

Lehrstuhl für Informatik II

Programmier- und Dialogsprachen sowie Compiler

Leiter: Prof. Dr. Hans Jürgen Schneider

Mitarbeiter:

Allendorf, Helmut	(Programmierer)	ab 01. 05. 97
Billing, Gunnar, Dipl.-Inf.	(wiss. Mitarb., FORSOFT)	ab 01. 03. 97
Dormeyer, Ricarda, Dipl.-Inf.	(wiss. Mitarb.)	
Fischer, Ingrid, Dipl.-Inf.	(wiss. Mitarb.)	
Hodek, Roman, Dipl.-Inf.	(wiss. Mitarb.)	
Jacob, Christian, Dr.-Ing.	(wiss. Mitarb.)	beurl. bis 31. 10. 97
Koch, Manuel, Dipl.-Inf.	(wiss. Mitarb.)	16. 04. - 31. 08. 97
Lührs, Erni	(Sekretärin - halbtags)	
Minas, Mark, Dr.-Ing.	(Akad. Rat)	
Nilson, Jörg, Dipl.-Inf.	(wiss. Mitarb.)	
Schörmal, Elfriede	(Sekretärin - halbtags)	
Uebler, Manfred	(Programmierer)	
Wilke, Peter, PD Dr.-Ing. habil.	(Akad. Rat)	

Gastwissenschaftler:

Billing, Gunnar, Dipl.-Inf.	(Bay. Kultusministerium)	bis 28. 02. 97
Kkai, Gabriella	(Bay. Kultusministerium)	ab 15. 10. 97
Schoberth, Andreas, M. Sc.	(FAST e.V.)	bis 31. 07. 97
Volle, Viktor, Dipl.-Inf.	(BMFT)	ab 01. 08. 97

Lehrbeauftragte:

Feder-Andres, Christiane, Dr.-Ing.	(Fa. sd&m, München)
Hindel, Bernd, Dr.-Ing.	(Fa. 3SOFT, Erlangen)
Kips, Detlef, Dr.-Ing.	(Fa. BASYS, Erlangen)
Schorr, Ruth, Dr.-Ing.	(Deutsche Bank, Frankfurt/Main)

Gäste:

Feder-Andres, Christiane, Dr.-Ing.	(Fa. sd&m, München)	27. 01. 97
Göttler, Herbert, Prof. Dr.-Ing.	(Universität Mainz)	28. 02. 97

Janssens, Dirk, Prof. Dr.

(Universität Antwerpen/Belgien)

27. - 31. 01. 97

Kkai, Gabriella

(Universität Szeged/Ungarn)

24. - 28. 02. 97

Kooperationspartner:

Arbeitsgemeinschaft Software-Qualität Franken e.V.

Bayerischer Forschungsverbund Software-Engineering

ESPRIT-Programme APPLIGRAPH (Applications of Graph Transformation)

EU - TMR Network GETGRATS (General Theory of Graph Transformations)

Jzsef Attila Universität Szeged, Ungarn (Dr. Tibor)

Rutgers University New Brunswick, NJ, USA (Dr. Shklar)

Technische Universität Berlin (Prof. Dr. Ehrig)

Technische Universität München (Prof. Dr. Broy)

Universität Antwerpen, Belgien (Prof. Dr. Janssens)

Universität Bremen (Prof. Kreowski)

Universität Karlsruhe (Prof. Rembold)

Universität Krakau, Polen (Dr. Grabska)

Universität Pisa, Italien (Prof. Montanari)

Universität Stuttgart, IPVR (Prof. Levi)

Universität Tübingen (Prof. Zell)

University of California, Berkeley (Dr. Berthold)

University of Calgary (Prof. Prusinkiewicz)

University of Western Australia (Prof. Brauml)

Astrum GmbH, Erlangen

Audi AG, Ingolstadt

Basys GmbH, Erlangen

Bell Communications Research, Morristown, NJ, USA

Beusen GmbH, Berlin

Carl-Cranz-Gesellschaft eV, Oberpfaffenhofen

Deutscher Wetterdienst, Offenbach/Potsdam

Dürr Systems GmbH, Bietigheim-Bissingen

FAST eV, München

FZI Forschungszentrum Informatik, Karlsruhe (Dr. Berns)

