AI Magazine Volume 24 Number 3 (2003) (© AAAI)

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

Liliana Ardissono, Alexander Felfernig, Gerhard Friedrich, Anna Goy, Dietmar Jannach, Giovanna Petrone, Ralph Schäfer, and Markus Zanker

■ For the last two decades, configuration systems relying on AI techniques have successfully been applied in industrial environments. These systems support the configuration of complex products and services in shorter time with fewer errors and, therefore, reduce the costs of a mass-customization business model. The European Union–funded project entitled CUSTOMER-ADAPTIVE WEB INTERFACE FOR THE CONFIGURATION OF PRODUCTS AND SERVICES WITH MULTIPLE SUPPLIERS (CAWICOMS) aims at the next generation of web-based configuration applications that cope with two challenges of today's open, networked economy: (1) the support for heterogeneous user groups in an open-market environment and (2) the integration of configurable subproducts provided by specialized suppliers.

This article 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 information processing–based virtual private networks.

Over the last decade, the market's demand for customer-individual, configurable products at a mass-production cost has constantly been increasing, and a mass-customization (Piller and Schaller 2002; Pine II, Victor, and Boynton 1997) 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 (Fleischanderl et al. 1998; McGuiness and Wright 1998; Stumptner 1997). As a result, powerful tools based on different AI techniques such as constraint satisfaction (Junker 2001; Mailharro 1998) 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 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 subproducts supplied by highly specialized providers. Because of organizational reasons or confidentiality issues, the integration of the complete configuration logic in one centralized knowledge base is undesirable. 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 because 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. 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 him/her through the configuration process in a personalized way (Benyon 1993; Fink and Kobsa 2000; Riecken 2000).

… the CAWICOMS WORKBENCH … offers a configuration shell supporting distributed configuration and personalization and a set of tools for the specification of the domain-dependent knowledge about the products and services to be configured and the types of end users to be served.

The CUSTOMER-ADAPTIVE WEB INTERFACE FOR THE CONFIGURATION OF PRODUCTS AND SERVICES WITH MULTIPLE SUPPLIERS (CAWICOMS) Project aims at enhancing the state of the art in configuration systems 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 and a set of tools for the specification of the domain-dependent knowledge about the products and 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 (ACM 2002; Brusilovsky 2001, 1996) for the management of personalized configuration interactions involving the cooperation between suppliers.

The design and development of the CAWICOMS WORKBENCH were mainly driven by our industrial partners' business cases.[1] In particular, BTexact Technologies provided the scenario concerning the joint provision and configuration of information processing–based virtual private networks with multiple suppliers. Because 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 (Felfernig et al. 2002b). We exploited the CAWICOMS WORKBENCH to develop two prototype systems assisting the configuration of items in both scenarios.

In this article, we focus on the first scenario as a concrete example to present the methodologies applied within our shell. We overview the application domain of information processing-based virtual private networks (IP-VPNs). We also describe the individual components of the workbench and relate them to existing and emerging approaches in the respective areas.

Configuration of IP-VPNs

IP-VPNs are used to extend a company's private network to remote offices, business partners, or roaming users. 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 multiprovider networks, as well as the provision of integrated services, that also include, for example, 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 nontechnical constraints.

In the envisioned business scenario, suppliers publish their services to the resellers (figure 1). The configuration system then starts the search process for an appropriate set of network segments, hardware components (for example, customer-site routers) and services (for example, 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 can be returned by such systems. These results are checked for consistency and incorporated into the final solution. In this scenario, the following key requirements, not addressed by current configuration technology, are identified.

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: When introducing complex and

