Product Configuration Systems: State of the Art, Conceptualization and Extensions

Thorsten Blecker and Nizar Abdelkafi and Gerold Kreutler and Gerhard Friedrich

University of Klagenfurt

2004

Product Configuration Systems: State-of-the-Art, Conceptualization and Extensions

Thorsten Blecker[*], Nizar Abdelkafi[*], Gerold Kreuter[**], Gerhard Friedrich[**]

[*] *Department of Production/Operations Management, University of Klagenfurt, Universitätsstrasse 65-67, A 9020 Klagenfurt, Austria*

blecker@ieee.org, nizar.abdelkafi@uni-klu.ac.at

[**] *Computer Science and Manufacturing, University of Klagenfurt, Universitätsstrasse 65-67, A 9020 Klagenfurt, Austria*

{gerhard.friedrich, gerold.kreutler}@uni-klu.ac.at

ABSTRACT - *Product configurators are considered to be among the most successful applications of artificial intelligence technology. In this paper, we determine different conceptualizations of configurators and condense them in a comprehensive morphological box, which should support configurator designers as well as decision makers in selecting the right system. The analysis of the criteria according to which configurators that are designed thus far reveals a neglect of the front-end perspective. Therefore, it is relevant to extend configurators with a front-end component assisting customers during product configuration through advisory. We develop a framework describing the main requirements on an advisory system and propose the technical infrastructure for its implementation. Finally, the advisory system and the configurator are integrated into a comprehensive interaction system.*

KEY WORDS: *product configurators, advisory system, product personalization*

Blecker, Th./Abdelkafi, N./Kreutler, G./Friedrich, G.: Product Configuration Systems: State of the Art, Conceptualization and Extensions, in: Hamadou, A. B./Gargouri, F./Jmaiel, M. (Hrsg.): Génie logiciel & Intelligence artificielle. Eight Maghrebian Conference on Software Engineering and Artificial Intelligence (MCSEAI 2004), Sousse/Tunisia, 9 - 12, May 2004, Centre de Publication Universitaire, Tunis 2004, S. 25 - 36

1. INTRODUCTION

Product configuration systems or configurators are important enablers of the mass customization (Pine 93) paradigm. They are considered to be among the most successful applications of artificial intelligence technology.

Configurators can be implemented at the interface between a supplier and its customers over the Internet in order to support the configuration task. Given a set of customer requirements and a product family description, the configuration task is to find a valid and completely specified product structure among all alternatives that the generic structure describes (Sabin *et al.,* 98). In this context, customers are provided with the possibility to alter a basic product and to graphically visualize the effects of these changes (e.g. http://www.customatix.com/).

2. DIFFERENT CONCEPTUALIZATIONS OF PRODUCT CONFIGURATORS

The artificial intelligence community generally addresses a software tool when speaking about configurators. (Bourke 00) defines a product configurator as „…software modules with logic capabilities to create, maintain, and use electronic product models that allow complete definition of all possible product option and variation combinations, with a minimum of data entries and maintenance". The main technical component of the configurator is the knowledge base which consists of the database and configuration logic. Whereas the database contains the total set of component types and their instances, the configuration logic specifies the set of restrictions on how components can be combined. In the following, different classifications of product configurators are presented.

– **Classification according to the configuration knowledge**

The conceptualizations of configuration knowledge can be classified to (a) rule-based, (b) model-based and (c) case-based approaches. Each approach relies on a different ontology that is required to represent the domain knowledge and describe the object types (classes) and the relations among object instances (Sabin *et al.,* 98).

Rule-based approach

Rule-based configurators work by executing rules with the following form: "if condition then consequence". The product solutions are derived in a forward-chaining manner. At each step, the system examines the entire set of rules and considers only the set of rules that can be executed next. Furthermore, there is no separation between directed relationships and actions. Thus, rules contain both the domain knowledge such as compatibilities between components and the control strategy that is necessary to compute the solution to a specific configuration problem (Sabin *et al.,* 98). The main drawbacks of rule-based systems are ascribed to the problems encountered during knowledge acquisition, consistency checking, knowledge maintenance, etc. (Guenter *et al.,* 99).

