Web-based Commerce of Complex Products and Services with Multiple Suppliers

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

Abstract. The sales of customisable products and services over the internet is a challenging task within the area of electronic commerce. In this chapter we will present a case study which shows how the offering and selling of complex products and services from the telecommunication industry is 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 special emphasis must be given to the customer interaction with the sales system. Therefore we sketch how a personalised Web-interaction may imitate a good salesperson that adapts his expert advice according to the customers 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 developed framework is designed generic enough to be also applicable to other industries with similar requirements for electronic commerce systems such as the areas of facility management equipment or building and construction industry.

1. Introduction

When commercialising complex products and services over the Web shortcomings of current technology become obvious. Configurators are employed to calculate product variants fulfilling customer requirements as well as technical and non-technical constraints on product solutions. However, Web-based commerce poses additional *requirements on the interaction with configuration systems*: customers with different needs, skill level, or organisational background interact with the system. Therefore interfaces must be provided that dynamically adapt to the needs of their vis-à-vis.

Another shortcoming of current configuration technology concerns the *co-operation between separate configurators*. As the application scenarios will show, there is no central point of knowledge and therefore a single configurator approach is

---

[1] Dipartimento di Informatica, Università di Torino, Corso Svizzera 185, 10149 Torino, Italy, {liliana. goy}@di.unito.it
[2] Computer Science and Manufacturing Research Group, Universitaet Klagenfurt, Universitaetsstr. 65-67, 9020 KlagenfurtAustria {felfernig, friedrich, jannach, zanker}@ifit.uni-klu.ac.at
[3] DFKI GmbH, Stuhlsatzenhausweg 3, 66123 Saarbrücken, Germany, Ralph.Schaefer@dfki.de

not appropriate. In addition to the distribution of the knowledge on product and service configuration, we have to accept the fact of heterogeneity among the employed knowledge representation formalisms.

In this section we give our ideas on extending current configurator technology. First the guiding application scenarios are outlined, followed by a discussion on adaptive Web interaction and distributed product configuration. Then, the environment of the ongoing IST research project CAWICOMS[4] is described.

2. Application Scenarios

The guiding application scenarios stem both from the domain of telecommunication: the configuration, ordering and provision of *telecommunication switches* (e.g. a TeCom) and 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 multi-national organisation via an IP network. An integrator/reseller company contracts with the customer and leads a federation of partnering network and other service providers. According to the geographic location of the different sites and the qualitative requirements with regards to bandwidth, quality of service or cost limits the layout of the network service has to be determined. This task requires to access the knowledge on the characteristics and the availability of services offered by each involved provider.

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

?? *Quotation phase*: Elicitation of basic requirements of the customer to determine an estimate of the capabilities, the availability and the price of the system or service. This can be accomplished solely by the Web interface or together 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 implementation alternatives for specific required features are taken and the order for production or accomplishment of services is determined.

?? *Production phase*: Support for this phase and the following phases is achieved through traditional systems (e.g., enterprise resource planning systems) and out of scope of this paper.

[4] CAWICOMS is the acronym for *Customer-adaptive Web interface for the configuration of products and services with multiple suppliers*. This work was partly funded by the European Union through the IST Programm (contract IST-1999-10688).

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

Requirements on system interaction: Due to the complexity of the application domains support to the user during the interaction is crucial. Therefore, the interaction has to be personalized according to the user's skills and needs. For example, during the parameter value 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 of particular importance for the user due to her/his characteristics 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 setting of business entities along the value chain and the sales process itself. The offering of customized and extensive problem solutions requires the flexible cooperation of several product and service providers. Each of them locally owns and maintains the knowledge necessary to configure its contribution to the overall solution.

