R2U2: Runtime Verification Takes Off!

Kristin Yvonne Rozier
Iowa State University
http://laboratory.temporallogic.org

VORTEX 2024

September 19, 2024

How is **Flight Software**

How is **Flight Software** Different from **Software**?

How is **Flight Software** Different from **Software**?

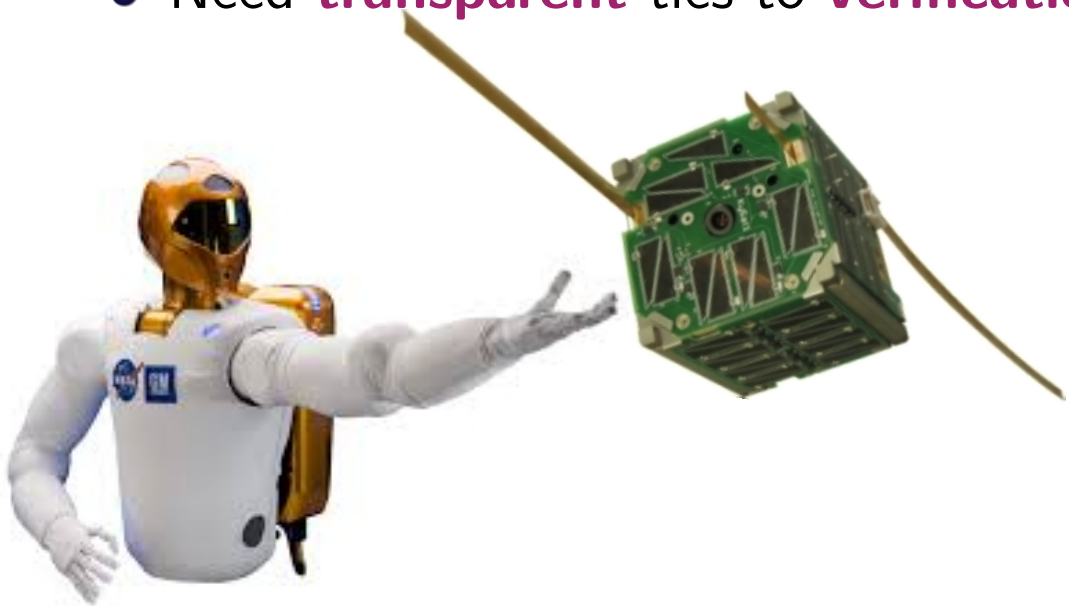
- **Has to** work

How is **Flight Software** Different from **Software**?

- **Has to** work
- Need capabilities for **independent checks**

How is **Flight Software** Different from **Software**?

- **Has to** work
- Need capabilities for **independent checks**
- Need **transparent** ties to **verification** tasks



Satisfying Requirements

Satisfying Requirements

RESPONSIVE
REALIZABLE
UNOBTRUSIVE
Unit

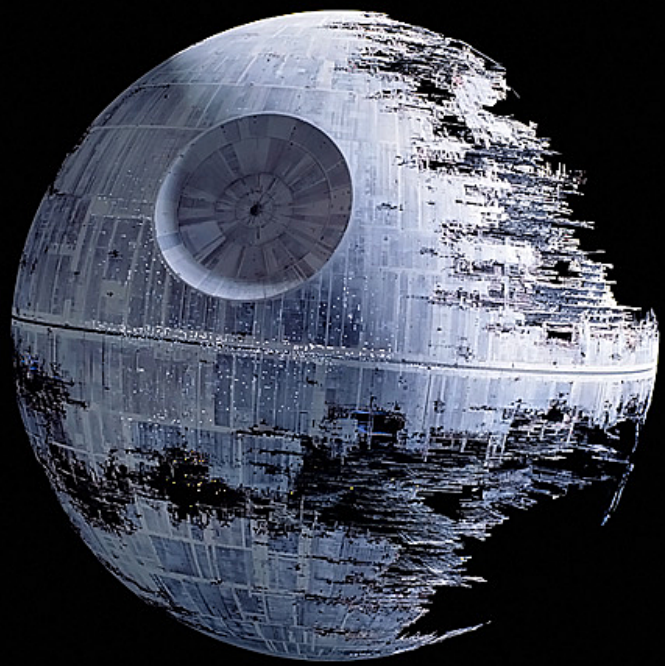
R2U2

Runtime Monitoring On-Board

Adding currently available runtime monitoring capabilities to the UAS would change its flight certification.

"Losing flight certification is like moving over to the dark side: once you go there you can never come back."

— Doug McKinnon,
NASA Ames' UAS Crew Chief



Requirements

Realizability:

- easy, *expressive* specification language
- *generic* interface to connect to a wide variety of systems
- *adaptable* to missions, mission stages, platforms

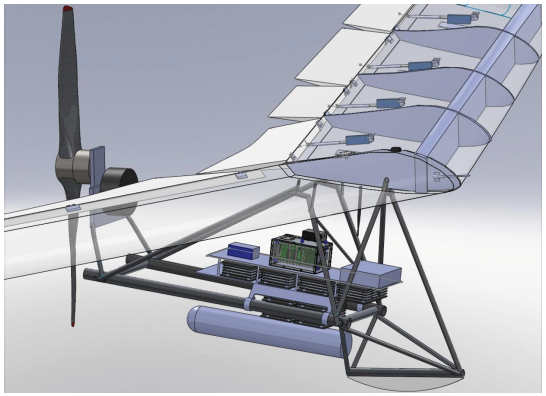
Responsiveness:

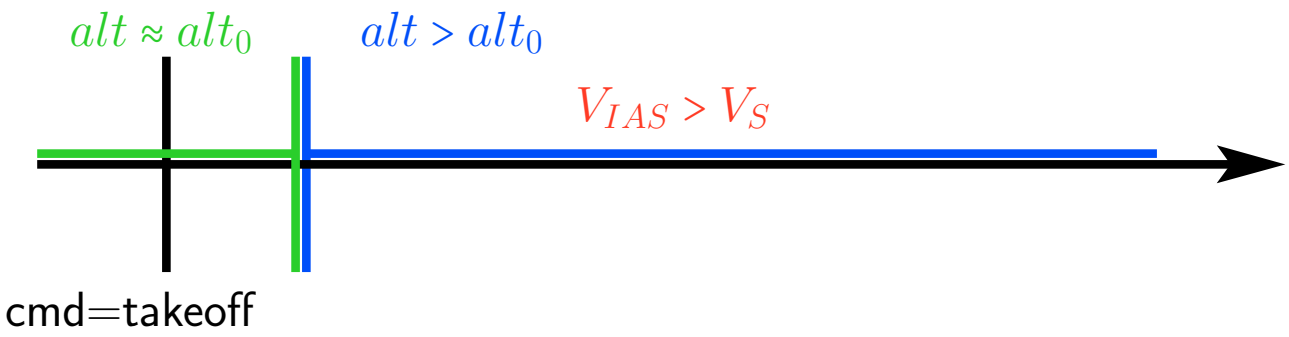
- *continuously monitor* the system
- *detect deviations* in *real time*
- *enable mitigation* or rescue measures

Unobtrusiveness:

- *functionality*: not change behavior
- *certifiability*: avoid re-certification of flight software/hardware
- *timing*: not interfere with timing guarantees
- *tolerances*: obey size, weight, power, telemetry bandwidth constraints
- *cost*: use commercial-off-the-shelf (COTS) components

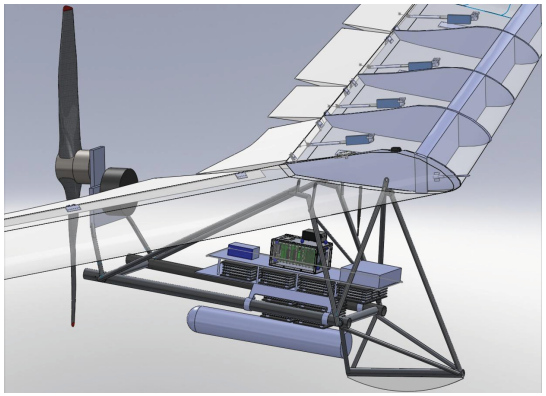
Runtime Observers for the Swift UAS

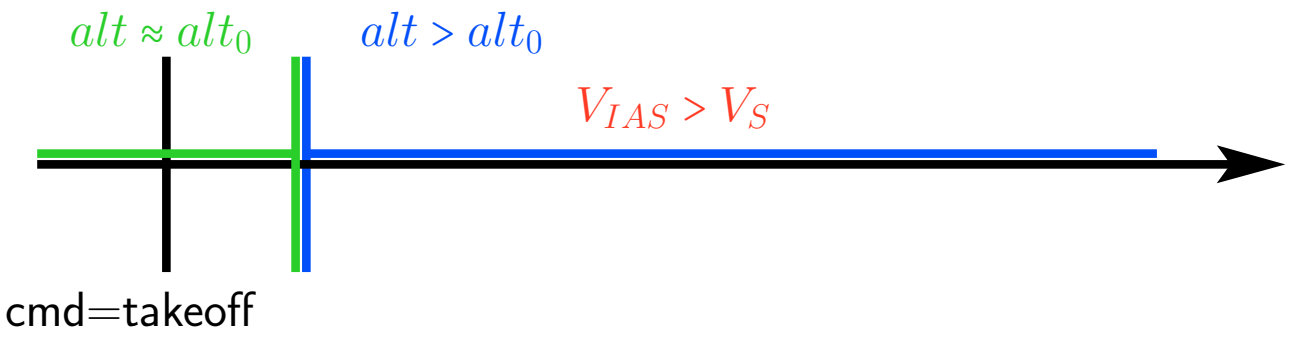




Whenever the Swift UAS is in the air, its indicated airspeed (V_{IAS}) must be greater than its stall speed V_S. The UAS is considered to be air-bound when its altitude *alt* is larger than that of the runway alt_0.

Runtime Observers for the Swift UAS





Whenever the Swift UAS is in the air, its indicated airspeed (V_{IAS}) must be greater than its stall speed V_S. The UAS is considered to be air-bound when its altitude *alt* is larger than that of the runway alt_0.

