



**TECHNISCHE FAKULTÄT DER
CHRISTIAN-ALBRECHTS-UNIVERSITÄT
ZU KIEL**

C | A | U

Christian-Albrechts-Universität zu Kiel

almanach 05

Programmiersprachen und Übersetzerkonstruktion

Die Arbeiten des Lehrstuhls für Programmiersprachen und Übersetzerkonstruktion konzentrieren sich auf den Bereich des Entwurfs, der Realisierung und der Anwendung von Programmiersprachen zur zuverlässigen Programmierung komplexer Systeme. Die Untersuchungen reichen dabei von objektorientierten Entwurfsmethoden über die Analyse verteilter Systeme bis zur Implementierung und Anwendung deklarativer Programmiersprachen. Im Berichtszeitraum wurden die wissenschaftlichen Arbeiten durch die DFG unterstützt. Die internationalen Kontakte zur Technischen Universität Valencia (Spanien), zur Portland State University (USA) und zur University of Kent (England) führten zu erfolgreichen gemeinsamen Forschungsarbeiten.

Ergebnisse

Im Berichtszeitraum lag der Schwerpunkt der Forschungsarbeiten in den Bereichen Konzeption, Implementierung und Anwendung deklarativer Programmiersprachen. Insbesondere wurde die Zusammenarbeit mit der Portland State University (Oregon, USA) durch einen längeren Forschungsaufenthalt von Prof. Michael Hanus am Partnerinstitut intensiviert.

Im Bereich der Konzeption deklarativer Programmiersprachen wurden in Zusammenarbeit mit der Portland State University konzeptuelle Fragestellungen beim Entwurf deklarativer Programmiersprachen behandelt. Zum einen wurde ein neues Konzept zum Pattern Matching entwickelt, genannt funktionale Muster, mit denen man Abstraktionen für Muster definieren kann, was in existierenden deklarativen Sprachen bisher nicht möglich ist. Andererseits erhöht sich die Ausdruckskraft, weil mit funktionalen Mustern Funktionen invertierbar und damit wiederverwendbar werden, ohne diese vollständig zu evaluieren. Darüberhinaus erhöhen funktionale Muster die Lesbarkeit und Wartbarkeit deklarativer Programme. In einem anderen Projekt haben wir uns mit dem minimalen Kern logisch-funktionaler Sprachen beschäftigt, d.h. der Fragestellung, ob bestimmte Sprachelemente eventuell durch andere dargestellt werden können. Dabei konnten wir zeigen, dass sowohl überlappende Regeln durch logische Variablen als auch logische Variablen durch überlappende Regeln modelliert werden können. Eine praktische Konsequenz dieser Erkenntnis ist die Tatsache, dass Implementierungen logisch-funktionaler Sprachen einfacher gestaltet werden können, da nur eines dieser Konzepte fest implementiert werden muss und das jeweils andere darauf abgebildet werden kann.

Weiterhin wurde die Zusammenarbeit mit der Portland State University (USA) im Bereich der Implementierung deklarativer Sprachen fortgeführt. Insbesondere wurde eine virtuelle Maschine zur Übersetzung deklarativer Programme entworfen und praktisch umgesetzt. Diese Maschine basiert auf neuesten Erkenntnissen zur Theorie der optimalen Auswertung deklarativer Programme. Darüberhinaus realisiert diese Maschine im Gegensatz zu bisherigen Implementierungen eine faire Suchstrategie zur Ausführung nichtdeterministischer Berechnungen.

