



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



Christian-Albrechts-Universität zu Kiel

almanach 14

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-orientated design methods and the analysis of concurrent and distributed systems to the implementation and application of declarative programming languages, particularly 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, semantics, implementation, and analysis 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 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 (constraint) logic programming. Because of our well-known activities in the area of multi-paradigm languages, we were invited to contribute a chapter on this topic in the new edition of the Computing Handbook that was published in 2014.

In order to evaluate the concepts of declarative programming in practice, we worked on the *application and implementation* of the functional logic programming language Curry. In particular, we continued the development of our recent implementation of Curry, called KiCS2, which is based on compiling Curry programs into purely functional Haskell programs. The advantages of this implementation are the good performance of purely functional computations and the explicit representation of non-deterministic computed results in a single data structure that supports a variety of search strategies. Due to the compilation into a purely functional target language, the implementation of features for constraint programming, like constraints over finite domains that have important applications in the area of operations research, is challenging. Therefore, we developed new techniques to integrate these features also into a functional implementation context; the promising results were published.

In another research project, we worked on improving the implementation of declarative programs by source code transformations. Declarative programs often consist of the implementation of general schemes, enabled by higher-order functions or polymorphic typing, that are used in various contexts in application programs. This style supports an efficient and reliable software development process. However, it could create a run-time overhead compared to writing similar code pieces that are tailored to their precise usage. Automatic source code transformations like partial evaluation can help to close this gap. If general code schemes are partially instantiated with parameters, a partial evaluator transforms these instantiations and produces new code specific to these situations. In the reported period, we worked on a partial evaluator for Curry that takes the specific language features (non-determinism, demand-driven evaluations) into account in order to perform correct and effective program transformations. The first results on this tool were published.

A good implementation of high-level programming languages like Curry requires advanced *program analysis* techniques. For instance, the KiCS2 compiler analyzes dynamic program properties, like potential non-deterministic computations, to produce efficient target code. However, building program analyzers for realistic applications is not an easy task. In order to support developers of language tools in this respect (e.g. compiler, editor, documentation generator), we developed a new framework and implementation for the generic and modular analysis of declarative programs. This tool, called CASS (Curry Analysis Server System), is based on a plug-in architecture to accommodate various program analyzers. Moreover, CASS provides various usage interfaces so that it can be used by a variety of tools such as compilers and documentation generators, as well as Eclipse plug-ins for Curry. Finally, CASS itself is implemented as a master/worker architecture in

