Visual Specification of Layout

Sonja Maier, Steffen Mazanek, and Mark Minas

Universität der Bundeswehr München, Germany

Abstract. We give an overview of a drawing approach that combines the concepts constraint satisfaction, attribute evaluation and transformation. The approach is tailored to an editor for visual languages, which supports structured editing as well as free-hand editing. In this paper, we focus on the visual specification of such a layout algorithm. As a running example, deterministic finite automata are used.

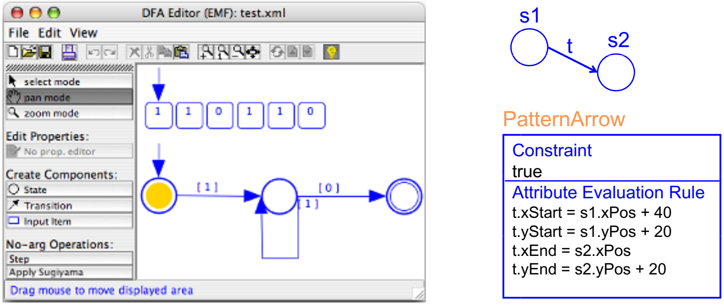
1 Visual Specification

When implementing an editor for a visual language, a challenging task is layout. The drawing approach should produce a good-looking result and should support the user. Additionally, the layout specification should be very easy. In the following, we introduce an approach that aims to achieve these concurrent goals.

Implementation. The approach was implemented and tested in DiaMeta [1], an editor generation framework. Figure 1(a) shows a DiaMeta editor for DFA's.

Aspects. The approach supports structured editing as well as free-hand editing. Structured editors offer the user some operations that transform correct diagrams into (other) correct diagrams. Free-hand editors allow to arrange diagram components from a language-specific set on the screen without any restrictions, thus giving the user more freedom.

The approach is best suited for layout refinement, which starts with an initial layout and performs minor changes to improve it while still preserving the "mental map" [2] of the original layout.



```
true
```

```
t.xStart = s1.xPos + 40
t.yStart = s1.yPos + 20
t.xEnd = s2.xPos
t.yEnd = s2.yPos + 20
```

(a) (b)

Fig. 1. (a) Editor for DFA's. (b) Visual Rule.

I.G. Tollis and M. Patrignani (Eds.): GD 2008, LNCS 5417, pp. 443–444, 2009.

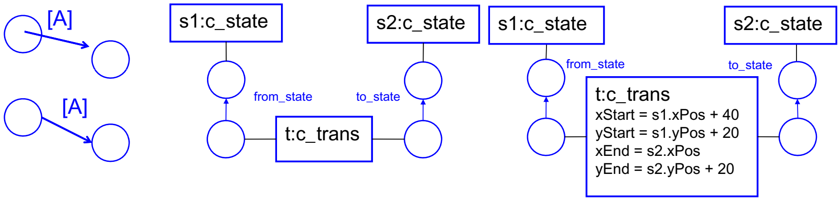


Fig. 2. Transitions (Editor, Rule (HGM))

Algorithm. We have introduced a dynamic layout algorithm, which combines the concepts constraint satisfaction and attribute evaluation [1]. This approach provides us with all we need for layout refinement. To simplify layout specification, we combined graph transformation with this dynamic drawing approach. With the approach, layout specification may be done on the abstract or on the concrete syntax level of a diagram language. In DiaMeta, this means changing an EMF model (abstract syntax level) or a hypergraph (concrete syntax level). Up to now, the layout specification is textual. To allow a more intuitive description of layout, we introduce a visual language for layout specification.

Example. We demonstrate our approach by specifying a rather simple layout for DFA's. Generally, a layout algorithm consists of a set of rules that modify the hypergraph. Each rule either changes attributes or the structure of the hypergraph. From this set, the drawing facility is generated and automatically included in the editor. The layout algorithm may be triggered manually (click “Apply Sugiyama”, Fig. 1(a)) or automatically.

Figure 2 shows a sample layouting rule that modifies attributes. Similarly, also rules that change the structure of the graph may be specified. On the left side, a DFA before and after applying rules that update the attributes *xStart*, *yStart*, *xEnd* and *yEnd* of the arrow *t* is shown. On the right side, the rule that is responsible for updating these attributes is presented. The rule checks for two states connected by a transition, if the arrow exactly starts and ends at the borderline of the circle. If this is not the case, the attributes are updated. Figure 1(b) shows this rule specified visually. An editor for specifying rules visually is automatically generated.

Conclusion. In this paper, we gave an overview of a drawing approach that supports structured editing as well as free-hand editing. It is possible to specify the layout algorithm visually. With this approach, an environment was created that allows us to conduct experiments easily and to identify the “best” layouting strategy.

References

1. Maier, S., Minas, M.: A generic layout algorithm for meta-model based editors. In: Schürr, A., Nagl, M., Zündorf, A. (eds.) AGTIVE 2007. LNCS, vol. 5088, pp. 66–81. Springer, Heidelberg (2008)
2. Purchase, H.C., Hoggan, E.E., Görg, C.: How important is the “mental map”? - an empirical investigation of a dynamic graph layout algorithm. In: Kaufmann, M., Wagner, D. (eds.) GD 2006. LNCS, vol. 4372, pp. 184–195. Springer, Heidelberg (2007)