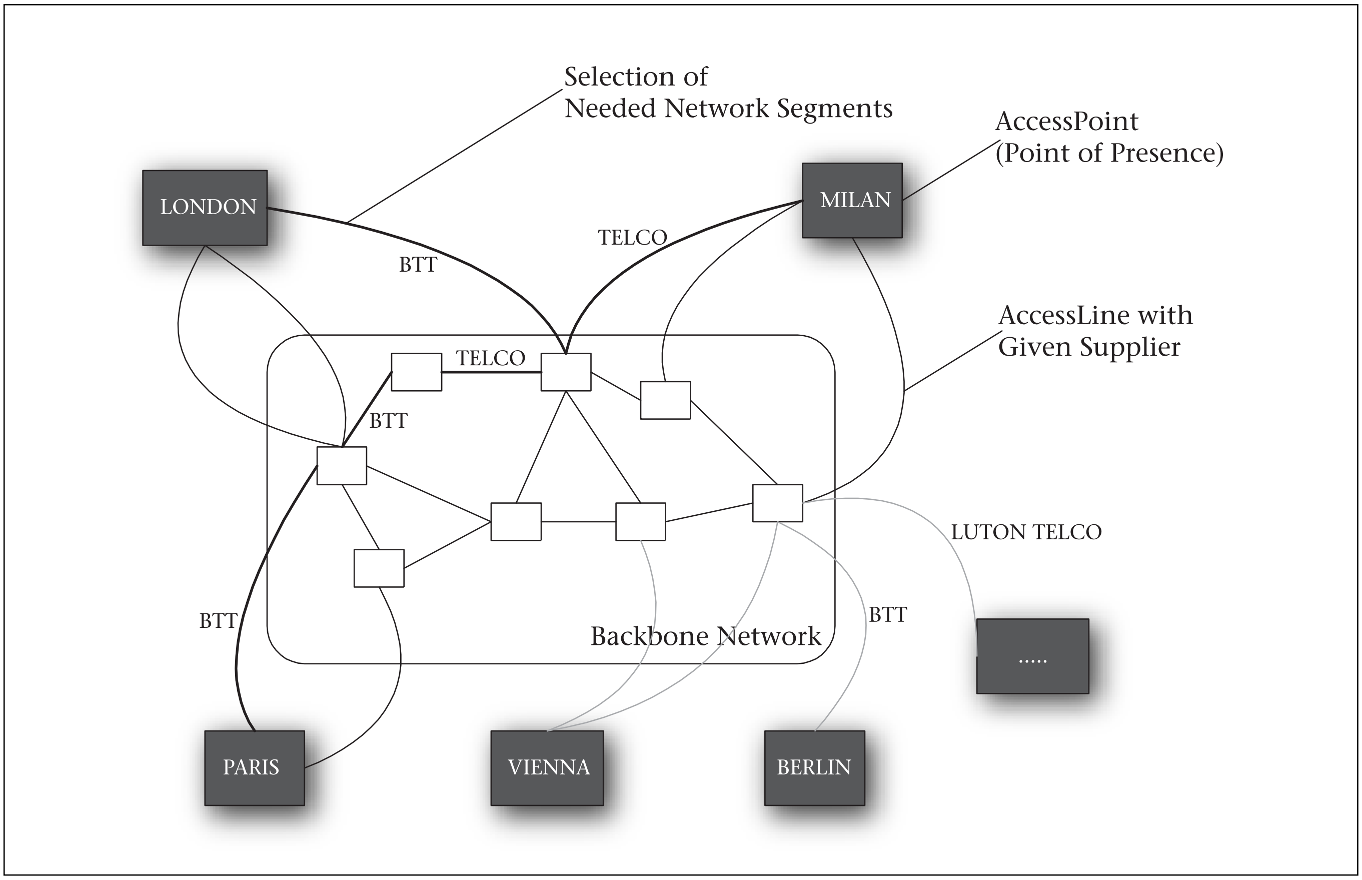

Figure 1. Reseller's View on IP-VPN Network.

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 vertexes within the backbone network represent interconnections between different companies' backbone segments.

evolving products, there is typically a shortage of trained sales people. Moreover, people playing different roles in their own organization, for example, 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. A technician might see it as a set of pluggable components, but an end user might be interested in the offered function or service. To satisfy the information needs of such different end users and adapt the configuration process to their capabilities, a dynamically generated user interface is needed (ACM 2002). 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 point of view, focused on the high-level properties offered by the final solution, for example, its quality and reliability.

Distributed reasoning: Recently, significant advances in the distributed AI, multiagent system, and distributed constraint-satisfaction areas have been made (Silaghi, Sam-Haroud, and Faltings 2000; Yokoo 2001). However, most current approaches do not fit well the characteristics of a distributed configuration. For example, configuration systems typically do not work autonomously or fully parallel, such as agents do. Moreover, they require 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 have to be used.

CAWICOMS WORKBENCH

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

Architecture of the CAWICOMS Configuration Shell

To develop a system satisfying the key requirements reported previously, different methodologies have to be integrated. Therefore, we de-

