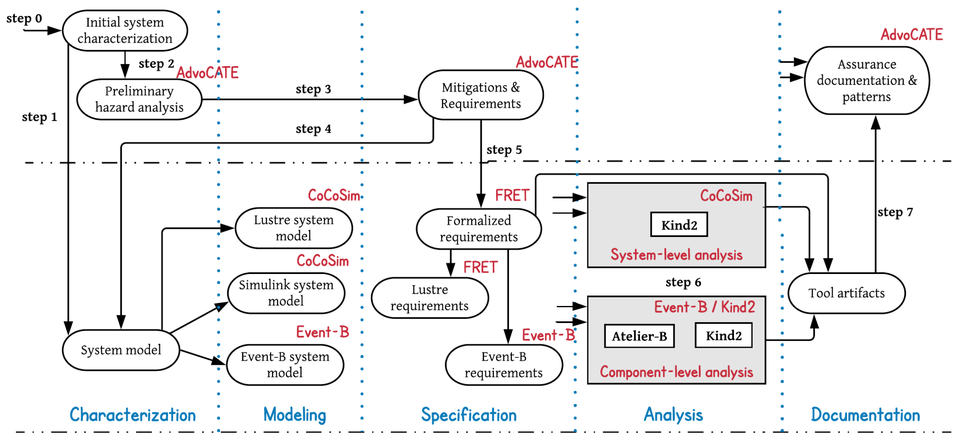
Integrating Formal Verification and Assurance: An Inspection Rover Case Study

Hamza Bourbouh **Marie Farrell** Anastasia Mavridou **Irfan Sljivo** Guillaume Brat Louise A. Dennis
Michael Fisher

Objective

- To study the **integration of formal verification results** via the **development of an assurance case**, as applied to a **robotic system,** using a tool palette that includes the three NASA Ames tools **FRET**, **CoCoSim**, and **AdvoCATE**, as well as **Event-B**.

Methodology



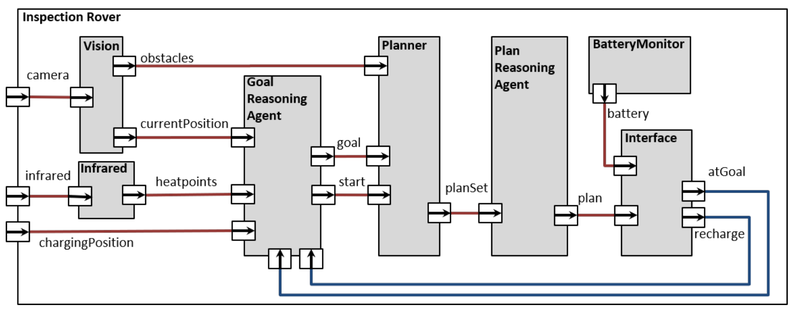
Step 0: Initial System Characterization

System: Autonomous rover undertaking an inspection mission.
System Context:

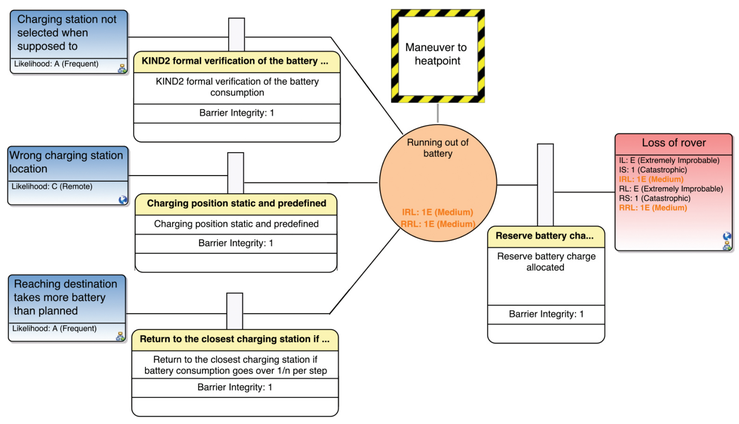
- The objective of the rover is to **explore a square grid** of known size.
- **Autonomously navigate** to points of interest whilst **avoiding obstacles** and **recharging** when necessary.
- The system operates **indoors**.

Our Focus: The navigation system of the rover.

Step 1: Create Initial System Model



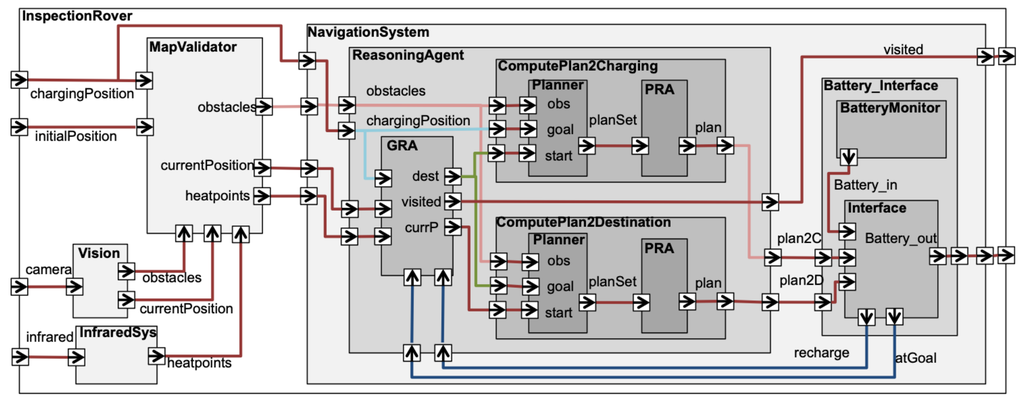
Step 2: Perform Preliminary Hazard Analysis



