

Programming Languages and Compiler Construction

The research group "Programming Languages and Compiler Construction" is interested in the design, implementation, and application of programming languages intended to support the reliable implementation of complex systems. The research ranges from object-oriented design methods and the analysis of concurrent and distributed systems, to the implementation and application of declarative programming languages, in particular, in the area of web-based systems.

During the period reported below, the research group worked on a new advanced implementation of the multi-paradigm language Curry and achieved new research results related to the design, implementation, analysis, and application of declarative programming methods.

Results

The scientific work of the research group involved all areas related to declarative programming languages, e.g. design, semantics, implementation, development tools, and the application of such languages. Declarative programming languages are based on clear mathematical foundations. They abstract from the underlying computer architecture and thus provide a higher programming level leading to more reliable systems. In particular, much of the research is focused on the integration of the most important declarative programming paradigms: functional and logic programming.



Fig. 1: The programming language and program development groups on a climbing trip

In relation to the *application* of declarative languages, we continued our work on the design and implementation of the web framework *Spicey*. *Spicey* enables the generation in a few minutes of complete web applications from a specification of the underlying complex data model. Since the generated implementation is a high-level declarative program, it is easy to adapt this program to various customer requirements. In contrast to other web frameworks, our framework exploits high-level declarative programming techniques so that it yields reliable implementations that avoid data inconsistencies at various levels. We applied this system to implement a web-based interface to access the module database of the Institute of Computer Science, which is used to plan the curriculum; it was quite positively evaluated by an accreditation committee.

To explore the integration of declarative and object-oriented programming techniques, we developed a new web-based survey system that combines surveys, quizzes, and self-tests. The flexible creation of such applications is enabled by a domain-specific language to describe surveys and tests. The complete system is implemented in the programming language *Scala*, which combines functional and object-oriented features and exploits the web framework *Play*. The developed survey

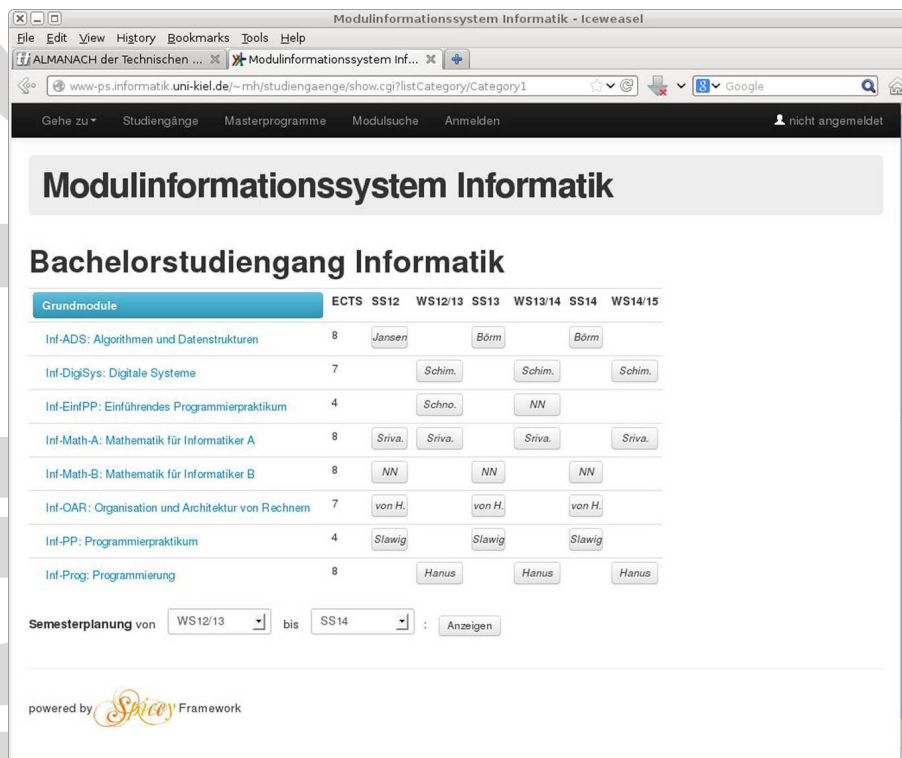


Fig. 2: The interface to the module database of the Institute of Computer Science

system, called SuSy, is freely available.

