

# Lehrstuhl für Informatik II

## (Programmier- und Dialogsprachen sowie Compiler)

### Leiter:

Prof. Dr. Hans Jürgen  
Schneider

### Mitarbeiter:

Arius, Peter, Dipl.-Inf.	(wiss. Mitarb., SFB)	
Behnsen, Wolfgang, Dipl.-Math.	(Programmierer)	
Betz, Wolfgang, Dipl.-Inf.	(wiss. Mitarb., SFB)	bis 30. 11. 95
Fischer, Ingrid, Dipl.-Inf.	(wiss. Mitarb.)	beurlaubt: 18. 02. - 08. 10. 95
Hodek, Roman, Dipl.-Inf.	(wiss. Mitarb.)	ab 01. 08. 95
Jacob, Christian, Dr.-Ing.	(wiss. Mitarb.)	
Keil, Martina, M.A.	(wiss. Mitarb.)	
Lührs, Erni	(Sekretärin - halbtags)	
Minas, Mark, Dr.-Ing.	(Akad. Rat)	
Nilson, Jörg, Dipl.-Inf.	(wiss. Mitarb.)	
Schörmal, Elfriede	(Sekretärin - halbtags)	
Uebler, Manfred	(Programmierer)	
Viehstaedt, Gerhard, Dr.-Ing.	(wiss. Mitarb.)	bis 31. 03. 95
Wilke, Peter, Dr.-Ing.	(Akad. Rat)	

### Gastwissenschaftler:

Heidenreich, Georg, Dipl.-Inf.	(BMFT)	
Kokai, Gabriella	(Bay. Kultusministerium)	ab 27. 10. 95
Shen, Bin	(Adenauer-Stiftung)	März bis Sept. 95

### Lehrbeauftragte:

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

### Gäste:

Bräunl, Thomas, Priv.-Doz. Dr. habil., Universität Stuttgart  
Burnett, Margaret, Dr., Oregon State University

Schürr, A., Dr., RWTH Aachen

Taentzer, Gabriele, TU Berlin

---

# 1 Evolvica: Simulierte Evolution natürlicher Entwicklungsprogramme

Die Evolution in der Natur hat eine faszinierende Vielfalt von Lebewesen mit erstaunlichen Fähigkeiten hervorgebracht und stellt ein herausragendes Beispiel natürlicher Adaptionsprozesse dar. Die Anpassung von Organismenpopulationen an ihre Umweltbedingungen ist dabei im wesentlichen auf ein Wechselspiel von Selektionen und Variationen zurückzuführen: Die Baupläne der Organismen, kodiert in den Erbanlagen der Zellen, werden im Laufe der Entwicklung der Generationen variiert, was dazu führt, daß sich schließlich diejenigen Individuen durchsetzen, die, meist aufgrund besonderer Fähigkeiten oder Merkmale, mit den vorherrschenden Umweltbedingungen am besten zurechtkommen.

Aus diesem evolutionären Anpassungsprinzip lassen sich eine Reihe von Konzepten und Strategien zur Lösung von Lernaufgaben und Optimierungsproblemen bei künstlich "intelligenten" technischen Systemen ableiten. Wie sich natürliche Populationen durch die Evolution an ihre Umweltbedingungen anpassen, so können Problemlösungsstrategien - beispielsweise in der Gestalt von Computerprogrammen - selektiv modifiziert und an eine vorgegebene Problemstellung, durch die die Randbedingungen einer "Umwelt" definiert sind, adaptiert werden.

Mit Hilfe evolutionärer Programmiermethoden (Stichwort: "Genetische Programmierung") versuchen wir, einen kleinen Ausschnitt der in der Natur beobachteten Selektions- und Modifikationsverfahren nachzubilden und für die simulierte Evolution problemspezifischer Programme einzusetzen. Kodiert man die Programme in Form von Termen oder symbolischen Ausdrücken, so können einerseits die für Programme typischen hierarchischen Strukturen (Verzweigungen, Schleifen, Module etc.) erfaßt und andererseits strukturmodifizierende Operatoren in einheitlicher Weise definiert werden. Die Erzeugung der "Programmgenome" erfolgt durch einen stochastischen Konstruktionsprozeß auf Grundlage elementarer Strukturbausteine, deren Definition sich aus der gegebenen Optimierungsaufgabe ableitet.

Auf Grundlage der Programmiersprache Mathematica haben wir ein interaktives Lern- und Entwicklungssystem - Evolvica - konzipiert und implementiert, mit dem einerseits verschiedene klassische evolutionsbasierte Algorithmen (Evolutionstrategien, Genetische Algorithmen) anschaulich demonstriert werden, und das andererseits zur Demonstration unserer Forschungen auf dem Gebiet der Programmevolution bzw. Nachbildung von Programmiermechanismen der Natur dient.

