# Intelligent Interfaces for Distributed Web-based Product and Service Configuration

L. Ardissono[1], A. Felfernig[2], G. Friedrich[2], D. Jannach[2], R. Schäfer[3], and M. Zanker[2]

[1] Dipartimento di Informatica, Università di Torino, Italy
liliana@di.unito.it
[2] Computer Science and Manufacturing Research Group, University of Klagenfurt, Austria
{felfernig, friedrich, jannach, zanker}@ifit.uni-klu.ac.at
[3] DFKI, Saarbrücken, Germany
ralph.schaefer@dfki.de

**Abstract.** This paper emphasizes on the enhancement of web-based selling technology for complex products. The approach of the EC-funded CAWICOMS[1] project is twofold: provision of technologies both for customer-adaptive Web-interfaces for the configuration of mass-customized products as well as for the integration of configuration systems along the supply-chain. Within this paper we first motivate the demand for personalized and adaptive Web-interfaces of product configurators as an efficient means for customer relationship management. In addition, we sketch scenarios where product configuration takes place at several stages in the supply chain and the involved configuration systems have to cooperatively solve a distributed configuration task.

## 1 Introduction

Web-based product configuration tools enable businesses to market complex customizable products and services by using the new technologies of electronic commerce, whereby customers can tailor the configurable products according to their specific needs and requirements. In a Web-based environment special emphasis must be given to the customer interaction with the sales system, i.e., such a selling system should be personalizable and adapt to heterogeneous customer interests and skills. In addition, the digital economy of the 21st century will be based on flexibly integrated webs of highly specialized solution providers, therefore the joint product configuration of organizationally and geographically distributed providers must be supported. This requires the extension of current configuration technology to include distributed knowledge bases and co-operative problem solving behavior. Based on real-world

---

[1] *Customer-Adaptive Web Interface for the Configuration of products and services with Multiple Suppliers. The work takes place with the financial support of the IST Programme of the European Union under contract IST-1999-10688.*

application scenarios, the CAWICOMS project is carried out in order to overcome obvious shortcomings of current configuration technology.

For presentation purposes we will shortly sketch these requirements and our approach based on the guiding application scenario *configuration of telephone switching systems:* Telephone switching systems consist of modules plugged into frames, which are mounted on racks. Cables connect the modules and frames, resulting in a network topology imposed on top of the hierarchical physical structure. In addition, several external hardware components and subsystems such as PCs or routers are connected to the switching node. Further the functionality of the system depends on a set of software applications that are installed on the hardware. The whole system can be decomposed into subsystems supplied by different organizational units or independent companies.

## 2 Adaptive user interfaces and personalized interaction

When commercializing complex customizable products online, there may be various classes of users of the configurator that differ in properties such as skills, needs and knowledge levels. In current Web-based configurators there is typically only one standard interface with a predefined interaction style that cannot be tailored to these different types of users. As an example, adaptive configurator interfaces that tailor the system behavior according to the level of expertise of the user, are able to match these requirements. Experienced users may configure the product at some detailed technical level, whereas a novice user might only enter some high-level features and will have a need for more guidance and extra help during the configuration process.

An adaptive configuration tool will classify the current user with respect to different properties (e.g., expertise or interest in product characteristics such as reliability) and generate an interface that adapts flexibly according to this classification. We address this problem by customising the following aspects of the interaction:

a) *Adaptation of the configuration process:* It aims at reducing the communication overhead for the customer during the interaction with the configuration system. Therefore, reasoning on the content and the sequence of the set of required interaction steps is necessary.

b) *Adaptation of the presentation of configuration solutions:* This refers to the selection of the information most relevant to the user and the presentation of this information at a level of technical detail suited to his/her domain expertise.

