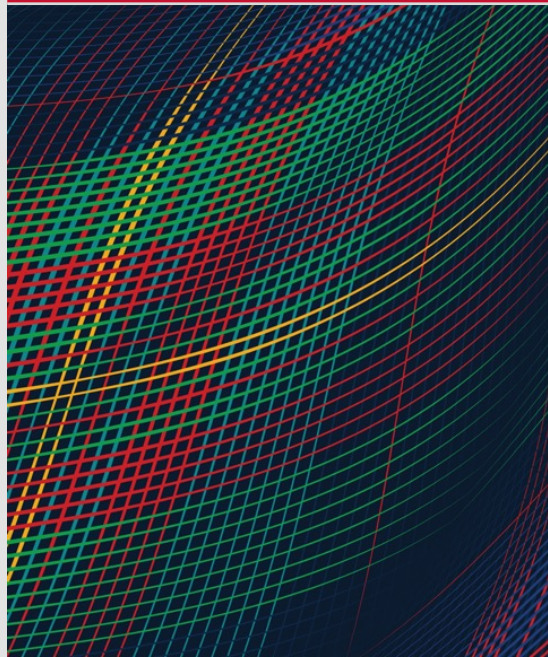
Probabilistic Verification to Support Next-Generation Certification

(ProVer-Cert)

APRIL 22, 2026

Sam Procter
Dio De Niz

Carnegie Mellon University
Software Engineering Institute



Document Markings

This material is based upon work supported by the Department of War under Air Force 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.

DM26-0399

Probabilistic Verification to Support Next-Generation Certification (ProVer-Cert)

Problem

Carnegie Mellon University
Software Engineering Institute

Problem: Software Certification is often *Process*-Based

Despite *Property*-Based Certification's advantages

Carnegie Mellon University Software Engineering Institute

DO-178C:

- Standardized guidance used by FAA
- Activities: Planning, verification, tracing

Benefits:
- Predictable, schedulable
- Good safety record

Drawbacks:
- Confidence subjective and not quant.
- Supplements required for new tech

Overarching Properties (OPs):

- Developed by NASA, FAA, others
- Properties: Sufficient for certification: Intent, Correctness, and Innocuity

Benefits:
- Flexible / Powerful

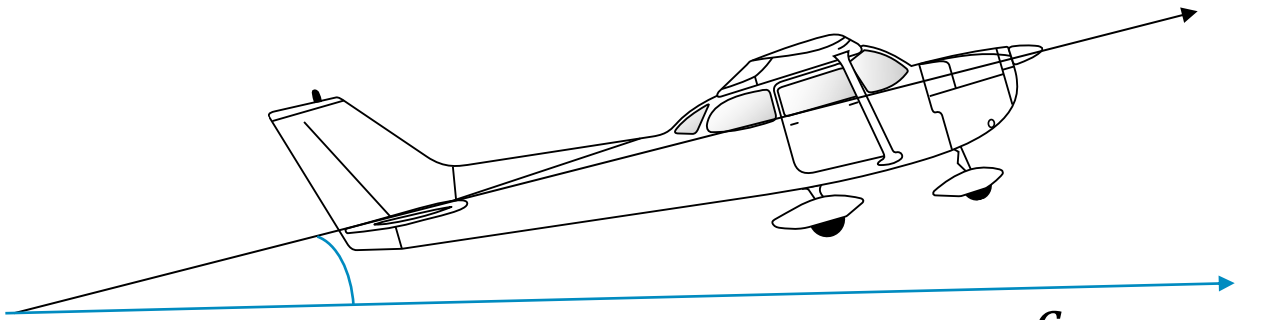
Drawbacks:
- New
- Needs clarity on how to apply

Establishing Correctness:

- Tests are the most common approach.
  - A failure indicates a bug
  - A passed test gives only an *unquantifiable* increase in confidence
- Need quantitative, statistical increase in confidence from passed tests
- How? FACT approach

Approach:

1. Model the current, process-based certification process
2. Develop and model a probabilistic assessment method to support property-based certification
3. Compare the models for inputs, outputs, resources required, etc.

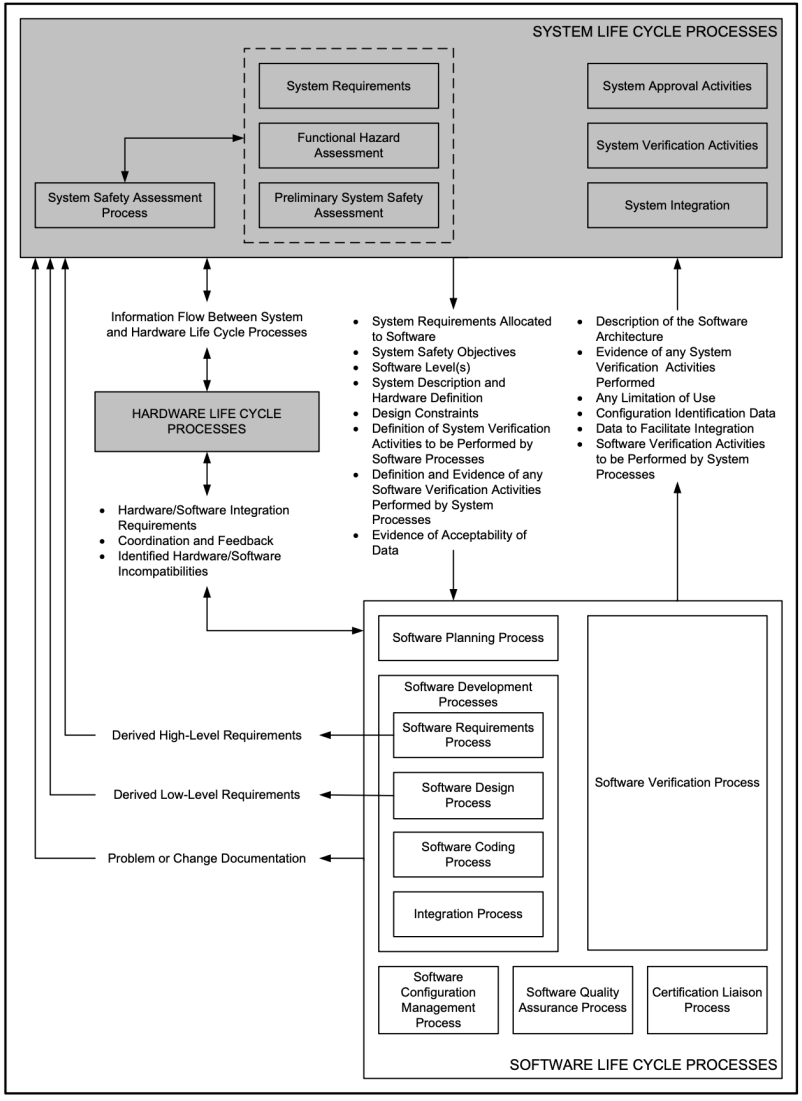


DO-178C

(The) Standard for Avionics Software

Process focus:
- Development
- Testing

Traceability, from:
- Requirements to code
- Code to tests
- Requirements to tests



RTCA DO-178C, "Software Considerations in Airborne Systems and Equipment Certification," December 2011.

