



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



Christian-Albrechts-Universität zu Kiel

almanach 06

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, insbesondere im Bereich der Internet-basierten Systeme. 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 Bereich der Konzeption deklarativer Programmiersprachen wurden in Zusammenarbeit mit der Portland State University konzeptuelle Fragestellungen beim Entwurf deklarativer Programmiersprachen behandelt. Zum einen wurden die Arbeiten für ein neues Konzept zum Pattern Matching (funktionale Muster), die im Jahr 2005 begonnen wurden, abgeschlossen und deren Ausdrucksstärke in verschiedenen Anwendungen (z.B. XML-Programmierung) demonstriert. Zum anderen 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.

Ein wesentlicher Schwerpunkt der Forschungsarbeiten im Berichtsjahr war die Entwicklung von Werkzeugen zur Entwicklung zuverlässiger deklarativer Programme. Hierzu wurde z.B. eine graphische Programmierumgebung (CurryBrowser) weiterentwickelt, mit deren Hilfe der Programmierer verschiedene Programmanalysen durchführen kann. Zum Aufspüren von Fehlern in deklarativen Programmen wurden mehrere Werkzeuge und Techniken entwickelt. Da deklarative Programmiersprachen stärker als herkömmliche Sprachen 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 verschiedene neue Debugging-Methoden entwickelt, die auf die Besonderheiten deklarativer Programmiersprachen zugeschnitten sind. Beim *Debugging durch Beobachten* werden bestimmte Programmobjekte (Daten und auch Funktionen) zur Beobachtung ausgezeichnet und deren Auswertung nach dem Programmablauf visualisiert. Um diesen Debugging-Prozess zu vereinfachen, wurde eine graphische Programmierumgebung entwickelt, mit der der Programmierer einfach neue Beobachtungsobjekte definieren und das Programm bezüglich dieser Beobachtungen ablaufen lassen kann.

Eine weitere Technik zum Auffinden von Fehlern ist das *Tracing*, d.h. die Verfolgung des Berechnungsverlaufes. Da die konkrete Berechnungsreihenfolge schwierig zu überblicken ist (s.o.), wurden Werkzeuge entwickelt, die zunächst die gesamte Programmausführung aufzeichnen, um sie dann geeignet zu visualisieren. Um die Kosten für das Aufzeichnen während der Ausführung in engen Grenzen zu halten, wurde eine neue Programmtransformation entwickelt, mit der Quellprogramme automatisch in effizient ausführbare Programme überführt werden können, die sich wie die Originalprogramme verhalten, aber gleichzeitig deren Ausführung protokollieren.

Zusicherungen (Assertions) sind eine weitere Alternative, um Fehler in Programmen zu finden. Zusicherungen sind Eigenschaften, die an bestimmten Programmstellen immer erfüllt sein sollen, andernfalls ist das Programm fehlerhaft, was entsprechend angezeigt werden soll. Somit sollen Zusicherungen bei der Programmausführung überprüft werden,

allerdings soll sich dabei die ursprüngliche Ausführungsreihenfolge nicht ändern. Dies ist gerade bei nicht-strikten Sprachen mit einer flexiblen Abarbeitungsstrategie ein nicht-triviales Problem. Aus diesem Grund wurde in Zusammenarbeit mit der University of Kent (England) eine spezielle Logik mit nicht-strikten Zusicherungen entwickelt.

