

Lehrstuhl für Informatik II

Programmier- und Dialogsprachen sowie Compiler

Leiter: Prof. Dr. Hans Jürgen Schneider

Mitarbeiter:

Allendorf, Helmut, Dipl.-Ing. (FH)	(Programmierer)
Billing, Gunnar, Dipl.-Inf.	(wiss. Mitarb.) FORSOFT
Brütting, Agnes	(Sekretärin - halbtags) ab 01.03.99
Dormeyer, Ricarda, Dipl.-Inf.	(wiss. Mitarb.) ab 02.04.99 Mutterschafts- und Erziehungsurlaub
Fischer, Ingrid, Dr.-Ing.	(wiss. Mitarb.) 01.01.99-31.12.99 Sonderurlaub
Gröbner, Matthias, Dipl.-Inf.	(wiss. Mitarb.) ab 01.11.99
Hodek, Roman, Dipl.-Inf.	(wiss. Mitarb.)
Jacob, Christian, Dr.-Ing.	(wiss. Assistent) bis 31.07.99)
König, Roman, Dr.-Ing.	(wiss. Mitarbeiter) ab 27.04.1999
Kókai, Gabriella, Dr.-Ing.	(wiss. Mitarb.)
Lühns, Erni	(Sekretärin – halbstag) bis 28.02.99
Minas, Mark, Dr.-Ing.	(Akad. Rat)
Schmidt, Harald, Dipl.-Inf.	(wiss. Mitarb.) ab 01.10.99
Schörmal, Elfriede	(Sekretärin - halbtags)
Uebler, Manfred	(Programmierer)
Wilke, Peter, PD Dr.-Ing. habil.	(Akad. Oberrat) Sonderurlaub ab 01.11.99

Gastwissenschaftler:

Volle, Viktor, Dipl.-Inf. (BMFT)

Lehrbeauftragte:

Hindel, Bernd, Dr.-Ing. (Fa. 3SOFT, Erlangen)

Kips, Detlef, Dr.-Ing. (Fa. BASYS, Erlangen)

Kreisel, Klaus, Dr.rer.nat. (Emmy-Noether-Gymnasium, Erlangen)

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)

József Attila Universität Szeged, Ungarn (Dr. Gyimóthy)

Technische Universität Berlin (Prof. Dr. Ehrig)

Technische Universität München (Prof. Dr. Broy)

Universität Antwerpen, Belgien (Prof. Dr. Janssens)

Universität Bremen (Prof. Kreowski)

Universität der Bundeswehr, Neubiberg (Prof. Dr. Schürr)

Universität Gent (Dr. Bellegham)

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. Bräunl)

Astrum GmbH, Erlangen
Audi AG, Ingolstadt
Basys GmbH, Erlangen
Deutscher Wetterdienst, Offenbach/Potsdam
FAST eV, München
FZI Forschungszentrum Informatik, Karlsruhe (Dr. Berns)
imbus GmbH, Erlangen
Ina Wälzlager Schaeffler KG, Herzogenaurach
Infoteam GmbH, Bubenreuth
Nureg, Nürnberg
Siemens A & D, Nürnberg
3SOFT GmbH, Erlangen

Forschungsergebnisse:

I. Requirements Engineering in der Automatisierungstechnik

Der Forschungsverbund Software Engineering FORSOFT ist ein interdisziplinärer Forschungsverbund von Hochschulen und Industrieunternehmen mit dem Ziel einer Weiterentwicklung des Software Engineering in Wissenschaft und Praxis (www.forsoft.de). Das Teilprojekt A4 "Requirements Engineering in der Automatisierungstechnik" bearbeitet zwei sich ergänzende Themenbereiche. Zum einen wird das Requirements Engineering komplexer Standardsoftware am Beispiel der Automatisierungstechnik und andererseits das Requirements Engineering automatisierter Produktionsanlagen behandelt. Dabei wird eng mit den Industriepartnern Siemens A&D AS, HOMAG AG und BMW AG zusammengearbeitet. Für das Requirements Engineering komplexer Standardsoftware wurde eine neuartige Methodik entwickelt und erprobt, die alle Aktivitäten von der Sammlung bis zur Verwertung von Anforderungen systematisch unterstützt. Diese Methodik umfaßt die Definition eines Vorgehensmodells, eines Datenmodells der zu verarbeitenden Daten und die Integration von Erfassungsmethoden für Anforderungen. Veröffentlichungen auf mehreren internationalen Workshops haben unter anderem dazu geführt, daß die Thematik von etablierten Workshops als Hauptthema formuliert wurde. Insbesondere wurde gemeinsam mit anderen internationalen Universitäten ein neuer Workshop im Bereich von Enterprise Resource Planning organisiert. Für das Requirements Engineering automatisierter Produktionsanlagen wurde in enger Kooperation mit dem Institut für Werkzeugmaschinen und Betriebswissenschaften der TU München (iwb) eine Methodik entwickelt, die über die Fachrichtungen hinweg eine

eindeutige Rollenverteilung mit klaren Schnittstellen definiert. Die Methodik ermöglicht einen frühzeitigen Beginn der Entwicklung des Automatisierungssystems und erlaubt ein Maximum an Parallelität in der Planung und Entwicklung. Zur Spezifikation, Visualisierung, Konstruktion und Dokumentation wurde eine integrierte Modellierungssprache für die Teilsysteme einer Produktionsanlage auf Grundlage moderner Beschreibungstechniken (z.B. UML, STEP/EXPRESS) entwickelt. Damit können wesentliche Nachteile der heute eingesetzten, sequentiellen Vorgehensweisen vermieden werden. Die Ergebnisse wurden auf verschiedenen praxisnahen Konferenzen und Messen einem umfassenden Fachpublikum des Anwendungsbereichs mit sehr positiver Resonanz präsentiert.

(Billing)

II. Die automatische Analyse von Idiomen

Die interessantesten und schwierigsten idiomatischen Beispiele enthalten lexikalische Variationen und syntaktische Modifikationen. Aber auch unveränderliche Idiome sollten bei der automatischen Sprachanalyse nicht vernachlässigt werden. Ein Mechanismus zur automatischen Interpretation von Idiomen muss die gesamte Bandbreite von Phänomenen abdecken und auch entscheiden können, warum bei einem Idiom eine bestimmte Modifikation oder Variation möglich ist und bei einem anderen nicht. Typische Beispiele sind "einen Korb geben, bekommen, einfangen, einhadeln, kriegen, austeilen,..." oder "The agency took the researcher under its well-muscled wing."

Unser Ansatz beruht auf zwei Vermutungen von Raymond Gibbs (*Poetics of the Mind*, 1994) und George Lakoff (*Woman, Fire and Dangerous Things*, 1987). Erstens können Idiome analysierbar sein, das heisst die einzelnen Komponenten können bei der Analyse des Idioms eine Rolle spielen, sie müssen es aber nicht. Zweitens sind Idiome nicht, wie oft behauptet "tote Metaphern", Metaphern spielen bei vielen Idiomen eine aktive Rolle im Verstehensprozess, besonders aber bei Idiomen die analysierbar und modifizierbar sind. Diesen Aussagen kann mit folgendem fünfteiligen System Rechnung getragen werden:

1. Zum einen wird eine "construction grammar" benutzt (A. Goldberg: *Constructions*, 1995), eine Sammlung von sogenannten Constructions, die in einheitlicher Weise das mentale Lexikon, die Idiomliste und die Grammatik darstellen. Jede dieser Constructions enthält wohlgeformte Bedingungen über verschiedene linguistische Domänen wie Syntax, Semantik und Pragmatik.
2. Ein auf Vererbung basierendes semantisches Netz dient der Repräsentation von Situationen und ihren Teilnehmern.