In the area of *software technique* related to declarative languages, we collaborated with the Portland State University (Oregon, USA) and developed a transformation tool to support the development of reliable declarative programs written in the multi-paradigm programming language Curry. First, we set up a framework with precise notions of specifications, contracts, and assertions for declarative programs. Since specifications, contracts, assertions, and programs are written in the same language (Curry), it is used as a wide-spectrum language for software development. For instance, specifications are written in Curry and are thus executable. Hence, they can be used as an initial implementation. If this implementation is not efficient enough, one can develop a more efficient one, e.g. by using specific data structures and algorithms for the given problem domain. In this case one can use the initial specification as a contract or assertion to check the validity of the new implementation. This general idea is supported by a new tool (DSDCurry), which transforms, if necessary, specifications into implementations or assertions. Based on other work of our group on different methods for assertion checking, DSDCurry also supports various kinds of assertion checks, like strict, lazy, or faithful assertions.

We also investigated several issues related to the *implementation* of functional logic programming languages. In particular, we continued our development of a new implementation of Curry, called KiCS2, that is based on compiling Curry programs into purely functional Haskell programs. The advantage of this implementation is the explicit representation of non-deterministic computed results in a single data structure. This provides for the application of various search strategies (depth-first, breadth-first, iterative deepening, parallel) to explore the search space. Since these opportunities are unique to KiCS2, we investigated the comparison of different search strategies in a single system. In most other systems, an incomplete depth-first search strategy is used for efficiency reasons. We showed that the preference of an incomplete strategy is not really necessary, since complete search strategies like breadth-first or iterative deepening are applicable in practice with a small overhead only. Based on the positive results, we started the distribution of our new implementation, which is available as open-source software.

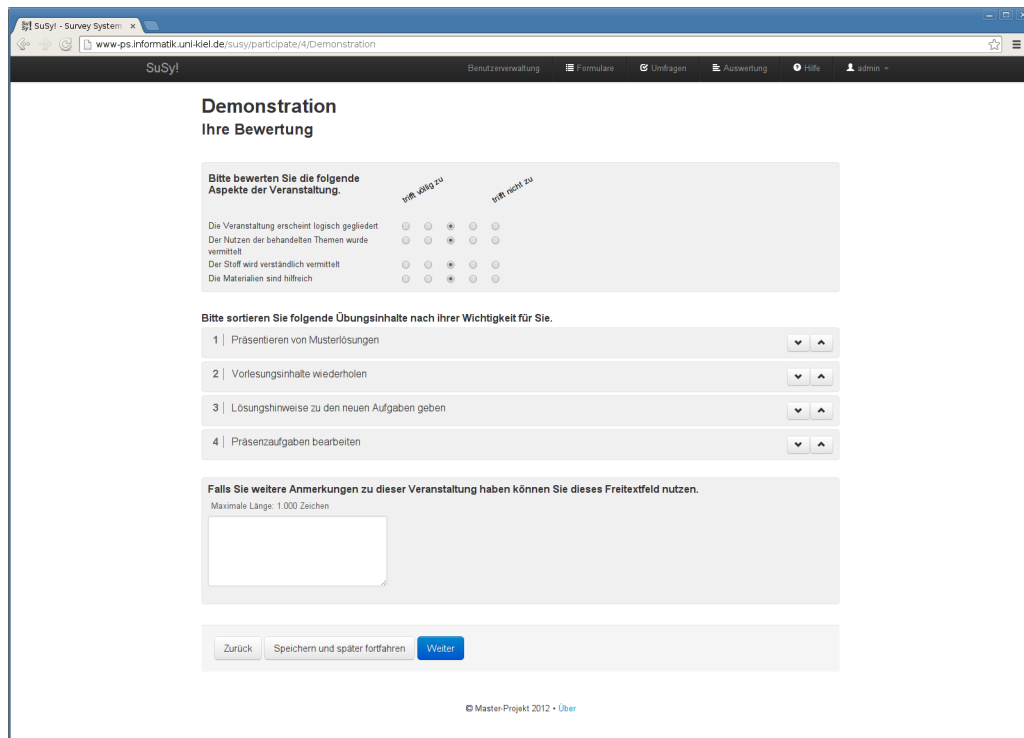


Fig. 3: A screenshot of the survey system SuSy

The practical use of our new implementation in various implementation projects (e.g. in the module database discussed above) also showed some performance problems in specific situations. In particular, the combination of non-deterministic evaluations that are lazily passed to other operations could cause a duplication of the work to traverse the underlying data structures. In order to avoid such problems, we developed a new program analysis to identify such points in a program together with a program transformation to optimize it.

