A Framework for the Development of Personalized, Distributed Web-Based Configuration Systems

Liliana Ardissono, Anna Goy, and Giovanna Petrone[1]

Alexander Felfernig, Gerhard Friedrich, Dietmar Jannach, and Markus Zanker[2]

Ralph Schäfer[3]

Contact: Dietmar Jannach *dietmar.jannach@uni-klu.ac.at*

Abstract

For the last two decades, configuration systems relying on Artificial Intelligence techniques have been successfully applied in industrial environments. These systems support the configuration of complex products and services in shorter time with fewer errors and therefore reduce costs of a mass-customization business model. The EU-funded project CAWICOMS[4] aims at the next generation of Web-based configuration applications, that cope with two challenges of today's open, networked economy: The support for heterogeneous user groups in an open market environment and the integration of configurable sub-products provided by specialized suppliers.

This paper describes the *CAWICOMS Workbench* for the development of configuration services offering personalized user interaction, as well as distributed configuration of products and services in a supply chain. The developed tools and techniques rely on a harmonized knowledge representation and knowledge acquisition mechanism, open XML-based protocols, and advanced personalization and distributed reasoning techniques. We exploited the workbench based on the real-world business scenario of distributed configuration of services in the domain of IP-based Virtual Private Networks.

1 Introduction

[1] Dipartimento di Informatica, Università di Torino, Italy
[2] Computer Science and Manufacturing Research Group, University of Klagenfurt, Austria
[3] DFKI, Germany.
Authors are grouped according to their affiliation.

[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 EU through the IST Programme under contract IST-1999-10688; see http://www.cawicoms.org.

Over the last decade the market's demand for customer-individual, configurable products at costs of mass-production has been constantly increasing and a mass-customization [30,29] business strategy has been adopted in many industrial sectors. Along with the growing complexity of configurable products, the required supporting software systems (configuration systems) have been improved and successfully applied in industrial environments [17,24,39]. As a result, powerful tools based on different AI techniques like, e.g., Constraint Satisfaction [22,23], are available on the market. These techniques are used to improve the corresponding business processes with respect to the reduction of the order lead time and of the number of faulty configurations. However, today's networked economy imposes new demands on intelligent software support that are not adequately addressed by current technology:

- More and more complex products and services that are offered to the customer as integrated *solutions* are themselves assembled from configurable sub-products supplied by highly-specialized providers. Due to organizational reasons or confidentiality issues, the integration of the complete configuration logic in one centralized knowledge base is not desirable. Therefore, adequate means for *distributed configuration* and integration of heterogeneous configuration systems have to be provided.
- The increased complexity of products makes the interaction with configuration systems lengthy, as lots of features have to be specified during the configuration process. Moreover, this complexity challenges the end-user, who needs deep technical knowledge to specify the required information. In order to make configuration systems usable not only by technical engineers, but also by less experienced end-users, such as sales representatives and customers, these systems have to be extended with *user-adaptive interfaces*, which tailor the interaction to the individual user, guiding her through the configuration process in a personalized way [6,16,33].

The CAWICOMS project aims at enhancing the state of the art in configuration systems in order to take such issues into account. Within this project, we have developed the *CAWICOMS Workbench*, an environment for the creation of user-adaptive Web-based configuration systems. This workbench offers:

- A *configuration shell* supporting distributed configuration and personalization;
- A set of *tools for the specification of the domain-dependent knowledge* about the products/services to be configured and the types of end-users to be served.

The shell is based on advanced distributed configuration and adaptive hypermedia techniques [7,8,1] for the management of personalized configuration interactions involving the cooperation between suppliers.

The design and development of the CAWICOMS Workbench was mainly driven by our industrial partners'[5] business cases. In particular, BTexact Technologies provided the scenario concerning the joint provision and configuration of IP-based Virtual Private Networks with multiple suppliers. Since we aimed at developing a general environment for the creation of Web-based configuration systems, we also considered a second scenario from the domain of telecommunication switching systems [12]. We exploited the CAWICOMS workbench to develop two prototype systems assisting the configuration of items in both scenarios. In this

[5] BTexact Technologies, Telecom Italia, ETIS, ILOG SA.

paper, we will focus on the first scenario as a concrete example to present the methodologies applied within our shell.

Section 2 overviews the application domain of IP-VPNs. In the subsequent sections, we describe the individual components of the workbench and relate them to existing and emerging approaches in the respective areas.

2 Configuration of IP-VPNs

IP-based Virtual Private Networks (IP-VPNs) are used to extend a company's private network to remote offices, business partners, or roaming users [31]. These networks are run over the public Internet and dedicated backbone networks owned by the providers, and are currently limited within country borders and the provider's backbone network. However, the future business model includes transnational multi-provider networks, as well as the provision of integrated *services*, that also include, e.g., installation support or router configuration at the customer's site. From a sales perspective, specialized resellers offer their customers integrated solutions consisting of basic services provided by the suppliers. The configuration problem for the reseller concerns assembling and configuring a set of basic services that satisfy the customer's demands and obey technical and non-technical constraints.

In the envisioned business scenario, suppliers *publish* their services to the resellers. The configuration system then starts the search process for an appropriate set of network segments, hardware components (e.g., customer-site routers) and services (e.g., installation and maintenance support) satisfying the customer's requirements. This high-level solution design is the basis for the quotation phase. In a second step, the detailed configuration of the VPN has to take place. The results of the first step are sent to the remote configuration systems at the suppliers' sites and additional low-level technical settings may be returned by such systems. These results are checked for consistency and incorporated in the final solution. In this scenario, the following key requirements, not addressed by current configuration technology, are identified:

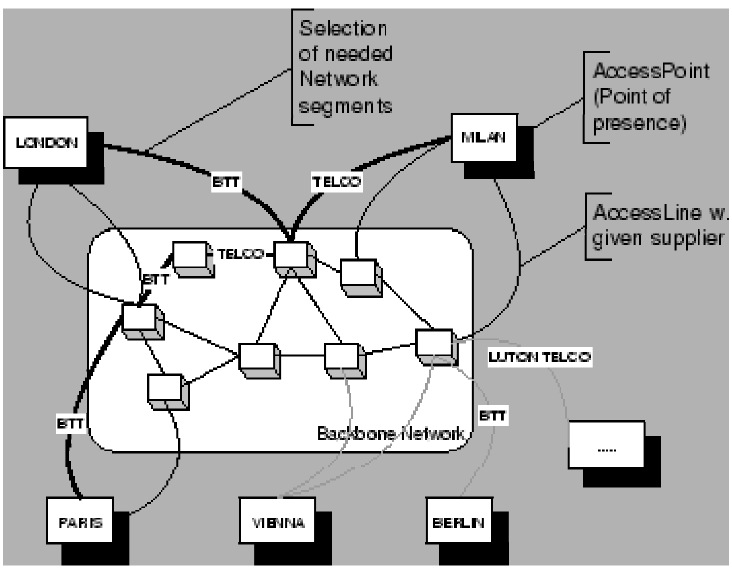

Figure 1: Reseller's view on IP-VPN Network. The edges of the network depict the available network segments. For each external access point there are several alternative connections to the backbone network. The vertices within the backbone network represent interconnections between different companies' backbone segments.

