Ein Werkzeug für die Implementierung von Diagrammsprachen*

Berthold Hoffmann[1] und Mark Minas[2]

[1] Universität Bremen, hof@informatik.uni-bremen.de
[2] Universität der Bundeswehr München, minas@acm.org

Diagrammsprachen wie die *Unified Modeling Language* (UML) gewinnen in der Informatik ständig an Bedeutung; sie werden – wie UML – zur Spezifikation von Software oder sogar zum *visuellen Programmieren* benutzt. Die Implementierung solcher Sprachen ist immer noch eine anspruchsvolle und langwierige Aufgabe.

Mit dem seit 1997 an der Universität Erlangen-Nürnberg entwickelten Diagrammeditor-Generator DiaGen [2] können nahezu beliebige Diagrammsprachen leichter implementiert werden. Mit Graphtransformationen wird die räumliche Anordnung von Diagrammzeichen und ihre Zusammenfassung zu Diagrammsymbolen beschrieben; die Syntax der Diagrammsprache wird mit Graphgrammatiken definiert. Daraus generiert DiaGen einen Editor, der sowohl freies Editieren als auch strukturierte Editier-Operationen unterstützt. (Solche Operationen können ebenfalls als Graphtransformationen spezifiziert werden.) Der hierfür notwendige Layoutmechanismus wird Constraint-basiert spezifiziert. Der arbeitsfähige Prototyp ist in Java implementiert. Mit ihm wurden Editoren für so unterschiedliche Sprachen wie Nassi-Shneiderman-Diagramme, Statecharts und UML-Klassendiagramme erstellt.

Bislang unterstützt DiaGen vorrangig die Implementierung der *Syntax* von Diagrammsprachen; ein Attributierungsmechanismus erlaubte es, die erkannten Diagramme in eine beliebige Form umzusetzen und danach außerhalb des Systems weiter zu verarbeiten. Mit DiaPlan [1] konzipieren wir nun eine Sprache zum regelbasierten Programmieren mit Graphen und Diagrammen, mit der die *Semantik* von Diagrammsprachen auch innerhalb des DiaGen-Systems beschrieben werden kann, beispielsweise die Animation oder Transformation von Diagrammen. DiaPlan ist eine regelbasierte Sprache, die Elemente logischer und funktionaler Sprachen enthält und dabei selbst visuell ist.

DiaPlan soll aber nicht nur als Spezifikationssprache für DiaGen dienen. Vielmehr soll DiaPlan zusammen mit DiaGen das visuelle Programmieren mit frei wählbaren, anwendungsabhängigen (visuellen) Darstellungen der manipulierten Datenstrukturen ermöglichen. Langfristiges Ziel ist es, mit DiaPlan und DiaGen eine visuelle Programmiersprache und -umgebung zu entwickeln, die an das jeweilige Anwendungsgebiet angepasst werden kann und so den Bruch zwischen dem vorwiegend visuellen Entwurf (z.B. mit UML) und der anschließend (immer noch) textuellen Codierung von Programmen zu überwinden hilft.

Im Vortrag wollen wir kurz die Basis des Werkzeugs, die Graphtransformation, resümieren, die Arbeitsweise von DiaGen an einem Beispiel erläutern, und die geplanten Erweiterungen, insbesondere DiaPlan, skizzieren.

Literatur

1. Hoffmann, B.: *Abstraction and Control for Shapely Nested Graph Transformation*. In: Corradini, A., H. Ehrig, H.-J. Kreowski und G. Rozenberg (Hrsg.): *1st Int'l Conference on Graph Transformation (*ICGT *'02)*, Nr. 2505 in *Lecture Notes in Computer Science*, S. 177–191. Springer, 2002.
2. Minas, M.: *Concepts and Realization of a Diagram Editor Generator Based on Hypergraph Transformation*. Science of Computer Programming, 44(2):157–180, 2002.

* Diese Arbeit wurde von der Europäischen Gemeinschaft im Rahmen der Arbeitsgruppe SeGraVis unterstützt.