Model-based approach

The most important model-based representation types that are implemented within configurators are: logic-based, resource-based and constraint-based approaches (Sabin *et al.,* 98). The most prominent family of logic-based approaches is based on description logic. Description logics are formalisms for representing and reasoning with knowledge. They are based on the notions of individuals (objects), concepts (unary predicates, classes), roles (binary relations) and constructors that allow complex concepts and roles to be built from atomic ones. The inference mechanism is based on subsumption. However, resource-based systems are based upon a producer-consumer model of the configuration task. Each technical entity is characterized by the amount of resources it supplies, uses and consumes. A product configuration is acceptable when a resource balancing is realized (Juengst *et al.,* 98). In constraint-based reasoning components are defined by a set of properties and a set of connection ports. Constraints among components restrict the ways components can be combined (Tsang 93). A restriction can forbid a combination of parts (Part A cannot be combined with part B) or can require a specific combination (Part A requires part C).

Case-based approach

Case-based reasoning takes a different view from rule-based and model-based approaches. It relies on the assumption that similar problems have similar solutions. The knowledge necessary for reasoning consists of cases that record a set of product configurations sold to earlier customers. The current configuration problem is solved by finding and adapting a previous solution to a similar problem.

– **Classification according to the business strategy**

From the point of view of mass customization, three main strategies with different requirements on configurators are distinguished, namely assemble-to-order, fabricate-to-order and engineer-to-order. The assemble-to-order concept enables customers to configure a product by combining a finite number of standard modules. However, fabricate-to-order and engineer-to-order may assume an infinite number of configuration possibilities. The technical realization of configurators for fabricate- and engineer-to order is more demanding than those for assemble-to-order because a parameterization of component dimensions should be made possible.

– **Classification according to organization**

The organization of a configurator can be either central or distributed. A central configurator works locally and its configuration knowledge is completely stored in one unique system. All potential product instances that may represent a solution to the customer configuration problem are derived from this local data. However, the knowledge base of a distributed configurator is locally incomplete. It is integrated with other configurators (e.g. suppliers' configurators) in order to generate consistent product instances for specific customer requirements.

– **Internal vs. external configurators**

Internal configurators are only implemented for a company's internal use. For example, internal configurators support sales' experts in capturing a customer's requirements and translating them into technical features without errors. However, external configurators are designed to provide customers with a direct assistance during product configuration. They are equipped with front-end interfaces to facilitate the configuration task for customers.

– **Classification of configurators according to the interaction nature**

Configurators can be classified according to the nature of interaction which can be either offline or online. Offline configurators work independently from networks. The necessary data for configuration is stored on a data carrier such as a floppy disk, CD-ROM or DVD-ROM. After product configuration, customers can send the specifications via e.g., e-mail or fax. However, online configurators enable a communication with customers over the web. The configuration knowledge is stored on a central web-server. Therefore, the knowledge base can be efficiently updated. Online configurators can be further divided into two categories: online configurators with central data processing and online configurators with local data processing. Online configurators with local data processing require the load of the configuration application (Java Applets, Full Java Applications) onto the customer's local unit. However, configurators with central data processing are characterized by continuous communication between the supplier's central unit and the customer's local unit.

– **Classification of configurators according to the updates' execution**

The updates' execution can be either push or pull. A push mode is realized when the supplier's central unit containing the product configuration logic communicates product updates to the customer's local unit. In this mode, the central unit imposes the updates that have to be accepted by the local unit. In contrast, one speaks about a pull mode when the local unit retrieves the updates if required.

– **Classification of configurators according to the scope of use**

Configurators can be categorized as single-purpose and general-purpose systems. A single-purpose system is developed to support the sales-delivery process of a product or a set of products of only one company or business field. Single-purpose configurators are called special-purpose configurators and may be designed for a particular industry such as e.g., window and door industry. However, general-purpose systems are used to configure diverse product types in different companies (Tiihonen *et al.*, 97).

– **Classification of configurators according to their complexity**

Product configurators can be classified according to their design complexity. (Tiihonen *et al.,* 97) distinguish between primitive, interactive and automatic configurators. Primitive configurators are the simplest ones. They merely record the configuration decisions made by the user without checking the validity of decisions. However, interactive configurators are capable of checking as to whether the configuration decisions are valid. They also guide users in making all of the necessary decisions. In addition to the functionalities of interactive configurators, automatic ones are able to provide full support and to automatically generate parts or even entire configurations.

– **Classification of configurators according to their integration level**

At the integration level, one can distinguish between stand-alone, data-integrative and application-integrative configurators. Stand-alone configurators cannot be integrated because they do not dispose of interfaces to other information systems. Data-integrative configurators enable one to avoid data redundancy. However, the application-integrative configurators enable the integration of whole applications. For example, when configurator and CAD-system are integrated, drawings of new parts or components can be automatically generated for customers.

