9. Web-Based Commerce in Complex Products and Services with Multiple Suppliers

Liliana Ardissono,[1] Alexander Felfernig,[2] Gerhard Friedrich,[2] Anna Goy,[1] Dietmar Jannach,[2] Ralph Schäfer,[3] and Markus Zanker[1]

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

Abstract

The sale of customisable products and services over the Internet is a challenging task within the area of electronic commerce, as it represents an important extension of the functionalities of electronic catalogues and Web stores, which normally do not support the dynamic configuration of the items to be purchased. In this chapter we shall present a case study which shows how the offering and selling of complex products and services from the telecommunication industry are supported within a generic framework for customer-adaptive distributed online configuration. Following the paradigm of mass customisation, products and services are nowadays sold to customers in many variants according to specific customer requirements. In a Web-based environment, the customer interaction with the sales system must be given special emphasis. Therefore we sketch how a personalised Web interaction may imitate a good salesperson who adapts expert advice according to the customer's interests and skills. The digital economy of the 21st century will be based on flexibly integrated webs of highly specialised solution providers. Regarding configuration technology itself, the joint configuration of organisationally and geographically distributed products and services must be supported. This requires the extension of current configuration technology to include distributed knowledge bases and co-operative problem-solving behaviour. The framework developed here is designed to be generic enough to be also applicable to other industries with similar requirements for electronic commerce systems, such as the areas of facility management equipment and the building and construction industry.

9.1 Introduction

When complex products and services are commercialised over the Web, the shortcomings of current technology become obvious. Configurators are employed to calculate product variants that fulfil customer requirements as well as technical and non-technical constraints on product solutions. However, Web-based commerce places additional requirements on the interaction with configuration systems: customers with different needs, skill levels or organisational backgrounds interact with the system. Therefore interfaces must be provided that dynamically adapt to the needs of the people facing them.

Another shortcoming of current configuration technology concerns the cooperation between separate configurators. As the application scenarios will show, there is no central point of knowledge and therefore a single-configurator approach is not appropriate. In addition to the distribution of the knowledge about configuration of products and services we have to accept the fact of heterogeneity among the knowledge representation formalisms employed.

In order to address the shortcomings of current configuration systems, we have developed an infrastructure for the creation of Web-based, user-adaptive configuration systems which can perform distributed configuration of products and services by interacting with remote suppliers. This infrastructure is a result of activities carried on in the ongoing IST research project CAWICOMS.[1]

This chapter is organised as follows: the guiding application scenarios are outlined in the next section, followed by a discussion of adaptive Web interaction and distributed product configuration. The environment of the CAWICOMS project is then described.

9.2 Application Scenarios and Architecture Overview

The guiding application scenarios both originate from the domain of telecommunication (cf. [4]): the configuration, ordering and provision of telecommunication switches (e.g. a TeCom) and the configuration of Internet protocol - virtual private networks (IP-VPNs).

Telephone switching systems consist of modules plugged into frames that 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 organisational units or independent companies.

An IP-VPN links a number of sites of a possibly multinational organisation via an IP network. An integrator/reseller company contracts with the customer and leads a federation of partnering network and other service providers. The layout of the network service has to be determined according to the geographic location of the different sites and the qualitative requirements with regards to bandwidth,

[1] CAWICOMS is an acronym for "customer-adaptive Web interface for the configuration of products and services with multiple suppliers". Specific information about this project can be found at the following URL: http://www.cawicoms.org.

quality of service and cost limits. This task requires knowledge about the characteristics and availability of the services offered by each provider involved.

In both scenarios the considered business process is roughly structured into the following phases:

– *Quotation phase*: Elicitation of basic requirements from the customer to provide an estimate of the capabilities, availability and price of the system or service. This can be accomplished solely by the Web interface or in conjunction with a physical sales representative.
– *Order generation phase*: In this phase, a technical engineer derives the bill of materials and the layout of the physical product or the structure of the service with the help of the configuration system. Here, decisions on alternative methods of implementation for specific required features are taken, and the order for production or accomplishment of services is generated.
– *Production phase*: Support for this phase and the following phases is achieved through traditional systems (e.g. enterprise resource planning systems) and is outside the scope of this chapter.