In the area of the *design and semantics* of declarative languages, we had developed, in a previous collaboration with the Portland State University (Oregon, USA), a new technique to encapsulate non-deterministic computations in functional logic programs. It is based on associating to each function a set-valued function encapsulating the non-determinism caused by the function's execution. It is the first referentially transparent approach to encapsulate non-deterministic computations and thus solves a long-standing problem in this area. Due to the advanced implementation techniques used in KiCS2, we were also able to implement set functions in this system. Thus, KiCS2 is the first implementation with a full support for set functions. During these implementation efforts, it turned out that the original proposal for set functions is underspecified so that various semantic options (empty result sets, nested set functions) are unclear. Therefore, we developed a new abstract description of all these details and obtained the first denotational description of set functions. This description is not only of theoretical interest, but is also the basis of our new implementation of set functions.

Finally, we finished a comprehensive survey on functional logic programming that might be used as a standard reference in this area in future research and teaching activities.

Personnel

Head of the group: Prof. Dr. Michael Hanus; Secretary: Jane-Maria Eitzen (50%)
 Technical Staff: Dipl.-Ing. (FH) Thomas Heß (50%)

Scientific Staff:

Dr. Sebastian Fischer	01.01.-31.12.2012	Guest
PD. Dr. Frank Huch	01.01.-31.12.2012	CAU (35%)
M. SC. Björn Peemöller	01.01.-31.12.2012	CAU
Dipl.-Inf. Lars Prädell	01.10.-31.12.2012	CAU
Dipl.-Inf. Fabian Reck	01.01.-31.12.2012	CAU
Dr. Friedemann Simon	01.01.-31.12.2012	CAU

Lectures, Seminars, and Laboratory Course Offers

Winter 2011/2012

Arbeitsgemeinschaft Informatik, Logik und Mathematik, 2 hrs Seminar/Week,
M. Hanus (+ R. Berghammer)

MSS0303: Masterabschlusssseminar - Programmiersprachen, 2 hrs Seminar/Week,
M. Hanus

Inf-Prog: Programmierung, 4 (+ 2) hrs Lecture (+ Exercises)/Week,
M. Hanus (+ F. Reck, L. Prädell)

MS0302: Übersetzerbau, 4 (+ 2) hrs Lecture (+ Exercises)/Week,
M. Hanus (+ F. Reck)

MS0304: Funktionale Programmierung, 4 (+ 2) hrs Lecture (+ Exercises)/Week,
F. Huch (+ B. Peemöller)

Inf-PtKI: Programmieretechniken für die Künstliche Intelligenz, 2 (+ 2) hrs Lecture (+ Exercises)/Week,
F. Simon

NF-Inf-3: Programmieretechniken für die Künstliche Intelligenz für Nebenfächler, 2 (+ 2) hrs Lecture (+ Exercises)/Week,
F. Simon

Summer 2012

Arbeitsgemeinschaft Informatik, Logik und Mathematik, 2 hrs Seminar/Week,
M. Hanus (+ R. Berghammer)

MSS0303: Masterabschlusssseminar - Programmiersprachen, 2 hrs Seminar/Week,
M. Hanus

Inf-FortProg: Fortgeschrittene Programmierung, 3 (+ 2) hrs Lecture (+ Exercises)/Week,
M. Hanus (+ B. Peemöller)

MS0301: Prinzipien von Programmiersprachen, 4 (+ 2) hrs Lecture (+ Exercises)/Week,
M. Hanus (+ F. Reck)

Inf-MP-PS: Mastermodul - Programmiersprachen und Programmiersysteme, 6 hrs Practical/Week,
M. Hanus (+ B. Peemöller)

Inf-PM-PS: Projektmodul - Programmiersprachen und Programmiersysteme, 6 hrs Practical/Week,
M. Hanus (+ B. Peemöller)

Winter 2012/2013

Arbeitsgemeinschaft Informatik, Logik und Mathematik, 2 hrs Seminar/Week,
M. Hanus (+ R. Berghammer)

MSS0303: Masterabschlusseminar - Programmiersprachen, 2 hrs Seminar/Week,
M. Hanus

Inf-MS-Sem-PS: Programmiersprachen und Programmiersysteme, 2 hrs Seminar/Week,
M. Hanus (+ F. Huch)

Inf-Prog: Programmierung, 4 (+ 2) hrs Lecture (+ Exercises)/Week,
M. Hanus (+ F. Reck, B. Peemöller, L. Prädell, N. Danilenko, D. Ehlers, P. Munstermann)