Ina Wälzlager Schaeffler KG, Herzogenaurach

Infoteam GmbH, Bubenreuth

Institut d'Investigaci en Intelligencia Artificial, Bellaterra, Spanien (J. Puigsegur)

ModulaWare, Meolans Revel, Frankreich

Nureg, Nürnberg

Siemens A & D, Nürnberg

Forschungsergebnisse:

1 Synchronisation von Dateibäumen mit Hilfe von Graphgrammatiken

Das Problem des Abgleichs von Dateien auf mehreren Rechnern anhand ihrer Modifikationszeiten kann anschaulich mit Hilfe von Graphgrammatiken dargestellt werden. Der Vergleich der einzelnen Modifikationszeiten und der Zeitpunkte der letzten Synchronisation ergibt einen Graphen, aus dem mit einer Graphtransformation die durchzuführenden Aktionen ermittelt werden können. Auch miteinander in Konflikt stehende Änderungen können erkannt und damit behandelt werden.

Weitere Untersuchungen hierzu befassen sich mit den Problemen partieller Abgleiche, einer effizienten Implementierung des Algorithmus und der Auswirkungen, wenn sich Modifikationszeiten in die Vergangenheit ändern (z. B. beim Einspielen von Backups). Des weiteren soll untersucht werden, inwieweit das Erstellen des Aktionsgraphen und seine Ausführung parallelisiert werden können.¹⁾

(Hodek)

2 Verteilte Systeme mit Graphgrammatiken

Graphtransformationssysteme alleine eignen sich wenig zur Modellierung komplexer Anwendungsbeispiele. Je komplizierter die zu modellierende Anwendung ist, desto mehr spielen auch Markierungen von Knoten und Kanten eine entscheidende Rolle. Es wurden bereits eine ganze Reihe verschiedener Ansätze zum Markieren von Graphen entwickelt. Von besonderem Interesse war die Kombination der Ansätze von Schied und Schneider. Zusätzlich wurde das in Berlin entwickelte Konzept des Tygraphen hinzugefügt sowie Möglichkeiten des Änderns der Markierung ohne vorhergehendes Löschen des Knotens bzw. der Kante entwickelt. Auch die Behandlung von uneindeutigen Pushout-Complementen ließ sich vereinfachen. Zwei Anwendungsgebiete waren von besonderem Interesse. Zum einen wurde die Anwendbarkeit von Graphtransformationen auf Neuronale Netze, zum anderen auf Aktorsysteme untersucht. Für die graphische Modellierung der Aktorsysteme wurden die von M. Korff (Berlin) und D. Janssens (Antwerpen) gemachten Vorschläge gemischt. Die Antwerpener Modellierung enthält keinerlei Vorschläge, wie Markierungen und Regeln über den Markierungen sinnvoll benutzt werden können. Dies wiederum ist im Berliner Ansatz enthalten. Anstelle des Single-Pushout wird ein Doppel-Pushout zur Auswertung der Regeln verwendet. Es wurde eine aktorbasierte Programmiersprache entwickelt, deren Semantik mit Hilfe der Graphtransformationen beschrieben wird. Die Verbindung von Neuronalen Netzen und Graphtransformationen wurde bisher wenig untersucht. Die ersten Versuche einer Kombination gingen in eher theoretische Richtung. Wie lassen sich graphbasierte Regeln verwenden, um das Verhalten von Neuronalen Netzen zu beschreiben? Dies erweist sich als sehr einfach. So genügen zwei Regeln, den bekannten Backpropagation-Algorithmus darzustellen. Besonders geeignet sind Graphtransformationen allerdings für Topologie-verändernde Algorithmen. Hier wurde exemplarisch das "Dynamic-Decay-Adjustment" verwendet. Mit Hilfe der Modellierung ließ sich einfach zeigen, daß dieser Algorithmus terminiert und konvergiert. Ein anderes Anwendungsgebiet

¹⁾ Ausführlichere Informationen finden sich im World Wide Web unter <http://www2.informatik.uni-erlangen.de/IMMD-II/Research/Jahresbericht/1997/>

ist der Vergleich von "Real-Time-Backpropagation" und "Backpropagation-Through-Time". Ein Regelsatz zur Beschreibung des einen Formalismus läßt sich durch Veränderung nur zweier Regeln auf den anderen übertragen. Mit Hilfe dieser Regeln ließ sich auch zeigen, daß unter bestimmten Voraussetzungen beide Algorithmen dasselbe Ergebnis liefern. ²⁾

