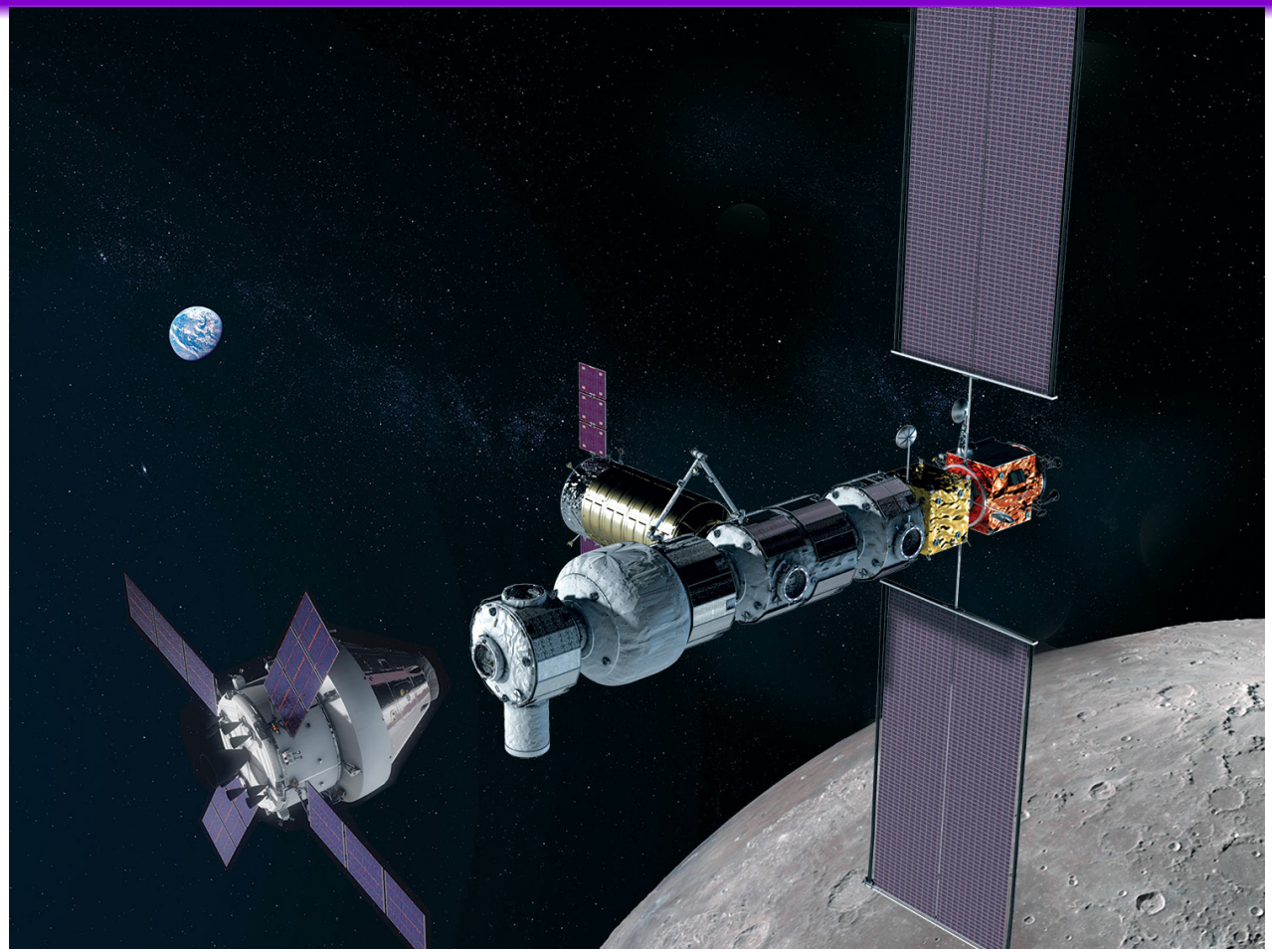
Improving Usability and Trust in Real-Time Verification of a Large-Scale Complex Safety-Critical System