– **Classification of configurators according to the solution searching approach**

There are two main solution-searching approaches: either by technical elements or by features. Searching by technical elements means that the configurator enables customers to start from a standard product and then to specify step-by-step product options. However, a configurator working by features provides customers with the possibility to specify their requirements in terms of product functionalities. Then, the configurator searches for product variants that best fit the features specified by customers.

– **Classification of configurators according to their support of the product life cycle**

The product life cycle support refers to product reconfiguration that is necessary when the customer would like to upgrade the product by including new or better functionalities or to replace non-functioning parts or modules for which identical replacements no longer exist (Sabin *et al.*, 98). The different cases that can be encountered are: (a) configurator without reconfigurator, (b) separate configurator and reconfigurator and (c) integrated configurator and reconfigurator.

3. CONFIGURATORS' MORPHOLOGICAL BOX AND MAIN DRAWBACKS OF CONFIGURATORS

The morphological box was introduced for the first time as an efficient tool for creativity and structuring of ideas by (Zwicky 66). The main advantage of morphological boxes is that they present in a straightforward manner all of the possible solution alternatives for a specific problem. Therefore, we present all of the results of the configurators' classification in a morphological box (figure 1). This model should provide software engineers and developers with the main dimensions to be considered when designing a configurator. The decisions to be made relate essentially to the values to be taken by each dimension. The zigzag line of figure 1 shows the relevant characteristics of an example of configurator software with respect to each dimension.

The morphological box reveals that the customer perspective is technically not strongly considered. The technical aspects addressing how it is able to lead customers in a fast-paced manner and with a low amount of effort to their optimal choice are not considered.

Knowledge base	Rule-based	Model-based	Case-based
Strategy	Assemble-to-order	Fabricate-to-order	Engineer-to-order
Organization	Central		Distributed
Internal/external	Internal		External
Interaction nature	Online central data processing	Online local data processing	Offline
Updates' execution	push		pull
Scope of use	Single-purpose		General purpose
Complexity	Primitive	Interactive	Automatic
Integration level	Stand-alone	Data-integrative	Application-integrative
Solution searching approach	Technical elements		Features
Product life cycle support	Configurator without reconfigurator	Separate configurator and reconfigurator	Integrated configurator and reconfigurator

Figure 1. *Morphological box: classification of configurators*

In the technical literature, configurators are more often criticized. For example, (Rogoll *et al.,* 02) have shown through a market study on configurators that there is no standard software solution that is able to fulfill optimal requirements from the supplier and customer's perspectives. (von Hippel 01) criticizes the implemented configurators in the automobile industry and points out that "…auto makers allow customers to select a range of options for their "custom" cars - but they do not offer the customer a way to learn during the design process or before buying". Learning during the design process means that customers should be provided with the possibility to verify, before placing their buying orders, as to whether the configured product meets their expectations exactly or not.

Designers of configurators considerably concentrate on the back-end technical aspects and neglect the customer perspective. Especially in the business-to-consumer field, customers generally do not have sufficient product expertise. They cannot express their preferences in terms of technical specifications. Therefore, we argue that it is relevant to better assist customers during configuration in order to help them find satisfying product variants. The next section focuses on the extension of configurators with a front-end component capable of leading customers to their optimal choices. So, we speak of an interaction system that consists of two main components, namely a configurator and a customer-oriented front-end component called an advisory system.

4. Advisory Systems as Front-End Extensions for Configurators

4.1 Main Requirements for Advisory Systems

We define advisory systems as front-end software systems that guide customers according to their profiles and preferences through a personalized consulting process resulting in the generation of product configurations that better fulfill customers' needs. In opposition to the commonly used product oriented interfaces in configurators, advisory systems are customer oriented and do not assume any specific technical knowledge of the product. To determine the main capabilities of an advisory system, the problems that may occur during interaction should be examined. (Blecker *et al.,* 03) mention three faults that may occur during interaction, namely when:

– The customers do not know their real needs.

– The customers cannot correctly express their real needs.

– The supplier wrongly interprets customer requirements.

In order to tackle these problems, two main levers are identified: (a) the dialogs with customers and (b) the mapping techniques which permit a translation of customer needs into product specifications and vice versa. These levers should be supported by adequate technologies and tools. Whereas the main technology is web mining, the relevant tools are customer interests' modeling and web metrics. Both tools aggregate the data provided by web mining in order to present them in an understandable goal-oriented form (Figure 2).

