Software Solutions Symposium 2017

March 20–23, 2017

Measuring Complexity for System Safety Assurance

Sarah Sheard, Mike Konrad, Bill Nichols, Charles B. Weinstock

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

© 2017 Carnegie Mellon University

[Distribution Statement A] This material has been approved for public release and unlimited distribution.

Copyright 2017 Carnegie Mellon University

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense.

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.

[Distribution Statement A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

This material 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.

DM-0004582

Purpose

Question: When is a system is too complex to certify as safe?

Possible Solution: Error propagation complexity algorithm

© 2017 Carnegie Mellon University

[Distribution Statement A] This material has been approved for public release and unlimited distribution.

Background

2014: FAA requested research on system complexity and safety, including definition and measurement

Requested avionics-specific definitions of complexity and complexity measure(s)

What threshold of that measure might make a system too complex to be able to assure safety?

Funded SEI research project

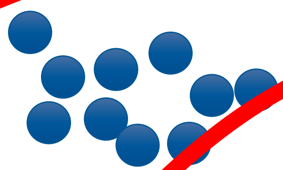
Output is Final Report and 5 white papers (Complexity overview, Candidate Measures, Safety Cases, Complexity Calculation Algorithm, Algorithm Test)

© 2017 Carnegie Mellon University
[Distribution Statement A] This material has been approved for public release and unlimited distribution.

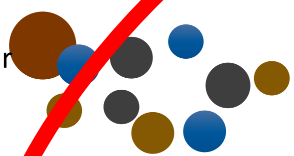
Complexity is complex

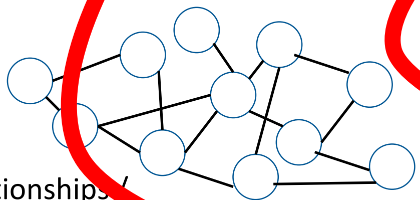
What is "Complexity"

Size (number)

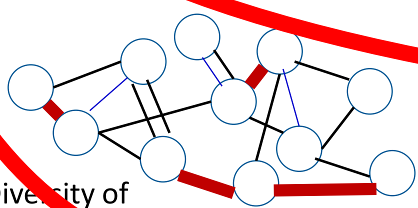


Diversity or Variety





Relationships / Interconnections



Diversity of Relationships / Interconnections

WHAT is complex?

Software

Hardware

Avionics

Plane ?

Requirements

Designs

Models

Tests

…?

How complex is it?

Cyclomatic Complexity

Fan-out and Fan-in

Requirements Churn

What about Complexity matters to Safety?

© 2017 Carnegie Mellon University

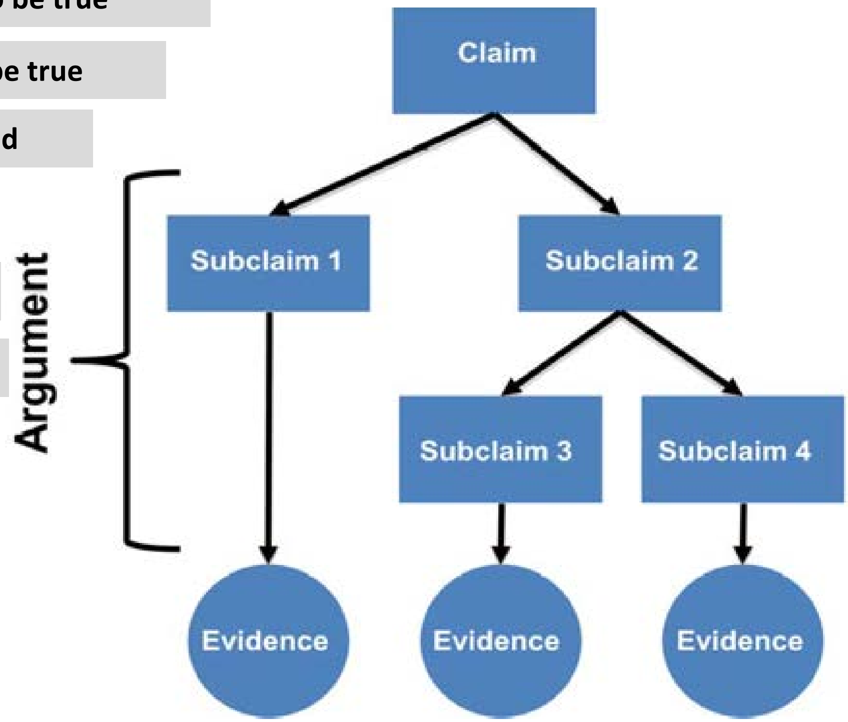
[Distribution Statement A] This material has been approved for public release and unlimited distribution.