Overarching Properties

If a system has these three properties, it should be certified:

Intent: The defined intended behavior is correct and complete with respect to the desired behavior.

Correctness: The implementation is correct with respect to its defined intended behavior, under foreseeable operating conditions.

Innocuity: Any part of the implementation that is not required by the defined intended behavior has no unacceptable impact.

Holloway, C. Michael. 2019. *Understanding the Overarching Properties*. Nos. NF1676L-33745. https://ntrs.nasa.gov/citations/20190029284.

Background: Research Areas

Carnegie Mellon University Software Engineering Institute

Certification

- Assurance Argumentation
  - Property-Based Certification: Overarching Properties [1,2]

Quantifying Reliability

- FACT: Formally verifies / computes confidence intervals of QoS properties for software Systems [3]
  - VERACITY: Incorporates FACT to guide testing [4]
- Incorporation of Subjective Assessments [5,6]

1. Wasson, Kimberly S., and C. Michael Holloway. 2022. An Introduction to Constructing and Assessing Overarching Properties Related Arguments (OPRAs): Version 1.0. https://ntrs.nasa.gov/citations/20210025425.
2. Holloway, C. Michael. 2019. *Understanding the Overarching Properties*. Nos. NF1676L-33745. https://ntrs.nasa.gov/citations/20190029284.

3. Calinescu, Radu, Carlo Ghezzi, Kenneth Johnson, Mauro Pezzé, Yasmin Rafiq, and Giordano Tamburrelli. 2016. "Formal Verification With Confidence Intervals to Establish Quality of Service Properties of Software Systems." *IEEE Transactions on Reliability* 65 (1): 107–25. https://doi.org/10.1109/TR.2015.2452931.
4. Alasmari, Naif, Radu Calinescu, Colin Paterson, and Raffaela Mirandola. 2022. "Quantitative Verification with Adaptive Uncertainty Reduction." *Journal of Systems and Software* 188 (June): 111275. https://doi.org/10.1016/j.jss.2022.111275.
5. Strigini, Lorenzo, and Andrey Povyakalo. 2013. "Software Fault-Freeness and Reliability Predictions." In *Computer Safety, Reliability, and Security. SAFECOMP 2013*, edited by Friedemann Bitsch, Jérémie Guiochet, and Mohamed Kaâniche. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-40793-2_10.
6. Bishop, Peter, Robin Bloomfield, Bev Littlewood, Andrey Povyakalo, and David Wright. 2011. "Toward a Formalism for Conservative Claims about the Dependability of Software-Based Systems." *IEEE Transactions on Software Engineering* 37 (5): 708–17. https://doi.org/10.1109/TSE.2010.67.

Probabilistic Verification to Support Next-Generation Certification (ProVer-Cert)

Tasks

Carnegie Mellon University
Software Engineering Institute

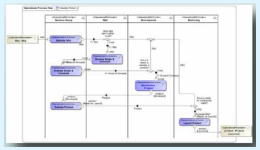
Process

Q1 Q2 Q3 Q4

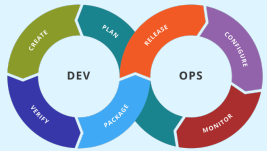
Define Use Case (Narrative)

- _____
- _____
- _____

Model (slice of) DO-178C Approach

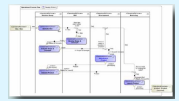


Model DevOps / Log++ Approach



Log++ ΦψΘ

Compare Modeled Processes

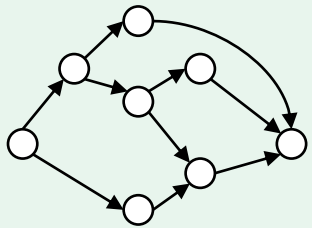


?

Encode Properties in PCTL

Φ…ψ…Θ…

Create Abstract Markov Chain



Create Observation Mechanism to Collect Transition Statistics from Runtime Data

Log++ ΦψΘ

Write & Publish

Technical Approach: Task 1
Define Use Case

We will define a use case to support our work in Tasks 2-4.

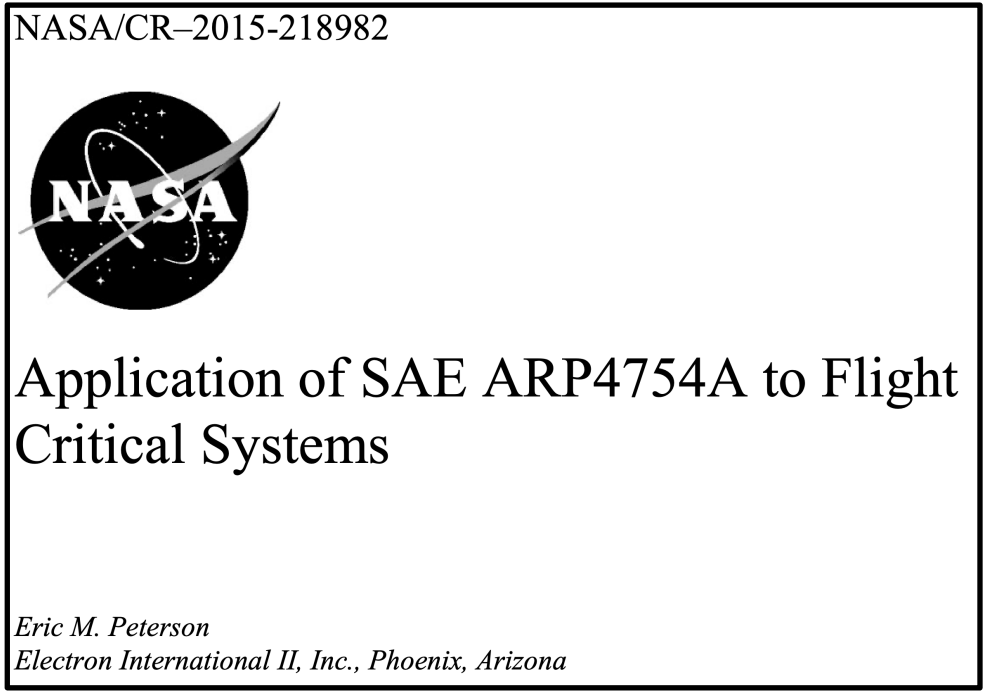
- The use case will be scoped carefully to:
  - Support (partial, mock) certification under DO-178C
  - Support (partial, mock) verification of correctness under the Overarching Properties
- The domain will be avionics, using a drone
  - This aligns us with the domain of DO-178 and many of the users of the Overarching Properties
- The structure will be narrative text augmented with models

Use Case: Source

Goal: Example certification from aviation domain

DO-178C: Software Considerations in Airborne Systems and Equipment Certification

- Coordinates with system development via, e.g., ARP4754A: “System life cycle processes can be found in other industry documents (for example, SAE ARP4754A).”


