Abstract

Formal methods (such as interactive provers) are increasingly used in software engineering. They offer a formal frame that guarantees the correctness of developments. Nevertheless, they use complex notations that might be difficult to understand for unaccustomed users. On the contrary, visual specification languages use intuitive notations and allow to specify and understand software systems. Moreover, they permit to easily generate graphical interfaces or editors for Domain Specific Languages (DSLs) starting from a meta-model. However, they suffer from a lack of precise semantics. We are interested in combining these two complementary technologies by mapping the elements of the one into the other.

In this paper, we present a generic transformation process from functional data structures, commonly used in proof assistants, to Ecore models and vice-versa. This translation method is based on Model-Driven Engineering and defined by a set of bidirectional transformation rules. These rules are presented with an illustrating example, along with an implementation in the Eclipse environment