A requirements analysis, performed for both scenarios, brought up issues that are not addressed by current configuration technology. These issues can be grouped into system interaction and configuration technology issues:

– *Requirements on system interaction*: Owing to the complexity of the application domains, support for the user during the interaction is crucial. Therefore, the interaction has to be personalised according to the user's skills and needs. For example, during the requirements elicitation phase, the system has to provide the user with reasonable default values and explanations of parameters. During the presentation and feedback phase, the output should be adapted to the user's interests. For example, information which is very interesting to the user should be highlighted.
– *Requirements on configuration technology*: The most innovative aspect of these scenarios is the distribution of configuration knowledge and problem solving-capability. This requirement does not stem from efficiency considerations, but is implied by the presence of several business entities along the value chain, and the sales process itself. The offering of customised, extensive solutions to problems requires the flexible cooperation of several product and service providers. Each of these providers owns and maintains locally the knowledge necessary to configure its contribution to the overall solution.

Figure 1 depicts the overall rationale of the CAWICOMS framework. The user interacts with the configuration system via a standard Web-browser. On the basis of a *model* of the current user (an estimate of the user's properties and interests), a personalised user interface is generated. The user requirements and the configuration results are transmitted back and forth to the *main configurator*, a facilitating component that integrates parts of the product models of the suppliers involved. This integration is done on a set of common ontologies for the domain of product configuration. At run time, the main configurator forwards the actual user requests to the corresponding configuration systems of the suppliers. The openness of the framework is guaranteed by using an open XML-based protocol for data exchange between the heterogeneous configuration systems. Therefore, existing applications can be integrated by implementing adapter components for the legacy systems that map the internal knowledge representation to the common language for run time data exchange. Finally, since in most cases configuration applications are not stand-alone applications but rather are embedded into an underlying business-to-business/enterprise resource planning (B2B/ERP) system providing *standard* functionalities, e.g. for order processing, adequate interfaces to these platforms are supported.

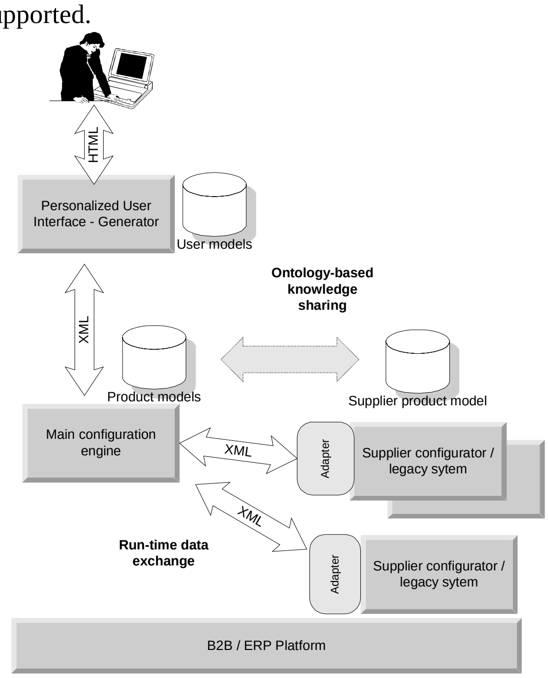

Fig. 1. Architectural sketch of the CAWICOMS framework

9.3 Adaptive Web Interaction

As noticed by Benyon in [2], the user interfaces of software systems suffer from a general problem: Owing to the high variability in user requirements, no interface can be designed which suits everybody, and there is a need for dynamic user interfaces that tailor the interaction style to the individual user. With the advent of Web-based services, such as search engines and electronic catalogues, Benyon's argument has become extremely relevant, as these systems are designed to be used by a very heterogeneous customer base, without relying on the assistance of technicians and domain experts. Thus, personalisation techniques have become central to the development of several electronic commerce systems, both at the academic and at the commercial level [1, 3, 5, 12].

In particular, dynamically generated, personalised interfaces are an essential requirement for Web-based configuration systems, as they are used by people who have different backgrounds and different requirements for the products/services to be configured. At present, these systems typically offer only a standard user interface and interaction style that cannot be tailored to the customers' needs and skills. Thus, their interfaces are either simple but cannot take advantage of all customisation possibilities, or are too complex and cannot be used by inexperienced customers. Moreover, current configuration systems cannot distinguish between those aspects of the configuration which need to be carried out in cooperation with the user and minor aspects which could be autonomously managed by the system, in order to reduce the user's effort in the configuration task. We have addressed these issues by customising the following aspects of the interaction:

- *Adaptation to the device used by the customer*, to support the interaction with standard browsers, graphical user interfaces and so forth.
- *Adaptation of the configuration process itself*, to reduce the overhead on the customer during the generation of configuration solutions. A non-personalised configuration process relies on the user for setting almost all the configuration parameters, which may turn into a long list of questions before a result can be presented. One main personalisation goal is therefore to reduce the list of questions, enabling the system to take the initiative and set as many parameters as it can without the user's intervention. The user's interests can be taken into account to anticipate the user's decisions whenever possible. Moreover, in order to assist the user during a configuration session, his/her skills can be taken into account to avoid difficult questions whenever the system can safely set the needed value. The user's expertise can also be taken into account when critical parameters must be set by the user. In that case, an explanation of the alternative values can be provided to support his/her decision.
- *Adaptation of the presentation of configuration solutions*, to select the information relevant to the evaluation of the solution and present such information at a technicality level suited to the user's domain expertise. Both aspects are essential for supporting the user's selection of the most suitable solution and the overall acceptability of the system's proposals, but the first one deserves some more comment. In particular, it should be noticed that, in non-trivial domains, configuration solutions are complex and full of details, only some of which represent relevant information from the viewpoint of the customer. Therefore, the system needs to be able to omit technical details that, although essential to the generation of a solution offering the functionalities desired by the user, do not represent relevant information about the properties of the resulting product/service.

These adaptation aspects rely on a user model that represents the system's beliefs about characteristics of the user. In the following, we shall describe the personalisation strategies for customising the interaction, focusing on the configuration process. The customisation of this process is based on the use of personalisation rules, represented within a rule-based system (ILOG JRules[2]). As

[2] See ILOG's Web pages (www.ilog.com) for reference.

the user interface is generated dynamically during the interaction, the level of detail addressed can be adapted to the most recent hypotheses about his/her knowledge and interests.

The configuration process is carried out, after the selection of the product/service needed by the customer, by asking him/her questions about configuration parameters and producing partial solutions, which may contain open parameters to be specified at a later stage of the process. At each step, given a partial solution, the system identifies the parameters that have not yet been set. It then invokes the rule-based system to identify the appropriate strategy for filling in the parameters. The personalisation rules discriminate among various alternatives, which either provide values for setting the parameters without questioning the user or suggest suitable questions to be asked, taking into account the user's expertise. Each rule corresponds to an alternative way to proceed and specifies, in its antecedent, the conditions (on parameter to be filled in and user's interests and expertise status) determining its own applicability.

The conditions specified in the antecedents of the rules are evaluated in order to determine which rule matches the current situation in the best way and represents the most promising strategy for carrying on the interaction. For example:

- One rule bases its own suggestions on the availability of individual defaults, i.e. default values explicitly selected by the user for the products and services he/she usually purchases (e.g. the quality-of-service level could represent an individual default, which the customer selects once and for all when interacting with the system). This rule suggests that, if an individual default is available in the user model to fill in a parameter, then the selected value(s) should be used to carry on the configuration process.
- Another rule specifies that personalised defaults, based on customer characteristics, can be exploited to set parameters. For instance, suppose that the instruction manuals for a product are available in English and in German. In this case, a personalised default could specify that German is the appropriate value if the customer is a German, while English is the appropriate value for all other nationalities.
- Yet another rule suggests that information about the user's interests and knowledge about the dependencies among such interests and the configuration parameters can be used to predict the user's choices and set the parameters. For example, if the user is interested in the reliability of a telecommunication switch, the system should propose the use of an additional power supply.
- In addition, there are rules suggesting that the system should ask the user a direct question about the parameter, or an indirect question concerning a more abstract concept related to the parameter. These strategies are applied when the parameter is too critical to be automatically set by the system, or when the estimates in the user model are too uncertain to support the use of defaults.

To support personalisation of the type described here, the user modelling component has to maintain user characteristics such as preferred language. In an idealisation, we assume that the user employs multi-attribute utility theory (MAUT) (see [18]) as an evaluation process. Based on this assumption, the user's interests are being estimated. MAUT is also used by consumer organisations for evaluating products. According to MAUT, an artefact can be evaluated by performing a weighted addition of the evaluations with respect to its relevant value dimensions. These are technical characteristics, such as reliability or economy. The evaluation of such a dimension is defined as a weighted sum of the evaluations of the attributes relevant to this dimension.