NASA/CR–2015-218982

NASA

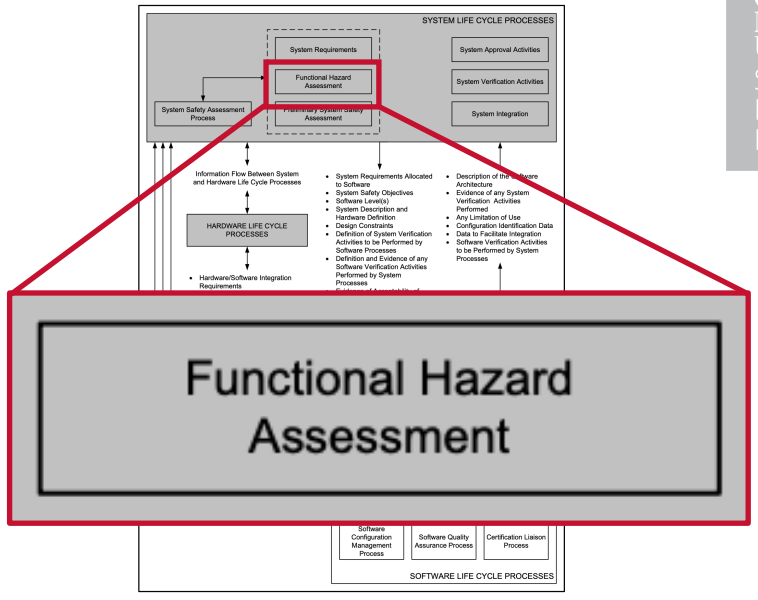
Application of SAE ARP4754A to Flight Critical Systems

Eric M. Peterson
Electron International II, Inc., Phoenix, Arizona

Use Case: Properties

Goal: Important property we can analyze quantitatively

We selected a property from the Functional Hazard Assessment in the example ARP4754A document



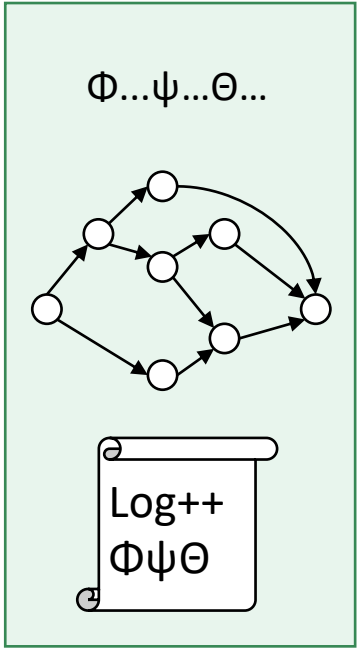
Function	FC #	Failure Condition (Hazard Description)	Flight Phase	Effect of Failure Condition on Aircraft/Crew	Classification
Provide Stability & Control: Automatic Stability & Control (2.5)	22.03	Erroneous autopilot command which exceeds authority limits	Flight	Airplane structural damage may result due to unrestricted pitch, roll or yaw commands. May result in rapid flight path responses, unsafe airplane flight paths and loss of altitude. Possible ground contact if occurs at low altitude resulting in loss of airplane.	Catastrophic

Technical Approach: Task 2
Extract and Quantify Probabilities

Currently the FACT approach verifies properties encoded in Probabilistic Computation Tree Logic (PCTL) on parameterized Markov chains.

In this task we will:

1. Encode specific safety / reliability properties from the use case developed in Task 1 in PCTL
2. Create the parameterized, abstract Markov chains which will support the establishment of confidence intervals for the the properties
3. Develop an instrumentation and observation approach (Log++) which will allow us to extract values for the parameters in the Markov chain. This will consist of two novel aspects:
    1. Determining what exactly needs to be observed to have coverage of the parameterized transitions in the abstract Markov chain
    2. Incorporating a sampling technique that ensures samples are independent and identically distributed (IID)