3. Der aktuelle Kontext kann nach Referenten abgefragt werden.

3. Metaphorische Abbildungen zwischen einer Start- und einer Zieldomäne werden aktiviert, wenn Konzepte beider Domänen vorhanden sind (Srinu Narayan: KARMA: Knowledgebased Action Representations for Metaphor and Aspect, Dissertation, Universität Berkeley).

4. Die Simulationsspezifikation dient als Schnittstelle zwischen den Constructions und einer auf Petri-Netzen basierenden Simulation (Srinu Narayan: KARMA: Knowledge-based Action Representations for Metaphor and Aspect, Dissertation, Universität Berkeley). In der Simulation werden die Angaben der Spezifikation wenn möglich ausgeführt und der Kontext aktualisiert. Idiome werden als Constructions spezifiziert, die die nötigen metaphorischen Abbildungen zwischen den beiden verwendeten Domänen auslösen. Mit Hilfe des semantischen Netzes und der Simulation werden mögliche Modifikationen und Variationen überprüft.

Die Arbeit wurde finanziert durch ein Stipendium des DAAD aus Mitteln des Gemeinsames Hochschulsonderprogramm III von Bund und Ländern. Sie wurde durchgeführt am International Computer Science Institute in Berkeley, Kalifornien in der Arbeitsgruppe Neural Theory of Language unter der Leitung von Prof. J. Feldmann und Prof. G. Lakoff.

(Fischer)

III. Synchronisation von Dateibäumen mit Hilfe von Graphgrammatiken

Bei den Arbeiten zum Abgleich von Dateibäumen wurde weiter an der Lösung von Detailproblemen gearbeitet.

Zur Aufgabenstellung: 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.

Aktuelle Entwicklungen befassen sich u.a. mit dem Einspielen von Backups, die ja "alte" Modifikationszeiten tragen, jedoch als neuere Versionen zu behandeln sind. Die dadurch entstehenden Probleme werden mit einer Ausnahmeliste umgangen, die bei Bedarf sog. Virtuelle Modifikationszeiten zu einzelnen Dateien speichert. Diese Ausnahmeliste leistet auch gute Dienste bei der Vermeidung von Kopierschleifen, die entstehen können, wenn einzelne Rechner zeitweise vom Netz getrennt sind und nicht an den Abgleichen teilnehmen.

Die Synchronisationen finden nun automatisch im Hintergrund statt, ohne dass sie manuell angestossen werden müssen. Auch die Behandlung von Konflikten wurde erweitert. Unter bestimmten Bedingungen ist sogar das automatische Zusammenmischen unabhängiger Änderungen möglich.

(Hodek)

IV. Beschreibung der Blutgefäße des Auges durch parametrisierte Lindenmayer-Systeme mit Hilfe evolutionärer Algorithmen

Wir entwickeln in Kooperation mit der Augenklinik der „Albert-Szent-Györgyi-Universität“ Szeged das GREDEA-System (Grammatical Retina Description with Evolutionary Algorithms).

Der Grundgedanke hinter GREDEA ist, patientenspezifische Überwachungsprogramme für das Überprüfen der Blutzirkulation der menschlichen Retina zu entwickeln. Das System wird bei Patienten mit Diabetes angewandt, die über lange Zeiträume überwacht werden müssen, da ihre Retina besonders gefährdet ist. Betroffen sind vor allem die kleinen Blutgefäße, die diese Membran ernähren. Im frühen Stadium treten meist noch keine Symptome auf, jedoch ist es notwendig, dass ein Facharzt die erweiterten Pupillen der Patienten regelmäßig untersucht.

Unser System hat im wesentlichen zwei Teile: Im ersten Teil, den man als Lernphase betrachten kann, wird eine individuelle Beschreibung der Blutzirkulation der menschlichen Retina für jeden Patienten erstellt. Um diese Beschreibung zu erhalten, beginnt der Prozess bei den Fundusbildern. Diese Bilder werden mit einem „scanning laser ophthalmoscop“ (SLO) aufgenommen und mit Hilfe von verschiedenen Bildverarbeitungsmethoden vorverarbeitet.