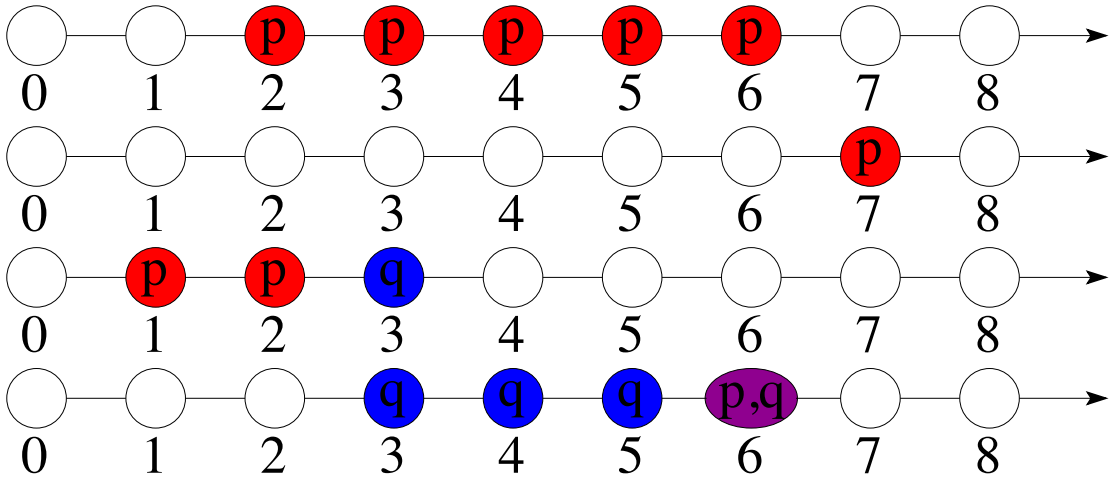
$$\text{ALWAYS}((alt > alt_0) \rightarrow (V_{IAS} > V_S))$$

Encoding Timelines: Mission-time Linear Temporal Logic

Mission-time LTL (MLTL) reasons about *bounded* timelines:

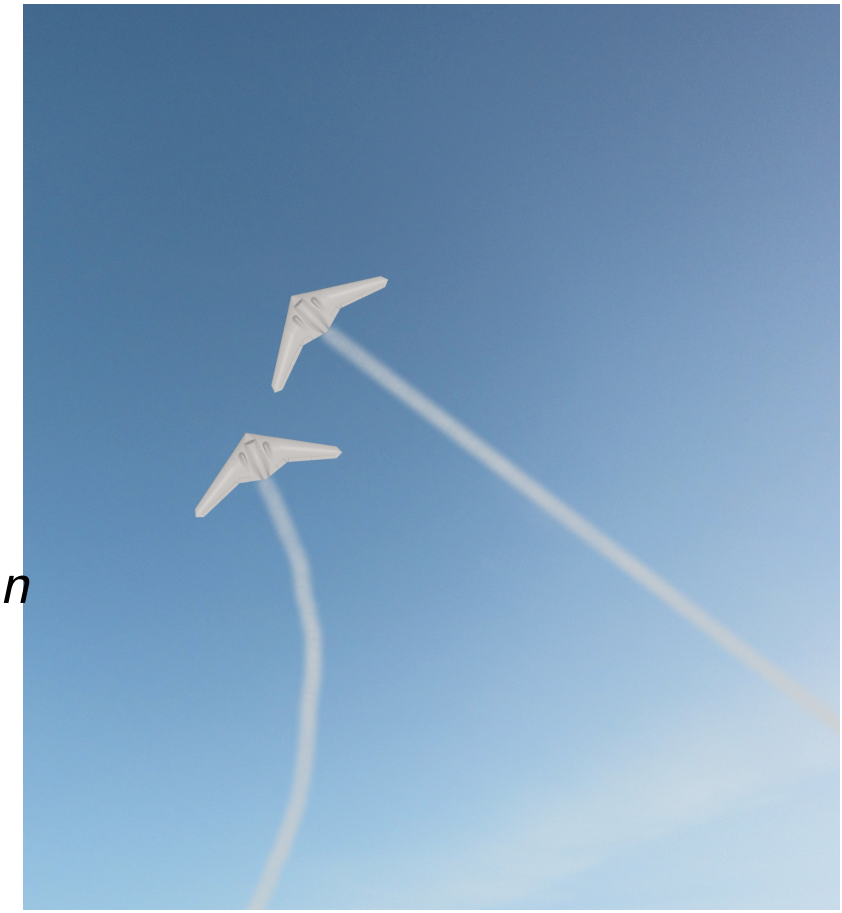
- finite set of atomic propositions {p q}
- Boolean connectives: ¬, ∧, ∨, and →
- temporal connectives *with time bounds*:

Symbol	Operator	Timeline
$\Box_{[2,6]} p$	$\text{ALWAYS}_{[2,6]}$	
$\Diamond_{[0,7]} p$	$\text{EVENTUALLY}_{[0,7]}$	
$p \mathcal{U}_{[1,5]} q$	$\text{UNTIL}_{[1,5]}$	
$p \mathcal{R}_{[3,8]} q$	$\text{RELEASE}_{[3,8]}$	



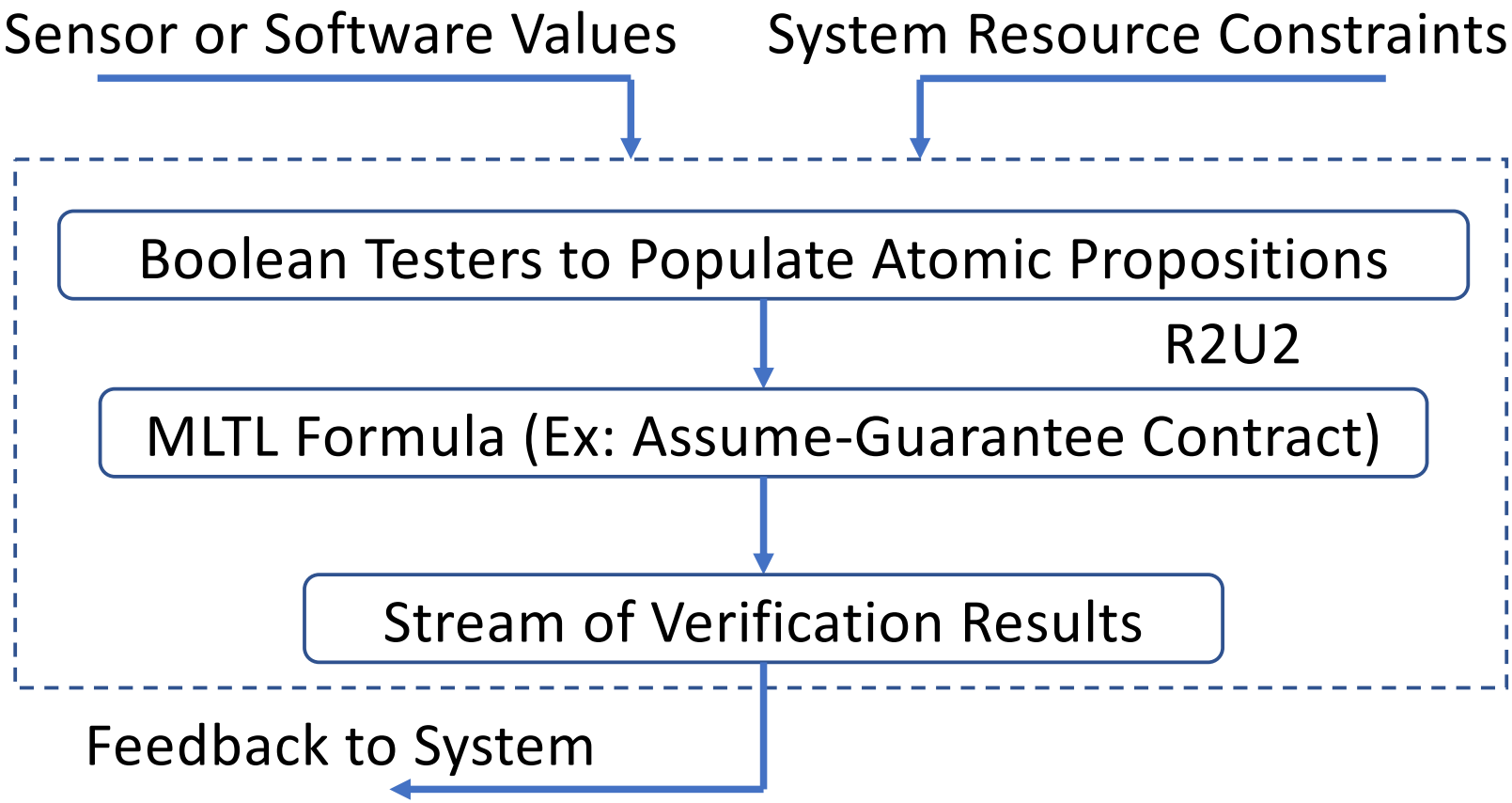
Asynchronous Observers (aka event-triggered)[1]

- *evaluate with every new input*
- 2-valued output: {true; false}
- resolve verdict *as early as possible* (a priori known time)
- for each clock tick, may resolve verdict for clock ticks prior to the current time *n* if the information required for this resolution was not available until *n*



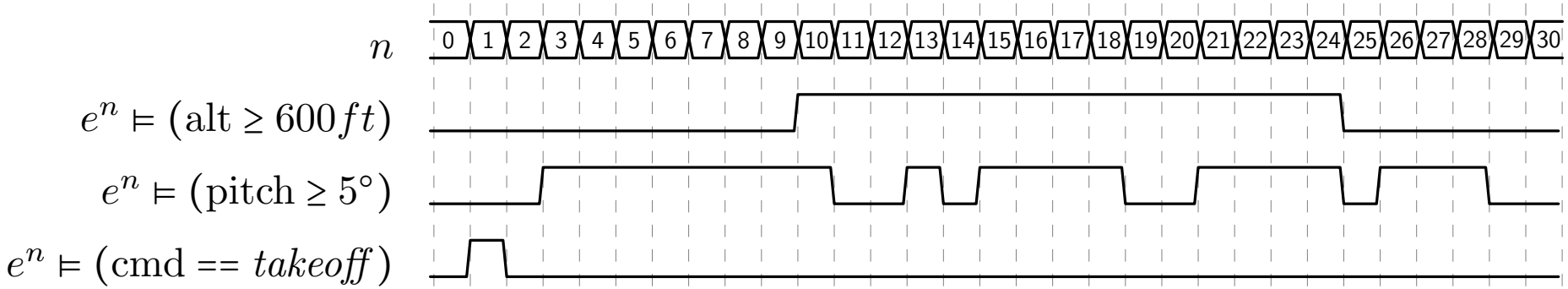
[1] Thomas Reinbacher, Kristin Y. Rozier, and Johann Schumann. "Temporal-Logic Based Runtime Observer Pairs for System Health Management of Real-Time Systems." In *Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, volume 8413 of Lecture Notes in Computer Science (LNCS), pages 357–372, Springer-Verlag, April, 2014.

