



Calhoun: The NPS Institutional Archive
DSpace Repository

Faculty and Researchers

Faculty and Researchers' Publications

2018-11-08

RT-176: Verification and Validation (V&V) of System Behavior Specifications

Giammarco, Kristin

Systems Engineering Research Center (SERC)

Giammarco, Kristin, "Verification and Validation (V&V) of System Behavior Specifications." Presentation to the 10th Annual SERC Sponsor Research Review (2018).
<http://hdl.handle.net/10945/70131>

This publication is a work of the U.S. Government as defined in Title 17, United States Code, Section 101. Copyright protection is not available for this work in the United States.

Downloaded from NPS Archive: Calhoun



Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

<http://www.nps.edu/library>

RT-176: Verification and Validation (V&V) of System Behavior Specifications

Sponsor: DASD(SE)

By

Dr. Kristin Giammarco

10th Annual SERC Sponsor Research Review

November 8, 2018

FHI 360 CONFERENCE CENTER

1825 Connecticut Avenue NW, 8th Floor

Washington, DC 20009

www.sercuarc.org

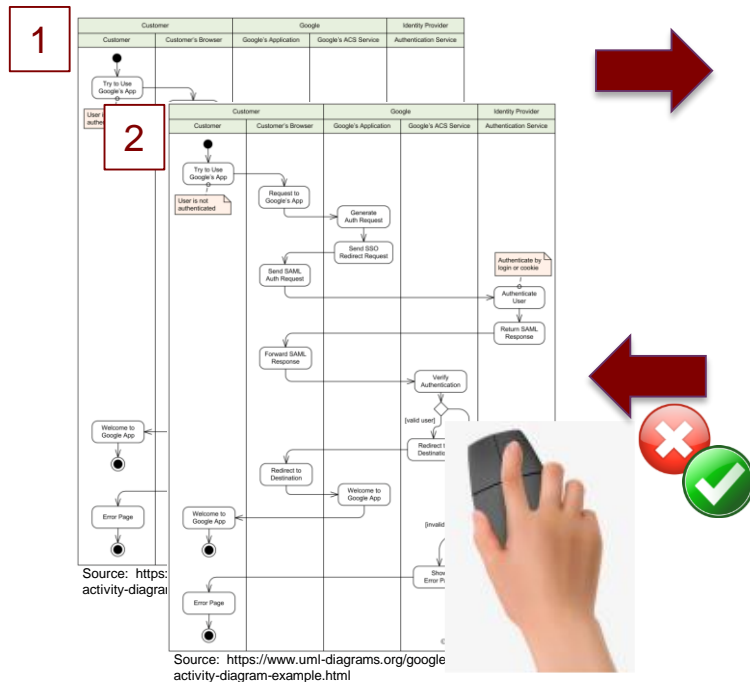


- Research Motivation & Objectives
- Technical Accomplishments
- Future Work

- Model-based methods tools and approaches on their own do not guarantee success
- The model may adhere to notational specifications while the design itself may be incomplete, ambiguous, inefficient, or contain unwanted system behaviors
- This research developed methods and tools to steer and shape *behavioral design*
 - to meet requirements (verification)
 - to meet expectations (validation)

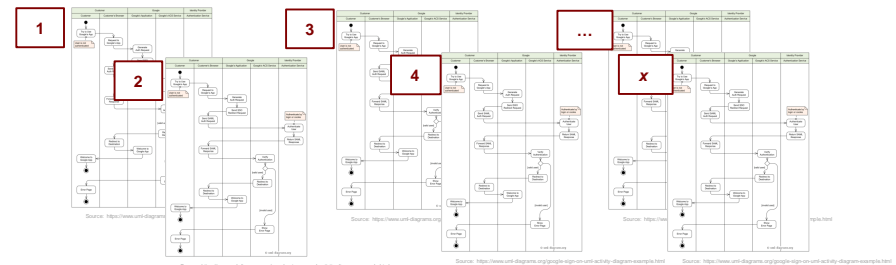
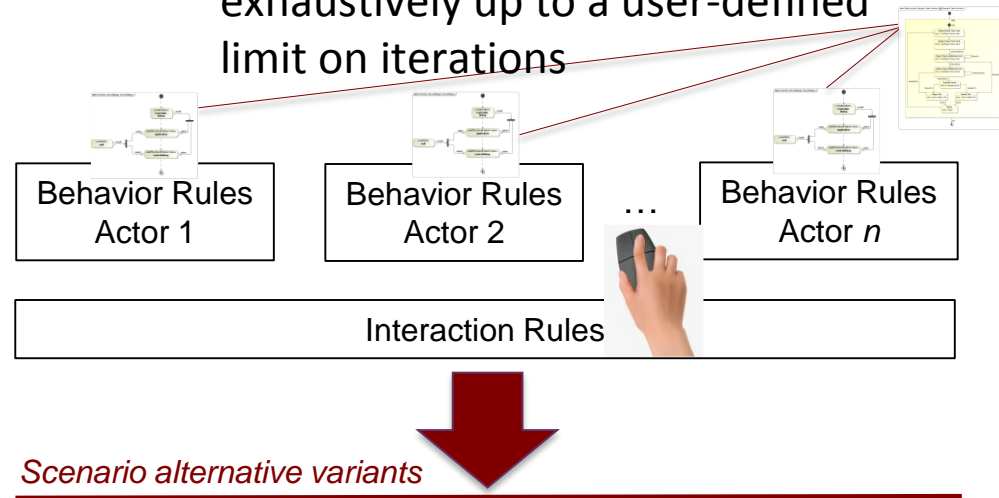
Prevailing Problem:

- Incompleteness
 - Only a subset of possible behaviors are included with actors and interactions drawn on the same diagram



MP Value Proposition:

- Scope-completeness
 - Generates full set of possible event traces (use case extensions) exhaustively up to a user-defined limit on iterations

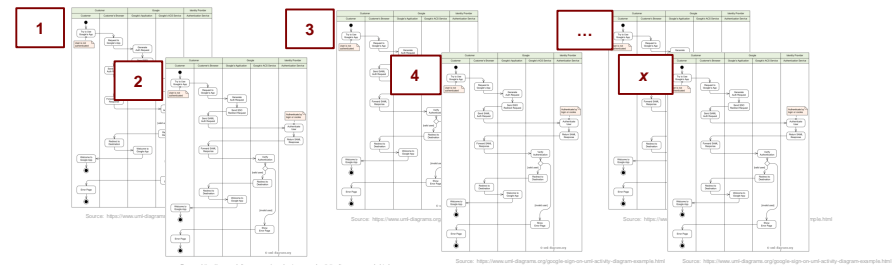
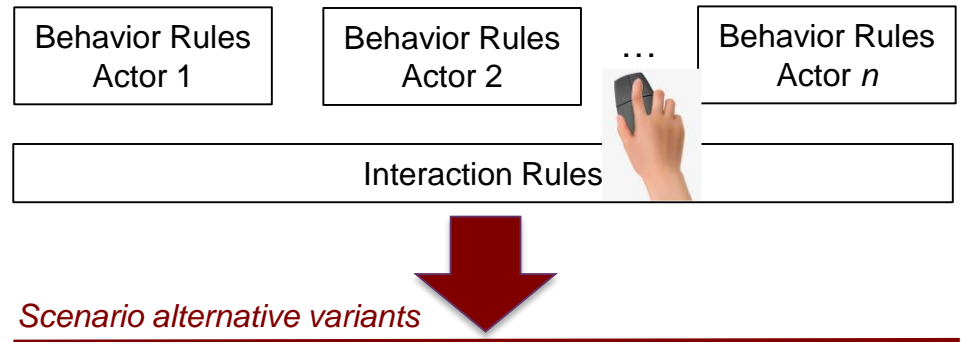
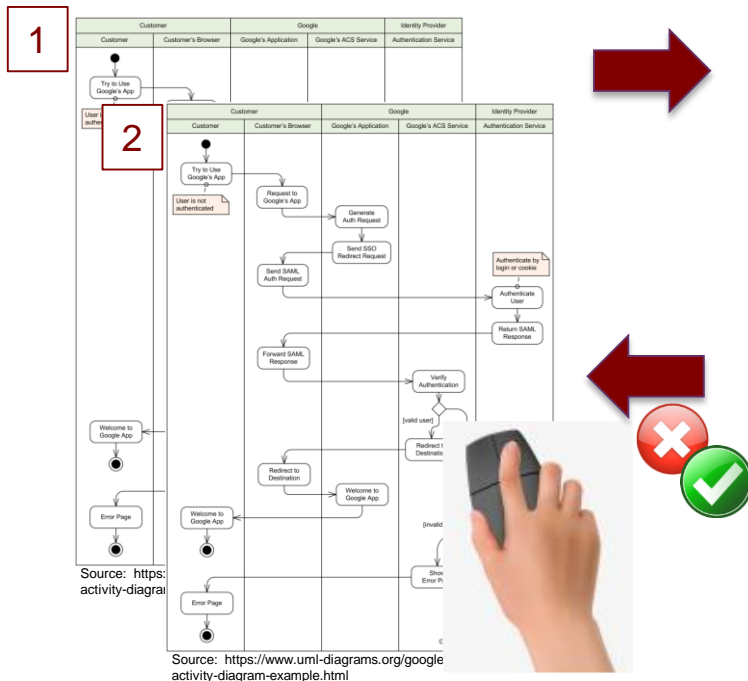


Prevailing Problem:

- Ambiguity
 - Behavior models that describe general activities but are unclear about who is doing each activity, or are otherwise unclear about activities performed

MP Value Proposition:

- Separation of concerns
 - Behaviors are separated by actor, and interactions between actors are separately layered on as constraints
 - Modeling in MP enables discussion and clarification of the behavior logic