3. Adaptive Web Interaction

As noticed in Benyon (1993), the user interfaces of software systems suffer from a general problem: Due to the high variability in user requirements, no interface can be designed which suits everybody and there is a need for dynamic user interfaces tailoring the interaction style to the individual user. This issue is particularly relevant for configuration systems, used by people having different backgrounds and requirements on the products/services to be configured. Nowadays, configurators typically offer only a standard user interface and interaction style that cannot adapt to the customers' needs and skills. Thus, their interfaces are either simple, but cannot take advantage of all customisation possibilities; or they are too complex and cannot be used by inexperienced customers. We address this problem by customising the following aspects of the interaction: *Adaptation to the device used by the customer*, to support the interaction with standard browsers, GUI interfaces, and so forth. *Adaptation of the configuration process itself* aims at reducing 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 considered to avoid difficult questions whenever the system can safely set the needed value. The user's expertise can also be considered when critical parameters must be set and they are too complex for him/her. In that case, an explanation of the alternative values can be provided to support his/her decision. *Adaptation of the presentation of configuration solutions* concerns the selection of the type of information most relevant to the user and the presentation of such information at a technicality level suited to his/her domain expertise. This aspect is essential to support the user's selection of the most suitable solution and the overall acceptability of the system's proposals.

These adaptation aspects rely on a user model representing the system's beliefs on the user characteristics. In the following, we will 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[5]). 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.

The configuration process is carried on, 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. Then, it 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 and user's interests and expertise status) determining their own applicability. The conditions specified in the antecedents of the rules are evaluated in order to determine which rule best corresponds to the current situation and represents the most promising strategy to carry on the interaction. For example, one rule suggests that, if an *individual default* is available in the user model to fill in the parameter, then the selected value(s) should be exploited to carry on the configuration. Another rule specifies that *personalised defaults*, based on user characteristics, can be exploited to set parameters. For instance, suppose that the instruction manuals for a product are available in English and in German. Then, a personalised default could specify that German is the appropriate value, if the customer is a German, while English is the appropriate one for all the other nationalities. Yet another rule suggests that the information about the *user's interests* and the 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 reliability of a TeCom, the system should propose to use an additional power supply. Then, there are rules suggesting to *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 autonomously set by the system, or the estimates in the user model are too uncertain to support the use of defaults.

For supporting the described personalisation, the user modelling component has to maintain user characteristics such as preferred language.

For estimating the user's interest's we ascribe *Multi-Attribute-Utility Theory (MAUT)* (see Winterfeld & Edwards, 1986) as evaluation process to the user. According to MAUT, the configurable artifact can be evaluated as weighted addition

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

of the evaluation with respect to its relevant *value dimensions*. These are technical characteristics, such as *reliability*. The evaluation on such a dimension is defined as an evaluation of the attributes relevant for this dimension.

For taking into account the uncertainty involved in the interpretation of the user's behaviour, we use *Bayesian networks* (BNs) as a probabilistic inference mechanism (see Pearl, 1988). For each interaction type, there is a BN interpreting the user's actions.

a) The user gives a self-assessment regarding his/her interests (*self characterisation*).
b) 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 a big impact on the overall evaluation of the product. In addition, the user expects that the chosen value for the parameter will improve the product compared with the current configuration, i.e. that there will be a positive evaluation shift (*parameter setting*).
c) When the system has configured a product, the product will be presented to the user, who has the choice to accept or reject it (*judgement*).
d) The user changes a parameter value of a presented solution. This means that the user probably assumes that the configured product will be better than if the parameter is left unchanged (*improvement*).

Information gained during interactions of these four types is used for estimating the user's knowledgeability which extends the approach of Jameson (1990). We use 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 "help").

4. Distributed Configuration

The 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. Our proposed architecture relates to previous research projects such as TSIMMIS in Garcia-Molina et al. (1997) or Infomaster from Genesereth et al. (1997), where an integrated access to multiple distributed and heterogeneous information sources on the internet is provided. As Figure 1 depicts, in our approach not only information sources but problem-solving agents with local knowledge (*Configuration KB*) are integrated. The customer has the illusion of interacting with a centralised, homogenous configuration system. The value chain 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. This is done by formulating requirements that must be met by the solutions that are then communicated back in response.

