System of Systems Architecture Evaluation with Concurrent Development

SATURN 2007
Mike Gagliardi, Bill Wood

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Problem

Severe integration and operational problems can arise due to inconsistencies in addressing system quality attributes between system and software architecture.

This is further exacerbated in a System of Systems (SoS) context where major system and software elements are developed concurrently and independently by a variety of contractors.

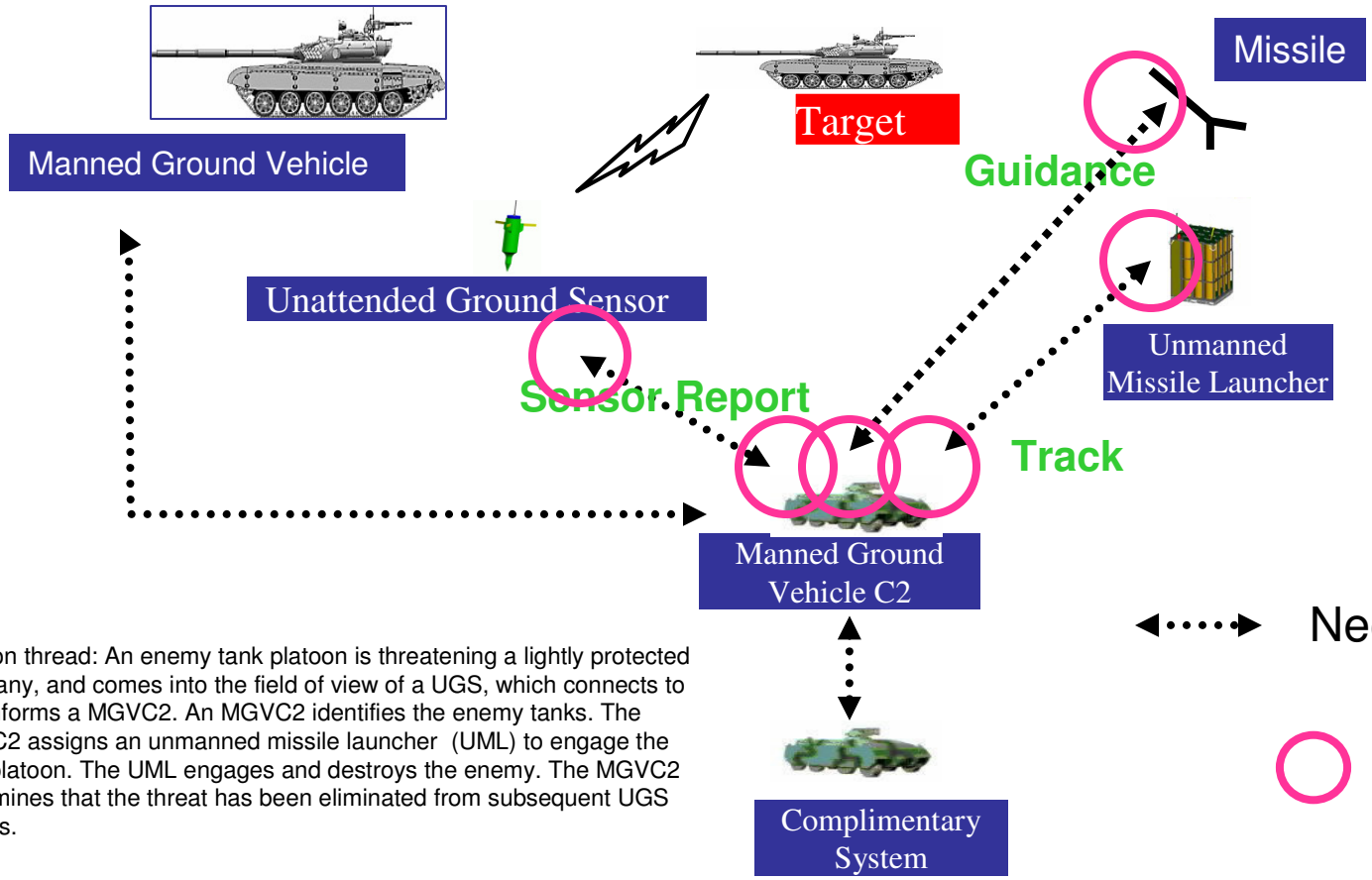
Approach

SEI began a two pronged approach to address the early identification of quality attribute inconsistencies within system and system of systems architectures.

- Develop a methodology to perform a "first pass" identification of inconsistencies across the constituent systems, at the SoS level, using existing mission threads that are augmented with quality attribute concerns.
- Expand the scope of the existing ATAM into system architecture; making minor enhancements to the methodology as needed to address the issue at the system level.