Abb. 1: Visualisierung einer Wachstumssimulation, modelliert durch L-Systeme. In Klammern sind die Iterationen der Regelsystemanwendungen angegeben.

Zu komplexen Entwicklungsprozessen führt die simulierte Evolution von Programmen zur Beschreibung naturähnlicher Gestaltbildungs- und Wachstumsvorgänge, die durch parallele Ersetzungssysteme in der Definition von Lindenmayer-Systemen (L-Systemen) modelliert werden. Die symbolischen Ausdrücke kodieren L-Systeme,

die in Analogie zu den Genomprogrammen natürlicher Zellen Entwicklungsvorgänge implizit beschreiben. Die Interpretation der L-System-Genome, in der Natur als Expression bezeichnet, führt beispielsweise zur simulierten Morphogenese pflanzenähnlicher Strukturen ([Abb. 1](#)).

Kann man eine Umwelt in Form einer die Entwicklungsprogramme bewertenden Funktion angeben, so gelangt man auf Grundlage dieser Bewertungen zu einem evolutionären Ausleseverfahren. Werden die Genome außerdem durch genetische Operatoren, im Sinne der Genetischen Programmierung, variiert, so ergibt sich ein Evolutionsverfahren für Entwicklungsprogramme. Evolution als Entwicklungsprozess findet somit auf zwei Ebenen statt: auf der Ebene der Gestaltbildung, gesteuert durch ein individuelles Genomprogramm, und auf einer (Meta-)Ebene von Populationen derartiger Programme, die miteinander in evolutionäre Konkurrenz treten.

Am Beispiel einfacher Gestaltbildungen bei zweidimensionalen fraktalen Strukturen haben wir die Möglichkeiten zur evolutionären Inferenz von L-Systemen aufgezeigt. Weitaus komplexere L-Systeme sind zur Modellierung von Wachstums- und Blütenbildungsvorgängen bei Pflanzen notwendig. Auf Grundlage solcher L-Systeme konnten wir zeigen, wie man auch hier zu einem evolutionsbasierten Verfahren zur Höherentwicklung dieser L-Systeme im Sinne einer impliziten Bewertung der Gestaltbildungen gelangt. Neben einer erstaunlichen Vielfalt an evolvierbaren (pflanzenähnlichen) Strukturen ergeben sich durch Anwendung der genetischen Operatoren, die die Genomstrukturen modifizieren, eine Reihe von Mutationseffekten bei den exprimierten Individuen, die analog auch in der Natur zu beobachten sind.

Abb. 2: Vergleich des Wachstumsverhaltens des besten Individuums der Initialpopulation und des am besten bewerteten Individuums der 13. Generation (Populationsgröße: 6). In Klammern sind die L-Systemiterationen angegeben.

[Abb. 2](#) zeigt das Ergebnis eines typischen Evolutionslaufes zur L-Systeminferenz. Das Ziel besteht in diesem Beispiel darin, L-Systeme zu "züchten", die das Wachstum von Pflanzen mit bestimmten Charakteristiken beschreiben; die Pflanzen sollen möglichst schnell wachsen und außerdem viele Blätter und Blüten erzeugen. Bei den zufällig erzeugten Individuen der Initialpopulation sind die geforderten Merkmale nur unzureichend ausgeprägt; die am besten bewertete Pflanzenstruktur aus Generation 13 weist jedoch bereits erstaunliches Wachstumsverhalten auf.

(Jacob)

## 2 *DiaGen* - Ein Generator für Diagrammeditoren

*DiaGen* ist ein am Lehrstuhl konzipierter und implementierter Generator für Diagrammeditoren. Mit diesem Werkzeug wird die Erstellung graphischer Editoren für jeweils vorzugebende Diagrammklassen stark vereinfacht, indem die Editoren aus einer Spezifikation automatisch generiert werden. Die gegenwärtig erste Fassung von *DiaGen* hat die folgende Funktionalität:

- Die Diagrammklassse, für die ein Editor erstellt werden soll, muß spezifiziert werden. Typische Diagrammklassen sind beispielsweise Struktogramme ("Nassi-ShneidermanDiagramme"), Flußdiagramme und Graphen endlicher Automaten. Die Spezifikation der Diagrammklassse erfolgt mittels einer kontextfreien Hypergraphgrammatik, die um kontextsensitive Elemente erweitert werden kann. Den einzelnen Grammatikproduktionen zugeordnete Constraints beinhalten Layoutinformation.
- Die Interaktion mit dem erstellten Diagrammeditor erfolgt syntaxgesteuert, d. h. jede Diagrammänderung hat nur korrekte Diagramme zur Folge. Beschreibungen der erlaubten Benutzeraktionen und der dadurch ausgelösten Diagrammtransformationen sind ebenfalls Bestandteil der Spezifikation.