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 Felfernig et al. (2000a) that enables communication by mapping the specific representations onto more general ontological concepts from the configuration domain (compare to Soininen et al. (1998)).

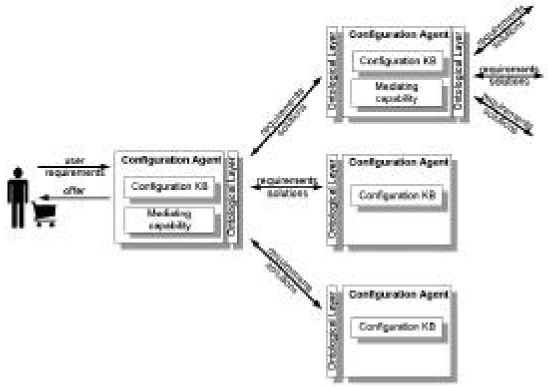

Figure 1: Architectural sketch

Problem solving itself can be compared to the work of Yokoo et al. (1992), where they describe how several agents can build major subassemblies on their own, but these subassemblies must "hook together" in a compatible way. As mechanism for problem representation they propose a distributed CSP and present several algorithms such as asynchronous backtracking. 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 towards configuring need to be supported. Several paradigms exist such as rule-based systems, various forms of constraint satisfaction (Fleischanderl et al. (1998)) or description logics (McGuiness and Wright (1998)).

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 Felfernig et al. (2001).

5. 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[6] 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 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 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[7]), 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 Felfernig et al. (2000b) 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 (Servlets, Java Server Pages, Enterprise JavaBeans). All components of the prototype are implemented within a three-tier architecture conformant to J2EE (Java 2 Enterprise Edition).

---

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

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

6. Conclusions

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

In this way, CAWICOMS will have benefits both for customers and suppliers: 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 co-operate along the supply chain by addressing the interoperability of product configuration systems.

For further information see *www.cawicoms.org*.

References

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

Felfernig, F., Friedrich, G., Jannach, D., and Stumptner M. (2000a). *Consistency-based Diagnosis of Configuration Knowledge Bases*. 14th ECAI, pp. 146-150.

Felfernig, F., Friedrich, G., and Jannach, D. (2000b). *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.

Felfernig, F., Friedrich, G., Jannach, D., and Zanker M. (2001). *Distributed Configuring*. Workshop on Configuration, 17th IJCAI.

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, vol. 13(4), pp. 59-68.

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, vol. 8(2), pp. 117-132.

Genesereth, M., Keller, A., and Duschka, O. (1997). *Infomaster: An Information Integration System*. 23rd ACM SIGMOD International Conference on Management of Data, pp. 539-542.

Jameson, A. (1990). *Knowing What Others Know – Studies in Intuitive Psychometrics*. PhD thesis, University of Amsterdam, Netherlands.

McGuinness, D. and Wright, J. (1998). *Conceptual Modeling for Configuration: A Description Logic-based approach*. Artificial Intelligence for Engineering, Design, Analysis and Manufacturing (AIEDAM), vol. 12(4), pp. 333-344.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* San Mateo, CA. Morgan Kaufmann.

Sabin, D. and Weigel, R. (1998). *Product Configuration Frameworks – A Survey*. IEEE Intelligent Systems, Issue on Configuration, vol. 13(4), pp. 50-58.

Soininen, T., Tiihonen, J., Männistö, T., and Sulonen, R. (1998). *Towards a general ontology of configuration*. Artificial Intelligence for Engineering, Design, Analysis and Manufacturing (AIEDAM), vol. 12(4), pp. 357-372.

Yokoo, M., Durfee, E., Ishida, T., and Kuwabara, K. (1992). *Distributed Constraint Satisfaction for Formalizing Distributed Problem Solving*. 12th IEEE Int. Conference on Distributed Computing Systems, pp. 614-621.

Winterfeld, D. von and Edwards, W. (1986). *Decision analysis and behavioral research*. Cambridge, England. Cambridge University Press.