

Tagungsbericht

Softwarevisualisierung 2000

Workshop der Gesellschaft für Informatik e.V. (GI)
Dagstuhl, Deutschland, 11.-12. Mai 2000

Veranstaltende Fachgruppen der GI: 2.1.1 Softwaretechnik, 4.1.2 Imaging und Visualisierungstechniken

Am 11. und 12. Mai fand in Schloß Dagstuhl der erste Workshop "Softwarevisualisierung" (SV2000) statt. Der Workshop wurde von Dr. Stephan Diehl und Andreas Kerren (beide Universität des Saarlandes) organisiert. Die Zielsetzung dieser Workshop-Reihe ist es, ein Forum zu schaffen für Gruppen im deutschsprachigen Raum aus dem Bereich der Softwarevisualisierung, und somit diente der erste Workshop auch dem Kennenlernen.

An dem Workshop nahmen 26 Teilnehmer aus ganz Deutschland teil, die im wesentlichen aus der Forschung kamen. Softwarevisualisierung wird in vielen Bereichen der Informatik betrieben, und so wundert es nicht, daß die Teilnehmer des Workshops ein breites Spektrum an informatischem Hintergrund aufwiesen, der vom Compilerbau und visueller Programmierung über Computergrafik und Softwareengineering bis hin zu Algorithmen und Datenstrukturen reichte. Das Programm umfaßte 10 Vorträge und 3 Kurzvorträge, sowie zwei Demo-Sessions, in denen fast alle der vorgestellten Systeme vorgeführt wurden.

Der überwiegende Anteil der Vorträge befaßte sich mit der Softwarevisualisierung für die *Informatikausbildung*. Hier reichte die Spannweite von der *Algorithmenanimation* und Visualisierung von *Datenstrukturen* bis zur Animation von *Berechnungsmodellen*. Mit dem System KIEL können funktionale Programme visuell ausgewertet werden (Rudolf Berghammer, Universität Kiel). Der Ablauf einfacher JAVA-Programme sowie Protokolle aus der Systemsoftware können mit dem System Vivaldi interaktiv visualisiert werden (Dieter Jahn, Rainer Öchsle, Fachhochschule Trier). Das System ViACoBi zur Algorithmenanimation (Wolfram Luther, Universität Duisburg) wurde insbesondere im breiteren Kontext von Lehr/Lernanwendungen diskutiert. In einem anderen Ansatz wurde der Grad der Lerneraktivität bzw. Lernerinteraktion thematisiert und eine Softwarearchitektur zur Unterstützung unterschiedlicher Interaktionsarten vorgestellt (Nils Faltin, Universität Oldenburg).

Auffallend war, daß derartige Anwendungen nicht mehr adhoc entstehen, sondern daß dafür *Generatoren* und *Frameworks* entwickelt wurden, wie z.B. GANIMAL zur Generierung interaktiver Animationen von Berechnungsmodellen aus dem Compilerbau (Stephan Diehl, Andreas Kerren, Universität des Saarlandes). Generatoren werden auch in der *visuellen Programmierung* eingesetzt, wie der Editorgenerator DiaGen, der auf Graphtransformationssystemen basiert (Berthold Hoffmann, Universität Bremen, Mark Minas, Universität Erlangen-Nürnberg). Mit Hilfe des Observer-Entwurfsmusters lassen sich Algorithmenanimationen mit minimalem Eingriff in Algorithmen erstellen, die mit der LEDA-Bibliothek für Algorithmen und Datenstrukturen programmiert wurden (Ste-

fan Näher, Universität Trier, Oliver Zlotowski, Universität Halle-Wittenberg).

Eine wichtige Rolle spielt die Softwarevisualisierung auch im Kontext des *Softwareengineering*. Hier lag der Schwerpunkt beim *Debugging*. Es wurde ein UML-basierter Ansatz vorgestellt, um Fehler bei der Synchronisation in nebenläufigen Programmen zu visualisieren (Katharina Mehner, Annika Wagner, Universität Paderborn). Das von vielen Entwicklern genutzte Debugging-Werkzeug DDD unterstützt u.a. die Visualisierung und Animation von Datenstrukturen (Andreas Zeller, Universität Passau). Grundlage für jede Art der Visualisierung von graphischen Strukturen für Probleme realistischer Größe sind *Layoutalgorithmen*. Vorgestellt wurde hierzu eine Erweiterung des bekannten Sugiyama-Algorithmus für Klassendiagramme (Holger Eichelberger, Universität Würzburg).

Der erste Kurzvortrag stellte ein Modul für die Animation von *Graphalgorithmen* vor (Roland Wiese, Universität Tübingen). Danach wurde das Gebiet des *Scientific Computing* beleuchtet, dessen Schwerpunkt auf der Echtzeitvisualisierung großer Datenmengen liegt. Dabei wurden insbesondere Erfahrungen hervorgehoben, die für die Softwarevisualisierung von Nutzen sein könnten (Hans Hagen, Universität Kaiserslautern). Zuletzt wurde die Technik der *Shape Analysis* vorgestellt, mit der Datenstrukturen eines Programmes statisch analysiert werden, um Invarianten in Form abstrakter Shape-Graphen zu finden. Diese können zur visuellen Algorithmenklärung verwendet werden (Reinhard Wilhelm, Universität des Saarlandes).

In der Diskussion wurde festgestellt, daß viele Beiträge ähnliche Techniken mit unterschiedlichen Zielen verwendeten und eine bessere Koordination der Projekte ein effektiveres Arbeit fördern würde. Daher wurde beschlossen, eine Webpage aufzubauen, die Links zu den im deutschsprachigen Raum aktiven Gruppen und Personen im Bereich der Softwarevisualisierung enthält, sowie Demos und eine "elektronische" Bibliothek (<http://www.softwarevisualisierung.de>).

Der Tagungsband ist erschienen als Technischer Bericht A/01/2000 der Universität des Saarlandes. Der nächste Workshop soll in zwei Jahren stattfinden. Darüber hinaus konnten die Organisatoren gemeinsam mit John Stasko (Georgia Institute of Technology) und Peter Eades (University of Sydney) erreichen, daß im nächsten Jahr erstmalig ein Dagstuhlseminar über Softwarevisualisierung stattfinden wird (20.-25. Mai 2001, <http://www.dagstuhl.de/>).

Stephan Diehl, Andreas Kerren, Universität des Saarlandes, Katharina Mehner, Universität Paderborn