To take into account the uncertainty involved in the interpretation of the user's behaviour, we have used Bayesian networks (BNs) as a probabilistic inference mechanism (see [15]). For each interaction type, there is a BN interpreting the user's actions:

- The user performs a self-assessment regarding his/her interests (*self-characterisation*).
- At the beginning and also during the configuration process, the user has the option to specify an up-to-now unspecified parameter. This means that this parameter will most probably have an impact on the overall evaluation of the product. In addition, the user expects that the value chosen for the parameter will improve the product compared with the current configuration, i.e. that there will be a positive evaluation shift (*parameter setting*).
- When the system has configured a product, the product will be presented to the user, who has the choice of accepting or rejecting it (*judgement*).
- The user changes a parameter value of a solution that has been presented. This means that the user probably assumes that the configured product will be better than if the parameter is left unchanged (*improvement*).

To estimate the user's knowledgeability the approach of [13] has been extended. We use BNs to interpret actions which indicate that the user probably knows or does not know the implications of a parameter for the relevant dimensions. For example, if the user selects or changes a parameter value, s/he probably will know the implications of the parameter for its value dimensions. On the other hand, if the user requests an explanation of the parameter by using “help”, this indicates that the user probably does not know the implications.

9.4 Distributed Configuration

The application scenarios show that there does not exist a single business entity in the value chain of the goods and services supplied that has complete pricing and product knowledge about the whole customer solution. Furthermore, this knowledge may be shared only partially among business partners for reasons of privacy and security. Therefore, we have to extend current configuration technology towards cooperative problem solving. Our proposed architecture relates to previous research projects such as TSIMMIS [10] and Infomaster [11], where integrated access to multiple distributed, heterogeneous information sources on the Internet was provided.

As Figure 1 depicts, in our approach not only information sources but also problem-solving agents with local knowledge (*product models*) are integrated. The customer has the illusion of interacting with a centralised, homogeneous configuration system. The value chain has a tree structure, where each node is represented by a configuration agent that represents either the main vendor or one of the suppliers. Except for the “leaves”, all nodes of the tree have mediating capabilities that allow them to decompose their configuration problem and assign subtasks to the configuration systems supplying them. This is done by formulating requirements that must be met by the solutions, which are then communicated back in response.

In a realistic supply chain setting, the configuration systems involved 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 [6] that enables communication by mapping the specific representations onto more general ontological concepts from the configuration domain (compare [17]). Problem-solving itself can be compared to the work of [19], where the authors of that publication describe how several agents can build major subassemblies on their own, but these subassemblies must “hook together” in a compatible way. As a mechanism for problem representation, they propose a distributed Constraint Satisfaction Problem (CSP) and present several algorithms, such as Asynchronous Backtracking [19]. However, in contrast to distributed CSP solving, the cooperation among configuration agents has to take two major extensions into account:

- On the one hand, configuration tasks have a dynamic nature in the sense that the set of problem variables changes depending on the initial requirements and on the decisions taken during the problem-solving process.
- On the other hand, different approaches to configuring need to be supported. Several paradigms exist, such as rule-based systems, various forms of constraint satisfaction [9] or description logics [14].

Each configuration agent comprises the local knowledge necessary to customise a specific product or service of the company behind it. Those products or services that are part of the distributed, configured overall solution may share resources and have defined connection points with each other. Agents that have to observe restrictions that refer to non-locally configured components need to have a limited view of those parts of the overall configuration solution. This view is provided by an agent with mediating capabilities. These mediating agents are also responsible for taking measures for resolution in case of conflict. For more detailed information on cooperation mechanisms, see [8].

9.4.1 Example: Distributed Configuration of IP-VPNs

We shall now describe the application of our architecture to distributed configuration of IP-VPNs. An IP-VPN based on the public IP network and dedicated backbone lines can be used to extend a company's network with reliable, secure connections to remote offices, roaming users and business partners. In a typical business case, a company interested in an IP-VPN solution contracts with a reseller that offers network solution based on subcontracts with, for example, public telecommunication companies or Internet service providers. Configuration of an

IP-VPN solution is done in a two-phase process, which is sketched in Fig. 2. First, given a set of sites to be interconnected and the specific requirements, for example on bandwidth for the individual connections, a high-level design of the overall network layout is performed. This step involves the selection of an adequate set of access lines from the customer sites to the backbone network and of a set of sections within the backbone. Note that this step also includes the selection of corresponding suppliers for the individual services, since for each customer site there may be more than one possible provider. The high-level network design serves as the basis for the quotation phase.

