

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)	
Dormeyer, Ricarda, Dipl.-Inf.	(wiss. Mitarb.)	
Fischer, Ingrid, Dipl.-Inf.	(wiss. Mitarb.)	
Hodek, Roman, Dipl.-Inf.	(wiss. Mitarb.)	
Jacob, Christian, Dr.-Ing.	(wiss. Assistent)	
Lühns, Erni	(Sekretärin - halbtags)	
Minas, Mark, Dr.-Ing.	(Akad. Rat)	
Nilson, Jörg, Dr.-Ing.	(wiss. Mitarb.)	bis 30. 06. 98
Schörmal, Elfriede	(Sekretärin - halbtags)	
Uebler, Manfred	(Programmierer)	
Wilke, Peter, PD Dr.-Ing. habil.	(Akad. Oberrat)	

Gastwissenschaftler:

Kkai, Gabriella, Dipl.-Inf.	(Bay. Kultusministerium, KAAD)
Volle, Viktor, Dipl.-Inf.	(BMFT)

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)

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)

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

Beusen GmbH, Berlin

Bissantz und Küppers, Nürnberg

Carl-Cranz-Gesellschaft eV, Oberpfaffenhofen

Deutscher Wetterdienst, Offenbach/Potsdam

FAST eV, München

FZI Forschungszentrum Informatik, Karlsruhe (Dr. Berns)

imbus GmbH, Erlangen

Ina Wälzlager Schaeffler KG, Herzogenaurach

Infoteam GmbH, Bubenreuth

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

Nureg, Nürnberg

Siemens A & D, Nürnberg

3SOFT GmbH, Erlangen

Forschungsergebnisse:

1 Graphgrammatikalische Beschreibung von Prozesssystemen

Graphen zur Darstellung von statischen Strukturen und dynamischen Vorgängen haben ihren festen Platz sowohl in der theoretischen als auch in der praktischen Informatik, aber auch in vielen anderen Wissenschaften. Seit Beginn der siebziger Jahre beschäftigt sich der Lehrstuhl intensiv mit formalen Graphtransformationssystemen, das sind Regelsysteme, die die Gewinnung neuer Graphen aus bereits bekannten beschreiben. Die Beschreibung dynamischer Veränderungen ist damit unmittelbar einsichtig. Diese Ideen haben weltweite Resonanz gefunden; an verschiedenen Stellen wurden Weiterentwicklungen vorgenommen und Alternativen untersucht.

Ein für theoretische Untersuchungen sehr fruchtbarer Ansatz wurde von uns 1973 vorgestellt. Er benutzt die Pushout-Konstruktion der Kategorientheorie, um den Begriff der Konkatenation von Zeichenreihen zu verallgemeinern. Die Ableitbarkeit wird mit einem Doppelpushout beschrieben, indem sowohl die linke als auch die rechte Seite einer Ersetzungsregel längs eines Klebegraphen mit dem verbleibenden Kontext verbunden werden. Dies ist eine geradlinige Verallgemeinerung des Chomskyschen Ableitbarkeitsbegriffes.

Im Berichtszeitraum stand die Anwendung dieser Technik auf die Beschreibung asynchroner Prozesssysteme im Vordergrund. In der Literatur gibt es eine Vielzahl von graphbasierten Techniken zur Beschreibung von Prozesssystemen, z. B. Petri-Netze, Statecharts, aber auch Teilaspekte im Bereich der Datenbanksysteme, der logischen Programmierung, der Termersetzung etc. werden gerne auf der Basis von Graphen beschrieben. In einer zusammenfassenden Darstellung wird gezeigt, dass sich alle diese formalen bzw. halbformalen Hilfsmittel als Sonderfälle der Graphtransformationssysteme auffassen lassen. Ein wesentlicher Vorteil dieser Darstellung ist, dass dadurch die Untersuchung von Prozesseigenschaften (z. B. parallele Unabhängigkeit der Teiloperationen) in den unterschiedlichen Modellen vereinheitlicht wird.

(Schneider)

2 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.

Die aktuellen Arbeiten hierzu befassen sich mit der Erhaltung der Modifikationszeit auf allen Zielrechnern (soweit dies möglich ist) und der Vermeidung von Kopierschleifen. Des weiteren wurde ein Algorithmus entwickelt, der die auszuführenden Aktionen ohne Umweg über tatsächlich ausgeführte Graphtransformationen ermittelt und daher

effizienter arbeitet.

(Hodek)

3 Die Modellierung neuronaler Netze mit Graphtransformationssystemen

Graphtransformationssysteme können mit Hilfe von High-Level-Replacement-Systemen definiert werden, bei denen die Anwendung einer Regel auf einen Graphen mit Hilfe zweier Pushoutdiagramme, einer Konstruktion aus der Kategorientheorie, beschrieben wird. Die Hauptvorteile der High-Level-Replacement-Systeme sind Theoreme zur Beschreibung parallel unabhängiger Ableitungsschritte. Graphtransformationen, so wie sie bisher in der Literatur definiert wurden, sind für die Modellierung und Simulation neuronaler Netze nicht ausreichend. Die den neuronalen Netzen zugrunde liegende Mathematik kann nicht mit reiner Graphersetzung modelliert werden; Mechanismen zur Markierung von Graphen sind nötig. Eine Markierung pro Knoten oder Kante ist nicht ausreichend, so dass ein neues Markierungskonzept auf Basis der High-Level-Replacement-Systeme sowie Kommakategorien entwickelt wurde. Es sollte sichergestellt werden, dass verschiedene Ersetzungssysteme gleichrangig miteinander kombiniert werden. Auf der Basis dieser Idee ist ein Graph nichts anderes als eine Menge (von Kanten), die mit einer weiteren Menge (von Knoten) markiert wird. Graphen sind also markierte Mengen. Die verwendeten Mengen können dann wiederum markiert werden, sei es mit Mengen oder Elementen anderer Kategorien. Kombinationsgraphen dienen dazu, die Markierungsstrukturen zu beschreiben. Zuletzt wurden Konzepte zum Ummarkieren von Knoten und Kanten entwickelt sowie zur Verwendung desselben Objekts in unterschiedlichen Markierungskontexten.

Methoden zur Analyse von High-Level-Replacement-Systemen und markierter Mengen lassen sich aus dem Bereich der allgemeinen Ersetzungssysteme übertragen. Die Terminierung wird mit Hilfe partieller, wohlfundierter Ordnungen gezeigt, wobei jedem Element des Ersetzungssystems eindeutig ein Element der partiellen Ordnung zugeordnet wird. Die Konfluenz sowie lokale Konfluenz werden auf der Basis einer Äquivalenzrelation definiert, da die Anwendung einer Regel nur ein bis auf Isomorphie eindeutiges Ergebnis liefert. Parallele Unabhängigkeit, wie für High-Level-Replacement-Systeme definiert, wird zum Beweis der lokalen Konfluenz verwendet.