(Fischer)

3 Spezifikation graphischer Diagrammeditoren

Diagramme sind ein weit verbreitetes Mittel zur Darstellung und Vermittlung komplizierter Sachverhalte. Sie zeichnen sich durch ihre problemspezifisch graphische Darstellung und eine vorgegebene Syntax sowie Semantik aus. Diagramme mit gemeinsamer Syntax und Semantik bilden so problemspezifische Diagrammklassen. Zur Erstellung und Bearbeitung von Diagrammen mit einem Rechner werden graphische Editoren benötigt. Diagrammeditoren sind spezialisierte graphische Editoren, die auf eine bestimmte Diagrammklasse zugeschnitten sind.

DiaGen ist ein am Lehrstuhl in der Entwicklung befindliches Framework, das es ermöglichen soll, Diagrammeditoren für möglichst viele Diagrammklassen mit verhältnismäßig geringem Aufwand automatisch aus einer Spezifikation zu erstellen. Grundlage der Spezifikation ist die Definition der Diagrammsyntax und die Beschreibung des Editorverhaltens.

In diesem Jahr wurde das *DiaGen*-Projekt in drei Richtungen vorangetrieben:

- Eine komplette Neuimplementierung des bereits existierenden Diagrammeditorgenerators und des *DiaGen*-Laufzeitsystems wurde begonnen. Statt C++ wird nun Java verwendet.
- Statt des streng syntaxgesteuerten Ediermodells erlauben mittels *DiaGen* erzeugte Diagrammeditoren nun auch freihändiges Zeichnen. Um trotzdem die Fähigkeiten eines Diagrammeditors nutzen zu können, ist ein Parser nötig, der aus der graphischen Benutzereingabe korrekte Diagramme identifiziert sowie ihre syntaktische Struktur rekonstruiert [Minas97].
- Um auch dynamische Aspekte von Diagrammen (z. B. das "Token-Game" bei Petri-Netzen) in Diagrammeditoren darstellen und bearbeiten zu können, wurde damit begonnen, dynamische, animierte Diagrammklassen zu untersuchen [Minas,Gottschall97]. ³⁾

(Minas)

4 Ein Entwicklungssystem für zwei- und dreidimensionale graphische Benutzeroberflächen

²⁾ <http://www2.informatik.uni-erlangen.de/IMMD-II/Persons/Researchers/idfische/index.html>

³⁾ Nähere Informationen sind unter <http://www2.informatik.uni-erlangen.de/DiaGen> zu finden.

Das Lehrstuhl-Projekt *NGUIDE*, ein Entwicklungssystem für graphische Benutzeroberflächen, wird fortgeführt. Insbesondere wird die Erweiterbarkeit des Systems untersucht. Hierzu wird das System exemplarisch um die Möglichkeit ergänzt, auch dreidimensionale Benutzeroberflächen zu behandeln. Die Implementierung beruht auf der Seite der Darstellung auf *OpenGL*. Ereignisbehandlung, Layout und die Verwaltung der Eigenschaften der Dialogelemente können unverändert übernommen werden. Einige neue Dialogelementklassen werden entwickelt. Es werden auch interaktive Texturen unterstützt. Anwendungen liegen beispielsweise im industriellen Bereich bei der Programmierung von automatischen Fertigungsanlagen. Anstatt den Roboterarm vor Ort zu programmieren, ist es möglich, diese Programmierung mit Hilfe dreidimensionaler Benutzeroberflächen am Rechner vorzunehmen.

