HELSINKI UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering

Laboratory of Information Processing Science

Timo Soininen

An Approach to Configuration Knowledge Representation and Reasoning

Licentiate thesis submitted for official examination for the degree of Licentiate in Technology in Espoo on December 9th, 1998.

Supervisor: Professor Reijo Sulonen

Instructor: Professor Reijo Sulonen

HELSINKI UNIVERSITY OF TECHNOLOGY

ABSTRACT OF THE LICENTIATE THESIS

Author: Timo Soininen

Title of the thesis: An Approach to Configuration Knowledge Representation and Reasoning

Date: December 9th, 1998 **Number of pages:** 110

Department: Department of Computer Science and Engineering

Professorship: Tik-76 Information Processing Science (ZE02)

Supervisor: Professor Reijo Sulonen

Instructor: Professor Reijo Sulonen

Configuration task can be roughly defined as the problem of designing a product individual using a set of predefined components while taking into account a set of well-defined restrictions on how the components can be combined. A configuration task can be automated or supported by a product configurator, i.e. a knowledge based information system.

In this work configurable products, configuration task related business processes and their impact on product configurators were analyzed. Companies were found to have similar processes and problems. Changing the business to configurable products may require major changes to the existing products and processes. It is not obvious when these changes are profitable. A product configurator is often necessary for the change to be profitable.

On the basis of the analysis a model of concepts for representing configuration knowledge was developed. This configuration ontology synthesizes several previous approaches and is extended with new concepts. In addition, a partial formal model of the computation and reasoning in configuration tasks was defined. It can be used to represent and reason on some aspects of the ontology. The configuration task was analyzed on the basis of the formal model and found to be **NP**-complete.

A prototype based on the formal model was implemented and the feasibility of the approach was studied using two simple examples. For these examples the implementation was efficient. It is argued that configuration tasks may be inherently well-structured. This would mean that the worst-case exponential computation implied by **NP**-completeness does not materialize in real world cases. This assumption, the ontology and the proposed formal model should each be empirically validated. Several extensions to the ontology and formal model and integrating them more tightly are pointed out as future work.

Keywords: Product configuration, configurable product, product configurator, ontology, knowledge representation and reasoning

teknisillä tietojärjestelmillä, tuotekonfiguraattoreilla.

Tässä työssä analysoitiin konfiguroitavia tuotteita ja konfigurointiin liittyviä liiketoimintaprosesseja sekä näiden merkitystä tuotekonfiguraattoreille. Eri yritysten konfigurointiin liittyvät prosessit ja niiden ongelmat havaittiin samankaltaisiksi. Yrityksen liiketoiminnan muuttaminen konfiguroitaviin tuotteisiin perustuvaksi voi vaatia huomattavia muutoksia yrityksen tuotteisiin ja prosesseihin. Ei ole kuitenkaan selvää milloin nämä muutokset ovat kannattavia. Kannattavuuden edellytyksenä on usein tuotekonfiguraattorin käyttöönotto.

Tuotteiden ja prosessien analyysin perusteella kehitettiin käsitteistö konfigurointitietämyksen esittämiseen. Tämä konfigurointiontologia yhdistää useat aiemmat lähestymistavat sekä laajentaa niitä uusilla käsitteillä. Käsitteistön lisäksi kehitettiin osittainen formaali malli konfigurointitehtäviin liittyvästä laskennasta ja päättelystä. Mallia voidaan käyttää joidenkin konfigurointiontologian käsitteiden esittämiseen ja niiden pohjalta tapahtuvaan päättelyyn. Formaalin mallin mukaisen konfigurointitehtävän laskennallinen vaativuus osoittautui **NP**-täydelliseksi.

Työssä kehitettiin prototyyppikonfiguraattori esitettyyn formaaliin malliin perustuen. Prototyypin todettiin toimivan tehokkaasti kahdella yksinkertaisella esimerkkituotteella. Konfigurointiongelmat näyttävät luonnostaan olevan rakenteellisia. Tämän takia **NP**-täydellisyydestä johtuva pahimmassa tapauksessa eksponentiaalinen laskennallinen vaativuus ei näyttäisi toteutuvan todellisissa tehtävissä. Tämä oletus, konfigurointiontologia ja esitety formaali malli tulisi kukin validoida kokeellisesti. Jatkotyöksi ehdotetaan useita laajennuksia ontologiaan ja formaalin malliin sekä näiden yhdistämistä.

Avainsanat:	Tuotekonfigurointi, konfiguroitava tuote, tuotekonfiguraattori, ontologia, tietämyksen esittäminen ja päättely

Preface

This work has been carried out in the research projects of Product Data Management Group (PDMG) of Helsinki University of Technology (HUT). It has been made possible by the financial support of Helsinki Graduate School in Computer Science and Engineering (HeCSE). The projects have also been funded by the Technology Development Centre Finland and the Academy of Finland.