Work with large-scale programs and contractors to develop techniques and participate in SoS and System architecture evaluations.

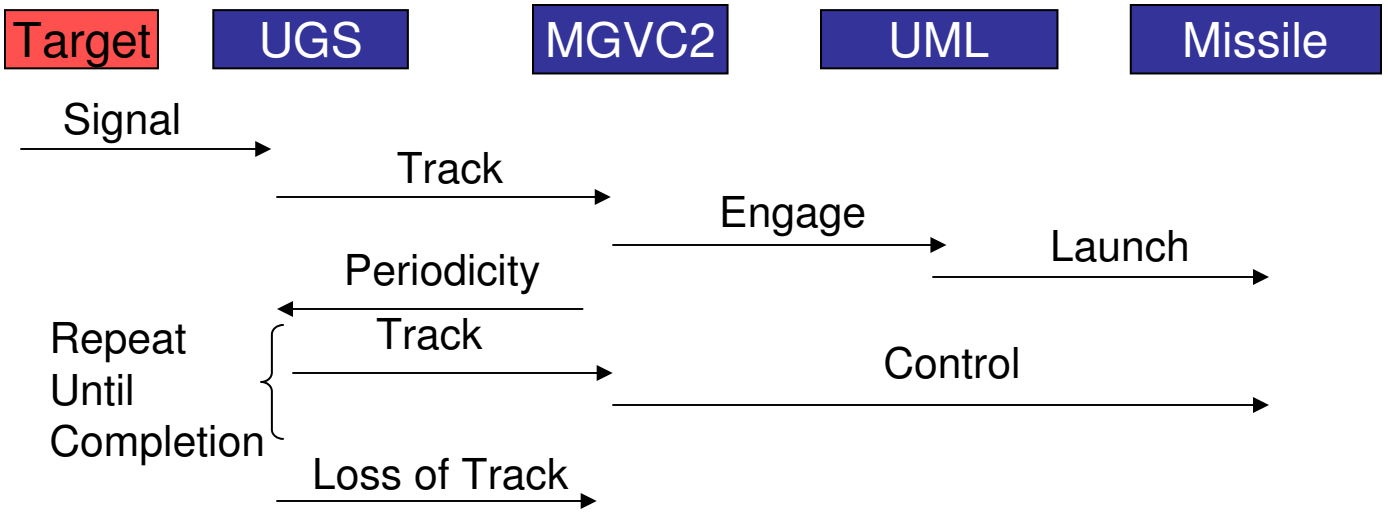
Operational View



Mission thread: An enemy tank platoon is threatening a lightly protected company, and comes into the field of view of a UGS, which connects to and informs a MGVC2. An MGVC2 identifies the enemy tanks. The MGVC2 assigns an unmanned missile launcher (UML) to engage the tank platoon. The UML engages and destroys the enemy. The MGVC2 determines that the threat has been eliminated from subsequent UGS signals.

Specific Mission Thread

An enemy tank platoon is threatening a lightly protected company, and comes into the field of view of a UGS, which connects to and informs a MGVC2. An MGVC2 identifies the enemy tanks. The MGVC2 assigns an unmanned missile launcher (UML) to engage the tank platoon. The UML engages and destroys the enemy. The MGVC2 determines that the threat has been eliminated from subsequent UGS signals.



Quality Attribute Augmentations

The engagement will be started within 10 seconds of sensing the enemy tank platoon

Any one failure does not impact any timelines

- Hardware, software, user, vehicle

This happens in an environment where there are 2000 BSEs being tracked

The security level between the MGVC2 and the UML is top secret

Include some measure of (importance, difficulty). Further refinements are necessary. Mission Thread Workshop (similar to QAW) to be used.