R2U2 High-Level Architecture[2]



[2] Rozier, Kristin Y., and Johann Schumann. "R2U2: tool overview." (2017)

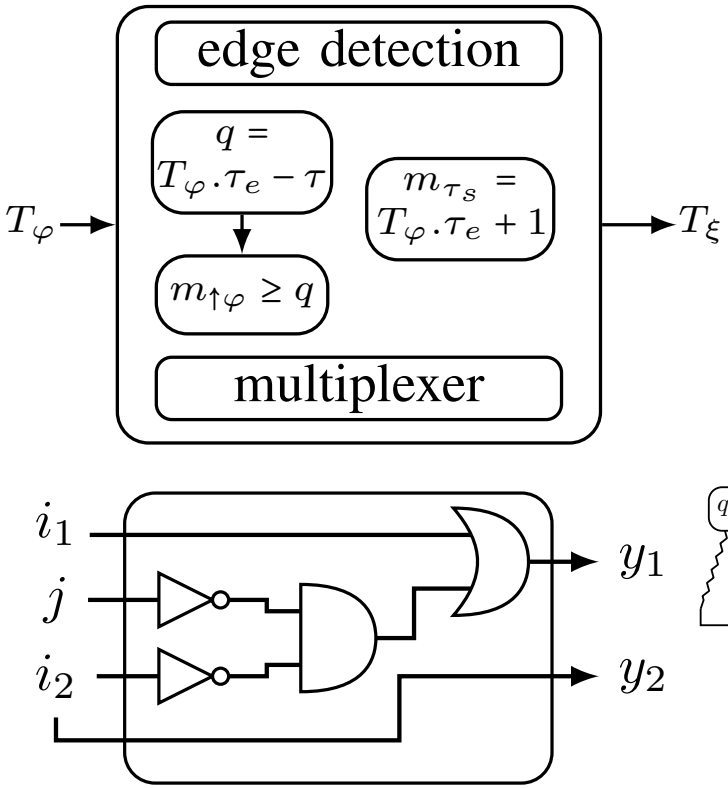
Asynchronous Observers Example

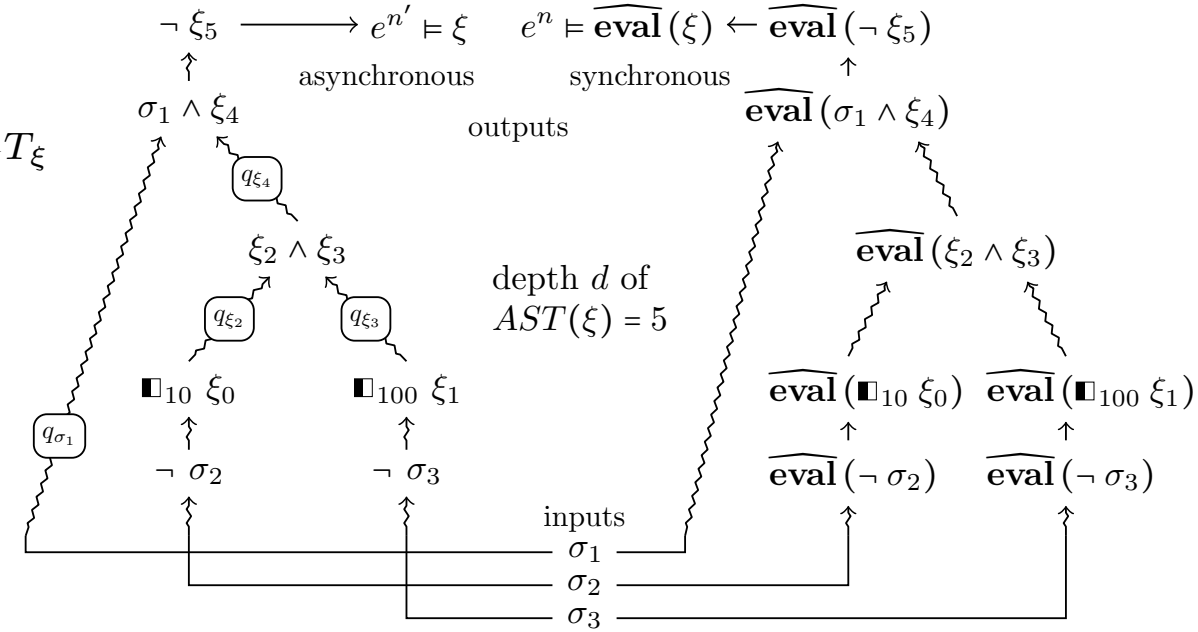


$\text{ALWAYS}_{[5]}(\text{pitch} \geq 5°)$

0	(false,0)	8	(true,3)
1	(false,1)	9	(true,4)
2	(false,2)	10	(true,5)
3	(␣, ␣)	11	(false,11) Resynchronized!
4	(␣, ␣)	12	(false,12)
5	(␣, ␣)	13	(␣, ␣)
6	(␣, ␣)	14	(false,14) Resynchronized!
7	(␣, ␣)	15	(␣, ␣)

FPGA Implementation of Temporal Observers[3]

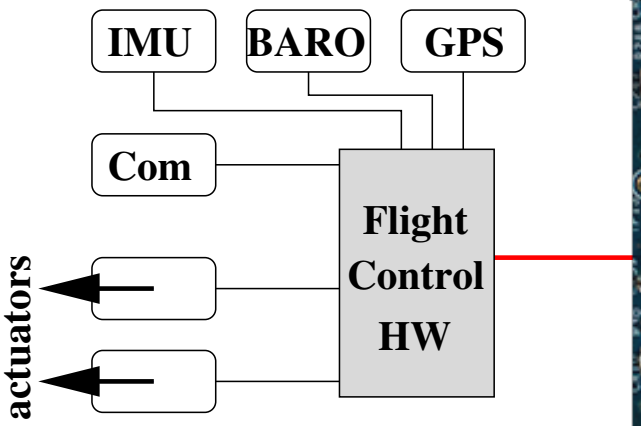




- asynchronous observers: substantial hardware complexity
- synchronous observers: small HW footprint

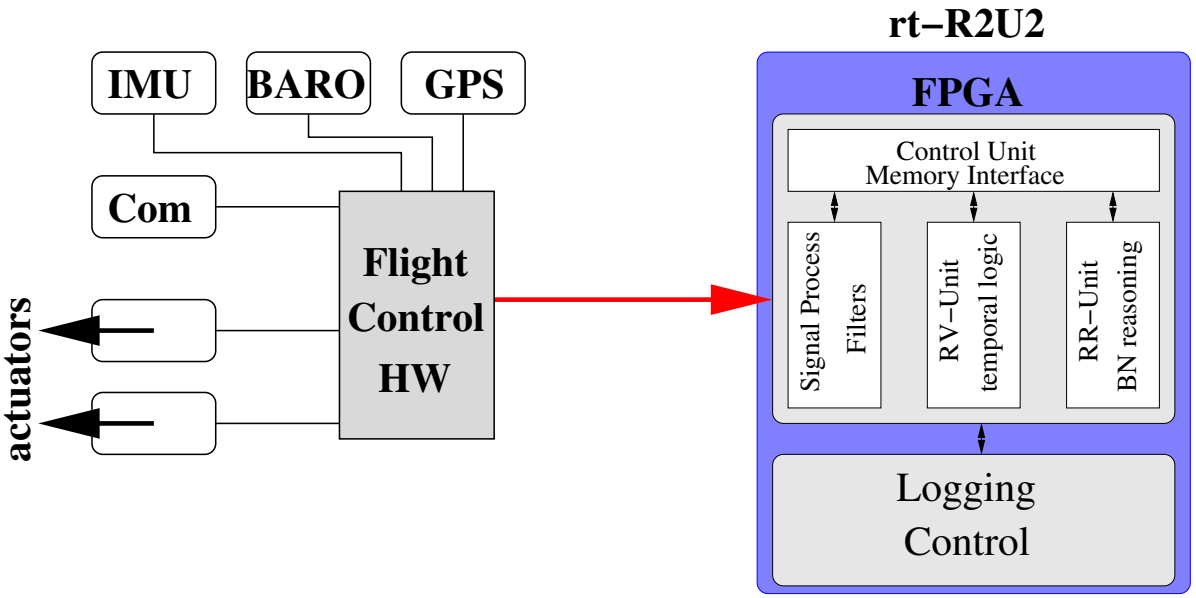
[3] Thomas Reinbacher, Kristin Y. Rozier, and Johann Schumann. "Temporal-Logic Based Runtime Observer Pairs for System Health Management of Real-Time Systems." In *Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, volume 8413 of Lecture Notes in Computer Science (LNCS), pages 357–372, Springer-Verlag, April, 2014.