Dann wird ein parametrisches Lindenmayer-System (L-System) mit Hilfe von evolutionären Algorithmen entwickelt. Diese Grammatik beschreibt ein Muster, das dem Gefäßbaum des Patienten sehr nahe kommt. Dieses L-System kann gespeichert werden (es braucht weniger Speicher als die ursprüngliche Abbildung) und später benutzt werden, um zu vergleichen.

Der zweite Teil ist dann die Anwendungsphase. Der Patient kommt erneut, um Untersuchungen vornehmen zu lassen und wieder werden Bilder von seinem Auge gemacht. Von diesen werden wiederum L-Systeme generiert. Aber jetzt dient das vorher generierte Ergebnis als Anfangspopulation des evolutionären Algorithmus. Als Ergebnis bekommen wir erneut ein L-System. Diese symbolische, parametrisierte Beschreibung kann mit früheren Ergebnissen verglichen werden und als Eingabe eines Diagnostik-Moduls dienen.

V. *DiaGen* - Ein Generator für graphische 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.

Im vergangenen Jahr wurde die Forschung in diesem Themengebiet in drei Richtungen vorangetrieben:

1. In Fortsetzung vorangegangener Arbeiten nutzen wir Hypergraphen als formale Modelle, um Diagramme in graphischen Editoren zu repräsentieren. In mehreren Anwendungsfällen hat es sich als vorteilhaft erwiesen, eine Idee aus dem herkömmlichen Compilerbau zu adaptieren; an Stelle eines einzelnen Hypergraph-Modells, das unmittelbar ein Diagramm repräsentiert und gleichzeitig Gegenstand der syntaktischen Analyse ist, kommen nun zwei Hypergraph-Modelle zum Einsatz. Das diagrammnahe Modell, das die einzelnen Diagrammkomponenten und ihre Beziehungen zueinander darstellt, wird einer *lexikalischen Analyse* unterzogen, in deren Verlauf ein reduziertes Hypergraphmodell entsteht. Dieses wiederum erlaubt eine deutlich effizientere *syntaktische* und, darauf basierend, *semantische Analyse*. Diagramme, die in einem Editor wie in einem Zeichenprogramm "freihändig", d.h. ohne Vorgabe fester Editieroperationen editiert werden dürfen, können so effizient analysiert und ihre Information weiterverwendet werden.
2. „*Freihändige*“ Editoren haben den Nachteil, daß sie dem Benutzer nur elementare Möglichkeiten zum Editieren von Diagrammen anbieten. Man kann Komponenten einfügen, verändern und löschen. Dagegen stellen *syntaxgesteuerte* Editoren teilweise komplexe Editieroperationen zur Verfügung, die größere Diagrammodifikationen in einem Schritt erlauben. Zudem haben sie den Vorteil, den Benutzer beim Editieren „an die Hand zu nehmen“, indem nur sinnvolle Veränderungen am Diagramm angeboten werden. Dies ist aber häufig auch ein Nachteil: Fast immer ist die Menge zur Verfügung stehender Operationen unbefriedigend und die Durchführung einzelner Diagrammodifikationen mit den existierenden Operationen zu umständlich. Im letzten Jahr wurde deshalb ein Konzept für Diagrammeditoren entwickelt, die Diagramme sowohl freihändig als auch syntaxgesteuert zu editieren gestatten. Damit werden die Vorteile beider Modi nutzbar: wo

keine geeigneten syntaxgesteuerten Editieroperationen zur Durchführung gewünschter Diagrammmodifikationen zur Verfügung stehen, können Diagramm „freihändig“ editiert werden.