As usual, I have several people to thank. All the people working in the PDMG have contributed to a stimulating and comfortable working environment. The ideas presented in this thesis have benefited from the numerous discussions with the members of the group. Without professor Reijo Sulonen's excellent advice, timely encouragement, critique and support this thesis would not have been possible. Juha Tiihonen has been an invaluable colleague in analyzing the practical issues related to product configuration and keeping the work relevant to the practice. Tomi Männistö and Hannu Peltonen have been sources of excellent support in providing critique to and further understanding of the issues covered by this work and how to best put them down in writing. Asko Martio has provided valuable insight of the industrial practices related to configuration and their effects on the issues covered in this work.

In addition to the PDMG, I am grateful to many people working in HUT and the industrial companies that PDMG has co-operated with. Foremost of these is Ilkka Niemelä, whose deep understanding of issues related to knowledge representation, reasoning and computational complexity and the ability to clearly communicate the intuition underlying them have inspired me to study these issues in the configuration domain. Professors Martti Mäntylä, Markku Syrjänen and Leo Ojala have been sources of valuable information on product modeling and artificial intelligence. I am also grateful to Patrik Simons and Tommi Syrjänen for implementing ideas that have come up in co-operation

A special loving thank you goes to Mia, whose patience and support made writing this thesis possible without severe damage to household relations.

Espoo, 9th December 1998

Timo Soininen

Table of Contents

List of Publications

This work is based on the following four publications, which are referred to in the text by their roman numerals.

I Tiihonen, Juha, Timo Soininen, Tomi Männistö, and Reijo Sulonen. 1996. State-of-the-practice in product configuration—A survey of 10 cases in the Finnish industry. In *Knowledge intensive CAD*, vol. 1, ed. Tetsuo Tomiyama, Martti Mäntylä, and Susan Finger, 95-114. London: Chapman & Hall.

II Tiihonen, Juha and Timo Soininen. 1997. *Product Configurators – Information System Support for Configurable Products*. Technical Report TKO-B137, Helsinki University of Technology, Laboratory of Information Processing Science. Also published in Richardson, Tom, ed. 1997. *Using Information Technology During the Sales Visit*. Cambridge, UK: Hewson Group.

III Soininen, Timo, Juha Tiihonen, Tomi Männistö and Reijo Sulonen. 1998. Towards a General Ontology of Configuration. In *AI EDAM (Artificial Intelligence for Engineering Design, Analysis and Manufacturing)* 12, no. 4.

IV Soininen, Timo and Ilkka Niemelä. Developing a declarative rule language for applications in product configuration. To appear in Gobal Gupta, ed., *Practical Aspects of Declarative Languages: First International Workshop*. Springer-Verlag, 1999.

1 Introduction

1.1 Background

The design and production of goods[1] that satisfy specific needs of individual customers are of central interest to the European and especially Scandinavian companies. The major trends in the business environment of these companies include diminishing lifetimes of products, increasing complexity and number of variants of products, and shorter lead-times in the sales-delivery processes. Moreover, there is increasing pressure to adapt product individuals according to customer requirements.

One way to cope with the changes in the business environment is to develop and deliver *configurable products*. A configurable product can be characterized by the following properties:

- Each delivered product individual is adapted to meet the requirements of a customer.
- The product has been pre-designed to meet a given range of different customer requirements. It is not meant to be adapted to meet requirements outside this range.
- Each product individual is specified as a combination of pre-designed components. New components are not designed in the sales-delivery process to adapt the product.
- The product has a pre-designed general structure.
- The adaptation in the sales-delivery process requires only routine design and can be done in a systematic manner.

A configurable product typically has a large number of variants. It allows systematic adaptation, *configuration*, through which the customer-specific variants are specified. The goal is to do this while keeping the adaptation easy, routine, and manageable, and the lead-time of the sales-delivery process short. In other words, a configurable product aims at combining some of the benefits of mass-produced and one-of-a-kind products. This type of operation has sometimes been called mass-customization (Hales 1992; Carson 1997).

A *product configurator*[2], or *configurator* for short, is an information system that configures a product or supports a human in doing it. The *configuration task* can be roughly defined as the problem of designing a product individual using a set

---

[1] In the following, the term “product” encompasses both physical and logical products, such as software, and services.

[2] The word “configurer” is sometimes used with the same meaning. There seems to be no standard terminology. The word “configurator” is used in this thesis for an information system. The word “configurer” is reserved for a person that does the configuration task, possibly supported by a configurator.

of pre-defined components while taking into account a set of well-defined restrictions on how the components can be combined. The inputs of the problem are a *configuration model*, which describes the components that can be included in the configuration and the rules on how they can be combined to form a working product, and *requirements* that specify some properties that the product individual should have. The output is a *configuration*, an accurate enough description of a product individual. The configuration must satisfy the requirements and be correct in the sense that it does not break any of the rules in the configuration model.

Product configurators have been used as an aid in the sales-delivery process at least from the beginning of the 1980s (McDermott 1982). In the last five years the number of vendors of product configurators has bloomed and the configurator business has grown to hundreds of millions of dollars annually (Richardson 1997). There are at least a dozen commercial systems available[3] (Richardson 1997). The commercial success of the field is also witnessed by the growing number of companies that have taken a configurator into use (Richardson 1997; Faltings and Freuder 1998).

