Artificial Intelligence for Engineering Design, Analysis and Manufacturing (2003), *17*, 1–2. Printed in the USA.
Copyright © 2003 Cambridge University Press 0890-0604/03 $16.00
DOI: 10.1017/S0890060403171016

GUEST EDITORIAL

Special Issue: Configuration

TIMO SOININEN[1] AND MARKUS STUMPTNER[2]

[1]Helsinki University of Technology, Department of Computer Science and Engineering, Software Business and Engineering Institute, FIN-02015 HUT, 02015 Espoo, Finland
[2]University of South Australia, Advanced Computing Research Centre, Adelaide, Australia

The configuration task is commonly defined as composing a complex product from a set of predefined component types while taking into account a set of well-defined restrictions on how components belonging to these types can be combined. Configuration, always a successful artificial intelligence (AI) application area ever since the R1/XCON system of the early 1980s, has recently attracted renewed research interest. This is also demonstrated by an annual series of workshops on configuration that have been held at the AAAI, ECAI, and IJCAI conferences since 1999. Important real-world industrial configuration tasks are encountered in marketing, manufacturing, and design. They usually involve physical products, such as telecommunication switches, computers, elevators, large diesel engines, automation systems, or vehicles (some of which appear as application domains in the articles in this issue), but can also pertain to financial or other services or software.

Because of their complexity, these tasks must often be supported by dedicated software tools, termed *configurators*. Efficient development and maintenance of configurators require sophisticated software development techniques. AI methods are central to the development of powerful configuration tools and to extending configurator functionality. A wide range of AI techniques such as constraint satisfaction and its extensions, description logics, logic programming, and different specialized problem solving methods are being used for this purpose. With the rise of E-business and web-based configuration services, the field presents many new and old challenges to researchers and application developers.

The goal of this Special Issue is to demonstrate the current state of the art in configuration research. The five papers collected in it reflect a cross section of current research and were selected from 14 initial submissions resulting from the Call for Papers. Each paper was subjected to double blind reviews by two to four expert reviewers, and those papers that gained preliminary acceptance were re-reviewed after resubmission to ensure that any critical comments by the reviewers had been considered. The review of manuscripts that included a coeditor as an author was managed exclusively in a screened manner by the uninvolved coeditor.

This issue follows 5 years after the first Special Issue on Configuration Design in this journal, and it seems appropriate to dwell on the progress that has been made in the meantime and the shifting or persisting areas of interest that are indicated by the paper topics. It is noticeable that the two traditional mainstays of configuration research, the search for appropriate reasoning mechanisms and suitable representation formalisms with clear formal foundations are still progressing. Likewise, the main streams of configuration languages that were in evidence in 1998 (constraint based and their close relatives, logic based on one hand and description logic based on the other) are still in evidence as well, and work is continuing with the goal of bringing them under one hat. However, new configuration related problems have also emerged and are being addressed.

Moving along this spectrum, the first two papers deal with extensions of constraint satisfaction methods for solving particular aspects of configuration reasoning. "The Goldilocks Problem," by Tudor Hulubei, Eugene C. Freuder, and Richard J. Wallace examines a value-ordering heuristic that is intended to effectively reduce the large number of solutions that often arise in configuration problems, because they are typically underconstrained: the Goldilocks problem consists of retaining only those solutions that are "just right." Experiments with randomly generated problems show that the heuristic provides the ability to still solve problems for which standard algorithms such as maintained arc consistency (MAC) fail. The second constraint-oriented paper also deals partially with the same problem. In "Preference Programming: Advanced Problem Solving for Configuration," Ulrich Junker and Daniel Mailharro describe a method for representing user preferences about the outcome of a computation in a set of rules that can also be

Reprint requests to: Timo Soininen, Helsinki University of Technology, Department of Computer Science and Engineering, Software Business and Engineering Institute, P.O. Box 9600, FIN-02015 HUT, 02015 Espoo, Finland. E-mail: timo.soininen@hut.fi

https://doi.org/10.1017/S0890060403171016 Published online by Cambridge University Press

used in an elegant fashion to reflectively regulate the preferences themselves, thus encapsulating reasoning strategies for improving computational efficiency.

The next two papers go beyond the view of a configuration problem as a monolithic entity and examine the interaction between multiple configuration (sub)problems from two different perspectives. The paper "Configuration Knowledge Representations for Semantic Web Applications," by Alexander Felfernig, Gerhard Friedrich, Dietmar Jannach, Markus Stumptner, and Markus Zanker, focuses on the requirements for enabling joint configuration tasks between multiple cooperating configuration solvers for E-commerce environments, in particular establishing interchange standards between different configuration formalisms. The paper examines requirements placed on such standards by *de facto* standard configuration ontologies, defines a mapping from the standard logic-based configuration formalisms to the DAML+OIL Semantic Web language, and points out areas where its expressiveness is currently too restricted and will need to be extended.

The other paper in this group, "Decomposition Strategies for Configuration Problems," by Diego Magro and Pietro Torasso, approaches the topic from a more algorithmic point of view, considering different approaches that can be used to improve configurator performance by automatically finding a promising decomposition of a configuration problem into (almost) separate subproblems and then efficiently solving them. The paper identifies the properties of certain configuration problems that will facilitate the application of particular methods. These configuration problems are modeled using a structured language that is similar to description logics.

The final paper, "Formal Methods for the Validation of Automotive Product Configuration Data," by Carsten Sinz, Andreas Kaiser, and Wolfgang Küchlin, is on one hand the most application-oriented paper in this issue, and at the same time ends the Special Issue by leading back to a logic-based representation. It describes a knowledge base consisting of propositional formulae, which are formulated on the basis of technical documentation, that are used to validate the output of a legacy configurator system (a rule-based system implemented in COBOL) that performs automated batch configuration tasks for automobile production. Using a SAT checker developed for this purpose, the system serves as a tool that can support the actual configuration process, support maintenance by identifying the ramifications of knowledge base changes, and help in validating the knowledge base by showing inconsistencies.

Together, these papers show the continuing potential for intriguing basic and applied research that has served to make configuration an attractive research area in the past and will continue to do so in the foreseeable future.

https://doi.org/10.1017/S0890060403171016 Published online by Cambridge University Press