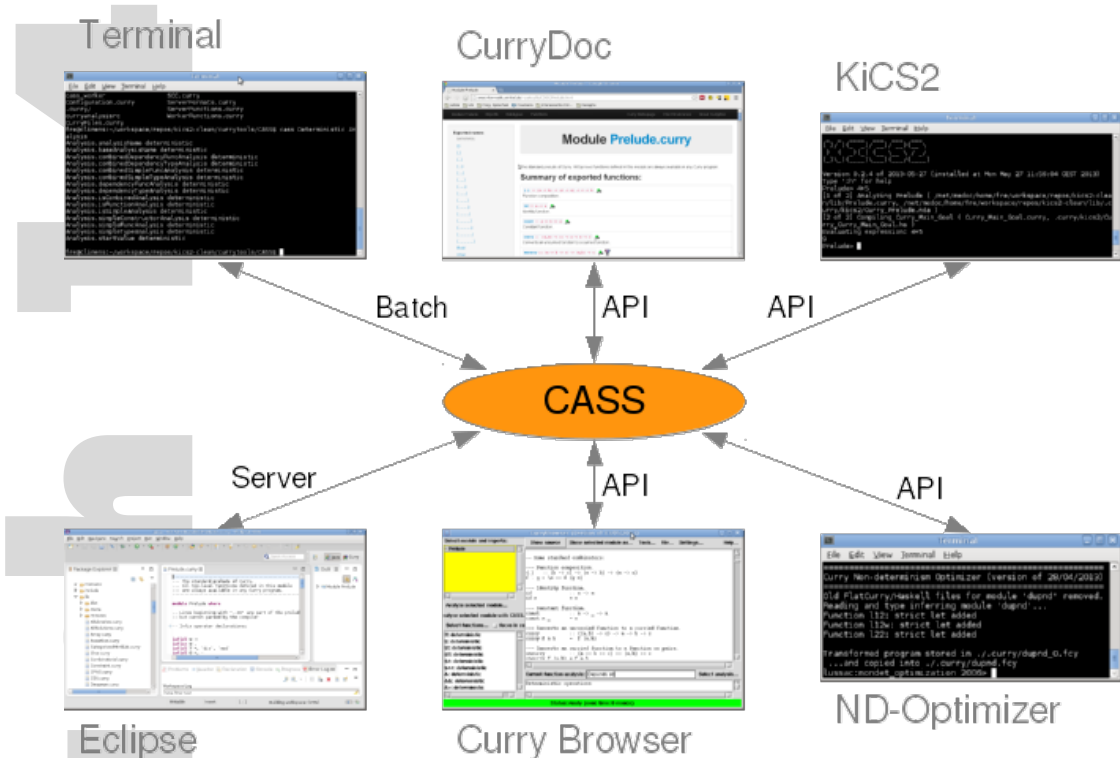


Fig. 1: The application structure of CASS

order to exploit parallel or distributed hardware environments. Since its publication and distribution in 2014, it has been used to develop new advanced program analysis techniques.

During the reported period, we also worked in the area of the *design* of Curry in a collaboration with the Portland State University (Oregon, USA). Since Curry integrates features from functional and logic programming, operations can be defined by case distinctions via pattern matching and conditional rules, as in functional programming, or also with unification and constraints, as in logic programming. This development led to similar but different concepts on the type level, namely Boolean values and constraints. Since the differences are subtle and choice of the right concept in a program might not be obvious, we proposed a radical change in the language design to simplify the language definition and join both concepts. Of course, this has consequences for the usage and implementation of Curry, which we discussed in a first paper. Nevertheless, the advantages being manifold we started to work on this new design in more detail.

On the practical side, we developed a system that should help people to play with the features of declarative languages without effort, i.e. without installing complex software systems. For this purpose, we developed Smap, a web-based editor and execution environment for programs written in various programming languages. Users can browse through example programs, execute or modify them, or develop their own programs and store them in Smap for other users. Smap supports web services for various programming languages and it can also be used in on-line books on programming languages to include example programs that are immediately executable. As a first practical use, all example programs contained in the tutorial on Curry are directly executable with Smap.

Personnel

Head of the group: Prof. Dr. M. Hanus; Secretary: M. Bradler (50%), L. Haberland (50%)
 Technical Staff: M. Gabriel (50%)