Im Bereich der Anwendung deklarativer Programmiersprachen ist die Zusammenarbeit mit der Technischen Universität Valencia im Berichtszeitraum fortgeführt worden, wobei der Schwerpunkt auf der Entwicklung neuer Techniken zum Debugging deklarativer Programmiersprachen lag. Da deklarative Programmiersprachen stärker als herkömmliche Programmiersprachen von konkreten Rechnerarchitekturen abstrahieren, bieten sie ein höheres Programmierniveau. Insbesondere liegt ihnen ein anderes Ausführungsmodell zugrunde, welches sich aus Gründen der Optimierung der Ausführungsreihenfolge nicht am sequentiellen Programmtext orientiert. In vielen Fällen führt dies zu einer effizienteren Programmentwicklung. Falls jedoch Programmierfehler auftreten, sind herkömmliche Debugging-Methoden durch Verfolgung des konkreten Berechnungsablaufs unzureichend. Aus diesem Grund wurden in diesem Projekt neue Debugging-Methoden entwickelt, die auf die Besonderheiten deklarativer Programmiersprachen zugeschnitten sind. Nachdem in den Vorjahren durch die Definition einer geeigneten Semantik ein theoretisches Fundament für Debugging-Methoden gelegt worden ist, wurden nun Tracing-Methoden entwickelt, mit denen man den Berechnungsverlauf auf einem abstrakten Niveau verfolgen und daher Fehler leicht aufspüren kann. Ein weiterer Schwerpunkt dieser Zusammenarbeit war die Entwicklung von Methoden und Werkzeugen zum Performance-Debugging, d.h. zur Beobachtung von Programmen bezüglich ihres

Ressourcenverbrauchs. Ziel ist dabei das Erkennen von Programmteilen, die besonders viel Zeit oder Speicher verbrauchen, um diese Programmteile gezielt zu optimieren. Zu diesem Zweck wurde ein Profiling-Werkzeug entwickelt, mit dem der Ressourcenverbrauch deklarativer Programme sowohl symbolisch (bezüglich einzelner Basisoperationen) als auch in realen Größen analysiert werden kann.

In einer weiteren Zusammenarbeit mit der University of Kent (England) wurde ein Programmierwerkzeug entwickelt, mit dem man Berechnungen in logisch-funktionalen Programmen beobachten kann, in dem man bestimmte Programmobjekte zur Beobachtung kennzeichnet. Da diese Beobachtungsobjekte sowohl Daten als auch Funktionen oder logische Variablen sein können, kann sich der Programmierer hiermit einen guten Überblick über die Auswertung der jeweiligen Programmobjekte in einem Programmablauf verschaffen. Dieses Werkzeug wurde im Berichtszeitraum mit einer neuen graphischen Schnittstelle versehen, die es dem Programmierer ermöglicht, sehr einfach neue Beobachtungsobjekte zu definieren und die Programme bezüglich dieser Beobachtungen ablaufen zu lassen.

Die Programmierung nebenläufiger und verteilter Systeme gestaltet sich in der Praxis oft schwierig, da Deadlocks und Lifelocks verhindert werden sollten. Neue Kommunikationsmechanismen können den Programmierer hierbei sehr gut unterstützen. Für Concurrent Haskell wurden die von Datenbanken bekannten Transaktionen als Kommunikationsabstraktion zur Synchronisation und Kommunikation nebenläufiger Prozesse erfolgreich eingesetzt. Wir haben im Berichtszeitraum eine Methode entwickelt, um dieses Transaktionskonzept auch in Concurrent Haskell selber zu implementieren. Diese Implementierung führt zum einen zu einem besseren Verständnis des Ansatzes, bildet zum anderen aber auch eine Basis für potentielle Erweiterungen bis hin zu einem neuen, abstrakteren Konzept zur verteilten Programmierung.

Um die Sicherheit bei der Programmierung mit deklarativen Sprachen zu verbessern, wurde eine neue Analysetechnik entwickelt, mit deren Hilfe man unerwünschte nichtdeterministische Berechnungen schon zur Übersetzungszeit entdecken kann. Zur besseren Unterstützung der Programmierung von Anwendungssystemen mit deklarativen Sprachen wurde ein neues Konzept zur Verwaltung persistenter Daten in deklarativen Programmen vorgeschlagen und mit Hilfe einer relationalen Datenbankimplementierung praktisch umgesetzt. Hierdurch wird eine für den Programmierer einfache und sichere Einbindung relationaler Datenbanken in deklarative Programme ermöglicht. Diese Konzepte wurden in Verbindung mit den schon vorhandenen Konzepten zur Programmierung dynamischer Web-Seiten in einem studentischen Fortgeschrittenenpraktikum verwendet, um ein neues System zur Unterstützung von E-Learning zu realisieren. Konkret wurde das System SOL entwickelt und praktisch implementiert, mit dem Lernende Übungen ortsungebunden bearbeiten und abgeben können. Dabei kann die Art der Übungsaufgaben von Multiple-Choice- über Lückentext- bis hin zu Programmieraufgaben variieren, wobei die Lösungen bei der Abgabe automatisch überprüft werden. Dieses System wurde bei der Anfängervorlesung in der Informatik im Wintersemester 2005/06 erstmalig eingesetzt.