Hard- and Software Architecture: Resource Estimation



- How do we fit in the resources left over?
- Choose between 3 R2U2 implementations:
  - Hardware: FPGA
  - Software: C emulation of FPGA
  - Software: Object-oriented C++

Hard- and Software Architecture: Resource Estimation



- How do we fit in the resources left over?
- Choose between 3 R2U2 implementations:
  - Hardware: FPGA
  - Software: C emulation of FPGA
  - Software: Object-oriented C++

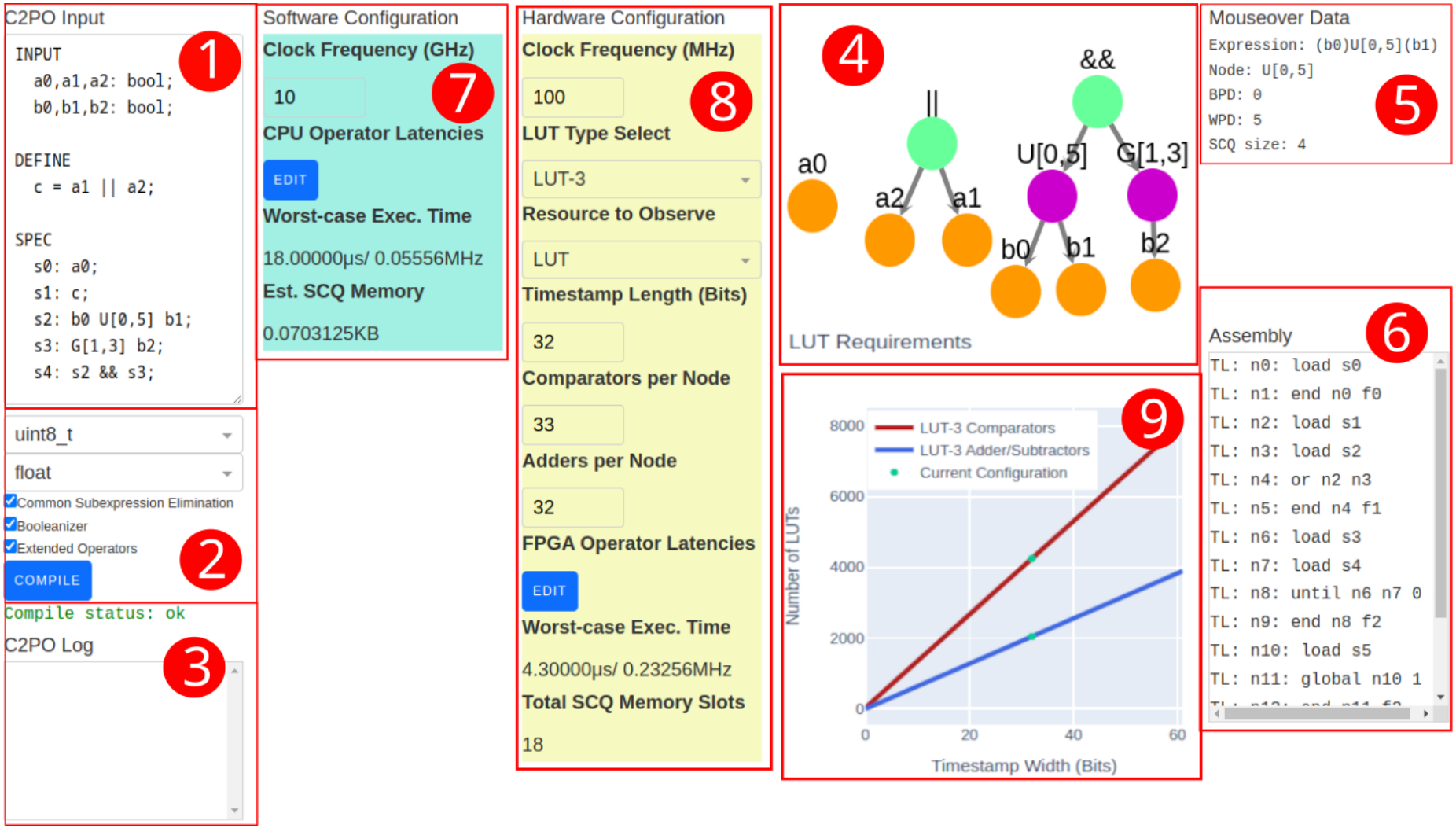


Figure: R2U2 Configuration Explorer web application: 1) C2PO specification input; 2) C2PO options; 3) C2PO output; 4) AST visualization; 5) AST node data; 6) R2U2 instruction; 7) C engine speed and memory calculator; 8) FPGA speed and size calculator; 9) FPGA design size vs maximum timestamp value.

Lifting Runtime Monitoring

Runtime Monitoring

[5] Ylies Falcone, Srdan Krstic, Giles Reger, and Dmitriy Traytel. "A taxonomy for classifying runtime verification tools." In International Conference on Runtime Verification, pp. 241-262. Springer, Cham, 2018.

Lifting Runtime Monitoring

Temporal Fault Disambiguation

Runtime Monitoring

[5] Ylies Falcone, Srdan Krstic, Giles Reger, and Dmitriy Traytel. "A taxonomy for classifying runtime verification tools." In International Conference on Runtime Verification, pp. 241-262. Springer, Cham, 2018.

Lifting Runtime Monitoring

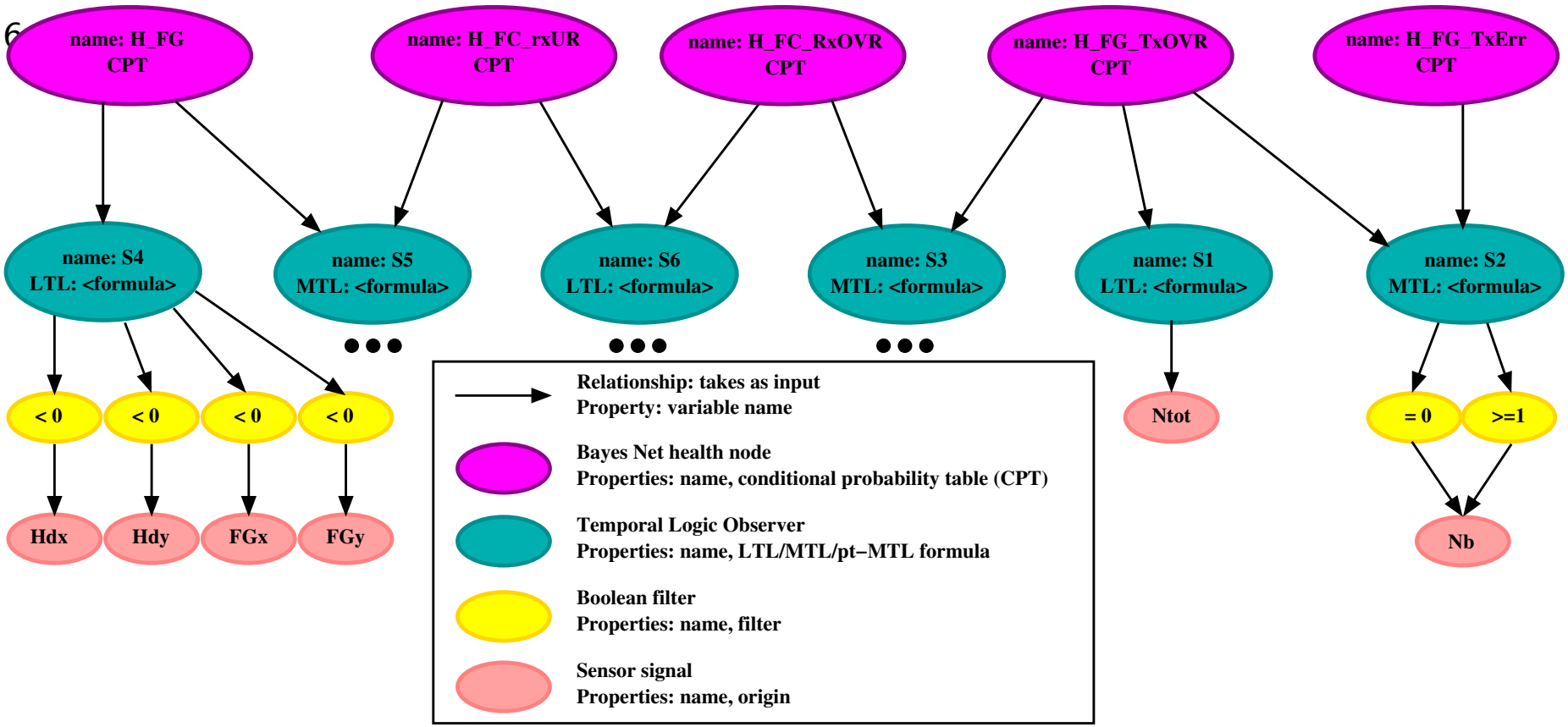
Temporal Fault Disambiguation

Runtime Monitoring

"R2U2 breaks our taxonomy; it is entirely application driven."
— Giles Reger, 11/13/2018 [5]

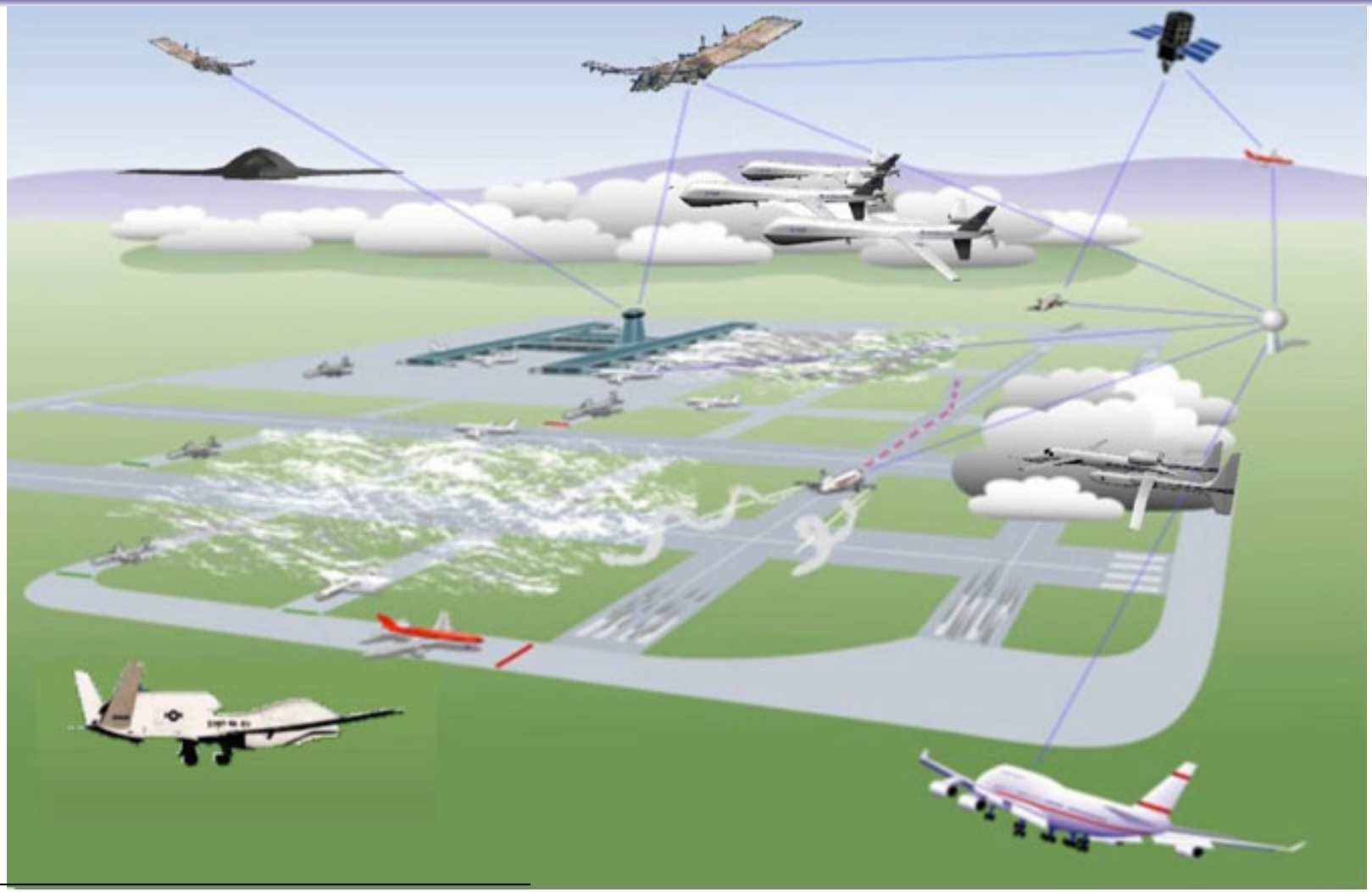
[5] Ylies Falcone, Srdan Krstic, Giles Reger, and Dmitriy Traytel. "A taxonomy for classifying runtime verification tools." In International Conference on Runtime Verification, pp. 241-262. Springer, Cham, 2018.