Safety Case (type of Assurance Case)

For "The System Is Safe" to be true

Subclaim 1 and 2 must be true

Argument must be sound



For Subclaim 1 to be true

There must be X evidence

For Subclaim 2 to be true

Subclaim 3 and 4 must be true

For Subclaim 3 and 4 to be true

There must be Y evidence

There must be Z evidence

Multiple technical exchange meetings

© 2017 Carnegie Mellon University

[Distribution Statement A] This material has been approved for public release and unlimited distribution.

2 Breakthroughs

1. Evaluate the complexity ***of the safety case***

   But: the safety case isn't "complete" until the aircraft is designed, built, tested, with all software on board…

2. **Estimate** the size of the safety case **early**

   How much work (analysis, documentation, meetings etc.)

   will it take to prove the system is safe?

   (# potentially cascading error conditions)

   - Assume component assurance process will remain as is
   - Big open question is errors cascading from one component to another
   - Order of magnitude probably ok

© 2017 Carnegie Mellon University

[Distribution Statement A] This material has been approved for public release and unlimited distribution.

Our Method

Primary Assumption:

Early design work on new system* has resulted in a model of the system architecture at a high level including

- system modes
- active components and their interconnections in each mode
- possible failure conditions that could propagate outward

Many additional assumptions made to arrive at notional thresholds for between systems that are assurable as safe and systems that are too complex to assure as safe

*For future research: precedented systems

© 2017 Carnegie Mellon University
[Distribution Statement A] This material has been approved for public release and unlimited distribution.

Assume

Multiple modes; errors can propagate in each

- Sum over all modes

Multiple components; errors can propagate from each one

- Sum over all components active in that mode

Multiple propagation points on components

- Sum over all (outward-) propagation points

Then,

For each propagation point, each component, each mode:

- Multiply number of failures that could propagate out by number of places the failures could reach (Fanout)

© 2017 Carnegie Mellon University
[Distribution Statement A] This material has been approved for public release and unlimited distribution.

Algorithm

Sum over all system modes:

Sum over all components active in a given mode:

Sum over all propagation points (p-points) for this component:

of:

$$\left\{\begin{array}{l}\text{Number of failures}\\ \text{that could propagate}\\ \text{out from this p-point}\end{array}\right\} \text{ times } \left\{\begin{array}{l}\text{Fanout from}\\ \text{this p-point}\end{array}\right\}$$