In a second step, the detailed design of the network is done, i.e. the corresponding suppliers (supplier systems) are contacted in parallel and the specific customer requirements are forwarded. The remote systems of these suppliers can then compute technical details of the final configuration, e.g. IP-settings, internal routing or additionally required hardware such as network routers. Note that this step is typically performed using some specialised routing or optimisation software or by a legacy system.

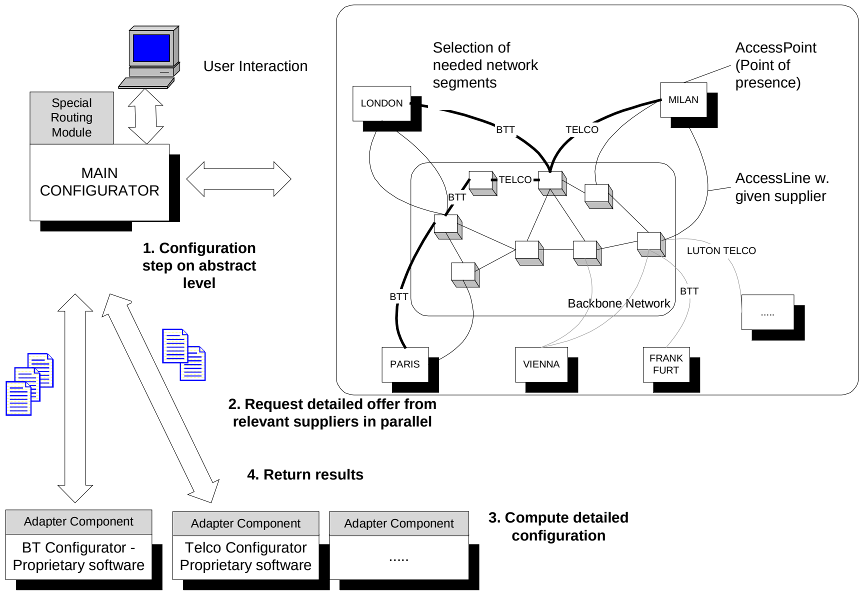

Fig. 2. Two-phase configuration scenario for IP-VPNs

The implementation of this scenario in our framework requires the following steps:

– Definition of a common ontology and vocabulary for the domain of IP-VPNs between the reseller and the suppliers involved. (By using UML (Unified Modeling Language) as ontology definition language.)

- “Advertisement” of concrete offerings (e.g. Internet service provision in some region) to the mediating agent at the reseller's site, where these offerings must conform to the common ontology. The sum of all these offers forms the high-level network known to the reseller.
- Implementation of “adapter components” for the suppliers' systems so that they are able to perform run-time data exchange (partial configuration solutions or notification that no solution is feasible) in the common XML-based protocol.

9.5 The CAWICOMS Environment

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

9.5.1 Knowledge-Sharing Support

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

9.5.2 Distributed Problem Solving

Besides 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 that allow cooperative problem-solving behaviour is supported. In the prototype implemented, ILOG's constraint-based JConfigurator[4] libraries are used as the problem solving mechanism in the mediating agent.

[3] See the World Wide Web consortium Web site (www.w3c.org) for reference.
[4] See ILOG's Web pages for more information (www.ilog.com).

9.5.3 Integration with Existing Platforms

The CAWICOMS environment supports seamless integration into existing e-commerce application platforms. Typical frameworks provide services such as product catalogue management, shopping carts, customer 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 into industrial standard business communication languages (e.g. cXML[5]), CAWICOMS supports the extension of basic framework functionalities with additional support for distributed and personalised configuration.

9.5.4 Improved Knowledge Acquisition

Owing to the increasing size and complexity of configuration knowledge bases, 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 and 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 to a large community of potential users and have been adopted in established industrial software development processes. As a consequence of the approach of [7] the application of configuration systems is no longer restricted to specialists with corresponding knowledge in the area of formal description languages (i.e., the basic representation languages of the underlying configuration systems).

9.5.5 Standard Components

State-of-the-art Internet technologies (servlets, Java server pages and Enterprise Java Beans) have been applied in the implementation of the CAWICOMS prototype. All components of the prototype have been implemented within a three-tier architecture conforming to the J2EE (Java 2 Enterprise Edition) specifications.