3. Das zuvor umrissene zweistufige Konzept der Diagrammanalyse und die Kombination des freihändigen mit dem syntaxgesteuerten Editierkonzept wurde im Rahmen einer Neukonzeption des schon existierenden *DiaGen*-Systems¹ in Java 2 realisiert. Das System besteht aus einem Editorframework und einem Programmgenerator, der aus einer textuellen Spezifikation einen gleichzeitig freihändigen und syntaxgesteuerten Editor auf der Basis des zugrundeliegenden Frameworks erzeugt. (13+14)

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

(Minas)

VI. Anwendung Neuronaler Netze

Mit dem „Centre for Intelligent Information Systems Processing“ an der University of Western Australia, Australien, wurde eine Forschungskoooperation zu praktischen Anwendungen von Soft-Computing-Systemen etabliert. Schwerpunkte werden die Einsatzmöglichkeiten in der Robotik und der Optimierung sein.

Schichtplanungsprogramme benutzen Zeitreihen-Vorhersagen, um den benötigten Bedarf an Arbeitskräften zu planen. Wir haben diese Problematik für Call-Center untersucht, die ein bestimmtes Service-Level halten müssen, d.h. wieviele Anrufe nach welcher Zeit beantwortet sein müssen. Dabei soll die Planung den Personaleinsatz und -dienstzeiten möglichst kostenoptimal gestalten.

Wir haben gezeigt, wie Künstliche Neuronale Netze benutzt werden können, um Vorhersagen für die Anzahl der Anrufe und deren Bearbeitungszeit aufgrund von Daten aus der Vergangenheit zu machen. Das entwickelte Neuronale Netz reflektiert die Datensequenz in der Dimension Zeit und die besten Ergebnisse wurden mit Netzen erzielt, die eine Auflösung von Tagen und Wochentagen hatten. Die Vorhersage der Bearbeitungszeit ist eine Vorhersage der mittleren zu erwartenden Bearbeitungszeit. Die Anzahl der Anrufe wird mit einer um 10% besseren Güte vorhergesagt als mit konventionellen Methoden. Obwohl die Ergebnisse der neuronalen Netze besser sind als die der nicht-lernenden System, ist dies noch nicht befriedigend, da die auftretenden Streuungen zum Teil erheblich sind. Wir erklären uns dies mit nicht modellierten Parametern, die das Verhalten beeinflussen. Für intelligente Roboter in Multi-Agenten-Systemen ist Kommunikation ein essentieller Faktor für kooperatives Verhalten. Als Anwendung diente uns ein Team von fußballspielenden Robotern (RoboCup) auf Basis der EyeBot-Plattform.

Der Vergleich von bereits implementiertem System zeigt die Problemabhängigkeit der Kommunikationsstruktur von der Anwendung. Es wurde daher eine auf die EyeBot-Plattform und Multi-Agenten-Systeme zugeschnittene Kommunikation entwickelt, die auf einer drahtlosen Übermittlung spezieller Datenpakete basiert. Besondere Merkmale sind die Selbst-Konfiguration und Fehler-Behandlung sowie die Möglichkeit individueller und Broadcast-Übertragungen. Die Steuerung des Multi-Agenten-Systems erfolgt durch einen in den Kommunikationsverbund integrierten Computer, auf dem dann das Soft-Computing-System ausgeführt wird, was wegen seines Raum-/Zeit-Bedarfes nicht lokal auf den Robotern erfolgen kann.

(Wilke)

Gäste:

Prof. Wit Forys, Jagiellonische Universität Krakau

Veröffentlichungen:

1. *Bardohl, R., Minas, M., Schürr, A., and Taentzer, G.:* „Application of graph transformation to visual languages.“ In: H. Ehrig, G. Engels, H.-J. Kreowski, and G. Rozenberg, editors, Handbook of Graph Grammars and Computing by Graph Transformation, Vol. II: Applications, Languages and Tools, pp. 105--180. World Scientific, 1999.
2. *Billing, G., Mauderer, M.:* „Enhancing Efficiency and Transparency of Manufacturing System Planning through Comprehensive and Integrated Tool Support, The 15th International Conference on Computer-Aided Production Engineering CAPE '99, Durham, 4/99 (BM99a)
3. *Billing, G., Mauderer, M.:* „Integrated Planning of Transfer Lines“, Werkzeugpräsentation, Internationale Werkzeugmaschinenmesse EMO, Paris, 5/99 (BM99b)
4. *Billing, G., Mauderer, M.:* „Integrated Planning and Development of Manufacturing Systems“, 15th International Conference on Production Research ICPR-15, Limerick, 8/99 (BM99c)
5. *Chandler, S. und Jacob, C.:* „Automata Containing Evolutionary Algorithms: Behavior and Learning Under Law“, IMS-99 Online Proceedings, Third International Mathematica Symposium

6. *Fischer, I.*: „Describing Neural Networks with Graph Transformations“, Arbeitsberichte des Instituts für Mathematische Maschinen und Datenverarbeitung, Band 31, Nummer 11, 1999.
7. *Fischer, I.; Koch, M., Berthold, M.R.*: „Learning and Rewriting, in Fuzzy Rule Nets in Tagungsband Applications of Graph Transformation with Industrial Relevance“ (AGTIVE'99), Kloster Rolduc, NL, 1.-3. September 1999
8. *Fischer, I., Taentzer, G.*: „Modellierung dynamischer Rekonfiguration in offenen, verteilten Systemen auf der Basis von Graphtransformation, in Tagungsband Formale Beschreibungstechniken fuer verteilte Systeme“ (FBT99), TU München, 1999
9. *Jacob, C.*: "Stochastic Search Methods". In: Michael R. Berthold und David J. Hand (Ed.), An Introduction to Intelligent Data Analysis, Springer Verlag, Berlin, 1999, Kapitel 9, S. 299-350.
10. *Jacob, C.*: „Evolution and Co-Evolution of Developmental Programs“ Computer Physics Communications 121-122, Special Issue: Modeling Collective Phenomena in the Sciences, S. 46-50 (1999)
11. *Kókai, G., Tóth, Z., Ványi, R.* : „Modelling Blood Vessels of the Eye with Parametric L-Systems Using Evolutionary Algorithms“. In the Proc Joint European Conference on Artificial Intelligence in Medicine and Medical Decision Making, AIMDM'99, 20th - 24th June 1999 Aalborg, Denmark published by Springer-Verlag LNCS series 1620 433-443
12. *Kókai, G., Ványi, R., Tóth, Z.*: „Parametric L-System Description of the Retina with Combined Evolutionary Operators“. In Proc. GECCO, Genetic and Evolutionary Computation“, Conference July 13 - 17, 1999 Orlando, Florida, USA, Vol. 2 1588-1596
13. *Kókai, G., Tóth, Z., Ványi, R.*: „Generic Evolution Algorithms Programming Library In Proc“. GECCO, Genetic and Evolutionary Computation“, Conference July 13 - 17, 1999 Orlando, Florida, USA, Vol. 2 1867
14. *Kókai, G., Tóth Z., Ványi, R.*: “Evolving Artificial Trees Described by Parametric L-system“ In Proc. IEEE Canadian Conference on Electrical & Computer Engineering Shaw Conference Centre, Edmonton, Alberta, Canada May 9 - 12, 1999, 1722-1728
15. *Kókai, G., Nilson, J., Niss, C.*: „GIDTS - A Graphical Programming Environment for Prolog“ In the Proc ACM SIGPLAN-SIGSOFT Workshop on Program Analysis for Software Tools and Engineering (PASTE'99) September 6-10, 1999, Toulouse, France, 95-105, ACM, in press

