



The Open Source AADL Tool Environment (OSATE)

Peter Feiler

Feb 2019

phf@sei.cmu.edu



Software Engineering Institute

Carnegie Mellon University

© 2019 Carnegie Mellon University

[Distribution Statement A] Approved for
public release and unlimited distribution.

Copyright 2019 Carnegie Mellon University. All Rights Reserved.

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

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.

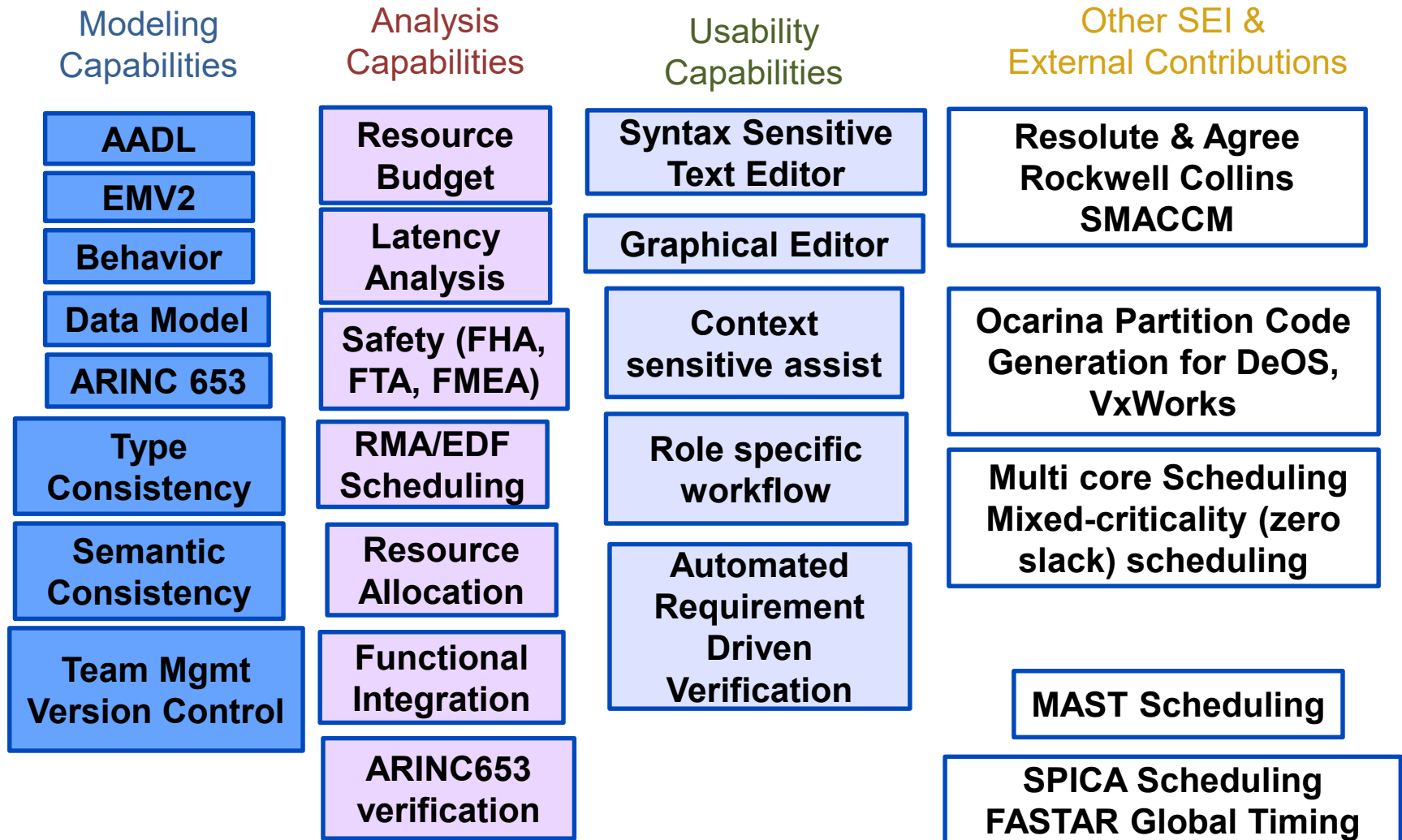
Carnegie Mellon® is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.
DM19-0065

OSATE

Eclipse-based Open Source AADL Tool Environment (OSATE)