R2U2 Observation Tree (Specification)



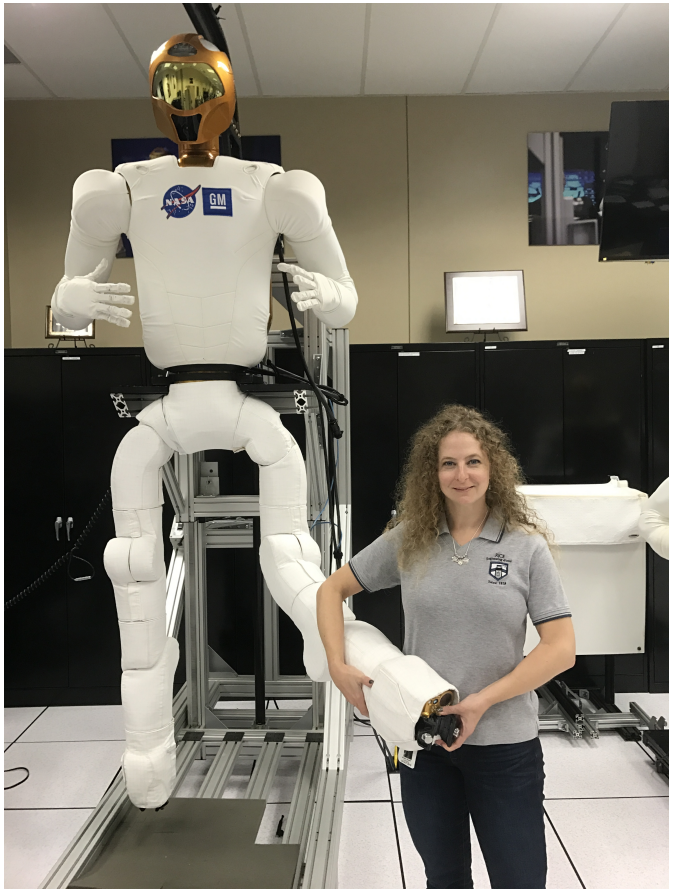
[6] Kristin Yvonne Rozier, and Johann Schumann. "R2U2: Tool Overview." In *International Workshop on Competitions, Usability, Benchmarks, Evaluation, and Standardisation for Runtime Verification Tools (RV-CUBES)*, held in conjunction with the 17th International Conference on Runtime Verification (RV 2017), Springer-Verlag, Seattle, Washington, USA, September 13–16, 2017.

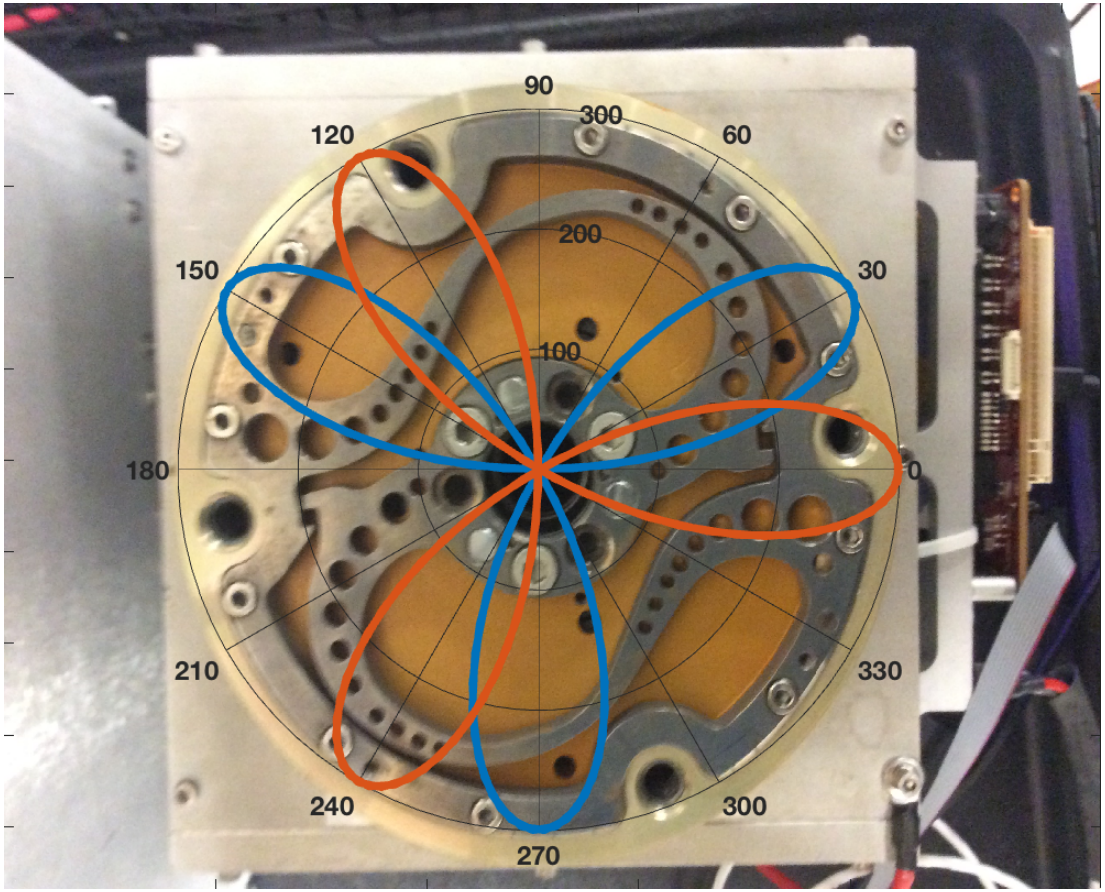
Adding UAS into the NAS?[7]



[7] Matthew Cauwels, Abigail Hammer, Benjamin Hertz, Phillip Jones, and Kristin Yvonne Rozier. "Integrating Runtime Verification into an Automated UAS Traffic Management System." *DETECT* 2020

Robonaut2's Knee[8]



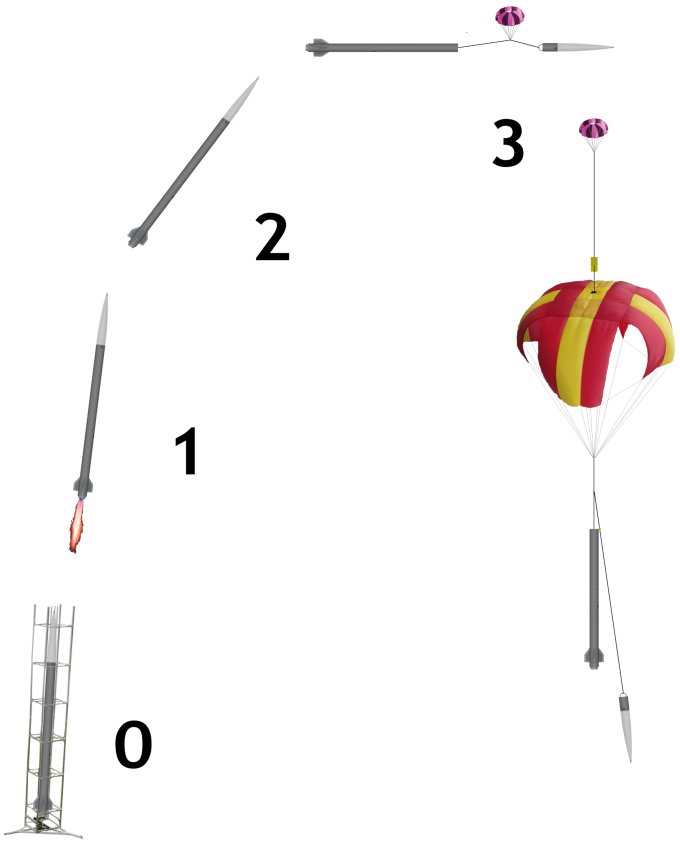


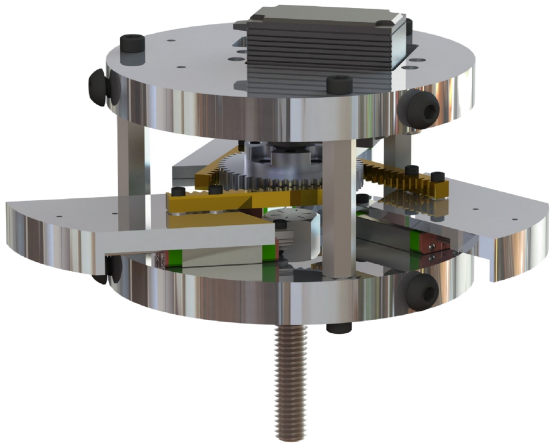
[8] Kempa, Brian, Pei Zhang, Phillip H. Jones, Joseph Zambreno, and Kristin Yvonne Rozier. "Embedding online runtime verification for fault disambiguation on Robonaut2." *FORMATS* 2020.

