



UC5 – Aircraft Engine Controller

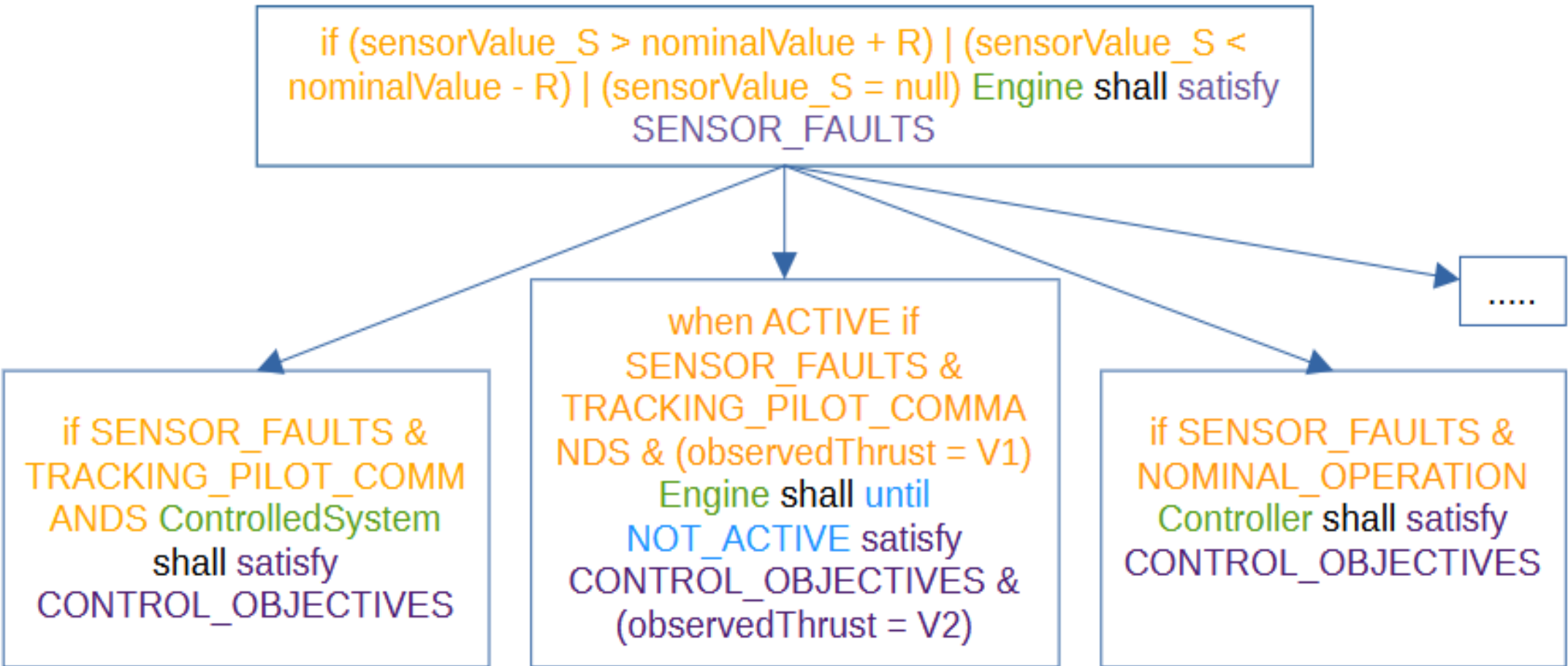


Refactoring Requirements and MU-FRET's place in the VeRFoR Toolchain

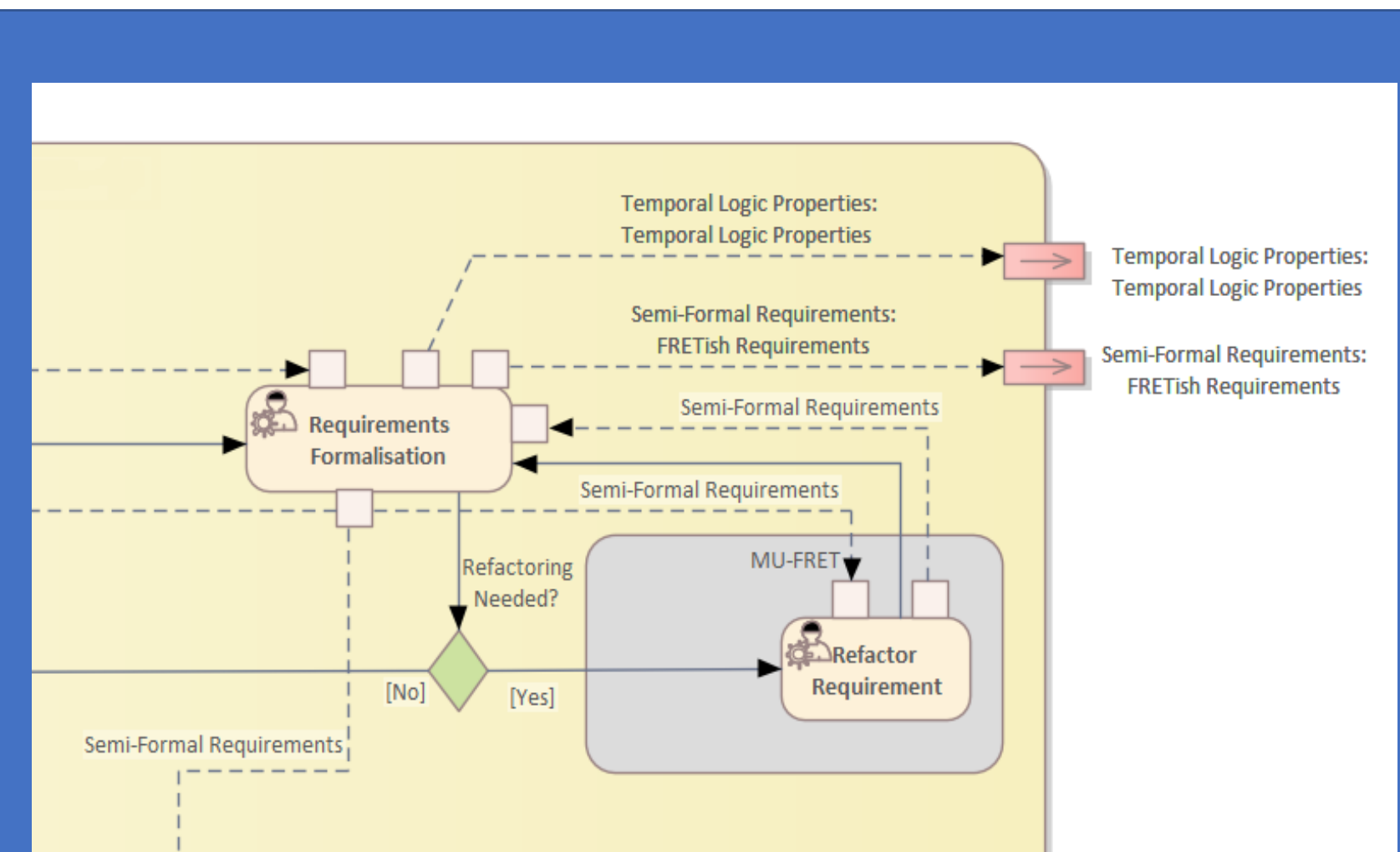
Cathal Peelo, Shi Hao To,
Oisín Sheridan,
Rosemary Monahan,
Maynooth University,
Ireland

Refactoring

While FRET provides for hierarchical relationships between system requirements, these "parent-child" relationships do not imply any functional interaction or dependency between requirements. By taking phrases which multiple requirements restate separately and refactoring them out as separate requirements, MU-FRET allows users to apply a more functional hierarchy to their projects. It also helps them state their requirements more concisely, while giving them semantic names increases human-readability. Altogether, this furthers the goal of representing requirements in a way that is more comprehensible across disciplinary boundaries. The diagram to the right illustrates the benefit of making a separate requirement (top) out of a string common to numerous existing requirements (below).