Shared knowledge representation. Currently, product classification and automated advertisement on electronic marketplaces for standard products is supported. In contrast, there is little support for complex, customizable products. To jointly configure such solutions, the supply chain participants have to share knowledge about the products and have to agree on a common methodology to model the products and exchange information at run-time.

Adaptive and dynamically generated User Interfaces (UI). When introducing complex and evolving products, there is typically a shortage of trained sales people. Moreover, people playing different roles in their own organization, e.g., managers and technical engineers, can use the same system to inspect the proposed solutions. These classes of users have a different view of the same configurable item. While a technician might see it as a set of pluggable components, an end-user may be interested in the offered functionality or service. In order to satisfy the information needs of such different end-users and to adapt the configuration process to their capabilities, a dynamically generated UI is needed [1]. This interface has to fill the gap between the system's point of view, focused on the implementation of the item to be configured, and the user's one, focused on the high-level properties offered by the final solution, e.g., its quality, reliability, and so forth.

Distributed reasoning. Recently, significant advances in the Distributed AI, Multi-Agent Systems and Distributed Constraint Satisfaction areas have been made [37,41]. However, most current approaches do not fit well the characteristics of distributed configuration. For instance, configuration systems typically do not work autonomously or fully parallel like, e.g., agents do. Moreover, they require very specific problem solving mechanisms, thus making the integration of legacy systems problematic. Therefore a simple and open protocol for distributed reasoning and an exchange format for complex data structures in a configuration scenario has to be used.

3 CAWICOMS Workbench

In the following, we present an overview of the CAWICOMS architecture and then we outline the details.

3.1 Architecture of the CAWICOMS Configuration Shell

To develop a system satisfying the key requirements reported in Section 2, different methodologies have to be integrated. Therefore, we designed the configuration shell offered by the CAWICOMS workbench as a modular system, including components specialized in the execution of complex tasks, such as the computation of configuration solutions, the dynamic generation of the personalized UI, or the communication with the suppliers.

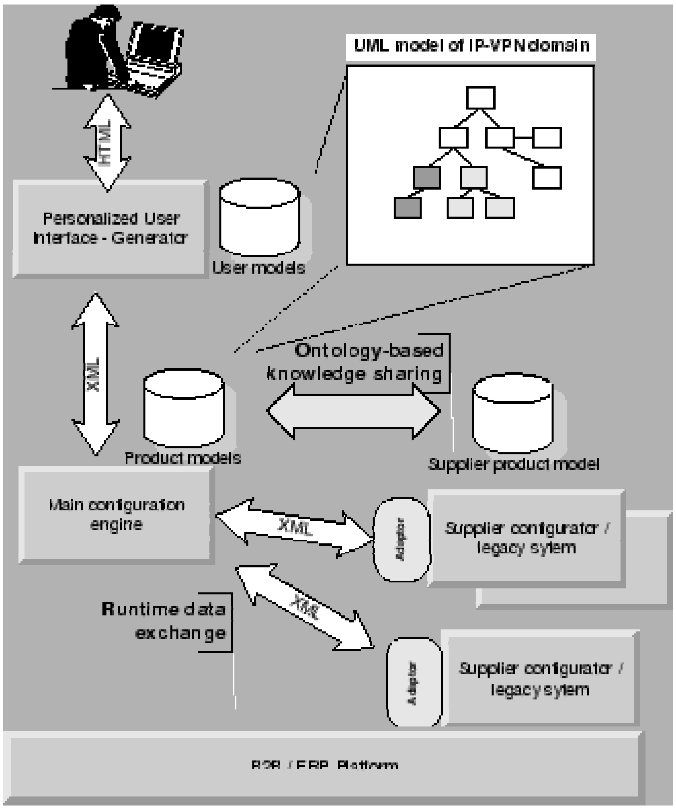

Figure 2: Architecture overview.

According to the CAWICOMS architecture [10], one designated *main configuration system* serves both as mediator for the distributed configuration process and as interaction point for the user. This system includes a domain-independent, Java-based configuration engine (ILOG's *JConfigurator* [22,23]), a User Modeling Component for the management of models representing the end-users' characteristics, and an Adaptive User Interface Generation module that mediates between the end-user and the configuration engine, automating the configuration process, whenever possible, and providing the user with assistance.

3.2 Knowledge Representation

An adequate representation of the configuration knowledge is the glue for the tools and techniques developed in the workbench. Knowledge acquisition and maintenance is critical to the development of centralized configuration systems [17] and especially here, because:

- Knowledge has to be shared among participants in the supply chain and different reasoning mechanisms and tools must be integrated;
- The adaptive User Interface (UI) has to be dynamically generated by applying business rules and personalization strategies based on the information about products and users stored in the knowledge base.

In CAWICOMS, the representation of the knowledge about products and services relies on the widely adopted *component-port* approach for configuration [17,23,27,39]. We developed a domain-independent meta-model and a UML-based (Unified Modeling Language) graphical notation with precise semantics [11], that follows the standard modeling concepts of the configuration domain [38]. The resulting *product models* can be further enriched with constraints expressed in OCL (Object Constraint Language) [34] and are independent of specific tools and their proprietary knowledge representation mechanisms. In addition, to improve the knowledge acquisition and maintenance process for centralized configuration systems, this modeling technique serves as a common language among the participants in the supply chain, thus leveraging the knowledge-integration process. Furthermore, we defined a human-readable, textual representation of the product models based on XML-Schema[6] documents. We also implemented tools for the automated document generation from commercial UML editors, such as Rational Rose[7]. These documents also serve as a starting point for further knowledge acquisition, i.e., knowledge about the suppliers.

Also the domain dependent knowledge needed to *personalize the interaction* with the user is represented by exploiting XML-Schema. This type of information includes the representation of the properties of products and services and the specification of the characteristics of the end-users:

- The representation of the features and the structure of products and services are extended with personalization-oriented information, such as the classification of features in metalevel information classes (technical, economic, and other types of information) and their complexity level. Moreover, a natural language explanation of their meaning is provided; see also [4]. Furthermore, in order to introduce a functional view of products and services, suitable for reasoning about configurable items at the end-user's abstraction level, some properties of products and services, such as their performance and reliability, may be specified to evaluate the configurable items from a qualitative point of view [21,9]. These properties have to be related to suitable sets of technical features, in order to describe the impact of such features on the evaluation of the product with respect to the corresponding properties. For instance, the inclusion of a certain component in a configuration may have a positive or negative impact on the evaluation of one or more properties. Figure 3 shows the evaluation of the reliability of an IP-VPN with respect to some of its technical features.
- The users are described by specifying general characteristics (e.g., nationality and type of company they represent), their expertise and their interests in the properties of products and services (Figure 4). The system also exploits a description of characteristics of user classes, e.g., managers, sales engineers, technical engineers, end-customers. This type of information is defined by a set of stereotypes [32] specifying skills and interests of the users belonging to the various classes.