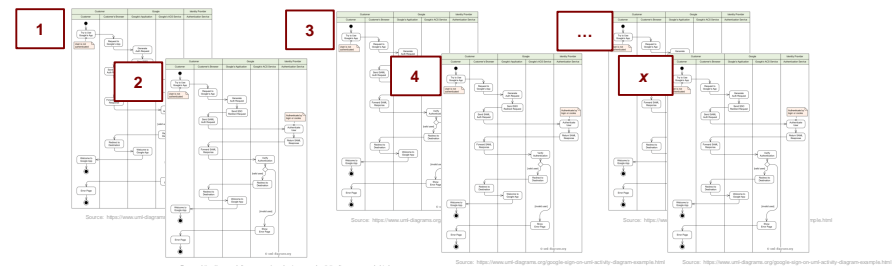
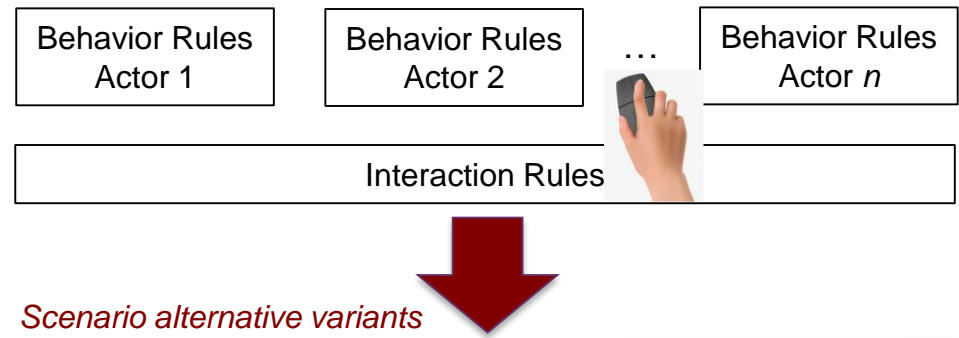
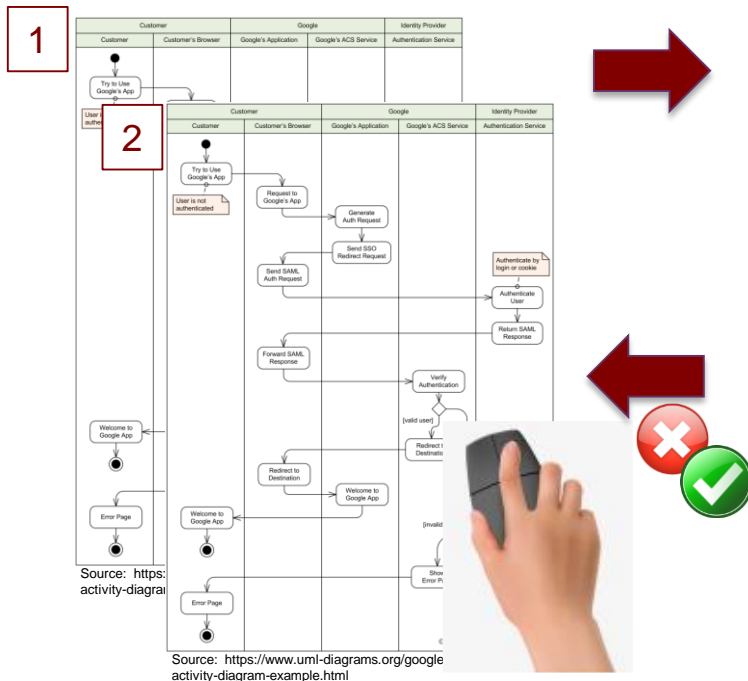


Prevailing Problem:

- Inefficiency
 - When people continue to do work that an automated computing device could do faster and with fewer errors

MP Value Proposition:

- Efficient task allocation
 - Humans focus on using their experience, creativity, and pattern detection skills to inspect and evaluate, and use automated tools to compute, generate, and search

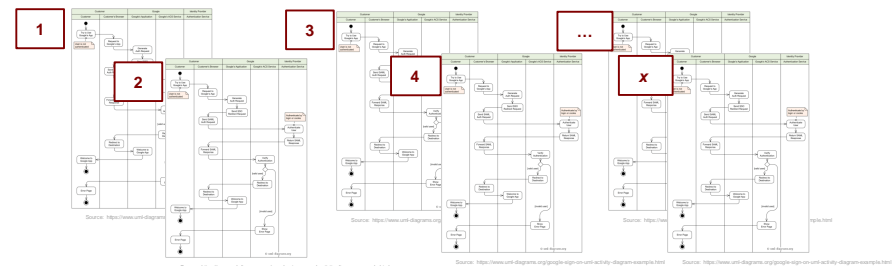
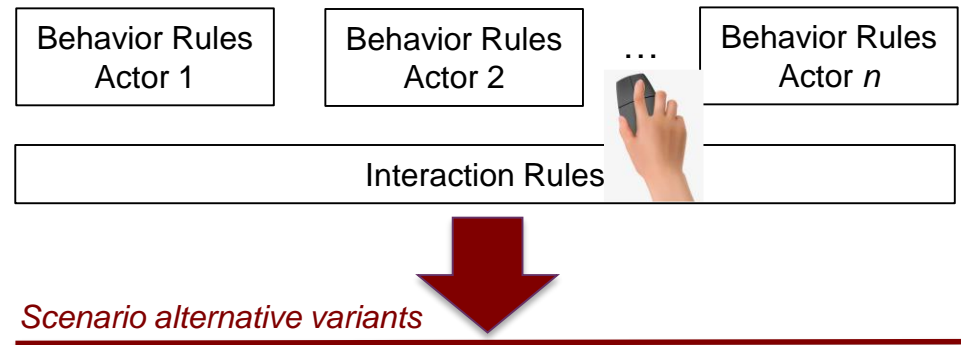
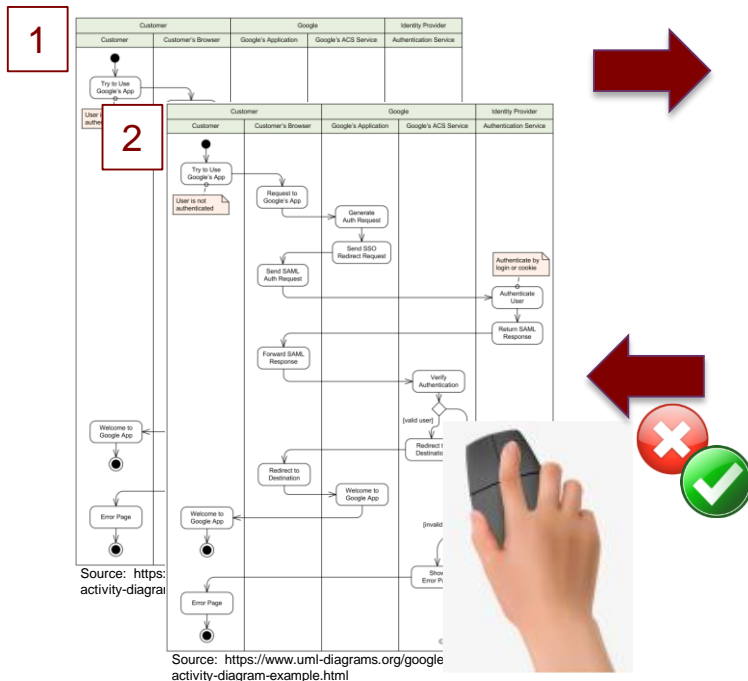