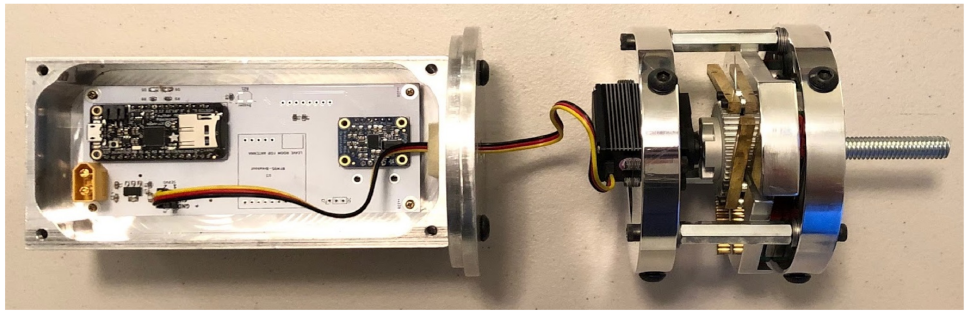
Robonaut2

http://temporallogic.org/research/R2U2/R2U2-on-R2_demo.mp4

Cyclone Rocketry's Sounding Rocket[9]





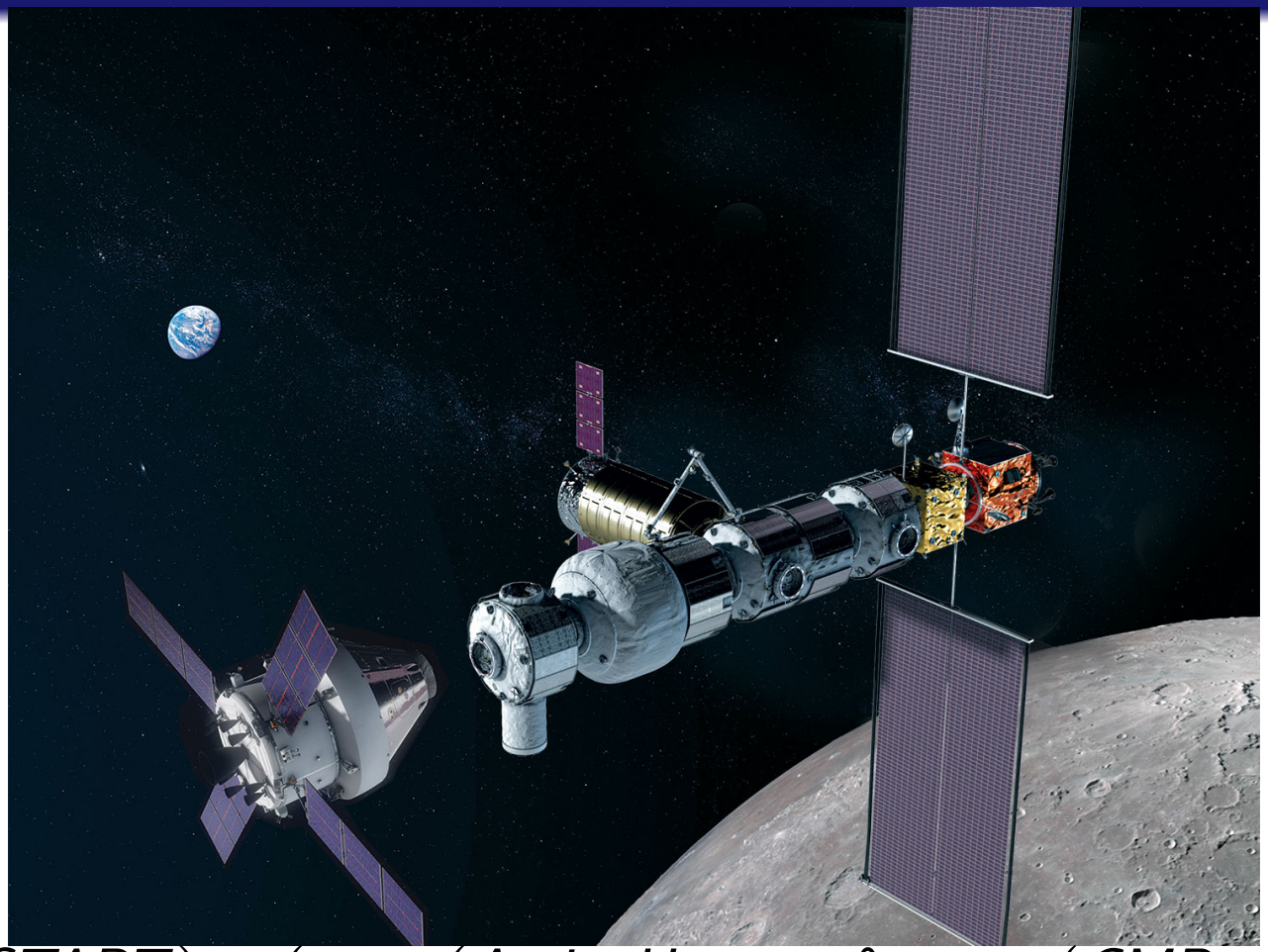


Left: Rocket mission states: *Launch Pad* (0), *Boost* (1), *Coast* (2), *Descent* (3). Right Top: Model of *Nova Somnium*'s ACS, Right Bottom: the physical ACS.

[9] B. Hertz, Z. Luppen, K.Y. Rozier. "Integrating Runtime Verification into a Sounding Rocket Control System." *NASA Formal Methods Symposium (NFM)*, 2021.

https://www.youtube.com/watch?v=p6dwT0sTdH0

NASA Lunar Gateway: Assume-Guarantee Contracts[10]



$$(CMD == START) \rightarrow (\square_{[0,5]}(ActionHappens \& \square_{[0,2]}(CMD = END)))$$

[10] Dabney, James B., Julia M. Badger, and Pavan Rajagopal. "Adding a Verification View for an Autonomous Real-Time System Architecture." In AIAA Scitech 2021 Forum, p. 0566. 2021.

Domain-Driven Adaptations

"Every file that gets opened eventually gets closed."

[11] C. Johannsen, B. Kempa, P. H. Jones, K. Y. Rozier, T. Wongpiromsarn. "Impossible Made Possible: Encoding Intractable Specifications via Implied Domain Constraints." FMICS, 2023.

Domain-Driven Adaptations

"Every file that gets opened eventually gets closed."

"At most X of these hold at the same time."

[11] C. Johannsen, B. Kempa, P. H. Jones, K. Y. Rozier, T. Wongpiromsarn. "Impossible Made Possible: Encoding Intractable Specifications via Implied Domain Constraints." FMICS, 2023.

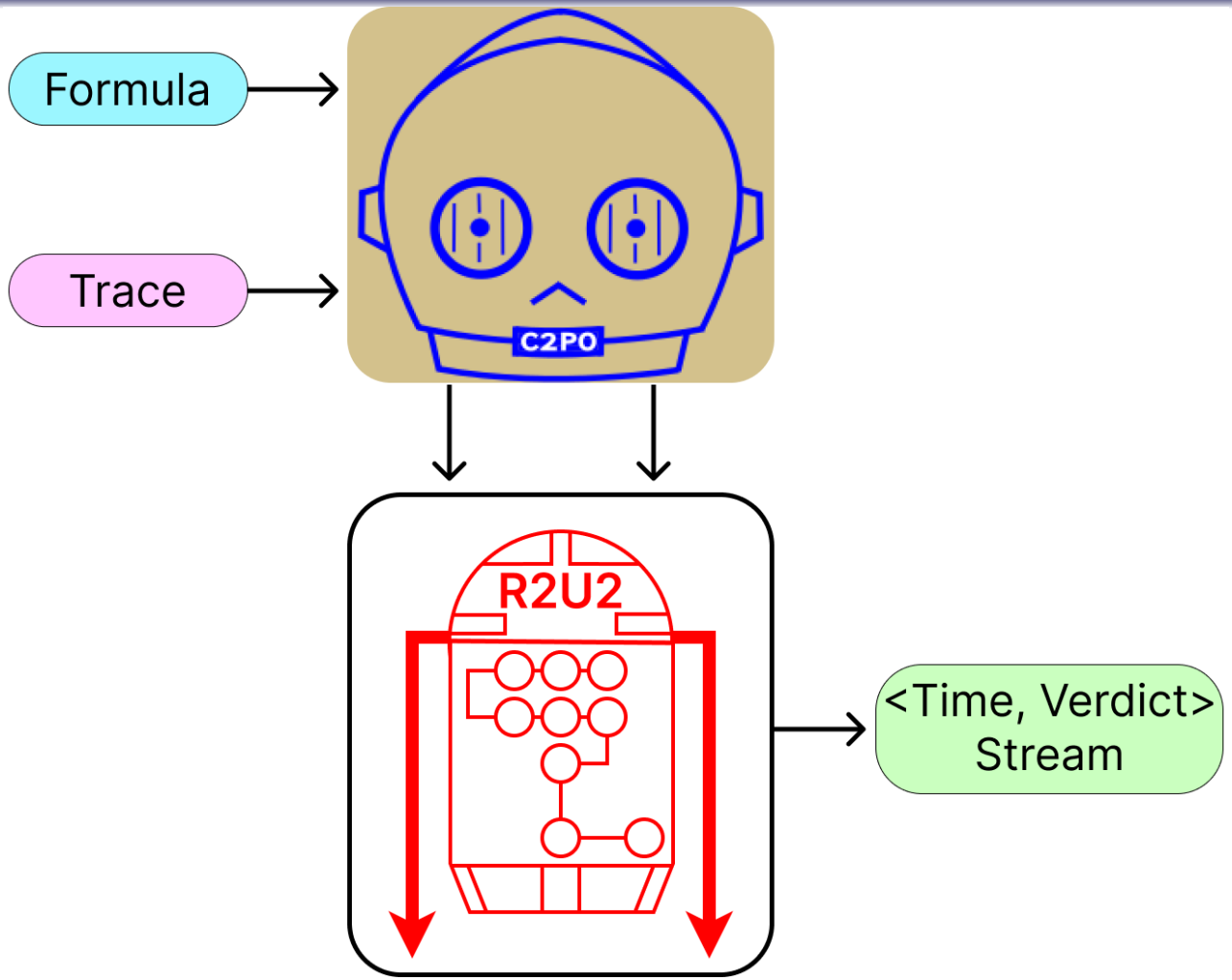
Domain-Driven Adaptations

"Every file that gets opened eventually gets closed."