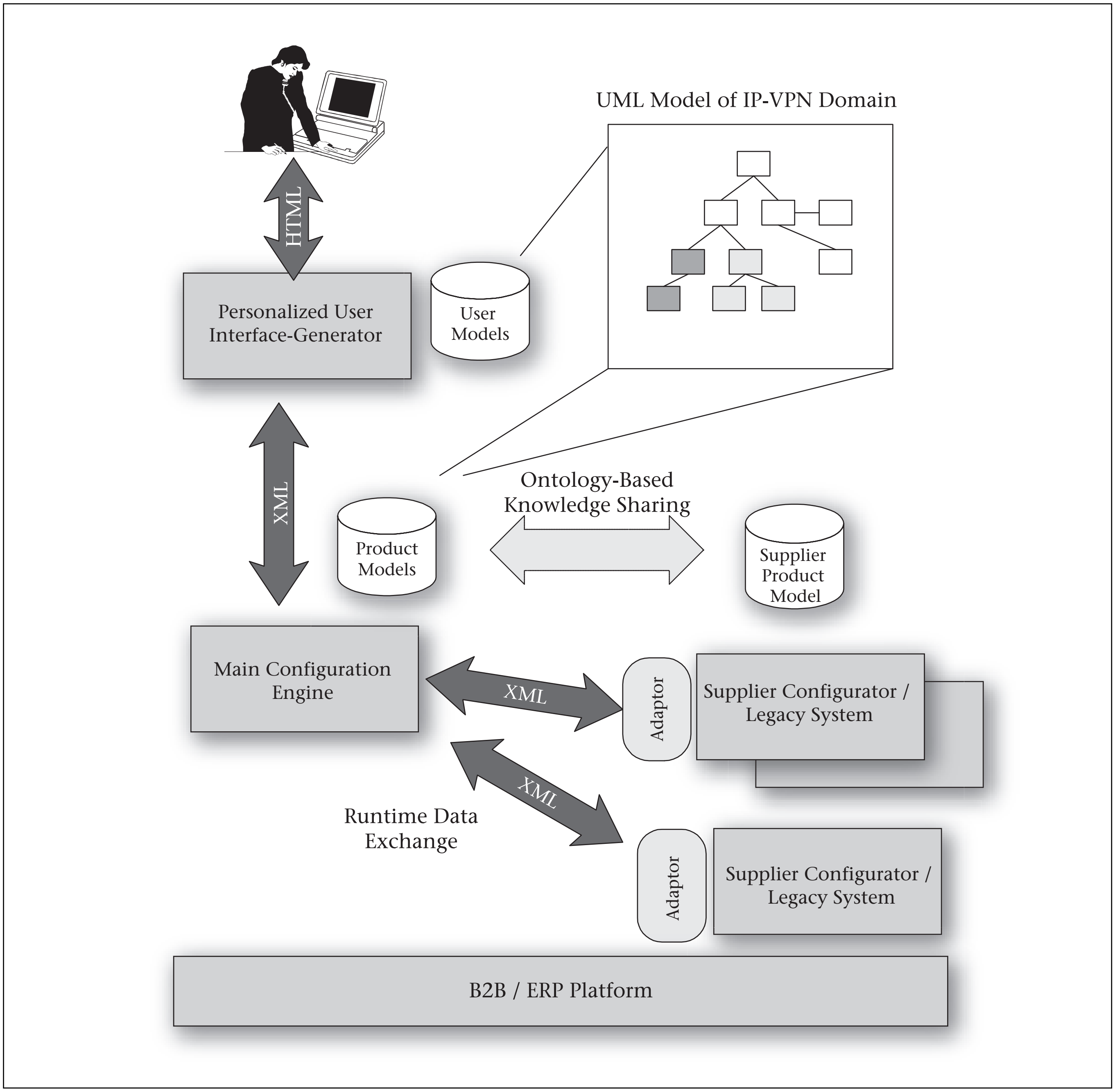

Figure 2. Architecture Overview.

signed the configuration shell offered by the CAWICOMS WORKBENCH as a modular system (figure 2), including components specialized in the execution of complex tasks, such as the computation of configuration solutions, the dynamic generation of the personalized user interface, and the communication with the suppliers.

According to the CAWICOMS architecture,[2] 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) (Junker 2001; Mailharro 1998), 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.

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 (Fleischanderl et al. 1998) and especially here because (1) knowledge has to be shared among participants in the supply chain, and different reasoning mechanisms and tools must be integrated, and (2) the adaptive user interface 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 (Fleischanderl et al. 1998; Mailharro 1998; Mittal and Frayman 1989). We developed a domain-independent metamodel and a UNIFIED MODELING LANGUAGE–based (UML) graphic notation with precise semantics (Felfernig, Friedrich, and Jannach 2000) that follows the standard modeling concepts of the configuration domain (Soininen et al. 1998). The resulting product models can be further enriched with constraints expressed in the OBJECT CONSTRAINT LANGUAGE (OCL) (Rumbaugh, Jacobson, and Booch 1998) 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 documents.[3] We also implemented tools for the automated document generation from commercial UML editors, such as RATIONAL ROSE.[4] These documents also serve as a starting point for further knowledge acquisition, that is, 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 Ardissono and Goy (2000b). Furthermore, 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, can be specified to evaluate the configurable items from a qualitative point of view (Carberry, Chu-Carroll, and Elzer 1999; Jameson et al. 1995). These properties have to be related to suitable sets of technical features to describe the impact of such features on the evaluation of the product with respect to the corresponding properties. For example, the inclusion of a certain component in a configuration can 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 (for example, 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, for example, managers, sales engineers, technical engineers, and end customers. This type of information is defined by a set of stereotypes (Rich 1989) that specifies the skills and interests of the users belonging to the various classes.

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 Ardissono et al. (2003, 2002). The user interface relies on the management of a user model that describes the capabilities and preferences of the individual end user (Schäfer 2001), the application of personalization techniques for the generation of customized content (Ardissono and Goy 2000b), and the dynamic generation of user interface pages (Ardissono and Goy 2000a). The methodologies underlying these two activities are outlined in the following subsections.

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 ….

Management of the User Model Some pieces of information about the individual end user, such as the type of company he/she represents, his/her nationality, and similar data, are retrieved by the system by explicitly questioning him/her. Other types of information, such as the user's interests and his/her knowledge about the configuration domain, are estimated by observing his/her behavior during the inter-

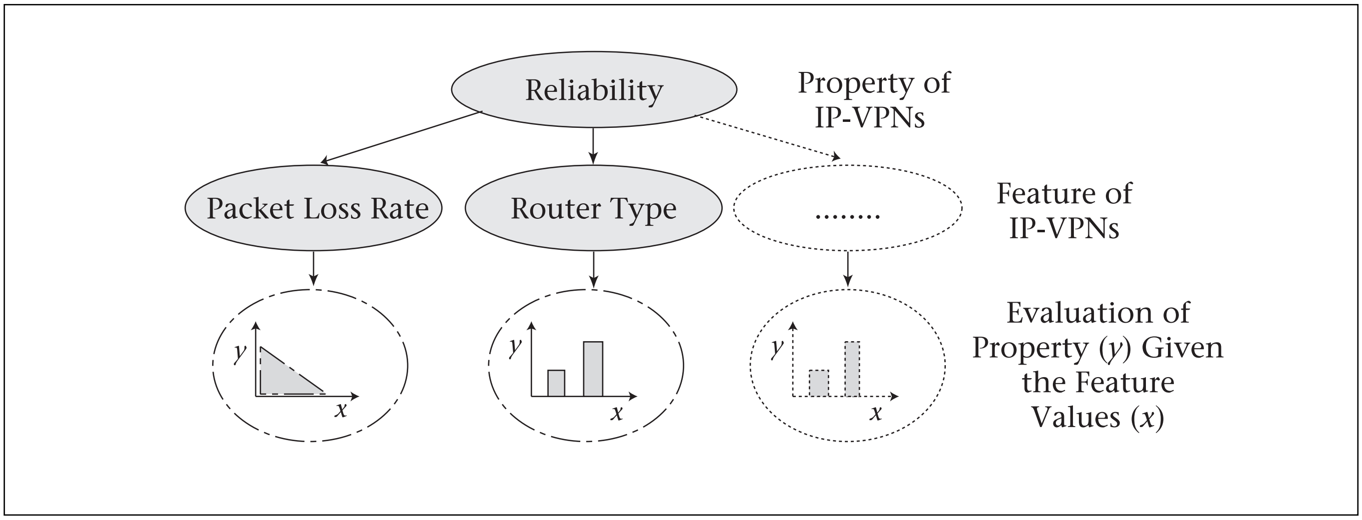

Figure 3. Evaluation of a Property of a Product-Service, Given Its Technical Features.