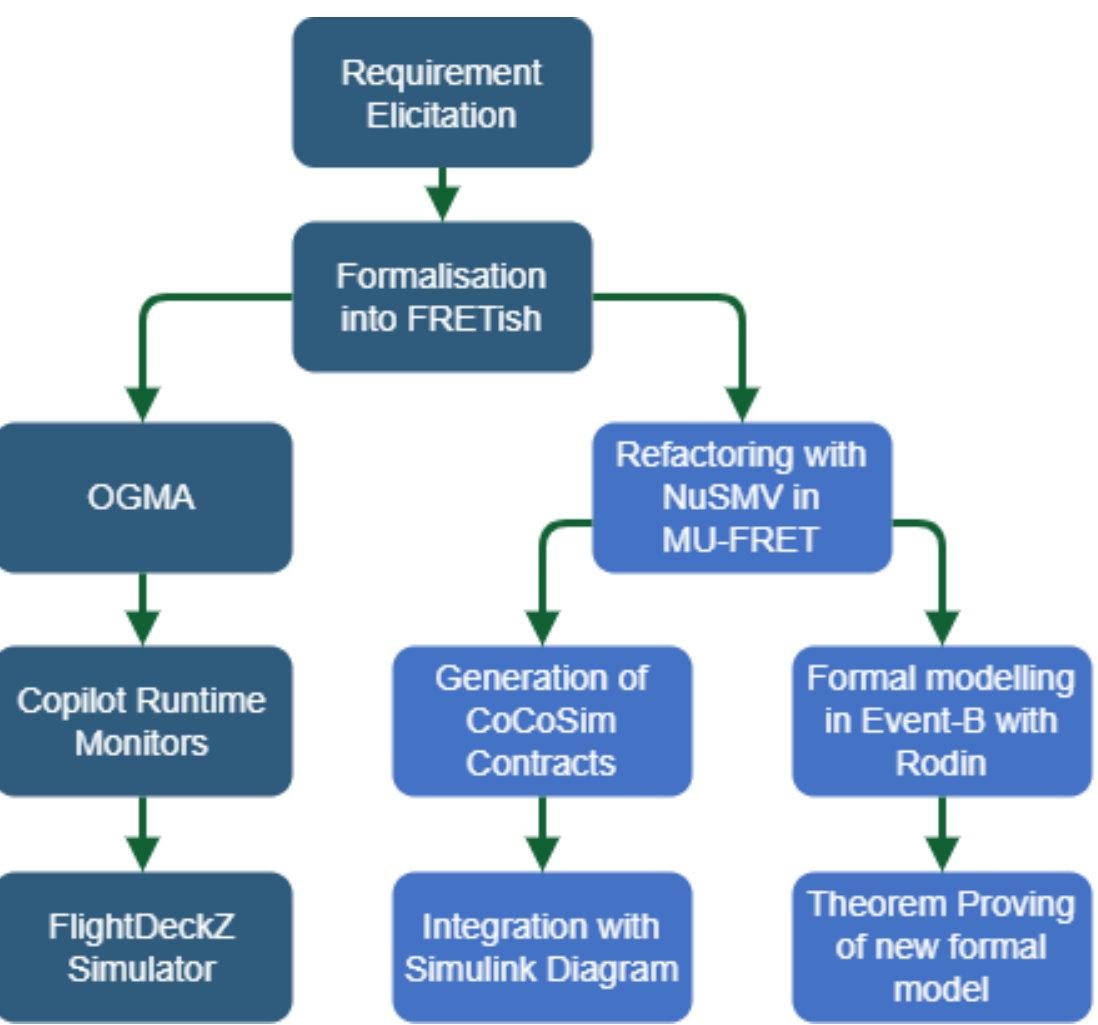
MU-FRET's Place within the VeRFoR Toolchain