Ein Ziel war nun, die Anwendbarkeit dieses Ansatzes zur Editorgenerierung zu zeigen und die vorliegende Version von *DiaGen* unter realen Bedingungen einzusetzen. In Kooperation mit Bell Communications Research wurde die dort entwickelte Sprache IRDL (InfoHarness Repository Definition Language) um eine graphische Variante VRDL (Visual Repository Definition Language) ergänzt. VRDL repräsentiert mit struktogrammähnlichen Diagrammen IRDL-Programme, die - stark vereinfacht - Filter für beliebige Datenbestände beschreiben und diese so auf dem Internet verfügbar und zugänglich machen. Der spezifizierte und mit *DiaGen* generierte graphische Editor für VRDL ermöglicht nun die einfache, anschauliche und suggestive graphische Erstellung dieser Filter. Die Anwendbarkeit von *DiaGen* konnte so in einer realen Anwendung unter Beweis gestellt werden.

Ferner wurde an einer Weiterentwicklung der gegenwärtigen *DiaGen*-Version gearbeitet. Es hat sich gezeigt, daß der bei weitem größte Spezifikationsaufwand auf die Beschreibung von Benutzerinteraktionen und Diagrammtransformationen entfällt. Die Spezifikation der eigentlichen Diagrammklassse schlägt demgegenüber kaum zu Buche (weniger als 20%). Im Moment laufende Arbeiten haben zum Ziel, auf den für die Interaktion zuständigen Spezifikationsanteil weitgehend zu verzichten, den Spezifikationsaufwand drastisch zu reduzieren und rapid prototyping von Diagrammeditoren zu ermöglichen. Die Interaktion mit dem erstellten Diagrammeditor muß hierfür standardisiert werden, so daß sich der Editor zum Teil wie ein Zeichenprogramm mit durch die Diagrammklassse vorgegebenen Bildelementen verhält. Ein Syntaxanalysealgorithmus ("Graphparser") hat fortwährend zu überprüfen, inwieweit die erstellten Diagramme syntaxkonform sind, d. h. den Regeln der Diagrammklassse entsprechen. Ein solcher inkrementeller Graphparser konnte erfolgreich durch Verallgemeinerung des bekannten textuellen Syntaxanalyseverfahrens nach Cocke, Younger und Kasami konzipiert und implementiert werden.

**Literatur:**

Shklar, L./Shah, K./Basu, C.: "Putting legacy data on the Web: a repository definition language", *Computer Networks and ISDN Systems* 27 (1995), pp. 939 - 951

(Minas)

## 3 Phraseo-Lex - eine Lexikondatenbank für Redewendungen

### 3.1 Redewendungen als komplexe sprachliche Mehrworteinheiten

Sprachliche Phänomene wie "ins Gras beißen", "einen Bock schießen", "jemanden auf die Palme bringen", "den Löffel abgeben", "jemandem einen Bären aufbinden" oder "jemanden ein Schnippchen schlagen" werden im Volksmund Redewendungen genannt - im Bereich der Linguistik werden sie auch als Phraseologismen bzw. Idiome bezeichnet.

Sie unterscheiden sich von einfachen Wörtern dadurch, daß sie aus mehreren Elementen bestehen, weisen aber - einfachen Wörtern ähnlich - als Komplex eine Gesamtbedeutung auf, die sich von der Bedeutung der einzelnen Wörter, aus denen die Redewendung zusammengesetzt ist, unterscheidet. So ist die Bedeutung "ins Gras beißen", die mit 'sterben' umschrieben werden kann, unabhängig von der Bedeutung der Wörter "ins", "Gras" und "beißen". Solche phraseologischen Mehrworteinheiten zeichnen sich durch eine Reihe weiterer spezifischer Eigenschaften aus, die bei nicht-phraseologischen Wortverbindungen nicht vorkommen. So treten beispielsweise in Phraseologismen oft Wörter auf, die außerhalb von Redewendungen nicht verwendet werden wie "Laufpaß", "Schnippchen", "Kerbholz", "Mißkredit" usw. Darüber hinaus unterliegen einzelne Redewendungen verschiedenen Modifikations- und Umformungsbeschränkungen, die nicht-phraseologische sprachliche Mehrworteinheiten ohne Probleme zulassen. Die folgenden Beispiele deuten allerdings an, daß diese Beschränkungen nicht für alle Redewendungen gleichermaßen gelten: (Der Stern \* kennzeichnet Sätze, die keine phraseologische Lesart zulassen und damit nur wörtlich zu verstehen sind, was in einigen Fällen zu absurden Inhalten führt.)