These adaptation aspects rely on a user model representing the system's beliefs on the user characteristics. The customisation of the configuration process is based on the use of personalisation rules, represented within a rule-based system (ILOG JRules[2]). As the user interface is dynamically generated during the interaction, the level of detail addressed can be adapted to the most recent hypotheses about his/her knowledge and interests. For estimating the user's interests we ascribe *Multi-Attribute-Utility Theory (MAUT)* (see [5]) as evaluation process to the user. According to MAUT, the

[2] See ILOG (www.ilog.com) for reference.

configurable artifact can be evaluated as weighted addition of the evaluation with respect to its relevant *value dimensions.* For taking into account the uncertainty occurring in the interpretation of the user's behaviour, we use *Bayesian networks* (BNs) as a probabilistic inference mechanism (see [4]). For each interaction type, e.g., user self-assessment or various kinds of inputs during the configuration process, there is a BN interpreting the user's actions. For estimating the user's knowledgeability the approach of [3] is extended. There are BNs which interpret both actions which indicate that the user knows (resp. does not know) the implications of a parameter for the relevant dimensions, e.g. by selecting or changing a parameter value (resp. by using the help function of the system).

## 3 Distributed configuration

The driving application scenarios show that there does not exist a single business entity in the value chain of supplied goods and services that has complete pricing and product knowledge on the whole customer solution. Further this knowledge may only be partially shared among business partners for reasons of privacy and security. Therefore, we have to enable current configuration technology towards co-operative problem solving.

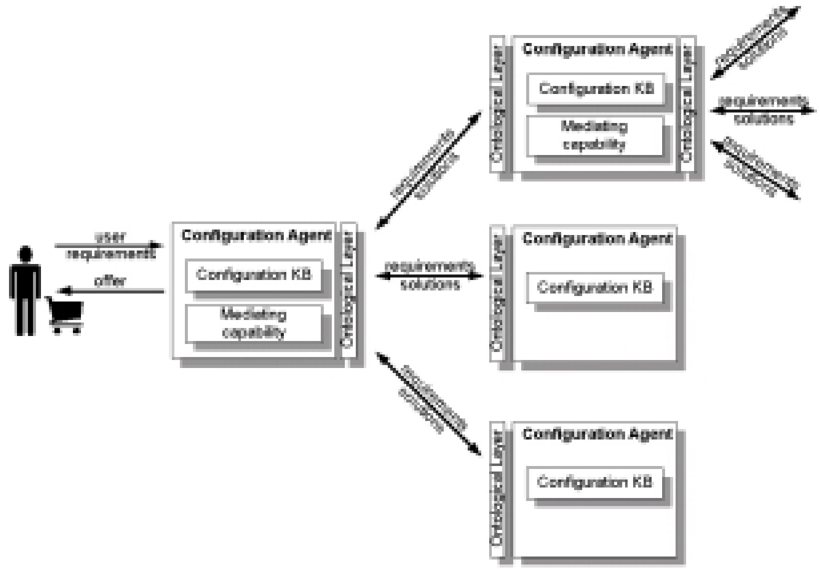


**Fig. 1:** Architectural sketch

As Figure 1 depicts, in our approach information sources as well as problem-solving agents with local knowledge (*Configuration KB*) are integrated. The value chain typically has a tree structure where each node is represented by a configuration agent that either represents the main vendor or one of the suppliers. Except for the leaves, all nodes of the tree possess *mediating capabilities* that allow them to decompose their configuration problem and assign subtasks to their supplying configuration systems.

In a realistic supply chain setting the involved configuration systems must be seen as legacy systems that have their proprietary knowledge representation mechanisms. Therefore, we employ an *ontological layer* based on a logic theory of configuration described in [1] that enables communication by mapping the specific representations onto more general ontological concepts from the configuration domain. Each configuration agent comprises the local knowledge necessary to customise a specific product or service of the company behind. These products or services that are part of the distributedly configured overall solution may share resources and have defined connection points vs. each other. Agents that have to observe restrictions that reference on not locally configured components need to have a limited view on these parts of the overall configuration solution. This view is provided to it by an agent with mediating capabilities. Further these mediating agents are responsible for taking measures for resolution in case of conflict occurrence. For more detailed information on co-operation mechanisms refer to [2].