Prevailing Problem:

- Unwanted behaviors
 - Built systems that may meet requirements, but also permit extra undesired behaviors

MP Value Proposition:

- Behavior pruning
 - Enforces the necessary model structure for exposing and purging unwanted behaviors in the design before they emerge in the actual system



- *Demonstrate use of the UAV behavior models for early V&V analysis of requirements*
 - using MP to expose positive and negative system behaviors permitted by the design
- *Formalize patterns of common design flaws or other model properties*
 - Catalog of anti-patterns catalog

- Research Motivation & Objectives
- Technical Accomplishments
- Future Work

Contract No. HQ0034-13-D-0004

UNCLASSIFIED



Verification and Validation (V&V) of System Behavior Specifications

Final Technical Report
October 31, 2018

Principal Investigator:

Dr. Kristin Giammarco, Naval Postgraduate School

Co-Principal Investigators:

Ron Carlson, Naval Postgraduate School
Dr. Mark Blackburn, Stevens Institute of Technology

Research Team:

Dr. Mikhail Auguston, Naval Postgraduate School
Dr. Rama Gehris, Naval Postgraduate School
Marianna Jones, Naval Postgraduate School
Bruce Allen, Naval Postgraduate School
David Shifflett, Naval Postgraduate School
John Quartuccio, PhD Student, Naval Postgraduate School
Kathleen Giles, PhD Student, Naval Postgraduate School
Chris Wolfgeher, PhD Student, Naval Postgraduate School
Gary Parker, PhD Student, Naval Postgraduate School

Sponsor: Naval Air Systems Command (NAVAIR)



Castle Point on Hudson, Hoboken, NJ 07030

A013: Final Technical Report

October 2018

Task Order 0076, RT 176

Appendix A: List of Publications and Invited Talks

Appendix B: References Cited

Appendix C: Collaborator Courses that Integrate or Contribute Research Results

Appendix D: Monterey Phoenix Overview

Appendix F: Instructions for Downloading MP Models

Appendix G: Model Based V&V (MCSE MPT) Demonstration

Appendix E: Catalog of Reusable Architecture Patterns

<https://sercuarc.org/project/?id=35&project=Verification+and+Validation+%28V%26V%29+of+System+Behavior+Specifications>

Run button

https://firebird.nps.edu

Scope of execution

Trace window

Number of traces

Console window

Code window

All traces

```

1 /*
2 Example1_simple_message_flow.mp
3
4 Event grammar rules for each root define derivations for event traces,
5 in this case a simple sequence of zero or more events for each root.
6
7 The COORDINATE composition takes two root traces and produces
8 a modified event trace, merging behaviors of Sender and Receiver
9 and adding the PRECEDES relation for the selected send/receive pairs.
10 The coordination operation behaves as a "cross-cutting" derivation rule.
11
12 Run for scopes 1 and up. The "Sequence" or "Swim Lanes" layouts are
13 the most appropriate for browsing traces here.
14
15 */
16 SCHEMA simple_message_flow
17
18 ROOT Sender: (* send *);
19 ROOT Receiver: (* receive *);
20
21 COORDINATE $x: send FROM Sender,
22             $y: receive FROM Receiver
23             DO ADD $x PRECEDES $y; OD;
24
25