(Nilson)

5 Evolution und Visualisierung von Gestaltbildungsprozessen

Während eines einjährigen Postdoc-Aufenthalts am Department of Computer Science der University of Calgary, Kanada, wurde in Zusammenarbeit mit Prof. P. Prusinkiewicz ein Softwaresystem zur evolutionsbasierten Entwicklung von Lindenmayer-Systemen (L-Systeme) auf Graphik-Workstations (Silicon Graphics) entworfen. Dieses "Virtuelle Labor" ermöglicht eine Darstellung für Populationen von animierbaren, dreidimensionalen Pflanzenstrukturen, deren Wachstumsprozesse über L-Systeme beschrieben werden. Die L-Systeme sind ihrerseits als symbolische Ausdrücke kodiert und unterliegen Mutationsoperatoren im Sinne der genetischen Programmierung. Welche Pflanzen im Züchtungsprozeß verbleiben, wird entweder durch Fitneßfunktionen oder durch explizite Benutzerauswahl geregelt. Dieses System erleichtert den Entwurf von L-Systemen zu realen Pflanzenstrukturen erheblich und soll schließlich auch für die Simulation von Interaktionen in Pflanzengemeinschaften im Sinne kleiner Ökosysteme eingesetzt werden. Der L-SystemSimulator *CPFG* (Continuous Plant and Fractal Generator) der kanadischen Forschergruppe wurde dabei um die im Rahmen des *Evolvica*-Projektes ⁴⁾ am hiesigen Lehrstuhl entwickelten evolutionären Algorithmen erweitert.

(Jacob)

6 Simulation biologischer Neuronen

Biologische neuronale Netze können auf Basis einer mathematischen Beschreibung simuliert werden. Einer dieser Ansätze wurde an der Univ. Marburg entwickelt und wurde als Marburg-Modell bekannt. Das Marburg-Modell ist ein biologisch-orientierter Ansatz, um natürliche Neuronen nachzubilden, die Impuls-kodierte Signalübertragung verwenden. Mit diesem Neuronentyp lassen sich praktische Anwendungen, z. B. die Objektseparation, simulieren. Diese Anwendung wurde von uns mit *NeuroGraph* ⁵⁾, einer Soft-Computing-Entwicklungsumgebung, implementiert. Dazu mußte die mathematische Beschreibung des Marburg-Modells in eine algorithmische umgeformt werden und eine entsprechende Datenstruktur entwickelt werden.⁶⁾

4) <http://www2.informatik.uni-erlangen.de/~jacob/Evolvica>

5) URL: <ftp://ftp.informatik.uni-erlangen.de/pub/NeuroGraph/>

6) Details finden sich in <http://www2.informatik.uni-erlangen.de/~wilke/Publications/Wilke.Publications.html> unter

7 Produktionsoptimierung mit genetischen Algorithmen

Das Flow-Shop-Problem ist von großem wirtschaftlichen Interesse, und zahlreiche Ansätze sind veröffentlicht worden. Von einem pragmatischen Standpunkt aus besehen, sind viele dieser Ansätze für reale Problemstellungen wegen ihrer Rechenzeit- und Speicherplatzkomplexität nicht geeignet. Unter den neueren Ansätzen sind genetische Algorithmen, die einen einfacheren Zugang zur Problembeschreibung ermöglichen, aber ebenso Nachteile besitzen. In einem realen Beispiel, der Produktion von Fahrrädern, wurden die Einsatzmöglichkeiten untersucht.⁷⁾

(Wilke)

8 Verteilte Konfigurationsverwaltung⁸⁾

In größeren Softwareprojekten ist der Einsatz einer Versions- bzw. Konfigurationsverwaltung unerlässlich. Immer häufiger werden jedoch Entwicklungsaufträge an externe Firmen vergeben, so daß sich die Probleme bei der Konsistenzerhaltung und der parallelen Entwicklung verschärfen.

Zur Unterstützung des Entwicklungsprozesses wurde eine verteilte Konfigurationsverwaltung entworfen, die eine möglichst einfache, intuitive Architektur besitzen sollte. Dabei werden nicht nur einzelne Dokumente in ihren verschiedenen Versionen archiviert, sondern Projekte, Subprojekte, Projektstrukturen und Abhängigkeiten zwischen einzelnen Versionen.

Diese Architektur erlaubt es, die Projektstruktur/Verzeichnishierarchien beliebig zu verändern und alte Versionen dennoch wiederzufinden. Die einzelnen Versionsarchive können repliziert werden, ohne daß die Projektstruktur in allen Archiven geändert werden muß, wenn in einem Archiv die Struktur geändert wurde.

