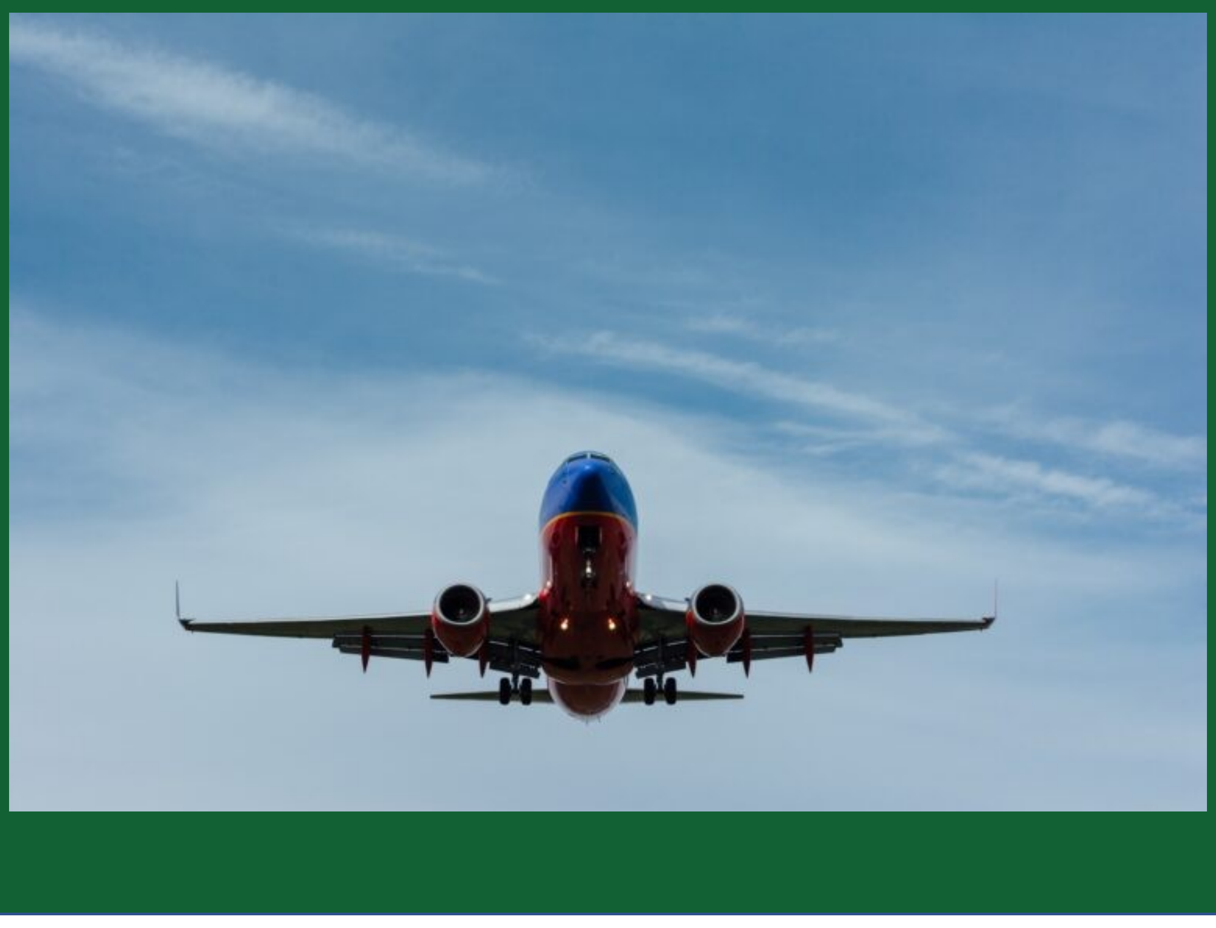