Im Bereich der Öffentlichkeitsarbeit beteiligte sich der Lehrstuhl an der Durchführung des Girls' Day am 28. April 2005. Hierbei wurde mit den Schülerinnen ein kleiner Programmierkurs durchgeführt mit dem Ziel, einfache grundlegende Programmiertechniken am Beispiel der Programmierung des Marienkäfers Kara zu erlernen. Dieses Beispiel wurde auch als Einführung in die Programmierung im Rahmen des Schnupperstudiums Informatik für Schülerinnen verwendet, das inhaltlich von Dr. Frank Huch ausgestaltet wurde. Hierbei erhielten die Schülerinnen auch eine Einführung in die verteilte Programmierung mit der Sprache Erlang und implementierten als Abschlussprojekt einen verteilten Chat.



Leiter/-innen: Prof. Dr. Michael Hanus; Sekretariat: Ulrike Pollakowski-Geuther
Technisches Personal: Dipl.-Ing. Thomas Heß (50%)

Wissenschaftliche Mitarbeiter/-innen:

Dr.phil. Bernd Braßel

01.01.-31.12.2005

DFG + CAU

Fehlersuche in deklarativen Programmen



Abb. 1: Die Kursleitersicht im Web-basierten Übungssystem SOL

Dipl.-Inf. Sebastian Fischer	15.09.-31.12.2005	CAU
Priv.-Doz. Dr. Wolfgang Goerigk	01.01.-31.12.2005	CAU
Dr. Frank Huch	01.01.-31.12.2005	CAU
Dipl.-Inf. Klaus Höppner	01.-31.01.2005	CAU
Dr. Friedemann Simon	01.01.-31.12.2005	CAU

Vorlesungen, Seminare und Praktika

Winter 2004/2005

Prinzipien von Programmiersprachen, 4 (+ 2) Std. Vorlesung (+ Übungen)/Woche,
Michael Hanus (+ Frank Huch)

Probleme beim Bau voll korrekter Übersetzer, 4 (+ 2) Std. Vorlesung (+ Übungen)/Woche,
Wolfgang Goerigk

Objektorientierte Programmierung, 2 (+ 4) Std. Vorlesung (+ Übungen)/Woche,
Friedemann Simon

Fortgeschrittenen-Praktikum Internet-Programmierung, 4 Std. Praktikum/Woche,
Michael Hanus (+ Bernd Braßel)