1. \*Er hat ins *nasse/tödliche* Gras gebissen.
2. \**Auf was für eine Palme* hast Du sie gebracht?
3. \*Der Löffel *wurde* von ihm *abgegeben*.
4. *Was für einen Bären* hast Du ihr aufgebunden?
5. Er hat einen *unglaublichen* Bock geschossen.

Für die maschinelle Verarbeitung natürlicher Sprache ergeben sich daraus zahlreiche Probleme. Denn die gewöhnlichen Informationen zu einzelnen Wörtern wie "Gras", "Löffel", "Bock", "abgeben", "schießen", "bringen" usw., die in maschinenlesbaren Lexika gespeichert sind, reichen für die Verarbeitung von phraseologischen Sprachstrukturen nicht aus.

### 3.2 Ein phraseologisches Informationssystem

Aus den aufgezeigten Problemen erwuchs die Idee, eine Lexikondatenbank - Phraseo-Lex genannt - zu entwickeln, in der die spezifischen Eigenschaften dieser komplexen Mehrworteinheiten adäquat repräsentiert werden können. Die Konzeptionierung des Systems PhraseoLex erfolgte im Hinblick auf verschiedene Zielrichtungen:

- **Systematische Erfassung**

Traditionellen Linguisten, die aus verschiedensten Gründen an der Systematisierung von Redewendungen und deren Vielgestaltigkeit interessiert sind, wird mit Phraseo-Lex ein Werkzeug an die Hand gegeben, mit dem zunächst einmal eine SYSTEMATISCHE ERFASSUNG dieser Phänomene ermöglicht wird. Die Ergebnisse des Erfassungsvorganges können beispielsweise als Grundlage einer darauf basierenden Wörterbuchschreibung dienen.

- **Analysewerkzeug**

Zudem ist es mit Hilfe von Phraseo-Lex möglich, die erfaßten Informationen zu Redewendungen STATISTISCH AUSZUWERTEN und INHALTLICH MITEINANDER ZU VERGLEICHEN, um weitere Aufschlüsse über Redewendungen zu gewinnen. Damit ist eine tiefergehende Klassifizierung des vielschichtigen Gebietes möglich.

- **Zusatzlexikon in der Sprachverarbeitung**

Darüber hinaus dient Phraseo-Lex Computerlinguisten in komplexen Sprachverarbeitungssystemen als Zusatzlexikon zum einfachen "Ein-Wort-Lexikon" (Simplex-Lexikon). Einen beispielhaften Einsatz erfährt Phraseo-Lex gegenwärtig in der Anbindung an ein Chart-Parsing-System, das am IMMD VIII entwickelt wurde.

### **3.3 Interne Struktur eines phraseologischen Eintrags**

Schlüssel eines jeden phraseologischen Eintrags ist das LEMMA, also die Nennform/Grundform einer Redewendung wie beispielsweise "jemandem einen Bären aufbinden" oder "einen Bock schießen". Während diese unflektierte Darstellungsform vor allem dem traditionellen Linguisten nutzt, ist sie beim Sprachverarbeitungsprozeß wenig hilfreich, da eine solche unflektierte Grundform von Redewendungen in realistischen Sätzen selten vorkommt. (Dies ist in etwa vergleichbar mit dem Auftreten des Infinitivs eines Verbs.) Von Interesse für den Sprachverarbeitungsprozeß ist daher vor allem der Zugriff auf die Informationen eines phraseologischen Eintrags über einzelne Wörter des Phraseologismus wie "Bär", "Bock", "aufbinden", "schießen", usw., die als BASISLEXEME bezeichnet werden. Daher werden sowohl Lemma als auch Basislexeme in jedem phraseologischen Eintrag angegeben.

Weiterhin kann die zu erfassende Information in syntaktische, semantische und pragmatische Information unterschieden werden.

- **Ebene der Syntax**

Während einfache Wörter in gewöhnlichen Wörterbüchern durch Angabe einer Wortart (Verb, Nomen, Adjektiv usw.) syntaktisch kategorisiert werden, weist eine Redewendung, die ja eine komplexe Phrase darstellt, auch intern eine



syntaktische Struktur auf, die mit Hilfe eines Phrasenstrukturbaums in Phraseo-Lex kodiert werden kann. Außerdem können auf syntaktischer Ebene Angaben zu Modifikations- und Umformungsmöglichkeiten und Angaben zu Valenzstellen und Varianten von Phraseologismen gemacht werden.