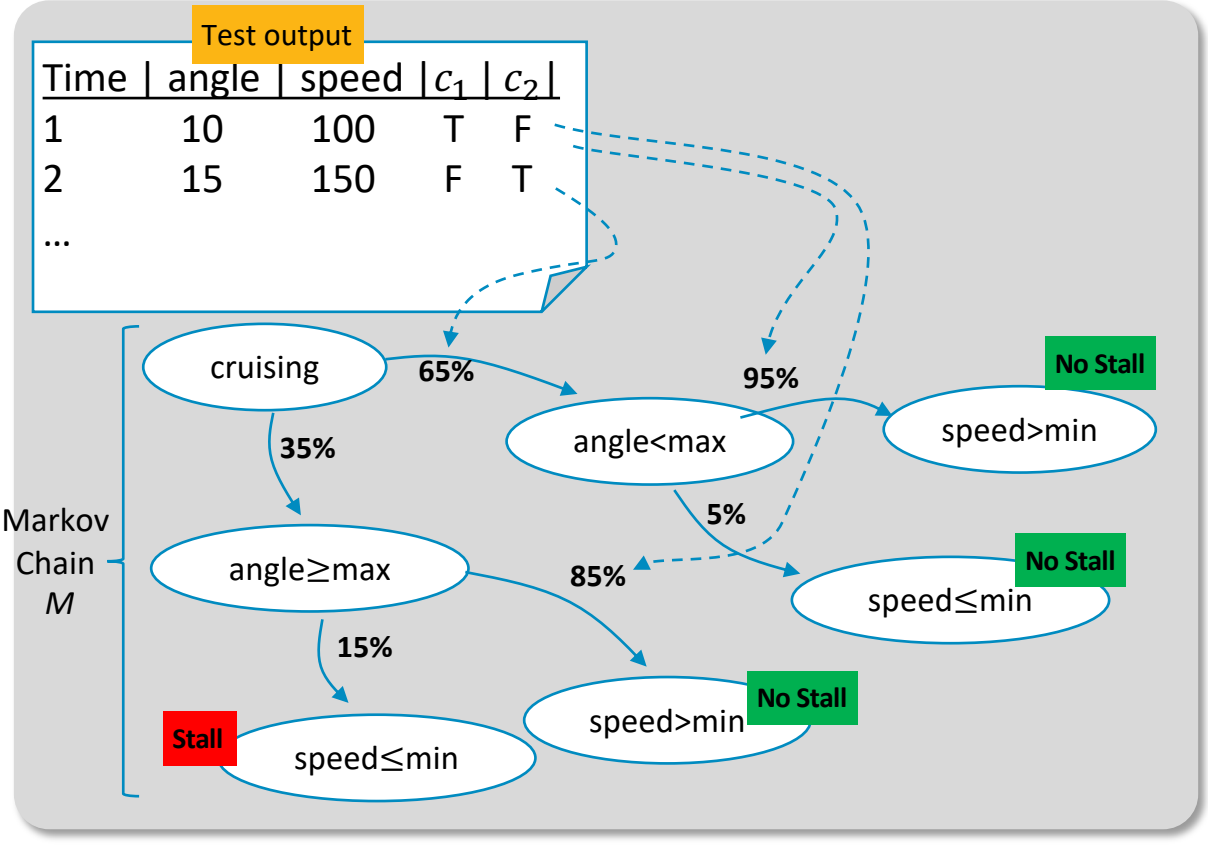


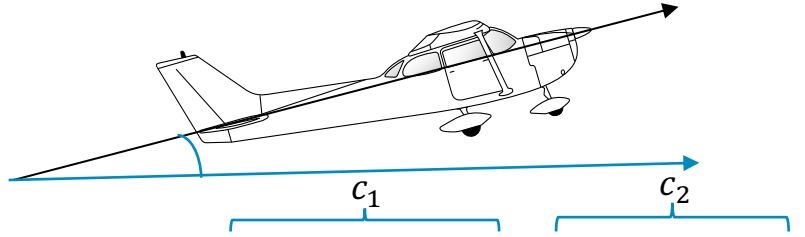
Example Safety Property

Carnegie Mellon University
Software Engineering Institute

Test output

Time	angle	speed	c_1	c_2
1	10	100	T	F
2	15	150	F	T
…				





No Stall (NS) := $G\ angleattack < maxangle\ U\ airspeed > minspeed$

More tests = Better confidence!

FACT: Probability that our property (*No Stall*) is within the confidence interval [a,b] is less than α

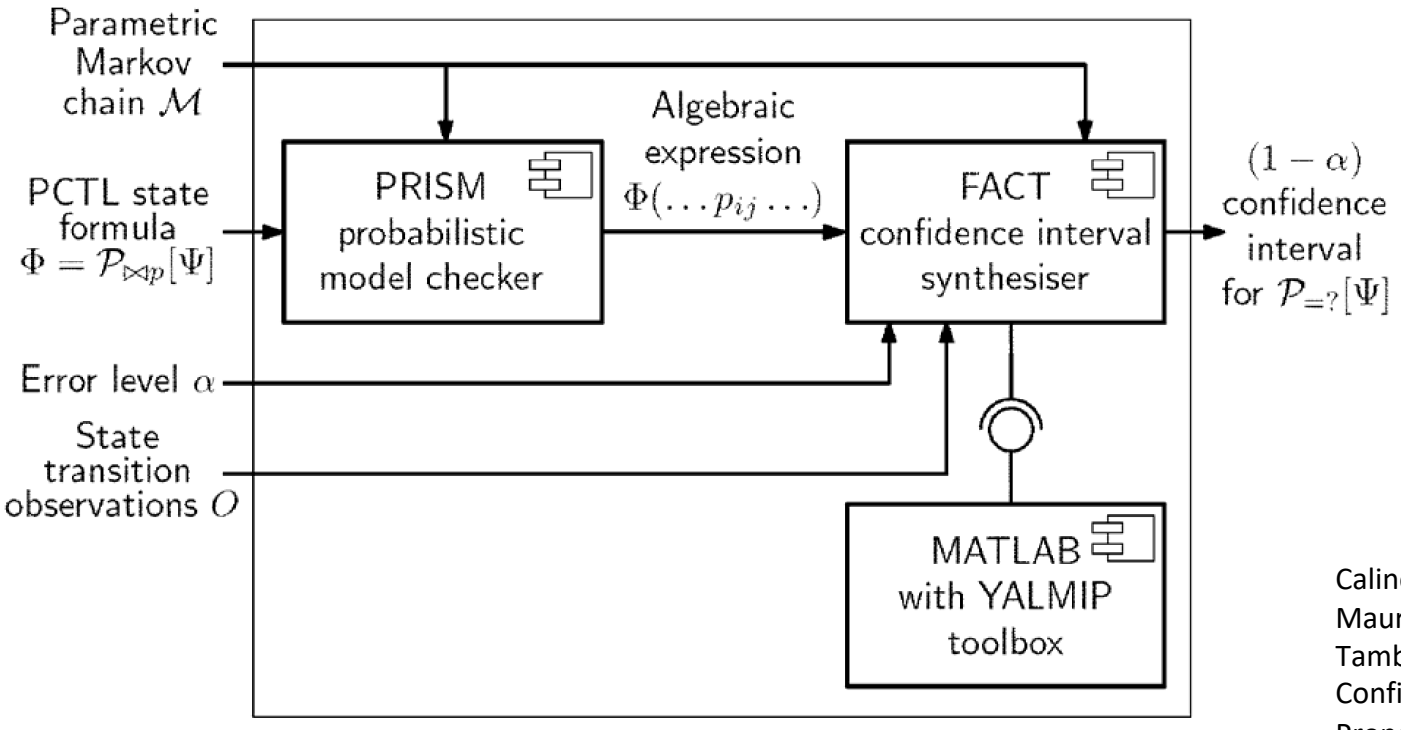
$$\text{Prob}(\text{Prop}(\pi \in paths^M(cruising) \mid \pi \vDash NoStall) \notin [a,b]) < \alpha$$

e.g.,

$$\text{Prob}(\text{Prop}(\pi \in paths^M(cruising) \mid \pi \vDash NoStall) \notin [.95,.99]) < .05$$

By Frank Murmann - Own work, CC BY 3.0, https://commons.wikimedia.org/w/index.php?curid=68211518

The FACT Technique: Architecture

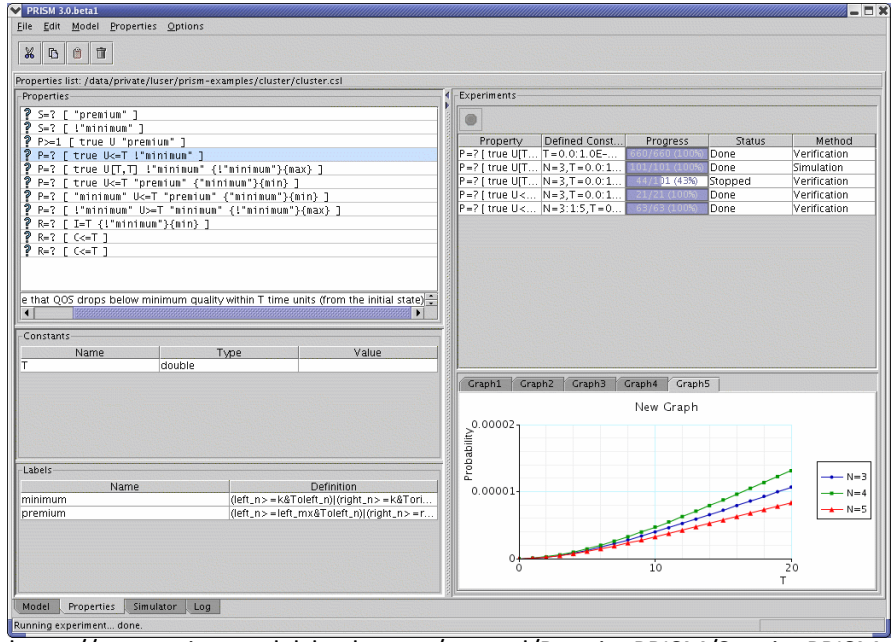


Calinescu, Radu, Carlo Ghezzi, Kenneth Johnson, Mauro Pezzé, Yasmin Rafiq, and Giordano Tamburrelli. 2016. "Formal Verification With Confidence Intervals to Establish Quality of Service Properties of Software Systems." *IEEE Transactions on Reliability* 65 (1): 107–25. https://doi.org/10.1109/TR.2015.2452931.