Dialogs with customers refer to the communication interface during interaction. Customers should not perceive complexity when specifying their requirements. When customers do not know their real needs, dialogs can be conceived according to *Kansei Engineering* which uses verbal language for requirement elicitation. It is a "translating technology of consumer's feeling and image for a product into design elements" (Nagamachi 95, p. 2).

Moreover, in order to simplify the interaction process, the advisory system has to guide customers to their optimal choices by following the shortest path. This is referred to as the interaction *process simplification*. If some product parameters are too difficult, the advisory system can set default values without asking questions.

In addition to process simplification, *personalization* is relevant when customers cannot correctly express their real needs. Personalization is based on data gained about customers. It aims at recognizing special customer characteristics such as desires or preferences to individualize the interaction process, e.g. by adapting the website layout and/or the formulation of the customer dialogs.

Note that dialogs are unsuitable to solve the problem arising when the supplier wrongly interprets the customer requirements, because dialogs in fact do determine the communication process, but not the way information is interpreted.

The captured customer requirements during the interaction process have to be correctly translated into product specifications. This is ensured by the mapping techniques that not only adequately transform customer preferences and requirements into product specific characteristics (*filtering*) but also guarantee that product specifications are adequately mapped to customer needs (*validation*).

When customers do not know or cannot express their real needs, filtering methods are suitable solutions. Filtering is a collective term for techniques that automatically select product attributes which meet customer profile and preferences by applying predefined rules, similarities or clustering. As opposed to filtering, validation methods ensure that the supplier did not wrongly interpret customer needs. Thus, the restricted solution space resulting from filtering can be further refined to make sure that the product specifications really correspond to customer requirements. For example, a clustering component can be used for validation.

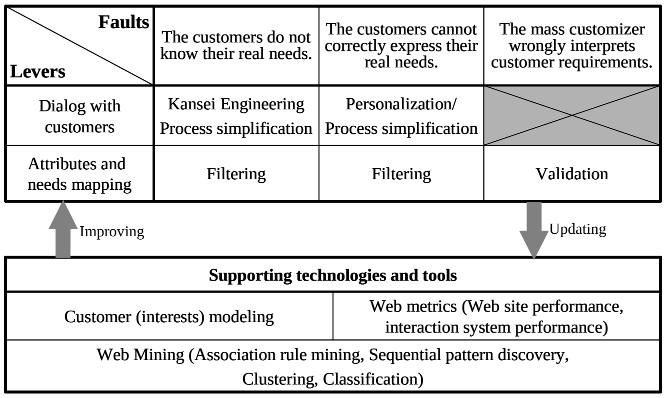

Figure 2. *Levers and supporting technologies and tools for an advisory system*

The described potential solutions are supported by web mining that aims at processing the raw data being stored in web server logs in order to extract statistical information, cluster users into groups and discover correlations between web pages and user groups. Web mining provides the information necessary to model customer interests and to compute relevant web metrics.

Customer interests' modeling is a tool enabling one to better understand customer preferences and thus, to correspondingly support the personalization of dialogs during the advisory process. Web metrics are necessary to measure the performance of the entire website and especially the performance of the interaction system, for instance how long one customer has spent on a certain web page. From the web metrics analysis, proposals for process simplification can be derived.

Recapitulating, with the proposed levers and supporting techniques and tools it is possible to cope with the described problems arising in the supplier-customer-interaction. The advisory system initiates a virtuous circle, which is ensured by a learning process consisting of continuous improvement of the presented solutions and an updating of the data processed by web mining techniques. Consequently, this leads to better product proposals and a better fulfillment of customer requirements.

4.2 Technical Implementation of the Advisory System

The main technical components being relevant for the implementation of the knowledge-based advisory system are depicted in figure 3. All relevant data for the advisory, i.e. the logic representing experts' knowledge, information for personalization as well as product and customer characteristics, are stored in a central *repository*. The knowledge is maintained by the expert through a graphical *knowledge acquisition component* which allows the expert to intuitively model the relevant knowledge. Then, a web-based, *graphical user interface* (GUI) is automatically generated by the *generation module*.

During the advisory process, the user communicates with the *dialog component* via the generated HTML pages. This component handles the user inputs and interfaces the *advisory component* which carries out the advisory on the basis of the stored knowledge. These components are supported online by a *CRM system* managing general preferences and properties of customers. Furthermore, the system relies on data stored in a web mining *data warehouse* where the web log files gained from the web server, as well as the data from the CRM system are processed in order to derive customers' models and web metrics.

