An ECLIPSE Framework for Rapid Development of Rich-featured GEF Editors based on EMF Models

Tony Modica, Enrico Biermann, Claudia Ermel
Institut für Softwaretechnik und Theoretische Informatik
Technische Universität Berlin
E-Mail: modica,enrico,lieske @cs.tu-berlin.de

Model-based development has an increasing importance in modern software engineering and other domains. Visual models such as Petri nets and UML diagrams proved to be an adequate way to illustrate many structural and behavioral system properties. However, while tooling for textual modeling is pretty mature now, visual tool builders are faced with a much higher complexity regarding the representation of model properties, and the interplay of the concrete syntax (the views) with the underlying abstract model representation, e.g. based on Java, XML or the Eclipse Modeling Framework (EMF). In order to ease the development of visual editors, the Graphical Editing Framework (GEF) oprovides the layout and rendering toolkit DRAW2D for graphics and follows the model-view-controller (MVC) architectural pattern to synchronize model changes with its views and vice versa. GEF's architecture allows to integrate models based on EMF, Java or XMI with their visual views and editors.

Generally, with a complex framework as GEF there are always many different possibilities to approach the implementation of a feature. Several frameworks support generation of code for from abstract editor specifications, like the ECLIPSE Graphical Modeling Framework (GMF) or MOFLON. Unfortunately, the visual languages we usually want to implement editors for and their simulation operations seem not appropriate to be specified in these frameworks, which assume models to be displayed in a single pane only. Moreover, if we used e.g. GMF to generate code as far as possible, GEF apprentices without deeper knowledge of the mechanisms in GEF would surely struggle when laying hand on the generated code to extend it with complex features.

Our research group has its main focus on applying formal techniques to visual modeling languages. We used our past experiences in teaching the development of GEF-based editors to generalize recurring code fragments for many editor features into code templates and to document them properly for simplifying the familiarization process for the students as well as the editor implementation. This development lead to our GEF-based framework MUVITORKIT (Multi-View Editor Kit). MUVITORKIT supports nested models, models needing multiple graphical viewers, and animated simulation of model behavior. Its architecture is designed in a way that encapsulates complex underlying mechanisms in GEF and simplifies the interaction with the ECLIPSE workbench.