Zur Definition von Software-Varianten wurde ein neues Beschreibungsmodell entwickelt, das den "Produktraum", aus dem eine Variante ausgewählt werden kann, graphisch veranschaulicht. Dabei wird ein Attributbaum gebildet. Wenn in diesem Baum ein Attribut selektiert wird, so gelten auch alle Attribute auf dem Pfad zu diesem Attribut als selektiert. Auf diesen Attributen wird eine Ordnung definiert, die den Begriff der "allgemeineren" Attributierung zu einer gegebenen spezifischen formal handhabbar macht.

Die Kernfunktionalität der Konfigurationsverwaltung wird erweitert durch Module zum Change-Management, Software-Workflow und zur Softwareverteilung.

Wilke97a

7) Details finden sich in
<http://www2.informatik.uni-erlangen.de/~wilke/Publications/Wilke.Publications.html> unter
Wilke97b

8) Diese Arbeit wurde durch Mittel der Basys GmbH, Erlangen, sowie des Bundesministeriums für Forschung und Technologie (BMFT) unter Förderkennzeichen FKP0004402B7A unterstützt.

9 Requirements-Engineering in der Automatisierungstechnik (FORSOFT/Teilprojekt A4)

Der Forschungsverbund Software-Engineering (FORSOFT)⁹⁾ ist ein interdisziplinärer Forschungsverbund von Hochschulen und Industrieunternehmen mit dem Ziel einer Weiterentwicklung des Software-Engineering in enger Kooperation zwischen Wissenschaft und Praxis¹⁰⁾. Im Rahmen des Teilprojekts A4 beschäftigen wir uns mit anwendungsorientierter Forschung zum Requirements-Engineering in der Automatisierungstechnik. Wir entwickeln Vorgehensweisen und Techniken für die Ermittlung, Dokumentation und Verwertung von Anforderungen in zwei Bereichen. Im ersten Anwendungsbereich untersuchen wir in Kooperation mit unserem Industriepartner Siemens A&D AS den Entwicklungsprozeß für die Softwareentwicklungswerkzeuge der Produktfamilie SIMATIC S7. Dabei werden insbesondere die Probleme des Requirements-Engineering für Software, die auf einem globalen Markt verkauft wird, sowie der Versions- und Variantenentwicklung behandelt. Im zweiten Anwendungsbereich beschäftigen wir uns in Zusammenarbeit mit dem Institut für Werkzeugmaschinen und Betriebswissenschaften (iwb) (FORSOFT/Teilprojekt D1) mit der Softwareentwicklung für automatisierte Produktionsanlagen, d. h. es wird die Vorgehensweise der potentiellen Anwender des Automatisierungssystems SIMATIC S7 untersucht. Die Schwerpunkte sind hier die zeitliche Parallelisierung von Mechanik- und Softwareentwicklung sowie die Formalisierung der Anforderungsdokumentation zur Unterstützung der Kommunikation und Zusammenarbeit der Entwickler. Weitere Informationen sind unter ¹¹⁾ zu finden.

(Billing)

Habilitation:

Wilke, P.: Soft-Computing - Prinzip, Simulation und Anwendung -

Studienarbeiten:

Datschne, P.: Die Modellierung von Tree Adjoining Grammars mit Graphtransformationssystemen

Degen, M.: Graphlayout in der Programmiersprache Java

9) gefördert von der Bayerischen Forschungstiftung und Industriepartnern

10) <http://www.forsoft.de>

11) <http://www.forsoft.de/public/a4>

- Gericke, J.: Spezifikation und Implementierung einer Ereignisbehandlung für dreidimensionale grafische Benutzeroberflächen
- Hadler, H.: Evolutionäre Algorithmen in 'Java'
- Hoenig, T.: Entwicklung hybrider Softwaresysteme: Integration funktions- und objektorientierter verteilter Systeme mit Hilfe des CORBA-Standards
- Ittner, J.: Evolution von Entwicklungsprogrammen auf der Basis von Zellularautomaten
- Kleß, W.: Ein Editor für Graphersetzungsgregeln
- Knorr, O.: Entwicklung einer Java-Schnittstelle für die linguistische Arbeitsumgebung Malaga
- Marczinik, J.: Konzeption und Realisierung einer Auswertungskomponente für die Benutzungsoberflächen-evaluation
- Raschke, B.: Genetische Programmierung in der Programmiersprache Java
- Reichelt, D.: Design eines Laufzeitsystems für graphische Editoren und seine Implementierung in der Programmiersprache Java
- Winterstein, A.: Spezifikation und Implementierung eines Editors für grafische Benutzeroberflächen

