Agenda

} What is an Assurance Case?
} Problems With Assurance Cases
} Hypotheses
} Notations and Tools
} Structured Assurance Case Process
} Assurance Standards Examined
} Practical Application Example
} Hypotheses Proved or Disproved
What Is an Assurance Case?
History of Assurance Cases

} Originally Only Safety Cases
   | Aerospace
   | Railways, automated passenger
   | Nuclear power
   | Off-shore oil
   | Defense

} Security Cases
   | Use compliance rules more than an assurance case

} Cases for Business Critical Systems
Definition of Safety Case

From Adelard’s ASCE manual:

“A documented body of evidence that provides a convincing and valid argument that a system is adequately safe for a given application in a given environment.”
Generalizing that definition

A documented body of evidence that provides a convincing and valid argument that a specified set of critical claims regarding a system’s properties are adequately justified for a given application in a given environment.
Where is an Assurance Case Used?

- Critical systems under regulation or acquisition constraints
- Third-party certification, approval, licensing, etc.
- Documented body of evidence required
- Need a compelling case that the system satisfies certain critical properties for specific contexts
- Examples: DO-178B, Common Criteria, MIL-STD-882D
- “safety case”, “certification evidence”, “security case”…

Collectively we’ll refer to them as “assurance cases”
Problems With Assurance Cases
Problems with Assurance Cases

There are problems in every aspect of assurance cases:
- Building them
- Reviewing them
- Maintaining them
- Reusing them

Problems result from:
- Volume of material
- Little structuring support
- Ad hoc “rules of evidence”
Building the Assurance Case – 1

} Most guidance is:
  | strong on excruciating detail for format
  | weak on gathering, merging, and reviewing evidence

} Guidance often uses the “cast a wide net” tactic
  | Assurance costs time and money
  | “Squandered diagnostic resources”
  | Some work on a “portfolio management” approach
With free format text and no tool support:

- coordination is hard
- tracking is hard
- workflow management is hard

Imagine building a 500 page project plan by hand, on paper
Reviewing the Assurance Case – 1

} Stacks of free-format text makes review tedious
    | Hard to see linkages or patterns
    | Hides key results in sheer volume

} Weak guidance on review of arguments and evidence
    often results in ad hoc criteria
    (be very nice to your reviewer!)

} Rarely is there explicit guidance for weighing conflicting or inconsistent evidence
“Often viewed as irrefutable, evidence is, in fact, an interpretive science, refracted through the varying perspectives of different disciplines. ... [Judging evidence requires] reasoning based on evidence that is incomplete, inconclusive, and often imprecise.”

_The Evidential Foundations of Probabilistic Reasoning_, David Schum
The one thing more brittle than software is – the associated assurance case

It is difficult to understand impact of a change on assurance structure because:

- volume of information is immense
- impact of a change on assurance structure is complex
Reasons for change

- The claims and/or evidence have changed
- Arguments no longer valid or new ones needed
- Evidence is irrelevant or new evidence needed
- “Weak link effect” of discrete systems compounds problem

Revalidation costs are a major burden

“Breakage” of successive dependencies
Reusing the Assurance Case – 1

} Assurance case frameworks are rarely the subject of study per se

} More attention for these would be useful
  | tool support
  | idioms and templates
  | extracting patterns for future use
Relationship among claims, arguments, and evidence
- not often explicit
- hard to distinguish the reusable from the project specific portions of assurance case

Compare this with building a deck with the help of a project planning tool
Hypotheses
Hypotheses

} All Assurance Cases Have Similar Components
Assurance is assurance is assurance…

} An Assurance Standard Implies the Structure
| The standards document implies some structure of an assurance case that would conform to it

actual or implied structure of an assurance standard

inherent structure of assurance case instantiated from that standard
Notations and Tools
Notations Considered

} Toulmin Structures (law domain), 1958
  | Claim, Qualifier, Data, Warrant, Backing, Reservation

} Goal Structuring Notation (GSN): T. Kelly, 1998
  | Main node types: Goal, Strategy, Solution
  | Supporting nodes: Assumption, Justification, Context, Model, Notes

} ASCAD (Adelard Safety Claims Arguments Data), 1998
  | described in the ESPRIT SHIP project
  | Claim, Argument, Evidence

Selected ASCAD for its simplicity
The Tool Selected

} Investigated and tested three tools
  | Structured Evidential Argumentation System (SEAS), SRI
  | Wisdom Pad, Expert Decision Systems (EXDS), Inc
  | The Adelard Safety Case Editor (ASCE), Adelard

} Selected Adelard Safety Case Editor (ASCE)
  | Supports both ASCAD and GSN
  | Graphical user interface to arrange and connect nodes
  | Rules that identify structure errors like a compiler
  | Structured and unstructured data behind each node
  | Hyperlinks to internal and external references
ASCAD Entities and Tool Notation

Claim = assertion to be proven

Argument = how evidence supports claim

Evidence = required document
Structured Assurance
Case Process
Developing a Structured Assurance Case

Non-Deterministic Flow

Specify Top-level Claim

Related claim needed?