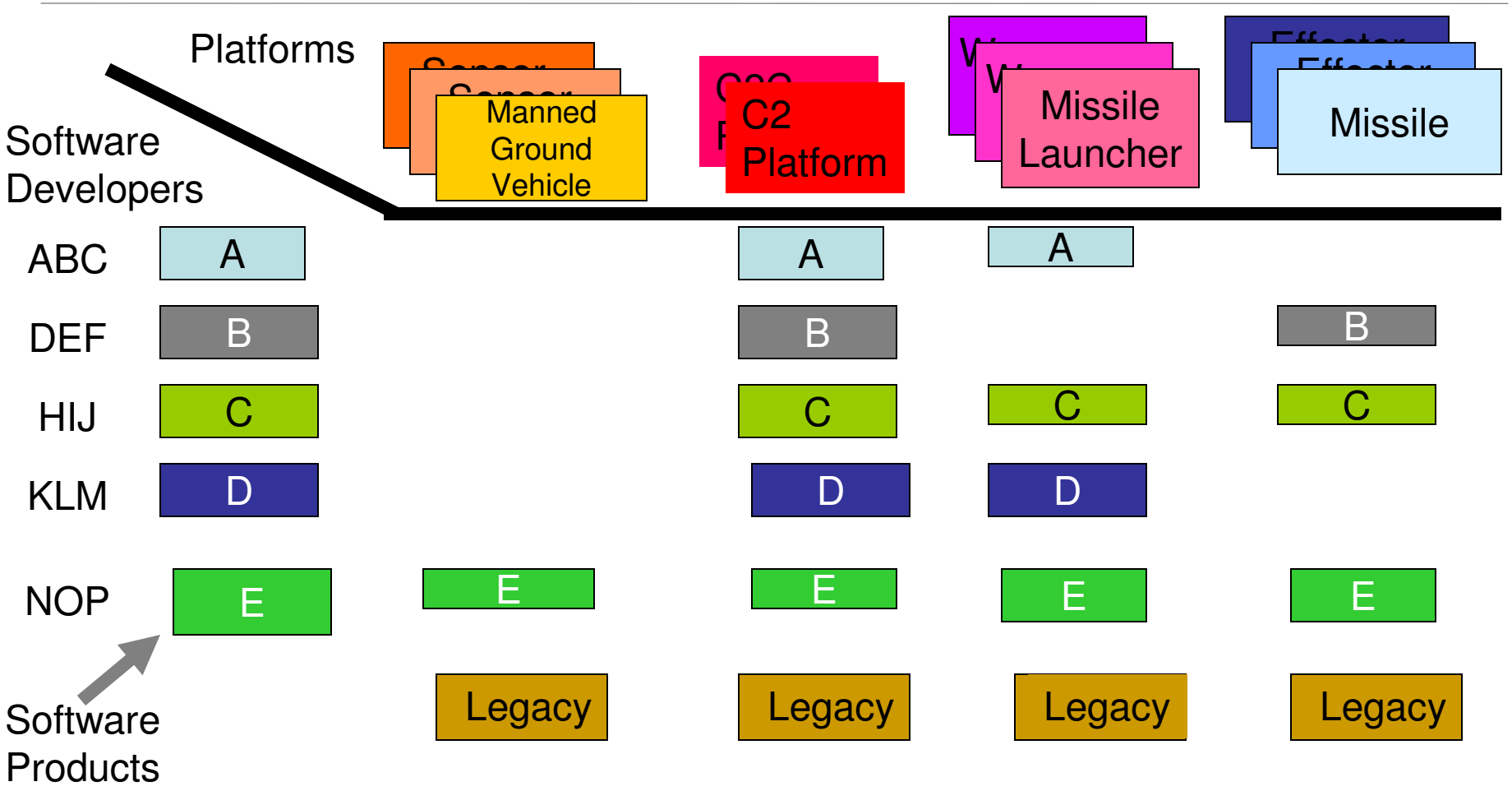
SoS Architecture Evaluation Steps

1) Review the SoS Architecture documentation and guidelines
   - Ensure sufficiency of architecture documentation

2) Mission Thread Selection and Quality Attribute Augmentation
   - Identify subset of system mission threads and augment mission them with quality attributes with stakeholders. Refine the details for the QAs.
   - Identify the systems which act upon the mission thread, provide them with augmented mission threads, have them prepare for evaluation.

3) For each augmented mission thread: (follow general ATAM approach with key stakeholders)
   - SoS Architect walking through architecture using the augmented mission thread
   - Each system provides short presentation of architecture for this thread, explaining how the architecture satisfies the QAs and functionality for the augmented mission thread
   - Evaluation team probes architecture at the areas where the systems interact to identify any gaps, inconsistencies, mismatches, etc.
   - Identify potential target systems for follow-on “focused evaluations” or ATAMs using the specific augmented mission threads
     - Derive scenarios from the augmented mission threads

Example: Multiple Contractors



Concurrent Development

SoS Architecture Guidelines are concurrently developed with some architecture components (views)