Neuronale Netze sind das Anwendungsgebiet der so definierten Ersetzungssysteme. Auf der einen Seite wurde versucht, einen einheitlichen Ansatz zur Beschreibung der oft sehr unterschiedlichen Netze zu finden, der zusätzlich auch die problemlose Visualisierung der Netze und der zugehörigen Algorithmen ermöglicht. Auf der anderen Seite sollten sich auf Ersetzungssystemen basierende Vorgehensweisen auf das Gebiet der neuronalen Netze übertragen lassen. Dieser theoretische Ansatz ermöglicht es, verschiedene Eigenschaften neuronaler Netze und ihrer Algorithmen zu beweisen. Zwei Netzwerktypen waren von besonderem Interesse: mehrschichtige Perzeptrone und probabilistische neuronale Netze. Es wurde ein Ersetzungssystem, basierend auf markierten Mengen, entwickelt, das netzunabhängig ist. Die Algorithmen für das Training eines Netzes und die Berechnungen in diesem, die durch Ersetzungsregeln beschrieben werden, können auf beide Netztypen problemlos angewandt werden.

Zwei Fallstudien, die sich ausführlicher mit beiden Netztypen beschäftigen, wurden durchgeführt. Zuerst wurde das Falten und Entfalten von mehrschichtigen Perzeptronen untersucht. Dies diente insbesondere der Modellierung von Zeit während des Trainings. Ersetzungssysteme für zwei unterschiedliche Trainingsalgorithmen, dem sog. *Offline Real Time Backpropagation* und dem *Backpropagation Through Time*, wurden entwickelt. Durch Kombination der beiden

Regelsysteme entstand eine ganze Klasse von Algorithmen, die sich in der Anzahl und Reihenfolge der Faltungen und Entfaltungen unterscheiden. Es wurde gezeigt, dass der gemischte Regelsatz terminiert und dass die möglichen Ableitungsschritte lokal konfluent sind. Lokale Konfluenz wird meist mit Hilfe eines Theorems der High-Level-Replacement-Systeme gezeigt, der parallelen Unabhängigkeit für Ableitungsschritte. Ist ein Ersetzungssystem lokal konfluent und terminiert, hat es auch Normalformen. Dies bedeutet für das gemischte Regelsystem, dass das Ergebnis bis auf Isomorphie eindeutig ist, unabhängig davon, in welcher Reihenfolge die Regeln angewandt werden. Beide zugrunde liegenden Algorithmen sind also äquivalent, da sie dasselbe Ergebnis produzieren.

Die zweite Fallstudie beschäftigte sich mit einem Trainingsalgorithmus für probabilistische neuronale Netze, bei dem die Netzwerktopologie durch das Einfügen von Neuronen in der verdeckten Schicht verändert wird. Es wurde bewiesen, dass dieser Algorithmus terminiert. Allerdings stellte sich heraus, dass die Methoden, die die Transformation markierter Graphen zur Verfügung stellt, nicht ausreichend waren. Deshalb wurde nur eine Hälfte des Beweises mit diesen Methoden geführt. Für die zweite Hälfte wurden Mechanismen, wie sie normalerweise in der Mathematik neuronaler Netze verwendet werden, eingesetzt. Dies unterstreicht, dass reine Graphersetzung nicht ausreichend ist; sie muss immer mit anderen Methoden kombiniert werden.

(Fischer)

4 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 im *DiaGen*-Projekt vorrangig versucht, die Klasse der mit *DiaGen* beschreibbaren Diagrammsprachen zu vergrößern. Dazu wurde die Modellierung von Diagrammen mittels Hypergraphen erweitert: Im ursprünglichen Modell repräsentierten Hyperkanten die graphischen Bestandteile eines Diagramms; gemeinsam besuchte Hyperknoten stellten zueinander in Beziehung stehende graphische Bestandteile dar. Diese einfache Modellierung erwies sich für bestimmte Diagrammklassen, die beliebige räumliche Beziehungen zwischen Diagrammkomponenten erlauben (z. B. "Block A liegt über Block B"), als zu eingeschränkt. Zur Lösung dieses Problems wurde das Hypergraphmodell erweitert, indem auch räumliche Beziehungen zwischen graphischen Diagrammbestandteilen durch Hyperkanten ausgedrückt werden können [21]. Für Diagrammeditoren, die auf diesem erweiterten Hypergraphenmodell basieren, und die es erlauben, Diagrammkomponenten frei zu verschieben und zu verändern (im Gegensatz zum eingeschränkten, syntaxgesteuerten Editoren), ergibt sich damit das Problem, solche räumlichen Beziehungen während des Editierens zu erkennen. Dies ist Aufgabe eines "graphischen Scanners", bei dem wegen der

interaktiven Arbeitsweise des Diagrammeditors die Effizienz seiner Arbeitsweise im Vordergrund steht [23], [24].¹⁾

(Minas)

5 *N*-GUIDE

N-GUIDE steht für New Graphical User Interface Developer's Environment. Es ist also eine Entwicklungsumgebung für graphische Benutzeroberflächen. Es unterscheidet sich von vergleichbaren Softwaresystemen durch seine hohe Effizienz, Portabilität und Konfigurierbarkeit. *N*-GUIDE wurde objektorientiert unter Verwendung der Programmiersprache C++ entwickelt. Das Kernsystem besteht aus einer Klassenbibliothek. Diese enthält bereits alle grundlegenden Funktionalitäten wie:

- Layoutberechnung: Die Layoutberechnung von *N*-GUIDE basiert zwar auf einem ähnlichen Ansatz wie die von Motif, aber *N*-GUIDE bietet mehr Möglichkeiten und eine effizientere Berechnung. Darüber hinaus bietet *N*-GUIDE ein zweistufiges Layout, dessen abstraktere Stufe auch dem mit Layout unerfahrenen Programmierer schnell die Erzeugung sinnvoll aufgeteilter Dialoge ermöglicht.
- Ereignishandhabung: Die Ereignishandhabung ist an der Hierarchie der Dialogstruktur ausgerichtet und zudem portabel ausgelegt. Selbst die Integration neuer Ereignistypen ist auf einfache Weise möglich.
- Dialogelemente: Dialogelemente sind die Grundlage jeder graphischen Benutzeroberfläche. Die wichtigsten Grundtypen befinden sich bereits in der Klassenbibliothek von *N*-GUIDE (Texteingabefelder, Beschriftungen, Rollbalken, Menüs, Register usw.)
- Möglichkeiten zur Verknüpfung von Objekten mit Eigenschaften: *N*-GUIDE bietet sehr vielfältige Möglichkeiten, Objekte mit Eigenschaften von Dialogelementen zu verknüpfen. So können neben Prozeduren und Variablen auch andere Eigenschaften oder sogar ganze Ausdrücke von Funktionen, Operatoren, Variablen und Eigenschaften verknüpft werden. *N*-GUIDE sorgt dabei stets für die Konsistenz aller beteiligten Objekte.