Probabilistic Model Checking

- “PRISM is a probabilistic model checker, a tool for formal modelling and analysis of systems that exhibit random or probabilistic behaviour.”

- “Models are described using the PRISM language, a simple, state-based language. PRISM provides support for automated analysis of a wide range of quantitative properties of these models… The property specification language incorporates the temporal logics PCTL, CSL, LTL and PCTL*, as well as extensions for quantitative specifications and costs/rewards.”

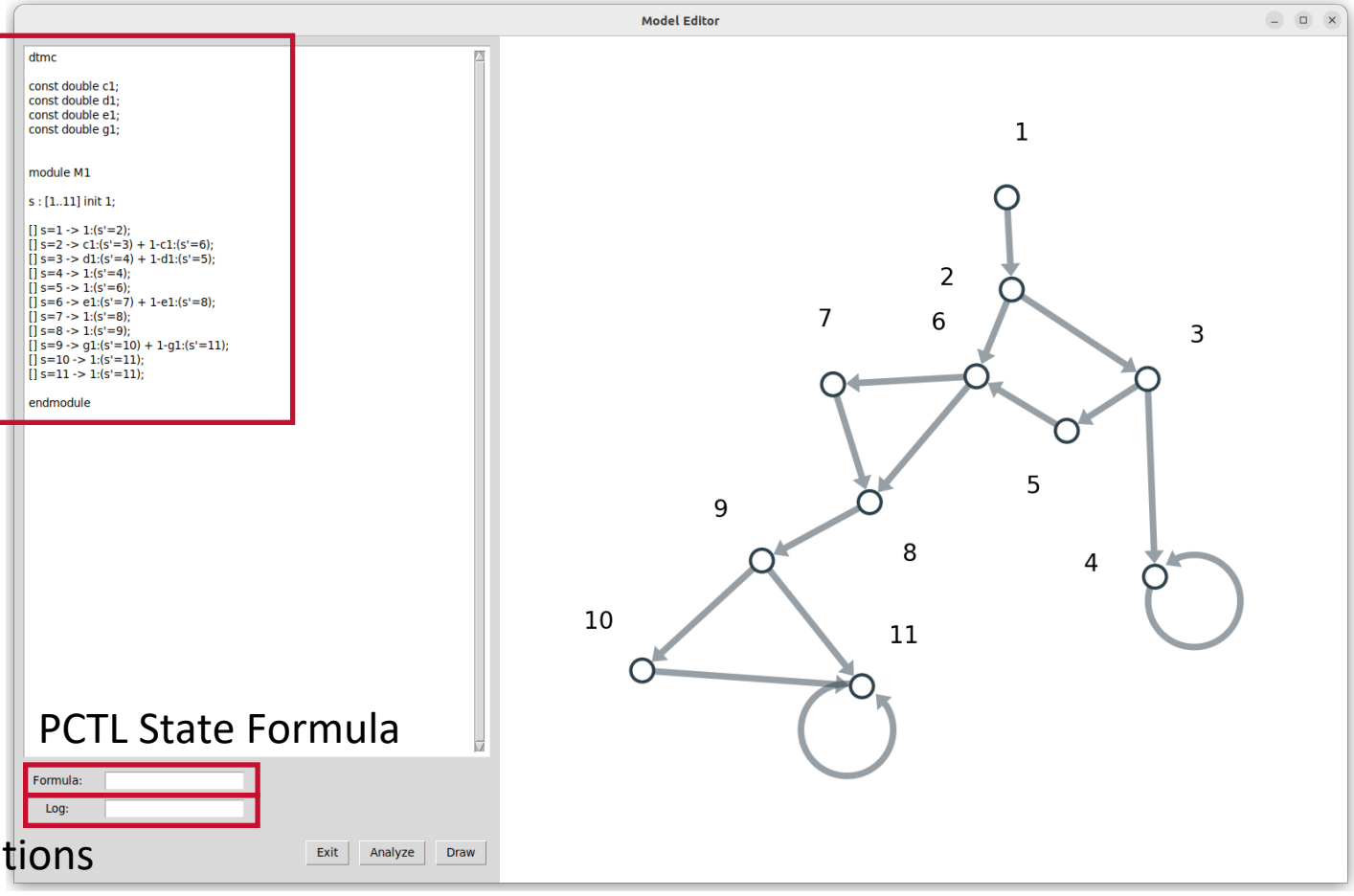


https://www.prismmodelchecker.org/manual/RunningPRISM/StartingPRISM

Marta Kwiatkowska, Gethin Norman and David Parker. PRISM 4.0: Verification of Probabilistic Real-time Systems. In *Proc. 23rd International Conference on Computer Aided Verification (CAV'11)*, volume 6806 of LNCS, pages 585-591, Springer, 2011