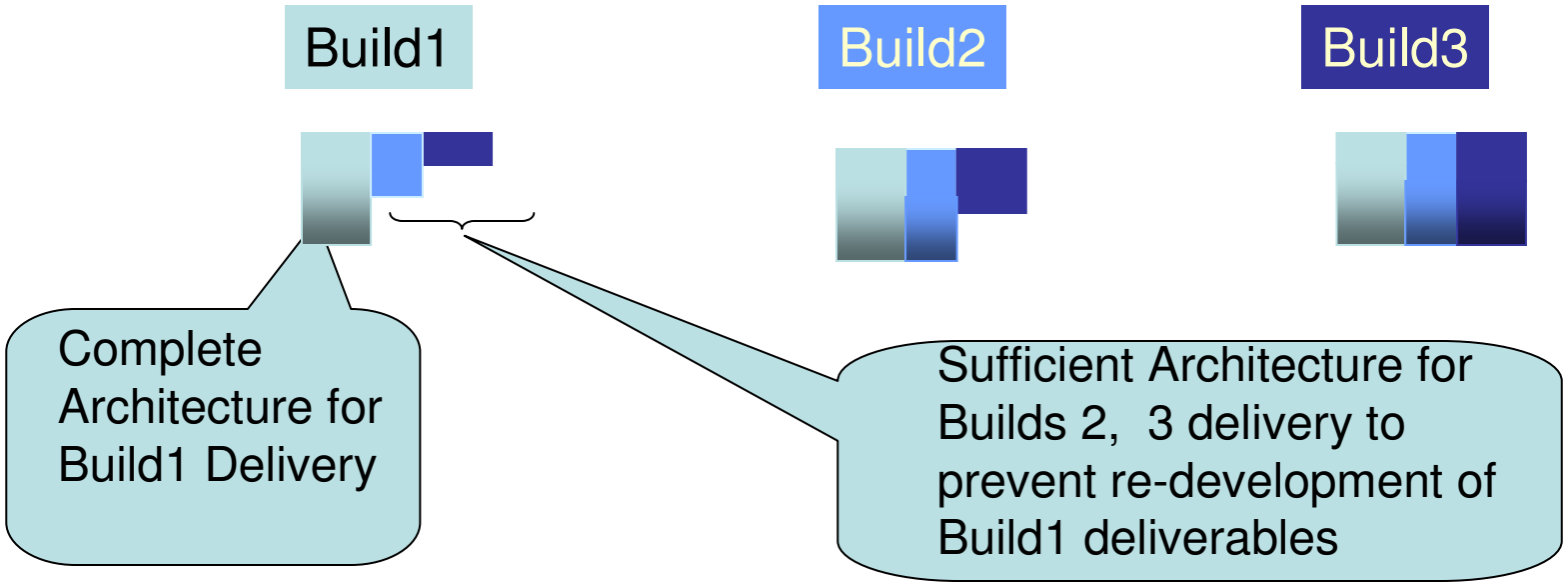
- Some folks “get ahead of the guidelines”
- Some legacy components may not be re-engineered to guidelines

Systems are architected and designed concurrently and independently, often with different views and tools. Must reconcile different architecture products as they become available.

Life-Cycle – Multiple Builds

Evaluation should focus on “Next Build”

- Including enough architecture of future build to minimize extensive re-work between builds (especially safety, security, fault tolerance)



Quality Attribute Augmentations (with builds)

The engagement will be started within 10 seconds of sensing the enemy tank platoon Build 1

Any one failure does not impact any timelines

- Hardware, software, user, vehicle – Build 2

This happens in an environment where there are 2000 BSEs being tracked

(1000 in Build 1)

The security level between the MGVC2 and the UML is top secret - Build 3

Include some measure of (importance, difficulty)

SoS Architecture Evaluation Steps (with builds)

1) Review the SoS Architecture documentation and guidelines

2) Mission Thread Selection and Quality Attribute Augmentation

3) For each augmented mission thread: (follow general ATAM approach with key stakeholders)

- SoS Architect walking through architecture using the augmented mission thread
- *Evaluation focuses on current and next builds*
  - *Including enough architecture of future builds to minimize extensive re-work between builds*
- Each system provides short presentation of architecture for this thread, explaining how the architecture satisfies the QAs and functionality for the augmented mission thread
- Evaluation team probes architecture at the areas where the systems interact to identify any gaps, inconsistencies, mismatches, etc.
- Identify potential target systems for follow-on "focused evaluations" or ATAMs using the specific augmented mission threads

Challenges

Technical

- Data / Information Architecture is critical and needs to be addressed
- Broader set of quality attributes
- SoS Patterns and Tactics
- SOA implications

Evaluation

- Evaluation team needs beefed up - SME's fast-tracked onto the evaluation team
- Each contractor may use diverse DoDAF products, software views and architecture toolsets
- SoS requirements often too vague
- Black box nature of many components
- Architecture/Design documentation does not match implementation

Programmatic

- Multiple Contractors' architectural products (views) for systems/software to be integrated into many weapons/sensor platforms and SILs
- Concurrent Development of system and software architectures – tension between system and software architects
- Life-cycle of many builds with spin-outs to the field from each build

Software Engineering Institute | Carnegie Mellon