Diplomandenseminar, 2 Std. Seminar/Woche,
Michael Hanus

Arbeitsgemeinschaft Informatik, Logik und Mathematik, 2 Std. Seminar/Woche,

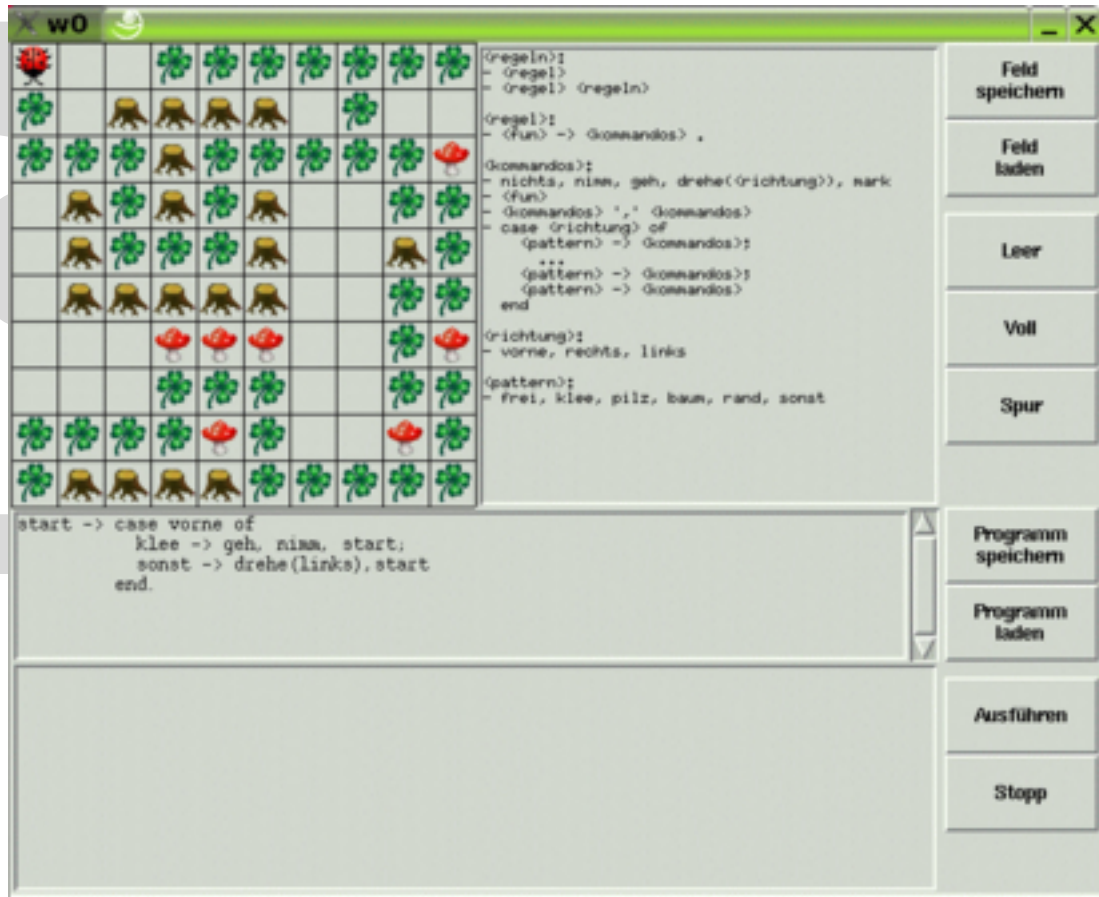


Abb. 2: Spielerisch programmieren lernen mit dem Marienkäfer Kara, Girlsday / Schnupperstudium Informatik für Schülerinnen

M. Hanus, R. Berghammer

Sommer 2005

Systematisches Programmieren, 2 (+ 4) Std. Vorlesung (+ Übungen)/Woche,
Friedemann Simon

Arbeitsgemeinschaft Informatik, Logik und Mathematik, 2 Std. Seminar/Woche,
M. Hanus, R. Berghammer

Funktionale Programmierung, 4 (+ 2) Std. Vorlesung (+ Übungen)/Woche,
Frank Huch (+ Michael Hanus)

Implementierung objektorientierter Sprachen, 4 (+ 2) Std. Vorlesung (+ Übungen)/Woche,
Wolfgang Goerigk

Winter 2005/2006

Diplomandenseminar, 2 Std. Seminar/Woche,
Michael Hanus

Arbeitsgemeinschaft Informatik, Logik und Mathematik, 2 Std. Seminar/Woche,
M. Hanus, R. Berghammer

Fortgeschrittene Techniken der funktionalen Programmierung, 2 Std. Seminar/Woche,
Michael Hanus (+ Frank Huch)

Informatik I (Programmierung/G1.1), 4 (+ 2) Std. Vorlesung (+ Übungen)/Woche,
Michael Hanus (+ Sebastian Fischer, Britta Kehden, Harald Fecher)

Programmierpraktikum (P1-G1.3), 3 Std. Praktikum/Woche,
Frank Huch (+ Michael Hanus)

Programmiertechniken für die künstliche Intelligenz, 2 (+ 4) Std. Vorlesung (+ Übungen)/Woche,
Friedemann Simon

◆ Drittmittel

DFG, *Systematische Fehlersuche in deklarativen Programmen*, 01.10.2004-31.12.2006 (127087 EUR)

◆ Weitere Zusammenarbeiten, Technologietransfers und Konsultationen

Während des Berichtszeitraumes erfolgten Kooperationen mit:

University of Kent (Olaf Chitil)

Universidad Complutense Madrid (Elvira Albert)

Technische Universität Valencia (Maria Alpuente, Salvador Lucas, Josep Silva, German Vidal)

Portland State University (Sergio Antoy und Andrew Tolmach)

