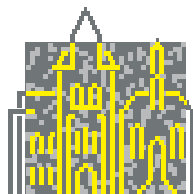


D. Batory (UT at Austin, USA), C. Consel (LaBRI, Bordeaux, F), C.
Lengauer (Univ. Passau, D), M. Odersky (EPFL, Lausanne, CH)
(Editors)

Domain-Specific Program Generation

Dagstuhl Seminar 03131 – March 23 to March 28, 2003
Dagstuhl-Seminar-Report No. 373



SCHLOSS DAGSTUHL

INTERNATIONALES
BEGEGNUNGS-
UND FORSCHUNGSZENTRUM
FÜR INFORMATIK

ISSN 0940-1121

Herausgegeben von IBFI gem. GmbH, Schloss Dagstuhl, 66687 Wadern, Germany.

Das Internationale Begegnungs- und Forschungszentrum für Informatik (IBFI) Schloss Dagstuhl ist eine gemeinnützige GmbH. Sie veranstaltet regelmäßig wissenschaftliche Seminare, welche nach Antrag der Tagungsleiter und Begutachtung durch das wissenschaftliche Direktorium mit persönlich eingeladenen Gästen durchgeführt werden.

Gesellschafter:

- Gesellschaft für Informatik e.V. – Bonn
- TH Darmstadt
- Universität Frankfurt
- Universität Kaiserslautern
- Universität Karlsruhe
- Universität Stuttgart
- Universität Trier
- Universität des Saarlandes

Summary Report

Dagstuhl Seminar 03131

Don Batory, Charles Consel, Christian Lengauer, Martin Odersky

1 Public Statement

Generative approaches have the potential to revolutionize software development as automation and components revolutionized manufacturing. This technology is particularly effective when applied with domain-specific techniques, since compact, domain-specific notations make programs easier to write and maintain and domain-specific knowledge allows for a more efficient implementation.

The purpose of the seminar was to promote scientific cooperation on the topic of domain-specific program generation. This topic has been pursued so far in a number of research communities which had insufficient contact with each other:

- *Domain-specific languages:* Language developers in a specific application domain have often been unaware of the domain-independent aspects of their domain-specific work. Vice versa, researchers that do not work in a specific domain are often unaware of the experiences made in application work.
- *High-performance parallelism:* This is one application domain, which has led to the development of a particular form of domain-specific language (so-called skeletons). Researchers in this community has been quite unaware of the wider aspects of domain-specific program generation.
- *Program generators:* This domain is concerned with the fast and reliable generation of members of a program family (so-called “product lines”). Researchers in this community are often in industry.
- *Metaprogramming:* researchers in this community develop a technology that can be used for customizing compilation and translation systems for domain-specific purposes.

As a main result of the seminar, initial steps were taken to form a working group. Also cooperation on an individual basis was fostered. A compendium of papers presented at the seminar is in preparation for the Lecture Notes in Computer Science series of Springer-Verlag.

2 Scientific Highlights and Perspectives

The seminar contributed significantly to the formation of a new community around the seminar topic:

- People from different language backgrounds –notably imperative languages (main representative: C++) and declarative languages (main representative: Haskell)– got to know each others’ work. People were very receptive of each other: everybody went away with a wider horizon; new cooperations were formed at several levels; some led to newly formed joint authorships for the compendium.
- Talks on applications made a special impact. In particular, the community on high-performance parallelism (the “skeletons” community) was recognized as a domain with a

special need for optimization. The integration of this domain in the wider context of the research community is a major result of the seminar.

- Plans were made to form a formal working group. A mission statement was being formulated (see Section 4).

3 Presentation and Publication

3.1 Numbers

Talks altogether: 35 Keynotes: 3 Talks of young researchers: 4

3.2 Titles

C. Conzel: Tutorial on Domain-Specific Languages (Keynote)
D. Wile: Domain-Specific Languages: Lessons Learned
C. Lengauer: Domain-Specific Optimization
K. Czarnecki: Technology Projections in Generative Programming
Y. Smaragdakis: Research Directions for the Generator Community
U. Eisenecker: Recent Research in Generative Programming
D. Batory (given by S. Krishnamurthi): The Road to Utopia (Keynote)
P. Devanbu: Aspects in Distributed Computing
A. Rauschmayer: Safe Multidimensional Software Evolution (young researcher)
E. Visser: Strategies for Program Transformation
C. Ramming: Language Support for Resilient Systems
W. Taha: Multi-Stage Programming
P. Feautrier: Tools for Metaprogramming (Keynote)
T. Veldhuizen: An Alternative to Staging and Metaprogramming
U. Assmann: Invasive Software Composition
M. Odersky: Types for Objects and Modules
M. Cole: Libraries for Skeletal Parallel Programming
H. Kuchen: A Skeleton Library
S. Gorlatch: Optimization of Compositions in Parallel and Distributed Programs
H. Bischof: A Generative C++ Library with Skeletons (young researcher)
S. Newhouse: ICENI – Applications on the Grid
K. Hammond: EC Funding in the Sixth Framework
A. Cohen: WRaP-IT: Metaprogramming in the Polytope Model
B. Fischer: Automatic Program Synthesis for the Data Analysis Domain
J. Striegnitz: Extensible Programming Environments
D. Gregg: Generating Efficient Virtual Machine Interpreters (young researcher)
P. Kelly: Run-Time Code Generation in C++
C. Herrmann: Using Haskell as a Metaprogramming Language (young researcher)
J. O'Donnell: Embedding DSLs with Template Haskell
K. Hammond: Real-Time Functional Programming
J. Lawall: Capturing OS Expertise in an Event Type System
L. Réveillère: DSLs for Device Drivers and/or Streaming
S. Krishnamurthi: Modular Verification of Product Lines
B. Haumacher: JavaParty: A DSL for Cluster Computing
C. Conzel: Scaling up the DSL Approach: The Nova Platform