Die reichhaltige Funktionalität der Klassenbibliothek wird von zwei dazugehörigen Werkzeugen unterstützt: dem Konfigurationsmanager und dem Dialogeditor. Der Konfigurationsmanager vereinfacht die Veränderung der individuellen Konfiguration. Der Dialogeditor ist dagegen ein Werkzeug, das nur für den Programmierer gedacht ist und der Erstellung beliebiger statischer Dialoge dient, deren Struktur bereits zum Zeitpunkt der Übersetzung festliegt. Der Dialogeditor ist in hohem Maße erweiterbar angelegt, so dass er auch ohne Funktionalitätseinbuße auf einfache Weise um neue Dialogelementklassen erweitert werden kann.

Als eines der ersten Softwaresysteme seiner Art versucht *N*-GUIDE, zweidimensionale Benutzeroberflächen auch mit dreidimensionalen zu verbinden. Dies gelingt einerseits durch Projektionen der zweidimensionalen Dialogelemente und andererseits durch neu implementierte dreidimensionale Dialogelemente. *N*-GUIDE hat in diesem Zusammenhang auch

¹⁾ Nähere Informationen sind unter <http://www2.informatik.uni-erlangen.de/DiaGen> zu finden.

seine hohe Erweiterbarkeit unter Beweis gestellt. Sehr viele Eigenschaften von *N-GUIDE* konnten unabhängig von der Dimensionsanzahl implementiert werden (Layout, Ereignisbehandlung, Verknüpfung von Eigenschaften usw.)

GIDTS (Graphical Interactive Debugging, Testing and Slicing System) ist das im Folgenden beschriebene IDTS, erweitert um eine graphische Benutzeroberfläche auf der Grundlage von *N-GUIDE*. Der graphische Anteil von GIDTS konnte mit erwartungsgemäß geringem Aufwand programmiert werden.

(Nilson)

Fehler! Textmarke nicht definiert.
PROLOG-Programmen

Debugging und maschinelles Lernen von

Kernpunkt dieser Arbeit ist eine integrierte Methode zum funktionalen Testen und zur Fehlersuche (*Debugging*) in logischen Programmen, die in Form des Systems IDTS (Interactive Debugging, Testing and Slicing System) implementiert wurde. IDTS basiert auf dem von *Shapiro* vorgeschlagenen Debugging-Verfahren, das den Programmierer durch gezielt gestellte Fragen zu Programmfehlern hinführt. IDTS ergänzt *Shapiros* Verfahren um zwei weitere Methoden (*Category Partition Testing* und *Slicing*) und verbessert die Effizienz der Fehlersuche durch die Verwendung von Testspezifikationen und Testdatenbanken. IDTS reduziert die Zahl der beim Debuggen gestellten Fragen dadurch, dass die Benutzereingaben kategorisiert und mit den Testfällen der Testdatenbank verglichen werden. Außerdem wird der Programmabhängigkeitsgraph genutzt, um fehlerfreie Programmteile von vornherein von der Fehlersuche auszuschließen. IDTS ist vollständig in PROLOG implementiert und kann mittels einer graphischen Benutzeroberfläche (GIDTS, s. 5) bedient werden.

Weiterhin wurde das SPECG-System (Specialized Prolog-based Electrocardiogram) entwickelt. Das SPECG-System kombiniert einen Elektrokardiogramm-Wellenklassifikator mit dem Debuggingssystem IDTS. Mit Hilfe dieses Verfahrens wird ein Fehler in einer Klassifikation gefunden, wenn das Klassifikatorprogramm die eingegebenen EKG-Wellen nicht analysieren kann.

Die IDTS-Methode wurde außerdem in eine maschinelle Lernmethode IMPUT (Interactive Multiple Predicate Learner Using Unfolding Transformation) integriert. Das IMPUT-System kann eine interaktive Revision der vielfältigen Prädikate eines logischen Programms durchführen. Die Hauptidee beim IMPUT-Verfahren ist, eine Verbesserung des Spezialisierungsprozesses durch die Anwendung von IDTS zu erreichen.

Schließlich wurde gezeigt, dass die Integration des IMPUT-Verfahrens in das SPECG-System ein Programm ergibt, das dem Benutzer hilft, ein Klassifikatorprogramm zu verbessern, wenn es eine vorgegebene Eingabe nicht klassifizieren kann. Das System kann damit sowohl Syntax- als auch Semantikregeln von *EKG*-Grammatiken lernen. Die hierbei verwendeten Regeln braucht der Anwender, um sog. QRS-Komplexe von den anderen *EKG*-Wellenformen trennen zu können.

(Kkai)

7 Verteilte Konfigurationsverwaltung²⁾

²⁾ Diese Arbeit wurde durch Mittel der Basys GmbH, Erlangen, sowie des Bundesministeriums für Forschung und Technologie (BMFT)

Heutzutage werden Produkte häufig an mehreren Standorten zugleich entwickelt, weil externes Know-How hinzugekauft wurde, aus Kostengründen bestimmte Bereiche ausgelagert werden (Outsourcing) oder durch Firmenzusammenschlüsse mehrere geographisch getrennte Abteilungen zusammen arbeiten sollen. Gerade im Fall derart verteilter Softwareentwicklung sind die bisherigen Werkzeuge zur Automatisierung der Organisation und Koordination, wie etwa Konfigurationsverwaltungen und Werkzeuge für das Change-Management und Software-Prozesse, unzulänglich. Deshalb wurde im Rahmen des Forschungsprojekts "Verteilte Konfigurationsverwaltung im Kontext des Software-Engineering-Prozesses" ein Reihe von Werkzeugen entwickelt, die insbesondere die verteilte Softwareentwicklung unterstützen. Dabei wurde besonderer Wert auf Möglichkeiten zur Anpassung an den eigenen Software-Prozess gelegt. So gibt es in der Konfigurationsverwaltung verschiedene Sperr-Granularitäten: Gesperrt werden können Konfigurationen, Dokumente, Varianten (Branches) und einzelne Revisionen. Die Fehlerdatenbank kann beliebig durch eigene Attribute erweitert werden. Außerdem lassen sich die einzelnen Bearbeitungsschritte (Änderungsprozess) beliebig definieren.

Um die Architektur auf ein möglichst sicheres Fundament zu stellen, wurde der Kern der verteilten Konfigurationsverwaltung mit einem formalen Modell spezifiziert. Als Grundlage dient ein in Berlin von *Gabi Taentzer* **Fehler! Textmarke nicht definiert.**³⁾ entwickeltes Modell zweistufiger Graphtransformationssysteme. Während mit der ersten Schicht die einzelnen Abteilungen modelliert werden, dient die zweite Schicht der Darstellung des aktuellen Stands (der Dokumente) in den Abteilungen. Abhängigkeiten zwischen einzelnen Projektpartnern können problemlos dargestellt werden. Mit Hilfe von Transformationsregeln können einzelne Arbeitsschritte innerhalb eines Teilprojekts modelliert werden, aber auch die Kommunikation zwischen den Projektpartnern.