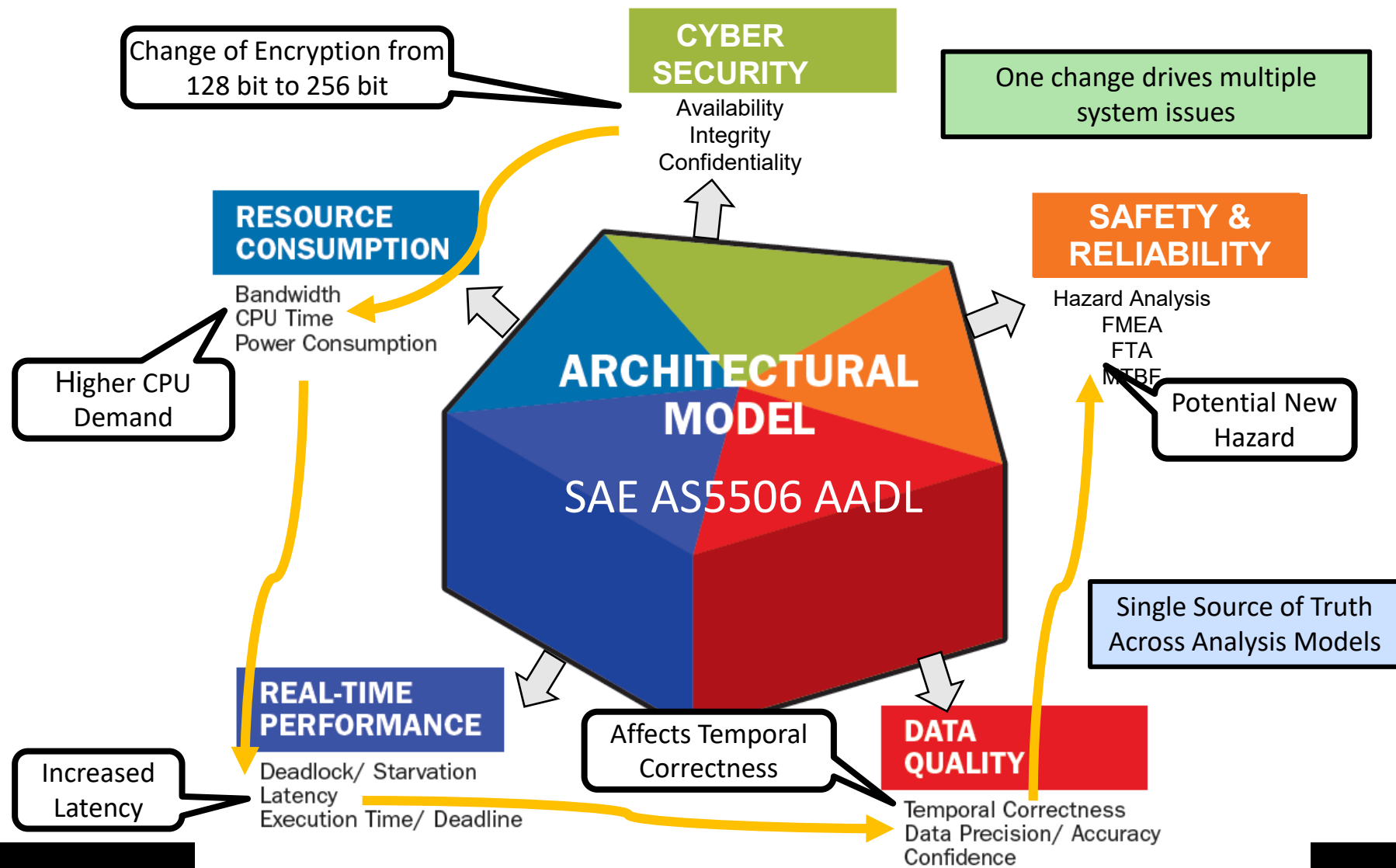
- No cost license under EPL license
- Download site: osate.org
- Issue tracking: <https://github.com/osate/osate2/issues>
- Release cycle: bi-monthly stable, nightly builds
- Forum for questions: <https://groups.google.com/forum/#!forum/osate>
- Reference implementation of core AADL and annex standards
 - AADL core language V2.2 plus any approved errata
 - Error Model V2 Annex, ARINC653 Annex, Behavior Annex, Data Modeling Annex, Code Generation Annex
- Research prototyping platform
 - University and industrial research and pilot projects at international level

