
Reference:	QSE:DSL
Topic:	Domain-Specific Engineering: Simple DSL Editor
Course Type:	Bakk-/Master Thesis, ASE or CS Project
Start:	As soon as possible
End:	To be defined
Contact:	Stefan Biffl (stefan.biffl@tuwien.ac.at) Arndt Lüder (arndt.lueder@ovgu.de)

Background

The engineering of technical systems, especially multi-disciplinary production systems, requires integration across product design, mechanical, electrical, control, and software engineering [2]. Engineers and domain experts use domain-specific terminologies and engineering models that are usually not tailored to other disciplines. To enable multi-model engineering, domain experts, who are non-experts in modelling, require simple methods to express domain knowledge (e.g., in 3D printing, assembly [3], or automotive welding processes [4]) in a **domain-specific language (DSL)** [3]. Figure 1 illustrates multidisciplinary system design and multi-model editing environments.

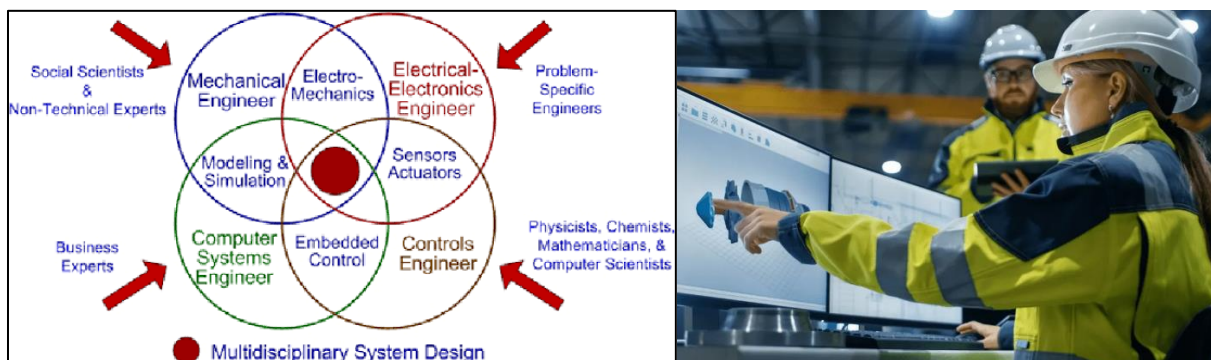


Figure 1: Left: multidisciplinary system design. Right: multi-model editing environment.

Project Goal

This project aims to explore model-based engineering of a user-friendly, simple **Domain-Specific Language (DSL) editor** with AI functions. The target tool shall be implemented as a plugin for the *AutomationML Editor* [1], enabling domain experts without modeling expertise to design, specify, and iteratively improve solution architecture models in simple language, e.g., 3D object structures and the 3D printing process. Built on lightweight DSL concepts, like the SIAM DSL [3], the editor shall facilitate creative ideation [5] while automatically translating structured inputs into robust AutomationML representations and downstream system artifacts, e.g., the validated 3D printing design.

The project will be conducted in cooperation with experts at the [AutomationML association](#) [1] and the Otto von Guericke University Magdeburg to provide sufficient knowledge and input on the application domain, the DSL, the AutomationML technology, and an AI skill description for AI-supported plug-in implementation. In case of a successful PlugIn implementation, the result shall be published as part of the *AutomationML Editor* [1] under the name of the authors.

Illustrative Use Case

Engineering real-world processes, such as 3D printing objects (including recycled materials and exception handling) or robotic welding in car production [4], requires collecting ideas and tacit knowledge from domain experts who lack formal modeling training [5].

The workflow facilitates knowledge capture and architecture optimization across four core aspects:

1. **Knowledge Collection:** Domain experts express solution architectures using simple structured language supported by procan.do Business Intelligence methods, Given-When-Then (GWT) scenarios, and lateral thinking [3, 4, 5], e.g., *Given* a 3D printed object, *When* the printing layer thickness is critically big, *Then* printing stability is insufficient.
2. **Architecture Trade-Off Analysis:** The system shall assist engineers in evaluating architecture options to shift operational complexity or liability to appropriate software or physical architecture elements [5], e.g., the stability problem for 3D printing can be handled by adapting (i) product design, (ii) filament pressure, or (iii) motion speed of the printing nozzle.
3. **AI-Assisted Plugin Integration:** Designed as a C# plugin within the leading *AutomationML Editor* [1], the tool shall leverage AI capabilities during modeling to assist the user in drafting, validating, and completing DSL specifications, e.g., AI skills for simple, typical 3D printing model patterns.
4. **Artifact Generation:** The simple DSL models serve as high-quality inputs to a wide range of engineering artifacts, including standard *AutomationML* models, *Neo4j* knowledge graphs for design-time and run-time graph queries [2, 4], and process monitoring configurations for high-quality data analysis [3], e.g., 3D printing code with monitoring to raise exceptions, if required, to trigger maintenance tasks.

