software-Reliant Systems

Can you afford to test everything?

What will you do if you cannot test your high assurance system's dependability long enough?

How can you be sure that your system of systems will perform as required?

Are your assurance processes more guesswork than engineering?

At the Software Engineering Institute (SEI), we are developing tools and methods to document and predict the dependability of a system *where it is infeasible or too costly to depend only on testing.*

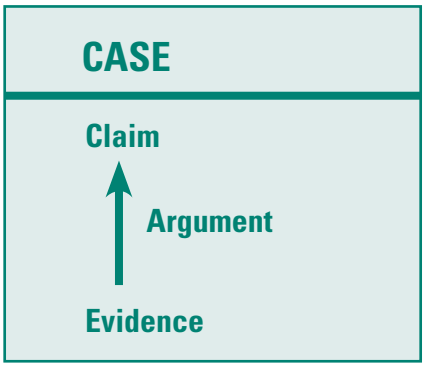
In particular, we are applying techniques for showing how software dependability claims can be supported by evidence derived from a combination of analysis and testing. The purpose of these techniques is to provide analysis-based assurance—the confidence, derived from means other than solely by testing, that a system or component of a system will perform as expected.

Analysis-based assurance can

- replace testing, where testing at all would be impossible
- augment testing where thorough testing would be infeasible or too costly
- reduce the number of tests that would be needed to assure the desired system dependability

Building an Assurance Case

Central to analysis-based assurance is the *assurance case* method. An assurance case is a structured set of arguments and a body of evidence showing that a system satisfies specific claims with respect to a given quality attribute such as reliability or security.



The assurance case is analogous to peer reviews for reducing defects. It provides a means to structure the reasoning that engineers use implicitly to gain confidence that systems will work as expected. It also becomes a key element in the documentation of the system and provides a map to more detailed information.

The concept of an assurance case has been derived from the safety case, a construct that has been used successfully for more than a decade to show that safety-critical systems meet their safety properties in areas such as flight control, railroad signaling, and nuclear reactor shutdown systems.

In support of the assurance case and analysis-based assurance concepts, we are developing case studies showing how to integrate software/system safety, security, and reliability assurance practices.

Assurance for Software-Reliant Systems

Related Web Site

www.sei.cmu.edu/dependability/consulting/assurance/

For More Information

For information about the SEI and its products and services, contact

Customer Relations

Phone: 412-268-5800

FAX: 412-268-6257

customer-relations@sei.cmu.edu

www.sei.cmu.edu

Software Engineering Institute

Carnegie Mellon University

4500 Fifth Avenue

Pittsburgh, PA 15213-2612

Applying Assurance Case Patterns

It is common for assurance case arguments with the same structure to appear in many different contexts. The recurring arguments are patterns of reasoning that encapsulate what evidence needs to be collected and what pitfalls are likely to occur in applying a particular type of argument. Such *patterns* also capture the expertise of people in an organization who know how to attain assurance.

Assurance case patterns provide a body of best practices in the form of ready-to-instantiate arguments where only the specific details need to be incorporated, reducing the effort needed to develop an assurance case. As a result, the use of these patterns helps an organization leverage scarce assurance resources.

An example of an assurance case pattern from the safety-critical milieu is the "as low as reasonably practicable" pattern that argues in this way: when a particular hazard cannot be eliminated or reduced to a negligible level, an argument needs to be developed showing that the risk has been reduced to a tolerable level and any further reduction is impractical.[1]

What Analysis-Based Assurance Offers Your Organization

- help in leveraging scarce assurance resources
- help in producing a reviewable artifact that provides
  - assurance of mission-critical properties
  - go/no-go criteria at different stages of development
- assessment of the impact of change
- development of and support for an engineering culture characterized by explicit articulation and verification of claims
- guidance for system-of-systems test and evaluation techniques
- evidence-based high-assurance practices

How the SEI Can Help Your Organization

- improve how your organization establishes and sustains a sense of assurance by reviewing the "as-is" state of your assurance practices and recommending a desired state and path forward[2]
- document assurance patterns used by expert engineers so that other engineers can use them repeatedly, making their tacit knowledge explicit
- apply best assurance practices from other organizations to address issues that are found in an examination of your organization's assurance processes
- train your engineers in the use of leading-edge assurance technologies
- provide services and tools for model-based analysis of embedded, real-time systems such as avionics, aerospace, automotive, and autonomous robotics[3]

1 In *Evaluating Hazard Mitigations with Dependability Cases*, John B. Goodenough (Software Engineering Institute) and Matthew R. Barry (Software-Intensive Systems, Inc.) discuss the valued added by a dependability case (an alternate name for an assurance case) to traditional hazard analysis (www.sei.cmu.edu/library/abstracts/whitepapers/dependabilitycase_hazardmitigation.cfm). Also, *Towards an Assurance Case Practice for Medical Devices* by Charles B. Weinstock and John B. Goodenough (Software Engineering Institute) explores the use of assurance cases for justifying claims of medical device safety (www.sei.cmu.edu/library/abstracts/reports/09tn018.cfm).

2 *Assurance Cases for Design Analysis of Complex Systems of Systems* by Stephen Blanchette (Software Engineering Institute) discusses the application of assurance cases as a means of building confidence that the software design of a complex system of systems will actually meet the operational objectives set forth in the project's top-level requirements (www.sei.cmu.edu/library/abstracts/whitepapers/Assurance-Cases-for-Design-Analysis-of-Complex-System-of-Systems-Software.cfm).

3 For more about these tools, see www.sei.cmu.edu/dependability/tools/aadl/index.cfm.

05/15/2010