

Reuse is considered as one of the most crucial software engineering concerns. It allows for delivering software systems faster with less effort. Therefore, the thesis explores limits of reuse in the context of component systems. It analyzes in depth contemporary component systems, finds their commonalities and variation points, and introduces a meta-component system -- a software product line which allows for producing a tailored component system based on a set of requirements.

The thesis addresses the meta-component system definition and focuses on its crucial aspects which play the key role in component systems preparation- (1) a configurable execution environment and (2) generation of implementation artifacts. To address the first aspect, the thesis proposes a model-driven method for creating configurable execution environments. Motivated by creating execution environments, the thesis contributes to (2) by introducing a notion of domain-specific languages interoperability in the context of the code generation. Furthermore, the thesis elaborates the proposed notion resulting into a family of interoperable domain-specific languages which is parametrized by a general purpose language.