[6] See http://www.w3.org for more information.

[7] See http://www.rational.com for details.

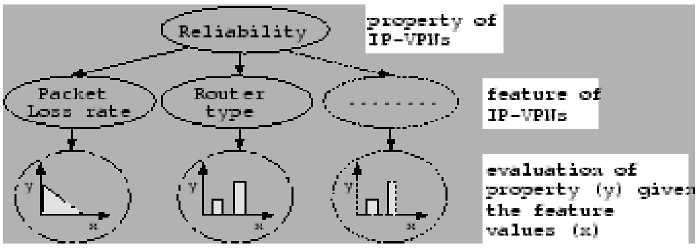

Figure 3: Evaluation of a property of a product/service, given its technical features. The impact of a set of features on the evaluation of a property is represented by defining *evaluation functions* that map evaluation values onto the feature values. For instance, this picture shows that the packet loss rate influences the reliability of an IP-VPN in a negative way. The higher is the rate, the lower is the reliability of the network.

3.3 Adaptive User Interaction

The CAWICOMS configuration shell manages a dynamic user interface that adapts the interaction style to the user, by customizing the elicitation of requirements and the presentation of information; see [2] and [5]. The UI relies on the management of a user model that describes the capabilities and preferences of the individual end-user [35], on the application of personalization techniques for the generation of customized content [4] and on the dynamic generation of UI pages [3]. The methodologies underlying these two activities are outlined in the following subsections.

```
Customer features:
   Customer name: Jane Doe;
   Enterprise type: Small Enterprise;
   Company type: Web Design;
Individual defaults:
   Quality of service: silver;
Interests in product/service properties:
    Quality:<H,0.7>,<M,0.2>,<L,0.1>
    Extensibility:<H,0.4>, <M,0.4>,<L,0.2>
    Economy:<H,0.3>, <M,0.4>,<L,0.3>
   ...
```

Figure 4: Representation of the user's data. The system manages an individual user model that represents the system's beliefs about the user interacting with the system. The user model describes the customer characteristics, such as the nationality and enterprise type. Moreover, this model describes the user's expertise about the products/services, and individual defaults, i.e., her preferences for specific feature values. The user model also stores the system's estimates about her interests in high-level properties of the product/service, such as its reliability and economy. As shown in the figure, the characteristics are represented as <feature,value> pairs. The estimates about the user's interests and expertise are represented as probability distributions on the values of variables associated to the knowledge items and the properties. For instance, a variable represents the user's interest in the reliability of a configuration solution.

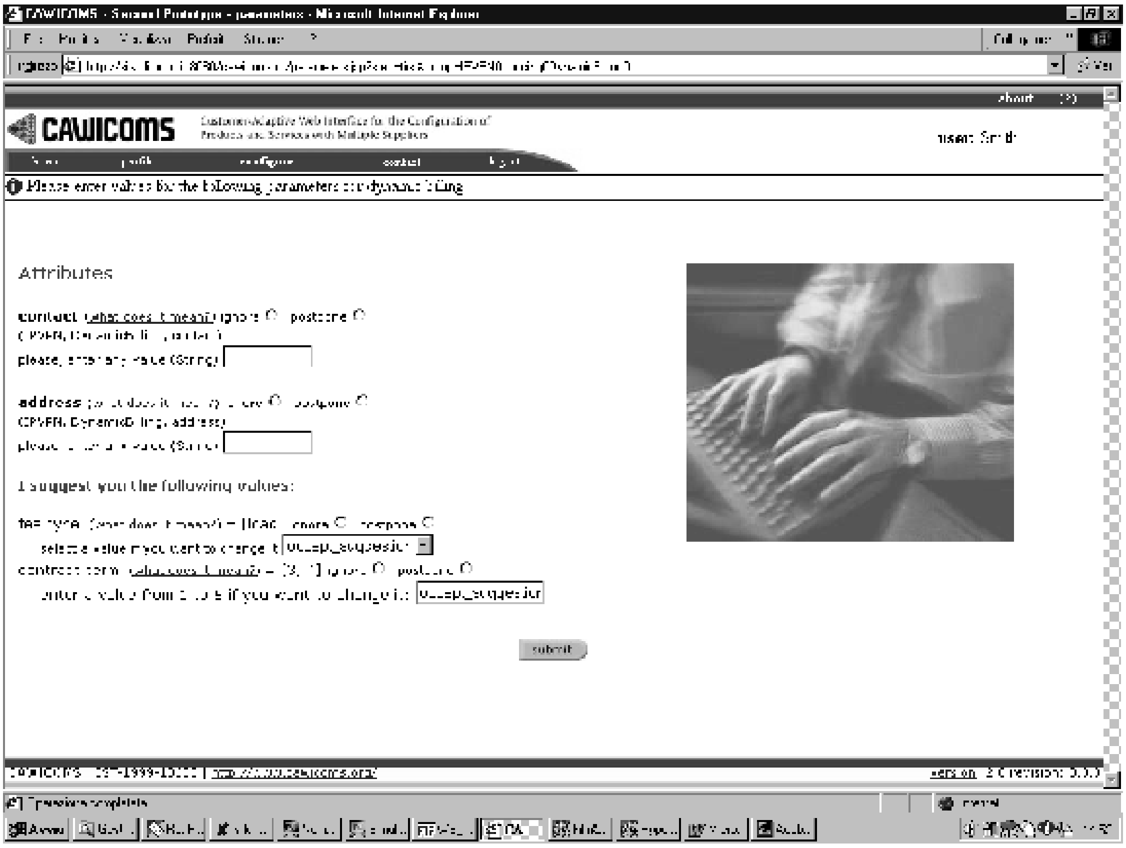

Figure 5: A step in the configuration of an IP-VPN. The system asks the user to specify contact point and address information and suggests default values for the fee type and the contract term. Each feature has a "what does it mean?" link, providing detailed information about the feature, and two buttons. The "ignore" button can be used to delegate the system to automatically set the feature value; the "postpone" one enables the user to set the value later on. Although not shown, the system may also elicit the required information in an indirect way, by asking the user about her preferences for one or more properties of an IP-VPN related to the feature to be set. Given the user's answer, the most suitable settings to achieve such properties are determined. Moreover, the system offers two advanced facilities, i.e. the auto-configuration option and the possibility to suspend and resume a configuration session.