OSATE Workbench Capabilities



Analysis of System Properties via Architecture Model

A Contribution to Single Source of Truth



Demonstration of Three OSATE Capabilities

Flow latency analysis early and throughout development life cycle

Safety and reliability analysis at different system levels

Architecture-Led Incremental System Assurance (ALISA) through requirements driven continuous integration



Flow Latency Analysis

Early and throughout development life cycle

Worst-case latency and latency jitter

Functional architecture with mapping to physical architecture

Impact of application task & communication and embedded platform design decisions

Latency Contributors

Specified component latency across multiple tiers

Specified connection, protocol, network latency

Sampling latency & port queuing

Network & protocol sampling, fixed & data size based

Partition allocation, rates, schedules

Synchronous and asynchronous platform components

Explore design alternatives

Synchronous vs asynchronous system

Partition end vs. major frame delayed send

Deadline vs. max execution time

Full vs. empty queue

Alternative deployment mappings

Alternative platforms

Alternative application configurations

Uses deadline rather than completion time

FASTAR includes scheduling analysis results

Sensitivity to completion time variation



Safety Analyses

Early and throughout development life cycle

Analyze impact of exceptional conditions at all levels of design

Based on Error Model V2 (EMV2) annotations in AADL models

Functional & physical architectures, application and platform systems including deployment mappings

Capabilities

Functional hazard assessment (FHA) reports in three formats

Reliability block diagram (RBD) like parts based system reliability

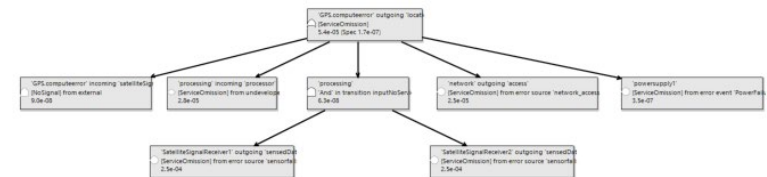
- Based on composite error state specifications

Fault impact analysis (FMEA) : forward chaining of type specific error sources

Probabilistic fault tree analysis (FTA): backward chaining to identify type specific contributors

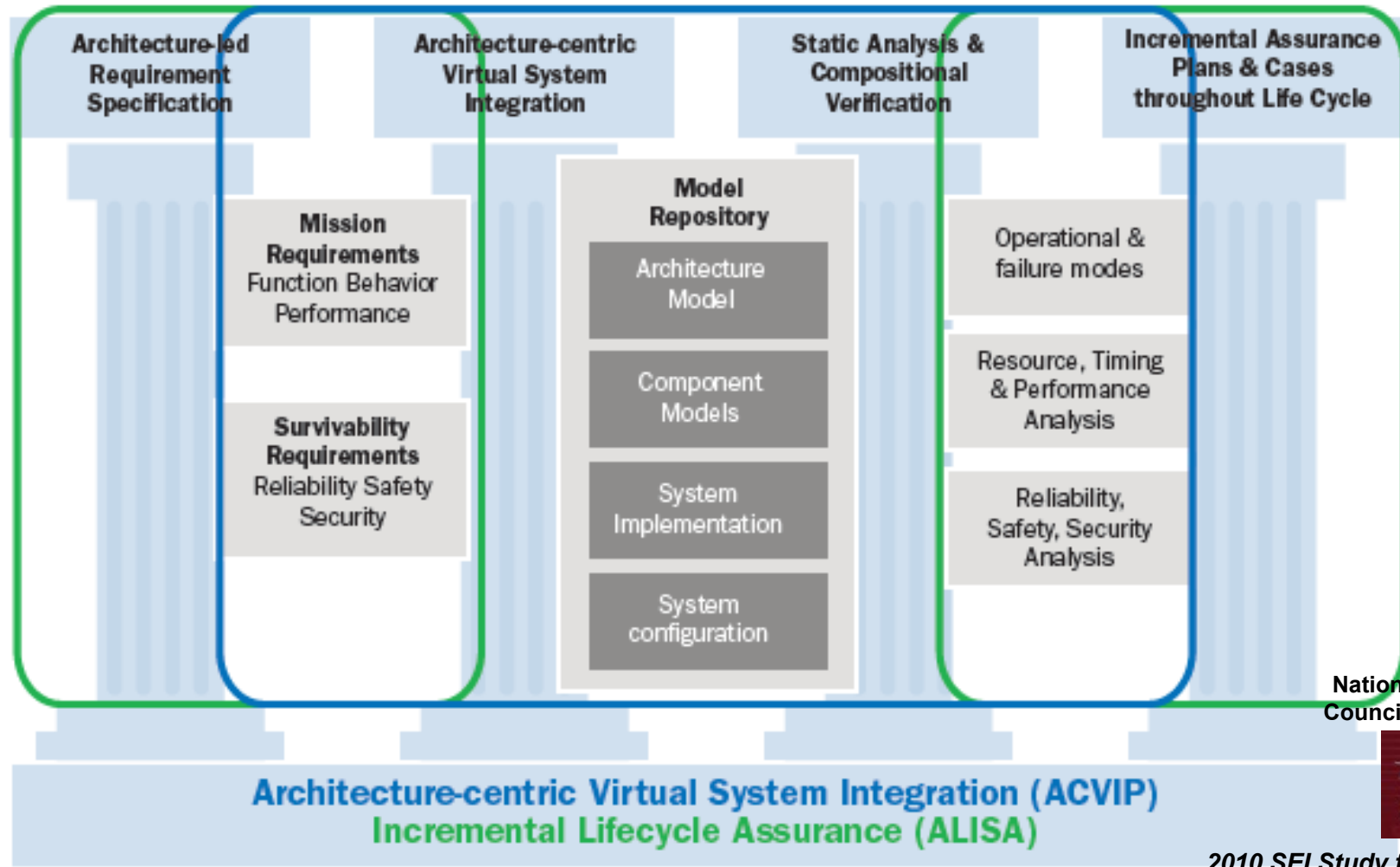
- Based on error flows and propagation paths
- Based on error events and specified component error behavior
- Fault trace, fault tree, and minimal cut sets
- Graphical and tabular results

