TLP **7** (1&2): 1–2, 2007. © 2007 Cambridge University Press
doi:10.1017/S1471068406002808 Printed in the United Kingdom

Introduction Special Issue on Multiparadigm Languages and Constraint Programming

MORENO FALASCHI
Dipartimento di Scienze Matematiche e Informatiche, University of Siena, Italy
(*e-mail:* moreno.falaschi@unisi.it)

MICHAEL MAHER
National ICT Australia, Kensington, Australia
(*e-mail:* mmaher@cse.unsw.edu.au)

In recent years much research and implementation effort has been devoted both to multiparadigm languages and constraint programming languages. Following up on a series of 11 workshops (WFLP) on multiparadigm languages and constraint programming, and as a result of an open call for submissions, the journal on Theory and Practice of Logic Programming is now publishing the results of the selection of the papers submitted to this special issue.

The aim of this issue was to attract high-quality research papers on all aspects related to multi-paradigm languages and constraint programming, such as semantics, analysis, extensions, implementations and applications. Papers addressing how to use constraints for integrating different paradigms, or extensions of multiparadigm languages with constraints were particularly welcome, although papers solely on multiparadigm programming or constraint programming were also of interest. Surveys and comparisons of constraint techniques and multiparadigm languages were also solicited. This special issue should hence stimulate the cross-fertilizing exchange of ideas and experiences among researchers from the different communities interested in the foundations, applications, and combinations of high-level, multiparadigm and constraint programming languages and related areas.

The competition was high. We received 16 papers in response to the call for papers. Each paper was revised by three anonymous referees and eight papers had to be rejected after the first round of reviews, in order to allow for timely publication of the issue. After a second round of reviews the following eight papers were finally accepted.

- *Removing Redundant Arguments Automatically.*
  Alpuente, Escobar and Lucas propose a method for the analysis and elimination of redundant arguments in term rewriting systems as a model for the programs that can be written in more sophisticated languages.
- *Combining Relational Algebra, SQL, Constraint Modelling, and Local Search.*
  Cadoli and Mancini show how to integrate constraint modelling and relational

https://doi.org/10.1017/S1471068406002808 Published online by Cambridge University Press

DBMSs. They propose (non-deterministic) extensions of standard query languages such as relational algebra and SQL, by adding constraint modelling capabilities to them.

- *Constraint-based Automatic Verification of Abstract Models of Multithreaded Programs.*
  Delzanno presents a technique for the automated verification of abstract models of multithreaded programs providing fresh name generation, name mobility, and unbounded control.
- *Integration of Declarative and Constraint Programming.*
  Hofstedt and Pepper show an approach for integrating different language paradigms and constraint programming into a unified framework.
- *Mapping Fusion and Synchronized Hyperedge Replacement into Logic Programming.*
  Lanese and Montanari compare three different formalisms that can be used in the area of models for distributed, concurrent and mobile systems. They analyze the relationships between a process calculus, the *Fusion Calculus*, a formalism for graph transformations in the *Synchronized Hyperedge Replacement* with Hoare synchronization (HSHR) and *logic programming*.
- *Demand Analysis with Partial Predicates.*
  Mariño, Herranz and Moreno-Navarro present a framework for a set constraint based *demand analysis* for integrated languages. This analysis can improve the implementation and performance of integrated languages.
- *A comparison between two logical formalisms for rewriting.*
  Palomino studies the relationships between Meseguer's rewriting logic and the rewriting logic CRWL. He shows that even though it is not possible to establish an entailment system map between the two logics, both can be naturally simulated in each other.
- *Forward Slicing of Functional Logic Programs by Partial Evaluation.*
  Silva and Vidal introduce the first forward slicing technique for declarative multi-paradigm programs which integrate features from functional and logic programming. This technique can be applied to a wide variety of software engineering tasks, like program understanding, maintenance, debugging, testing, code reuse, etc.

We want to thank the anonymous referees for their invaluable work, Marco Comini from the University of Udine for having organized the special issue web site, Krzysztof Apt and Maurice Bruynooghe, as the editor in chief of Theory and Practice of Logic Programming, for their support from the very beginning to the idea of this special issue.

https://doi.org/10.1017/S1471068406002808 Published online by Cambridge University Press