Figure 2 illustrates an *AutomationML* multi-screen engineering environment. Goal of this project is to implement and validate the simple prototype DSL editor plugin, with friendly domain experts for practical 3D printing or manufacturing scenarios.

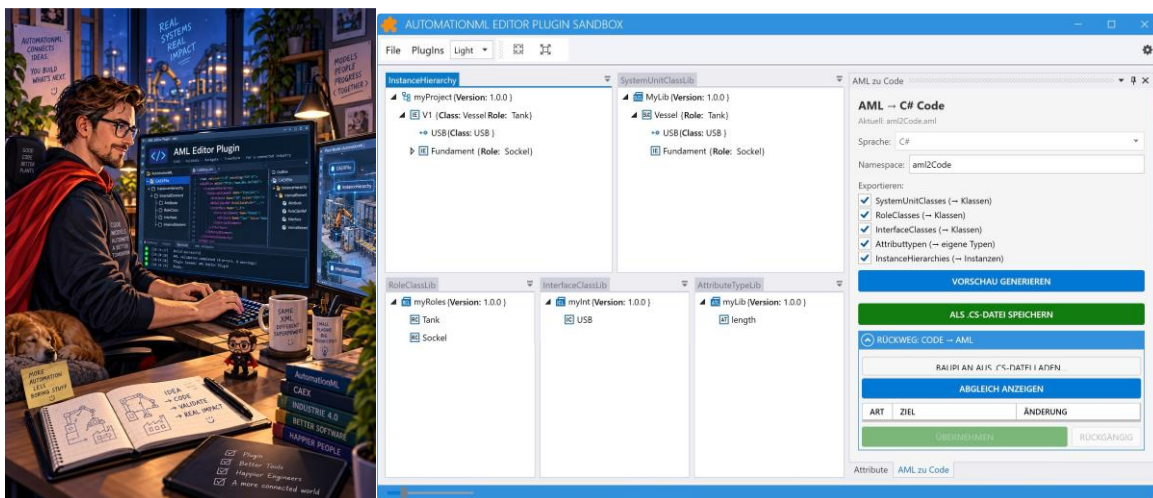


Figure 2: AutomationML multi-screen engineering environment.

Tasks

Depending on the application area and inputs, plan the project with the following task candidates:

- Elicit requirements for the DSL editor plugin [1] and target DSL syntax [2,3], for graphic editing of models following the selected DSL.
- Design the solution architecture for the plugin interface [1].
- Implement the DSL editor as a C# plugin for the *AutomationML Editor* [1].
- Apply or integrate AI functions to assist model creation and exception handling.
- Map DSL constructs to *AutomationML* representations and *Neo4j* knowledge graphs [3, 4].
- Validate plugin functionality and test the DSL with friendly domain experts in an illustrative use case, e.g., 3D printing.

Expertise

For this topic, a set of skills is recommended (at least two are mandatory):

- Programming skills in C# and software plugin development.
- Model-Based Engineering and Domain-Specific Language (DSL) design.
- Knowledge of *AutomationML* or system engineering data structures.
- Graph database concepts (e.g., Neo4j, Cypher queries).
- Interest in practical engineering application domains, e.g., 3D printing or smart production.
- Interest in AI-assisted software engineering and editor capabilities.

References

- [1] AutomationML e.V., "[AutomationML Editor and Plugin Concept](#)". Web link.
- [2] Hünecke, P., Binder, C., Hoffmann, D., Hoang, T., & Lüder, A. (2025). [Common Concept Method for Citizen Modelers in Cross-Domain Engineering](#). In *IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)* (pp. 1-8). IEEE.
- [3] Biffel, S., Bein, T., Kropatschek, S., Meixner, K., Lüder, A. (2026) [SIAM DSL: Accessible Structured Text for Bridging the Architecture-Impact Gap in Cyber-Physical Production Systems](#). WiPiEC Journal Vol. 12, No. 2.
- [4] Biffel S., Rahmani H., Meixner K., Hoffmann D., Lüder A. (2026) [Value-based Change Impact Analysis in Production with System Impact Architecture Modeling](#). Proc. ETFA, IEEE.
- [5] Biffel, S., Marhold, K., Musil, J., Meixner, K. (2026) [Business intelligence for value-based process and systems engineering and operation with procan.do](#). In: Business Intelligence for Multi-Domain Systems Engineering and Operation. Springer. > Section 5.2.x