Weitere Arbeiten erfolgten im Bereich der praktischen Anwendungen deklarativer Programmierkonzepte. So wurde das bisher entwickelte Konzept zur Verwaltung persistenter Daten in deklarativen Programmen mit Hilfe einer relationalen Datenbankimplementierung und einer geeigneten Schnittstelle praktisch umgesetzt. Im Bereich der Internet-Programmierung wurde ein neues Konzept zum verteilten Zugriff und Manipulation von Daten mit standardisierten Internet-Protokollen entwickelt. Zur effizienten Programmierung Web-basierter Benutzerschnittstellen zur Manipulation komplexer Daten wurde ein neues Konzept entwickelt, mit dessen Hilfe die Benutzerschnittstellen direkt aus der Typstruktur der Daten erzeugt werden können. Diese Technik erlaubt eine bisher nicht erreichte Kompaktheit von Programmen, die die typkorrekte Manipulation von Daten mit Standardbrowsern garantieren und dadurch die Anwendungslogik von der Schnittstellenlogik trennen. Dieses Konzept wurde in der Sprache Curry umgesetzt. Als Demonstrationsbeispiel wurde ein Programm zur Lösung von Sudoku-Rätseln mit einer Web-basierten Eingabeschnittstelle entwickelt, was aus nur 20 Codezeilen besteht, wobei die Hälfte davon den Sudoku-Löser implementiert und die andere Hälfte die Web-basierte Ein-/Ausgabeschnittstelle realisiert. Eine ähnliche Kompaktheit ist mit nicht-deklarativen Programmiersprachen kaum erreichbar.

Die Programmierung nebenläufiger und verteilter Systeme gestaltet sich in der Praxis oft schwierig, da Deadlocks und Livelocks verhindert werden sollten. Neue Kommunikationsmechanismen können den Programmierer hierbei sehr gut unterstützen. Für Concurrent Haskell wurde 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. Als Erweiterung verwenden wir diesen Ansatz nun auch als Kommunikationsmechanismus für verteilte Systeme, wodurch sich der Ansatz homogen auf die Programmierung verteilter Systeme fortsetzt.

Im Bereich der Öffentlichkeitsarbeit beteiligte sich der Lehrstuhl an der Ringvorlesung des Instituts im Rahmen des Informatikjahres 2006 und an der Durchführung des Girls' Day am 27. April 2006. An diesem Tag wurde mit den Schülerinnen ein kleiner Programmierkurs durchgeführt, bei dem einfache grundlegende Programmierkonzepte am Beispiel der Programmierung des Marienkäfers Kara erlernt wurden. 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.

Personal

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.-31.12.2006	DFG
Fehlersuche in deklarativen Programmen		
Dipl.-Inf. Sebastian Fischer	01.01.-31.12.2006	CAU
Dr. Frank Huch	01.01.-31.12.2006	CAU
Dr. Friedemann Simon	01.01.-31.12.2006	CAU