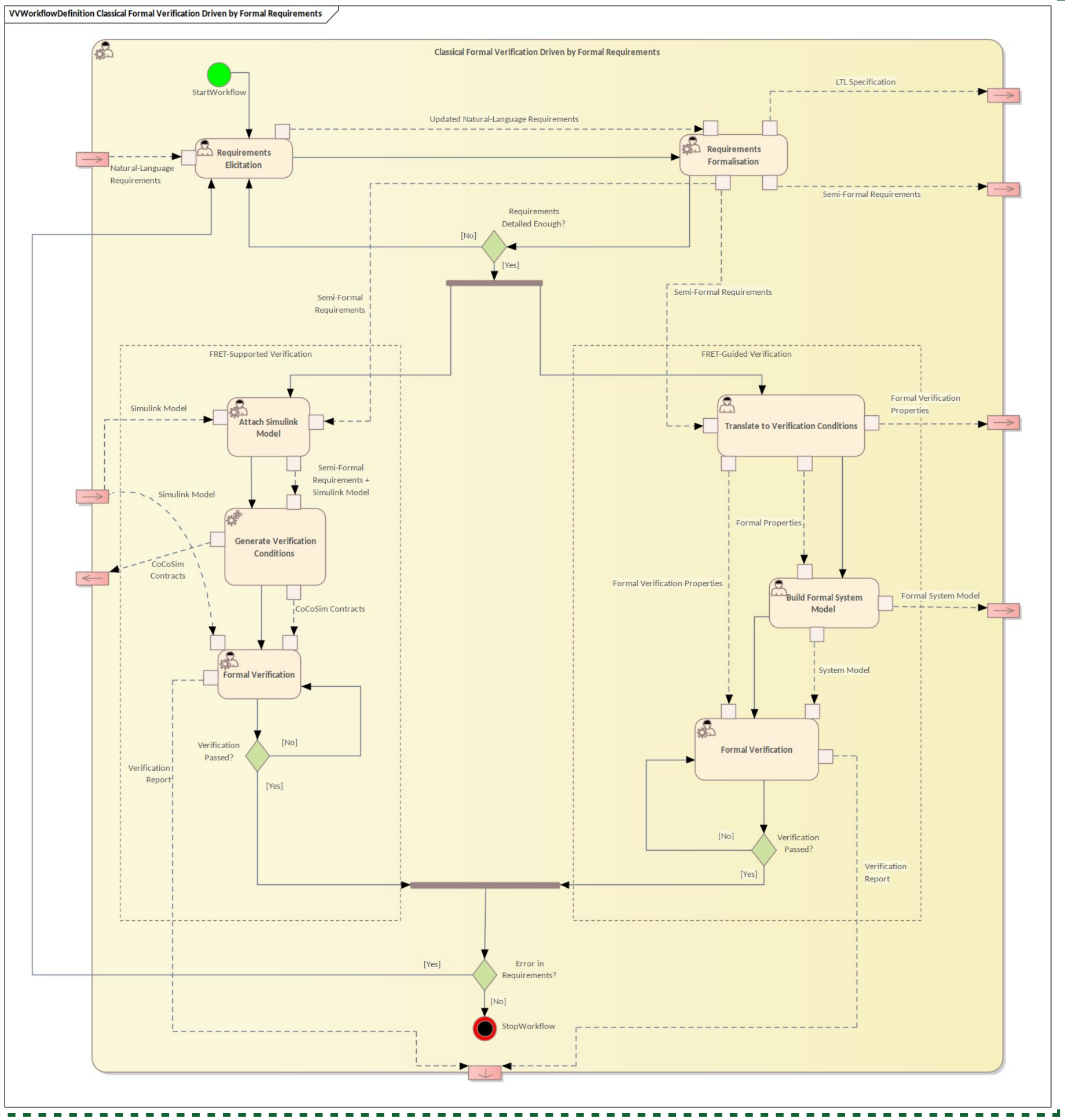


