Automated Test Generation Techniques for Systems Engineering Tools

Imad Dodin, Nada Marawan, Muhammad Huzaifa Elahi, Ahmed Elehwany
DP 46, Professor Daniel Varro, McGill University

1 - Overview

- Systems Engineering tools validation is a critical process in Model Based Systems Engineering (**MBSE**).
- Manual Efforts to certify these tools lack in input diversity and are time and labour costly.
- Automated Model Generation methods (e.g. Viatra Generator) are shown to significantly reduce required effort and increase test diversity.
- There is a significant lack of open-source Systems Models for use by the research community.
- Systems Engineering tools like Capella are demonstrated within the context of the design of Cyber-Physical Systems (here an Aircraft Flight Control System)

2- Capella

Capella is a MBSE tool to design system, software, and hardware architecture. Capella helps systems architects to formalize specification and master architectural design. Capella models consist of several layers, following a top-bottom approach.

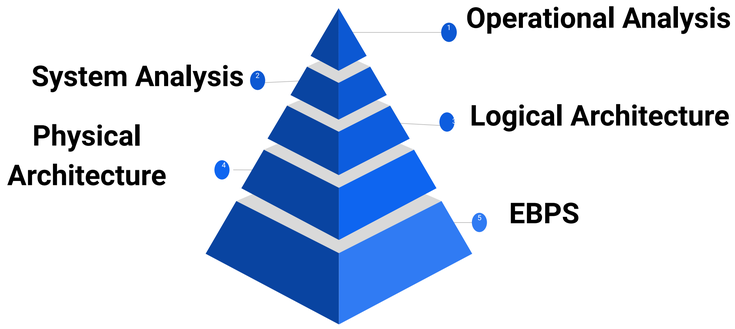

Figure 1: Capella Layers, abstraction decreases down the pyramid

Operational Analysis: Relationships b/w actors & stakeholders
System Analysis: Assign actions & responsibilities to actors
Logical Architecture: Assign the system its expected functions
Physical Architecture: Explain how system operates physically
EBPS: Component level breakdown of the entire system

Capella Implements "**Arcadia**": a method based on the use of models to define user and system needs, design and validate system requirements and flesh out software and hardware architectures through layered approach.

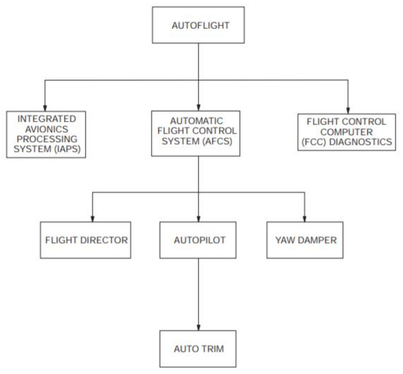

Figure 2. AFCS Architecture

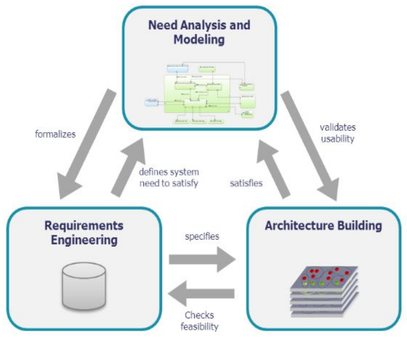

Figure 3. Arcadia Method

3 - Traceability & Fault Tolerance

The Capella layers are linked together using **Traceability**: An entity created in one layer is linked to functions in other layers, keeping the model concise up to date.
To demonstrate fault tolerance, we created feedback loops. By polling data from sensors and using it to affect subsequent readings we can smooth out outliers and create a robust system