3.3 Publication

23 submissions have been announced by participants for a planned compendium on the seminar topic. The submissions are based on talks given or discussions conducted at the seminar. Springer-Verlag has been approached for publication. We expect the book to appear some time next year.

4 Working Group on Domain-Specific Program Generation: Mission Statement

4.1 Aim

Generative approaches have the potential to revolutionize software development as automation and components revolutionized manufacturing. Such approaches are particularly effective when combined with domain-specific techniques, since compact, domain-specific notations make programs easier to write and maintain, and domain-specific knowledge allows for a more efficient implementation.

The aim of this Working Group of researchers and practitioners is to promote innovation in

- foundations
- design
- engineering
- techniques
- tools
- applications

for domain-specific program generation.

4.2 Scope

The scope of this WG covers all aspects of design, analysis, generation, and quality control of generative programs and the programs that they generate, with emphasis on the use of domain-specific knowledge.

Specific research themes include (but are not limited to the following areas):

- *Foundations*: language design, semantics, type systems, formal methods, multi-stage and multi-level languages, validation and verification.
- *Design*: models of generative programming, domain engineering, domain analysis and design, system family and product line engineering, model-driven development, separation of concerns, aspect-oriented modeling, feature-oriented modeling.
- *Engineering*: practices in the context of program generation, such as requirements elicitation and management, software process engineering and management, software maintenance, software estimation and measurement

- *Techniques*: meta-programming, staging, templates, in-lining, macro expansion, reflection, partial evaluation, intentional programming, staged configuration, stepwise refinement, software reuse, adaptive compilation, runtime code generation, compilation, integration of domain specific languages, testing.
- *Tools*: open compilers, extensible programming environments, active libraries, frame processors, program transformation systems, program specializers, aspect weavers, and tools for domain modeling.
- *Application*: IT infrastructure, finance, telecom, automotive, aerospace, space applications, scientific computing, health, life sciences, manufacturing, government, systems software and middle-ware, embedded and real-time systems, generation of non-code artifacts.

4.3 Objectives

- Foster collaboration and interaction between researchers from domain engineering, and on language design, meta-programming techniques, and generative methodologies.
- Demonstrate concrete benefits in specific application areas.
- Develop techniques to assess productivity, reliability, and usability.

Participants

- Aßmann, Uwe (TU Dresden)
- Bischof, Holger (TU Berlin)
- Cohen, Albert (UPMC – Paris)
- Cole, Murray I. (University of Edinburgh)
- Consel, Charles (Université Bordeaux)
- Czarnecki, Krzysztof (University of Waterloo)
- Devanbu, Premkumar T. (University of California – Davis)
- Eisenecker, Ulrich W. (Universität Leipzig)
- Ellmenreich, Nils (Universität Passau)
- Feautrier, Paul (ENS – Lyon)
- Fischer, Bernd (NASA / RIACS – Moffett Field)
- Gorlatch, Sergei (Universität Münster)
- Gregg, David (University College Dublin)
- Größlinger, Armin (Universität Passau)
- Hammond, Kevin (University of St Andrews)
- Haumacher, Bernhard (Universität Karlsruhe)
- Herrmann, Christoph A. (Universität Passau)
- Kelly, Paul H. J. (Imperial College London)
- Krishnamurthi, Shriram (Brown University – Providence)
- Kuchen, Herbert (Universität Münster)
- Lawall, Julia (University of Copenhagen)
- Lengauer, Christian (Universität Passau)
- Newhouse, Steven (Imperial College London)
- O’Donnell, John (University of Glasgow)
- Odersky, Martin (EPFL – Lausanne)
- Philippsen, Michael (Universität Erlangen-Nürnberg)
- Ramming, J. Christopher (DARPA – Arlington)
- Rauschmayer, Axel (LMU München)
- Réveillère, Laurent (INRIA – Bordeaux)
- Smaragdakis, Yannis (Georgia Institute of Technology)
- Striegnitz, Jörg (Jülich Supercomputing Centre)
- Taha, Walid (Rice University – Houston)

03131 – Domain-Specific Program Generation

- Veldhuizen, Todd (Chalmers – Göteborg)
- Visser, Eelco (Utrecht University)
- Wile, David (Teknowledge Corporation)