16. *Minas M.*: „Creating semantic representations of diagrams“. In: Proc. of the International Workshop on Applications of Graph Transformation with Industrial Relevance (AGTIVE'99) at Monastery Rolduc, NL, 1999. Erscheint in LNCS, April 2000.
17. *Minas, M. and Köth, O.*: „Generating diagram editors with diagen.“ In: Proc. of the International Workshop on Applications of Graph Transformation with Industrial Relevance (AGTIVE'99) at Monastery Rolduc, NL, 1999. Erscheint in LNCS, April 2000.
18. *Taentzer, G., Koch, M., Fischer I., Volle V.*: “Distributed Graph Transformation with Application to Visual Design of Distributed Systems“. Handbook of Graph Grammars and Computing by Graph Transformation III: Concurrency, Parallelism, and Distribution (Hrsg.: G. Rozeberg et al.), World Scientific, 1999, S. 269 - 341
19. *Schneider, H.J.*: „Describing systems of processes by means of high-level replacement“. Handbook of Graph Grammars and Computing by Graph Transformation III: Concurrency, Parallelism, and Distribution (Hrsg.: G. Rozenberg et al.), World Scientific, Singapore, 1999, pp. 401-450
20. *Schneider, H.J./Ehrig, H. (Hrsg.)*: „Applied Categorical Structures“ Vol. 7, No. 4 (1999): Special Issue on Categorical Approaches to Computing by Graph Transformation

Vorträge:

- Fischer, I.*: „*Learning and Rewriting in Fuzzy Rule Nets, Applications of Graph Transformation With Industrial Relevance*“, (AGTIVE'99) Kloster Rolduc, NL, 2. September 1999
- Jacob, C.*: "*Nature´s Ways of Programming*", ASCI'99, Fifth Annual Conference of the Advanced School for Computing and Imaging, Heijen, Niederlande, 15. Juni, 1999
- Jacob, C.*: "*Lindenmayer systems and growth program evolution*", GECCO 99, Genetic and Evolutionary Computation Conference, Workshop on Advanced Grammar Techniques within Genetic Programming and Evolutionary Computation, Orlando, Florida, July 13, 1999
- Jacob, C.*: „*Wie programmiert die Natur? - Ein Streifzug durch die Natur aus dem Blickwinkel der Informatik*“, Tag der Erlanger Informatik 1999

- König, R.: „*Codes and Factorizations of Finite Cyclic Groups*“, Second International Conference on Semigroups, St. Petersburg, Russland, 06.07.99
- König, R.: Lehraufenthalt im Rahmen des Socrates-Erasmus-Programms an der Università degli studi di Cagliari, Italien 13.-03. – 10.04.99
- Kókai, G.: „*Error diagnosis an overview of the existing debugging techniques*“, 3Soft GmbH, Erlangen 20.01.1999
- Kókai, G.: „*Monitoring and debugging of distributed real time systems*“, 3Soft GmbH, Erlangen 05.02.1999
- Kókai, G.: „*IDTS: Ein System zur automatischen Testen und Debuggen von PROLOG-Programmen*“ ER-INF'99 Tag der Erlangener Informatik 30.04.1999
- Kókai, G.: „*IEEE Canadian Conference on Electrical and Computer Engineering Shaw*“, Conference Centre, Edmonton, Alberta, Canada AIMDM'99 09.05 – 12.05.1999
- Kókai, G.: „*Joint European Conference on Artificial Intelligence in Medicine and Medical Decision Making*“, 9, Aalborg, Denmark 20.06. – 24.06.1999
- Kókai, G.: „*GECCO, Genetic and Evolutionary Computation Conference*“, Orlando, Florida, USA 13.07. – 17.07.1999
- Kókai, G.: „*ACM SIGPLAN-SIGSOFT Workshop on Program Analysis for Software Tools and Engineering*“, (PASTE'99) Toulouse, France
- Minas, M.: Workshop Spezifikationstechniken (Erlangen) im Rahmen des DFG-Schwerpunktprogramms „Rapid Prototyping für integrierte Steuerungssysteme mit harten Zeitbedingungen“: *Spezifikation von Diagrammsprachen und -editoren mit Hypergraphgrammatiken* 19.02.1999
- Minas, M.: GRACE-Treffen (Erlangen): „*Modul-Klassen in GRACE*“ 23.02.1999