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 example, this picture shows that the packet loss rate influences the reliability of an IP-VPN in a negative way: the higher the rate, the lower the reliability of the network.

action with the system. In particular, the user can self-assess his/her own expertise and interests, but this information has to continuously be updated to take into account the fact that the user's skills and preferences might evolve. As shown in figure 4, 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 with the knowledge items and the properties. In CAWICOMS, the user model is managed in two phases:

When a new user interacts with the system, his/her interests and skills are estimated by exploiting the predictions provided by the stereotypes describing the user classes he/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 the has existed since the beginning of the interaction.

During the interactive configuration sessions, the estimates of the user's interests are updated by interpreting his/her behavior as an attempt to maximize the utility of a configuration solution (cf. Jameson et al. [1995]) in typical interaction situations, such as expression of interest, change of some proposed default value, and acceptance or rejection of a proposed configuration (Schütz and Schäfer 2001).[5]

The second point deserves further discussion. Being the user of a rational agent, we assume that he/she tries to maximize his/her own utility by setting the item features to satisfy his/her own needs in the best way. This assumption enables us to reason about the user's actions to assess his/her interests in the properties of the product-service, thus revising the user model accordingly. The system's functional knowledge about features and properties (figure 3) supports the identification of the user's interests in the properties related to the features his/she sets during the configuration process. If the user selects a specific value for a feature *F,* or he/she rejects a default value suggested 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 his/her a model for the evaluation of items based on the widely-adopted multiattribute utility theory (MAUT) (von Winterfeldt and Edwards 1986). 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 Schäfer (2001). 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 or 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 (Pearl 1988) are used to perform this task.