◆ Diplom- und Master-Arbeiten

M. Engelke, *Refactoring für logisch-funktionale Programme*, 10.02.2005

S. Herhut, *Ein Modulsystem zur Unterstützung von subtypbasierter Überladung unter Wahrung separater Namensräume*, 16.02.2005

B. Bahnens, *Ein allgemeiner Ansatz zur effizienten Generierung von Datenbanken für perfektes Spiel*, 14.04.2005

S. Fischer, *Functional Logic Programming with Databases*, 19.04.2005

S. Strauß, *Managementunterstützung durch Business Intelligence: Eine kritische Bestandsaufnahme*, 19.04.2005

T. Gerhardts, *Evolutionäre Algorithmen und das gradbeschränkte, minimale Spannbaum-Problem (Studienarbeit)*, 01.03.2005

H. Theman, *Graphische Visualisierung von Java-Kommunikationsabstraktionen (Studienarbeit)*, 10.01.2005

J. Christiansen, *Erweiterung von Web-Fun (Studienarbeit)*, 02.02.2005

◆ Veröffentlichungen

erschienen im Jahre 2005

Elvira Albert, Michael Hanus, Frank Huch, Javier Oliver, German Vidal, *Operational Semantics for Declarative Multi-Paradigm Languages*, *Journal of Symbolic Computation*, **40(1)**, 795 - 829 (2005)

Maria Alpuente, Michael Hanus, Salvador Lucas, German Vidal, *Specialization of Functional Logic Programs Based on Needed Narrowing*, *Theory and Practice of Logic Programming*, **5(3)**, 273 - 303 (2005)

Sergio Antoy, Michael Hanus, *Proceedings of the ACM SIGPLAN 2005 Workshop on Curry and Functional Logic Programming*, ACM Press, (2005)

Sergio Antoy, Michael Hanus, *Declarative Programming with Function Patterns*, *Proceedings of the International Symposium on Logic-based Program Synthesis and Transformation (LOPSTR'05)*, 3 - 18 (2005)

- Sergio Antoy, Michael Hanus, *Overlapping Rules and Logic Variables in Functional Logic Programs*, Technischer Bericht, Univ. Kiel, (2005)
- Sergio Antoy, Michael Hanus, Jimeng Liu, Andrea Tolmach, *A Virtual Machine for Functional Logic Computations*, Proceedings of the 16th International Workshop on Implementation and Application of Functional Languages, IFL 2004, Springer LNCS 3474, 108 - 125 (2005)
- Bernd Braßel, Michael Hanus, *Nondeterminism Analysis of Functional Logic Programs*, Proc. of the 21st International Conference on Logic Programming, ICLP 2005, Springer LNCS 3668, 265 - 279 (2005)
- Bernd Braßel, Michael Hanus, Frank Huch, Josep Silva, German Vidal, *Run-Time Profiling of Functional Logic Programs*, Proceedings of the International Symposium on Logic-based Program Synthesis and Transformation, LOPSTR'04, Springer LNCS 3573, 182 - 197 (2005)
- Robby Findler, Michael Hanus, Simon Thompson, *Proceedings of the ACM SIGPLAN 2005 Workshop on Functional and Declarative Programming in Education*, ACM Press, (2005)
- Sebastian Fischer, *A Functional Logic Database Library*, Proceedings of the ACM SIGPLAN 2005 Workshop on Curry and Functional Logic Programming, WFLP 2005, 54 - 59 (2005)
- Dietmar Seipel, Michael Hanus, Ulrich Geske, Oskar Bartenstein, *Applications of Declarative Programming and Knowledge Management*, Springer Lecture Notes in Artificial Intelligence, **3392**, (2005)
- Michael Hanus, *Functional Logic Programming: From Theory to Curry*, Technischer Bericht, Univ. Kiel, (2005)
- Michael Hanus, *CurryTest: A Tool for Testing Declarative Programs*, Proc. 19th Workshop on (Constraint) Logic Programming, WLP 2005, (2005)
- Michael Hanus, *A Generic Analysis Environment for Declarative Programs*, Proceedings of the ACM SIGPLAN 2005 Workshop on Curry and Functional Logic Programming, 43 - 48 (2005)
- Bernd Braßel, Frank Huch, *Translating Curry to Haskell (System Demo)*, Proceedings of the ACM SIGPLAN 2005 Workshop on Curry and Functional Logic Programming, 60 - 65 (2005)
- Frank Huch, Frank Kupke, *Composable Memory Transactions in Concurrent Haskell*, Proc. of the 17th International Workshop on Implementation and Application of Functional Languages (IFL 2005), (2005)
- Frank Huch, Parissa H. Sadeghi, *The COOSY Debugging Environment – Tool Demo*, Proc. of the 17th International Workshop on Implementation and Application of Functional Languages (IFL 2005), (2005)
- C. Grelck, Frank Huch, Greg J. Michaelson, Phil Grinder, *Proceedings of the 16th International Workshop on Implementation and Application of Functional Languages*, Springer Lecture Notes in Computer Science, **3474**, (2005)
- Frank Huch, *Programmiersprachen und Rechenkonzepte - Proceedings des 22. Workshop der GI-Fachgruppe 2.1.4*, Technischer Bericht, CAU Kiel, **0513**, (2005)
- Bernd Braßel, *Das Programm der idealen Logik*, Königshausen & Neumann, (2005)
- Volker Stolz, Frank Huch, *Runtime Verification of Concurrent Haskell Programs*, Electr. Notes Theor. Comput. Sci., **113**, 201 - 216 (2005)

