Assurance Cases for Design Analysis of Complex System of Systems Software

Presented at
AIAA Infotech@Aerospace Conference
Software Assurance Session
8 April 2009

Stephen Blanchette, Jr.
Assurance Cases for Design Analysis of Complex System of Systems Software

Carnegie Mellon University, Software Engineering Institute, Pittsburgh, PA, 15213

Presented at the American Institute for Aeronautics and Astronautics (AIAA) Infotech@Aerospace Conference, 7-9 Apr 2009, Seattle, WA.
Problem: SoS are Common, but Difficult

Huge systems of systems (SoS) development efforts

- Have become quite commonplace as aerospace/defense solutions
  - The Global Earth Observation System of Systems (GEOSS)
  - The US Army’s Future Combat Systems
  - The US Coast Guard’s Deepwater
  - Etc.
- Exhibit extreme complexity – they’re hard to understand in detail
- Are difficult, even impossible, to test adequately using traditional methods
- Use software as an enabler of SoS functionality

Often, SoS elements are dispersed geographically, adding complexity and making predictability of behavior difficult

How can project management, stakeholders, and decision makers achieve some reasonable level of confidence that software for a large-scale, dispersed SoS will meet operational needs, even before development of much of the software?
Recent Experience Led to Assurance Case Use

An SEI team was tasked with answering this question in the face of:

- *the* software design was actually *many* software designs documented in many places and with a *tremendous* volume of data
- the designs were not yet complete and would not be complete until after many production decisions had to be made
- the need to relate the software to operational needs
- the desire to base conclusions as much as possible on *actual data* rather than on optimistic plans and confident assertions

Constraints

- short time frame: a project level review had a hard deadline
- limited availability of personnel, all with varying levels of domain knowledge
We Had to Relate Operational Needs to Software

The top-level SoS requirements were expressed in military terms.

- A requirements analysis would be repeating work already done
- Errors in requirements traceability could skew the entire analysis

We decided against re-casting the operational needs in terms of software and instead analyzed the software contributions to the definitive characterization of those needs—the Key Performance Parameters (KPPs).

The size of the analysis space, the complexity of the task, and the desire to leverage data suggested an assurance case approach

- The analysis was a structured decomposition of each KPP into more precise statements that could be more readily assessed in terms of evidence.
  - Challenge: be logical and consistent but avoid accumulating too much detail
  - Use engineering judgment to leap from higher-level concepts to lower-level ones
An Assurance Case is a Structured Argument

An assurance case

- is a generalization of a safety case
- presents an argument (similar to a legal case) that a system has or satisfies some property in a given context
- requires claims, evidence, and an argument linking evidence to claims
- should be sound and complete to justify belief in the main claim
- should be based on objective evidence

Goal Structuring Notation (GSN)

- graphically presents the argument by showing how claims are broken down into sub-claims until arriving at a sub-claim supported by evidence.

For our purposes, we used assurance cases to demonstrate that a SoS software design supported each of the SoS KPPs, as in the claim “The SoS will satisfy KPP k”
An Example

Assume we’re developing a SoS for a DoD project

• The SoS must exchange information with other systems
  – Our SoS is subject to the Net Ready Key Performance Parameter (NR KPP):
  • The system...must support Net-Centric military operations. The...system...must be able to enter and be managed in the network, and exchange data in a secure manner to enhance mission effectiveness. The...system...must continuously provide survivable, interoperable, secure, and operationally effective information exchanges to enable a Net-Centric military capability.

Chairman of the Joint Chiefs of Staff Instruction, “Interoperability and Supportability of Information Technology and National Security Systems,” CJCSI 6212.01E, 2008.
We Can Express the NR KPP Diagrammatically

A1
It is assumed that hardware components of the network can be ignored in this analysis.