Personalization of the Interaction with the User The CAWICOMS configuration shell man-

```
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, that is, his/her preferences for specific feature values.

ages 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 user interface 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 that 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.[6]

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 he/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 (for example, a specific IP-VPN) by focusing the description on the features suitably fitting his/her interests and expertise (Ardissono and Goy 2000b). 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 he/she is very interested in. The less important features, those too complex for the user, and those outside his/her interests usually fall into the supplementary information to keep the presentation of solutions as synthetic as possible.

A rule-based mechanism is utilized 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 (Ardissono et al. 2002). The generation of the user interface is based on the JAVA SERVER PAGES (JSP) technology,[7] which enables the system to dynamically generate the content of the web pages on the basis of the status of the configuration process and the user's characteristics. Because the user model is continuously revised by the system, the application of personalization techniques and the dynamic generation of the user interface support a reactive adaptation of the interaction, so that the system can repair the possibly wrong personalization behavior (because of 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.

Figure 5. A Step in the Configuration of an IP-VPN.

The system asks the user to specify configuration parameters and suggests default values. The system can also elicit the required information in an indirect way, by asking the user about his/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.

Distributed Reasoning

Distributed configuration problem solving in a supply chain in CAWICOMS relies on three pillars, described in the following subsections: (1) ontology-based knowledge sharing, (2) open XML-based communication protocols and other interfaces, and (3) advanced constraint-satisfaction techniques.

Ontology-Based Knowledge Sharing Distributed problem solving approaches always require ontological commitments on domain-specific terms and conditions 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. To establish a common language for representing configurable products, we defined a hierarchy of related ontologies (Gruber 1992) and structured these ontological commitments, according to the proposal of Gangemi, Pisanelli, and Steve (1999), into three hierarchical levels: (1) the generic ontology level, (2) the intermediate level, and (3) the domain level.

In the *generic ontology level*, the basic representation concepts and modeling primitives are introduced. We used UML, whose metamodel defines the basic concepts such as Class and Association. 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 do-

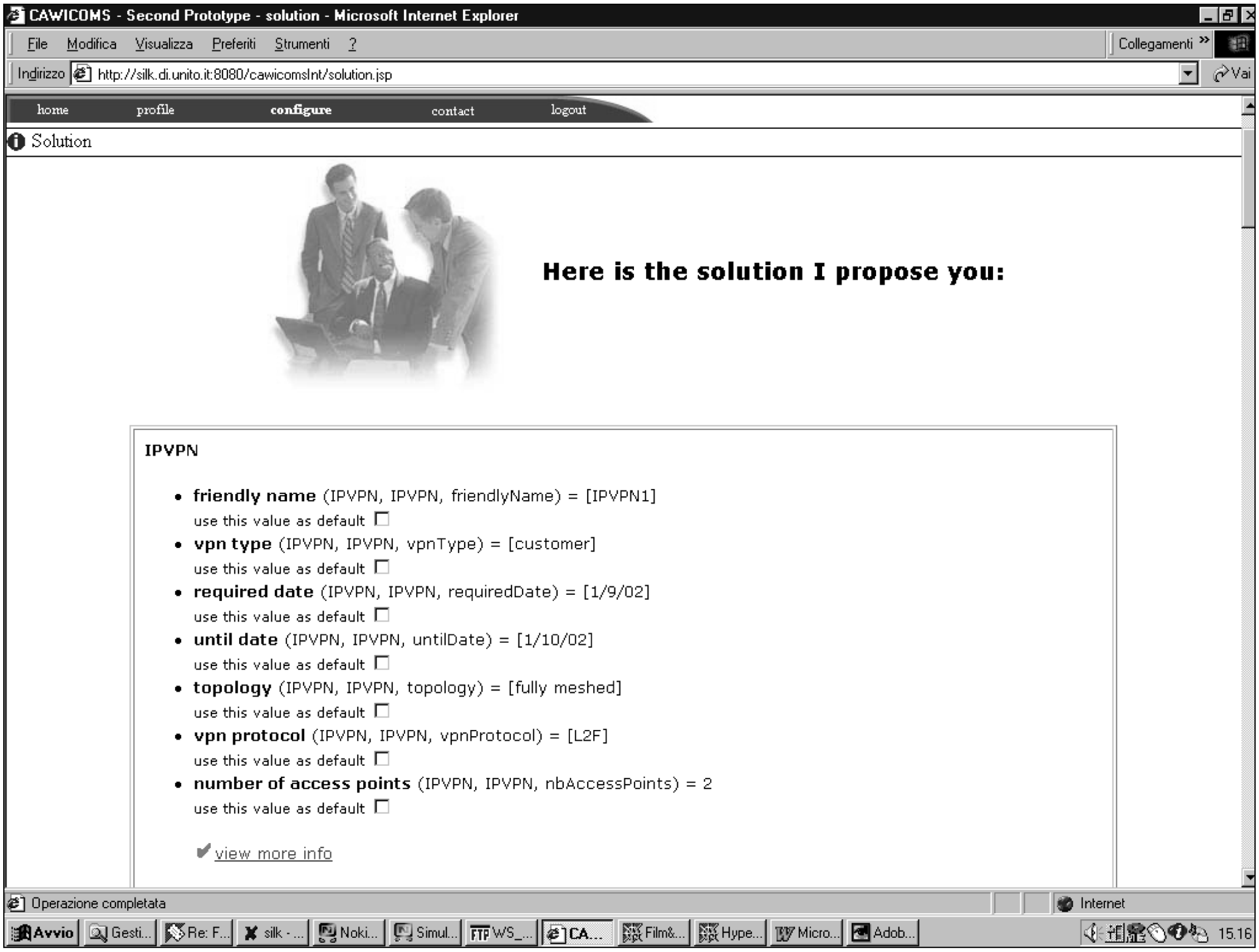

Figure 6. Presentation of a Configured IP-VPN.

The specification of the whole set of features is not visible in the picture because this type of service is rather complex. The 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.

main have been proposed (Felfernig, Friedrich, and Jannach 2000; Soininen et al. 1998). 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, for example, component types or ports, and is based on precise semantics that 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.[8]

The key innovation for seamless supply-chain integration in business-to-business (B2B) electronic-commerce (e-commerce) is to tackle the integration problem at the conceptual level; that is, 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, graphic 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 easily be transformed or mapped to the representation of an existing (knowledge-based) configuration system.

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 such as in distributed constraint satisfaction (Silaghi, Sam-Haroud, and Faltings 2000; Yokoo 2001). However, the focus of such approaches lies in intelligent, distributed reasoning and the process of setting up a multiagent scenario, but the acquisition and maintenance of the knowledge is neglected.

For nonstandard, proprietary products such as telecommunication switches (Felfernig et al. 2002c), knowledge is integrated in a knowledge engineering process, and the cooperation is based on long-term business relationships and contracts. However, in domains such as IP-VPN provision, where the configurable basic services are quite similar (for example, internet service provision) and there are many different suppliers for the same service, the goal is to come to an agreed-on product model for these basic services. Given such an agreement, the involved communities can establish electronic marketplaces (e-marketplaces) for configurable products in a specific domain. Therefore, one future goal is to extend existing e-marketplaces with a standardized (semantic) configuration web service (McIlraith, Son, and Zeng 2001) for its participants. For details on this approach, refer to Felfernig et al. (2002a).

Open-Communication Protocols and Platform A key issue for centralized configuration applications is the integration into the corporate software infrastructure such as enterprise resource planning (ERP) systems or B2B platforms. This issue is addressed in CAWICOMS and ILOG's JCONFIGURATOR by providing adequate interfaces,[9] for example, by allowing the configuration knowledge base to contain references to external data sources using a database connectivity interface; therefore, data do not have to be kept redundantly in the knowledge base. Moreover, the workbench is completely developed on the J2EE platform,[10] which is designed for easy integration of components into enterprise environments.