```

Monterey Phoenix Version 3.5

Zoom + Layout Sequence Show Hidden 4 of 4

1 p=0.25

2 p=0.25

3 p=0.25

4 p=0.25

Sender Receiver

send receive

Sender Receiver

send receive

Sender Receiver

send receive

Sender Receiver

send receive

Console

Generated 4 event traces

completed Sender: 4 traces (0 MARKed) 10 events
average 2.5 ev/trace min 1 max 4

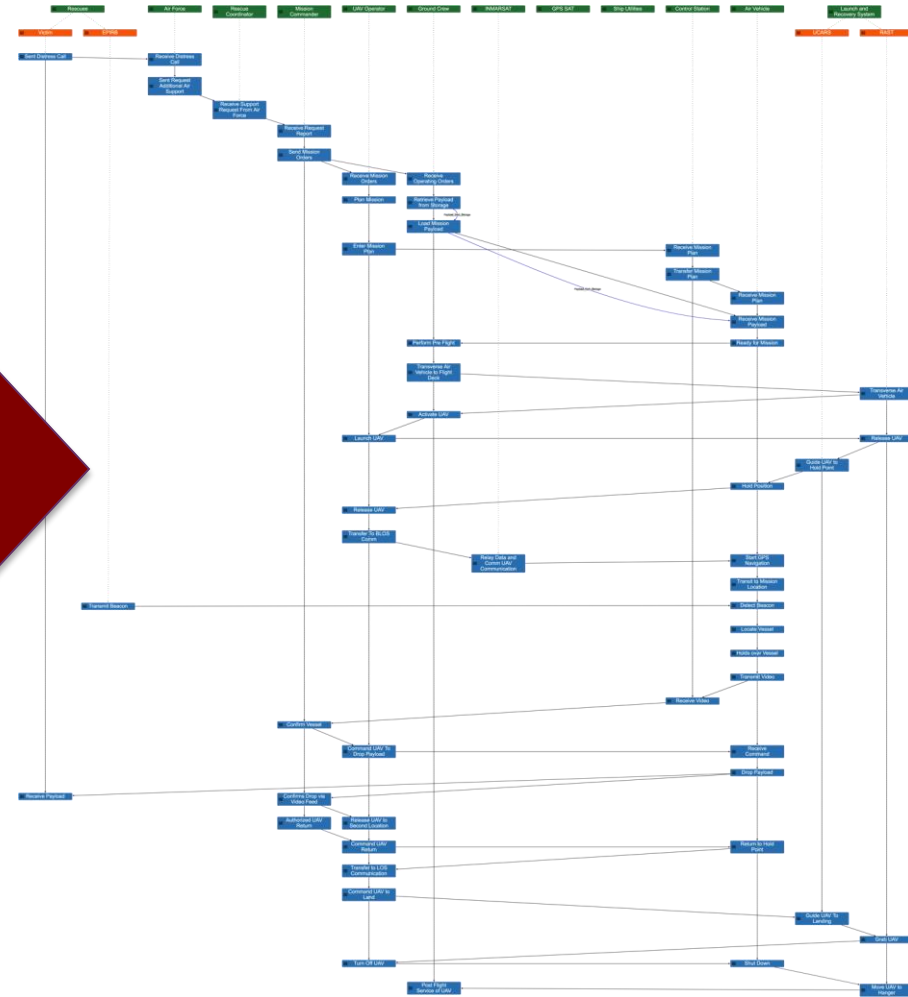
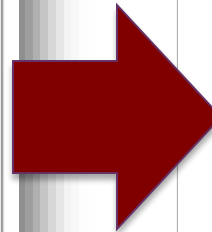
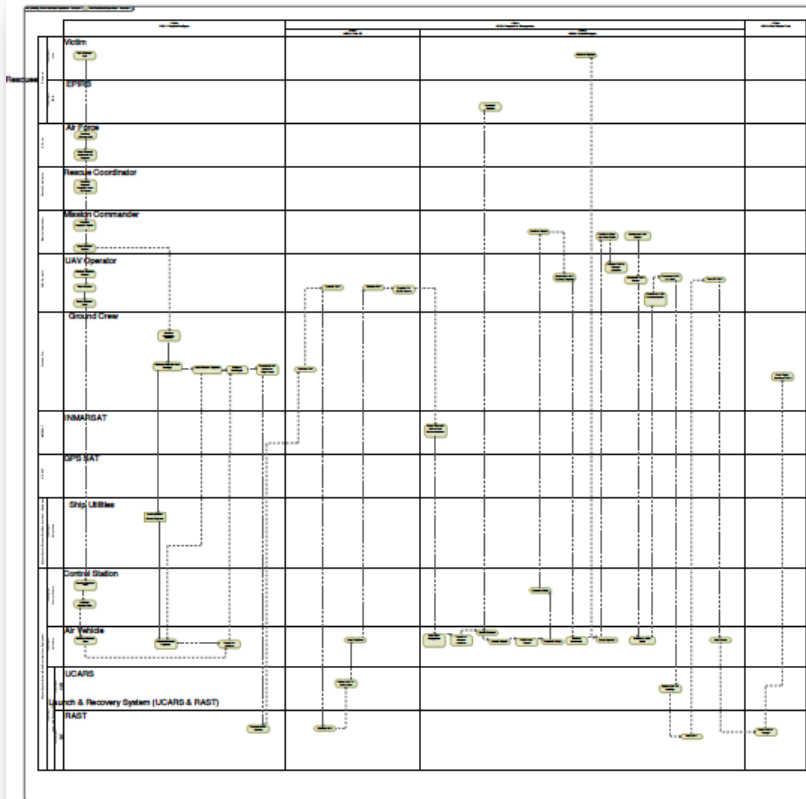
completed Receiver: 4 traces (0 MARKed) 10 events
average 2.5 ev/trace min 1 max 4

completed S_c365fb75081d629ba2078f58abce5bff: 4 traces (0 MARKed) 24 events
average 6 ev/trace min 3 max 9