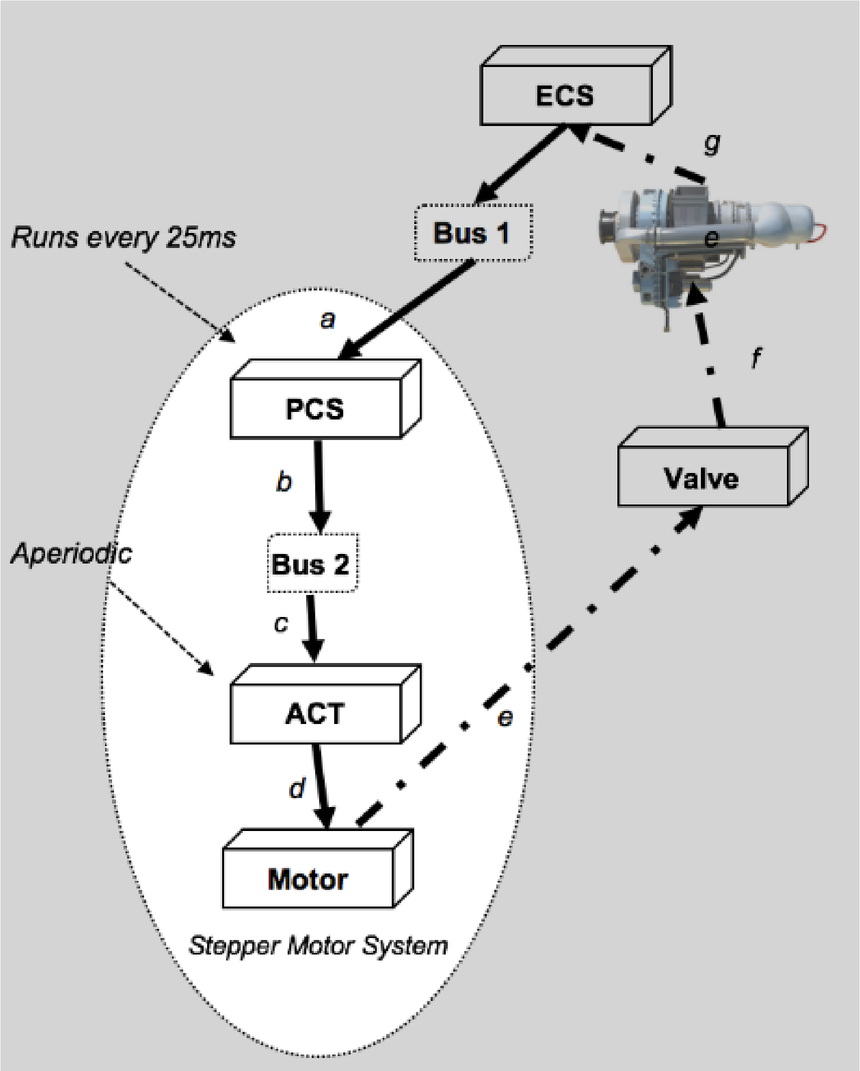
© 2017 Carnegie Mellon University

[Distribution Statement A] This material has been approved for public release and unlimited distribution.

Example 1: Stepper Motor System

1. From High Level design:
- 1 mode
- Interfaces shown
- Treat Bus 2 as a component*
- 4 components plus Environment
- #P-points = 1 for all components
- Fanout always = 1
- \- - - - - - - - - - - - - -

2. From Error Model:
- Errors from Environment to SMS: 3
- Errors from PCS to Bus 2: 4
- Errors from Bus 2 to ACT: 3
- Errors from ACT to motor: 3
- Errors from Motor to Envt.:3



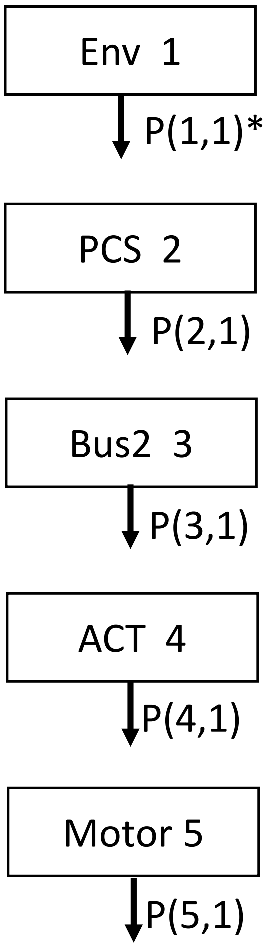
Ref: Konrad 2015b of Final Report

*Since it can be a source of a failure condition

© 2017 Carnegie Mellon University
[Distribution Statement A] This material has been approved for public release and unlimited distribution.

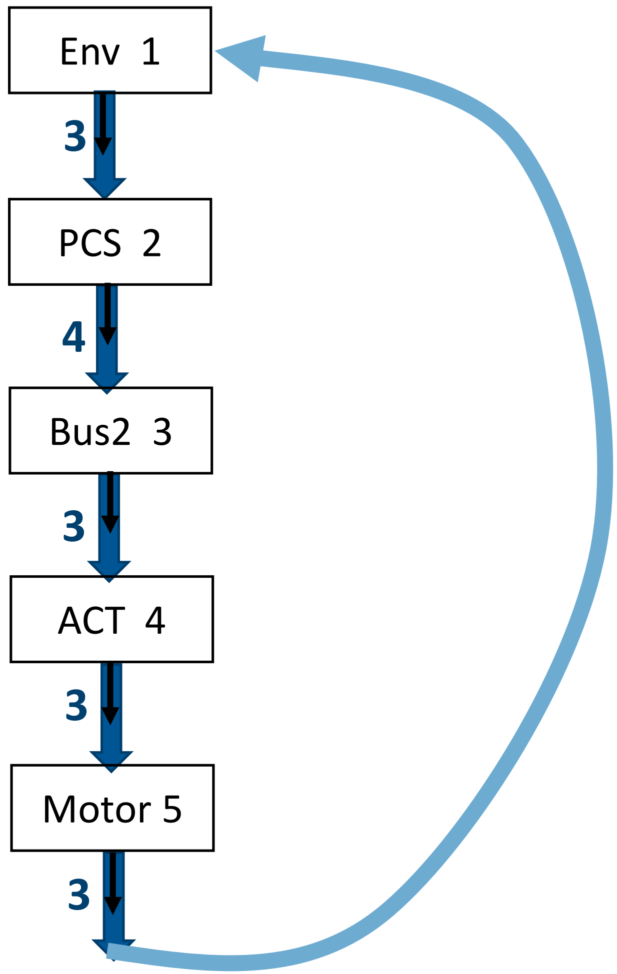
Calculating EPC (for one mode)

First step



*Notation P(component#, p-point#)

Second step



Third step

Sum of (#failures*Fanout for all P-points of Component x)

x	Sum
1	3*1 = 3
2	4*1 = 4
3	3*1 = 3
4	3*1 = 3
5	3*1 = 3

Total all components

Error Propagation Complexity = **16**

© 2017 Carnegie Mellon University

[Distribution Statement A] This material has been approved for public release and unlimited distribution.