To support 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 and web services,[11] defines (1) a fixed set of methods with defined semantics for the configuration domain, such as creating components, setting values for parameters, initiating the search process, or retrieving results; (2) a mechanism to exchange complex data structures such as configuration results and a language for expressing navigation expressions within these data structures (compare XML-SCHEMA and XPATH); and (3) extensibility mechanisms for special domains and support for transactions for hypertext transfer protocol (HTTP)–based configuration sessions.

The main advantages of this approach lie in the independence from individual programming languages and in security-related issues, for example, the possibility of communicating over HTTP connections without the need for opening corporate firewalls for information processing–based connections.[12] Beside the definition of the protocol and the interfaces, the CAWICOMS WORKBENCH includes a set of corresponding software tools, for example, components for transaction support and tools for interface development based on standard technologies such as EXTENSIBLE STYLESHEET LANGUAGE (XSL) transformations and JAVA SERVER PAGES (JSPS).[13]

In figure 7, a schematic overview of knowledge integration and communication in a simple supply chain is given. 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 UML. 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 user interface 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 example, set a value for different parameters or initiate the search process at the supplier system. ILOG's WEBCONNECTOR component is utilized to parse and transform the protocol messages into the internal data structures of the underlying constraint-based configuration engine.

Distributed Problem Solving Historically, several reasoning techniques from the field of AI were developed and used for the core configuration tasks, such as consistency checking, automatically completing a partial configuration, or optimizing a solution. Compared to procedural or rule-based implementations of the

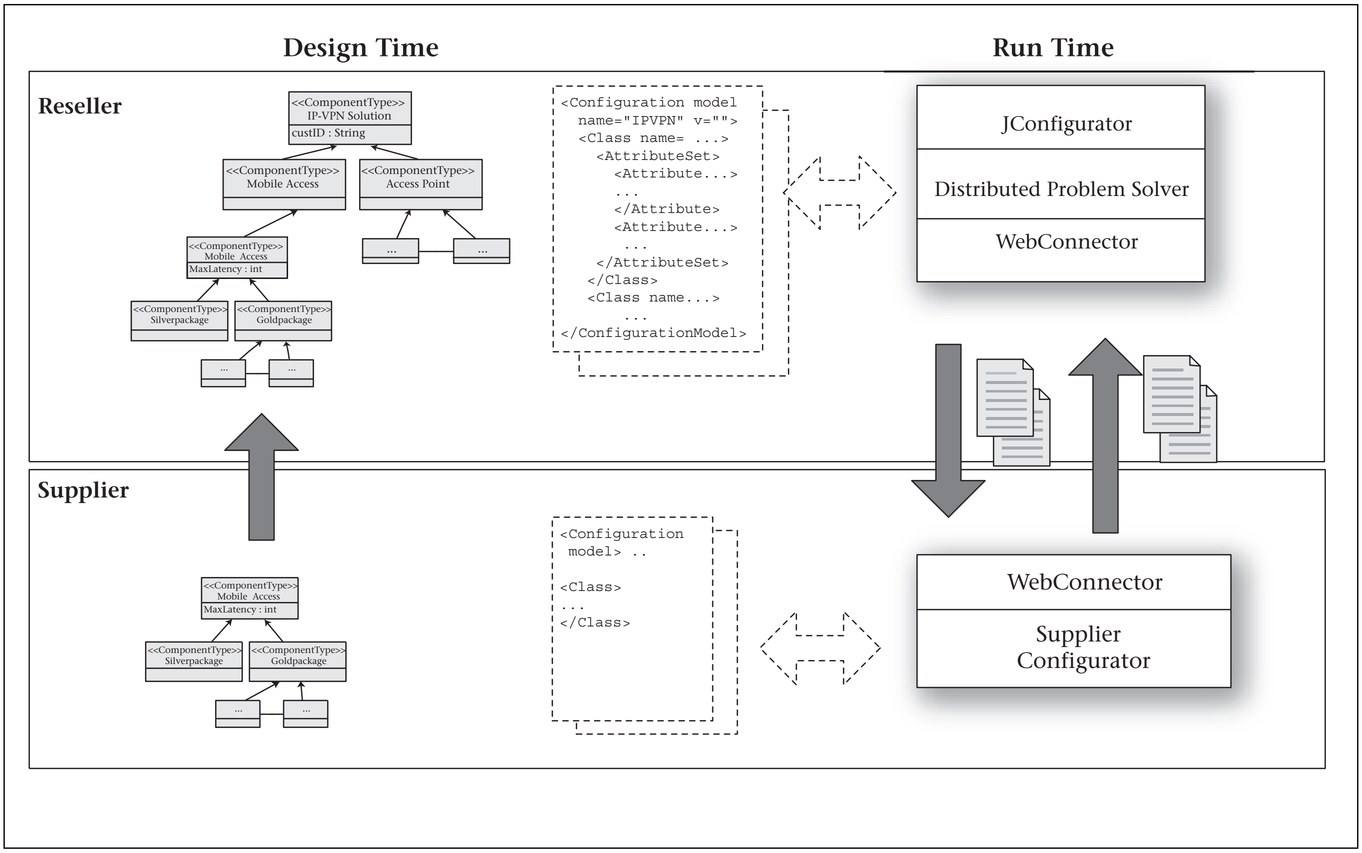

Figure 7. Schematic Knowledge Integration and Deployment Scenario for a Simple Supply Chain.

configuration logic, modern approaches, based on techniques such as 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 AI made substantial progress in the last decade, accompanied by the emergence of the agent paradigm, existing algorithms for distributed reasoning (for example, asynchronous backtracking [Yokoo 2001]) 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 multiagent 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 setup, 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 uses 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 (Fleischanderl et al. 1998) mechanism, whereby this core function is implemented in ILOG's commercial JCONFIGURATOR libraries. Each agent is capable of accepting requirements from its clients (using the defined protocol) and solving the local configuration problem. In cases where this local subproblem can only be solved by involving additional supplier systems, each mediating agent can, 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, where in each case the client system

Within the CAWICOMS framework, the novelty of the approach lies in the personalization of the interaction process for an underlying configuration search routine.

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 Felfernig et al. (2002a, 2002c); the formal requirements when implementing such algorithm variants are described in Felfernig et al. (2001). Beside these two reference implementations, the framework includes utility libraries for, for example, 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 (such as component generation) to steer the distributed search process. Finally, the use of an open-configuration protocol and message-exchange format allows us to integrate non-CAWICOMS configurators at the leaf nodes of the supply chain that 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.

Evaluation

The prototype system was evaluated a number of times throughout the development process, and the evaluation results were used to guide the system's further development. 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; and some were completely new to this area of technology. These users gave feedback by filling out a questionnaire 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 is a fast-growing technology startup company with offices across Europe and needs to purchase a VPN to replace its existing collection of ad hoc communications links. They decided to go to a reseller who uses 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 additional sites and dial-up access for mobile users. The results of the tests 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, we found the following to be true.

First, 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 nearly all the users because they can speed up the configuration process.[14]

Second, the elicitation of qualitative user requirements on the items (preference elicitation) represented valuable feedback for the less experienced user because 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.

Third, the users found that the explanation capabilities of the system (which provides the reasons for a failure by highlighting the violated constraints) are useful in understanding the causes of the failures in the configuration and repairing such failures by selecting different values for the problematic features.

Fourth, the possibility of suspending a configuration session and resuming it later was useful to the users who participated in the most complex test scripts.

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