Diplomarbeiten:

- Billstein, M.: Implementierung von Treibern für Neuro-Beschleunigerkarten für PCs mit PCI-Bus
- Binder, M. A.: Programmierung eines Graphersetzungssystems in der Programmiersprache Java
- Dauerlein, M.: Die Darstellung von Aktorsystemen mit Graphgrammatiken
- Gottschall, J.: Entwurf und Realisierung eines objektorientierten Frameworks zur Algorithmenanimation in der Programmiersprache Java

- Hopmann, J.: Simulierte Neurogenese auf Basis von Lindenmayer-Systemen
- Kleß, W.: Bereitstellung einer Schnittstelle für Berechnungsdienste im Internet
- Köhn, J.: Spezifikation und Implementierung von Dialogelementen für dreidimensionale grafische Benutzeroberflächen
- Krüger, J.: Entwicklung eines Graphersetzungssystems basierend auf dem Doppel-Pushout-Ansatz
- Lattka, M.: 'BlocksWorld' - Untersuchung zur Koevolution und Kommunikation künstlicher Organismen
- Maltz, W.: Implementierung einer Visualisierungskomponente für Genetische Algorithmen und Fuzzy-Logik
- Melchers, D.: Entwurf und Implementierung eines Internet-Ladens
- Mette, A.: Definition einer praxisorientierten Ablaufsprache gemäß IEC1131-3
- Mohr, V.: Portierung des GUI-Systems N-GUIDE auf das Betriebssystem MacOS
- Plückhahn, S. P.: Spezifikation und Implementierung eines Konfigurationsmanagers für grafische Benutzeroberflächen
- Schmölz, H. J.: Einsatz eines inkrementellen Hypergraph-Parsers in Diagrammeditoren
- Strauß, T.: Entwurf und Realisierung eines Computerprogramms zur Benutzerunterstützung bei der Parametrierung automatischer Lackieranlagen im Automobilbau
- Tietze, D.: Entwicklung genetischer Algorithmen für ein Mehrdipolmodell zur Lokalisierung von Gehirnströmen mit der Magnetoenzephalographie

Veröffentlichungen

- Berthold, M./
Fischer, I.: "Modelling Neural Networks with Graph Transformation Systems", Proc. of the IEEE International Joint Conference on Neural Networks, Houston, Texas, June 1997
- Fischer, I.: "Combining different Approaches to Labeling", Proc. of First GetGrats Workshop , Bordeaux, October 8 - 10, 1997
- Jacob, C.: "Evolutionary Algorithms with Mathematica". In: V. Keränen, P. Mitic und A. Hietamäki (Eds.), Innovation in Mathematics, Proc. IMS'97, 2nd International Mathematica Symposium, Rovaniemi, Finland, Computational Mechanics Publications, Southampton, UK, 1997, S. 263 - 272
- Jacob, C.: "Principia Evolvia - Simulierte Evolution mit Mathematica", dpunktVerlag, Heidelberg, 1997
- Keil, M.: "Wort für Wort - Repräsentation und Verarbeitung verbaler Phraseologismen", Niemeyer-Verlag, Reihe 'Sprache und Information', 1997
- Minas, M.: "Diagram editing with hypergraph parser support". In 'Proc. 1997 IEEE Symposium on Visual Languages (VL'97)', Capri, Italien, IEEE Computer Society Press, September 1997, S. 226 - 233
- Minas, M./
Gottschall, J.: "Specifying animated diagram languages". In: 'Proc. International Workshop on Theory of Visual Languages (TVL'97)', Capri, Italien, September 1997
- Nilson, J./
Kkai, G.: "PECG-GUI - A Graphical User Interface for an ECG Classifier System". In: G. Brewka, C. Habel, and B. Nebel, editors, KI-97: Advances in Artificial Intelligence, Freiburg, Germany, LNAI 1303, Springer-Verlag, September 1997, S. 397 - 400
- Wilke, P.: "Soft-Computing - Prinzip, Simulation und Anwendung". Habilitationsschrift, Erlangen, 1997. 432 S.
- Wilke, P./
Gröbner, M.: "Simulation of Marburg-Model Neural Networks using NeuroGraph". Proc. IMACS97, 1997
- Wilke, P./
Zobel, M.: "Optimization of Flow-Shop Problems using Evolutionary Algorithms - A Case Study". Proc. EUFIT97, 1997