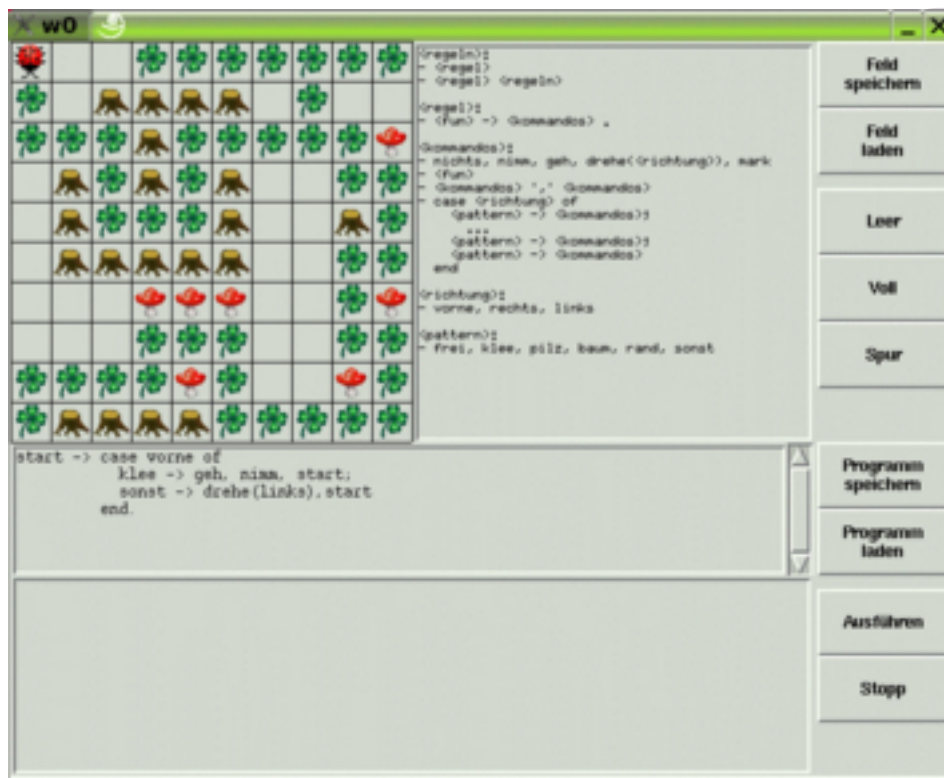


Abb. 1: Spielerisch programmieren lernen mit dem Marienkäfer Kara, Girls' Day/Schnupperstudium Informatik für Schülerinnen

Vorlesungen, Seminare und Praktika

Winter 2005/2006

Diplomandenseminar, 2 Std. Seminar/Woche,
Michael Hanus

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 (+ Britta Kehden, Sebastian Fischer, 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

Informatik, Logik und Mathematik, 2 Std. Arbeitsgemeinschaft/Woche,
Michael Hanus (+ Rudolf Berghammer)

Sommer 2006

Diplomandenseminar, 2 Std. Seminar/Woche,
Michael Hanus



Abb. 2: Teilnehmerinnen am Schnupperstudium Informatik 2006

Logikprogrammierung, 4 (+ 2) Std. Vorlesung (+ Übungen)/Woche,
Michael Hanus (+ Sebastian Fischer)

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

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

Webbasierte Systeme, 4 Std. Seminar/Woche,
Frank Huch (+ Sebastian Fischer)

Informatik, Logik und Mathematik, 2 Std. Arbeitsgemeinschaft/Woche,
Michael Hanus (+ Rudolf Berghammer)

Winter 2006/2007

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

Diplomandenseminar, 2 Std. Seminar/Woche,
Michael Hanus

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

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

Programmierwerkzeuge, 2 Std. Seminar/Woche,

Michael Hanus

Informatik, Logik und Mathematik, 2 Std. Arbeitsgemeinschaft/Woche,
Michael Hanus (+ Rudolf Berghammer)

▀ Drittmittel

DFG, *Systematische Fehlersuche in deklarativen Programmen*, 01.10.2004-28.02.2007 (127087 EUR)
DFG, *Teilnahme ICLP 2006 und Exkursion zur Portland State University*, 15.-27.08.2006 (2215 EUR)

▀ Weitere Zusammenarbeiten, Technologietransfers und Konsultationen

Während des Berichtszeitraumes erfolgten Kooperationen mit:

University of Kent (Olaf Chitil)

Technische Universität Valencia (German Vidal)

Portland State University (Sergio Antoy)

Westfälische Wilhelms-Universität Münster (Herbert Kuchen)

▀ Diplom- und Master-Arbeiten

Aneta Neumann, *X3D als Fortentwicklung von VRML - eine Bestandsaufnahme und Fallstudien*, 09.01.2006
Jan Christiansen, *A purely functional implementation of ROBDDs in Haskell*, 09.02.2006
Mirko Sarach, *Entwurf und Implementierung einer JMX-Management-Konsole*, 11.02.2006
Lennart-C. Ehlers, *SOA-Anwendungen zur technischen Unterstützung von Basel II*, 18.04.2006
Antje Nageler, *Integration von sicherheitsrelevanten Informationen in ein Identitätsmanagementsystem*, 22.05.2006
Ingmar Puschnig, *Anpassung und Implementierung des ITIL Configuration Management-Prozesses*, 31.07.2006
Björn Teegen, *Web-Services in Haskell*, 22.09.2006
Melanie Bargstädt, *Implementierung von Lazy Assertions mit Template Haskell*, 15.09.2006