Präsentationen

- Michael Hanus, *CurryTest: A Tool for Testing Declarative Programs*, 19th Workshop on (Constraint) Logic Programming (WLP 2005), Ulm, 22.02.2005
- Michael Hanus, *Declarative Programming with Persistent Information*, Kolloquium, Oregon State University, Corvallis, Oregon, USA, 16.05.2005
- Michael Hanus, *Declarative Programming with Databases*, Kolloquium, Portland State University, Portland, Oregon, USA, 06.06.2005
- Michael Hanus, *Declarative Programming with Function Patterns*, International Symposium on Logic-based Program Synthesis and Transformation (LOPSTR'05), London, England, 07.09.2005
- Michael Hanus, *A Generic Analysis Environment for Declarative Programs*, ACM SIGPLAN 2005 Workshop on Curry and

- Functional Logic Programming, Tallinn, Estland, 29.09.2005
- Michael Hanus, *Future Logic Programming Languages: Close the Gap between Theory and Practice*, International Conference on Logic Programming (ICLP 2005), Sitges, Spanien, 04.10.2005
- Michael Hanus, *Multiparadigmen-Programmierung*, Kolloquium, Universität Karlsruhe, Karlsruhe, 11.10.2005
- Sebastian Fischer, *A Functional Logic Database Library*, ACM SIGPLAN 2005 Workshop on Curry and Functional Logic Programming, Tallinn, Estland, 29.09.2005
- Sebastian Fischer, *A Functional Logic Database Library*, Kolloquium über Programmiersprachen und Grundlagen der Programming, Fischbachau, 06.10.2005
- Frank Huch, *Searching for deadlocks while debugging Concurrent Haskell programs*, Eingeladener Vortrag, University of Kent, Canterbury, England, 28.02.2005
- Frank Huch, *Searching for deadlocks while debugging Concurrent Haskell programs*, Eingeladener Vortrag, University of Hertfordshire, Hatfield, England, 09.03.2005
- Frank Huch, *Composable Memory Transactions in Concurrent Haskell*, 22. Workshop der GI-Fachgruppe 2.1.4, Bad Honnef, 04.05.2005
- Frank Huch, *Composable Memory Transactions in Concurrent Haskell*, 17th International Workshop on Implementation and Application of Functional Languages, Dublin, Irland, 19.09.2005
- Parissa H. Sadeghi, *The COOSY Debugging Environment – Tool Demo*, 17th International Workshop on Implementation and Application of, Dublin, Irland, 21.09.2005
- Bernd Braßel, *Curry To Haskell - Erste Ideen für einen Compiler*, 22. Workshop der GI-Fachgruppe 2.1.4, Bad Honnef, 03.05.2005
- Bernd Braßel, *Nondeterminism Analysis of Functional Logic Programs*, International Conference on Logic Programming, ICLP 2005, Sitges, Spanien, 02.10.2005
- Bernd Braßel, *Translating Curry To Haskell*, ACM SIGPLAN 2005 Workshop on Curry and Functional Logic Programming, Tallinn, Estland, 29.09.2005
- Bernd Braßel, *Q.E.D. - Beweis durch Selbstbezug*, IV. Graduiertenkonferenz der Gesellschaft für Philosophie und Wissenschaft (GPW), Heidelberg, 08.10.2005