- **Ebene der Semantik**

Zentral an der Semantik von Redewendungen ist ihre nicht-wörtliche Bedeutung. Daher wird in den Mittelpunkt der semantischen Beschreibungsebene die Erfassung der Bedeutung mit Hilfe einer Paraphrase gestellt. So kann beispielsweise der Phraseologismus "einen Bock schießen" durch "einen Fehler machen" umschrieben werden. Zudem erfolgt eine Kodierung der Bedeutung durch logische Formeln: "jemand schießt einen Bock" wird repräsentiert durch "machen(X,Y)" und "fehler(Y)". Eine solche Kodierung spielt für die Bedeutungsrepräsentation in Sprachverarbeitungssystemen (beispielsweise in der Diskursrepräsentationstheorie) eine große Rolle. Darüber hinaus werden auf semantischer Ebene unter anderem Angaben zur Motiviertheit und Mehrdeutigkeit und zu Synonymen und Antonymen von Phraseologismen erfaßt.

- **Ebene der Pragmatik**

Auf der Ebene Pragmatik können diasystematische Angaben und Konnotationen von Phraseologismen spezifiziert werden. So kann beispielsweise angegeben werden, ob es sich um einen normalsprachlichen ("jemandem den Rücken stärken"), umgangssprachlichen ("einen Bock schießen"), bildungssprachlichen ("in medias res gehen"), derben ("jemandem in den Arsch kriechen") oder anderen Phraseologismus handelt. Solche Informationen sind insbesondere für Fremdsprachler wichtig und sind bei der Wörterbuchschreibung zu berücksichtigen.

### **3.4 Implementierung**

Der Datenbankkern von Phraseo-Lex wurde zunächst in SCHEME und alternativ in PROLOG implementiert, da somit eine einfache Einbindung in existierende Sprachverarbeitungssysteme gewährleistet ist. Die SCHEME-Version ist mit einer komfortablen Bedienoberfläche (OSF/Motif) ausgestattet. In jüngster Zeit wurde das überarbeitete PhraseoLex-Konzept auch mit Hilfe des geschachtelt relationalen Datenbanksystems Postgres umgesetzt. Auch hierzu wird gegenwärtig eine geeignete Bedienoberfläche unter OSF/Motif entwickelt.

(Keil)

---

# Dissertationen:

Jacob, C.:

*MathEvolvica* - Über die simulierte Evolution von Entwicklungsprogrammen der Natur

Keil, M.:

Modell zur Repräsentation verbaler Phraseologismen - Konzept und Implementierung eines Werkzeugs zum Einsatz in der Phraseologieforschung, Phraseographie und maschinellen Sprachverarbeitung

---

# Diplomarbeiten:

Beck, R.:

Entwurf und Implementierung einer ereignisgesteuerten Protokollierung für ein Simulatorsystem für künstliche neuronale Netze (NeuroGraph)

Billing, G.:

Erweiterung einer objektorientierten Datenstruktur um eine Codeerzeugungskomponente für ein Simulatorsystem für künstliche neuronale Netze (NeuroGraph)

Evert, S.:

Klassendesign für technische Berechnungsprogramme

Förther, R.:

Entwurf und Implementierung einer generischen, graphischen Benutzerschnittstelle für Debugger

Gold, S. S.:

Parameteroptimierung für ein Kühlstreckenmodell

Hodek, R.:

Eine graphische Oberfläche für die Flowback-Analyse in HERAKLIT

Holland, B.:

Objektorientierte Modellierung flexibler Fertigungssteuerungen in HERAKLIT

Landauer, J.:

Visual AWK - Programming by Demonstration Based on the Pattern-Action Paradigm

Mangold, T.:

Erkennung von existentiell quantifizierten globalen Prädikaten in parallelen Programmen

Minge, R.:

Integration eines C++-Sourcecode-Analysewerkzeugs in die Entwicklungsdatenbank einer kommerziellen Software-Produktionsumgebung

Özdemir, R.:

Vergleich verschiedener Debuggingverfahren bzgl. ihrer Anwendbarkeit für verteilte objektorientierte Programmiersprachen

Rehder, J.:

Entwurf einer Komponente für genetische Algorithmen und Fuzzy-Logik eines Simulatorsystems (NeuroGraph)

Sandig, T.:

Paralleles Rechnen mit Mathematica

Schmidt, H.:

Archivierungs-, Analyse- und Visualisierungskonzepte für die evolutionäre Programmierung

Siegert, O.:

Datenbankeinsatz im Rahmen von Phraseo-Lex

Stumpp, M.:

Realisierung der Flowback-Analyse für HERAKLIT - Debuggingphase

Tlili, N.:

Vergleichende Darstellung verbreiteter Systeme zur Konfigurationsverwaltung

Utz, W.:

Realisierung der Flowback-Analyse für HERAKLIT - Statische Phase -

---

# Studienarbeiten:

Balzer, P.:

Implementierung eines Grammatikregelsystems zur Analyse modifizierter verbaler Phraseologismen und deren Lesarten

Billstein, M.:

3-dimensionale Anzeigefenster und Editoren für ein Simulatorsystem für künstliche neuronale Netze (NeuroGraph)

Engel, K. D.:

Entwurf und Implementierung eines 'Requirement Inspection and Validation Editors' (RIVALE)

Frisch, N.:

Untersuchungen zur Programmevolution in Tierra - Visualisierungsmodul -

Gälli, M.:

Entwicklung von Fraktalen durch Genetische Programmierung von L-Systemen

Henter, R.:

Untersuchungen zur Programmevolution in Tierra - Analysemodul -

Herbst, M.:

Modellierung und Visualisierung genetisch programmierter zellulärer neuronaler Netze

Köhn, P.:

Genetische Algorithmen und neuronale Netze: Das Kodierungsproblem

Krüger, J.:

Parsing verbaler Phraseologismen durch eine mit 'Phraseo-Lex' kombinierte Chart

Lenke, D.:

Modellierung von Strukturen und Regulationsmechanismen natürlicher Genome

Maltz, W.:

Evolutionsbasierte interaktive Konstruktion Neuronaler Netze

Melchers, D.:

Implementierung von Unix-Werkzeugen zur Installation und Ausführung eines Simulators für künstliche neuronale Netze (NeuroGraph)

Meyer-Kayser, J.:

Entwurf und Implementierung einer Visualisierungskomponente für ein Simulatorsystem für künstliche neuronale Netze (NeuroGraph)

Neeb, J.:

Untersuchungen zur Kodierung und evolutionären Entwicklung hierarchischer neuronaler Netze

Tietze, D.:

Visualisierung der Gestaltbildung zellulärer neuronaler Netze im zwei- und dreidimensionalen Raum

---

# Vorträge:

Bräunl, T.:

"Parallelverarbeitung und Mobile Roboter", Informatik-Kolloquium der Univ. Erlangen-Nürnberg  
02. 06. 1995

Burnett, M.:

"Integrating Algorithm Animation into a Visual Programming Language", Informatik-Kolloquium der Univ. Erlangen-Nürnberg  
12. 09. 1995

Heidenreich, G.:

"Kompatibilitätsvektoren", Betriebsseminar der Fa. BASYS GmbH, Erlangen, Schwend/Opf.  
24. 11. 1995

Jacob, C.:

"Evolution und Züchtung von Computerprogrammen", Graphik-Kolloquium des Lehrstuhls für Graphische Datenverarbeitung der Univ. Erlangen-Nürnberg  
13. 01. 1995

Jacob, C.:

"Evolution und Züchtung von Computerprogrammen - Simulierte Evolution von Entwicklungsprogrammen der Natur", Informatik-Kolloquium der Univ. Leipzig, Fakultät für Mathematik und Informatik, Leipzig  
13. 07. 1995

Jacob, C.:

"Genetic L-system programming: breeding and evolving artificial flowers with Mathematica", IMS'95, First International Mathematica Symposium, Southampton (UK)  
20. 07. 1995

Keil, M.:

"Warum man auf einer Sitzung zwar einen großen Bock, aber keinen jungen Bock schießen kann", Fachgruppentreffen 'Frauenarbeit und Informatik' der Gesellschaft für Informatik (GI), Hagen  
05. 03. 1995

Keil, M.:

"'Nicht auf die lange Bank geschoben' - Systematische Erfassung und Analyse von Phraseologismen in der Lexikondatenbank Phraseo-Lex", Kolloquium im Institut für Semantische Informationsverarbeitung der Univ. Osnabrück  
21. 11. 1995

Keil, M.:

"'Da geht dem Rechner der Hut hoch' - Ein Ausflug in die Computerlinguistik", Betriebsseminar der Fa. BASYS GmbH, Erlangen, Schwend/Opf.  
24. 11. 1995

Minas, M.:

"Graphical Representation and Manipulation of Complex Structures Based on a Formal Model", CAiSE'95, Jyväskylä (Finnland)  
15. 06. 1995

Minas, M.:

- "*DiaGen: A Generator for Diagram Editors Based on a Hypergraph Model*",  
NGITS'95, Naharia (Israel)  
29. 06. 1995
- Minas, M.:  
"Generating Editors for Direct Manipulation of Diagrams", EWHCI'95, Moskau  
(Rußland)  
04. 07. 1995
- Minas, M.:  
"Detecting Quantified Global Predicates in Parallel Programs", EURO-PAR'95,  
Stockholm (Schweden)  
30. 08. 1995
- Minas, M.:  
"*DiaGen: A Generator for Diagram Editors Providing Direct Manipulation and  
Execution of Diagrams*", VL'95, Darmstadt  
08. 09. 1995
- Schneider, H. J.:  
"Graphgrammatiken als Werkzeug zur Beschreibung von Prozeßsystemen -  
Von Petri-Netzen bis Linda -", Informatik-Kolloquium der Univ. Erlangen-Nürnberg  
30. 01. 1995
- Schneider, H. J.:  
"Wege zwischen Wissenschaft und Wirtschaft", ESPITI-Fachforum 'Qualität  
führt zusammen', Erlangen  
20. 06. 1995
- Schneider, H. J.:  
"Programmiersprachen als Werkzeug der Kommunikation", Collegium  
Alexandrinum, Erlangen  
22. 06. 1995
- Schneider, H. J.:  
"A note on outward and inward productions in the categorical graph-grammar  
Approach and Delta-grammars", SEGRAGRA Workshop, Volterra (Italien)  
28. 08. 1995
- Schürr, A.:  
"PROGRES: Eine Programmierumgebung für Graphgrammatiken",  
Informatik-Kolloquium der Univ. Erlangen-Nürnberg  
23. 02. 1995
- Taentzer, G.:  
"Verteilte Graphtransformation und ihr Nutzen zur Systementwicklung",  
Informatik-Kolloquium der Univ. Erlangen-Nürnberg  
28. 11. 1995
- Wilke, P.:  
"Integration of Genetic Algorithms and Fuzzy-Logic into a Neural Network  
Simulation Environment", Mascots, Durham, USA  
19. 01. 1995
- Wilke, P.:  
"Neuronale Netze", RRZE-Kolloquium der Univ. Erlangen-Nürnberg  
31. 01. 1995
- Wilke, P.:  
"A Simulation Environment for Neural Networks, Genetic Algorithms and Fuzzy

Logic", ICANNGA 95 Ales, Frankreich

21. 04. 1995

Wilke, P.:

"Neuronale Netze: Einführung und Überblick", Carl-Cranz-Gesellschaft CCG,  
Oberpfaffenhofen

04. 07. 1995

Wilke, P.:

"Netzwerkmodell: Kohonen Feature Map", Carl-Cranz-Gesellschaft CCG,  
Oberpfaffenhofen

05. 07. 1995

Wilke, P.:

"Hard- und Software für neuronale Netze", Carl-Cranz-Gesellschaft CCG,  
Oberpfaffenhofen

06. 07. 1995

Wilke, P.:

"NeuroGraph - An Integrated Development Environment for Neural Networks,  
Genetic Algorithms and Fuzzy Logic", Eufit 95, Aachen

30. 08. 1995



# Veröffentlichungen:

Fischer, I./ Geistert, B./ Görz, G.:

"Chart-based Incremental Semantics Construction with Anaphora Resolution Using lambda-DRT". - In: *Proc. of the Fourth International Workshop on Parsing Technologies*, 20. - 24. 09. 1995, S. 87 - 88

Heidenreich, G./ Kips, D.:

"A General Model for Engineering Databases". - In: *Proc. 1st World Conf. on Integrated Design & Process Technology*, Austin, TX, Dec. 7 - 9, 1995, Society for Design & Process Science, Dallas, Texas, S. 347 - 352

Jacob, C.:

"Modeling Growth with L-Systems & Mathematica". - In: *Mathematica in Education and Research*, Vol. 4, No. 3 (1995), TELOS-Springer, New York, 1995, S. 12 -19

Jacob, C.:

"Genetic L-system programming: breeding and evolving artificial flowers with Mathematica". - In: V. Keränen, P. Mitic, *Mathematics with Vision, Proc. of the First International Mathematica Symposium, IMS'95*, Computational Mechanics Publications, Southampton, 1995

Jacob, C.:

"*MathEvolvica - Simulierte Evolution von Entwicklungsprogrammen der Natur*", Dissertation. - Arbeitsber. des IMMD der Univ. Erlangen-Nürnberg, Band 28, Nr. 10, 1995

Jacob, C./ Rehder, J./ Siemandel, J./ Friedmann, A.:

"XNeuroGene: A system for evolving artificial neural networks". - In: *Proc. of the Third Intern. Workshop 'Modeling, Analysis, and Simulation of Computer and Telecommunication Systems' (MASCOTS'95)*, Durham, North Carolina, Jan. 18 - 20, 1995. - Los Alamitos, CA, USA: IEEE Computer Society Press, 1995, S. 436 - 439

Keil, M.:

"*Modell zur Repräsentation verbaler Phraseologismen - Konzept und Implementierung eines Werkzeugs zum Einsatz in der Phraseologieforschung, Phraseographie und maschinellen Sprachverarbeitung (Phraseo-Lex)*". - Erscheint voraussichtlich in der Reihe 'Sprache und Information', Niemeyer-Verlag, Tübingen, Frühjahr 1997

Kips, D./ Heidenreich, G.:

"Project Flow Graphs - A Meta Model for Quality Assurance in Software Engineering". - In: *Proc. 2nd Int. Conf. on Industrial Engineering and Production Management*, Marrakesch, 03. - 07.04.1995, FuCaM, Mons (Hrsg.), Band 1, S. 347 - 357

Landauer, J./ Hirakawa, M.:

"Visual AWK: A Model for Text Processing by Demonstration". - In: Haarslev, V. (Hrsg.), *Proc. 11th IEEE International Symposium on Visual Languages (VL'95)*, Darmstadt, S. 267 - 274, IEEE Computer Society Press, September 1995

Minas, M.:

"Detecting quantified global predicates in parallel programs". - In: Haridi, S./Ali, K./Magnusson, P. (Hrsg.), *Proc. 1st International EURO-PAR Conference*

(EURO-PAR '95), *Parallel Processing*, Stockholm (Schweden), LNCS 966, S. 403 - 414, Springer-Verlag, August 1995

Minas, M./ Viehstaedt, G.:

"DiaGen: A generator for diagram editors providing direct manipulation and execution of diagrams". - In: Haarslev, V. (Hrsg.), *Proc. 11th IEEE International Symposium on Visual Languages (VL'95)*, Darmstadt, S. 203 - 210, IEEE Computer Society Press, September 1995

Schneider, H. J.:

"Parallele Programmiersprachen". - In: *'Parallelrechner'*, K. Waldschmidt (Hrsg.). - Stuttgart: Teubner, 1995. S. 411 - 470

Schneider, H. J.:

"A note on outward and inward productions in the categorical graph-grammar Approach and Delta-grammars". *Electronic Notes in Theoretical Computer Science* 2 (1995)

Viehstaedt, G.:

"A Generator for Diagram Editors", Dissertation. - Arbeitsber. des IMMD der Univ. Erlangen-Nürnberg, Band 28, Nr. 2, 1995

Viehstaedt, G./ Minas, M.:

"Graphical representation and manipulation of complex structures based on a formal model". In: Iivari, J./Lyytinen, K./Rossi, M. (Hrsg.), *Proc. 7th International Conference on Advanced Information Systems Engineering (CAiSE'95)*, Jyväskylä (Finnland), LNCS 932, S. 243 - 254, Springer-Verlag, Juni 1995

Viehstaedt, G./ Minas, M.:

"DiaGen: A generator for diagram editors based on a hypergraph model". - In: Motro, A./Tennenholtz, M. (Hrsg.), *Proc. 2nd International Workshop on Next Generation Information Technologies and Systems (NGITS '95)*, Naharia (Israel), S. 155 - 162, Juni 1995

Viehstaedt, G./ Minas, M.:

"Generating editors for direct manipulation of diagrams". - In: Blumenthal, B./Gornostaev, J./Unger, C. (Hrsg.), *Proc. 5th International Conference on Human-Computer Interaction (EWHCI'95)*, Moskau (Rußland), LNCS 1015, S. 17 - 25, Springer-Verlag, Juli 1995

Wilke, P./ Billing, G./ Mansfeld, C./ Nilson, J.:

"Integration of Genetic Algorithms and Fuzzy-Logic into a Neural Network Simulation Environment". - In: *Proc. of the Third Intern. Workshop 'Modeling, Analysis, and Simulation of Computer and Telecommunication Systems' (MASCOTS'95)*, Durham, North Carolina, Jan. 18 - 20, 1995. - Los Alamitos, CA, USA: IEEE Computer Society Press, 1995, S. 330 - 333

Wilke, P.:

"A Simulation Environment for Neural Networks, Genetic Algorithms and Fuzzy Logic", *Proc. ICANNGA Ales*, Frankreich

Wilke, P./ Rehder, J./ Billing, G./ Nilson, J./ Mansfeld, C.:

"NeuroGraph - An Integrated Development Environment for Neural Networks, Genetic Algorithms and Fuzzy Logic". - In: *Proc. of the 'Third European Congress on Intelligent Techniques and Soft Computing' (EUFIT'95)*, Zimmermann, H.-J. (Hrsg.), Aachen, Aug. 28 - 31, 1995. - Aachen: Verlag Mainz, Wissenschaftsverlag, S. 1879 - 1883