Although the use of configurators in companies started and is probably most wide-spread in the USA, already at least a dozen Finnish companies have taken a configurator into use. Several Finnish companies have also developed and delivered configurable products for some time without configurator support (I). This is natural as products that can be adapted to customer specific requirements are very important for the Finnish industry. It seems that the Scandinavian companies are among the first in Europe to acquire product configurators. There are also a few Finnish companies that produce configurators, one of which was recently acquired by a major US company.

This thesis is organized as follows: In Section 1.2 previous research on configuration is briefly discussed. The research problem and goals of this research are defined in Section 1.3. The scope of the research and the research methods are discussed in Sections 1.4 and 1.5. Then, in Section 2 brief summaries of the annexed publications are given. In Section 3 some conclusions are given and further work is outlined. The publications annexed to the thesis follow as appendices.

1.2 Research in Configuration

Numerous theoretical models of product configuration tasks and reports on implemented configurators have been presented (see, e.g., Cunis et al. 1989; Heinrich and Jüngst 1991; Mittal and Frayman 1989; Faltings and Freuder 1996; Baader et al. 1996; Faltings and Freuder 1998; Hales 1992; Darr, McGuinness and Klein 1998; Schreiber and Birmingham 1996). In these, the

---

3 The configuration homepage at http://www.cs.unh.edu/ccc/config/ provides a list of references to the homepages of vendors of commercial systems under the heading Who's Who/Companies.

configuration task has been usually characterized as a subclass of design that can be done in a routine manner. The term "product configuration" has usually been used for this type of routine design that take place in the sales-delivery process, whereas the term *configuration design* may also encompass more design-oriented activities that take place in the product development process.

Configuration tasks and product configurators have been studied within the field of artificial intelligence (AI) for at least two decades. Most of the research has concentrated on problem solving methodologies, such as constraint satisfaction-type (Haselböck and Stumptner 1993; Mittal and Falkenhainer 1990; Gelle and Weigel 1996; Weigel and Faltings 1996; Sabin and Freuder 1996), resource-based (Heinrich and Jüngst 1991) and propose-and-revise type approaches (Balkany, Birmingham and Tommelein 1993; Schreiber and Birmingham 1996). Knowledge-based systems (KBS) employing techniques such as constraint satisfaction, its generalizations and description logics have been successfully applied to real world product configuration tasks (Faltings and Freuder 1996; Faltings and Freuder 1998). The reason for the success of KBSs in this area is that the relevant knowledge is usually well-defined and complete.

In addition to the AI-centric study of product configuration, *configurable products* have recently emerged as a new field of study more closely linked to the research on product design and industrial management. The study of configurable products includes analyzing what types of products are configurable, how easy it is to configure a product, what factors affect the feasibility of operating with configurable products and what are the methods and processes that companies use to develop and deliver configurable products (Tiihonen et al. 1998; Jorgensen and Raunsbaek 1998). The importance of this research field has become apparent due to the commercial success of product configurators, although configurable products were important already before configurators became successful.

Despite the research there is no widely accepted model of configuration tasks that would cover all the relevant aspects in a satisfactory manner. The theoretical models presented so far have different viewpoints. Most of them also lack a sound formal basis that would allow a rigorous analysis of the configuration tasks and comparison of the models. Notable formal exceptions are (Najman and Stein 1992), (Klein, Buchheit and Nutt 1994), some of the approaches in (Baader, Bürckert, Günter and Nutt 1996) and the constraint based approaches mentioned above. In addition, there has been relatively little research on configurable products. The differences in the models, lack of formal models, and the practical significance of the field make further research on configuration tasks, product configurators and configurable products important. This has lead to a surge of new symposia, workshops and journal issues devoted to configuration in the past few years (Faltings and Freuder 1996; Baader, Bürckert, Günter and Nutt 1996; Faltings and Freuder 1998; Darr, McGuinness and Klein 1998).

1.3 Research Problem and Goals

The problem that this work tries to partially solve was twofold:

- What are the requirements posed by real-world configuration tasks on product configurators?
- Is there a formal model of real-world configuration tasks? Which aspects of them can be efficiently supported by computers?

The first problem is related to the practical processes and methods through which companies manage and deliver configurable products, i.e. research on configurable products. The second problem is more related to developing a conceptual and computational model of real world configuration tasks. The model should be analyzed from computational complexity (see, e.g., Papadimitriou 1994) point of view to characterize how hard the configuration task is, whether it can be done efficiently on a computer, and to identify which aspects make the problem hard or easy.