(Volle)

8 Evolution und Koevolution in Multiagenten- und Pflanzenökosystemen

Die Modellierung, Visualisierung und Animation von Pflanzen eignet sich sehr gut zur Illustration und zur Untersuchung allgemeiner evolutionärer Mechanismen von Wachstums- und Musterbildungsprozessen. Zur Modellierung von Wachstumsvorgängen bei Pflanzen verwenden wir Lindenmayer-Systeme (L-Systeme), die sich als parallele Ersetzungssysteme besonders gut dafür eignen, dynamische Prozesse (Zellteilung, Längen- und Breitenwachstum, Blütenbildung etc.) zu modellieren. Den Aspekt, dass die "Wachstumsprogramme" natürlicher Zellen im Genom kodiert sind, berücksichtigen wir, indem die L-Systeme als symbolische Ausdrücke (Terme, Bäume) repräsentiert werden. Auf diesen Ausdrücken werden Mutationsoperatoren im Sinne der sog. genetischen Programmierung definiert, durch die sich Variationen und Rekombinationen bisheriger Wachstumsprogramme erzeugen lassen - in Analogie zur Evolution in der Natur. Mit diesem System lassen sich auf relativ einfache Weise L-Systeme bzw. Pflanzen mit bestimmten Wachstumscharakteristiken "züchten" (siehe auch Jahresbericht 1997).⁴⁾

unter dem Förderkennzeichen FKP0004402B7a unterstützt.

3) "Parallel and Distributed Graph Transformation: Formal Description and Application to Communication-Based System". Dissertation, TU Berlin, Shaker Verlag, Germany, 1996

4) Nähere Informationen siehe auch unter <http://www2.informatik.uni-erlangen.de/~jacob>

Diesen Ansatz haben wir im vergangenen Jahr um einige Aspekte erweitert:

1. Bisher haben wir L-Systeme zwar in Populationen organisiert, jedoch wurde jede Pflanze unabhängig von den übrigen bewertet. Ein nächster Schritt besteht demnach darin, Pflanzengruppen im Sinne von Ökosystemen miteinander in Interaktion treten zu lassen. Somit lassen sich auch koevolutive und Konkurrenzeffekte mit berücksichtigen. Die bisherige explizite Fitnessfunktion wird nunmehr durch eine implizite Fitnessbewertung ersetzt; das "Überleben" in Konkurrenz und in Kooperation bestimmt nun die Dynamik des Evolutionssystems. Ein erstes Prototypsystem zur Untersuchung derjenigen Faktoren, die für die Modellierung einfacher Pflanzenökosysteme von Bedeutung sind, wurde bereits in einem Artikel zum Sonderband "Modeling Collective Phenomena in the Sciences" [12] vorgestellt.
2. Ökosysteme sind im Kontext komplexer adaptiver Systeme als Multiagentensysteme zu betrachten. Mit Zwei- und Mehragentensystemen befasst sich insbesondere die evolutionäre Spieltheorie. In diesem Zusammenhang untersuchen wir im Rahmen einer Diplomarbeit (*C. Miss*) Effekte der Kooperation und Konkurrenz am Modell von zellulären Automaten. Mit Hilfe der genetischen Programmierung sollen Spiel- und Lernstrategien evolviert werden.
3. In einem Gemeinschaftsprojekt mit *Prof. Seth Chandler* vom University of Houston Law Center, USA, wird versucht, das Lernverhalten von Personengruppen unter einer Reihe von gesetzlichen Randbedingungen zu modellieren. Auch hier kommen zelluläre Automaten zum Einsatz, die mit evolutionären Algorithmen kombiniert sind. An einfachen Beispielen konnten wir bereits zeigen, wie verschiedene gesetzliche Regelungen hinsichtlich der Interaktionen zwischen den Zellen (Personen, Personengruppen) sich auf die Evolution des Lernens und die Entwicklung eines ökonomischen Systems auswirken.⁵⁾

(Jacob)

9 Lösung praxisrelevanter Probleme mit neuronalen Netzen

Im vergangenen Jahr wurden mittels neuronaler Netze verschiedene Probleme aus der Praxis gelöst, die mittels herkömmlicher Algorithmen entweder nicht oder nur sehr ineffizient gelöst werden können.

Optimierung der Einsatzplanung für Personal im Schichtdienst

In Kooperation mit einem Softwarehaus wurde die Problematik der Einsatzplanung für Personal im Schichtdienst bearbeitet. Verschiedene Rahmenbedingungen in unterschiedlichen wirtschaftlichen Bereichen wurden untersucht und eine Abgrenzung des Problems vorgenommen. Anschließend wurden die informationstheoretischen Grundlagen der Aufgabenstellung analysiert und mögliche Problemlösungsstrategien aufgezeigt. Die Diskussion der verschiedenen Verfahren kam zu dem Ergebnis, dass für die Implementierung einer automatischen Planungskomponente evolutionäre Verfahren genutzt werden sollen.

Es wurden zwei unterschiedliche evolutionäre Verfahren, nämlich auf Basis genetischer Algorithmen und evolutionärer Strategien, konzipiert. Aufgrund der verschiedenen Grundalgorithmen und der daraus resultierenden unterschiedlichen Anwendungsgebiete beider Techniken wurde der genetische Algorithmus der evolutionären Strategie für eine Imple-

⁵⁾ S. Chandler und C. Jacob, Automata Containing Evolutionary Algorithms: Behavior and Learning Under Law (submitted), IMS-99, International Mathematica Symposium, Linz, Österreich, 1999

mentierung vorgezogen.

Nach Erläuterung der allgemeinen Erfordernisse an einen genetischen Algorithmus im Rahmen der Personaleinsatzplanung wurden verschiedene Vorschläge für das Design der Codierung und der Operatoren des genetischen Algorithmus gemacht. Für ein konkretes Beispiel, nämlich der Stationsplanung in einem Krankenhaus, wurde eine Implementierung in C++ vorgenommen. Sowohl aufgrund der komplexeren Rahmenbedingungen als auch aufgrund der unterschiedlichen Codierung und somit Vorgehensweise zur Lösung des Problems unterscheidet sich die Implementierung von bisher auf dem Gebiet der Personaleinsatzplanung gemachten Versuchen mit genetischen Algorithmen.

Durch den Einsatz hybrider Operatoren, die innerhalb des genetischen Algorithmus problemspezifisches Wissen einsetzen, gelang die Erstellung eines Schichtplans für einen gesamten Monat unter Einhaltung aller gegebenen Rahmenbedingungen.

