CMU Task RT-119
—
Systemic Assurance

Bill Scherlis
Professor and Director
Institute for Software Research (ISR)
School of Computer Science (SCS)
scherlis@cmu.edu

6th Annual SERC Sponsor Research Review
Dec 2014

CMU RT-119 faculty and team

- Bill Scherlis, PI
- Claire Le Goues
- Travis Breaux
- Christian Kästner
- David Garlan
- Bradley Schmerl
- Joshua Sunshine
- Jonathan Aldrich
- Multiple PhD students

We are prepared for the risks, the uncertainties, and the dangers of software-intensive systems engineering
RT-119 Systemic Assurance – Focus and Themes

• Assurance with scale, complexity, variability, uncertainty
  – Quality attributes:ilities, security, etc.

• Focus on software-reliant system engineering domains
  – Dynamism and self-adapting systems
  – Complex framework-based and web-based systems
  – CPS

• Themes
  – Evidence and traceability
    – Architecture, resiliency, variabilities, coupling
    – Evolving systems and “agile intent”
    – Technical models for requirements
    – Productivity and affordability – “pushing the curve out”

• Primary technical areas
  – Chains of evidence to support ongoing reevaluation for evolving systems
  – Language extensions for assurance assertions and context metadata
  – Dynamic adaptability and resiliency in architectural design
  – Evidence-based design, development, and decision support building on engineering data

Summary: Provide practices for systemic assurance of safety, reliability, availability, maintainability, evolvability, and adaptability

Impact: Advancement in modeling, analysis, tooling, and process in support of rapid and effective certification of systems in development and recertification of systems in sustainment/modernization.

Validation: Achieve this through a strategy that links technical advancement with validation effort including prototyping, case studies and field trials, development of measures, engagement with assurance stakeholders, and evaluation of baseline standards.

Status: Project initiation in 2014. This effort builds on a long record of engagement on the challenges of assurance at scale for component-based complex systems, including architecture, resiliency, modeling, analysis, tooling, concurrency, and other areas.
RT-119 – the Seven Steps

1. Identify baseline and intervention models for a selection of current standards
2. Advance capability for traceability to support explicit modeling and chains of evidence.
3. Design and implement experiments to address the challenge of rapid recertification.
4. Develop a framework for assessment of architecture-derived quality attributes
5. Develop requirements elicitation and management approaches that better address quality and policy objectives.
6. Augment and collaborate with diverse existing efforts focused on technical means to address particular quality criteria.
7. Identify and advance areas to support increasing automation. The key hypothesis is that assurance-related interventions will increase productivity throughout the lifecycle, leading to a "positive benefit" model.

Year 1 and Task 1

- **Year 1 focus**
  - Tasks 1 (baselining), 4 (arch), 5 (RE), 6 (tech attrs)

- **Note on Task 1 – standards baselining**
  - Identify baseline and intervention models for a selection of current standards
  - Candidate DoD standards and activities for consideration
    - 5000.02 and PPP, 8500.01 and RMF, 800-53, etc.
    - Representative compliance stds: FISMA, ISO 26262, etc.
    - Evaluation standards: NIAP/CC, DO 178C, etc.
    - OT&E, C&A, etc.
    - Selected jointly with OSD stakeholders
  - Commercial approaches and identified best practices
    - Process frameworks: SDL and BSIMM
    - Internal commercial practices
The RT-119 task matrix – initial areas of focus

