Finding Models in Model-Based Development
(Abstract)

Wolfram Schulte and Ethan Jackson

Microsoft Research, Redmond, WA
{schulte,ejackson}@microsoft.com

Model-based development focuses on creating and manipulating domain models. We present the FORMULA language and its tool environment for specifying, documenting, and analyzing models.

The FORMULA language is based on the observation that constraints are ubiquitous. For instance, a real-time system must meet its deadlines, a software deployment must obey resource constraints, a compiler must preserve the meaning of its source language. Each design problem is defined w.r.t. some abstraction; in FORMULA these abstractions are called domains. A domain encapsulates a set of data structures used to formalize key concepts, and *logic programming* is used to describe restrictions on the set of possible solutions. Complex systems have a multitude of facets. Our language provides a rich set of domain composition operators for building new abstractions. Similarly, transforms synthesize other models at the same or different level of abstractions. FORMULA has a standard first-order logic semantics.

The FORMULA solver answers queries under the *open-world-assumption*, which considers that not all facts are known *a priori*. Evaluating a query under this assumption means searching for a finite set of facts where the program satisfies the query. These missing facts are the solutions to our modeling problems, *e.g.* legal instances of schedules, feasible deployments, necessary synchronization constraints. In the end, FORMULA translates to state-of-the-art satisfiability-modulo-theory solvers, which search through complex spaces in the presence of many constraints.

FORMULA draws on methods from type theory, logic programming, and automatic theorem proving. It has successfully been applied to a number of domains from scheduling, meta-modeling and configuration management, to software deployment. More information at: `http://research.microsoft.com/formula`. Joint work with Nikolaj Bjorner, Dirk Seifert, Markus Dahlweid, and Thomas Santen.

References

1. Jackson, E.K., Kang, E., Dahlweid, M., Seifert, D., Santen, T.: Components, Platforms and Possibilities: Towards Generic Automation for MDA. In: EMSOFT, pp. 39–48 (2010)
2. Jackson, E.K., Balasubramanian, D., Levendovszky, T.: Reasoning about Metamodeling with Formal Specifications and Automatic Proofs. In: Whittle, J., Clark, T., Kühne, T. (eds.) MODELS 2011. LNCS, vol. 6981, pp. 647–661. Springer, Heidelberg (2011)
3. Jackson, E.K., Bjørner, N., Schulte, W.: Canonical regular types. In: ICLP (Technical Communications), pp. 73–83 (2011)

J. Whittle, T. Clark, and T. Kühne (Eds.): MODELS 2011, LNCS 6981, p. 591, 2011.