The primary goals of this work were to:

- Analyze and describe the processes in companies through which configurable products are developed and delivered, the factors that affect the configuration related processes, and their impact on the required information system support. (Addressed in (I) and (II).)
- Develop a model of the *configuration concepts*, i.e. a *configuration ontology*, that can be used to represent configuration models, configurations and requirements. The model should enable accurate communication of the knowledge and its computer based manipulation. It should synthesize the previous research. (Addressed in (III).)
- Develop a formal model of configuration tasks, i.e. of the computation and reasoning required in them. In addition, the computational complexity of configuration tasks should be analyzed on the basis of the formal model. (Addressed in (IV).)
- Implement a prototype that carries out configuration tasks on the basis of the formal model to study the feasibility of the approach. (Addressed in (IV).)

1.4 Scope

The results on configurable products originate from the Finnish discrete manufacturing industry. The companies studied were interested in developing their processes, products, information systems and knowledge management related to product configuration. They were neither a random sample nor a collection of state-of-the-art companies. This restriction was caused by the pre-study nature of the research due to limited understanding of the problem, which did not allow a more rigorous statistical study. It was not clear what the relevant parameters were and how they should be measured.

Only few companies had begun using a configurator. Others had configurable products and were only beginning to consider product configurator support. This means that the whole impact of product configurators may not be covered by the research. Only those impacts that can be logically expected could be identified, but unexpected side effects could not yet be seen.

In developing the configuration ontology a product configuration point of view was taken. This means that the ontology may not cover all the things required for configuration design. Nor does it cover the geometry, pricing and optimality of configurations, which are important for some products. Construction and control knowledge on how to accomplish the configuration task, i.e. what actions and in which order can be taken to configure a product (Günter, Cunis and Syska 1990) were also excluded. These restrictions were made to simplify the problem to a manageable size. It is assumed that the lacking aspects can be defined on top of the ontology developed in this work.

The formal model of configuration tasks and its implementation was also based on a simplification of the real world configuration tasks and the configuration concepts. It is assumed that the more complex tasks and configuration concepts can be defined on top of this model. The justification for this restriction was again the complexity of the whole problem. The implementation was tested on a few small examples only instead of modeling a set of real world products. This was done to get preliminary results on the feasibility of this approach fast before proceeding with any more large scale experiment.

1.5 Method

The analysis and definition of the processes and methods by which companies develop and deliver configurable products was a synthesis of a survey of ten companies that deliver configurable products and experiences gained in joint-projects with half a dozen other companies that use or plan to use product configurators (I; Soininen 1996; Tiihonen 1994).

The development of the configuration ontology was based on a synthesis of a set of previously presented theoretical models, which had little in common except the central notion of a component. These models were chosen because they all have gained popularity in the research community or have been used in commercial applications. Experiences on configurable products had indicated that a synthesis of these models is needed to compactly and adequately represent the knowledge on products. The ontology was defined using the Frame Ontology of the Ontolingua approach to ontology development (Gruber 1992).

The formal model of configurations tasks was defined through developing a rule-language for expressing typical configuration knowledge elements and giving it a formal declarative semantics. The language was defined with the goal that some of the concepts in the configuration ontology can be straightforwardly represented in it. In addition, the aim was that it should be possible to represent the rest of the concepts by extending the language. For formalizing the language, techniques developed in non-monotonic reasoning and logic programming research (see, e.g. Przymusinska and Przymusinski 1990) were relevant. The complexities of the main decisions problems for the language were analyzed using the techniques developed in the field of computational complexity (see, e.g., Papadimitriou 1994).

The semantics of the rule language was found to be closely related to the declarative semantics of logic programs. This relation was exploited in developing the first implementation of the language, which is based on an existing logic programming system. In order to estimate the feasibility of this approach two simple imaginary configuration problems were studied.

2 Summary of Publications

2.1 Configurable Products and Configuration Processes

2.1.1 State of the Practice in Finland

In (I) product configuration is argued to constitute a broader subject of research than had previously been studied. It is noted that there was surprisingly little research on the way product configuration problems are understood in the industry.

A framework for characterizing and analyzing product configuration was developed. The framework consists of five problem areas that are further refined into a number of factors characterizing the areas. The problem areas cover the *economic importance* of product configuration, the complexity of the *configuration task*, the nature of the *configuration process*, *long term management of configuration related product knowledge* and the *interfaces* to other processes and systems. The proposed framework can be used to analyze product configuration tasks and processes. The framework can also be utilized as an extended checklist when assessing the usability of proposed models, solutions and tools.

The proposed framework was used in a survey of ten case studies. The areas, factors and related questions proved to be useful in collecting data from the companies and subsequently analyzing the cases. The preliminary results of the survey indicated that configurable products and the related configuration processes were important to many companies in Finland. The main reasons for operating with configurable products were the ability to meet a wide range of customer requirements, increased control of production and reduced lead times in the delivery process. All the companies had room for improvement in at least one area, usually several.

The companies had approached configurable products from both mass-produced products and one-of-a-kind products for different reasons. The annual number of delivered product individuals varied from a few dozen to a few thousand. In addition to configurable products, many companies delivered tailored products that involved both configuration and innovative engineering design.

