

論文の内容の要旨

A Language-based Approach to Model Synchronization in Software Engineering

(ソフトウェア工学におけるモデル同期に関する言語論的研究)

氏名 熊 英飛

(本文) Software development often involves a set of heterogeneous artifacts, such as requirement documents, design models and implementation code. Maintaining these artifacts has been a notorious problem in software engineering. When we update part of an artifact, we need to propagate the update across all artifacts to make them consistent. Such synchronization of heterogeneous artifacts is known to be time-consuming and error-prone.

Recently, meta model technique emerges to give a unified data represent of artifacts by capturing them as models, enabling a unified way for programs to access artifacts. Based on models, people develop tools to automate some synchronization tasks, and these model synchronization tools have shown their usefulness in software development. However, as synchronizing artifacts may be very complex, such tools are usually difficult to develop and maintain.

In this thesis we propose a language-based approach to facilitating the development of model synchronization tools. We design specification languages for model synchronization. The specification languages mainly describe the consistency relations between artifacts, with a small amount of additional information to confine the synchronization behavior. When users give high-level specification of synchronization in these languages, we generate a synchronizer from the specification, and the synchronizer automatically synchronizes models to keep them consistent. We also identify the requirements for model synchronization, which consists of three properties to ensure the correctness of synchronization. We prove that our generated synchronization procedures satisfy the requirements we proposed.

We design the specification languages from the languages that developers use to describe

the consistency relation in practice. In this way developers need less effort to learn our languages and can reuse their existing programs. We identify two typical types of model synchronization and design different specification languages for the two types. The first type, off-site synchronization, is used to integrate different software development tools. In tool integration developers usually describe the consistency relation between models in model transformation languages. A program in such languages converts models from one format into another, but cannot synchronize the models after transformation. We adopt a unidirectional model transformation language as specification language and use a trace-based technique to derive a synchronizer from a transformation program. In cases where there are already two existing transformations to transform between two models forwardly and backwardly, we also design an algorithm to wrap them into a synchronizer that allows parallel updates on the two models.

The second type is about the synchronization within one tool. In such cases developers usually describe the consistency relation over models in a logic-based specification language. We design Beanbag, a specification language whose syntax is similar to first-order logic but developers can customize the synchronization behavior by adjusting their programs. We also discuss the implementation issues of the Beanbag language.

All our languages have been implemented, and we have applied them to real-world cases. The result shows that our approach is useful in practice.