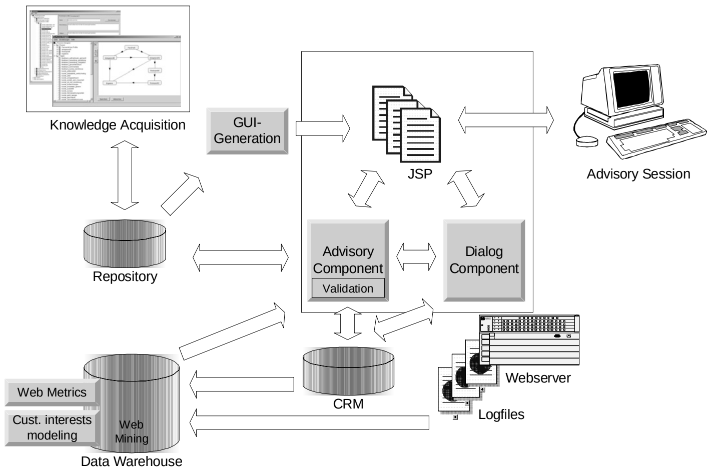

Figure 3. *Advisory system technical structure*

The personalization task is based on customer data already existing in a CRM system. As an example, we can analyze as to whether the customer is in general interested in a low price or premium products, and we can then adapt the interaction flow for a specific advisory session, for instance the content or the presentation style. The overall goal lies in simplifying the process and shortening the dialog.

Regarding the filtering techniques applied, we propose the adoption of a multi-technique approach where the advisory component automatically selects one specific recommendation technique such as collaborative or content based filtering, depending on the customer's knowledge and the product domain. Collaborative techniques are suitable when the match between customer preferences and products is based on quality and taste. On the other hand, content-based approaches are suitable for customers with sufficient technical knowledge. Finally, case or critique based techniques allow the customers to express their preferences by comparing and rating product alternatives in an intuitive way. Consequently, these mapping techniques incorporate information deduced by web mining-techniques in the external data warehouse. Besides mapping tasks, the results of such a data mining process can be used for personalization (e.g. by clustering users) or for deducing new business rules from customers' past behaviour.

5. Interfacing the Advisory System with a Configurator

In this section, we propose to interface the advisory system with the configurator. As previously described, the customer interface is very important and its technical structure can be complex. Therefore, it is advantageous to consider the advisory system as an independent software system. The configurator contains the product model, whereas the advisory system takes over the consulting role.

It is noteworthy that common configurators are equipped with customer interfaces just to make the product model accessible to customers. The user interface reflects the internal logic of the configurator. However, by means of an interface between the advisory system and the configurator, it is possible to decouple both systems. The advisory system proposes product configurations that would best fulfill customer needs. Consequently, the representation method of the knowledge base will not affect the advisory process. Thus, the method with which the product is modeled is a decision that can be made without considering the customer's perspective. For example, when the product has modular product architecture, it is advantageous to implement a resource-based configurator. This representation may be not optimal for the customer who has not enough technical product knowledge. However, the advisory system is capable of interacting with customers in a non-technical language and translates their needs into technical product specifications.

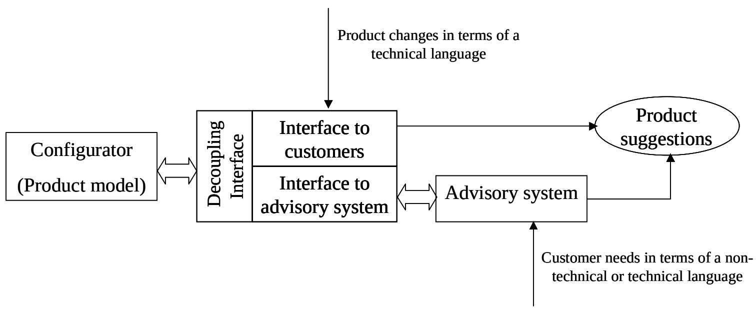

Figure 4. *Conceptual architecture of the interaction system*

The decoupling interface provides an abstraction layer between the configurator and the advisory system and ensures that both systems work as a unit that constitutes the whole interaction system. However, it can happen that the suggested product alternatives do not exactly correspond to a customer's needs because he prefers to introduce some changes to the proposed product variants. Therefore, the interface should enable customers to have direct access to the configurator in order to change the product's technical specifications. Schematically, this is described in Figure 4.