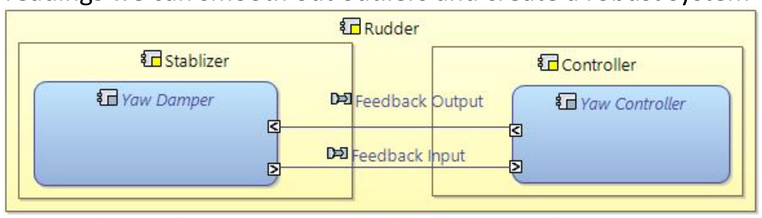

Figure 4: Rudder Component with Feedback Loop in Physical Layer

4 - Tool Qualification

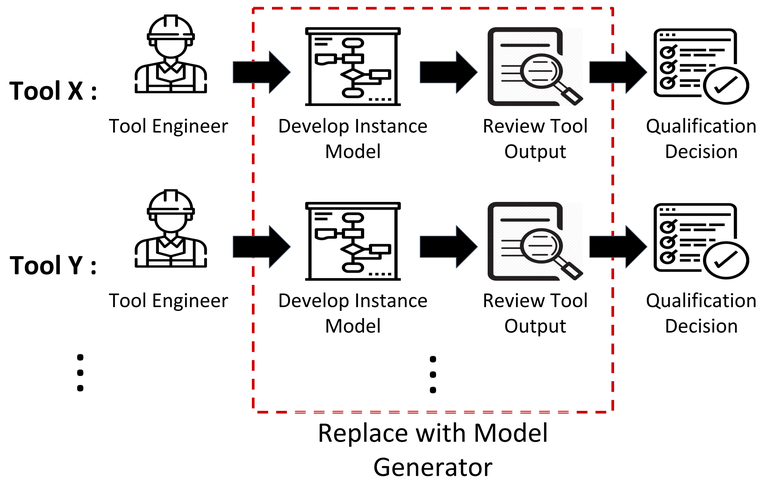

Figure 5: Standard tool qualification process.

This process requires precious time budget spent by the Tool Engineer developing the instance model and reviewing it. The use of model generator allows to shave precious time off the testing process by automating these 2 steps.

5 - Model Generation

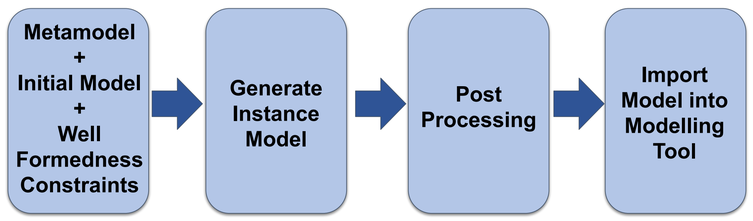

Figure 6: Tool Qualification process using Viatra Generator.

The Viatra Generator is a model generator that leverages a Logic Solver to generate **Instance Models** conforming to an Input **Metamodel**. One may additionally provide an **Initial Model** and additional **constraints** that constitute a model being regarded as **"Well-Formed"**.

6 - Post Processing

We take, as a Proof of Concept, the Yakindu Statecharts Tool and explore the practicality of following the preceding approach. Significant efforts were required in post-processing:

- **Concrete Syntax:** a concrete syntax (diagrammatic representation) must be generated from the Abstract Syntax output:
  - **Addition of name tags:** the generator is lacking in the generation of attribute values.
  - **Concrete Elements**: recursively generated off of the Abstract Model
  - **Bounds:** constraint attributes generated with a Constraint Solver (Choco-Solver)

7 - Future Work

Capella Modelling

- Perform component level breakdown for the AFCS Model by delving into the final EBPS layer
- Constructing customized templates for automated model documentation generation
- Expand AFCS scope and peripheral subsystems

Model Generation

- Create an automated testing pipeline that performs the whole testing process programmatically
- Use the test suite to find implicit constraints not defined by the metamodel, that would crash the tool.
- Converge on a generalized set of post processing steps required across different tools.

8 - Outcomes

- Development of a complete Capella Systems Model, made available to the open source community.
- Exploration of the use of Model Based Systems Engineering tools in cyber-physical system design.
- Demonstration of the capabilities of Model Generation techniques within the context of Systems Engineering Tool Qualification.
- Exploration of future considerations to be taken when using Model Generation tools to generate complex Systems Models.

9 - Acknowledgments

We would like to thank the following individuals, without whom's efforts, this project would not be possible:

- Professor Daniel Varro
- Mr. Aren Babikian
- Dr. Oszkar Semerath
- Dr. Gabor Szarnyas
- The VIATRA Development Team