Brian Kempa, Chris Johannsen, **Kristin Y. Rozier**
Iowa State University

26^{th} **Ada-Europe: Reliable Software Technologies**
June 16, 2022

NASA Lunar Gateway: Assume-Guarantee Contracts[1]



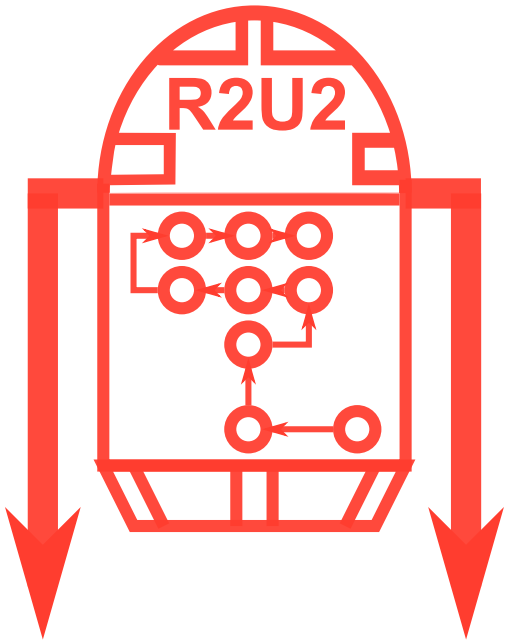
$$(CMD == START) \rightarrow (\Box_{[0,5]}(ActionHappens \& \Box_{[0,2]}(CMD = END)))$$

[1] 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.

Real-time, Flight-certifiable, Embedded Runtime Verification

RESPONSIVE
REALIZABLE
UNOBTRUSIVE
Unit

R2U2



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

Matching Input Specifications to Use-Cases

Old Syntax

```
a0 && ((a1 && !a2 && !a3) || // AGC:
       (!a1 && a2 && !a3) || // TRUE
       (!a1 && !a2 && a3));
!a0; // AGC: INACTIVE
a0 && !((a1 && !a2 && !a3) || // AGC:
        (!a1 && a2 && !a3) || // FALSE
        (!a1 && !a2 && a3));

a0 = bool(s0) == 1;
a1 = bool(s1) == 1;
a2 = bool(s2) == 1;
a3 = bool(s3) == 1;
```

New Syntax

```
RVALID: resRactive => resRvalid;

taskAactive = bool(Aactive) == 1;
taskBactive = bool(Bactive) == 1;
taskCactive = bool(Cactive) == 1;
resRactive = bool(Ractive) == 1;
resRvalid =
  exactly-one-of(active_tasks) == 1;

active_tasks = {taskAactive,
                taskBactive,
                taskCactive};
```

Example: Assume-Guarantee Output

Propositional Logic:

$False \rightarrow True \equiv True$

$True \rightarrow True \equiv True$

$True \rightarrow False \equiv False$

User-friendly:

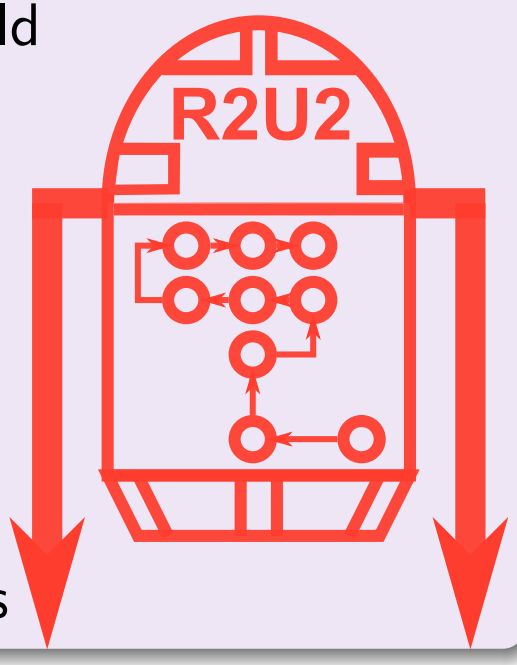
$(Assumption = False) \rightarrow (Guarantee = True) \equiv Inactive$

$(Assumption = True) \rightarrow (Guarantee = True) \equiv True$

$(Assumption = True) \rightarrow (Guarantee = False) \equiv False$

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/