The configuration activities and related knowledge were usually not systematized. The companies lacked good methods and tools to represent the configuration models. Typical concepts that were used to describe products were the components of the product, their compositional structure, requires and incompatibility relations between these, and the functions that the product provides to the customer. Some customer requirements were given as a functional specification, while some other requirements were formulated as component selections. There was a clear need for a general methodology and tools for representing configuration knowledge. It seemed possible to develop a general model that would be feasible for a range of companies.

Most companies did not even attempt to reconfigure existing product individuals using the same methods as in configuring new ones. Rather, the reconfiguration task was done on a case by case basis. Long term management of configuration models was also a problem for all the companies. The processes that developed the products usually did not create the configuration related information as a part of the development effort. Rather, this was an additional task done by persons that are not product experts. This may lead to loss of data and erroneous configuration knowledge being used in the configuration process.

The configurations were manually checked by product experts or by sales persons. The case companies were not yet using product configurators, but nearly all of them intend to have computer support within the next few years. There was considerable variation in the maturity of the companies to make this transition. Companies had relatively strong disbelief in and bad experiences with expert systems. The long term management of products and product knowledge was considered a risk factor.

One of the primary motives for building a support system for product configuration was to assist in the transfer of up-to-date product configuration knowledge to the sales units and to enforce its proper use. Another driving force for automated product configuration was the desire to reduce the number of errors to improve quality.

It was argued that the general configuration problem cannot be solved solely with better models and tools. Flexible configuration of products must be considered already while products and components are designed. (*Design for Configuration*). Modularity of a product seemed to have a favorable effect on the complexity of the product configuration task.

2.1.2 Configurable Products

In (II) the types of configurable products and configuration related processes that can gain the most from utilizing product configurators are analyzed. It was noted that configurable products transfer much of the design work from the sales-delivery process to the product development process. This requires systemizing the product and the related product knowledge. The systemized product knowledge produced by the product development process can be re-used several times in the sales-delivery process.

However, in order to accomplish this the product should be relatively easy to configure. A modular product architecture (Ulrich and Eppinger 1995) seems to facilitate this, whereas an integral one (Ulrich and Eppinger 1995) seems more difficult from the configuration point of view. It is also crucial to identify the most common requirements to which the product needs to be adapted through configuration. This knowledge should be documented in the early stages of the development process.

The sales-delivery process of configurable products has in general two stages: sales and engineering configuration. In the sales stage the product individual may be specified in terms of the abstract functions or modules that satisfy the customer requirements, which results in a sales specification or order. In the engineering stage the output of the preceding stage is used in connection with the configuration model for this stage to produce a more concrete definition of the product individual. The configuration process in a company may not contain all these stages, for example all of the configuration task may be done in the sales or engineering stage.

Companies had moved to configurable product from mass products and one-of-a-kind products. This direction affects the necessary changes to the processes of the company. When making the transition from one-of-a-kind products the main effort probably goes to pre-designing and systemizing the products so that they are configurable. The investment is profitable only if the volume of delivered product instances is high enough.

Introducing a configuration process to a delivery process which has previously operated with fixed mass products can cause problems, as more of the specification work is expected to be done by the sales persons and in the engineering configuration. The problems in the transition to configurable products from either direction are to large extent due to the fact that the increased effort and the benefits gained are experienced by different functions within the company.

The product and process-oriented view was used as a basis for discussing product configurators. Configurators with up-to-date product knowledge allow non-product-experts to make error-free sales specifications and production orders. The reduction in the lead-times of the sales-delivery process was analyzed. Most of the reduction is caused by reducing the number of the iterations between customers, sales persons and engineers in producing a correct configuration. In this respect a configurator functions as an essential enabler for business process re-engineering (Hammer and Champy 1993).

Long-term management and maintenance of the product knowledge as product models and product individuals evolve was identified as the major problem that can prohibit successful use of a configurator. The long term management capability of configurators was analyzed. The current commercial systems were found to support only few aspects to varying degrees. Support for reconfiguration of product individuals was almost non-existent.

The underlying thesis in (II) is that product configurators on their own are not enough to make the sales and order fulfillment processes more efficient. The success of a configurator in any company is based on adequate systemization of the product, in some cases even re-designing the product for configurability, and the systemization and reengineering of the configuration related processes. If these aspect are properly accomplished, a product configurator can be a major reengineering asset whose benefits can be truly dramatic. However, choosing between the different commercially available configurators is not easy as they operate on different principles and ideas. Long-term management of products should be a major criterion in this.

2.2 Configuration Ontology

In (III) a configuration ontology is presented. It was argued that a general ontology is needed to re-use and share configuration knowledge. Despite the research on configuration, such an ontology had not emerged. It was further argued that a general ontology of configuration is an equally important research issue as the problem solving methodologies for configuration tasks. These issues are connected, as the ontology affects the computational methods that can be used to carry out the configuration task and vice versa, but they should be given equal attention.