Vorträge:

- Feder-Andres, C.: "Als Software-Ingenieur bei sd&m"
Informatik-Kolloquium der Univ. Erlangen-Nürnberg
27. 01. 1997

- Fischer, I.: "Modelling Neural Networks with Graph Transformations"
Graphgrammatik-Arbeitsgemeinschaft, Technische Universität Berlin
23. 05. 1997
- Fischer, I.: "Modelling Neural Networks with Graph Transformations"
IEEE International Conference on Neural Networks, Houston, Texas
11. 06. 1997
- Fischer, I.: "Neural Networks and Graph Transformation"
APPLIGRAPH-Workshop, Schloss Etelsen, Bremen
27. 06. 1997
- Fischer, I.: "Formalizing Neural Networks with Graph Transformations"
Neuro AG, Universität Karlsruhe
05. 09. 1997
- Fischer, I.: "Combining different approaches to labeling"
First Getgrats Workshop, Bordeaux
09. 10. 1997
- Göttler, H.: "Graphen in der Kunst, Picasso im Internet"
Festvortrag anl. des 60. Geburtstages von
Herrn Prof. Dr. H. J. Schneider, Erlangen
28. 02. 1997
- Jacob, C.: "Evolutionary Algorithms with Mathematica"
IMS'97, 2nd Intern. Mathematica Symposium, Rovaniemi, Finland
01. 07. 1997
- Jacob, C.: "Evolutionary Algorithms - An Overview" (Tutorial)
IMS'97, 2nd Intern. Mathematica Symposium, Rovaniemi, Finland
03. 07. 1997
- Jacob, C.: "Evolutionary Algorithms and Mathematica" (Tutorial)
GP'97, 2nd Intern. Genetic Programming Conf., Stanford University, Palo Alto, CA, USA
15. 07. 1997
- Jacob, C.: "Genetic Programming and the Evolution of Ecosystems"
Digital Burgess Conference, Banff, AB, Kanada
01. 09. 1997
- Janssens, D.: "ESM graph rewriting"
Informatik-Kolloquium der Univ. Erlangen-Nürnberg

29. 01. 1997

- Kkai, G.: "IDTS: Ein interaktiver Tester und Debugger für Prolog-Programme"
Informatik-Kolloquium der Univ. Erlangen-Nürnberg
27. 02. 1997
- Minas, M.: "Defining graphical object-oriented modelling methods by graph rewriting systems"
APPLIGRAPH-Workshop, Schloß Etelsen, Bremen
26. 06. 1997
- Minas, M.: "A visual modelling and simulation framework"
APPLIGRAPH Workshop, Schloß Etelsen, Bremen
27. 06. 1997
- Minas, M.: "Diagram Editing with Hypergraph Parser Support"
IEEE Symp. on Visual Languages (VL'97), Capri, Italien
25. 09. 1997
- Minas, M.: "Specifying animated diagram languages"
International Workshop on Theory of Visual Languages (TVL'97),
Capri, Italien
27. 09. 1997
- Schneider, H. J.: "Concurrent Graph Transformations
- Reflecting their use in designing distributed programs -"
TU Berlin
06. 03. 1997
- Schneider, H. J.: "A Categorical Interpretation of an Operational Approach to Graph Transformation"
Universität Krakau, Polen
29. 09. 1997
- Schneider, H. J.: "Graph Transformation as a Model of Concurrent Programming"
Universität Krakau, Polen
01. 10. 1997
- Wilke, P.: "Optimierung der Maschinenbelegung mit genetischen Algorithmen"
Habitationsvortrag, Universität Erlangen-Nürnberg
08. 01. 1997
- Wilke, P.: "Wenn Darwin produzieren würde - Ein Erbfall"
Universität Bamberg
18. 06. 1997

Wilke, P.: "Optimization of Flow-Shop Problems using Evolutionary Algorithms - A Case Study"
EUFIT 97, Aachen
29. 08. 1997

Wilke, P.: "Simulation of Marburg Model Neural Networks using NeuroGraph"
IMACS 97, Berlin
30. 09. 1997