MS0303: Deklarative Programmiersprachen, 4 (+ 2) hrs Lecture (+ Exercises)/Week,
M. Hanus (+ B. Peemöller)

MS0306: Nebenläufige und verteilte Programmierung, 4 (+ 2) hrs Lecture (+ Exercises)/Week,
F. Huch (+ F. Reck)

Inf-Sem-PS: Programmierung und Programmiersprachen, 2 hrs Seminar/Week,
M. Hanus

Further Cooperation, Consulting, and Technology Transfer

During the reported period, the research group collaborated with:

Sergio Antoy (Portland State University, Oregon, USA)

Jan Christiansen (Kiel)

Sebastian Fischer (Kiel)

Daniel Seidel (Universität Bonn)

Diploma, Bachelor's and Master's Theses

Ercan Küçükcaraca, *Requirements Traceability in Model Based Systems Engineering*, 07.03.2012

Lutz Seemann, *Entwicklung einer formalen Beschreibungssprache für Algorithmen im medizinischen Kontext*,
22.03.2012

Lutz Häußler, *Melodiesuche in digitaler Notenschrift - Entwicklung einer in Haskell eingebetteten Sprache zur
musterbasierten Suche in Musik als temporales Medium*, 30.03.2012

Christoph Wulf, *Typsichere Einbettung von Datenbankanfragen in Scala*, 30.03.2012

Jan Rasmus Tikovsky, *Integration eines Finite-Domain-Constraint-Solvers in KiCS2*, 02.08.2012

Sandra Dylus, *An API Search Engine for Curry*, 13.09.2012

Stefan Exner, *Ein universeller Anfrageeditor für Ruby on Rails Datenmodelle*, 24.09.2012

Jonas Oberschweiber, *Type Inference for a Declarative Intermediate Language*, 25.09.2012

Arne Kowalewski, *Entwicklung und Vergleich von maschinellen Spielern für Gomoku, Gobang und Pente*, 27.09.2012

Christian Wischmann, *Erweiterung eines Web-basierten Umfragesystems um eine XML-Schnittstelle und um
XSL-Transformationen*, 27.09.2012

Jens-Uwe Bahr, *A computer player for billiards based on artificial intelligence techniques*, 28.09.2012

Timo von Holtz, *Integration of HaskellDB in Hascat*, 28.09.2012

Sebastian Schäfer, *A Compiler for Functional Languages on GPUs*, 07.11.2012

Marian Palkus, *An Eclipse-Based Integrated Development Environment for Curry*, 04.12.2012

Heiko Hoffmann, *Eine verteilte generische Analyseumgebung für deklarative Programme*, 04.12.2012
 Maïke K. Hargens, *Erweiterung des Ruby-on-Rails Testframeworks um randomisierte Ausführungen*, 04.12.2012
 Christian Kitzlinger, *Entwicklung einer Layoutbeschreibungssprache für Smartphone Anwendungen*, 15.12.2012



Publications

Published in 2012

- Sergio Antoy, Michael Hanus, *Contracts and Specifications for Functional Logic Programming*, Proc. of the 14th International Symposium on Practical Aspects of Declarative Languages, PADL 2012, **Springer LNCS 7149**, 33 - 47 (2012)
- Sven Efftinge, Moritz Eysholdt, Jan Köhnlein, Sebastian Zarnekow, Robert von Massow, Wilhelm Hasselbring, Michael Hanus, *Xbase: Implementing Domain-Specific Languages for Java*, Proc. of the 11th International Conference on Generative Programming and Component Engineering, GPCE'12, 112 - 121 (2012)
- Sebastian Fischer, Michael Hanus, Yuki-yoshi Kameyama, Chung-chie Shan, Naoki Takashima, *Lazy Delimited Nondeterminism*, Proceedings of the 21st International Workshop on Functional and (Constraint) Logic Programming, WFLP 2012, (2012)
- Michael Hanus, *Improving Lazy Non-Deterministic Computations by Demand Analysis*, IfI, CAU Kiel, **Research Report 1209**, (2012)
- Michael Hanus, *Improving Lazy Non-Deterministic Computations by Demand Analysis*, Technical Communications of the 28th International Conference on Logic Programming (ICLP 2012), **LIPICs Vol. 17**, 130 - 143 (2012)
- Michael Hanus, *Functional Logic Programming: From Theory to Curry*, Ganzinger Festschrift, **Springer LNCS 7797**, 123 - 168 (2012)
- Michael Hanus, Sven Koschnicke, *An ER-based Framework for Declarative Web Programming*, Theory and Practice of Logic Programming, (2012)
- Michael Hanus, Björn Peemöller, Fabian Reck, *Search Strategies for Functional Logic Programming*, Proceedings of the 5th Working Conference on Programming Languages (ATPS 2012), **Springer LNI 199**, 61 - 74 (2012)
- Jan Christiansen, Michael Hanus, Fabian Reck, Daniel Seidel, *A Denotational Semantics for Weak Encapsulation in Curry*, Proceedings of the 26th Workshop on Logic Programming, WLP 2012, 139 - 153 (2012)