The ontology presented is a synthesis of the main approaches to configuration. It consists of a set of concepts for representing the knowledge on a configuration and the restrictions on possible configurations. The concepts include components, attributes, resources, ports, contexts, functions, constraints and relations between these. Earlier approaches were extended with new concepts arising from practical experience on configurable products. The main extensions were in the detailed conceptualization of knowledge on product structures and in extending the resource concept with contexts for limiting the availability and use of resources. The concepts were treated uniformly with respect to classification, which had not been the case in previous work. In addition, constraint sets representing different views on the product were introduced.

The ontology was compared with previous work on configuration. It was found be the most generic ontology presented so far in the sense that it covered all the principal approaches to configuration, i.e. connection-based, structure-based, resource-based and function-based approaches. In addition, the ontology made as few commitments as possible to enable extending and refining the ontology. This was accomplished through flexibility in the dependencies between the different concepts.

It was noted that the ontology contains overlapping concepts for representing some phenomena. In particular, the concepts could be shown to be overlapping in the sense of formal expressiveness. However, this was argued to be less important than preserving the clarity of configuration models, which should not be compromised by minimizing the number of concepts in a modeling language.

2.3 Formal Model of Configuration

In (IV) a partial formal model of product configuration tasks is developed. A rule-based language was proposed for expressing typical forms of knowledge elements in configuration models, i.e. choices, dependencies between choices, optionality and defaults. These knowledge elements seem to underlie many of the models of configuration tasks. The developed formal model can be considered a common denominator of several models of configuration tasks. Consequently, the fundamental aspects of these models can be represented in the proposed language. It was also a goal that the language can be extended to cover other aspects of configuration. Several constraint based formalisms CSP, DCSP (Mittal and Falkenhainer 1990) and GCSP (Haselböck and Stumptner 1993) can also be considered such denominators.

The language was equipped with a simple declarative semantics that provided formal definitions for the main concepts in product configuration, i.e., configuration model, requirements, configuration, satisfiability of requirements and validity of a configuration. A key feature was that the semantics ensures that valid configurations are tightly grounded in the configuration rules without resorting to an explicit minimality condition on configurations. This type of groundedness had not been considered in previous work on product configuration. It is important that the elements of a configuration are grounded in the configuration model, i.e. that they have a justification for being in a configuration. Otherwise, a configuration may have superfluous elements that are not needed for the product to function and satisfy requirements.

The use of groundedness instead of minimality conditions had a favorable effect on the complexity of the configuration tasks. For example, the validity of a configuration as well as satisfiability of requirements can be decided efficiently, in linear time, and also other computational task remain in **NP** (see, e.g., Papadimitriou 1994), which has been the usual computational complexity of the formalisms used for configuration tasks. Formalizing groundedness as minimality would have lead to higher computational complexity, where even the problem corresponding to the validity of a configuration cannot be efficiently solved. For practical configuration tasks, it is a minimal requirement that the validity of a configuration and satisfiability of requirements can be decided efficiently.

It was argued that from a knowledge representation point of view the rule-based language is more attractive for representing configuration knowledge than constraints. This is done by showing that CSP and dynamic CSP (Mittal and Falkenhainer 1990) can be embedded in the proposed language, but the mapping in the other direction is not straightforward. This is due to the difficulty of capturing justifications in CSP and to more expressive rules that seem difficult to capture in DCSP.

The semantics of the rule-based language was shown to be closely related to the stable model semantics of normal logic programs (Gelfond and Lifschitz 1988). This connection was exploited in the first implementation which is based on a translator from rules to normal programs and on a high performance implementation of the stable model semantics, the Smodels system (Niemelä and Simons 1997).

In order to estimate the feasibility of this approach two simple configuration problems based on an example presented in (Mittal and Falkenhainer 1990) were modeled and solved. Such examples were found to be straightforward to model in the language. The implementation exhibited reasonable performance for interactive applications on the examples.

There are indications that the proposed formal model provides a basis for solving practically relevant product configuration problems. Experiences in other domains have shown that efficient implementations of the stable model semantics are capable of handling tens of thousands of ground rules. Compiling a practically relevant configuration model from a high-level representation based on the configuration ontology (III) into the rule language would seem to generate rule sets of approximately that size.

3 Conclusions and Future work

There seems to be a general model of developing and delivering configurable products. This model is followed by the numerous companies for whom configurable products are important. Changing the business to follow the model may require major changes in the business processes and practices of a company. There are significant potential benefits of such operation compared to delivering fixed mass products or one-of-a-kind products. These include capability to fulfill a wide range of customer requirements and shorter lead times and increased quality in the sales-delivery process.

However, it is not clear when operation with configurable products is more profitable or when it is useful to acquire a product configurator to support in configuration tasks. It is also not evident how a product should be designed to be configurable. The model presented in this work indicates some aspects that are prerequisites for operating with configurable products. One important topic of further work is to analyze more thoroughly the issues that make configurable products feasible for a company. The relevant issues that should be studied in real world cases include requirements posed by the business environment and organization, degree of systemization of product development and sales-delivery process with respect to configuration related activities, modularity of product, degree to which product knowledge in configuration models can be re-used, complexity of the resulting configuration model and pace of changes to the configuration model.