Ctx1
Source Document: CJCSI 6212.01E

Clim1
The system of systems supports Net-Centric military operations

Nte1
This diagram focuses on threshold level KPP satisfaction

Clim2
The SoS is able to enter and be managed in the network

Clim6
The SoS is able to exchange data in a secure manner to enhance mission effectiveness

Clim13
The SoS continuously provides survivable, interoperable, secure, and operationally effective information exchanges
Primary Sub-claim #2

Clm3
The SoS network conforms to the relevant standards

Clm2
The SoS is able to enter and be managed in the network

Clm5
The SoS network management software is able to establish, control, and terminate connections to the network

Clm4
The SoS communications software and protocol stack conform to the relevant standards

Ev1
Architecture documents

Ev2
Test plans/results related to standards compliance

Ev3
Architecture documents

Ev4
Architecture evaluation results

Ev5
Results of preliminary field tests and experiments
Primary Sub-claim #6

Clm6
The SoS is able to exchange data in a secure manner to enhance mission effectiveness

Clm7
The security requirements (e.g., data security, information assurance, access control) for the SoS are met

Clm8
Security requirements are identified adequately across the SoS

Clm9
Security components within the design are consistent across the SoS

Clm10
Verification/validation of SoS security features is adequately planned and resourced

Clm11
The SoS supports information exchanges

Clm12
Information to be exchanged is defined

Ev6
Requirements database showing security requirements

Ev7
Architecture and design documents

Ev8
Architecture evaluation results

Ev9
Test plans and descriptions

Ev10
Formally tracked risks

Ev11
Requirements database showing information transfers

Ev12
Results of preliminary field tests and experiments
Primary Sub-claim #13

Clm13
The SoS continuously provides survivable, interoperable, secure, and operationally effective information exchanges

Clm11
The SoS supports information exchanges

Clm12
Information to be exchanged is defined

Ev11
Requirements database showing information transfers

Ev12
Results of preliminary field tests and experiments

Ev13
Requirements database showing applicable timelines

Clm14
The SoS information exchanges occur within prescribed timelines

Clm15
Timelines are defined

Ev14
Results of preliminary field tests and experiments

Ev15
Modeling & simulation results

Ev16
Formally tracked risks
A Diagram Helps Visualize the Completed Case

Logic Should Hold…Even Without the Diagram
Scoring Can be Used to Express Risk

First, Develop Scoring Rules

<table>
<thead>
<tr>
<th>For Evidence</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Green</strong></td>
<td>Evidence is complete and adequate</td>
</tr>
<tr>
<td><strong>Yellow</strong></td>
<td>Evidence is incomplete or planned for the future</td>
</tr>
<tr>
<td><strong>Red</strong></td>
<td>Evidence is complete but inadequate, planned but now late, or non-existent</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>For Claims</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Green</strong></td>
<td>All lower-level claims and supporting evidence are green</td>
</tr>
<tr>
<td><strong>Yellow</strong></td>
<td>Some lower-level claims and supporting evidence are a combination of yellow and red</td>
</tr>
<tr>
<td><strong>Red</strong></td>
<td>All, or an overwhelming majority of, lower-level claims and supporting evidence are red</td>
</tr>
</tbody>
</table>

Then, Work Upward from Evidence…
Scored Diagram Provides a Roadmap…

Quality of the Evidence Drives Assessment of Claims…and Relative Risk

Legend:
- Green = Low Risk
- Yellow = Medium Risk
- Red = High Risk
Assurance Cases are Helpful in the SoS Space

Assurance cases gave us a way of organizing a nebulous task and gave us a means of selecting among innumerable artifacts to study. They brought order to complexity.

- Due to time constraints we had to focus on big picture risks
- A more thorough analysis might have identified additional risks or strong points

The assurance case technique is a powerful tool for analyzing large and complex SoS software design.

- It provides a means of taking a crosscutting look at SoS
- It gives managers answers about design progress that are rooted in facts and data instead of opinions based upon hope and best intentions.
Contact Information

Stephen Blanchette, Jr.
Senior Member of the Technical Staff
Acquisition Support Program
Telephone: +1 412-268-6275
Email: sblanche@sei.cmu.edu

U.S. mail:
Software Engineering Institute
4500 Fifth Avenue
Pittsburgh, PA 15213-2612
USA

World Wide Web:
www.sei.cmu.edu

Customer Relations
Email: customer-relations@sei.cmu.edu
Telephone: +1 412-268-5800
SEI Phone: +1 412-268-5800
SEI Fax: +1 412-268-6257
NO WARRANTY

THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN “AS-IS” BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

Use of any trademarks in this presentation is not intended in any way to infringe on the rights of the trademark holder.

This Presentation may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

This work was created in the performance of Federal Government Contract Number FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013.