Step 3: Define Mitigations and Safety Requirements

- Defining **mitigations for the different hazards** in order to **minimize the risk** of those hazards and their consequences.
- **System-level requirements**:
  - R1: The rover shall not run out of battery
  - R2: The rover shall not collide with an obstacle
  - R3: The rover shall visit all reachable heat points

Step 4: Refine System Model According to Mitigations



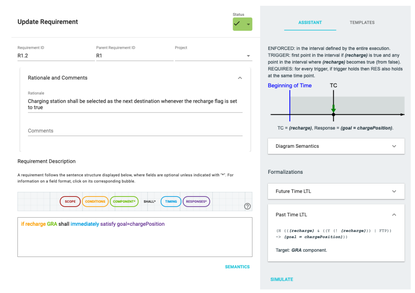
Step 5: Formalize Requirements and Create Specifications

Formal Requirements Elicitation Tool (FRET)

- FRET bridged the gap between the informal and formal steps.

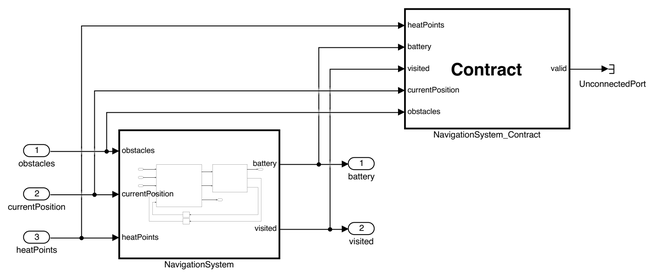
[R1]: Navigation shall always satisfy battery $>$ 0

[R3.3]: GRA shall always satisfy if ! recharge then (if forAll i & i_inGrid then (if ! visited[i] then heatpoints[goal] $>=$ heatpoints[i]))



Step 6: Verification at System- and Component-Levels

Compositional Verification with CoCoSim

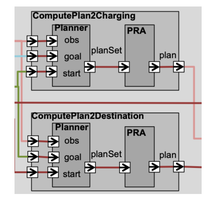


number of cells	[R1]	[R3]
9	0.487 seconds	0.487 seconds
16	2.395 seconds	9.08 hours
25	4.298 seconds	Timeout ($>$ 2 days)

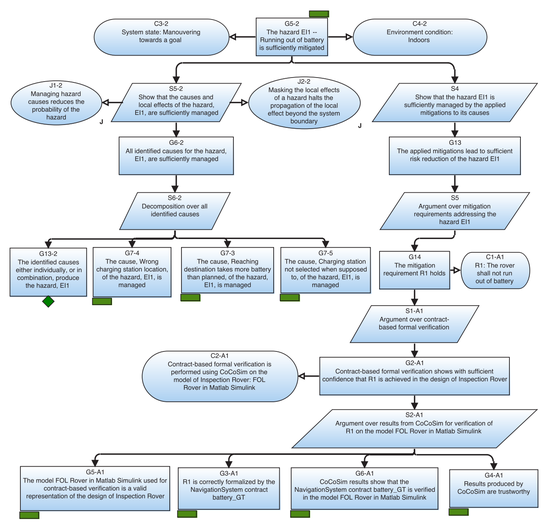
```
guarantee "R3.3" not recharge
=> (forall (i:int) (0 <=
i and i < width) => (not
visited[i] => heatpoints[goal]
>= heatpoints[i] )));
```

Component-Level Verification with Event-B

Event PRA $\widehat{=}$ **ordinary**
any
p
when
grd1: $p \in PlanSet$ **not theorem**
grd2: $\forall\ q \cdot q \in PlanSet \Rightarrow card(p) \leq card(q)$ **not theorem**
grd3: $planningCompleted = TRUE$ **not theorem**
then
act1: $chosenplan := p$
act2: $returnplan := TRUE$



Step 7: Document Verification Results and Build Safety Case in AdvoCATE



Our Paper

SCAN ME

Bourbouh, H., Farrell, M., Mavridou, A., Sljivo, I., Brat, G., Dennis, L. A., & Fisher, M. (2021). Integrating Formal Verification and Assurance: An Inspection Rover Case Study. In *NASA Formal Methods Symposium* (pp. 53-71). Springer.

Work supported by NASA ARMD System-Wide Safety Project, UK Research and Innovation and EPSRC Hubs for “Robotics and AI in Hazardous Environments”: EP/R026092 (FAIR-SPACE), and the Royal Academy of Engineering.