Presentations

- Michael Hanus, *Contracts and Specifications for Functional Logic Programming*, 14th International Symposium on Practical Aspects of Declarative Languages, Philadelphia, USA, 23.01.2012
- Fabian Reck, *Search Strategies for Functional Logic Programming*, 5. Arbeitstagung Programmiersprachen (ATPS 2012), Berlin, 27.02.2013
- Björn Peemöller, *Specialising Functional Patterns*, 29. Workshop GI-FG Programmiersprachen und Rechenkonzepte, 02.05.2012
- Michael Hanus, *Improving Lazy Non-Deterministic Computations by Demand Analysis*, 29. Workshop GI-FG Programmiersprachen und Rechenkonzepte, 04.05.2012
- Fabian Reck, *Negation as failure with Set Functions*, 29. Workshop GI-FG Programmiersprachen und Rechenkonzepte, Bad Honnef, 04.05.2012
- Michael Hanus, *Improving Lazy Non-Deterministic Computations by Demand Analysis*, 28th International Conference on Logic Programming, Budapest, Ungarn, 06.09.2012
- Fabian Reck, *A Denotational Semantics for Weak Encapsulation in Curry*, 26th Workshop on Logic Programming, Bonn, 25.09.2012

Further Activities and Events

M. Hanus: Programme committee member of the Fifth Working Conference on Programming Languages (ATPS 2012), Berlin, February 2012 (part of the conference Software Engineering 2012)

M. Hanus: Programme committee member of WFLP 2012 (21st International Workshop on Functional and (Constraint) Logic Programming), Nagoya (Japan), May 2012

M. Hanus: Programme committee member of WLP 2012 (26th Workshop on (Constraint) Logic Programming), Bonn, September 2012

M. Hanus: Member of the Editorial Board of the Journal of Functional and Logic Programming

M. Hanus: Member of the steering committee of the Symposia on Logic-based Program Synthesis and Transformation

M. Hanus: Chair of the steering committee of the ACM SIGPLAN Symposia on Principles and Practice of Declarative Programming

M. Hanus: Member of the executive committee and vice-chair of the GLP (Gesellschaft für Logische Programmierung), German-speaking branch of the Association for Logic Programming (ALP)

M. Hanus: Member of the advisory board of the GLP (Gesellschaft für Logische Programmierung), German-speaking branch of the Association for Logic Programming (ALP)

M. Hanus: Member of the selection committee of the DAAD (German Academic Exchange Service) for the project-related support to scientific cooperation with Spain and Portugal

M. Hanus: Member of the advisory board of the "Berufsakademie an der Wirtschaftsakademie Schleswig-Holstein"

M. Hanus: Member of the executive board of the "Fakultätentag Informatik der Bundesrepublik Deutschland"

M. Hanus: Chair of the selection committee of the award for the best diploma or master thesis in computer science in Germany 2012

M. Hanus: Member of the selection committee of the b+m-Preis Software- und Systems-Engineering 2012

M. Hanus: Chair of the examinations board of computer science studies, University of Kiel

M. Hanus: Member of the Senate Curriculum Committee, University of Kiel

M. Hanus: Vice-member of the Senate Equal Opportunities Committee, University of Kiel

F. Huch: Organization of the 29th Workshop of the GI-Fachgruppe Programmiersprachen und Rechenkonzepte, Bad Honnef (Germany), May 2012

F. Huch: Member of the Steering CommitteeCommittee of Symposia on Implementation and Application of Functional Languages (IFL)

F. Huch: Chair of the executive committee of the Fachgruppe "Programmiersprachen und Rechenkonzepte" of the Gesellschaft für Informatik e.V.

F. Simon: Participation in seminars for students planning professional careers

F. Simon: "Computer Museum", representative of the Faculty of Engineering in the board of control