▀ Dissertationen / Habilitationen

Frank Huch, *Debugging Lazy Declarative Programs (Habilitationsschrift)*, 29.11.2006

▀ Veröffentlichungen

erschieden im Jahre 2006

Sebastian Fischer, *Resource-Based Web Applications*, Trends in Functional Programming, **7**, (2006)
Bernd Braßel, Sebastian Fischer, Frank Huch, *A Program Transformation for Tracing Functional Logic Computations*,
International Symposium on Logic-based Program Synthesis and Transformation (LOPSTR 2006), Springer LNCS
4407, 160 - 176 (2006)
Sebastian Fischer, *Lazy Database Access with Persistent Predicates*, Proc. of the 15th International Workshop on
Functional and (Constraint) Logic Programming (WFLP 2006), 167 - 181 (2006)
Rudolf Berghammer, Sebastian Fischer, *Combining Relation Algebra with Functional and Constraint Logic Programming*,
Proc. of the 15th International Workshop on Functional and (Constraint) Logic Programming (WFLP 2006), 153 -
166 (2006)
Michael Hanus, *Reporting Failures in Functional Logic Programs*, Proc. of the 15th International Workshop on Functional
and (Constraint) Logic Programming (WFLP 2006), 49 - 62 (2006)

- Sergio Antoy, Michael Hanus, *Declarative Programming with Function Patterns*, Proceedings of the 15th International Symposium on Logic-based Program Synthesis and Transformation, **Springer LNCS 3901**, 6 - 22 (2006)
- Michael Hanus, *Type-Oriented Construction of Web User Interfaces*, Proc. of the 8th International ACM SIGPLAN Conference on Principles and Practice of Declarative Programming, 27 - 38 (2006)
- Sergio Antoy, Michael Hanus, *Overlapping Rules and Logic Variables in Functional Logic Programs*, Proc. of the 22nd International Conference on Logic Programming (ICLP 2006), **Springer LNCS 4079**, 87 - 101 (2006)
- Michael Hanus, *Adding Constraint Handling Rules to Curry*, Proc. 20th Workshop on Logic Programming, WLP 2006, 81 - 90 (2006)
- Michael Hanus, *Type-Oriented Construction of Web User Interfaces*, Proc. 21. Workshop GI-Fachgruppe Programmiersprachen und Rechenkonzepte, Softwaretechnik-Trends, **26(2)**, (2006)
- Michael Hanus, *CurryBrowser: A Generic Analysis Environment for Curry Programs*, Proc. of the 16th Workshop on Logic-based Methods in Programming Environments (WLPE 2006), 61 - 74 (2006)
- Sergio Antoy, Michael Hanus, *Overlapping Rules and Logic Variables in Functional Logic Programs*, Forschungsbericht, **0608**, (2006)
- Frank Huch, Frank Kupke, *Composable Memory Transactions in Concurrent Haskell*, Proc. of the 17th International Workshop on Implementation and Application of Functional Languages, **Springer LNCS 4015**, 124 - 141 (2006)
- Olaf Chitil, Frank Huch, *A Pattern Logic for Prompt Lazy Assertions in Haskell*, Proceedings of the 18th International Symposium on Implementation and Application of Functional Languages (IFL 2006), (2006)
- Parissa Sadeghi, Frank Huch, *The Interactive Curry Observation Debugger COOISY*, Proc. of the 15th International Workshop on Functional and (Constraint) Logic Programming (WFLP 2006), 93 - 107 (2006)
- Jan Christiansen, Frank Huch, *A Purely Functional Implementation of ROBDDs in Haskell*, Proceedings of the Seventh Symposium on Trends in Functional Programming (TFP 2006), **4**, (2006)
- Andrew Butterfield, Clemens Greck, Frank Huch, *Proceedings of the 17th International Workshop on Implementation and Application of Functional Languages*, Springer LNCS, **4015**, (2006)
- Bernd Braßel, *A Framework for Interpreting Traces of Functional Logic Computations*, Proc. of the 15th International Workshop on Functional and (Constraint) Logic Programming (WFLP 2006), 77 - 91 (2006)
- Bernd Braßel, Sebastian Fischer, Frank Huch, *A Program Transformation for Tracing Functional Logic Computations*, Pre-Proceedings of the International Symposium on Logic-based Program Synthesis and Transformation (LOPSTR'06), **TR CS-2006-5, Univ. of Venice**, 141 - 157 (2006)