- Minas, M.: AGTIVE Workshop 1199 (Subarea Meeting on Tools, Rolduc, Kerkrade, The Netherlands): „*Computational Category Theory with Java and Applications to Graph Transformation*“
31.08.1999
- Minas, M.: AGTIVE Workshop 1999 (Rolduc, Kerkrade, The Netherlands): „*Generating Diagram Editors with DiaGen*“ (Systemdemonstration)
01.09.1999
- Minas, M.: AGTIVE Workshop 1999 (Rolduc, Kerkrade, The Netherlands): „*Creating semantic representations of diagrams*“
02.09.1999
- Minas, M.: Firmenkolloquium der 3soft GmbH (Tennenlohe): „*Automatisierte Erstellung graphischer Editoren*“
19.11.1999:
- Wilke, P.: The University of Western Australia, Dep. of Electrical and Electronic Engineering, ENTRE FOR INTELLIGENT INFORMATION PROCESSING SYSTEMS: „*Neural Networks: Artificial Brain or Dr Frankenstein's Creature?*“
05.10.1999
- Wilke, P.: The University of Western Australia, Dep. of Electrical and Electronic Engineering, ENTRE FOR INTELLIGENT INFORMATION PROCESSING SYSTEMS: „*Simulation of Neural Network: ... or the dark side of computational power*“
05.10.1999
- Wilke, P.: The University of Western Australia, Dep. of Electrical and Electronic Engineering, ENTRE FOR INTELLIGENT INFORMATION PROCESSING SYSTEMS: „*Evolutionary Algorithms: ... and in the semi-finals you are going to meet*“
07.10.99
- Wilke, P.: The University of Western Australia, Dep. of Electrical and Electronic Engineering, ENTRE FOR INTELLIGENT INFORMATION PROCESSING SYSTEMS: „*Evolutionary Programming: „If Darwin had been a programmer*“
07.10.99

Studienarbeiten:

Andres Anna-Maria	Rechnen mit DNA Studienarbeit zum Themaberreich DNA Computing
Bickermann Gabriele	Ein visuelles diff-/merge Tool in der Programmiersprache Java
Bischewski Stefan	Erweiterung der Maschinenbefehlsanordnung im GNU-C-Compiler
Degen Markus	Graphlayout in der Programmiersprache Java
Eckstein Christian	Implementation eines Attributierungsschemas für Varianten in einem Versionsarchiv
Gast Daniel	Konzepte zur Entwicklung von Benutzeroberflächen eines technischen Berechnungsprogrammes für das Internet
Köth Oliver	Entwurf und Implementierung eines generischen graphischen Editors
Güngör Mesut	Die Re-Implementation der Oberfläche einer Datenbank in der Programmiersprache Java
Mihaylov Kristian	Optimierungsalgorithmen für Multi-Agenten-Systeme

Diplomarbeiten:

Bloß Jürgen	Entwurf und Implementierung eines Client/Server-Systems für ein Versionsarchiv
Bosch Christian	Statische Codeanalyse von PROLOG-Prgrammen
Foldenauer Ralf	Entwurf und Implementierung einer linguistischen Datenbank in der Programmiersprache Java
Glaser Günter	Zeitreihenvorhersage mit künstlichen neuronalen Netzen am Beispiel der Personalbedarfsprognose eines Call Centers
Gast Daniel	XML-basierte Definition und Verwaltung von Software Entwicklungsprozessen
Hofer Joachim	Werkzeuggestützte Java-Programmierung
Knorr Oliver:	Rechteverwaltung im Rahmen einer Konfigurationsverwaltung
Niss Christian	Evolution von Kooperation und Konkurrenz – Modellierungen mit zellulären Automaten