Die Ergebnisse lassen darauf schließen, dass genetische Algorithmen unter Zuhilfenahme von problemspezifischem Wissen sehr gut dafür geeignet sind, Einsatzpläne für Personal im Schichtdienst automatisch zu erstellen.

Entwicklungsumgebung für die Konvertierung von Datenformaten

Datenkonvertierung ist die Methode, Daten eines bestimmten Formats so zu manipulieren, dass sie ohne Verlust ihrer Bedeutung mit anderer Software oder in einem anderen Kontext verwendet werden können. Es wird der Frage nachgegangen, was die Besonderheiten von Datenformaten sind und in welchem Zusammenhang die Konvertierung von Daten mit anderen strukturtransformierenden Methoden steht.

Zu diesem Zweck wurde unter einer graphischen Oberfläche eine Entwicklungsumgebung zur Konvertierung textorientierter Daten geschrieben, die grammatikbasiert ist und dem Benutzer Methoden zur regelbasierten Formatbeschreibung anbietet. Neben der Formatbeschreibung können auf einfache Weise Verknüpfungen zwischen Quell- und Zielformat hergestellt werden, die die Konvertierung des Quellformats in einem automatisch erzeugten Konverter steuern. Für die Transformation der syntaktischen Struktur wurde ein neuer Ansatz gefunden, der nicht auf der Manipulation des Ableitungsbaums der Eingabe basiert, sondern bei dem die Ausgabe gemäß der Zielformatbeschreibung unabhängig generiert wird. Die zu übertragenden Quelldaten werden dann nach einer festen Vorschrift in die Zielstruktur eingefügt.

Anhand zweier Beispiele wurde gezeigt, dass die Entwicklungsumgebung dazu geeignet ist, kleinere Datenkonvertierungsprobleme mit geringem zeitlichen Aufwand zu bewältigen.

Lösung des Partitionierungsproblems mit evolutionären Algorithmen in einem betriebswirtschaftlichen Geo-Informationssystem

Ziel dieser Arbeit war es, einen Algorithmus zu finden, der in der Lage ist, das Partitionierungsproblem mit mehrdimensionalen Objekten zu lösen. Dieses Problem ist NP-vollständig und damit nicht effizient berechenbar. Für ein aktives Managementinformationssystem, dem DeltaMiner der Firma Bissantz & Company in Nürnberg, sollen die Postleitzahlgebiete der Bundesrepublik Deutschland in Gruppen eingeteilt werden, so dass diese bezüglich mehrerer

Kriterien gleich groß sind. Herkömmliche Ansätze liefern in einer vertretbaren Zeit keine guten Ergebnisse.

Mit einem genetischen Algorithmus konnte eine brauchbare Lösung für die geforderte Problemgröße mit zufällig erzeugten eindimensionalen Testdaten in weniger als 30 Minuten berechnet werden. Der Speicheraufwand hierfür war kaum mehr als 8 MB. Für das Partitionierungsproblem kann eine meist zu optimistische Schätzung über den zu erwartenden Gruppenwert gemacht werden. Von diesem Wert war die schlechteste Gruppe bei zufällig erzeugten Testdaten zwischen 3% und 6% entfernt. Die mittlere Abweichung betrug unter 1%. Bei realen Daten waren die Ergebnisse gewöhnlich wesentlich besser.

Für diese Arbeit wurden zusätzlich neu erarbeitete Techniken erfolgreich getestet. Neben der Verwendung von Vorgehensweisen der evolutionären Strategien wurde auch ein neuer Steuerungsmechanismus eingeführt, der sich vorteilhaft auf die Laufzeit auswirkt. Durch eine andere Darstellung der Entwicklung der Population konnten neue Erkenntnisse über das Verhalten von genetischen Algorithmen gewonnen werden.

Ein Controller für eine Fahrzeugsteuerung auf Basis neuronaler Netze

In dieser Studienarbeit konnte gezeigt werden, dass künstliche neuronale Netze prinzipiell in der Lage sind, als Regler eines Regelkreises zu fungieren. In einer konkreten Anwendung sollte ein Netzwerk die Steuerung eines Kraftfahrzeuges übernehmen. Als Experimentalumgebung für die Fahrdynamik wurde RARS verwendet, ein System zur Simulation von Autorennen, wobei durch Kontrollfunktionen gesteuerte Kraftfahrzeuge auf Rundkursen gegeneinander antreten. Ein mit Hilfe von NeuroGraph erstelltes und trainiertes Netzwerk sollte die neuronale Steuerung eines der Rennwagen übernehmen.

Die erfolgreiche Umsetzung der gestellten Aufgabe erforderte eine geeignete Modifikation der Eingangsdaten und eine angemessene Netzwerkarchitektur. Da sich der Validierungsfehler als ungeeignetes Gütemaß für ein Netz erwies, wurde er durch die in der realen Rennsituation erzielten Rundenzeiten ersetzt, was sich äußerst positiv auf den Entwicklungsprozess auswirkte.

Eine Automation des Bewertungsvorgangs für neuronale Netzwerke ermöglichte schließlich die Realisierung von mehreren mit Backpropagation trainierten Neuro-Controllern, die auf einzelnen Rundkursen konkurrenzfähige Fahrleistungen erbrachten. Allerdings gelang es nicht, einen Universal-Controller zu trainieren, der auf allen Strecken mit konventionellen Ansätzen vergleichbare Ergebnisse erzielte.

Für die Lösung dieses Problems waren die verwendeten Netzwerke offensichtlich von zu geringer Komplexität. Die erreichten Teilerfolge zeigen jedoch die prinzipielle Eignung von künstlichen neuronalen Netzen bei der Kontrolle dieser Art von Regelkreisen.

(Wilke)

10 Maschinelle Verarbeitung von Idiomen

Die festen Wortverbindungen, die in der Linguistik als Phraseologismen oder Idiome bezeichnet werden, stellen innerhalb der maschinellen Sprachverarbeitung eine besonders zu behandelnde Gruppe dar. Sie zeichnen sich durch eine

übertragene, oft bildliche Bedeutung aus, die sich nicht aus den literalen Bedeutungen ihrer Komponenten erschließen lässt. Während die ersten Ansätze zur maschinellen Verarbeitung von Idiomen sie als starre, unveränderliche Gebilde betrachteten, wurde in den letzten Jahrzehnten deutlich, dass sie syntaktisch sehr flexibel sind. Weiterhin wurde die traditionelle Ansicht, wonach sie grundsätzlich semantisch nichtkompositionell sind, von verschiedenen Forschern in Frage gestellt.