Präsentationen

- Sebastian Fischer, *Resource-Based Web Applications*, Seventh Symposium on Trends in Functional Programming (TFP 2006), Nottingham, Großbritannien, 20.04.2006
- Sebastian Fischer, *Resource-Based Web Applications*, 23. Workshop Programmiersprachen und Rechenkonzepte der GI-Fachgruppe 2.1.4, Bad Honnef, 05.05.2006
- Sebastian Fischer, *Lazy Database Access with Persistent Predicates*, 15th Workshop on Functional and (Constraint) Logic Programming (WFLP 2006), Madrid, Spanien, 16.11.2006
- Michael Hanus, *Adding Constraint Handling Rules to Curry*, 20th Workshop on Logic Programming, Wien, 22.02.2006
- Michael Hanus, *Type-Oriented Construction of Web User Interfaces*, 23. Workshop Programmiersprachen und Rechenkonzepte der GI-Fachgruppe 2.1.4, Bad Honnef, 05.05.2006
- Michael Hanus, *Type-Oriented Construction of Web User Interfaces*, 8th International ACM SIGPLAN Conference on Principles and Practice of Declarative Programming (PPDP 2006), Venedig, Italien, 10.07.2006
- Michael Hanus, *CurryBrowser: A Generic Analysis Environment for Curry Programs*, 16th Workshop on Logic-based Methods in Programming Environments (WLPE'06), Seattle, USA, 16.08.2006
- Michael Hanus, *Overlapping Rules and Logic Variables in Functional Logic Programs*, 22nd International Conference on Logic Programming (ICLP 2006), Seattle, USA, 17.08.2006

- Michael Hanus, *Reporting Failures in Functional Logic Programs*, 15th International Workshop on Functional and (Constraint) Logic Programming (WFLP 2006), Madrid, Spanien, 16.11.2006
- Michael Hanus, *Von Mozart zur Informatik – Wie programmiert man Musik?*, Ringvorlesung, Kiel, 17.05.2006
- Parissa Sadeghi, *Interactive Curry Observation Debugger*, 15th Workshop on Functional and (Constraint) Logic Programming (WFLP 2006), Madrid, Spanien, 16.11.2006
- Frank Kupke, *Haskell Transaktionen in verteilten Systemen*, 23. Workshop Programmiersprachen und Rechenkonzepte der GI-Fachgruppe 2.1.4, Bad Honnef, 03.05.2006
- Frank Huch, *A Program Transformation for Tracing Functional-Logic Computations*, International Symposium on Logic-based Program Synthesis and Transformation, Venedig, Italien, 13.07.2006
- Frank Huch, *A Pattern Logic for Prompt Lazy Assertions in Haskell*, 18th International Symposium on Implementation and Application of Functional Languages, Budapest, Ungarn, 04.09.2006
- Frank Huch, *Transaktionsbasierte Kommunikation für nebenläufige Systeme*, Habilitationsvortrag, Kiel, 29.11.2006
- Frank Huch, *Dynamische Deadlock-Suche in nebenläufigen funktionalen Programmen*, 23. Workshop Programmiersprachen und Rechenkonzepte der GI-Fachgruppe 2.1.4, Bad Honnef, 04.05.2006
- Bernd Braßel, *Searching Bugs by Visualizing Computations*, 23. Workshop Programmiersprachen und Rechenkonzepte der GI-Fachgruppe 2.1.4, Bad Honnef, 05.05.2006
- Bernd Braßel, *A Framework for Interpreting Traces of Functional Logic Computations*, 15th Workshop on Functional and (Constraint) Logic Programming (WFLP 2006), Madrid, Spanien, 16.11.2006
- Bernd Braßel, *Freiheit und Notwendigkeit in der logischen Entwicklung*, Kolloquium des Arbeitskreises Hegels Naturphilosophie - Die Freiheit des Willens, Lübeck, 24.06.2006
- Jan Christiansen, *A Purely Functional Implementation of ROBDDs in Haskell*, Seventh Symposium on Trends in Functional Programming (TFP 2006), Nottingham, Großbritannien, 20.04.2006