9.6. Conclusions

CAWICOMS aims at the next generation of electronic-commerce solutions for customisable products and services. Techniques developed for enabling integration and collaboration of distributed Web-based configurators, and for enabling adaptation and personalisation of user interaction with configurators.

[5] See Commerce XML Resources (www.cxml.org) for reference.

CAWICOMS will have benefits both for customers and for suppliers in the following ways: Personalisation will help consumers to better specify their needs and to select the most appropriate solution when buying complex goods and services over the Web. CAWICOMS also enables flexibly integrated webs of suppliers to cooperate along the supply chain by addressing the interoperability of product configuration systems.

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

Acknowledgements.

This work was partly funded by the European Union through the IST Programme under contract IST-1999-10688 CAWICOMS. We thank the other project members, especially Jonathan Mitchener, for their valuable comments.

References

1 Ardissono, L. and Goy, A. (2000): Tailoring the interaction with users in Web stores. User Modeling and User-Adapted Interaction, **10**(4), 251–303.

2 Benyon, D.R. (1993): Adaptive systems: a solution to usability problems. User Modeling and User-Adapted Interaction, **3**, 65–87.

3 Brusilovsky, P. (2001): Adaptive hypermedia. User Modeling and User-Adapted Interaction, **11**(1–2), 87–110.

4 CAWICOMS Consortium (2000): *Deliverable D01: Requirements, Application Scenarios, Overall Architecture, and Test Specification*, available via http://www.cawicoms.org.

5 Riecken, D. (ed.) (2000): Communications of the ACM, Special Issue on Personalization, **43**(8).

6 Felfernig, A., Friedrich, G., Jannach, D., and Stumptner M. (2000): Consistency-based diagnosis of configuration knowledge bases. *Proc. 14th European Conference on Artificial Intelligence*, Berlin, IOS Press, Amsterdam, 146–150.

7 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, **10**(4), 449–469.

8 Felfernig, A., Friedrich, G., Jannach, D., and Zanker M. (2001): Towards distributed configuration. In: Baader, F., Brewka, G., and Eiter, T. (Eds.): KI 2001: Advances in Artificial Intelligence, Lecture Notes in Artificial Intelligence 2174, Springer, Berlin, Heidelberg, 198-212.

9 Fleischanderl, G., Friedrich, G., Haselböck, A., Schreiner, H., and Stumptner, M. (1998): Configuring large systems using generative constraint satisfaction. IEEE Intelligent Systems, Special Issue on Configuration, **13**(4), pp. 59–68.

10 Garcia-Molina, H., Papakonstantinou, Y., Quass, D., Rajaraman, A., Sagiv, Y., Ullman, J., Vassalos, V., and Wisom, J. (1997): The TSIMMIS approach to mediation: data models and languages. Journal of Intelligent Information Systems, **8**(2), pp. 117–132.

11 Genesereth, M., Keller, A., and Duschka, O. (1997): Infomaster: an information integration System. 23rd ACM SIGMOD International Conference on Management of Data, 539–542.
12 Kobsa A., Koenemann J., and Pohl W. (2001): Personalized hypermedia presentation techniques for improving online customer relationships, The Knowledge Engineering Review, **16**(2), pp. 111–155.
13 Jameson, A. (1990): *Knowing What Others Know – Studies in Intuitive Psychometrics*. PhD thesis, University of Amsterdam.
14 McGuinness, D. and Wright, J. (1998): Conceptual Modeling for configuration: a description logic-based approach. Artificial Intelligence for Engineering, Design, Analysis and Manufacturing, **12**(4), pp. 333–344.
15 Pearl, J. (1988): *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, San Mateo, CA.
16 Sabin, D. and Weigel, R. (1998): Product configuration frameworks – a survey. IEEE Intelligent Systems, Special Issue on Configuration, **13**(4), pp. 50–58.
17 Soininen, T., Tiihonen, J., Männistö, T., and Sulonen, R. (1998): Towards a general ontology of configuration. Artificial Intelligence for Engineering, Design, Analysis and Manufacturing, **12**(4), pp. 357–372.
18 von Winterfeldt, D. and Edwards, W. (1986): *Decision Analysis and Behavioral Research*, Cambridge University Press, Cambridge.
19 Yokoo, M., Durfee, E., Ishida, T., and Kuwabara, K. (1992): Distributed constraint satisfaction for formalizing distributed problem solving, 12th IEEE International Conference on Distributed Computing Systems, pp. 614–621.