From a technical perspective, the decoupling interface provides an independent layer between both systems providing a well-defined interface. With regard to the proposed advisory system that is already implemented in a stand-alone version, following communication steps are conceivable:

1. Initially, the product attributes contained in the configurator's product model must be exported to the repository of the advisory system and additionally annotated. This is relevant for the advisory process that captures customers' preferences and maps them to product specifications.

2. Data collected during the advisory process provide input for configuration. The advisory system initiates configuration by transferring the product specifications to the configurator that accordingly creates valid product variants.

3. Finally, the configuration results are returned to the advisory system that presents the product suggestions to the user. In order to refine the product suggestions, the advisory system allows customers to re-execute step 2.

Note that communication scenario 1 is an asynchronous task. An implementation approach could be realized by XML based communication. In contrast, steps 2 and 3 are real-time. Therefore, remote function calls would be a suitable solution.

6. CONCLUSIONS

Different conceptualizations of configurators are proposed and condensed in a morphological box. Furthermore, it has been shown that the back-end perspective generally plays a more important role than the front-end perspective. Therefore, we introduced advisory systems to support customers during the configuration task. In order to tackle the faults that may occur during the interaction process, a framework outlining the levers and technologies was developed. Subsequently, the technical infrastructure that is needed to implement the advisory system is outlined. In order to ensure the integration of the advisory system and the configurator, a conceptual architecture of a comprehensive interaction system was proposed. However, the advisory system has some limitations. It reposes on some statistical methods requiring an amount of data only available after a certain time. Moreover, it is relevant to continuously adapt the content of dialogs, e.g. depending on the product life cycle. From a technical point of view, the approach is manageable, but its complexity should not to be underestimated, especially due to the high number of different interfaces.

7. REFERENCES

[Blecker et al. 03] Blecker, Th., Abdelkafi, N., Kaluza, B., Friedrich, G., “Key Metrics System for Variety Steering in Mass Customization”, in: *Proceedings of the 2nd Interdisciplinary World Congress on Mass Customization and Personalization* - MCPC’03, Munich, October 6-8, 2003.

[Bourke 00] Bourke, R., “Product Configurators: Key Enabler for Mass Customization – An overview”. Available online, URL: http://www.pdmic.com/articles/midrange/-Aug2000.html (Retrieval 10. Nov. 2003).

[Guenter et al. 99] Guenter, A., Kuehn, C., “Knowledge-Based Configuration – Survey and Future Directions – Technical report.”, Also available online, URL: http://www.hitec-hh.de/ueberuns/home/aguenter/literatur/xps-99.pdf (Retrieval 10. Nov. 2003).

[Juengst et al. 99] Juengst, W. E., Heinrich, M., “Using Resource Balancing to Configure Modular Systems”, *IEEE intelligent systems*, Vol. 13, No. 4, pp. 50-58.

[Nagamachi 95] Nagamachi, M., “Kansei Engineering: A new ergonomic consumer-oriented technology for product development”, *International Journal of Industrial Ergonomics*, Vol. 15, No. 1, pp. 3 - 11.

[Pine 93] Pine II, J., “Mass Customization: The New Frontier in Business Competition”, Boston: Harvard Business School Press 1993.

[Rogoll et al. 02] Rogoll, T., Piller, F. T., “Konfigurationssysteme fuer Mass Customization und Variantenproduktion”, Muenchen: ThinkConsult 2002.

[Sabin et al. 98] Sabin, D., Weigel, R., “Product Configuration Frameworks – A Survey”, *IEEE intelligent systems*, Vol. 13, No. 4, pp. 42-49.

[Soininen et al. 98] Soininen, T., Tiihonen, J., Männistö, T., Sulonen, R., “Towards a general ontology of configuration”, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, Vol. 12, No. 4, pp. 357-372.

[Tsang 93] Tsang, E., “Foundations of Constraint Satisfaction”, Academic Press, London, 1993.

[Tiihonen et al. 97] Tiihonen, J., Soininen, T., “Product Configurators – Information System Support for Configurable Products”, *Increasing Sales Productivity through the Use of Information Technology during the Sales Visit. A survey of the European Market*, Hewson Consulting Group.

[von Hippel 01] von Hippel, E., “User Toolkits for Innovation, MIT Sloan School of Management.”, Available on line, URL: http://ebusiness.mit.edu/research/papers/134%20vonhippel,%20Toolkits.pdf (Retrieval 10. Nov. 2003).

[Zwicky 66] Zwicky, F., “Entdecken, Erfinden, Forschen im Morphologischen Weltbild.”, Muenchen, Zuerich, 1966.