Im Rahmen dieser Erkenntnisse wurde in den letzten Jahren von *Fischer/Keil* die semantische Verarbeitung modifizierter, kompositioneller Idiome, bei denen eine zur Nennform parallel aufgebaute Paraphrase gefunden werden kann, untersucht. Der verwendete Formalismus ist eine Kombination aus λ -Kalkül und Diskursrepräsentationstheorie. Aufbauend auf ihren Arbeiten wurde diese Idiomdarstellung um eine möglichst ähnliche Darstellung semantisch leerer Idiomkomponenten erweitert, die mit der syntaktischen Modifikation derartiger Komponenten durch Adjektive oder Relativsätze semantisch adäquat umgehen kann. Die gewonnenen Erkenntnisse wurden mit Hilfe eines am Lehrstuhl für Künstliche Intelligenz entwickelten Chartparsers implementiert. Die folgenden Phänomene wurden untersucht und in die formale Darstellung der Idiome aufgenommen, aber noch nicht implementiert: die Rolle des Artikels, syntaktische Flexibilität und Semantik von Negationspartikeln, Idiome, bei denen sich Lemma und Paraphrase im syntaktischen Aufbau unterscheiden, Varianten von Idiomen bzw. Familien verwandter Idiome. Weiterhin wurden Kriterien dafür entwickelt, für welche Idiome eine kompositionelle Darstellung angemessen ist, sowie das von *Martina Keil* entwickelte computergestützte Idiomlexikon Phraseo-Lex kritisch daraufhin überprüft, inwieweit es sich für die automatische Generierung von Lexika für die maschinelle Sprachverarbeitung eignet.

(Dormeyer)

11 Requirements-Engineering in der Automatisierungstechnik (FORSOFT/Teilprojekt A4)⁶⁾

Im Teilprojekt A4 des Forschungsverbunds Software Engineering⁷⁾ befassen wir uns mit dem Requirements Engineering in der Automatisierungstechnik. Dabei werden die Anwendungsbereiche des Requirements Engineering komplexer Standardsoftware und des Requirements Engineering automatisierter Produktionsanlagen behandelt. Im ersten Anwendungsbereich befassen wir uns mit dem Entwicklungsprozess der Engineering Software des Produktsystems SIMATIC unseres Industriepartners Siemens A&D AS. Ziel ist dabei die Entwicklung einer Methodik für die Erfassung, Verhandlung, Spezifikation und Validierung von Anforderungen an Software, die in Versionen und Varianten entwickelt und auf einem globalen Markt verkauft wird. Besondere Schwerpunkte sind die integrierte Verwaltung von Anforderungen in einem gemeinsamen Datenmodell und die Kosten/Nutzen-Priorisierung zur Schaffung einer fundierten Grundlage für die Auswahl von Anforderungen, die in der Software umgesetzt werden sollen. Im zweiten Anwendungsbereich befassen wir uns in enger Zusammenarbeit mit dem Institut für Werkzeugmaschinen und Betriebswissenschaften der TU München (iwb) mit der integrierten Planung und Entwicklung automatisierter Produktionsanlagen, wobei die Integration des Requirements Engineering für das Automatisierungssystem einer Anlage den Schwerpunkt unserer Arbeiten bildet. Dazu wurde ein Prozessmodell für die integrierte Planung und Entwicklung einer Anlage entworfen, das auf einen frühzeitigen

⁶⁾ gefördert von der Bayerischen Forschungsstiftung und Industriepartnern

⁷⁾ <http://www.forsoft.de>

Beginn der Automatisierungsentwicklung, eine verstärkte Integration des Kunden und auf eine verbesserte Zusammenarbeit zwischen Entwicklern verschiedener Fachgebiete abzielt. Weiterhin wurde ein Informationsmodell abgeleitet, das die integrierte Verwaltung der für die Planung und Entwicklung benötigten Informationen (z. B. Teilefamilien, Bearbeitungsfolgen, Produktionsprogramme, Anlagendesign) ermöglicht. Die Methodik wird in naher Zukunft im Rahmen einer Fallstudie beim Transferstraßenhersteller HOMAG AG unter Einsatz eines dafür entwickelten Werkzeugsystems erprobt.

(Billing)

Dissertationen:

- | | |
|------------------|--|
| Fischer, Ingrid: | Describing Neural Networks with Graph Transformations |
| Kkai, Gabriella: | Debugging und maschinelles Lernen von PROLOG-Programmen |
| Nilson, Jörg: | <i>M</i> -GUIDE - Ein System zur Erstellung grafischer Benutzeroberflächen |

Habilitation:

- | | |
|-----------------|------------------------------------|
| Fritzke, Bernd: | Vektorbasierte Neuronale Netzwerke |
|-----------------|------------------------------------|

Studienarbeiten:

- | | |
|---------------------|--|
| Fuchs, Matthias: | Lösung des Partitionierungsproblems mit evolutionären Algorithmen in einem betriebswirtschaftlichen Geo-Informationssystem |
| Ganzleben, Andreas: | Datenkonvertierung von Excel zu NeuroGraph |
| Glaser, Günter: | Ein Controller für eine Fahrzeugsteuerung auf Basis neuronaler Netze |
| Lee, Ki-Cheol: | Die Re-Implementation der Oberfläche einer Datenbank in der Programmiersprache Java |
| Niss, Christian: | Spezifikation und Implementierung einer grafischen Benutzeroberfläche für das Prolog-Debugging-System IDTS |
| Poetzsch, Jörg: | Entwurf eines Generators für graphische Editoren und seine Implementierung in der Programmiersprache Java |

Diplomarbeiten:

Chen, Ningsui:	Entwurf und Implementation einer Fehlerdatenbank in der Programmiersprache Java (JDBC)
Farnschläder, Frank:	Eine Grammatik zur Verarbeitung von Idiomen
Goebel, Hartmut:	64-Bit-Portierung des Alpha-Oberon-Systems und des Oberon-2-Compilers
Gröbner, Matthias:	Optimierung der Einsatzplanung für Personal im Schichtdienst
Lenke, Doris:	Die Darstellung verteilter Systeme mit Graphen
Metin, Mine:	Implementierung eines interaktiven Auskunftssystems
Stingl, Bettina:	Spezifikation animierter Diagrammklassen mittels Graphtransformationssystemen
Wetzel, Christian:	Entwicklungsumgebung für die Konvertierung von Datenformaten
Winterstein, Alexander:	Erweiterung eines Editors für graphische Benutzeroberflächen
Wopmann, Hans:	Modellierungen zur Evolution komplexer Interaktionen zwischen Umwelt und Organismen

