

# A Domain Specific Language for Project Execution Models

Eugen Wachtel, Marco Kuhrmann, Georg Kalus

Technische Universität München, Institut für Informatik – I4  
Boltzmannstr. 3, 85748 Garching, Germany  
{wachtel|kuhrmann|kalus}@in.tum.de

The modeling of strategies for deriving valid project plans is a core task in development process's design. Strategies are used for planning and for the validation of concrete project plan-instances. Especially if using delivery-oriented planning, project managers need to know if they have all required deliverables available or if a scheduled project satisfies the process's requirements.

In this paper, we present a domain specific language (DSL) to model and validate a project plan as well as its execution, which is specified in form of a template called process execution strategy. The modeling of strategies is usually done by process engineers who require a deep understanding of the required workflows as well as knowledge about the underlying meta-model and the allowed processes. Especially the second part can be enforced or at least validated by specific software, making the modeling of complex strategies more manageable. Our DSL supports process engineers during the process design, in process improvement or introduction projects, allowing them to define and modify strategies and by providing real-time validation feedback visualizing inconsistencies.

To give a relevant example, we introduce a Microsoft DSL-Tools-based language for the V-Modell XT 1.3 describing its project execution strategies (PES). The meta-model for PES in the V-Modell XT is hierarchical and highly modular, which makes the modeling of strategies for the process engineer rather complicated and fault-prone. Therefore, the V-Modell XT is a good example for our DSL.

To introduce the implementation of our prototype we present a domain model with its two facets for modeling project execution strategies and project plans. As the strategy definitions part is based on the V-Modell XT meta-model, we investigate and propose the required extensions and/or restrictions needed to guarantee a valid modeling of strategies. Further, we explain how the domain model can be transformed into the Microsoft DSL-Tools domain model. We also present the DSL-Tools designer and highlight a couple of noteworthy implementation details.