Elapsed time 0 sec, Speed: inf events/sec

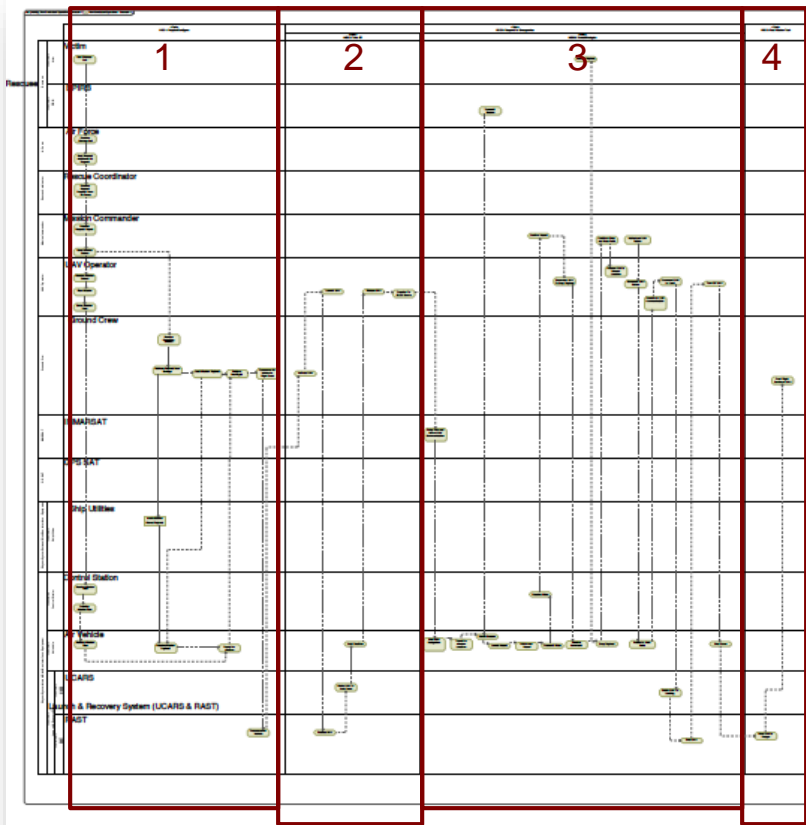
Finished Compiling! Graphing 4 event traces...

Non-Combat Operations Scenario 1

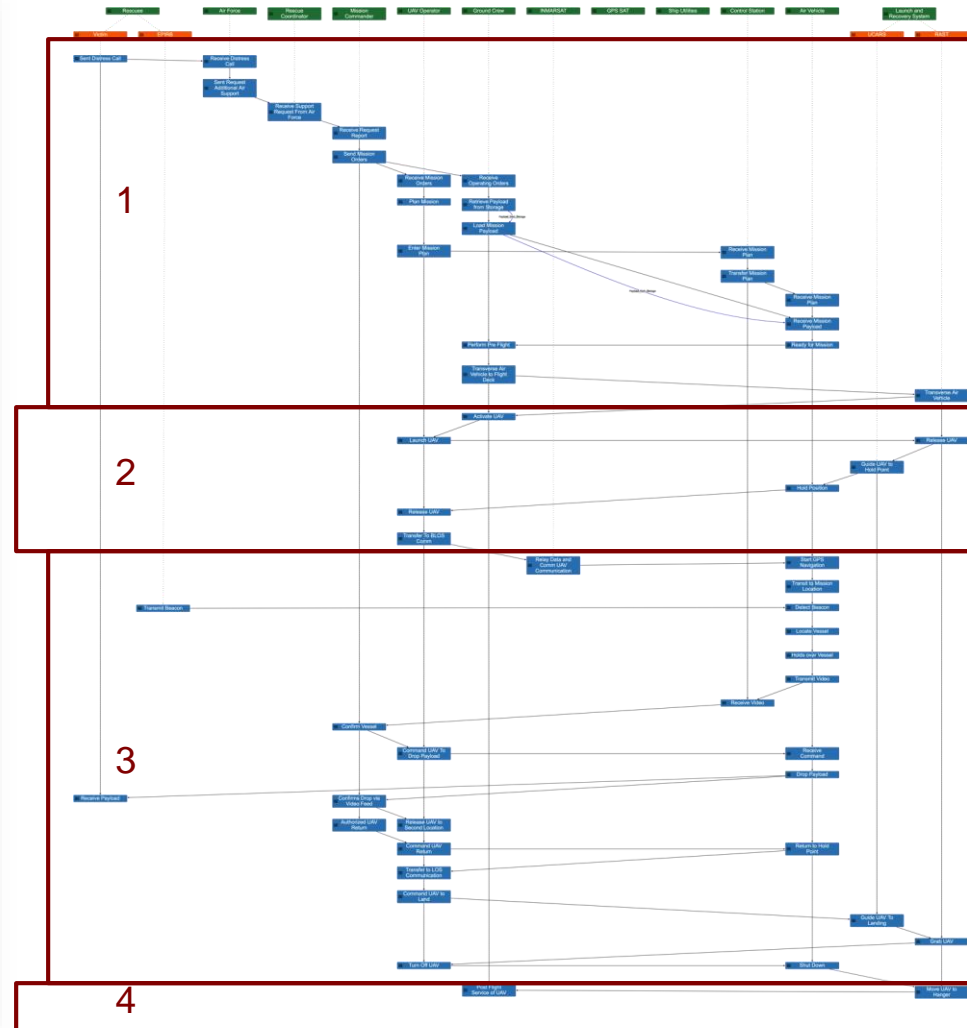


Cameo Systems Modeler, 1-1 Skyzer_nkf1_IM20_etc Non-Combatant Operations - Scenario 1 Aug 6, 2018 9:22:33 AM