3.3.1 Management of the User Model

Some pieces of information about the individual end-user, such as the type of company she represents, her nationality, and similar data, are retrieved by the system by explicitly questioning her. Other types of information, such as the user's interests and her knowledge about the configuration domain, are estimated by observing her behavior during the interaction with the system. In particular, the user may self-assess her own expertise and interests, but this information has to be continuously updated to take into account the fact that the user's skills and preferences may evolve. In CAWICOMS, the user model is managed in two phases:

- When a new user interacts with the system, her interests and skills are estimated by exploiting the predictions provided by the stereotypes describing the user classes she belongs to. The user is matched against the possible user classes and the best matching stereotypes are applied to initialize the user model. The stereotypical information enables the system to personalize the configuration process on the basis of an approximated user model since the beginning of the interaction.

- During the interactive configuration sessions, the estimates of the user's interests are updated by interpreting her behavior as an attempt to maximize the utility of a configuration solution (cf. [21]) in typical interaction situations, such as expression of interest, change of some proposed default value, acceptance or rejection of a proposed configuration [36].[8]

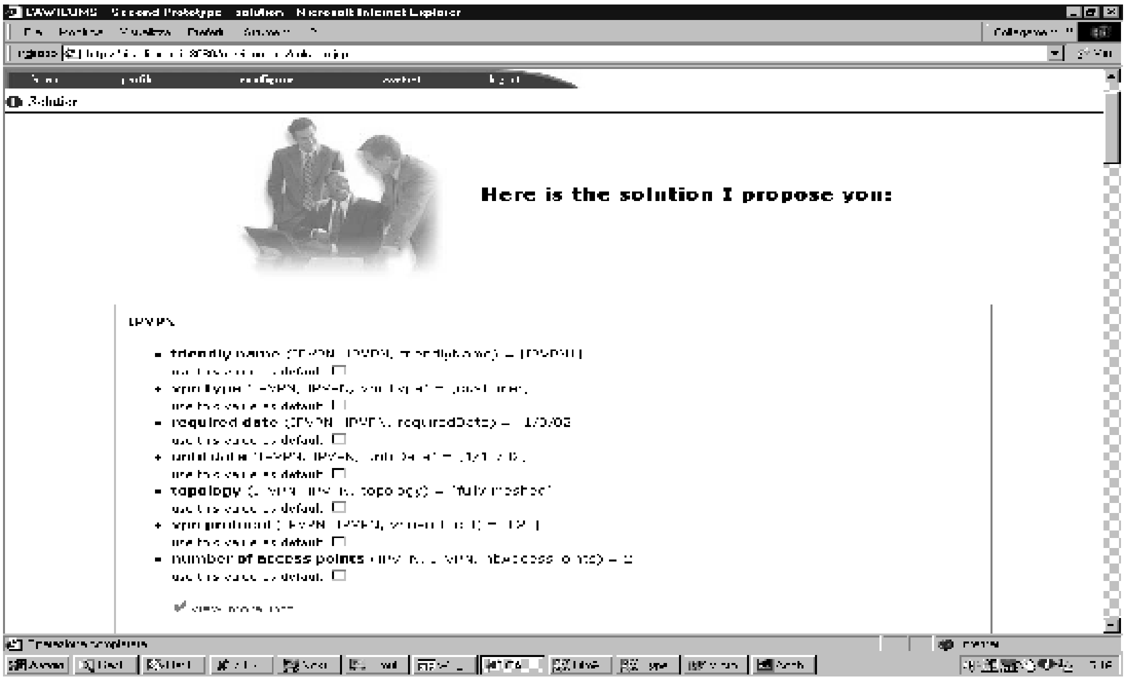

Figure 6: Presentation of a configured IP-VPN. The specification of the whole set of features is not visible in the picture because this type of service is rather complex. The presentation is organized in sections related to the high-level structure of the IP-VPN. First, the general features of the IP-VPN are shown; then, the type of support to be provided by the reseller is specified, and so forth. The system hides some details to focus the presentation on the most relevant information, given the user's interests. The user can reach such details by following the "view more info" link.

The second point deserves further discussion. Being the user a rational agent, we assume that she tries to maximize her own utility by setting the item features to satisfy her own needs in the best way. This assumption enables us to reason about the user's actions in order to assess her interests in the properties of the product/service, thus revising the user model accordingly. The system's functional knowledge about features and properties (see Figure 3) supports the identification of the user's interests in the properties related to the features she sets during the configuration process. If the user selects a specific value for a feature *F*, or she rejects a default value suggested

[8] We omit the description about the estimation of the user's expertise, which is performed as described in [20].

by the system and selects a different one, this fact can be interpreted as an attempt to improve the utility of the item. Therefore, the system can infer that the user has a certain interest in the properties influenced by *F*.

Our system estimates the user's interests in the properties of products and services by ascribing her a model for the evaluation of items based on the widely-adopted Multiattribute Utility Theory (MAUT, [40]). This model defines the evaluation of items as a function of the evaluation of their properties, given the user's interests in such properties; see [35]. The exploitation of this theory enables the system to interpret the acceptance or the rejection of a specific configuration solution as a sign that the user's evaluation of the solution is good/bad. In turn, this information is relevant to assess the user's interests in the item properties. As the interpretation of the user's actions is affected by uncertainty, Bayesian networks [28] are used to perform this task.

3.3.2 Personalization of the Interaction with the User

The CAWICOMS configuration shell manages the interaction with the user as a dynamically generated sequence of configuration steps. At each step, the configuration engine is queried about the current set of features to be set and the module responsible for the generation of the UI selects the most convenient way to set them. When the features are set, their values are used by the configuration engine to refine the state of the configuration. This process continues until no more features have to be set. At that point, the engine generates a solution which is presented to the user. The selection of the features to be considered at each configuration step and the presentation of the solution are based on the conceptual structure of the product model, which provides a component-based view of the item to be configured[9].

The estimates on the user's actual interests are exploited to steer the configuration process by proposing feature values and components that maximize the user's expected evaluation of the solution. In this way, the system reduces the number of features she has to set. The system also uses other personalization strategies; for example, the features can be set by applying individual defaults expressing the user's specific preferences for feature values, or personalized defaults describing business rules based on the customer's characteristics.

Finally, the system personalizes the presentation of a solution to be proposed to the user (e.g., a specific IP-VPN), by focusing the description on the features suitably fitting her interests and expertise [4]. Given a feature, the system decides whether it has to be shown in the main presentation page or it can be linked as supplementary information about the item. The system shows the critical features, which the user must be aware of, and those related to properties she is very interested in. The less important features, those too complex for the user and those outside her interests, usually fall into the supplementary information, in order to keep the presentation of solutions as synthetic as possible.

A rule-based mechanism is employed to select personalization strategies determining the generation of defaults to be proposed during the configuration process and the presentation style to be applied to the features of the configuration solution [2]. The generation of the UI is based

[9] In contrast to the internal view of the configuration engine, which may be completely flat, as it happens for constraint-based engines.

on the JSP (Java Server Pages) technology[10], which enable the system to dynamically generate the content of the Web pages, on the basis of the status of the configuration process and of the user's characteristics. As the user model is continuously revised by the system, the application of personalization techniques and the dynamic generation of the UI support a reactive adaptation of the interaction, so that the system can repair the possibly wrong personalization behavior (due to errors in the estimation of the user's properties) and also adapt to changes in the user's attitudes. Figures 5 and 6 show two sample pages generated by the prototype for the configuration of IP-VPNs we developed by exploiting the CAWICOMS workbench.