Sixth, 10 percent 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. 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 a failure (choosing a different value for the problematic features) but also allows the user to change his/her mind before the solution is generated. Although we could not test the usability of this facility, we are aware that it might challenge the user who is unfamiliar with configuration technologies. Nevertheless, it represents a powerful tool, helping the expert users to configure items in a quicker and more flexible way.

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 for openness and integratability in a dynamically changing, heterogeneous environment was taken into account, complying with existing and emerging standards and relying on state-of-the-art software platforms and XML-based protocols.

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

From a research perspective, this project provided an excellent opportunity for integrating the results separately achieved by the configuration systems and intelligent user interfaces research communities. To this point, they had not joined 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, this attempt was the first 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. One 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, 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 and generalizing these 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 base technologies such as the choice of the programming language and environment, we made good use of state-of-the-art technology in 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 ex-

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.

ploiting standard tools for, for example, dynamic web page generation or XML parsing, these technologies have the advantage that many of the software tools from the surrounding information technology infrastructure such as databases or e-commerce platforms, can easily be integrated. Even more, the use 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, for example, 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 ongoing trends in the market: (1) personalization, (2) mass customization, and (3) 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. In addition, 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: First, once a customer is identified in a configuration session, we can retrieve the user profile from the CRM knowledge base. Second, we can upload additionally learned or entered information into the CRM system. Such integration improves the communication with the user that potentially hops 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 to discover new market opportunities.

The mass-customization approach, which can also be seen as personalization of the products themselves, evolved over the last two decades and today is the standard business model for many industrial sectors. As a result, every major vendor of Enterprise Software (such as 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 this area in two ways: 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 accom- panying services rather than the product itself. However, the techniques developed in the CAWICOMS framework advance current B2B e-commerce technology. Current systems support cooperation in the supply chain by exchanging orders, publishing product catalogs, or 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.

Acknowledgments

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

This work was partly funded by the European Union through the IST Programme under contract IST-1999-10688.

Notes

1. BTexact Technologies, Telecom Italia, ETIS, ILOG S.A.

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

3. See www.w3.org for more information.

4. See www.rational.com for details.

5. We omit the description of the estimation of the user's expertise, which is performed as described in Jameson (1990).

6. This is in contrast to the internal view of the configuration engine, which can be completely flat, as it happens for constraint-based engines.

7. See java.sun.com/products/jsp/ for details.

8. Note that actual configurations can be seen as instance models of the domain-specific model.

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

10. Sun Microsystems. 2002. JAVA Web Pages. java.sun.com.

11. See www.w3.org for details.

12. ILOG's WebConnector tool kit also supports java-

based communication for homogeneous environments.

13. Sun Microsystems. 2002. JAVA Web Pages. java.sun.com.

14. The CAWICOMS system can run in personalized and nonpersonalized modes. In the personalized mode, the system provides suggestions for about half the features to be configured.

15. The project's budget was about €3.5 million.

References

ACM. 2002. *Communications of the ACM* (Special Issue on the Adaptive Web) 45(5).

Ardissono, L., and Goy, A. 2000a. Dynamic Generation of Adaptive Web Catalogs. In *Adaptive Hypermedia and Adaptive Web-Based Systems*, eds. O. Stock, P. Brusilovsky, and C. Strapparava, 5–16. Lecture Notes in Computer Science 1892. Berlin: Springer Verlag.

Ardissono, L., and Goy, A. 2000b. Tailoring the Interaction with Users in Web Stores. *User Modeling and User-Adapted Interaction* 10(4): 251–303.

Ardissono, L.; Felfernig, A.; Friedrich, G.; Goy, A.; Jannach, D.; Meyer, M.; Petrone, G.; Schaefer, R.; Schütz, W.; and Zanker, M. 2002. Personalizing Online Configuration of Products and Services. Paper presented at the Fifteenth European Conference on Artificial Intelligence (ECAI-02), 21–26 July, Lyons, France.

Ardissono, L.; Goy, A.; Holland, M.; Petrone, G.; and Schäfer, R. 2003. Customising the Interaction with Configurators. Paper presented at the International Conference on User Modeling (UM-2003), 22–26 July, Pittsburgh, Pennsylvania.

Benyon, D. 1993. Adaptive Systems: A Solution to Usability Problems. *User Modeling and User-Adapted Interaction* 3(1): 65–87.

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

Brusilovsky, P. 1996. Methods and Techniques of Adaptive Hypermedia. *User Modeling and User-Adapted Interaction* 6(2-3): 87–129.