"At most X of these hold at the same time."

Need a Configuration Compiler for Property Organization (C2PO)
11

[11] C. Johannsen, B. Kempa, P. H. Jones, K. Y. Rozier, T. Wongpiromsarn. "Impossible Made Possible: Encoding Intractable Specifications via Implied Domain Constraints." FMICS, 2023.

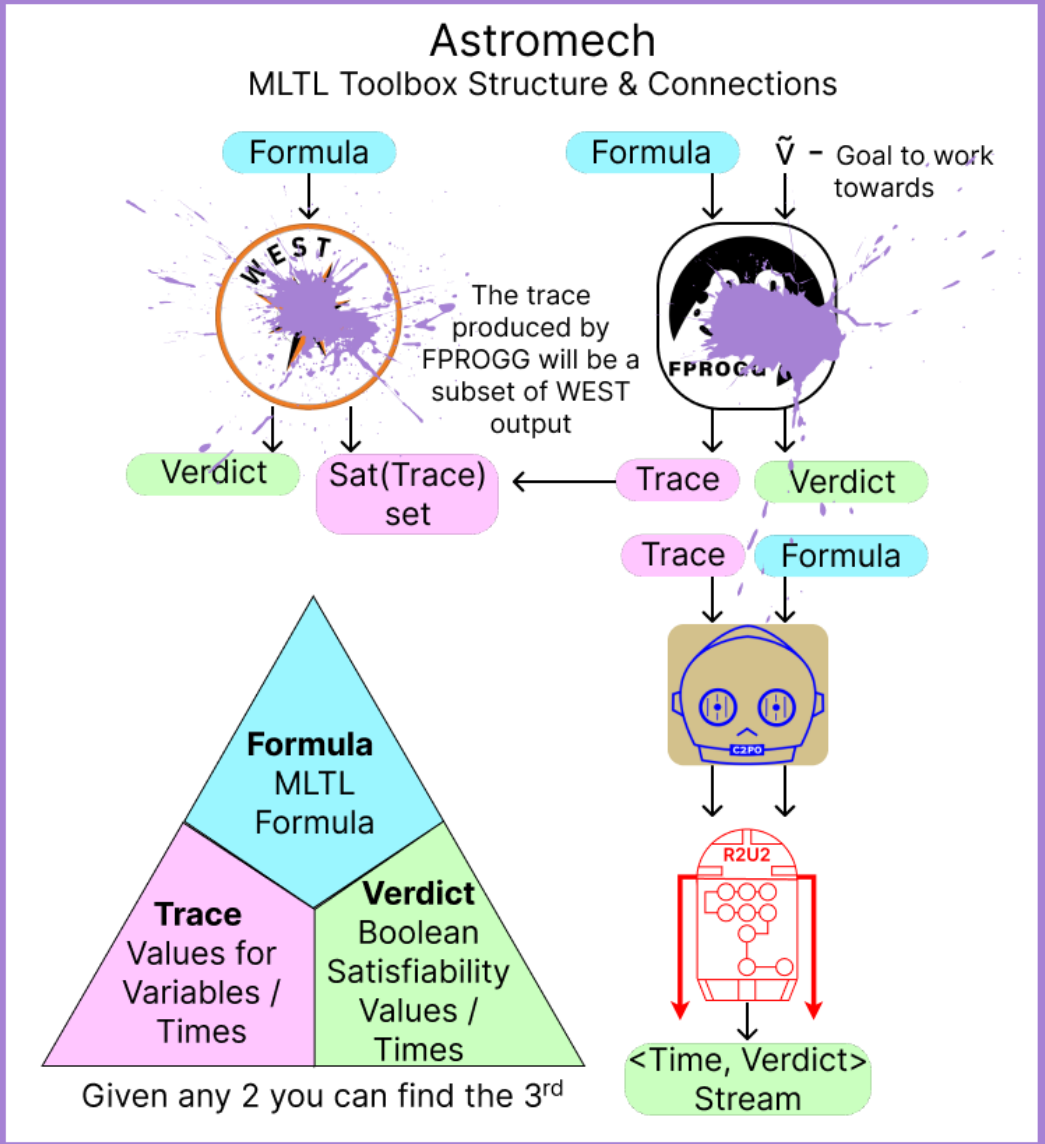


12 13

[12] C. Johannsen, P. H.Jones, B. Kempa, K. Y. Rozier, P. Zhang. "R2U2 Version 3.0: Re-imagining a Toolchain for Specification, Resource Estimation, and Optimized Observer Generation for Runtime Verification in Hardware and Software." CAV, 2023.

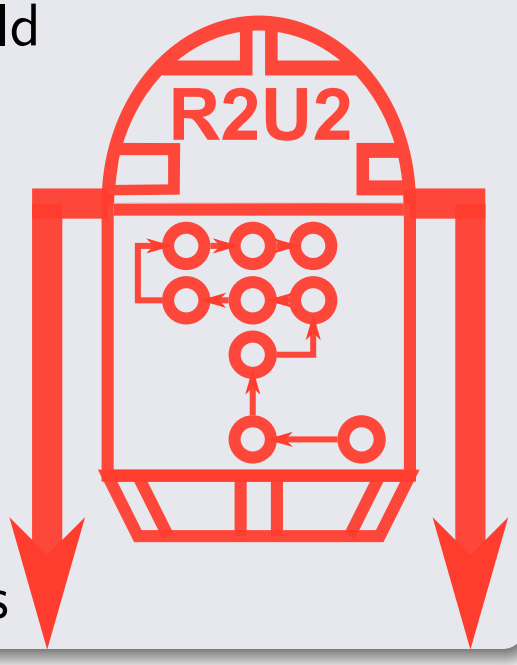
[13] C. Johannsen, B. Kempa, P. H. Jones, K. Y. Rozier, T. Wongpiromsarn. "Impossible Made Possible: Encoding Intractable Specifications via Implied Domain Constraints." FMICS, 2023.

Isabelle Theorem Proving



R2U2: Realizable Responsive Unobtrusive Unit

- Data Integrity: data is consistent, coherent, within expectations
- Sanity Checking: common-sense assumptions hold
- Fault Mitigation: real-time monitoring for fault signatures
- Security Monitoring: complex temporal patterns indicative of breaches
- Mission Integration: automatically catch mis-configured, or otherwise tenuous/faulty connections that elude system integration checks

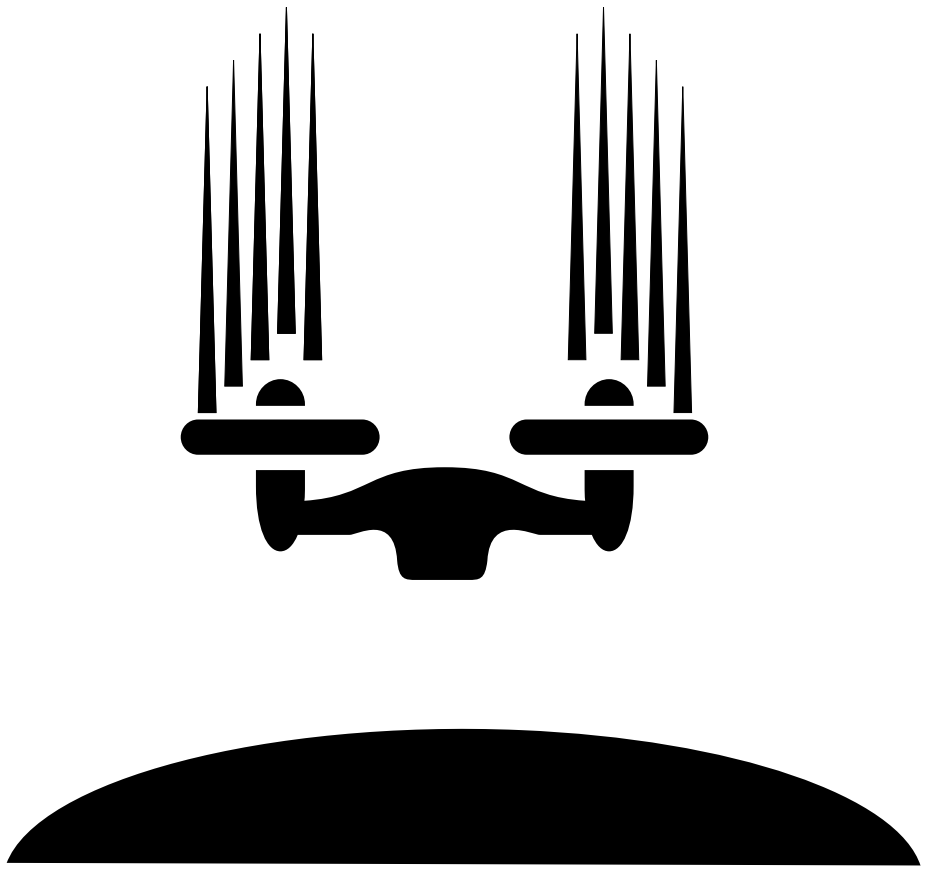


http://r2u2.temporallogic.org/

BACKUP SLIDES

Runtime Functional Specification Patterns[14]