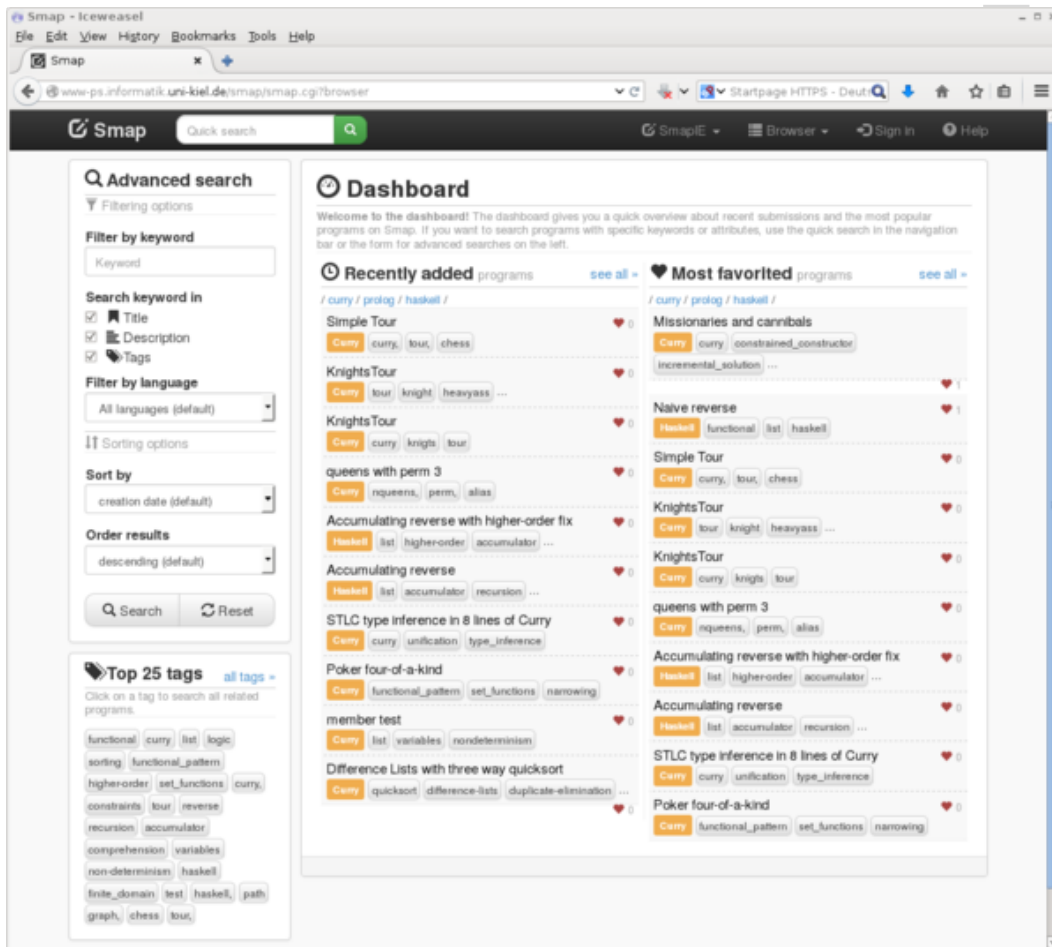


Fig. 2: The browser view of the web-based program development and execution environment Smap

Scientific Staff:

M. Sc. S. Dylus	01.11.-31.12.2014	CAU
Dr. S. Fischer	01.01.-31.12.2014	Guest
PD. Dr. F. Huch	01.01.-31.12.2014	CAU (35%)
M. Sc. B. Peemöller	01.01.-31.12.2014	CAU
Dr. F. Simon	01.01.-31.12.2014	Guest
Dipl.-Inf. F. Skrlac	01.01.-31.12.2014	CAU / Guest
Dipl.-Inf. J. R. Tikovsky	01.01.-31.12.2014	CAU

Lectures, Seminars, and Laboratory Course Offers

Winter 2013/2014

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

MSS0303: Masterabschlussseminar - 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 (+ J. R. Tikovsky, B. Peemöller, P. Munstermann)

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

Inf-MP-PS: Masterprojekt - Programmiersprachen und Programmiersysteme, 6 hrs Practical/Week,
M. Hanus

NF-Inf-1: Informatik für Nebenfächler, 2 (+ 2) hrs Lecture (+ Exercises)/Week,
F. Huch (+ F. Skrlac, J. Piwonski)

NF-Inf-1v: Informatik für Nebenfächler (vertiefend), 4 (+ 2) hrs Lecture (+ Exercises)/Week,
F. Huch (+ F. Skrlac, J. Piwonski)

Summer 2014

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

Inf-MP-PS: Masterprojekt - Programmiersprachen und Programmiersysteme, 4 hrs Exercise/Week,
M. Hanus (+ F. Skrlac)

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

MS0303: Deklarative Programmiersprachen, 4 (+ 2) hrs Lecture (+ Exercises)/Week,
M. Hanus (+ J. R. Tikovsky)

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

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

Inf-FortProg: Fortgeschrittene Programmierung, 3 (+ 2) hrs Lecture (+ Exercises)/Week,
F. Huch (+ B. Peemöller, J. R. Tikovsky)

Winter 2014/2015

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

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 (+ J. R. Tikovsky, B. Peemöller)

MS0302: Übersetzerbau, 4 (+ 2) hrs Lecture (+ Exercises)/Week,
M. Hanus (+ B. Peemöller, S. Dylus)