Verification Tool

Parametric Markov Chain



State Observations

Instrumentable Implementation

Goal: Executable code we can instrument

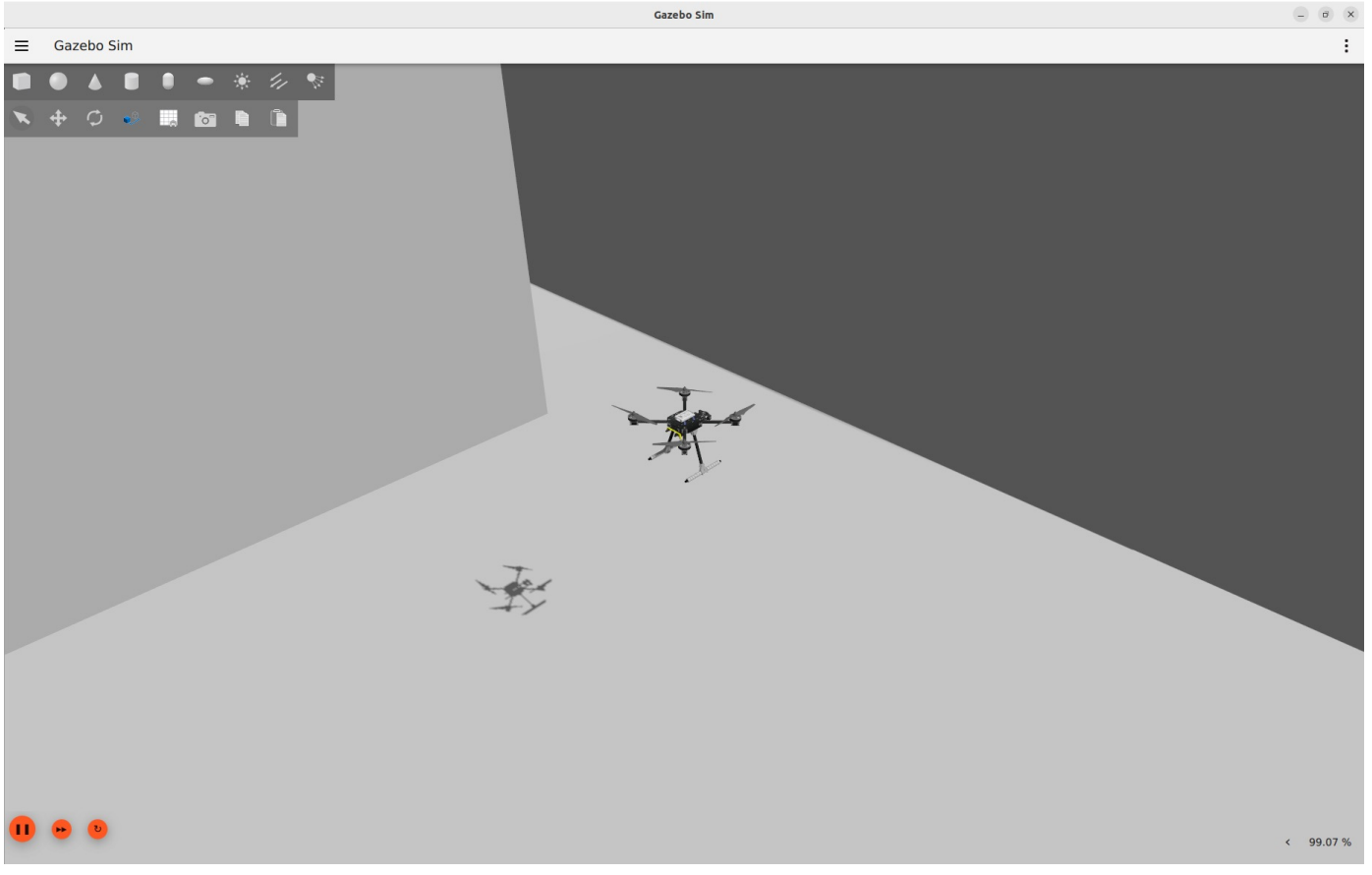
- Need to be able to observe / map to transitions in Markov Chain

Use PX4 Autopilot and the Gazebo simulator.

Logo from https://github.com/PX4/PX4-Autopilot

Logo from https://gazebosim.org

PX4 + Gazebo Simulator



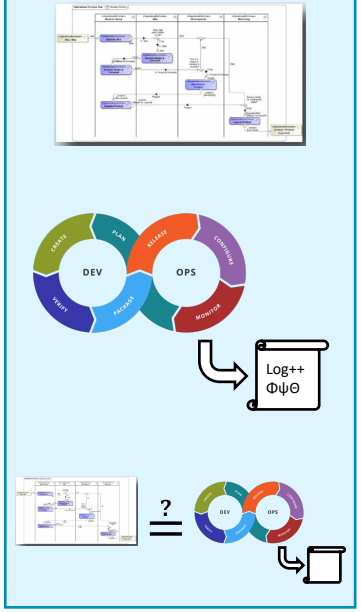
Technical Approach: Task 3
Model Certification Processes

Currently DO-178C, the Overarching Properties, and their assessment methods (aka Related Arguments (OPRAs)) are described textually.

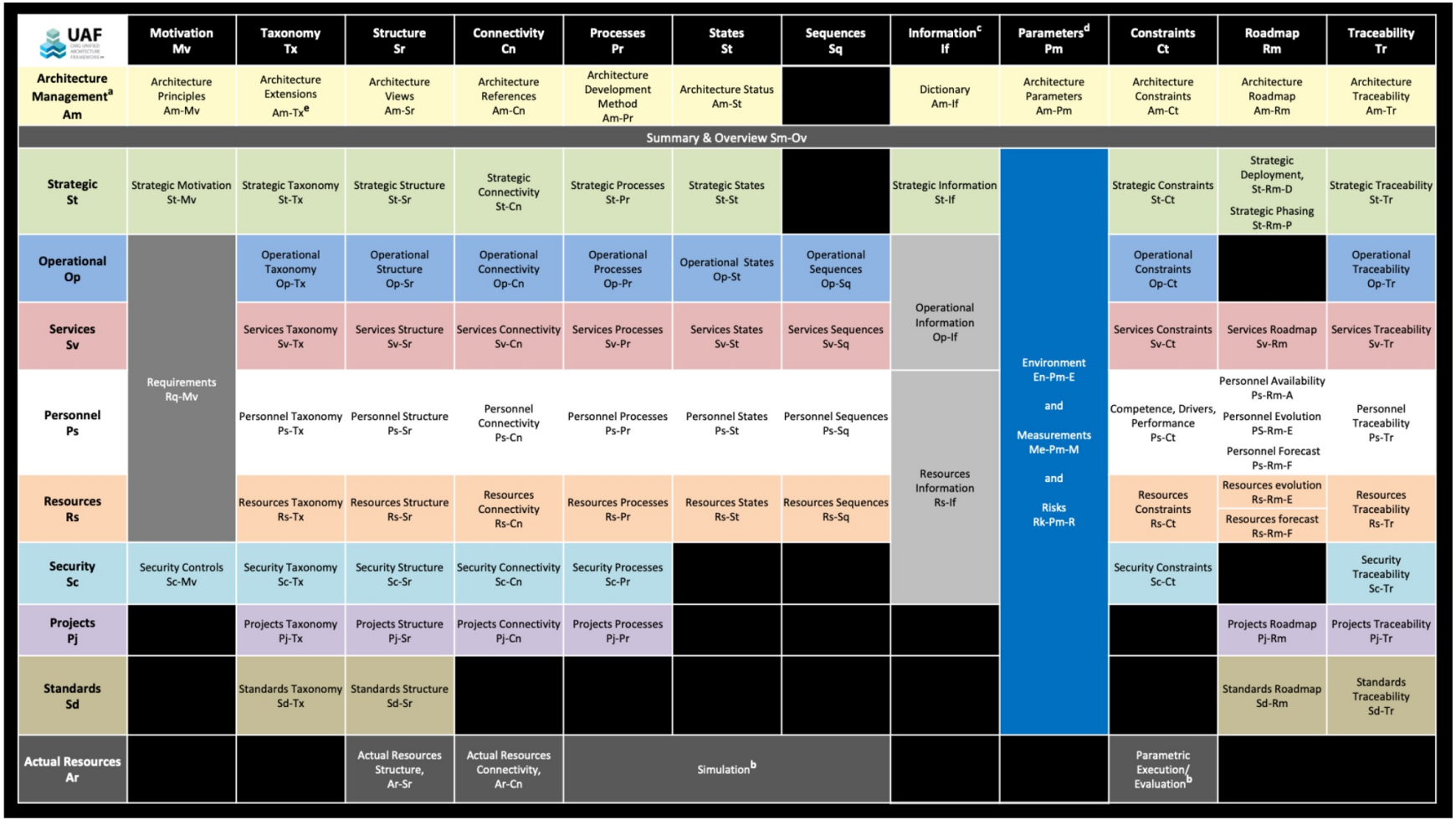
In this task we will model in UAF using Cameo:

1. A slice of DO-178C activities necessary for certification of the example system
2. The usage of Operations data, as part of a larger system evaluation activities, to collect inputs necessary for the FACT approach

We will then compare the modeled processes to calculate the feasibility of our approach and the increase in support for verification of probabilistic quantitative system requirements.