<https://repo.valu3s.eu/use-cases/aircraft-engine-controller/workflow/verfor>

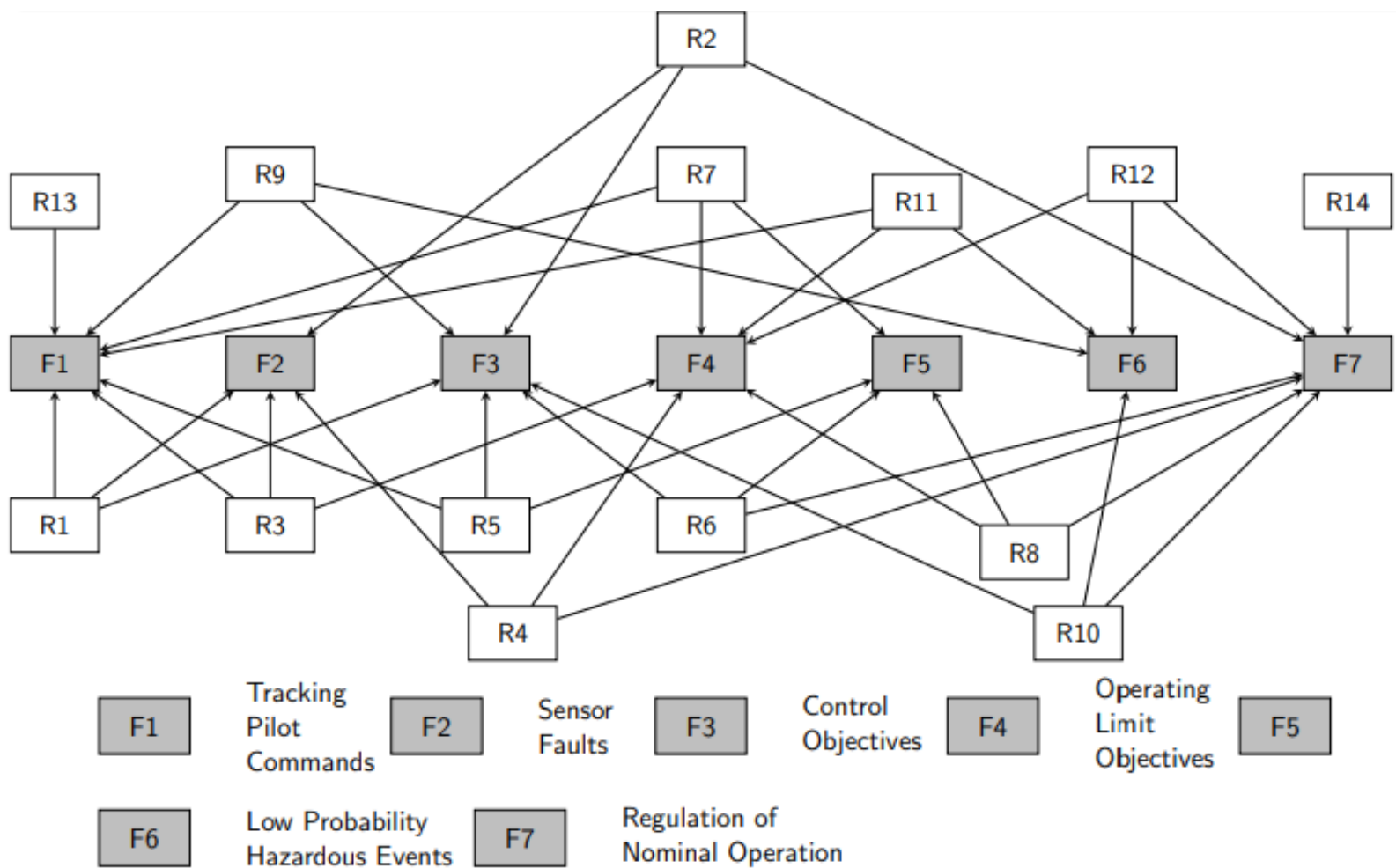
- The VeRFoR toolchain brings requirements from the vagaries of elicitation in natural language through to formal verification.
- Requirements are formalised into the structured natural language FRETISH, encouraging refinement of ambiguities.
- These then allow generation of CoCoSim contracts, which are used to formally verify a Simulink model of the system.
- MU-FRET provides a GUI for much of VeRFoR, allowing refactoring with the aid of NuSMV.
- The new functionality of the Mu-FRET project enables a rationalised and more easily read set of requirements.

MU-FRET on GitHub:



Origins and Expansion

- NASA developed the FRET tool to allow engineers with varying levels of experience in formal methods to contribute to V&V processes.
- As part of the VALU3S project, the MU-FRET fork has been developed. It has extended FRET's functionality, most notably with the addition of "extract requirement" refactoring.
- An industrial use case has been documented with a toolchain differing from that used by NASA and featuring two independent and complementary flows to allow for redundant verification.



Industrial Feedback

- FRETish requirements were 'much more clear' than the natural-language requirements.
- FRET was 'very easy to use; interface is intuitive'.
- Useful 'because it forces you to think about the actual meaning behind the natural-language requirements'.
- Issues with the GUI were highlighted. Some of these we have improved as part of VALU3S, others remain the focus of planned projects at MU

Planned Developments

- Adding inline requirement refactoring (opposite of extract requirement)
- Adding (deep) rename requirement refactoring
- Adding move definition refactoring

Involved VALU3S Partners

Leader:



Participating Partners:



VALU3S HAS RECEIVED FUNDING FROM THE ECSEL JOINT UNDERTAKING UNDER GRANT AGREEMENT NO 876852. THE JOINT UNDERTAKING RECEIVES SUPPORT FROM THE EUROPEAN UNION'S HORIZON 2020 RESEARCH AND INNOVATION PROGRAMME AND AUSTRIA, CZECH REPUBLIC, GERMANY, IRELAND, ITALY, PORTUGAL, SPAIN, SWEDEN, TURKEY.