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

NF-Inf-1: Informatik für Nebenfächler, 2 (+ 2) hrs Lecture (+ Exercises)/Week,
F. Huch (+ Ch. D. Schulze, T. Wilke)

Inf-InfNat: Informatik für Naturwissenschaftler, 4 hrs Lecture/Week,
F. Huch (+ Ch. D. Schulze, T. Wilke)

NF-Inf-1v: Informatik für Nebenfächler (vertiefend), 4 (+ 2) hrs Lecture (+ Exercises)/Week,
F. Huch (+ Ch. D. Schulze, T. Wilke)

Further Cooperation, Consulting, and Technology Transfer

During the reported period, the research group collaborated with:

Sergio Antoy (Portland State University),

Ricardo Rocha (University of Porto).

Diploma, Bachelor's and Master's Theses

Mirko Heinold, *Automatisiertes Testen von Webanwendungen im Scala-Framework Play*, 25.11.2013

Torsten Krause, *Entwicklung eines sprachunabhängigen Tools zum modellbasierten schnellen Prototyping*, 24.03.2014

Lasse K. Meyer, *Entwicklung einer Webanwendung zur Erstellung, Verwaltung und Ausführung von Programmen in Curry*, 25.03.2014

Jan Bracker, *A Web-Based Editor for Cloud-Based Programming*, 27.03.2014

Lennart Spitzner, *Extending an Eclipse-Plugin for Curry by Features for Program Analysis, Type-Checking and Debugging*, 27.03.2014

Jasper P. Sikorra, *Foreign Code Integration in Curry*, 31.03.2014

Max A. Deppert, *Integration von Auszeichnungssprachen in Curry*, 31.03.2014

Bastian Holst, *Integration of Parallel and Fair Search Strategies for Non-Deterministic Programs into the Curry System KiCS2*, 27.05.2014

Florian Micheler, *A Transaction Framework for Web Applications in Haskell*, 10.06.2014

Daniel Stark, *Konfigurierbare Übersetzer von Java nach C++ und C# zur Unterstützung betrieblicher Anwendungen*, 11.08.2014

Folke Will, *Maschinencode-Obfuscation als Schutz vor Reverse Engineering*, 25.09.2014

Mike Tallarek, *Implementierung einer Datenbank-Schnittstelle für Curry*, 30.09.2014

Sandra Dylus, *Lenses and Bidirectional Programming in Curry*, 30.09.2014

Publications

Published in 2014

M. Hanus, F. Skrlac, *A Modular and Generic Analysis Server System for Functional Logic Programs*, Proc. of the ACM SIGPLAN 2014 Workshop on Partial Evaluation and Program Manipulation (PEPM'14), **ACM Press**, 181 - 188 (2014)

M. Hanus, B. Peemöller, J. R. Tikovsky, *Integration of Finite Domain Constraints in KiCS2*, Proc. of the 7th Working Conference on Programming Languages (ATPS 2014), 151 - 170 (2014)

M. Hanus, S. Koschnicke, *An ER-based Framework for Declarative Web Programming*, Theory and Practice of Logic Programming, **14(3)**, 269 - 291 (2014)

M. Hanus, *Multiparadigm Languages*, Computing Handbook: Computer Science and Software Engineering, **Third Edition**, 6601 - 6617 (2014)

- M. Hanus, *Declarative Multi-Paradigm Programming*, Proceedings of the 23rd International Workshop on Functional and (Constraint) Logic Programming (WFLP 2014), 5 - 7 (2014)
- S. Antoy, M. Hanus, *Curry without Success*, Proceedings of the 23rd International Workshop on Functional and (Constraint) Logic Programming (WFLP 2014), 40 - 54 (2014)
- M. Hanus, B. Peemöller, *A Partial Evaluator for Curry*, Proceedings of the 23rd International Workshop on Functional and (Constraint) Logic Programming (WFLP 2014), 55 - 71 (2014)
- M. Hanus, R. Rocha, *Declarative Programming and Knowledge Management*, Springer Lecture Notes in Artificial Intelligence, **8439**, (2014)