Error Model Element/Type	Computed Probability	Specified Probability
'GPS.computeerror' outgoing 'location' (ServiceOmission)	5.4e-05	1.7e-07
'GPS.computeerror' incoming 'satelliteSignal' (NoSignal) from external		9.0e-08
'processing' incoming 'processor' (ServiceOmission) from undeveloped		2.8e-05
'And' in transition inputNoService (ServiceOmission) from error source 'sensorfailure'	6.3e-08	2.5e-04
'SatelliteSignalReceiver1' outgoing 'sensedData' (ServiceOmission) from error source 'sensorfailure'		2.5e-04
'SatelliteSignalReceiver2' outgoing 'sensedData' (ServiceOmission) from error source 'sensorfailure'		2.5e-05
'network' outgoing 'access' (ServiceOmission) from error source 'network_access_es'		2.5e-05
'powersupply1' (ServiceOmission) from error event 'PowerFailure'		3.5e-07



Automated Verification of Functional and Non-functional System Properties

Assurance: Sufficient evidence that a system implementation meets system requirements



National Research
Council Study 2007

2010 SEI Study for AMRDEC

Automated Incremental Assurance with ALISA

Identify Assurance Hotspots throughout Lifecycle

Stakeholder Goals

Verifiable Requirements

High Abstraction

Abstraction Level

Low Level
Close to Implementation

Tier 0



Model

Ver Plan

Req

Tier 1



Model+1

Ver Plan

Req+1

Code

Tier 2



Code

Model+2

Model+2'

Ver Plan

Req+2

Continuous Assurance
Case Record



Incremental Multi-Tier Assurance of an Aircraft

Aircraft: (Tier 0)

Aircraft system: (Tier 1)
Engine, Landing Gear, Cockpit, ...
Weight, Electrical, Fuel, Hydraulics,...

LRU/IMA System: (Tier 2)
Hardware platform, software partitions
Power, MIPS, RAM capacity & budgets
End-to-end flow latency

System & SW Engineering:
Mechatronics: Actuator & Wings
Safety Analysis (FHA, FMEA)
Reliability Analysis (MTTF)

Subcontracted software subsystem: (Tier 3)
Tasks, periods, execution time
Software allocation, schedulability
Generated executables

OEM & Subcontractor:
Subsystem proposal validation
Functional integration consistency
Data bus protocol mappings

Repeated Virtual Integration Analyses:
Power/weight
MIPS/RAM, Scheduling
End-to-end latency
Network bandwidth

Proof of Concept Demonstration and Transition by Aerospace industry initiative

- Architecture-centric model-based software and system engineering
- Architecture-centric model-based acquisition and development process
- Multi notation, multi team model repository & standardized model interchange

- Multi-tier system & software architecture (in AADL)
- Incremental end-to-end verification of system properties