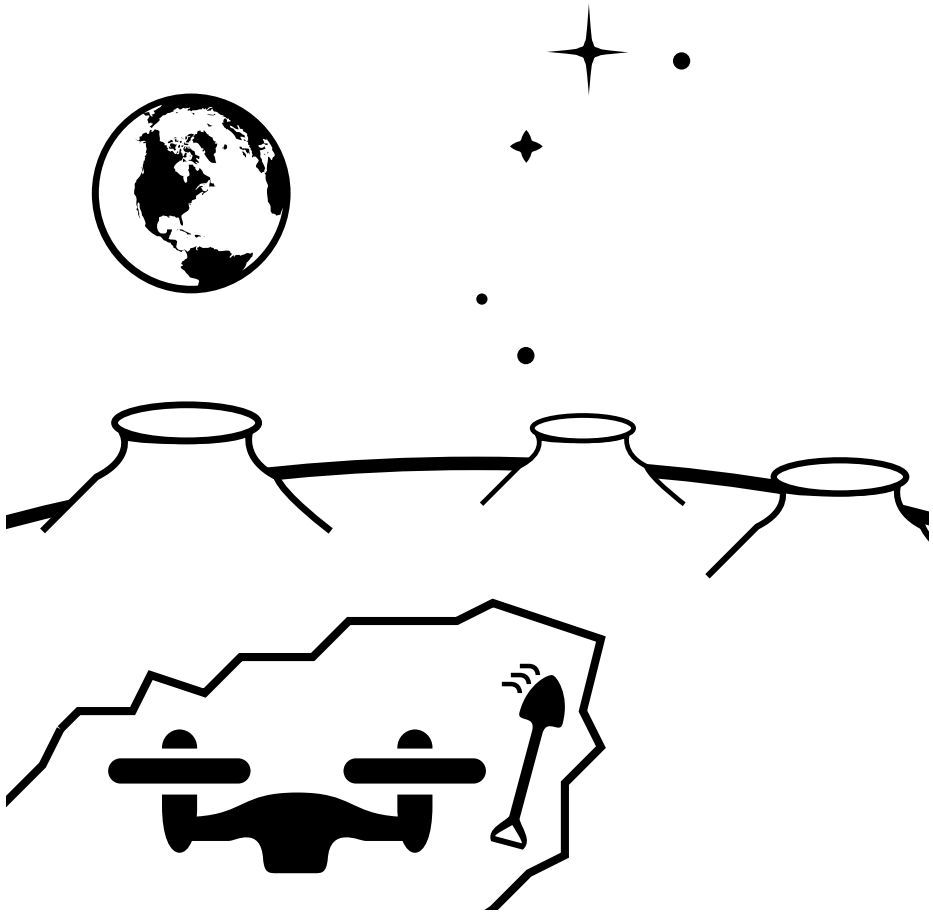
- **Rates**
- **Ranges**
- **Relationships**
- **Control Sequences**
- **Consistency Checks**



[14] K.Y.Rozier. "Specification: The Biggest Bottleneck in Formal Methods and Autonomy." VSTTE, 2016.

Runtime Functional Specification Patterns[14]

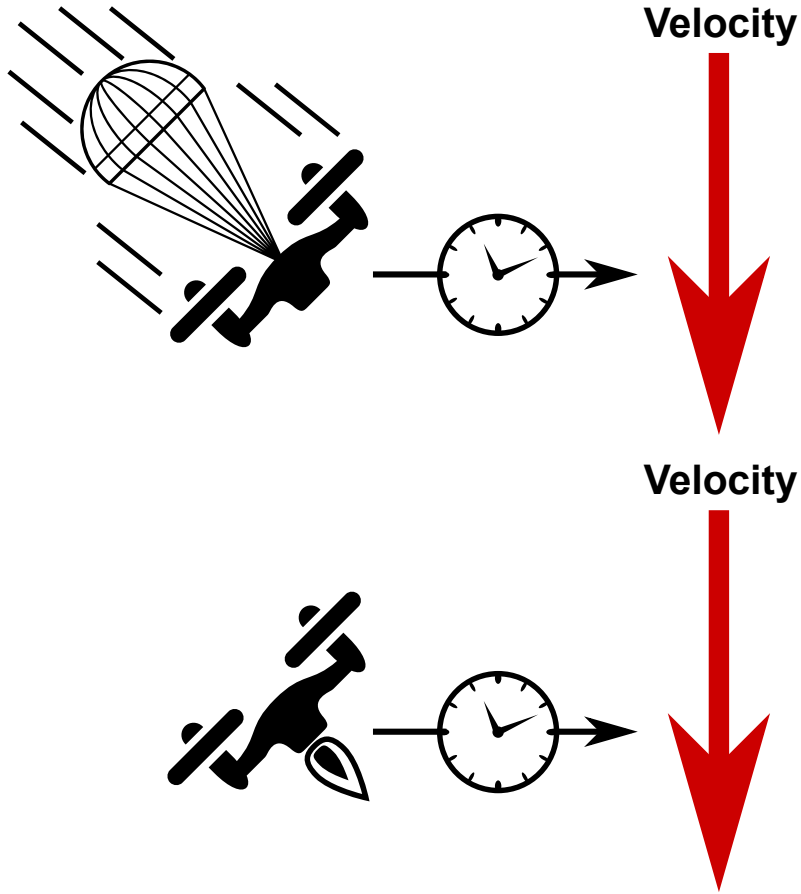
- **Rates**
- **Ranges**
- **Relationships**
- **Control Sequences**
- **Consistency Checks**



[14] K.Y.Rozier. "Specification: The Biggest Bottleneck in Formal Methods and Autonomy." VSTTE, 2016.

Runtime Functional Specification Patterns[14]

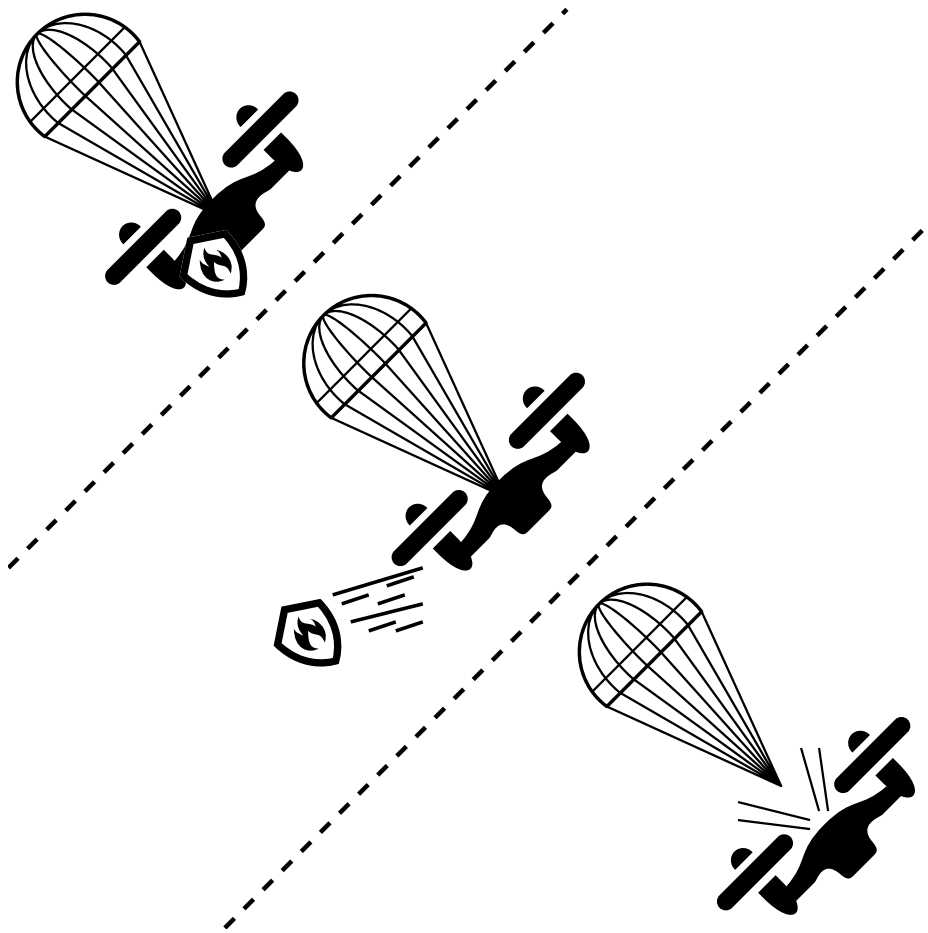
- Rates
- Ranges
- Relationships
- Control Sequences
- Consistency Checks



[14] K.Y.Rozier. "Specification: The Biggest Bottleneck in Formal Methods and Autonomy." VSTTE, 2016.

Runtime Functional Specification Patterns[14]

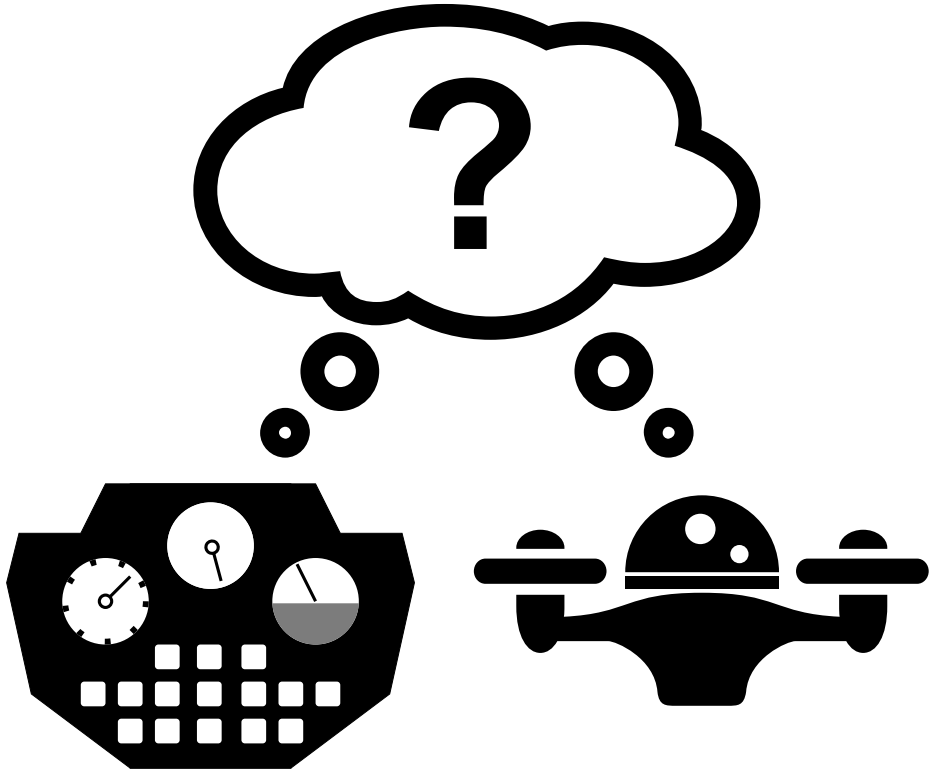
- **Rates**
- **Ranges**
- **Relationships**
- **Control Sequences**
- **Consistency Checks**



[14] K.Y.Rozier. "Specification: The Biggest Bottleneck in Formal Methods and Autonomy." VSTTE, 2016.

Runtime Functional Specification Patterns[14]

- **Rates**
- **Ranges**
- **Relationships**
- **Control Sequences**
- **Consistency Checks**



[14] K.Y.Rozier. "Specification: The Biggest Bottleneck in Formal Methods and Autonomy." VSTTE, 2016.