Veröffentlichungen

- [1] *Bardohl, R.; Taentzer, G.; Minas, M.; Schürr, A.*: "Application of graph transformation to visual languages". Erscheint in: H. Ehrig, G. Engels, H.-J. Kreowski, and G. Rozenberg (eds.), Handbook of Graph Grammars and Computing by Graph Transformation, Vol. II: Applications, Languages and Tools. World Scientific, 1999
- [2] *Dormeyer, R.; Fischer, I.*: "Building Lexicons out of a Database for Idioms". In: Antonio Rubio, Natividad Gallardo, Rosa Castro and Antonio Tejada (eds.), Proc. of the First Intern. Conference on Language Resources and Evaluation, Granada, Spanien, May 28 - 30, 1998, S. 833 - 838
- [3] *Dormeyer, R.; Fischer, I.; Keil, M.*: "A Database for Verbal Idioms". In: Thierry Fontenelle, Philippe Hillmann, Archibald Michiels, Andre Moulin, Siegfried Theissen (eds.), Euralex'98, Proc., Univ. de Lie`ge, Department d'anglais et de neerlandais, Belgien, Aug. 4 - 8, 1998, S. 99 - 109
- [4] *Dotzel, G.; Goebel, H.*: "64-Bit Oberon", ACM SIGPLAN Notices 33,2 (Febr. 1998), S. 56 - 58
- [5] *Fischer, I.*: "Describing Neural Networks with Graph Transformations". Arbeitsber. des IMMD der Univ. Erlangen-Nürnberg, Band 31, Nr. 11, 1998
- [6] *Fischer, I.; Koch, M.; Berthold, M.R.*: "Formalizing Neural Networks". In: Wilfried Brauer (ed.), Proc. of the 5. GI-Workshop Fuzzy-Neuro-Systems '98, Proc. in Artificial Intelligence, infix, March 1998, S. 218 - 225
- [7] *Fischer, I.; Koch, M.; Berthold, M.R.*: "Proving Properties of Neural Networks with Graph Transformations". In: Proc. of the IEEE Intern. Joint Conference on Neural Networks, 1, Anchorage, Alaska, 1998, S. 457 - 456
- [8] *Fischer, I.; Koch, M.; Berthold, M.R.*: "Showing the Equivalence of Two Training Algorithms". In: Proc. of the IEEE Intern. Joint Conference on Neural Networks, 1, Anchorage, Alaska, 1998, S. 441 - 446
- [9] *Fischer, I.; Koch, M.; Taentzer, G.*: "Local Views on Distributed Systems and their Communications". Preliminary

Proc. of TAGT'98, Theory and Application of Graphtransformation, Technical Report, Univ. of Paderborn, tr-ri-98-201, 1998, S. 40 - 47

- [10] *Fischer, I.; Koch, M.; Taentzer, G.*: "Visual Design of Distributed Object Systems by Graph Transformation". Intern. Workshop on Communication Based Systems, Technical Univ. of Berlin, Germany and Shanghai Jiao Tong Univ., China, Technical TU Berlin 98/15, ed.: Guenter Hommel
- [11] *Jacob, C.*: "Evolution and Co-Evolution of Developmental Programs", Proc. CCP 1998, Intern. Conference on Computational Physics, Granada, Spanien, Sept. 1998
- [12] *Jacob, C.*: "Evolution and Coevolution of Developmental Programs". In: Modeling Collective Phenomena in the Sciences, Computer Physics Communications, Elsevier, 1998
- [13] *Jacob, C.*: "Evolving Developmental Programs", Proc. 2nd Workshop on Frontiers of Evolutionary Algorithms (FEA), Joint Conference on Information Sciences (JCIS), Research Triangle Park, North Carolina, USA, Okt. 1998
- [14] *Jacob, C.*: "Stochastic Search Methods". In: Michael R. Berthold und David J. Hand (eds.), An Introduction to Intelligent Data Analysis, Springer Verlag, Berlin, 1999, Kapitel 9
- [15] *Keil, M.*: "The lexical entry of phrasal idioms". In: Ulrich Heid (ed.), Proc. of the 3rd Intern. Symposium on Phraseology ISP-3, Technical Report, Univ. of Stuttgart, Inst. für Maschinelle Sprachverabtg., Ber. Lexika/Textcorpora, 1998, S. 149 -155
- [16] *Koch, M.; Fischer, I.; Volle, V.*: "Graphtransformation zur visuellen Beschreibung einer transaktionsgesteuerten Konfigurationskontrolle". GI Rundbrief Datenbanken, Mai 1998
- [17] *Kkai, G.*: "Error Diagnosis in Prolog Programs, A Critical View". In: Proc. of Conference of PhD Students on Computer Sciences Szeged, Ungarn, July 18 - 22, 1998, S. 64 - 67
- [18] *Kkai, G.*: "Debugging und maschinelles Lernen von PROLOG-Programmen". Arbeitsber. des IMMD der Univ. Erlangen-Nürnberg, Band 31, Nr. 8, 1998
- [19] *Kkai, G., Thury, A., Csirik, J., Gyimthy, T. and Csandy, M.*: "Diagnosis of Heart Diseases with the Help of PEECG". 8th Intern. Congress on Holter and Noninvasive Electrocardiology, Ulm, May 22 - 23, 1998 In Annals of Noninvasive Electrocardiology, Vol. 3, Number 3, Part 2, July 1998, S. 58

- [20] *Kkai, G.; Vnyj, R.; Tth, Z.*: "Application of Genetic Algorithms with more Populations for Lindenmayer Systems". In: Proc ICSC98, Intern. Symposium on Engineering of Intelligent Systems, Univ. of Laguna, Teneriffa/Spanien, Febr. 11 - 13, 1998, S. 324 - 331
- [21] *Minas, M.*: "Specifying diagram languages by means of hypergraph grammars". In: Proc. Thinking with Diagrams '98 (TwD'98), Aberystwyth, United Kingdom, S. 151 -157, Aug. 1998
- [22] *Minas, M.*: "Automatically generating environments for dynamic diagram languages". In: Proc. 1998 IEEE Symposium on Visual Languages (VL'98), Halifax, Kanada, IEEE Computer Society Press, 1998, S. 70 - 71.
- [23] *Minas, M.*: "Hypergraph representation of diagrams in diagram editors". In: Proc. AAAI Fall Symposium 1998, Workshop on Formalizing Reasoning with Visual and Diagrammatic Representations, Orlando, Florida, AAAI Technical Report FS-98-04, Okt. 1998, S. 79 - 85
- [24] *Minas, M.*: "Hypergraphs as a uniform diagram representation model". In: Preliminary Proc. 6th Intern. Workshop on Theory and Application of Graph Transformations (TAGT'98), Paderborn, S. 24 - 31, Technical Report, Univ. of Paderborn, tr-ri-98-201, Nov. 1998
- [25] *Nilson, J.*: "*N*-GUIDE - Ein System zur Erstellung grafischer Benutzeroberflächen". Arbeitsber. des IMMD der Univ. Erlangen-Nürnberg, Band 31, Nr. 2, 1998
- [26] *Schneider, H. J.*: "Foundations of Parallel Algorithms", Arbeitsber. des IMMD der Univ. Erlangen-Nürnberg, Band 31, Nr. 3, 1998
- [27] *Schneider, H. J.*: "Describing Systems of Processes by the Double-Pushout Approach: Jungles as Distributed Data Structures". Preliminary Proc. of TAGT'98, Theory and Application of Graphtransformation, Technical Report, Univ. of Paderborn, tr-ri-98-201, 1998, S. 1 - 8
- [28] *Schneider, H. J.*: "Describing Systems of Processes by Means of High-Level Replacement", erscheint in: G. Rozenberg (ed.), Handbook of Graph Grammars and Computing by Graph Transformation, Vol. III: Concurrency, World Scientific, 1999
- [29] *Taentzer, G.; Fischer, I.; Koch, M.; Volle, V.*: "Distributed graph transformation with application to visual design of distributed systems. Erscheint in: G. Rozenberg, U. Montanari, H. Ehrig, H.-J. Kreowski (ed.). Handbook of Graph Grammars and Computing by Graph Transformation, Vol. III: Concurrency, Parallellism, and Distribution. World Scientific, 1999
- [30] *Thury, A.; Kkai, G.; Csirik, J.; Gyimthy, T.; Csandy, M.*: "Development of SPECG System for Automated ECG Analysis". In: Proc 25th Intern. Congress on Electrocardiology and 39th Intern. Symposium on Vectorcardiography, Budapest, Hungary June 3 - 6, 1998, S. 37