There was a clear need for configuration knowledge representation methods and product configurators. However, there is no generally accepted commercial or even theoretical solution that would cover the different products and needs of companies. The configuration ontology presented in this work shows that the previous approaches can be unified in a flexible manner to cover the practical needs better. Like the previous approaches, the ontology and the formal model of configuration task presented in this work are still only partial ones. The formal model does not even cover the ontology presented. They should be both developed further and integrated to adequately cover configuration knowledge and the configuration tasks in a formal model. This would facilitate a rigorous analysis of the ontology and different modeling languages based on subsets of the ontology. The computational complexity of the configuration tasks for such languages should be studied.

A more general ontology and model of configuration task should include geometric, pricing, scheduling and optimality related issues and the knowledge on how to configure a product. There is also a need to extend the formal model to cover interactive configuration, where the user makes the hard decisions and the computer only the efficiently computable ones. This may be the only feasible alternative for very large or complex problems. In addition, this type of assistance in configuration tasks seems to be more acceptable to companies than completely automatic configurators. Another important extension would be to extend the ontology and model to cover the long-term management of configuration models and configurations. This aspect, which may be orthogonal to the other extensions, seems very important for the practical applications.

The ontology and formal model and their extensions should be validated by empirically modeling different kinds of products and testing how the implementation approach scales for larger, real world problems. The relevance of the ontology depends mostly on how easy it is to model different kinds of products. The relevance of the formal model and its implementation is determined by whether it can be used to efficiently configure real products.

In the proposed formal model a computer can efficiently check whether a configuration is correct and satisfies a set of requirements. The configuration task, however, is **NP**-complete, which means that in the worst case the configuration task cannot probably be done efficiently by a computer. The extensions to the formal model should be designed so that it remains efficiently computable to check if a configuration is correct and satisfies a set of requirements. This implies that the configuration task must be in **NP**. If this goal is not achieved, the formal model may become irrelevant for practical purposes, as a minimum requirement of a practical system is that the checking tasks can be done efficiently.

On the other hand, even though the configuration task is **NP**-complete, the simple test problems were solved satisfactorily. Experiences from other domains indicate that the approach taken here scales up to larger problems. It may turn out that configuration problems are also so well structured that the exponential worst-case behavior implied by **NP**-completeness does not materialize. This structure could be a result of the product being designed by humans. Designers very often recursively decompose a design problem into relatively independent parts. Thus, a product is typically not an ill-structured system where everything depends on everything else. Further research is needed to validate this assumption.

The approach taken in this work was to translate a configuration specific language to another formalism which is used in the problem solving. It may be possible to develop a more efficient algorithm for configuration tasks that avoids the overhead and loss of information incurred by the translation. Devising such an algorithm would be an interesting subject of further work. A practically important task would also be to identify additional syntactically restricted but still useful subsets of the language that would allow more efficient computation.

References

Baader, Franz, Hans-Jürgen Bürckert, Andreas Günter, and Werner Nutt, eds. 1996. *Proceedings of the Workshop on Knowledge Representation and Configuration WRKP'96*. Document D-96-04. DFKI GmbH (German Research Center for Artificial Intelligence).

Balkany, Alan, William P. Birmingham, and Iris D. Tommelein. 1993. An analysis of several configuration design systems. *AI EDAM* 7, no. 1: 1-17.

Carson, C. 1997. Intelligent Sales Configuration. *PC AI* (January/February).

Cunis, R., A. Günter, I. Syska, H. Peters, and H. Bode. 1989. PLAKON — An approach to domain-independent construction. In *The Second International Conference on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems IEA/AIE-89*, 866-874.

Darr, Timothy, Deborah McGuinness, and Mark Klein, guest editors. 1998. Special Issue on Configuration Design. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 12, no. 4.

Faltings, Boi, and Eugene Freuder, eds. 1996. *Configuration—Papers from the 1996 AAAI fall symposium*. Technical report FS-96-03. Menlo Park, California: AAAI Press.

Faltings, Boi, and Eugene Freuder, guest editors. 1998. Special Issue on Configuration. *IEEE intelligent systems & their applications* (July/August): 29-85.

Gelfond, M., and V Lifschitz. 1988. The stable model semantics for logic programming. In *Proceedings of the 5th International Conference on Logic Programming*, 1070-1080. The MIT Press.

Gelle, Esther, and Rainer Weigel. 1996. Interactive Configuration using Constraint Satisfaction Techniques. In *Configuration—papers from the 1996 AAAI Fall Symposium*, ed. Boi Faltings and Eugene Freuder, 37-44. AAAI Press.

Gruber, Thomas. 1992. *Ontolingua: A Mechanism to Support Portable Ontologies. Version 3.0*. KSL 91-66. Stanford University, Knowledge Systems Laboratory.

Günter, Andreas, Roman Cunis, and Ingo Syska. 1990. Separating control from structural knowledge in construction expert systems. In *Proc. of the Third International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE 90)*, 601-610. ACM.