3.4 Distributed Reasoning

Distributed Configuration problem solving in a supply chain in CAWICOMS relies on three pillars, described in the following subsections:

- Ontology-based knowledge sharing;
- Open XML-based communication protocols and other interfaces;
- Advanced Constraint Satisfaction techniques.

3.4.1 Ontology-Based Knowledge Sharing

Distributed Problem Solving approaches always require ontological commitments on domain specific terms and conditions in order to assure the same interpretation of the content of communication by all participants. Therefore, the involved configuration systems must establish a shared view on the product which requires an integration and harmonization process during the setup of the supply chain. In order to establish a common language for representing configurable products, we defined a hierarchy of related ontologies [19] and structured these ontological commitments, according to the proposal of [18], into three hierarchical levels: The generic ontology level, the intermediate level, and the domain level.

In the generic ontology level, the basic representation concepts and modeling primitives are introduced. We used the Unified Modeling Language (UML), whose metamodel defines the basic concepts like, e.g., Class, Association, and so forth. Based on these definitions, specific ontologies can be created that refine the generic concepts for a certain domain. For the easy integration of models of complex configurable products and knowledge reuse, different general ontologies for the configuration domain have been proposed [11,38]. Within the CAWICOMS workbench, a UML profile for the configuration domain is applied, which is a synthesis of structure-based, function-based, and resource-based approaches to configuration. The developed profile consists of a set of modeling concepts for the configuration domain, e.g., component types or ports, and is based upon precise semantics which are needed for automating the knowledge acquisition and exchange process. Finally, the configuration profile (stereotyped modeling concepts in UML) is used to define the domain-specific terms, vocabulary and relations. In our application case, examples for specific concepts in the IP-VPN domain are "Network segment" or "Access Point"[11].

[10] See htt://java.sun.com/products/jsp/ for details.
[11] Note that actual configurations can be seen as instance models of the domain-specific model.

The key innovation for seamless supply chain integration in B2B eCommerce is to tackle the integration problem at the conceptual level, i.e., the supply chain partners agree on a common way of conceptualizing the configuration problem. The advantage of this approach is that, during the integration effort, low-level technical details and proprietary knowledge representation mechanisms can be neglected and the integration can be done on an easy-to-understand, graphical basis. Furthermore, the proposed conceptualization allows us to be independent of proprietary or legacy configuration systems. Given a defined set of modeling concepts in the intermediate ontology layer, the resulting conceptual product models can be easily transformed or mapped to the representation of an existing (knowledge-based) configuration system.

In order to participate in the supply chain of configurable products, the public (shared) portions of the supplier products are modeled using the defined UML configuration profile. These models are then integrated into the view of the mediating main configuration system. Having a partial view on the local problem of other agents is a typical situation in distributed problem solving like in, e.g., Distributed Constraint Satisfaction ([37,41]). However, the focus of such approaches lies in intelligent, distributed reasoning and the process of setting up a Multi-Agent scenario, while the acquisition and maintenance of the knowledge is neglected.

For non-standard, proprietary products like in the domain of telecommunication switches [12], knowledge is integrated in a knowledge engineering process and the cooperation is based on long-term business relationships and contracts. However, in domains like IP-VPN-provision, where the configurable basic services are quite similar (e.g., Internet Service Provision) and there are many different suppliers for the same service, the goal is to come to an agreed-upon product model for these basic services. Given such an agreement, the involved communities can establish electronic marketplaces for configurable products in a specific domain. One future goal is therefore to extend existing eMarketplaces with a standardized (semantic) configuration Web Service [25] for its participants. For details on this approach, refer to [14].

3.4.2 Open Communication Protocols and Platform

A key issue for centralized configuration applications is the integration into the corporate's software infrastructure like ERP systems or B2B platforms. This issue is addressed in CAWICOMS and ILOG's JConfigurator by providing adequate interfaces [10], e.g., by allowing the configuration knowledge base to contain references to external data sources via a database connectivity interface; therefore, data does not have to be kept redundantly in the knowledge base. Moreover, the workbench is completely developed on the J2EE [26] platform, which is designed for easy integration of components into enterprise environments.

To support the communication within the workbench, the CAWICOMS consortium has developed an XML-based protocol that relies on ILOG's *WebConnector*. This protocol, according to the evolving paradigm of XML-based SOAP messaging[12] and *Web Services*, defines:

- A fixed set of methods with defined semantics for the configuration domain, like creating components, setting values for parameters, initiation of the search process, or retrieving results;

[12] See http://www.w3.org for details.

- A mechanism to exchange complex data structures like configuration results and a language for expressing navigation expressions within these data structures (compare XML-Schema and XPath);
- Extensibility mechanisms for special domains and support for transactions for HTTP-based configuration sessions.

The main advantages of this approach lie in the independence from individual programming languages and in security-related issues, e.g., the possibility to communicate over HTTP connections without the need to open corporate firewalls for IP-based connections.[13] Beside the definition of the protocol and the interfaces, the CAWICOMS workbench includes a set of corresponding software tools, e.g., components for transaction support and tools for interface development, based on standard technologies such as XSL Transformations and JSP [26].