Non-Combat Operations Scenario 1

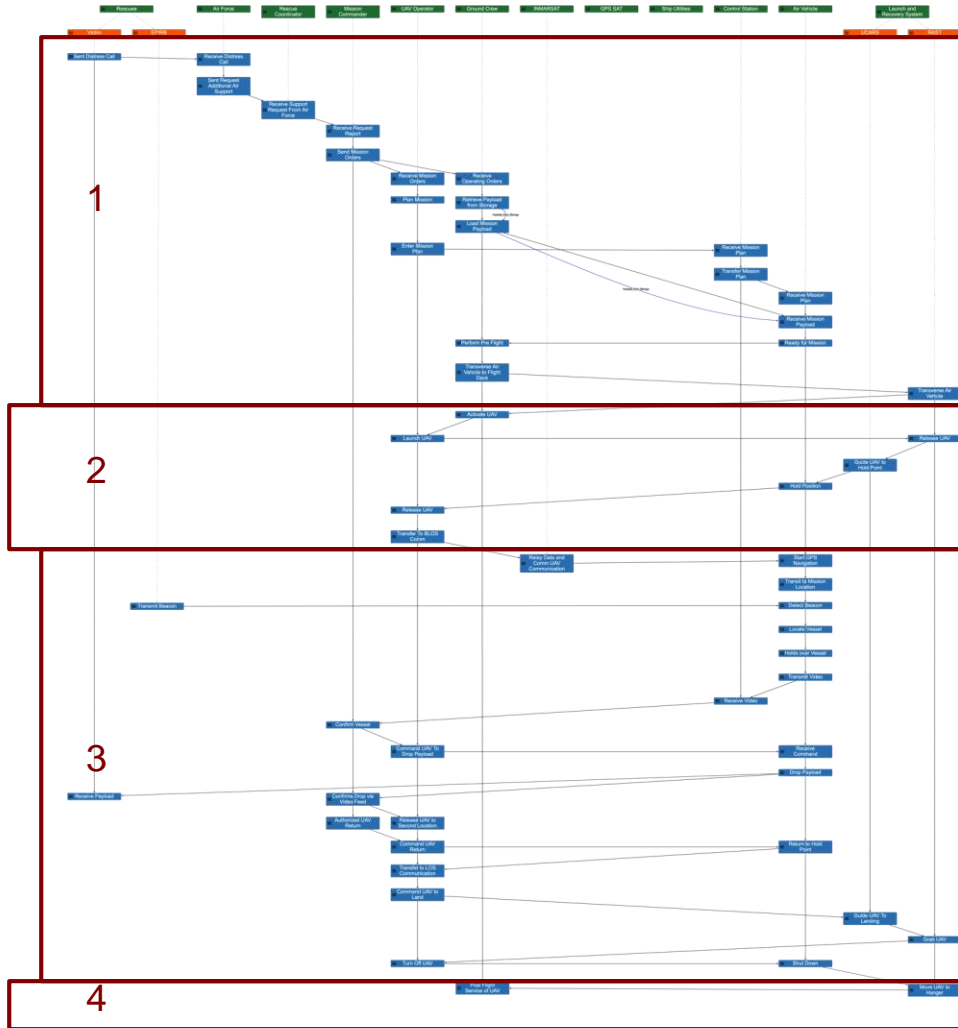


Cameo Systems Modeler, 1-1 Skyzer_nkf1_IM20_etc Non-Combatant Operations - Scenario 1 Aug 6, 2018 9:22:33 AM