Presentations

- M. Hanus, *A Modular and Generic Analysis Server System for Functional Logic Programs*, ACM SIGPLAN 2014 Workshop on Partial Evaluation and Program Manipulation (PEPM'14), San Diego, USA, 21.01.2014
- J. R. Tikovsky, *Integration of Finite Domain Constraints in KiCS2*, 7th Working Conference on Programming Languages (ATPS 2014), Kiel, Germany, 26.02.2014
- B. Peemöller, *A Partial Evaluator for FlatCurry*, 31st GI-Workshop Programmiersprachen und Rechenkonzepte, Bad Honnef, Germany, 28.04.2014
- F. Will, *Kianxali: Kieler Analyzer for Executables and Libraries*, 31st GI-Workshop Programmiersprachen und Rechenkonzepte, Bad Honnef, Germany, 28.04.2014
- F. Huch, *Stolpersteine bei der Integration funktionaler Konzepte in die imperative Programmierung*, 31st GI-Workshop Programmiersprachen und Rechenkonzepte, Bad Honnef, Germany, 28.04.2014
- S. Fischer, *Backtracking in unterschiedlichen Programmierparadigmen*, 31st GI-Workshop Programmiersprachen und Rechenkonzepte, Bad Honnef, Germany, 28.04.2014
- J. R. Tikovsky, *Solving FD constraints incrementally using KiCS2*, 31st GI-Workshop Programmiersprachen und Rechenkonzepte, Bad Honnef, Germany, 28.04.2014
- M. Hanus, *Declarative Multi-Paradigm Programming*, 23rd International Workshop on Functional and (Constraint) Logic Programming and the 28th Workshop on (Constraint) Logic Programming, Wittenberg, Germany, 16.09.2014
- M. Hanus, *Curry without Success*, 23rd International Workshop on Functional and (Constraint) Logic Programming and the 28th Workshop on (Constraint) Logic Programming, Wittenberg, Germany, 16.09.2014
- B. Peemöller, *A Partial Evaluator for Curry*, 23rd International Workshop on Functional and (Constraint) Logic Programming and the 28th Workshop on (Constraint) Logic Programming, Wittenberg, Germany, 16.09.2014

Further Activities and Events

- M. Hanus is a programme committee member of the following: -Seventh Working Conference on Programming Languages (ATPS 2014), Aachen, February 2014 (part of the conference Software Engineering 2014),
- International Joint Workshop on Implementation of Constraint and Logic Programming Systems and Logic-based Methods in Programming Environments 2014 (CICLOPS-WLPE 2014), Vienna (Austria), July 2014,
- 30th International Conference on Logic Programming (ICLP 2014), Vienna (Austria), July 2014.

He also chairs the programme committees below:

- 28th Workshop on (Constraint) Logic Programming (WLP 2014), Wittenberg, September 2014,
- 23rd International Workshop on Functional and (Constraint) Logic Programming (WFLP 2014), Wittenberg, September 2014.

M. Hanus also has the following positions:

- Chair of the steering committee of the ACM SIGPLAN Symposia on Principles and Practice of Declarative Programming,

- Member of the steering committee of the Symposia on Logic-based Program Synthesis and Transformation,
- 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),
- Member of the advisory board of the GLP (Gesellschaft für Logische Programmierung), German-speaking branch of the Association for Logic Programming (ALP),
- Member of the advisory board of the „Berufsakademie an der Wirtschaftsakademie Schleswig-Holstein“ ,
- Member of the executive board of the „Fakultätentag Informatik der Bundesrepublik Deutschland“ ,
- Chair of the selection committee of the award for the best diploma or M.Sc. thesis in computer science in Germany 2014,
- Chair of the examinations board of computer science studies, University of Kiel
- Member of the Senate Curriculum Committee, University of Kiel,
- Vice-member of the Senate Equal Opportunities Committee, University of Kiel.

S. Dylus received an ACM-W Scholarship Award in order to support her stay in Gothenborg, Sweden, to participate in the 19th ACM SIGPLAN International Conference on Functional Programming (ICFP 2014).

M. Gabriel was interviewed by FLOSS Weekly (Episode 295, <http://twit.tv/show/floss-weekly/295>) on May 21, 2014 about the remote desktop solution X2Go.