Available online at www.sciencedirect.com
SCIENCE DIRECT®

ELSEVIER

Science of Computer Programming 62 (2006) 1–2

Science of Computer Programming
www.elsevier.com/locate/scico

Preface

It is a pleasure for the editors of this special issue to present this collection of papers as the proceedings of the First MetaOCaml Workshop held in 2004. MetaOCaml (http://www.metaocaml.org) is an extension of OCaml with basic support for type-safe program generation. This support takes the form of a special-purpose quasi-quotation mechanism for constructing programs as well as a construct for evaluating such dynamic generated programs. The quasi-quotations are hygienic in the sense that they can only be used to generate programs, and all binding occurrences of variables are renamed to avoid accidental variable capture. The novelty of this support lies in statically checking the quasi-quotes to ensure that generated programs are always well-typed, and in statically checking the evaluation construct to ensure that it is never invoked on program fragments that contain unbound variables. Over the last ten years, the use of such quasi-quotations along with a run construct for program generation – called multi-stage programming (MSP) – has attracted the attention of both the programming languages and program generation communities. More recently, it has been applied in the development of embedded software and digital circuit design.

The five papers presented here were selected from eight submissions based on papers presented at the first MetaOCaml Workshop:

- Carette shows how a Gaussian Elimination is in fact a family of algorithms, and how MetaOCaml can express this family concisely as one parameterized implementation *without* paying a runtime overhead for the parameterization.
- Cohen et al. report on their experience implementing high-performance computing transformations using staging. They identify some instances where this can already be done, as well as others which remain challenging.
- Herrmann and Langhammer present a detailed case study on implementing a domain-specific language (DSL) for image processing in MetaOCaml. The study uses the staged interpreter technique combined with a number of optimizations.
- League shows how web publishing can be usefully viewed as staged computation. He presents MetaOCaml Server Pages, a domain-specific language for web applications programming, and uses it to construct several compelling arguments for why web publishing is a particularly promising domain for multi-stage programming.
- Neverov and Roe present a detailed and insightful analysis of their experience with taking key ideas underlying MetaOCaml and applying them to the Java/C# setting.

We would like to take this opportunity to thank the rest of the MetaOCaml team (Cristiano Calcagno, Edward Pizzi, Oleg Kiselyov, Roumen Kaiabachev, Jason Eckhardt, Xavier Leroy and Emir Pasalic) for their work on the implementation. We would also like to thank the Program Committee of the first MetaOCaml Workshop, as well as the external reviewers. Finally, MetaOCaml would not have been possible without the generous support of the National Science Foundation through award #0113569 entitled “ITR/SY(CISE): Putting Multi Stage Annotations to Work”.

Program Committee of the MetaOCaml 2004 Workshop:
Kedar Swadi, Persistent Systems
Cristiano Calcagno, Imperial College
Walid Taha, Rice University (Chair)
Rowan Davies, University of Western Australia
Todd Veldhuizen, Chalmers University

doi:10.1016/j.scico.2006.05.001

Ralf Hinze, University of Bonn
Stephanie Weirich, University of Pennsylvania
Robert Glück, University of Copenhagen
Hongwei Xi, Boston University
Oleg Kiselyov, Fleet Numerical Meteorology and Oceanography Center

External Reviewers:
Stephan Ellner, Google Technologies
Eric Allen, Sun Microsystems
Peter Sestoft, Royal Veterinary and Agricultural University
Jim Larus, Microsoft Research
Jeremy Condit, University of California at Berkeley
Sam Kamin, University of Illinois at Urbana-Champaign
John Mellor-Crummey, Rice University
Olav Beckmann, Imperial College
Emir Pasalic, Rice University
Scott Crosby, Rice University
Jeremy Siek, Rice University
Alexander Nanevski, Harvard University

Guest Editors
Christian Lengauer
University of Passau,
Innstrasse 33, Room 129,
94030 Passau,
Germany
E-mail address: lengauer@fmi.uni-passau.de.

Walid Taha*
Rice University,
Computer Science,
6100 Main,
77005-1827 Houston, TX,
United States
E-mail address: taha@rice.edu.

Available online 6 June 2006

* Corresponding editor.