1 scenario

Non-Combat Operations Scenario 1



Phase 1 –
Prepare/C
onfigure

Phase 1 scenarios

Phase 2 –
Take Off

Phase 2 scenarios

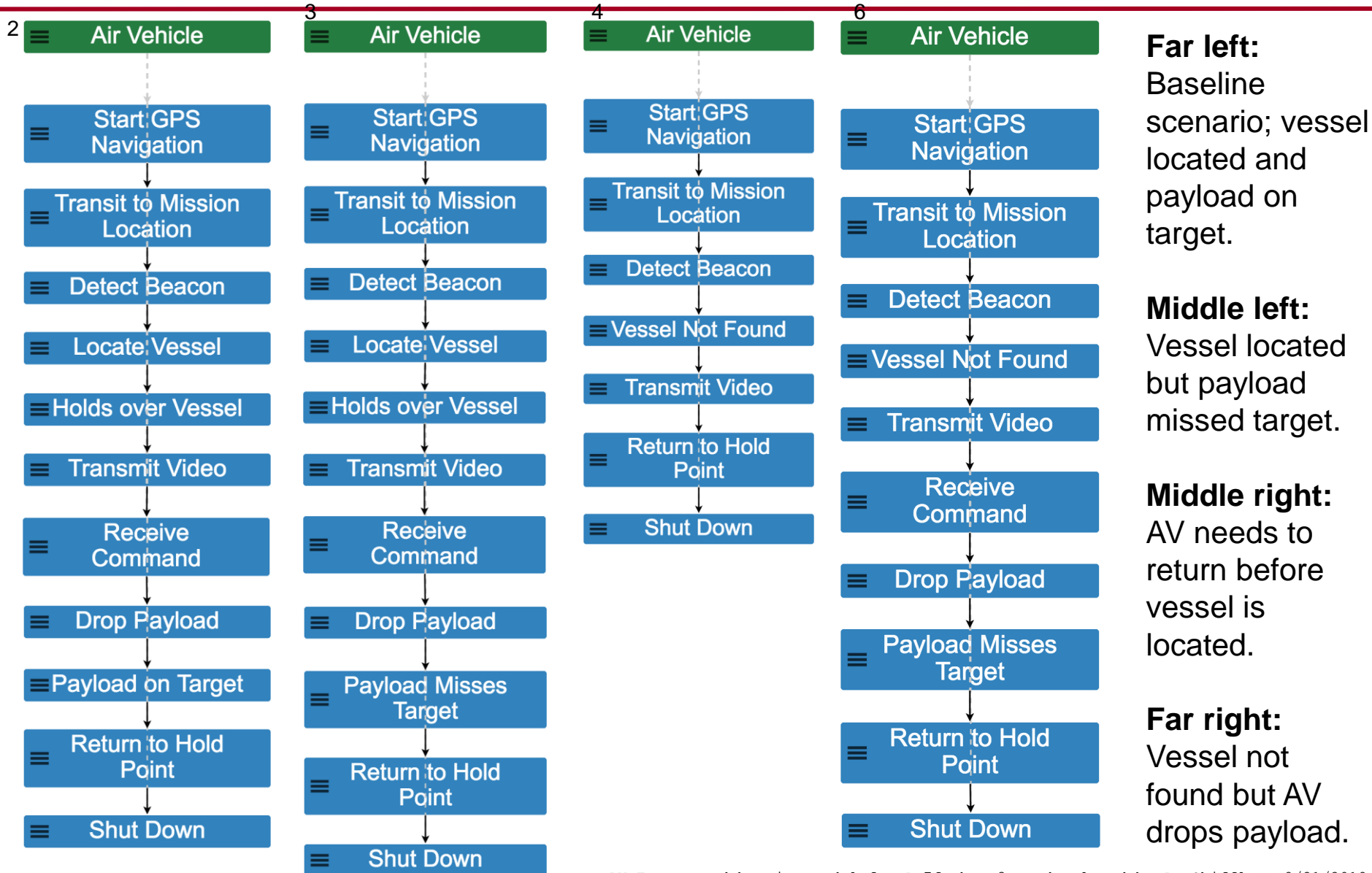
Phase 3 –
Transit/N
avigate

Phase 3 scenarios

Phase 4 –
Post Mission
Task

Phase 4 scenarios

Phase 3 Alternative Emergent Behaviors



AV_Temp.mp, debugging model for Av7f_phase3.mp developed by D. Shifflett 8/21/2018

- What should happen if the payload just misses the target (trace 3)?
 - Could the payload still be retrieved by target vessel? What would help?
- What should happen if the AV has to return before locating/reaching the vessel (trace 4)?
 - Could the payload be dropped at max range with a means for vessel retrieval?
- What should happen if the AV drops the payload prematurely, enroute to the vessel (trace 6)?
 - Though unintended by the modeler, does trace 6 contain an idea for handling out of range vessels or AVs experiencing a return to base condition?

All of these operational “what ifs” were exposed through MP modeling of the provided baseline scenario.

MP modeling of SysML behavior diagrams can help to expose requirements that may otherwise not be considered until later in the lifecycle.

Architecture Model Anti-Patterns (Examples in Four Languages)

No.	DM2/UJPD	UPIA	SDL	LML
H.1.1	Activities with no child and no parent	Operational tasks with no child and no parent	Functions with no child and no parent	Actions with no child and no parent
H.2.4	Requirements with more than one parent	Requirements with more than one parent	Requirements with more than one parent	Requirements with more than one parent
H.5.1	Performers having itself as a child	Capability roles having itself as a child	Components having itself as a child	Assets having itself as a child
FPA.1.1	Activities that are not performed by any performer	Operational tasks that are not performed by any capability role	Functions that are not performed by any component	Actions that are not performed by any asset
FI.3.1	Activities that do not produce or consume any resources	Operational tasks that do not produce or consume any information elements	Functions that do not produce or consume any items	Actions that do not generate or receive any input/outputs
PI.6.1	Performers that exchange some resource, but are not connected to any common connectors	Capability roles that exchange some information element, but are not connected to any common headlines	Components that exchange some item, but are not connected to any common links	Assets that exchange some input/output, but are not connected by any common conduits
T.2.1	Activities that do not trace to any requirement	Operational tasks that do not trace to any requirement	Functions that are not based on any requirement	Actions that do not satisfy/verify/trace to any requirement
S.5.1	Performers that interact with each other through exchange of resources, but are not subject to a common standard	Capability roles that interact with each other through exchange of information elements, but are not subject to a common standard	Components that interact with each other through exchange of items, but are not specified by a common standard-labeled requirement	Assets that interact with each other through exchange of input/outputs, but satisfy no common standardizing requirement

Technical report contains a total of 46 anti-patterns

- Research Motivation & Objectives
- Technical Accomplishments
- Future Work

- Further test the Monterey Phoenix approach on MBSE pilot projects
- Formalize the types and definitions of emergent behavior for use in risk analysis
- Train model developers how to verify and validate SysML models from other tools using MP
- Generate SysML sequence, activity, and state transition views from MP models
- Develop a graphical gateway to MP (enable code generation from diagrams)

