Master

Co-Evolution von Modellen und Code in einem simulierten automatisierten Produktionssystem

Co-Evolving Models and Code in a Simulated Automated Production System

Motivation — English

Automated production systems (aPS) are production systems that are accompanied by digital assets that help to control and monitor them. Model-based development and Analysis of aPS before building them is important, because changing a prescriptive model is considerably cheaper than changing an aPS that is already built physically. Still, when running these systems, you must observe whether the implementation acts as prescribed. Otherwise, the analysis results do not hold for the implementation.

Our framework Codeling[1] synchronizes software design models and code by eliminating the need for models as separate artifacts. The tool translates models into source code patterns and back. This allows models such as state machines to be reliably extracted from the source code when needed, and changes in the extracted models can be propagated to the code. The generated code has well-defined interfaces for interaction with its contextual code, i.e. the rest of the system. For model/code mappings, the framework describes translations, code libraries and stubs for executing the models at runtime. The development of these artifacts in this framework is time-consuming and error-prone. By describing explicit mappings the generation of the artifacts can be automated and thus significantly improved.

The goal of this thesis is to develop a language for model/code mappings and a prototypical tool to synchronize models and code based on that language.

As a guidance, we provide the following research questions:

- RQ 1: How must a mapping language be specified to map deltas of software design models to code deltas and vice versa?
- RQ 2: How can these mappings be implemented to be executable effectively and efficiently?

You will address the following tasks during your thesis project:

1. Describe the XAPS demonstrator with our PPR meta model
2. Prototype simulating code
3. Invent a mapping description for co-evolving models and code combined (create,update,delete), which is executable (as stub)
4. Develop tool to synchronize models and code based on your mapping description
5. Evaluate your approach with the XAPS demonstrator

Knowledge required to carry out the work: Java
Helpful knowledge: Code generation, e.g., with XText; Code Analysis, e.g., with JDT, model-based software development with Ecore

Organisatorisches

Kontakt:
Dr. Marco Konersmann <konersmann@uni-koblenz.de>

[1] `https://codeling.de`

Literatur

[1] Fabian Glittenberg, Synchronisation von Designmodellen und Programmcode im Model Integration Concept, Master Thesis, 2018, Universität Duisburg-Essen (on request from konersmann@uni-koblenz.de)

[2] Birgit Vogel-Heuser, Marco Konersmann, Thomas Aicher, Juliane Fischer, Felix Ocker, and Michael Goedicke. Supporting evolution of automated Material Flow Systems as part of CPPS by using coupled meta models. In 1st IEEE International Conference on Industrial Cyber-Physical Systems (ICPS-2018), May 2018. [DOI] [pdf]

[3] Marco Konersmann. On executable models that are integrated with program code. In Proceedings of the 4th International Workshop on Executable Modeling co-located with ACM/IEEE 21st International Conference on Model Driven Engineering Languages and Systems (MODELS 2018), Copenhagen, Denmark, October 14, 2018., 2018. [pdf]