Specify Related Claim

Evidence needed for this claim?

Identify Supporting Evidence

Argument needed for this claim?

Develop Argument

Are all child claims “necessary & sufficient” for their parent claims?

Done

© 2009 The MITRE Corporation. All rights reserved
Assurance Standards Examined
Standards Selected for Mapping into Structures

| represents the biggest divergence from Adelard’s safety-critical domain

} RTCA/DO-178B Software Considerations in Airborne Systems and Equipment Certification
| the only one of the three that sits firmly within Adelard’s territory

} ISO 14971 Medical devices – Application of risk management to medical devices
| in a domain for which Adelard’s tool has not yet been used
| risk management approach is different from the other selected standards
Process Mechanics – 1

} Goals:
   | Avoid misrepresenting the standard with our own ideas
   | Be consistent in structuring each standard

} Methods:
   | Devised a minimal set of rules for mapping each standard
   | Tried to apply those rules mechanically in our mapping

} Mapped the entire standard or a usable subset
   | DO-178B and ISO 1497 completely
   | Common Criteria – only EAL4

Result: Each mapping should still be recognizable
ASCAD notation requires
- Quasi-hierarchical (multiple parents are allowed)
- One claim at top of hierarchy
- Subordinate claims below
- Evidence nodes at the bottom
- Argument positioned between claim and its evidence

We deviated somewhat
- Arguments also positioned between claim and sub-claims
The Common Criteria – Top Level

CLAIM EAL4
[Confidence in Security because the product has been] methodically designed, tested, and reviewed

ARGUMENT ACM Configuration Management
ARGUMENT ACM AUT CM Automation
ARGUMENT ACM_CAP CM Capabilities
ARGUMENT ACM SCP CM Scope
ARGUMENT ADV Development
ARGUMENT ADV FSP Development
ARGUMENT ADV HLD High-Level Design
ARGUMENT ADV IMP Implementation Representation
ARGUMENT ADV LLD Low-Level Design
ARGUMENT AGD Guidance Documents
ARGUMENT AGD ADM Administrator Guidance
ARGUMENT AGD ADP Administrator Data Protection
ARGUMENT AGD ASM Administration Support Model
ARGUMENT AGD ASI Administration Security Implementation
ARGUMENT AGD USR User Guidance
ARGUMENT ALC Life Cycle Support
ARGUMENT ALC DVS Development Security
ARGUMENT ALC RCR Representation Corespondence
ARGUMENT ALC TAT Tools and Techniques
ARGUMENT ALC LCD Life Cycle Definition
ARGUMENT ALC OPT Depth
ARGUMENT ALC OFF Functional Tests
ARGUMENT ATE Tests
ARGUMENT ATE COV Coverage
ARGUMENT ATE IND Independent Testing
ARGUMENT ATE OPT Depth
ARGUMENT ATE FUN Functional Tests
ARGUMENT AVA Vulnerability Assessment
ARGUMENT AVA_MSU Misuse
ARGUMENT AVA VLA Vulnerability Analysis
ARGUMENT AVA SOF Strength of TOE Security Functions
ARGUMENT AGD ADM Administrator Guidance
ARGUMENT AGD USR User Guidance
ARGUMENT ALC Life Cycle Support
ARGUMENT ALC DVS Development Security
ARGUMENT ALC RCR Representation Corespondence
ARGUMENT ALC TAT Tools and Techniques
ARGUMENT ALC LCD Life Cycle Definition
ARGUMENT ALC OPT Depth
ARGUMENT ALC OFF Functional Tests
ARGUMENT ATE Tests
ARGUMENT ATE COV Coverage
ARGUMENT ATE IND Independent Testing
ARGUMENT ATE OPT Depth
ARGUMENT ATE FUN Functional Tests
ARGUMENT AVA Vulnerability Assessment
ARGUMENT AVA_MSU Misuse
ARGUMENT AVA VLA Vulnerability Analysis
ARGUMENT AVA SOF Strength of TOE Security Functions

© 2009 The MITRE Corporation. All rights reserved
Issues Encountered While Structuring Common Criteria

} Highly structured
   | Easy to map one assurance level into ASCAD
   | Introductory paragraphs worded like justifications
      { Fit better as argument nodes
      { No claims except at the top

} No “objectives” paragraph at component/bottom level
   | Leaving an empty argument at that level
   | Only evidence requirements for those components

} More complex evidence requirements than our mechanical rules allowed for
CLAIM: 7.0 SCM process is properly established and executed

