Integration of the NASA tool FRET into Polarion & MATLAB/Simulink Workflows

Combine the features of NASA’s Formal Requirements Elicitation Tool (FRET) with the full-grown application life-cycle management platform Polarion and the Model-Based Design tool suite MATLAB/Simulink.

Motivation:

We at the Institute of Flight System Dynamics are using the application life-cycle management platform Polarion by Siemens Digital Industry Software to manage our requirements for various research projects as well as MATLAB/Simulink by The MathWorks for model-based design.

Unfortunately, Polarion does not provide any wording constraints or formalization and analysis features for text out of the box. NASA’s Formal Requirements Elicitation Tool (FRET) is a framework that provides such features. Requirements formulated in FRET need to follow a structured natural language (FRETish) and can then be translated into formal mathematical logics, diagrams and can be visualized in interactive simulations.

Instead of using FRET’s own database to manage the requirements, we want to store the requirement text and the mathematical logics directly in Polarion to benefit from the lifecycle and traceability features of the platform. It will be your task to establish a communication interface between Polarion (runs on a server) and FRET (runs locally on client computers as Electron JavaScript app). The NASA team behind FRET will happily help you with any FRET related questions you might have.

Now that the requirement data is stored in Polarion, it is accessible by other tools. In the second part of your assignment, you will create a dynamic subsystem block in Simulink that loads the information of a requirement from Polarion and, based on its mathematical logics, automatically transforms into an observer block for this requirement. This subsystem can then be used for model-based validation of the requirements and respectively verification of the implementation.
Work packages:

- WP1 Familiarize yourself with FRET’s command line features and code
- WP2 Implement the communication interface between Polarion and FRET
- WP3 Familiarize yourself with Simulink model block customization
- WP4 Implement the data retrieval from Polarion and the update mechanism
- WP5 Implement the dynamic Simulink subsystem block
- WP6 Document the Polarion/FRET interface and the Simulink subsystem block
- WP7 Demonstrate the functionality using examplary requirements

Lecture for IDP (not for semester thesis):

The lecture Safety and Certification of Avionics and Flight Control Systems (5 ECTS including exercise) addresses the certification process of avionics and flight control systems in commercial aviation. The lecture begins with a general overview of the development and certification of flight control systems, along with the content of relevant development standards and recommended practices and the resulting process structure including requirements management. Based on this, profound knowledge of the process and methods of safety assessment of complex technical systems in aircraft is conveyed. The goal of the lecture is to convey the capabilities of structuring the development process of a complex, technical system according to the relevant regulations and standards and developing safety proofs for certification in a mathematically founded and methodically sound way.

The project work helps the developers not only with requirement management by introducing a structured natural language but also automatically transforms the requirements into correct models for model-based validation and verification.

Your Profile:

- Interest in software development, model-based design, development processes and formal methods
- Very good skills in JavaScript and web development tools
- Advanced skills in MATLAB/Simulink
- Optional: Experience with the Electron framework

Start: Immediately