

Interpretation of Visual Notations in the Recopla Editor Generator

Bernd Meyer
bernd.meyer@acm.org

Hubert Zweckstetter
hsz@fast.de

Ludwig-Maximilians Universität, Institut für Informatik
Oettingenstr. 67, D-80538 München, Germany

Even in application domains where diagrammatic notations are a natural element of discourse and their meaning is well-understood, their usage in the human-computer interface is still very limited. Mainly this is because of the difficulties encountered when implementing a diagrammatic interface. Despite the fact that a number of editor generators or compilers are available, only a handful of research prototypes support the high-level specification of syntax and interpretation of diagrams. These issues in the context of incremental interactive systems are addressed by the Recopla editor generator that we are demonstrating.

Recopla is a Java-based editor generator used to construct interactive graphical front-ends (so-called Recopla instances) that are specialized for particular types of diagrams, like e.g. an editor for electronic circuit diagrams. Recopla lets the user specify appearance, syntax and semantics of a diagram language and compiles this into a Java implementation of an interactive editor for this particular notation which is capable of automatically interpreting the user's input.

A vast number of diagrammatic notations used in practice essentially have a (hierarchical) graph as their underlying backbone structure. The Recopla environment is therefore designed to offer special support for the definition of and interaction with graph structures. However, the system also supports editing and interpretation of other diagrammatic notations.

The appearance of the graphical entities, i.e. the vocabulary, that may be used by an editor instance is specified interactively on the meta-level by drawing prototype entities with a normal object-oriented graphics editor. The syntax, i.e. the permissible ways to arrange these graphical entities spatially and topologically in a valid diagram, is specified by *conditional relational grammars*, a multidimensional grammar formalism that inherits from both relational grammars and constraint multiset grammars (Marriott, Meyer, & Wittenburg 1998). The interpretation and translation of diagrams is specified in the same formalism by using attributed productions and/or by attaching actions to productions. The following production is an example taken from a simple grammar for state transition diagrams:

```
Transition --> Arrow, Label ;      % where
exists(State_1, State_2),         % such that
starts_at(State_1, Arrow),
ends_at(State_2, Arrow),
close_to(Label, Arrow) ;         % and set
Transition.condition = Label.string,
Transition.from      = State_1.name,
Transition.to        = State_2.name.
```

The formalism allows the usage of context-sensitive productions and of arbitrary user-defined predicates that are evaluated as conditions over the objects' attributes. Such a grammar is processed in bottom-up fashion by simultaneously rewriting a set of graphical objects and a set of spatial relations between these objects.

In interactive systems it is desirable to support immediate feedback from the interpretation without restricting the admissible user-interactions. For systems working on the basis of a formal syntax definition this is a difficult problem since for this they have to address the issue of incremental parsing and, more importantly, of non-monotonic incremental interpretation.

In Recopla's interpretation component this is solved by viewing parsing as logical deduction: The grammar execution can essentially be understood as a bottom-up derivation of horn-clauses with set-valued attributes. This perspective allows us to employ deduction techniques developed in the area of Truth Maintenance Systems by which a consistent interpretation can incrementally be maintained when the diagram is changing without having to initiate a complete reinterpretation from scratch.

The demonstration will show some editors built with Recopla as well as the process of constructing a new editor.

Acknowledgments

The authors would like to thank Luis Mandel and Zoltan Gassmann who have designed and implemented the original Recopla environment on top of which we have built the interpretation component.

References

Marriott, K.; Meyer, B.; and Wittenburg, K. 1998. A survey of visual language specification and recognition. In Marriott, K., and Meyer, B., eds., *Visual Language Theory*. New York: Springer-Verlag.