

20 Years of Concurrent Model Driven Application Engineering with Triple Graph Grammars (Abstract)

Andy Schürr

FG Real-Time Systems
Technische Universität Darmstadt
Germany

Today model-driven system development is a well-established and successfully used paradigm in many engineering disciplines including e.g. mechanical, electrical, automation, and software engineering. Models are, therefore, used and concurrently manipulated in different disciplines at different levels of abstractions across the whole lifecycle of a single product. Keeping all these models and related other engineering artefacts synchronized often turns out to be a nightmare. The situation even becomes worse when product line engineering principles are used to develop a family of products instead of a single product on top of a common platform.

Model transformation techniques promise to be a silver bullet for the construction and validation of reliable and efficiently working model synchronizers on a new level of abstraction. Triple Graph Grammars (TGGs) are a bidirectional model transformation formalism, where a single specification generates a language of related graph tuples (pairs of models) together with an intermediate correspondence graph (traceability link database). A single TGG specification is used as input for a compiler that generates batch transformations as well as incrementally working synchronizers that assist engineers in their daily work to keep networks of evolving models and their traceability relationships in a consistent state.

This talk starts with an introduction to concurrent model driven application engineering in general, explains how triple graph grammars are specified and used to generate model transformation and integration tools, and finishes with an assessment of the state-of-the-art of 20 years of TGG related research activities.