ARGUMENT: (not explicit)
Satisfactory SCM process requires three characteristics

CLAIM: 8.0 SQA process is properly established and executed

ARGUMENT: (not explicit)
Satisfactory SQA process includes six elements

CLAIM: 2.0 System Aspects are Taken into Account

ARGUMENT: (not explicit)
Certification expects all systems considerations to be addressed

CLAIM: 3.0 Software Life Cycle is properly defined

ARGUMENT: (not explicit)
All three areas are based on “best practices” and have detailed sub-claims

CLAIM: 4.0 Software Planning Process is executed

CLAIM: 5.0 Software Development Process is executed as planned

CLAIM: 6.0 Software Verification [low-level] and Integration Process

ARGUMENT: (not explicit)
Satisfactory verification covers products of all processes

CLAIM: 9.0 Certification Liaison process comprises three factors

ARGUMENT: (not explicit)
Satisfactory Certification Liaison process comprises three factors

CLAIM: DO-178B Software Considerations are taken into account

ARGUMENT: (not explicit)
Certification expects all factors be included
Issues Encountered While Structuring DO-178B

} Time is not inherently an element in ASCAD notation
   | sub-claims, and evidence were laid out in approximately their chronological order of use from left to right

} DO-178B does not include linkages between the generation of one artifact and its later use
   | We consulted an expert authorized to perform certification, a Designated Engineering Representative (DER)
   | DO-178B does not specify all of the artifacts that the certification evaluator expects to examine
   | Supplier knows that the DER expects to see the implied documentation
Issues Encountered While Structuring ISO 14971

} No direct relation between document structure and the structure of the intended assurance case

| Major sections correspond to legs on the hierarchy
| Statements representing claims, arguments, or evidence, have to be identified by analyzing the words and phrases

} One generic evidence type referenced in many places

} Document defines once what is instantiated several times

} Risk Control: under Risk Evaluation in the hierarchy, but is an optional level of decomposition
Practical Application
Example
Practical Application Example – Background

Government experiment in formal methods
- multiple authentication systems and an access log
- software developed for them by a contractor
- using formal methodology, validated by a third party

The researchers provided us with
- their Common Criteria Protection Profile document
- related Security Target document, from developer
- EAL5 targeted

Documents addressed Common Criteria components plus
- hierarchical arrangement of assumptions and policies
- objectives that address the policies
- threats, as they relate to the assumptions
Structuring the Experiment’s Assurance Case

} Combined Protection Profile with Security Target

} Created three separate structures in ASCAD
  | Security assurance requirements
    { Enhanced our EAL4 structure to EAL5
    { Amended it according to the protection profile and security target
  | Security functional requirements
  | Security threats, assumptions, and security policies
    { From tables in the protection profile
Lessons Learned from the Experiment

- Structuring revealed missing dependencies
  - Many security functional requirements list dependencies

- Identified security threats went unanswered
  - Most threats were connected to requirements below
  - At least one had nothing below
  - Others may be insufficiently answered
Hypotheses Proved or Disproved
Second Hypothesis Proved or Disproved?

} An Assurance Standard Implies the Structure

| DO-178B indicates structure in text and tables
| Common Criteria implies structure through its own structure
| ISO 14971 text suggests a reasonable approach for organizing

} Based on this limited trial …

One way or another, cases based on a given standard will inherently tend to be similar in structure
First Hypothesis Proved or Disproved?

All assurance cases have similar components

- Not clear that standards or assurance cases will have similar components
- Use of a structuring notation helps identify gaps
- Allows the applicant to present a case in a consistent manner
- Rigor of a claim-argument-evidence structure creates fulfillment of the original hypothesis for a given standard, regardless of product
- Makes consistency across different assessors more likely
- Use of a consistent notation across standards is at least feasible
- Opens possibility of tool use to identify gaps
Using Our Results

- Structured Standards can serve as templates – especially if we
- **Enhance structure of Common Criteria EAL4**
  - Create empty argument nodes where they are needed
  - Document those nodes to be filled in for each assurance case
  - Divide evidence nodes to reference one thing each
- **Enhance structure of DO-178B**
  - Explicitly incorporate implied documentation and precedences
- **Enhance structure of ISO 14971**
  - Create empty argument nodes where they are needed
  - Document those nodes to be filled in for each assurance case
  - Document the need to create a set of “risk evaluation” nodes for each risk
  - The generic evidence node might become several nodes in the template
Conclusion

} Tools such as ASCE and its notation are applicable to a broad range of assurance standards

} Mapping a standard into a notation may be a little time-consuming but is not difficult

} Using mappings as assurance-case templates is only a side benefit

} Structuring a new standard as it is being written can help to ensure completeness and avoid complexity
Credits

} Principal Investigators
   | Chuck Howell
   | Jim Moore

} Research Assistant
   | Samarina Lowrie