Potential Applications of This Research

- FAA uses as evidence that they need to ask manufacturers to provide documented safety cases rather than just standards compliance
- Manufacturers (1st and lower tiers) use estimate of design complexity to estimate their own QA effort
- Comparison of designs by how complex are their error propagation potentials
- Complexity as an indicator of risk, to be tracked using standard techniques
- Future research into “how much can we discount the complexity of a system given that X% has been used before?” can be framed as “Credit for Precedence” and ties to “Recertification” questions. Much interest across SEI and at CMU for this topic

© 2017 Carnegie Mellon University
[Distribution Statement A] This material has been approved for public release and unlimited distribution.

Contributions

First tie of system complexity to safety that we know of

Use Safety Case review time estimate as a proxy for complexity

With architecture model, program, can estimate complexity of different alternatives as they will relate to safety, and can compare them

© 2017 Carnegie Mellon University

[Distribution Statement A] This material has been approved for public release and unlimited distribution.

Recommended Future Research

1) Apply and validate to larger system at real-life scale.
2) Study special cases, assumptions, and limitations more specifically
   a) Including what about precedented system components: should these count as less complex because we are familiar with them? How?
   b) Including tweak numbers for whether the Applicant has provided an organized assurance case or not. How does this affect FAA effort?
   c) Determine effect of having models to different levels of detail. Is there a notional "complexity reduction" curve?
3) Expand fault model to include more than error propagation: emergent behavior, concurrency, and cybersecurity
4) Develop guidelines for safe assurance practices and design guidelines to reduce software complexity

© 2017 Carnegie Mellon University

[Distribution Statement A] This material has been approved for public release and unlimited distribution.

For More Information: Report and White Papers

http://resources.sei.cmu.edu/library/asset-view.cfm?assetID=483758

Report:

Sheard 2016a. Sheard, Sarah, Michael D. Konrad, Charles B. Weinstock, and William Nichols. “Definition and Measurement of Complexity in the Context of Safety Assurance.”

White Papers

Konrad 2016a. Konrad, Michael D. and Sarah Sheard. “FAA Research Project on System Complexity Effects on Aircraft Safety: Literature Search to Define Complexity for Avionics Systems.”

Nichols 2016. William Nichols and Sarah Sheard. “FAA Research Project on System Complexity Effects on Aircraft Safety: Candidate Complexity Metrics.”

Sheard 2016b. Sarah Sheard, Charles B. Weinstock, Michael D. Konrad, and Donald Firesmith. “FAA Research Project on System Complexity Effects on Aircraft Safety: Identifying the Impact of Complexity on Safety.”

Konrad 2016b. Michael D. Konrad and Sarah Sheard. “FAA Research Project on System Complexity Effects on Aircraft Safety: Estimating Complexity of a Safety Argument.”

Konrad 2016c. Michael D. Konrad, Sheard, Sarah, Charles B. Weinstock, and William Nichols. “FAA Research Project on System Complexity Effects on Aircraft Safety: Testing the Identified Metrics.”

© 2017 Carnegie Mellon University

[Distribution Statement A] This material has been approved for public release and unlimited distribution.

Contact Information

Sarah A. Sheard, Ph.D.
Principal Engineer
Office: (412) 268-7612
sheard@sei.cmu.edu

Michael D. Konrad, Ph.D.
Principal Researcher
Office: (412) 268-5813
mdk@sei.cmu.edu

Charles B. Weinstock, Ph.D.
Principal Researcher
Office: (412) 268-7612
weinstock@sei.cmu.edu

William R. Nichols, Ph.D.
Senior Member, Technical Staff
Office: (412) 268-1727
mdk@sei.cmu.edu

Software Engineering Institute
Carnegie Mellon University

© 2017 Carnegie Mellon University

[Distribution Statement A] This material has been approved for public release and unlimited distribution.

#1 Recommended Future Research: Precedence

- Study complexity "discounts" that we should give to known or precedented system components because they are familiar
  - How many error propagations (from model) have already been proven not to be unsafe and thus need less review?
  - How can this be applied to, say, *slightly* different configurations? How do you measure "slightly"?
  - How can this be applied to slightly different hazards?
  - What is safety effect of higher-capability component compared to existing?
- Other areas can contribute:
  - How organizations today currently allow credit for testing already done
    - FAA and aircraft re-certification (e.g. longer fuselage)
    - FDA and medical devices
    - Regression testing
  - Estimate of the amount of impact caused by a change (hardware, then software)
  - Understanding how much of the problem could be solved by nearly-independent, modularized, proven-correct components

© 2017 Carnegie Mellon University

[Distribution Statement A] This material has been approved for public release and unlimited distribution.