Introduction to the Special Issue on Abstraction and Automation in Constraint Modelling

Alan M. Frisch
University of York
United Kingdom
Email: frisch@cs.york.ac.uk

Ian Miguel
The University of St Andrews
United Kingdom
Email: ianm@cs.st-and.ac.uk

Constraints are a powerful and natural means of knowledge representation and inference, and have proven successful at solving a wide range of combinatorial problems. Constraint solving of a combinatorial problem proceeds in two phases. First, the problem is modelled by a set of constraints on decision variables that its solutions must satisfy. Second, a constraint solver is used to search for solutions to the model.

In general, many models are possible for a given problem. The model chosen can have a substantial effect on the efficiency of the solving phase, but selecting an effective model is difficult, often requiring a great deal of expertise. Hence, there is a modelling bottleneck, which research must address for constraint solving technology to be more widely used by non-experts.

One way of improving usability is by extending constraint technology to enable models to be formulated at a higher level of abstraction. For instance, support for set variables (variables whose domain values are sets) in many constraint languages and solvers has abstracted away from the low-level details of how the set variable is represented; the user no longer needs to know these details. Recent work on constraint languages, such as Hnich's $\mathcal{F}$, Flener *et al.*'s Esra, and Cadoli *et al.*'s NP-Spec, has been attacking the direct support of other types of values, such as functions and relations. However, with the exceptions of set and multiset variables, many of these abstract variable types are not yet supported directly by constraint solvers.

In this case, the abstract variable can be refined into a representation that comprises a set of more primitive variables and a collection of constraints among them. In order to avoid forcing the user to perform this step manually, automated refinement is a key goal.

Automation can also aid the modelling process by transforming a constraint model into one that can be solved more effectively. Such transformations include: adding implied constraints; recognising symmetries in models so that symmetry-breaking methods can be used to reduce solution time; adding constraints to exploit dominances in optimisation problems; removing propagation-redundant constraints; and creating relaxed versions of the initial problem.

This special issue contains five papers that present research that addresses the constraint modelling bottleneck through abstraction and automation. Two of these, "The Design of the Zinc Modelling Language", by Marriott *et al.*, and "Essence: A Constraint Language for Specifying Combinatorial Problems", by Frisch *et al.*, describe general abstract constraint languages. A third, "Domain Specific High-level Constraints for User Interface Layout", by Lutteroth, Strandh and Weber, is more focused, offering abstractions tailored to the specification of the layout of graphical user interfaces. "The Power of Abstraction in Essence", by Mitchell and Ternovska, analyses the expressive power of abstract constraint languages, and the Essence language in particular. Finally, Cambazard and O'Sullivan's paper, "Reformulating Table Constraints using Functional Dependencies - An Application to Explanation Generation" describes an automated technique for reformulating high-arity extensional (or 'table') constraints into lower-arity constraints.

We gratefully acknowledge the help of all the reviewers that contributed their time and expertise to this special issue. We are espcially grateful to Peter van Beek, Editor-in-chief of Constraints, for inviting us to edit this special issue and for offering expert guidence throughout the process. Peter served as editor for the paper "Essence: A Constraint Language for Specifying Combinatorial Problems" ; Peter and Ian Miguel served as editor for the paper "The Design of the Zinc Modelling Language."

Alan M. Frisch & Ian Miguel, February 2008.