<table>
<thead>
<tr>
<th>Group 1</th>
<th>Kastner, Aldrich, Sunshine</th>
<th>DDS (OMG's widely adopted data distribution service), CC/NIAP. Ecosystem stds for Android, node.js, Eclipse.</th>
<th>1. Standards baselining and evaluation</th>
<th>2. Traceability and models and evidence</th>
<th>3. Rapid recertification (later: Modules and composition benefits)</th>
<th>4. Architecture derived quality attributes (e.g., BA) Grant capabilities to modules to control resources (e.g., ambient authority).</th>
<th>5. Requirements elicitation and analysis</th>
<th>6. Technical means for quality criteria</th>
<th>7. Automation yielding positive benefit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group 2</td>
<td>Breaux</td>
<td>IA focus with CNSII 1253, DoDI 8500.01, NIST 800-53 and others; Interview AO's</td>
<td>Description and temporal logic for IA policy constraints</td>
<td>Dynamic checking</td>
<td>Assessment of compositions</td>
<td>IA requirements using logic (as specified)</td>
<td>Reasoning within the logics (as specified)</td>
<td>Focus on (composable) interface specifications, looking at existing module systems</td>
<td></td>
</tr>
<tr>
<td>Group 3</td>
<td>Garlan, Schmerl</td>
<td>Review Do 178C (aviation flight controls and avionics), and FDA also; ODAF, OMG coordination.</td>
<td>Models for resilient architectures.</td>
<td>Multiple design models.</td>
<td>Resilience. (HLA case study?)</td>
<td>Modeling and analysis of architectural models. Runtime monitoring and repair</td>
<td>Integration of tools targeted at CPS.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Group 4</td>
<td>LeGoues</td>
<td>(D.O.T.E) criteria.</td>
<td>Evidence in the form of models, analyses, tests</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Group 1 (Christian Kästner, Jonathan Aldrich, Josh Sunshine)

- **Initial stds focus: OMG’s Data Distribution Service (DDS)**
  - Widely adopted in Defense (e.g., Navy systems at Lockheed, Raytheon, GD)

- **Framework studies**
  - Key question
    - How to achieve assurance judgments for "payloads and platforms" systems and their open architectures?
  - Approach
    - Composition of modules in sample ecosystems (Android, Wordpress)
    - Framework enforcement of plug-in constraints
    - Configuration experience and modeling (from engineer experience)

- **Isolation in architecture**
  - Key question
    - What are architectural and engineering techniques to better support significant trust gradients among components in a system
  - Approaches
    - Encapsulation. Monitoring and logging.

- **Architecture-derived quality attributes**
  - Key question
    - How to advance architecture-derivedilities (security, reliability, evolvability)?
  - Approach
    - Modeling based on a new module focused on quality attributes and traceability
**Group 2 (Travis Breaux)**

- **Initial stds focus**
  - DoD 5000.1, 8500.1, 8500.2
  - Relevant STIGs

- **Requirements and specifications**
  - Analysis of requirements for selected technical properties,ilities

- **Stakeholder engagement**
  - Interviews with DIACAP stakeholders regarding C&A

- **Requirements to support artifact-based evaluation**
  - Alignment of identified requirements with other project teams
  - Focus on opportunities for artifact-focused evaluation as complement/alternative to existing process-based approaches

---

**Group 3 (David Garlan, Bradley Schmerl)**

- **Initial stds focus:** DODAF, DO-178C
  - Emphasis on architectural practices

- **SE architectural design-space trades (w/ Kevin Sullivan, U VA)**
  - Architectural model analysis, model checking

- **Modeling and checking of structural and semantic consistency**
  - CPS focus

- **Resilience and self-adaptive systems**
  - Modeling using statistical multiplayer games

- **Multi-disciplinary SE**
  - Interviews with stakeholders to understand interactions among disciplines with focus on assurance issues
Group 4 (Claire LeGoues)

- [Partnering with Group 1]
- [Initiating effort]
  - Role of architectural frameworks to enforce behaviors

No engineer or programmer, no programming tools, are going to help us, or help the software business, to make up for a lousy design.

— NATO Software Engineering report 1969
The aim of any testing scheme is to ensure that the customer gets substantially the software that he ordered and it must provide the customer with convincing evidence that this is so.

— NATO Software Engineering report 1968

**Assurance: Incentives and counter-incentives**

- **Augmenting process compliance with direct evidence**
  - Process *begets* quality
  - Evidence *affirms* quality

- **Challenges and impediments** to evidence-based approaches
  - IP exposure, acceptance evaluation, and aggregate judgments
  - Safe harbors and perverse incentives
  - False trades: performance, cost (both lifecycle and devt), security, etc

- **Drivers of evidence-based approaches**
  - Process
    - Evolution / modernization / sustainment
  - Structures
    - Dynamic architectures, resiliency, autonomy
    - Frameworks, granular components, rich supply chains, ecosystems
    - Internal trust gradients
  - Data-intensive modern tooling, linking models and artifacts
    - Modeling and analytic evidence structures
    - Multiple kinds of models, attributes,ilities, etc.
    - Traceability and links – mixed formal and informal
    - Explicit management of attribute trades