Vorträge

- Dormeyer, R.: "Building Lexicons out of a Database for Idioms" (Posterpräsentation), 1st Intern. Conference on Languages Resources and Evaluation, Granada, Spanien
29. 05. 1998
- Dormeyer, R.: "A Database for Verbal Idioms", Euralex'98, Univ. Liege, Belgien
08. 08. 1998
- Fischer, I.: "Formalizing Neural Networks", 5. Intern. GI-Workshop 'Fuzzy-Neuro', Systems '98, München
20. 03. 1998
- Fischer, I.: "Proving Properties of Neural Networks with Graph Transformations" (Posterpräsentation), Intern. Joint Conference on Neural Networks, Anchorage, Alaska
07. 05. 1998
- Fischer, I.: "Showing the Equivalence of Two Training Algorithms" (Posterpräsentation), Intern. Joint Conference on Neural Networks, Anchorage, Alaska
07. 05. 1998
- Fischer, I.: "Local Views on Distributed Systems and their Communications", Theory and Application of Graph Transformation, Paderborn
16. 11. 1998
- Fischer, I.: "Die Modellierung Neuronaler Netze mit Graphtransformationen", Promotionsvortrag, Univ. Erlangen-Nürnberg
18. 12. 1998
- Fritzke, B.: "Neuronale Netze und das Bias-Varianz Dilemma", Habilitationsvortrag, Univ. Erlangen-Nürnberg
07. 01. 1998
- Jacob, C.: "Evolutionary Algorithms and Complex Adaptive Systems", (Kompakt-Vorlesung), Department of Informatics, Univ. Szeged, Ungarn
16. 04. bis 27. 04. 1998
- Jacob, C.: "Programming with Evolution", Department of Informatics, Univ. Szeged, Ungarn
17. 04. 1998
- Jacob, C.: "Evolution and Coevolution of Developmental Programs", Conference on Computational Physics, Granada, Spanien
04. 09. 1998
- Jacob, C.: "Evolutionary Computing and Agent-Based Distributed Systems" (Tutorial), Parallel Problem Solving from Nature V (PPSN V, 1998), Amsterdam
27. 09. 1998

- Jacob, C.: "Evolution and Coevolution of Developmental Programs", Faculty of Mathematics, Computer Science, Physics & Astronomy, Univ. Amsterdam
01. 10. 1998
- Jacob, C.: "Evolving Developmental Programs", 2nd Workshop on Frontiers of Evolutionary Algorithms (FEA), Joint Conference on Information Sciences (JCIS), Research Triangle Park, North Carolina, USA
25. 10. 1998
- Jacob, C.: "Simulierte Evolution von Entwicklungsprogrammen der Natur" MINT 2000, Mathematik, Informatik, Naturwissensch. und Technik 2000, Lehrer-Fortbildungskonferenz des Ernst Klett-Verlags, Hamburg
07. 11. 1998
- Kkai, G.: "Application of Genetic Algorithms with more Populations for Lindenmayer Systems", ICSC98, Intern. Symposium on Engineering of Intelligent Systems, Univ. Laguna, Tenerife, Spain
12. 02. 1998
- Kkai, G.: "Development of SPECG System for Automated ECG Analysis", 25th Intern. Congress on Electrocardiology and 39th Intern. Symposium on Vectorcardiography, Budapest, Ungarn
04. 06. 1998
- Kkai, G.: "Diagnosis of Heart Diseases with the Help of PEEG" (Posterpräsentation), 8th Intern. Congress on Holter and Noninvasive Electrocardiology, Ulm
23. 05. 1998
- Kkai, G.: "Entwicklung des SPECG-Systems zur automatisierten EKG-Analyse", Promotionsvortrag, Univ. Erlangen-Nürnberg
01. 09. 1998
- Minas, M.: "Objektorientierte Konzepte für GRACE", GRACE-Treffen, Januar 1998
30. 01. 1998
- Minas, M.: "Werkzeugunterstützte Erstellung graphischer Diagrammeditoren als Mittel zur Softwarequalitätsverbesserung", ASQF-Arbeitskreis, 3SOFT, Erlangen
28. 04. 1998
- Minas, M.: "Specifying Diagram Languages by Means of Hypergraph Grammars", Workshop "Thinking with Diagrams" (TwD'98), Aberystwyth, Wales
23. 08. 1998
- Minas, M.: "Automatically Generating Environments for Dynamic Diagram Languages" (Posterpräsentation), 1998 IEEE Symposium on Visual Languages (VL'98), Halifax, Nova Scotia, Kanada
02. 09. 1998
- Minas, M.: "Hypergraphs as a Uniform Diagram Representation Model", 6th Intern. Workshop on Theory and Application of Graph Transformation (TAGT '98), Paderborn
16. 11. 1998
- Nilson, J.: "N-GUIDE - ein System zur Erstellung grafischer Benutzeroberflächen", Promotionsvortrag, Univ. Erlangen-Nürnberg
05. 05. 1998
- Wilke, P.: "Anwendungen des Soft-Computing - Ein kurz(weilig)er Streifzug durch die Welt der neuronalen

Netze, evolutionären Strategien und Fuzzy-Logik",
Tag der Informatik, Univ. Erlangen-Nürnberg
24. 04. 1998

Wilke, P.: "Lernen", Seminar "Kognitionswissenschaft und kognitive Psychologie", Univ. Passau
23. 06. 1998

Wilke, P.: "Modellierung und Simulation von Soft-Computing-Systemen",
Univ. Rostock
10. 12. 1998