Hales, H. L. ed. 1992. Automating and Integrating the Sales Function: How to Profit From Complexity and Customization. *Enterprise Integration Strategies* 9, no. 11: 1-9.

Hammer, M., and J. Champy. 1993. *Re-engineering the Corporation - A Manifesto for Business Revolution*. New York: HarperCollins.

Haselböck, Alois, and Markus Stumptner. 1993. An integrated approach for modelling complex configuration domains. In *Proc. of the AI'93, Avignon*.

Heinrich, M., and E. W. Jüngst. 1991. A resource-based paradigm for the configuring of technical systems from modular components. In *Proc. of the Seventh IEEE Conference on Artificial Intelligence Applications*, 257-264. IEEE.

Jorgensen, Kaj A., and Thomas Raunsbaek. 1998. Design of Product Configuration Management Systems. In *Proceedings of 2nd International Conference on Engineering Design and Automation*. Integrated Technology Systems, Inc.

Klein, R., M. Buchheit, and Werner Nutt. 1994. Configuration as Model Construction: The Constructive Problem Solving Approach. In *Proceedings of Artificial Intelligence in Design 1994*, 201-218.

McDermott, John. 1982. R1: a rule-based configurer of computer systems. *Artificial Intelligence* 19, no. 1.

Mittal, Sanjay, and Brian Falkenhainer. 1990. Dynamic constraint satisfaction problems. In *Proc. of the 8th National Conference on Artificial Intelligence (AAAI-90)*, 25-32. MIT Press.

Mittal, Sanjay, and Felix Frayman. 1989. Towards a generic model of configuration tasks. In *Proc. of the 11th International Joint Conference on Artificial Intelligence (IJCAI)*, 1395-1401.

Najman, O., and B. Stein. 1992. A theoretical framework for configurations. In *Proc. of IEA/AIE 92*, 441-450.

Niemelä, Ilkka, and Patrik Simons. 1997. Smodels - an implementation of the stable model and well-founded semantics for normal logic programs. In

Proceedings of the 4th International Conference on Logic Programming and Non-monotonic Reasoning, 420-429. Springer-Verlag.

Papadimitriou, Christos H. 1994. *Computational complexity*. Reading, Massachusetts: Addison-Wesley Publishing Company, Inc.

Przymusinska, Halina, and Teodor Przymusinski. 1990. Semantic Issues in Deductive Databases and Logic Programs. In *Formal Techniques in Artificial Intelligence A Sourcebook*, ed. R. B. Banerji, 321-367. Elsevier Science Publishers B.V. (North-Holland).

Richardson, Tom, ed. 1997. *Using Information Technology During the Sales Visit*. Cambridge, UK: Hewson Group.

Sabin, Daniel, and Eugene Freuder. 1996. Configuration as Composite Constraint Satisfaction. In *Configuration—papers from the 1996 AAAI Fall Symposium*, ed. Boi Faltings and Eugene Freuder, 28-36. AAAI Press.

Schreiber, A. Th, and William P. Birmingham, guest editors. 1996. The Sisyphus-VT initiative. *International Journal of Human-Computer Studies* 44, no. 3.

Soininen, Timo. 1996. Product configuration knowledge: Case study and general model. Master's thesis, Helsinki University of Technology.

Tiihonen, Juha. 1994. Computer-assisted elevator configuration. Master's thesis, Helsinki University of Technology.

Tiihonen, Juha, Timo Soininen, Tomi Männistö, and Reijo Sulonen. 1998. Configurable products - Lessons learned from the Finnish Industry. In *Proceedings of 2nd International Conference on Engineering Design and Automation*. Integrated Technology Systems, Inc.

Ulrich, K., and S Eppinger. 1995. *Product Design and Development*. McGraw-Hill, Inc.

Weigel, Rainer, and Boi Faltings. 1996. Abstraction Techniques for Configuration Systems. In *Configuration—papers from the 1996 AAAI Fall Symposium*, ed. Boi Faltings and Eugene Freuder, 55-60. AAAI Press.

The Author's Contribution to the Publications

For publications I and II the author collaborated with the first author in the development of all the aspects of the framework, the survey of ten companies and the analysis of configurable products and configuration related processes. The first author was the principal developer of the framework and lead the survey. Especially configuration knowledge and sales-delivery processes of configurable products were analyzed by the author of this work. The development of configurable products and long-term management of configuration models and configurations were analyzed and surveyed to somewhat lesser extent by the author of this work.

The author was the principal developer of the configuration ontology in publication III. The second author collaborated in the development of the ideas presented in publication III to some extent and the other authors reviewed and polished the ontology.

The configuration aspect of the formal model and its relations to CSP and DCSP presented in publication IV were principally developed by the author, while the connection to non-monotonic reasoning and logic programming was analyzed in collaboration with the second author. The complexity analysis, modularity investigation and implementation were principally done by the second author.

Publication I

Publication II

Publication III

Publication IV