GenEd – A Generic Editor for Reasoning about Visual Notations

Volker Haarslev and Michael Wessel
University of Hamburg, Computer Science Department,
Vogt-Köln-Str. 30, 22527 Hamburg, Germany
http://kogs-www.informatik.uni-hamburg.de/~haarslev/
http://kogs-www.informatik.uni-hamburg.de/~mwessel/

Summary
We demonstrate the object-oriented editor GenEd supporting the design of specifications for visual notations. Prominent features of GenEd are (1) it is generic, i.e. domain-specific syntax and semantics are specified by users; (2) built-in parser for actual drawings, driven by formal specifications; (3) reasoning capabilities about diagrams and their specification. GenEd's specification language is based on a fully formalized theory for describing visual notations. An example with place-transition petri nets is presented in Figure 1 and 2.

The design and evaluation of the generic editor GenEd for visual notations is fully presented in (Haarslev & Wessel 1996). The term “generic” refers to the characteristic that users can specify domain-specific syntax and semantics for visual notations. We think of formal notations as a very general notion. Examples for visual notations can be found in areas such as music, geography, mathematics, chemistry, etc. Visual programming languages are viewed as a subset of formal notations.

GenEd is intended for supporting the design and analysis of visual notations or visual programming languages. Usually the design of new visual notations is an exploratory approach. Our experience with the design of formal semantics for a completely visual programming language (Haarslev 1995) has strongly motivated the approach presented in this paper. We argue that designers of notations should be offered an almost freeform, purely declarative style for specifications with immediate support for analysis and reasoning about specifications and for verification through parsing of example drawings. GenEd's parser can operate in two modes. The incremental mode validates drawings after every relevant modification and reports parsing successes and errors to the user. The second mode performs parsing only when demanded by the user.

Our approach is based on a fully formalized theory about visual notations (Haarslev 1998) that can be defined from "first principles" by starting with point-sets and topology. GenEd's specification language and its reasoning capabilities depend on description logic theory (e.g. see (Brachman & Schmolze 1985)). Our approach is in contrast to other approaches that use only syntax specifications or favor a generative solution, i.e. they create specialized editors for particular visual languages.

GenEd is implemented in Common Lisp using the Common Lisp Object System (CLOS) and the Common Lisp Interface Manager (CLIM) as interface toolkit. The classification of concepts and the parsing of actual drawings take place by using CLASSIC (Brachman et al. 1991; Brachman 1992) as description logic reasoner. CLASSIC is also implemented in Common Lisp. GenEd consists of 28 modules with a total of about 300 KB source code (without CLIM, CLOS, and CLASSIC).

References
Figure 1: GenEd: petri net for reader-writer problem (simplified)

Figure 2: GenEd: magnified selection of petri net from Figure 1