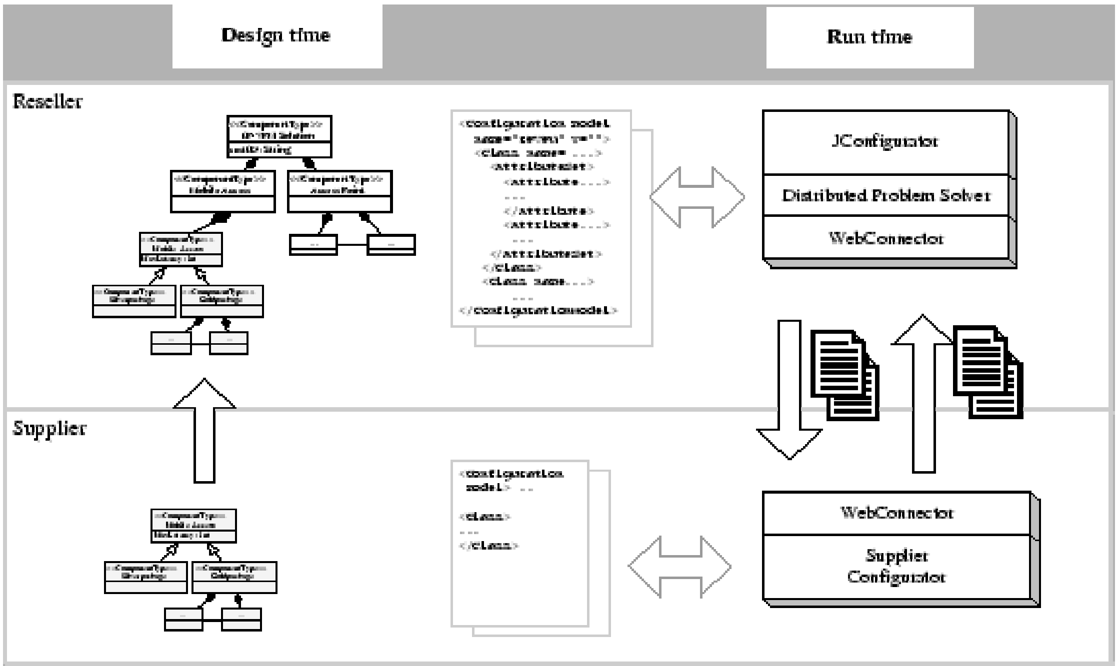

Figure 7: Schematic knowledge integration and deployment scenario for a simple supply chain. At design time, knowledge integration between the reseller's and the supplier's view on the configurable product is done based on the same conceptualization using the Unified Modeling Language. These models are transformed into an XML-Schema representation and the resulting knowledge base is enriched with personalization information or additional constraints that cannot be expressed graphically. Before run time, the knowledge bases are loaded by the configuration engines, the distributed problem solving component, and the UI generator. During the distributed search process, the agents communicate using the XML-based *WebConnector* protocol. Using a defined set of performatives and a message content format, the reseller can, for instance, set a value for different parameters or initiate the search process at the supplier system. ILOG's WebConnector component is used to parse and transform the protocol messages into the internal data structures of the underlying constraint-based configuration engine.

[13] ILOG's WebConnector toolkit also supports Java-based communication for homogeneous environments.

3.4.3 Distributed Problem Solving

Historically, several reasoning techniques from the field of Artificial Intelligence where developed and employed for the core configuration tasks, like consistency checking, automatic completion of a partial configuration, or optimizing a solution. Compared to procedural or rule-based implementations of the configuration logic, modern approaches, based on techniques like Constraint Satisfaction, offer the advantages of simple and declarative knowledge representation and the availability of high-performance search and optimization algorithms. Nonetheless, these algorithms and techniques are designed to run as centralized processes and therefore cannot cope with the advanced distributed negotiation requirements among partners in an informational supply chain for configurable products. Although the field of Distributed Artificial Intelligence made substantial progress in the last decade, accompanied by the emergence of the agent paradigm, existing algorithms for distributed reasoning (see, e.g., Asynchronous Backtracking, [41]) have different shortcomings when applied to real-world supply chain settings: Their notion of parallelism does not fit the client-server interaction style in a (typically tree-structured) supply chain; moreover such multi-agent systems require that the involved systems implement complex algorithms and data structures; finally, the issues of knowledge acquisition and maintenance, as well as supply-chain set-up, are neglected. In the CAWICOMS framework, a parameterizable, distributed algorithm is implemented that trades a reduction in achievable agent parallelism for more simplicity in the algorithm and easier maintenance of the supply chain. The framework utilizes the given supply chain structure and introduces integrating agents that coordinate the work of their supplier systems. The integrating agents also support dynamic supply-chains by computing appropriate sets of suppliers as part of the configuration process. Technically, the mediating agents solve their local problems based on an extension of the Generative Constraint Satisfaction [17] mechanism, whereby this core functionality is implemented in ILOG's commercial JConfigurator libraries. Each agent is capable of accepting requirements from its clients (via the defined protocol) and solving the local configuration problem. In cases where this local sub-problem can only be solved by involving additional supplier systems, each mediating agent may in turn contact its own suppliers and forward the current requirements that are determined by the current search process. The capabilities of the supplier systems include consistency-checking, computation of a result, or detection of inconsistencies, in which cases the client system has to initiate a backtracking process. The CAWICOMS framework includes two sample implementations of such complete algorithms, that take special characteristics of their real-world problem domains into account. These implementations are described in detail in [12] and [15] respectively; the formal requirements when implementing such algorithm variants are described in [13]. Beside these two reference implementations, the framework includes utility libraries for, e.g., generation of synchronous and asynchronous messages according to the protocol, access to local knowledge bases on suppliers, or methods for reacting on events in the local search process (like component generation) in order to steer the distributed search process. Finally, the usage of an open configuration protocol and message exchange format allows us to integrate non-CAWICOMS configurators at the leaf nodes of the supply chain which do not have to implement the required mediating capabilities. Legacy configurators can be integrated by using wrapper components that map the small set of protocol messages into their internal knowledge representation format.

4 Evaluation

The prototype system was evaluated a number of times throughout the development process and the evaluation results were used to guide its further development. In order to test features of the adaptive interface, we used role-playing exercises involving 50 users with different levels of expertise. Some of them were familiar with existing configuration systems, but not necessarily IP-VPN, some were familiar with the domain, some completely new to this area of technology. These users gave feedback by filling in a questionnaire presenting questions about the usability of the system, aimed at assessing the usefulness of the personalization and explanation facilities and the flexibility of the user interface.

We performed tests using two sets of scenarios. One for the Telecommunications Switches domain and one for the IP-VPN domain. Here, we concentrate on the IP-VPN domain, which was used for the final evaluation. Our IP-VPN test scenario involved a fictitious company named WooCorp. WooCorp are a fast growing technology start-up company with offices across Europe and need to purchase a VPN to replace their existing collection of ad-hoc communications links. They have decided to go to a reseller who employs a CAWICOMS configuration system. We developed a number of different test scripts within this scenario, going from the initial specification of a small network to connect WooCorp's major sites, then adding in more complexity such as further sites and dial-up access for mobile users. The results of the test provide strong evidence that the personalized user interface improves the usability of a configuration system, concerning both the length and the quality of the interactions with the users. More specifically:

- The personalized suggestion of values for the features of items and the advanced configuration facilities, such as the automatic configuration of products and services, were approved by the nearly all users, because they can speed up the configuration process.[14]
- The elicitation of qualitative user requirements on the items (preference elicitation) represented valuable feedback for the less experienced user, as they support an intuitive specification of requirements on products and services. Moreover, novice users particularly enjoyed the ability to let the system automatically set certain item features.
- The users found that the explanation capabilities of the system (which provides the reasons for a failure by highlighting the violated constraints) are very useful to understand the causes of the failures in the configuration and to repair such failures by selecting different values for the problematic features.
- The possibility to suspend a configuration session and resume it later on was useful to the users who participated in the most complex test scripts.

The personalized presentation of solutions was appreciated, but the users also provided feedback for further improvement, asking to support a navigation within the structure of the solution.