## 4 The CAWICOMS environment

The outcome of the CAWICOMS project is an integrated environment supporting development, execution, and maintenance of (distributed) Web-based configuration applications. This environment consists of a set of components, which entail a set of improvements concerning the applicability in real world settings.

**Knowledge Sharing Support.** One of the major aims of CAWICOMS is the integration of heterogeneous configuration environments to support a distributed configuration process. A prerequisite for such a process is knowledge sharing between the engaged configuration systems. CAWICOMS provides a set of standardised XML Schema[3] definitions forming an ontology for distributed configuration. This ontology can be seen as a standard interchange format for configuration knowledge bases, which significantly reduces efforts of knowledge interchange.

**Distributed Problem Solving.** Beside an effective support for knowledge interchange between configuration environments supported by the knowledge acquisition component, CAWICOMS provides mechanisms for integrating those systems at the execution level. An ontological layer is imposed on each (remote) supplier configuration platform, which maps the generic configuration concepts onto the proprietary representation of the supplier system. Furthermore, a set of protocols implementing distributed problem solving algorithms (e.g., based on Distributed Constraint Satisfaction, see e.g., [6]) is supported that allow co-operative problem solving behaviour.

**Integration with existing Platforms.** The CAWICOMS environment supports seamless integration into existing e-commerce application platforms. Typical frameworks provide services like product catalogue management, shopping cart, customer

[3] See the World Wide Web Consortium (www.w3c.org) for reference.

management, procurement, purchase orders, payment transactions, and pricing. The CAWICOMS architecture relies on these services provided by the underlying layer. By providing a standardised schema for representing complex product structures and by integrating this schema in industrial standard Business Communication Languages (e.g. cXML, see www.cxml.org for reference), CAWICOMS supports the extension of basic framework functionalities with additional support for distributed and personalised configuration.

**Improved Knowledge Acquisition.** Due to the increasing size and complexity of configuration knowledge bases an effective design and maintenance support for configuration knowledge bases is required. In order to offer a more user-oriented knowledge acquisition process, the configuration knowledge is represented in UML (Unified Modeling Language) – the corresponding constraints are represented in OCL (Object Constraint Language). The major advantage of applying those languages in the configuration context is that they are comprehensible for a large community of potential users and are adopted in established industrial software development processes. As a consequence of the approach of [1] the application of configuration systems is no more restricted to specialists with corresponding knowledge in the area of formal description languages (basic representation languages of the underlying configuration systems).

**Standard Components.** For the implementation of the CAWICOMS prototype state-of-the-art Internet technologies are applied (Java Server Pages, Enterprise JavaBeans). All components of the prototype are implemented within a three-tier architecture conformant to J2EE (Java 2 Enterprise Edition).

For further information see http://www.cawicoms.org.

## References


1. Felfernig, A., Friedrich, G. and Jannach, D. (2000). *UML as domain-specific language for the construction of knowledge-based configuration systems*. International Journal of Software Engineering and Knowledge Engineering (IJSEKE), vol. 10 (4), pp. 449-469.
2. Felfernig, A., Friedrich, G., Jannach, D. and Zanker M. (2001). *Towards distributed configuration*. Proc: Joint German/Austrian Conference on Artificial Intelligence, 24th German / 9th Austrian Conference on Artificial Intelligence, KI-2001, Vienna.
3. Jameson, A. (1990). *Knowing What Others Know – Studies in Intuitive Psychometrics.* PhD thesis, University of Amsterdam, Netherlands.
4. Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* San Mateo, CA. Morgan Kaufmann.
5. Winterfeld, D. von and Edwards, W. (1986). *Decision analysis and behavioral research*. Cambridge, England. Cambridge University Press.
6. Yokoo, M., Durfee, E., Ishida, T. and Kuwabara, K. (1992). *Distributed Constraint Satisfaction for Formalizing Distributed Problem Solving*. $12^{th}$ IEEE Int. Conference on Distributed Computing Systems, pp. 614-621.