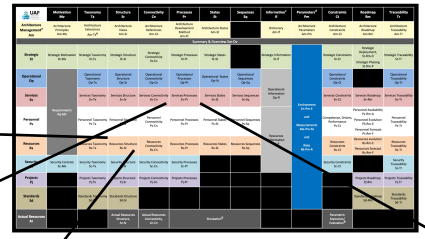
UAF Overview

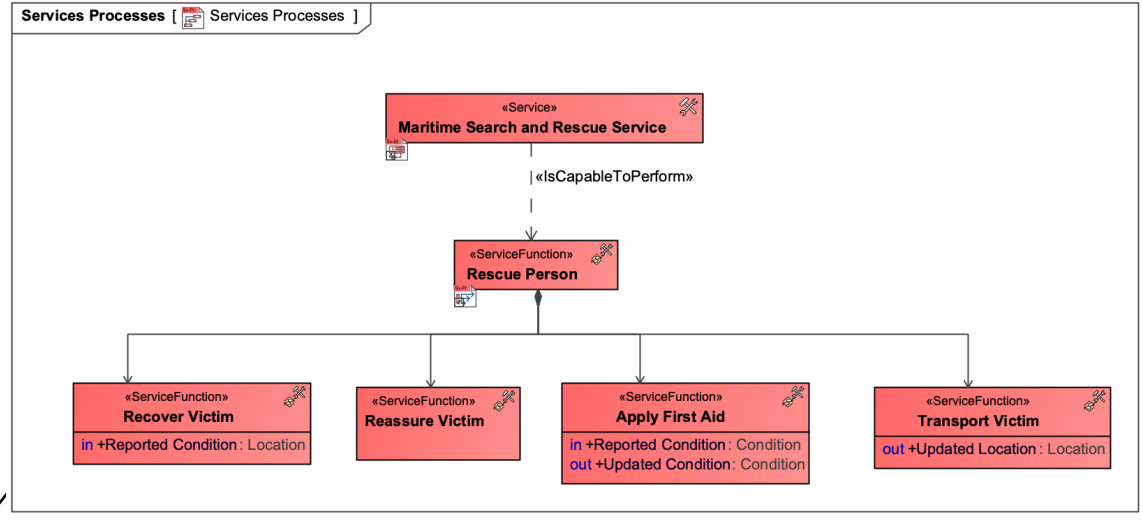


UAF	Motivation Mv	Taxonomy Tx	Structure Sr	Connectivity Cn	Processes Pr	States St	Sequences Sq	Information[c] If	Parameters[d] Pm	Constraints Ct	Roadmap Rm	Traceability Tr
Architecture Management[a] Am	Architecture Principles Am-Mv	Architecture Extensions Am-Tx[e]	Architecture Views Am-Sr	Architecture References Am-Cn	Architecture Development Method Am-Pr	Architecture Status Am-St		Dictionary Am-If	Architecture Parameters Am-Pm	Architecture Constraints Am-Ct	Architecture Roadmap Am-Rm	Architecture Traceability Am-Tr
Summary & Overview Sm-Ov												
Strategic St	Strategic Motivation St-Mv	Strategic Taxonomy St-Tx	Strategic Structure St-Sr	Strategic Connectivity St-Cn	Strategic Processes St-Pr	Strategic States St-St		Strategic Information St-If	Environment En-Pm-E and Measurements Me-Pm-M and Risks Rk-Pm-R	Strategic Constraints St-Ct	Strategic Deployment, St-Rm-D Strategic Phasing St-Rm-P	Strategic Traceability St-Tr
Operational Op	Requirements Rq-Mv	Operational Taxonomy Op-Tx	Operational Structure Op-Sr	Operational Connectivity Op-Cn	Operational Processes Op-Pr	Operational States Op-St	Operational Sequences Op-Sq	Operational Information Op-If		Operational Constraints Op-Ct		Operational Traceability Op-Tr
Services Sv		Services Taxonomy Sv-Tx	Services Structure Sv-Sr	Services Connectivity Sv-Cn	Services Processes Sv-Pr	Services States Sv-St	Services Sequences Sv-Sq			Services Constraints Sv-Ct	Services Roadmap Sv-Rm	Services Traceability Sv-Tr
Personnel Ps		Personnel Taxonomy Ps-Tx	Personnel Structure Ps-Sr	Personnel Connectivity Ps-Cn	Personnel Processes Ps-Pr	Personnel States Ps-St	Personnel Sequences Ps-Sq	Resources Information Rs-If		Competence, Drivers, Performance Ps-Ct	Personnel Availability Ps-Rm-A Personnel Evolution PS-Rm-E Personnel Forecast Ps-Rm-F	Personnel Traceability Ps-Tr
Resources Rs		Resources Taxonomy Rs-Tx	Resources Structure Rs-Sr	Resources Connectivity Rs-Cn	Resources Processes Rs-Pr	Resources States Rs-St	Resources Sequences Rs-Sq			Resources Constraints Rs-Ct	Resources evolution Rs-Rm-E Resources forecast Rs-Rm-F	Resources Traceability Rs-Tr
Security Sc	Security Controls Sc-Mv	Security Taxonomy Sc-Tx	Security Structure Sc-Sr	Security Connectivity Sc-Cn	Security Processes Sc-Pr					Security Constraints Sc-Ct		Security Traceability Sc-Tr
Projects Pj		Projects Taxonomy Pj-Tx	Projects Structure Pj-Sr	Projects Connectivity Pj-Cn	Projects Processes Pj-Pr						Projects Roadmap Pj-Rm	Projects Traceability Pj-Tr
Standards Sd		Standards Taxonomy Sd-Tx	Standards Structure Sd-Sr								Standards Roadmap Sd-Rm	Standards Traceability Sd-Tr
Actual Resources Ar			Actual Resources Structure, Ar-Sr	Actual Resources Connectivity, Ar-Cn	Simulation[b]					Parametric Execution/ Evaluation[b]		

OMG 2021, Unified Architecture Framework Domain Metamodel, Version 1.2, Object Management Group, https://www.omg.org/spec/UAF/1.2/Beta1/DMM/PDF