A 10% of the users could not successfully end the configuration process because they were not able to repair the conflicts that were caused by inconsistent requirements. In order to enhance the flexibility of our system, we recently extended the CAWICOMS system with an "undo" facility. This operation is aimed at retracting configuration values, so that a different item can be configured without starting the interaction from the beginning. The "undo" of a feature setting can be performed to repair to a failure (choosing a different value for the problematic features),

[14] The CAWICOMS system may run in personalized and non-personalized mode. In the personalized mode, the system provides suggestions for about half of the features to be configured.

but also to enable the user to change her mind, before the solution is generated. Although we could not test the usability of the this facility yet, we are aware that it might challenge the user who is not familiar with configuration technologies. Neverthless, it represents a powerful tool, helping the expert users to configure items in a quicker and more flexible way.

5 Summary and conclusion

Driven by the new requirements of today's networked markets and based on emerging business models in the telecommunication domain, a framework for rapid development of advanced Web-based configuration systems was developed. The CAWICOMS workbench includes technologies and tools addressing the demands of personalized and adaptive user interfaces and advanced cooperative problem solving in the supply chain. The workbench's components are based on the implementation of best practices in the corresponding fields, as well as on newly developed mechanisms and algorithms. Furthermore, the need of openness and integrability in a dynamically changing, heterogeneous environment were taken into account by complying to existing and emerging standards and by relying on state-of-the-art software platforms and XML-based protocols.

Let us conclude this article with a broader view on some experiences we made during the project and an outlook on future developments and the next steps.

From a research perspective, this project provided an excellent opportunity to integrate the results separately achieved in the Configuration Systems and Intelligent User Interfaces research communities which, up to now, did not join their efforts to enhance the state of the art in the development of complex problem solving applications. The integration of such results was not easy, for two main reasons. First of all, this was the first attempt to design an interaction model mediating human and automated configuration reasoning. Second, the scenarios provided by our application partners corresponded to future or emerging business models and we could not fully rely on experiences or precise requirements statements. Consequently, permanent and efficient knowledge exchange concerning both the common overall goals of the project as well as lower-level technical issues had to be practiced during the whole project life cycle. On particular interesting challenge we encountered was the knowledge transfer among AI research and industry. One the one hand, there is little awareness of the already existing capabilities of AI technology in industry, particularly in our case, where new algorithms and techniques were developed. On the other hand - in order to work with real-world application scenarios - we had to elicit the specific business requirements of the application partners. One of the major difficulties in this process lies in understanding the peculiarities of the respective domains while, at the same time, generalizing such requirements to a level such that a more general software framework can be designed.

As far as technical issues are concerned, and in particular the employed base technologies like the choice of the programming language and environment, we made good experiences in using state-of-the-art technology from industrial environments. The whole workbench is developed using Java-based Web and component technology and XML-based knowledge representation and exchange formats. Beside the possibility of exploiting standard tools for e.g., dynamic Web page generation or XML-parsing, these technologies have the advantage that many of the software tools from the surrounding information technology infrastructure like databases or e-commerce platforms, can be quite easily integrated. Even more, the usage of a pure Java-based constraint solver can also be seen as a key success factor for industrial applicability of such systems. Using standard programming languages (such as object-oriented languages) in such a project will typically result in lower project costs for e.g., interface programming; in addition, the expertise of the existing development staff can be exploited without additional training or paradigm shifts in the programming style.

Concerning the long term business perspectives of the challenges dealt with in the CAWICOMS project, we can identify that there are three major still ongoing trends in the market: Personalization, mass-customization, and increased supply-chain integration of highly-specialized solution providers.

Many companies see the provision of personalized content in the Web as an additional means to improve their customer services and generate added value for their customers. Within the CAWICOMS framework, the novelty of the approach lies in the personalization of the interaction process for an underlying configuration search routine. Above this, in future work, the collected information about the customer's needs and preferences (the user profile) can be integrated into a Customer Relationship Management (CRM) system in two different ways. Once a customer is identified in a configuration session, we can retrieve the user profile from the CRM knowledge base. On the other hand we can, vice versa, upload additionally learned or entered information to the CRM system. Such an integration improves the communication with the user that potentially hop between several sales channels (internet, sales representative, or customer care center). In addition, the information about (un)realizable customer requirements can be exploited to revise the product portfolio in order to discover new market opportunities.

The *mass-customization approach*, which can be also seen as personalization of the products themselves, evolved in the last two decades and is nowadays the standard business model for many industrial sectors. As a result, every major vendor of Enterprise Software (like SAP) has incorporated software modules that support the business processes related with mass-customization and product configuration. The CAWICOMS project tries to advance current technology in that area in two dimensions. First, the application scenario deals with the configuration of *services* rather than products. In fact, many companies try to differentiate themselves from their competitors that manufacture comparable products by providing additional services. In many cases the main profit for the companies stems from the accompanying services rather than the product itself. On the other hand, the techniques developed in the CAWICOMS framework advance current B2B eCommerce technology. Current systems support co-operation in the supply chain by exchanging orders, publishing product catalogs, or by supporting billing transactions. In the CAWICOMS approach, the goal is to support more sophisticated communication and negotiation facilities in an open network of cooperating suppliers for mass-customizable products and services over the Web.

Acknowledgements

We thank BTexact Technologies, in particular Jonathan Mitchener and Matthew Holland, for having provided us with the IP-VPN scenario. Moreover, we are grateful to ILOG, in particular to Xavier Ceugniet and Frédéric Pesquet, for having designed and developed the WebConnector protocol. Finally, we thank the European Union for having supported the research and development activity carried out in the CAWICOMS project.[15]

[15] The project's budget was around 3.5 million EUR.

Bibliography

1 The adaptive Web. *Communications of the ACM*, 45(5), 2002.

2 L. Ardissono, A. Felfernig, G. Friedrich, A. Goy, D. Jannach, M. Meyer, G. Petrone, R. Schaefer, W. Schuetz, and M. Zanker. Personalising on-line configuration of products and services. In *Proc. 15th Conf. ECAI*, pages 225-229, Lyon, 2002.

3 L. Ardissono and A. Goy. Dynamic generation of adaptive Web catalogs. In Oliviero Stock Peter Brusilovsky and Carlo Strapparava, editors, *Lecture Notes in Computer Science n. 1892: Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 5-16. Springer Verlag, Berlin, 2000.

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

5 L. Ardissono, A. Goy, M. Holland, G. Petrone, and R. Shäfer. Customising the interaction with configurators. In *Proc. of 9th Int. Conf. on User Modeling (UM 2003)*, to appear, Pittsburgh, 2003.

6 D. Benyon. Adaptive systems: a solution to usability problems. *User Modeling and User-Adapted Interaction*, 3:65-87, 1993.

7 P. Brusilovsky. Methods and techniques of adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 6(2-3):87-129, 1996.

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

9 S. Carberry, J. Chu-Carroll, and S. Elzer. Constructing and utilizing a model of user preferences in collaborative consultation dialogues. *Computational Intelligence*, 15(3):185-217, 1999.

10 CAWICOMS Consortium. *D01: Requirements, Application Scenarios, Overall Architecture, and Test Specification*. www.cawicoms.org, 2001.