Andere Aktivitäten und Ereignisse

- S. Fischer: Forschungsaufenthalt bei Prof. Herbert Kuchen, Westfälische Wilhelms-Universität Münster (27.8.-3.9.2006)
- M. Hanus: Mitglied im Programmkomitee des 20th Workshop on (Constraint) Logic Programming, Wien, 2006
- M. Hanus: Organisation des 23. Workshop der GI-Fachgruppe Programmiersprachen und Rechenkonzepte, Bad Honnef, Mai 2006
- M. Hanus: Mitglied im Programmkomitee des 20th Workshop on Logic Programming, Wien, 2006
- M. Hanus: Mitglied im Programmkomitee von LOPSTR 2006 (16th International Symposium on Logic-based Program Synthesis and Transformation), Venedig (Italien), 2006
- M. Hanus: Mitglied im Programmkomitee von WFLP 2006 (14th International Workshop on Functional and (Constraint) Logic Programming), Madrid, 2006
- M. Hanus: Vorsitzender des Programmkomitees der 9th International Symposium on Practical Applications of Declarative Languages, Nizza, 2007
- 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 „Programmiersprachen und Rechenkonzepte“ der Gesellschaft für Informatik e.V.
- M. Hanus: Vorstandsmitglied und stellvertretender Sprecher der GLP (Gesellschaft für Logische Programmierung)
- M. Hanus: Mitglied im Fachbeirat der GLP (Gesellschaft für Logische Programmierung)

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

M. Hanus: Begutachtung von Projektanträgen für die Deutsche Forschungsgemeinschaft

M. Hanus: Mitglied im wissenschaftlichen Beirat der Berufsakademie an der Wirtschaftsakademie Schleswig-Holstein

M. Hanus: Forschungsaufenthalt am Department of Computer Science, Portland State University, U.S.A., August 2006

M. Hanus: Vorträge im Rahmen der Lehrerfortbildung des IQSH

M. Hanus, F. Huch: Programmkomitee-Chairs der 4. Arbeitstagung Programmiersprachen (ATPS'06)

F. Huch: Mitglied im Programmkomitee von WFLP 2006 (14th International Workshop on Functional and (Constraint) Logic Programming), Madrid, 2006

F. Huch: Mitglied im Steering Committee der Symposia on Implementation and Application of Functional Languages (IFL)

F. Huch: Miniprogrammierkurs mit Kara, Girls' Day der TF, 27.4.2006

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

F. Huch (mit Th. Wilke): Organisation des Schnupperstudiums Informatik für Schülerinnen (16.-20.10.2005, Kiel). Darin Durchführung eines Kurses zur Einführung in die Programmierung (Steuerung des Marienkäfers Kara mit Hilfe der Programmiersprache Erlang) und Abschlussprojekt: Entwicklung und Implementierung eines verteilten Chats.

F. Simon: Vorträge im Rahmen der Lehrerfortbildung des IQSH

F. Simon: Referent auf Berufsinformationsseminaren für Schülerinnen und Schüler der gymnasialen Oberstufe