

**Martin, Julien; Martinez, Thierry; Fages, François**

**Procedural code generation vs static expansion in modelling languages for constraint programming.** (English) [\[Zbl 1322.68184\]](#)

Larrosa, Javier (ed.) et al., Recent advances in constraints. 14th annual ERCIM international workshop on constraint solving and constraint logic programming, CSCLP 2009, Barcelona, Spain, June 15–17, 2009. Revised selected papers. Berlin: Springer (ISBN 978-3-642-19485-6/pbk). Lecture Notes in Computer Science 6384. Lecture Notes in Artificial Intelligence, 38-58 (2011).

Summary: To make constraint programming easier to use by the non-programmers, a lot of work has been devoted to the design of front-end modelling languages using logical and algebraic notations instead of programming constructs. The transformation to an executable constraint program can be performed by fundamentally two compilation schemas: either by a static expansion of the model in a flat constraint satisfaction problem (e.g. Zinc, Rules2CP, Essence) or by generation of procedural code (e.g. OPL, Comet). In this paper, we compare both compilation schemas. For this, we consider the rule-based modelling language Rules2CP with its static expansion mechanism and describe with a formal system a new compilation schema which proceeds by generation of procedural code. We analyze the complexity of both compilation schemas, and present some performance figures of both the compilation process and the generated code on a benchmark of scheduling and bin packing problems.

For the entire collection see [\[Zbl 1209.68005\]](#).

**MSC:**

- 68T20** Problem solving in the context of artificial intelligence (heuristics, search strategies, etc.)
- 68N30** Mathematical aspects of software engineering (specification, verification, metrics, requirements, etc.)
- 68P05** Data structures

**Software:**

[Rules2CP](#)

**Full Text:** [DOI](#)