Andere Aktivitäten und Ereignisse

- M. Hanus: Mitglied im Programmkomitee des 19th Workshop on (Constraint) Logic Programming, Ulm, 2005
- M. Hanus: Mitglied im Programmkomitee von PPDP 2005 (7th International ACM SIGPLAN Conference on Principles and Practice of Declarative Programming), Lissabon (Portugal), 2005
- M. Hanus: Mitglied im Programmkomitee von LOPSTR 2005 (15th International Symposium on Logic-based Program Synthesis and Transformation), London (England), 2005
- M. Hanus: Organisation des Workshops "Functional and Declarative Programming in Education 2005" im Rahmen der IFCP 2005 (10th ACM SIGPLAN International Conference on Functional Programming), Tallin, 2005 (zusammen mit Robby Findler, University of Chicago, und Simon Thompson, University of Kent)
- M. Hanus: Vorsitzender des Programmkomitees und Organisation des "Workshop on Curry and Functional Logic Programming" (WCFLP 2005) im Rahmen der IFCP 2005 (10th ACM SIGPLAN International Conference on Functional Programming), Tallin, 2005 (zusammen mit Sergio Antoy, Portland State University)
- M. Hanus: Mitglied im Programmkomitee des 20th Workshop on (Constraint) Logic Programming, Wien, 2006
- M. Hanus: Eingeladener Podiumssprecher zur International Conference on Logic Programming, Sitges (Spanien), 2005
- M. Hanus: Mitglied im Steering Committee der PPDP-Konferenzen (International ACM SIGPLAN Conference on Principles and Practice of Declarative Programming)

M. Hanus: Mitglied im Editorial Board des Journal of Functional and Logic Programming

M. Hanus: Sprecher der Fachgruppe FG 2.1.4 Programmiersprachen und Rechenkonzepte der Gesellschaft für Informatik e.V.

M. Hanus: Mitglied in der DAAD-Auswahlkommission zur projektbezogenen Förderung des Wissenschaftleraustausches mit Spanien und Portugal

M. Hanus: Begutachtung von Projektanträgen für die Deutsche Forschungsgemeinschaft

M. Hanus: Forschungsaufenthalt am Department of Computer Science, Portland State University, U.S.A., März bis August 2005

F. Huch: Stellvertretender Sprecher der Fachgruppe FG 2.1.4 Programmiersprachen und Rechenkonzepte der Gesellschaft für Informatik e.V.

F. Huch: Organisation des 22. Workshop der GI-Fachgruppe 2.1.4 Programmiersprachen und Rechenkonzepte, Bad Honnef, 2. - 4.05.2005

F. Huch: Mitglied im Programmkomitee der 17th International Workshop on Implementation and Application of Functional Languages, Dublin, Irland, 2005

F. Huch: Mitglied im Programmkomitee des Workshop on Curry and Functional Logic Programming (WCFLP 2005), Tallinn, Estland, 2005

F. Huch: Forschungsaufenthalt an der University of Kent, Canterbury

F. Huch: Miniprogrammierkurs mit der Marienkäferdame Kara, Girls´ day der TF, 28.4.2005

F. Huch: Organisation (zusammen mit Thomas Wilke) des Schnupperstudiums Informatik für Schülerinnen, 10. - 14. Oktober 2005, Kiel.

F. Huch: Kurs im Rahmen des Schnupperstudiums Informatik zur Einführung in die Programmierung: Steuerung des Marienkäfers Kara mit Hilfe der Programmiersprache Erlang, Abschlussprojekt: Entwicklung und Implementierung eines verteilten Chats.

F. Simon: Vorträge im Rahmen der Lehrerfortbildung des IQSH

W. Goerigk: Stellvertretender Sprecher der Fachgruppe FG 2.1.4 Programmiersprachen und Rechenkonzepte der Gesellschaft für Informatik e.V.

W. Goerigk: Stellvertretender Sprecher der Fachgruppe FG 1.1.1 Deklarative Sprachen der Gesellschaft für Informatik e.V.