VALU3S

Classical Formal Verification Driven by Formal Requirements

Combined Method


VVWorkflowDefinition Classical Formal Verification Driven by Formal Requirements
Classical Formal Verification Driven by Formal Requirements
StartWorkflow
Natural-Language Requirements
Requirements Elicitation
Updated Natural-Language Requirements
Requirements Formalisation
LTL Specification
Semi-Formal Requirements
Requirements Detailed Enough?
[No]
[Yes]
Semi-Formal Requirements
Semi-Formal Requirements
FRET-Supported Verification
Simulink Model
Attach Simulink Model
Semi-Formal Requirements + Simulink Model
Simulink Model
Generate Verification Conditions
CoCoSim Contracts
CoCoSim Contracts
Formal Verification
Verification Passed?
[No]
[Yes]
Verification Report
FRET-Guided Verification
Translate to Verification Conditions
Formal Verification Properties
Formal Properties
Formal Verification Properties
Build Formal System Model
Formal System Model
System Model
Formal Verification
Verification Passed?
[No]
[Yes]
Verification Report
Error in Requirements?
[Yes]
[No]
StopWorkflow

Base Method Description

Requirements are often expressed in natural language, and often at a level of detail that is not suitable for direct formalisation [SFM1]. A semi-formal language may be used as an intermediate between natural- and formal-languages, which avoids slowing down the requirements elicitation process, but still provides enough formalisation to both reduce ambiguities and make the requirements easier to (fully) formalise later on [SFM2].

Improved/Combined Method

One common criticism of formal methods is that they are too abstract and too far removed from realistic models used for designs and simulations. Also, building a formal specification is still the main bottleneck in using formal methods [SFM1]. We intend to improve selected tool chains for model checking and theorem proving approaches by linking their models more closely with the associated Simulink models. We will also improve the learning curve improved via better integration and inclusion of more user-friendly tools such as FRET.

Layers of the multi-dimensional framework

Evaluation Environment Type	Evaluation Type	Type of Component	Evaluation Stage	Purpose of Component	Type of Requirement	Evaluation Performance Indicator
• **In-the-lab** • Closed • Open	• Experimental – Testing • Experimental – Monitoring • Experimental – Simulation • **Analytical – Formal** • Analytical – Semi-Formal	• **Model** • **Software** • Hardware	• Concept • **Requirement Analysis** • **System Design** • **Architecture Design** • **Detail Design** • Implementation • **Unit Testing** • **Integration Testing** • **System Testing** • Acceptance Testing • **Operation** • Risk Analysis • Other	• **Sensing** • **Thinking** • **Acting** • **Other**	• **Non-functional – Safety** • Non-functional – Security • Non-functional – Privacy • Non-functional – Other • **Function**	• V&V Process criteria • Time of test execution • Number of test cases • Joint Management of SCP Requirements • Effort needed for test • Reduced cost and time for work on certification process and functional safety • SCP criteria • Error coverage • Number of safety/security requirement violations • Number of malicious attacks and faults detected

Gaps & Limitations Addressed

- GAPM-DEV02, GAPM-MCH02, GAPM-TPS02, GAPM-FRV02: Accuracy
- GAPM-DEV04, GAPM-FRV04: Deployment
- GAPM-DEV06: Costs
- GAPM-FRV01, GAPM-TPS01: Functionality
- GAPM-DEV05, GAPM-FRV05, GAPM-MCH05, GAPM-TPS04, GAPM-TPS05: Learning Curve

Foreseen Impacts

Formalised requirements are easier to input/translate into languages used by formal methods. The act of formalising the requirements highlights ambiguities that may cause problems later in the development process.

Connection to VALU3S Use Cases

- UC5: Aircraft Engine Controller

Connection to V&V Tools

- FRET (https://github.com/NASA-SW-VnV/fret)

References

- [SFM1] Rozier, Kristin Yvonne, "Specification: The Biggest Bottleneck in Formal Methods and Autonomy" (2016). Verified Software. Theories, Tools, and Experiments (pp. 8-26). Springer. https://doi.org/10.1007/978-3-319-48869-1_2 ·
- [SFM2] Giannakopoulou, D., Pressburger, T., Mavridou, A., & Schumann, J. (2020). Generation of formal requirements from structured natural language. In International Working Conference on Requirements Engineering: Foundation for Software Quality (pp. 19-35). Springer. https://doi.org/10.1007/978-3-030-44429-7_2

Involved VALU3S Partners

Maynooth University
National University of Ireland Maynooth

ENTERPRISE IRELAND

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.

ECSEL Joint Undertaking
Electronic Components and Systems for European Leadership