11 A. Felfernig, G. Friedrich, and D. Jannach. UML as domain specific language for the construction of knowledge-based configuration systems. *Int. Journal of Software Engineering and Knowledge Engineering (IJSEKE)*, 10(4):449-469, 2000.

12 A. Felfernig, G. Friedrich, D. Jannach, C. Russ, and M. Zanker. Multi-site product configuration of telecommunication switches. pages 113-118, Innsbruck, Austria, 2002. Acta Press.

13 A. Felfernig, G. Friedrich, D. Jannach, and M. Zanker. Towards distributed configuration. pages 198-212, Vienna, Austria, 2001. Springer.

14 A. Felfernig, G. Friedrich, D. Jannach, and M. Zanker. Semantic configuration Web Services in the CAWICOMS project. pages 192-205, Sardinia, I, 2002. Springer.

15 A. Felfernig, G. Friedrich, D. Jannach, and M. Zanker. Web-based configuration of virtual private networks with multiple suppliers. pages 41-62, Cambridge, UK, 2002. Kluwer.

16 J. Fink and A. Kobsa. A review and analysis of commercial user modeling servers for personalization on the World Wide Web. *User Modeling and User-Adapted Interaction, Special Issue on Deployed User Modeling*, 10(2-3):209-249, 2000.

17 G. Fleischanderl, G. Friedrich, A. Haselböck, H. Schreiner, and M. Stumptner. Configuring large systems using generative constraint satisfaction. *IEEE Intelligent Systems*, 13(4):59-68, 1998.

18 A. Gangemi, D.M. Pisanelli, and G. Steve. An overview of the onions project: Applying ontologies to the integration of medical terminologies. *Data and Knowledge Engineering*, 31(2):183,220, 1999.

19 T. Gruber. Ontolingua: A mechanism to support portable ontologies. *Technical Report KSL 91-66, Stanford University, Knowledge Systems Laboratory*, 1992.
20 A. Jameson. *Knowing What Others Know: Studies in Intuitive Psychometrics*. PhD thesis, University of Amsterdam, 1990.
21 A. Jameson, R. Schäfer, J. Simons, and T. Weis. Adaptive provision of evaluation-oriented information: tasks and techniques. In *Proc. 14th IJCAI*, pages 1886-1893, Montreal, 1995.
22 U. Junker. Preference-programming for configuration. In *IJCAI Configuration Workshop*, pages 50-56, Seattle, 2001.
23 D. Mailharro. A classification and constraint-based framework for configuration. *AI in Engineering, Design and Manucturing*, 12:383-397, 1998.
24 D.L. McGuiness and J.R. Wright. Conceptual modeling for configuration: A description logic-based approach. *AI in Engineering, Design and Manufacturing (AI EDAM)*, 12(98):33-344, 1998.
25 S. McIlraith, T.C. Son, and H. Zeng. Semantic Web Services. *IEEE Intelligent Systems*, 16(2):46-53, 2001.
26 Sun Microsystems. Java Web Pages. http://java.sun.com, 2002.
27 S. Mittal and F. Frayman. Towards a generic model of configuration tasks. *Proc. Intl. Joint Conference on Artificial Intelligence (IJCAI'89)*, pages 1395-1401, 1989. Morgan Kaufmann.
28 J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, San Mateo, CA, 1988.
29 F. Piller and C. Schaller. Individualization based collaborative customer relationship management: Motives, structures, and modes of collaboration for mass customization and CRM. Technical Report 29, Dept. of General and Industrial Management, Technische Universität München, 2002.
30 B.J. PineII, B. Victor, and A.C. Boynton. Making Mass Customization Work. *Harvard Business Review*, Sep./Oct. 1993:109-119, 1993.
31 Infonetics Research. Virtual private networks, White Paper. http://www.infonetics.com, 2002.
32 E. Rich. Stereotypes and user modeling. In A. Kobsa and W. Wahlster, editors, *User Models in Dialog Systems*, pages 35-51. Springer Verlag, Berlin, 1989.
33 D. Riecken, editor. *Special Issue on Personalization*, volume 43. Communications of the ACM, 2000.
34 J. Rumbaugh, I. Jacobson, and G. Booch. *The Unified Modeling Language Reference Manual*. Addison-Wesley, 1998.
35 R. Schäfer. Rules for using multi-attribute utility theory for estimating a user's interests. In *Proc. 9. GI-Workshops: ABIS-Adaptivität und Benutzermodellierung in interaktiven Softwaresystemen*, 2001.
36 W. Schütz and R. Schäfer. Bayesian networks for estimating the user's interests in the context of a configuration task. In R. Schäfer, M. E. Müller, and S. A. Macskassy, editors, *Proceedings of the UM2001 Workshop on Machine Learning for User Modeling*, pages 23-36, July 2001.
37 M.C. Silaghi, D. Sam-Haroud, and B. Faltings. Asynchronous search with aggregations. In *Proc. of 17th National Conf. on Artificial Intelligence (AAAI)*, pages 917-922, Austin, TX, 2000.
38 T. Soininen, J. Tiihonen, T. Männistö, and R. Sulonen. Towards a General Ontology of Configuration. *AIEDAM - AI Engineering Design Analysis and Manufacturing Journal,*

Special Issue: Configuration Design, 12,4:357-372, 1998.
39 M. Stumptner. An overview of knowledge-based configuration. *AI Communications*, 10(2), June, 1997.
40 D. von Winterfeldt and W. Edwards. *Decision Analysis and Behavioral Research*. Cambridge University Press, Cambridge, UK, 1986.
41 M. Yokoo. *Distributed Constraint Satisfaction*. Springer, 2001.

CVs of authors

Liliana Ardissono
Anna Goy
Giovanna Petrone

Alexander Felfernig is a research assistant in Computer Science at the University Klagenfurt, Austria, where he obtained his MS degree and PhD in Applied Informatics. His research interests are in the fields of knowledge-based configuration, knowledge representation and the Semantic Web. Within the European Union project CAWICOMS he focused on design and implementation of the knowledge acquisition component.

Gerhard Friedrich is full professor of Computer Science at the University Klagenfurt, Austria, where he directs the Computer Science and Manufacturing Research Group. His research interests include configuration, knowledge representation and diagnosis. He received his PhD and MS in Computer Science from the Vienna University of Technology.

Dietmar Jannach works as research assistant in Computer Science at the University Klagenfurt, Austria, where he obtained his MS degree and PhD in Applied Informatics. His current research areas include knowledge-based configuration and the integration of model-based diagnosis techniques into the configurator development and deployment process. Within the CAWICOMS project he focused on the development and implementation of distributed problem solving algorithms.

Markus Zanker is a research and teaching assistant in Computer Science at the University Klagenfurt, Austria, where he obtained his PhD and master degrees in Applied Informatics and in Business Administration. His research interests are in the field of configuration, distributed constraint satisfaction and the Semantic Web. He participated in the CAWICOMS project and focused on distributed problem solving

Ralph Schäfer