UAF Example Diagrams





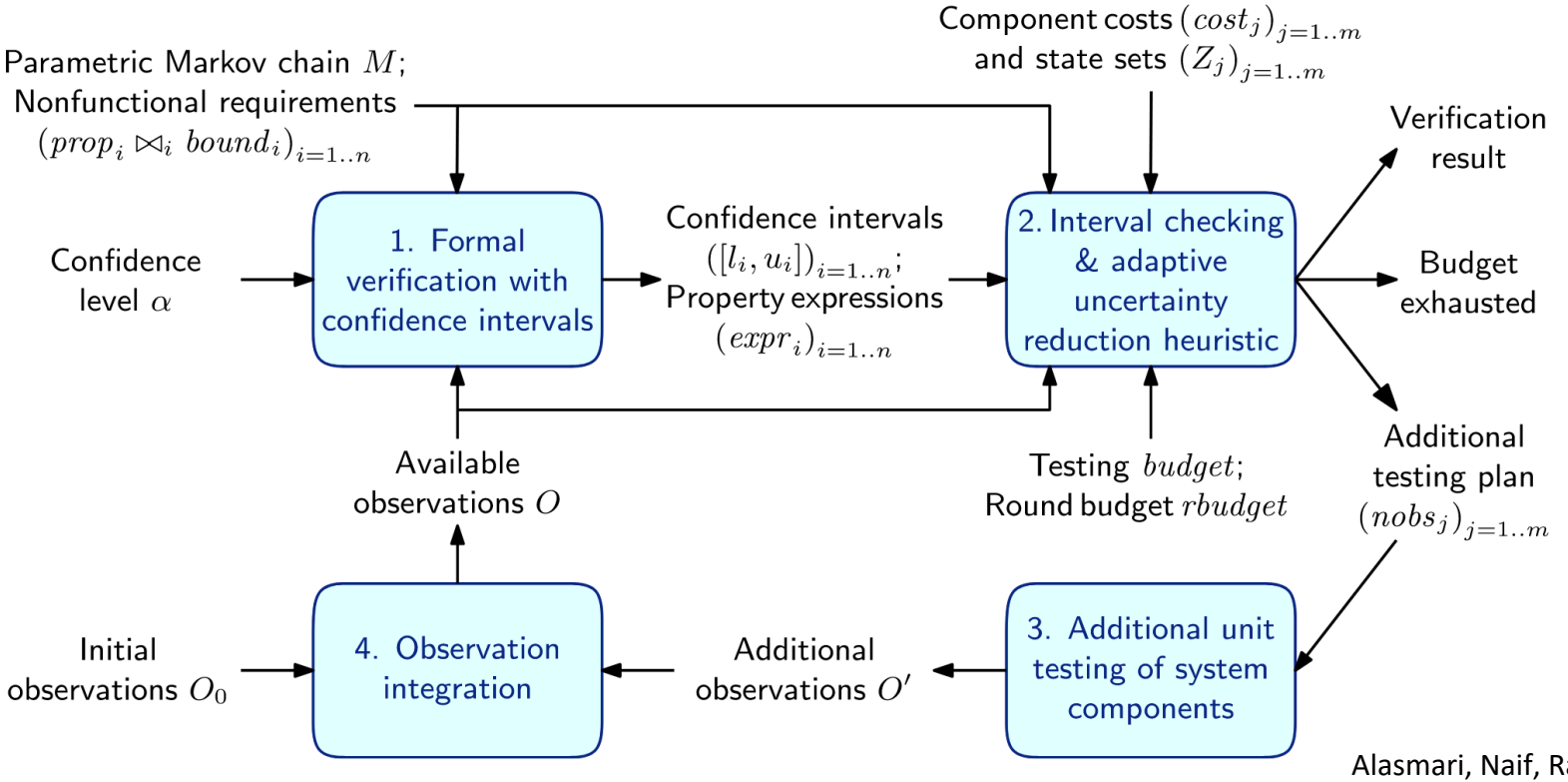
UAF Sample, Cameo Enterprise Architecture Samples.
Derived from concept for UK MOD by VEGA.

Carnegie Mellon University
Software Engineering Institute

Probabilistic Verification to Support Next-Generation Certification (ProVer-Cert)

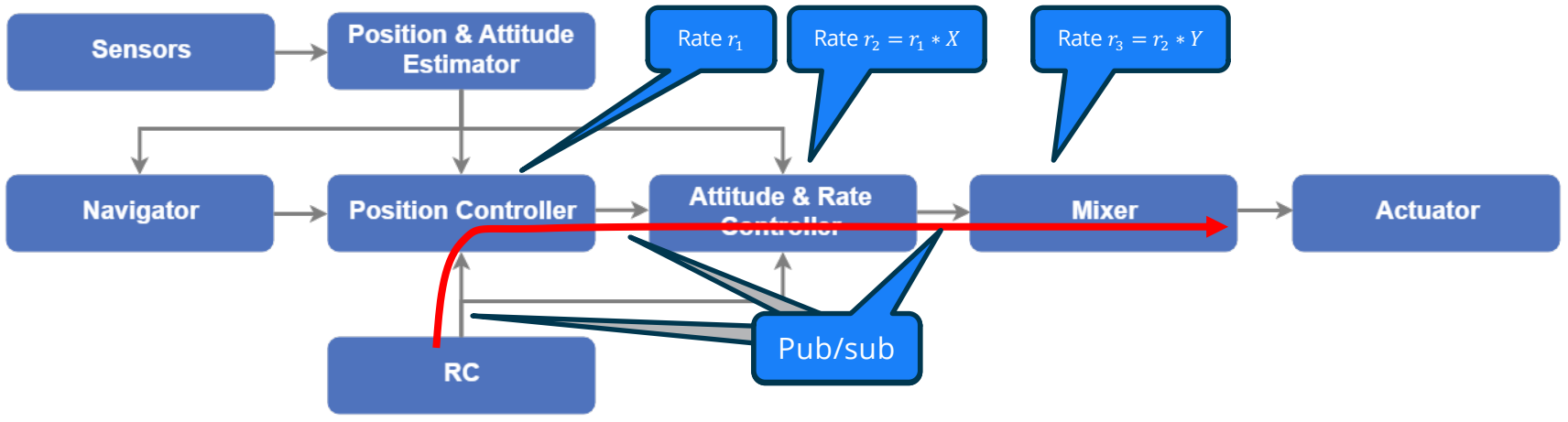
Next Steps

Next Steps: VERACITY



Alasmari, Naif, Radu Calinescu, Colin Paterson, and Raffaela Mirandola. 2022. "Quantitative Verification with Adaptive Uncertainty Reduction." *Journal of Systems and Software* 188 (June): 111275. https://doi.org/10.1016/j.jss.2022.111275.

Next Steps: Pub/Sub & PMC



Pub/sub creates asynchrony in data and control flow. This breaks assumptions of Probabilistic Model Checking!

Controller architecture use multiple control loops at different rates

Next Steps: Publications

Carnegie Mellon University Software Engineering Institute

Publications

- May: FACCT26: Extended Abstract (this talk)
- Late Summer: International Test and Evaluation Association (ITEA) journal

Presentations

- May: FACCT26

Source code

- https://github.com/dionisiodeniz/PX4-Autopilot
- https://github.com/dionisiodeniz/ProVerCert

Next Steps: Collaborations

Carnegie Mellon University Software Engineering Institute

Users

- Want to try this out?

Improvements

- Are there certification aspects we haven't considered?
- Statistical techniques that might be helpful?
- System properties you'd like to see included?

Let's talk! You can reach me at: sprocter@sei.cmu.edu