Carberry, S.; Chu-Carroll, J.; and Elzer, S. 1999. Constructing and Utilizing a Model of User Preferences in Collaborative Consultation Dialogues. *Computational Intelligence* 15(3): 18–217.

Felfernig, A.; Friedrich, G.; and Jannach, D. 2000. UML as Domain-Specific Language for the Construction of Knowledge-Based Configuration Systems. *International Journal of Software Engineering and Knowledge Engineering (IJSEKE)* 10(4): 449–469.

Felfernig, A.; Friedrich, G.; Jannach, D.; and Zanker, M. 2002a. Semantic Configuration Web Services in the CAWICOMS Project. In *Proceedings of the First International Semantic Web Conference,* 192–205. Lecture Notes in Computer Science 2342. New York: Springer.

Felfernig, A.; Friedrich, G.; Jannach, D.; and Zanker, M. 2002b. Web-Based Configuration of Virtual Private Networks with Multiple Suppliers. In *Proceedings of the Seventh International Conference on Artificial Intelligence in Design,* 41–62. Cambridge, United Kingdom: Kluwer.

Felfernig, A.; Friedrich, G.; Jannach, D.; and Zanker, M. 2001. Toward Distributed Configuration. In *Proceedings of the Twenty-Fourth German/Ninth Austrian Conference on Artificial Intelligence,* 198–212. Heidelberg, Germany: Springer-Verlag.

Felfernig, A.; Friedrich, G.; Jannach, D.; Russ, C.; and Zanke, M. 2002c. Multisite Product Configuration of Telecommunication Switches. In *Proceedings of the Twentieth IASTED International Multiconference—Applied Informatics,* 113–118. Chicago, Ill.: Acta.

Fink, J., and Kobsa, A. 2000. 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.

Fleischanderl, G.; Friedrich, G.; Haselböck, A.; Schreiner, H.; and Stumptner, M. 1998. Configuring Large Systems Using Generative Constraint Satisfaction. *IEEE Intelligent Systems* 13(4): 59–68.

Gangemi, A.; Pisanelli, D.; and Steve, G. 1999. An Overview of the ONIONS Project: Applying Ontologies to the Integration of Medical Terminologies. *Data and Knowledge Engineering* 31(2): 183–220.

Gruber, T. 1992. ONTOLINGUA: A Mechanism to Support Portable Ontologies. Technical Report, KSL 91-66, Knowledge Systems Laboratory, Stanford University.

Jameson, A. 1990. Knowing What Others Know: Studies in Intuitive Psychometrics. Ph.D. dissertation, University of Amsterdam.

Jameson, A.; Schäfer, R.; Simons, J.; and Weis, T. 1995. Adaptive Provision of Evaluation-Oriented Information: Tasks and Techniques. In Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI), 1886–1893. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence.

Junker, U. 2001. Preference Programming for Configuration. Paper presented at the Fourteenth International Conference on Artificial Intelligence Configuration Workshop, 4–10 August, Seattle, Washington.

McGuiness, D. L., and Wright, J. R. 1998. Conceptual Modeling for Configuration: A Description Logic–Based Approach. *AI in Engineering, Design, and Manufacturing (AI EDAM)* 12(98): 333–344.

McIlraith, S.; Son, T. C.; and Zeng, H. 2001. Semantic Web Services. *IEEE Intelligent Systems* 16(2): 46–53.

Mailharro, D. 1998. A Classification and Constraint-Based Framework for Configuration. *AI in Engineering, Design, and Manufacturing* 12(98): 383–397.

Mittal, S., and Frayman, F. 1989. Toward a Generic Model of Configuration Tasks. In Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, 1395–1401. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence.

Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Francisco, Calif.: Morgan Kaufmann.

Piller, F., and Schaller, C. 2002. Individualization-Based Collaborative Customer Relationship Management: Motives, Structures, and Modes of Collaboration for Mass Customization and CRM. Technical Report, 29, Department of General and Industrial Management, Technische Universität München.

Pine II, B. J.; Victor, B.; and Boynton, A. C. 1993. Making Mass Customization Work. *Harvard Business Review* 71(5): 109–119.

Rich, E. 2000. Stereotypes and User Modeling. In *User Models in Dialog Systems,* eds. A. Kobsa and W. Wahlster, 35–51. Berlin: Springer Verlag.

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

Rumbaugh, J.; Jacobson, I.; and Booch, G. 1998. *The UNIFIED MODELING LANGUAGE Reference Manual*. Reading, Mass.: Addison-Wesley.

Schäfer, R. 2001. Rules for Using Multiattribute Utility Theory for Estimating a User's Interests. Paper presented at the Ninth Workshop Adaptivität und Benutzermodellierung in Interaktiven Softwaresystemen, 8–10 October, Dortmund, Germany.

Schütz, W., and Schäfer, R. 2001. Bayesian Networks for Estimating the User's Interests in the Context of a Configuration Task. Paper presented at the UM2001 Workshop on Machine Learning for User Modeling, 13–17 July, Sonthoven, Bavaria, Germany.

Silaghi, M. C.; Sam-Haroud, D.; and Faltings, B. 2000. Asynchronous Search with Aggregations. In Proceedings of the Seventeenth National Conference on Artificial Intelligence, 917–922, Menlo Park, Calif.: American Association for Artificial Intelligence.

Soininen, T.; Tiihonen, J.; Männistö, T.; and Sulonen, R. 1998. Toward a General Ontology of Configuration. *AI Engineering Design Analysis and Manufacturing* (Special Issue on Configuration Design) 12(4): 357–372.

Stumptner, M. 1997. An Overview of Knowledge-Based Configuration. *AI Communications* 10(2): 111–126.

von Winterfeldt, D., and Edwards, D. 1986. *Decision Analysis and Behavioral Research*. Cambridge, U.K.: Cambridge University Press.

Yokoo, M. 2001. *Distributed Constraint Satisfaction*. New York: Springer Verlag.

Liliana Ardissono is an associate professor at the Dipartimento di Informatica of the Università degli Studi di Torino, where she graduated and obtained her Ph.D in computer science. Her research interests include adaptive hypermedia, multiagent systems and web services. Her e-mail address is liliana@di.unito.it.

Alexander Felfernig is a research assistant in computer science at the University Klagenfurt, Austria, where he obtained his M.S. and Ph.D. 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. His e-mail address is alexander.felfernig@uni-klu.ac.at.

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 Ph.D. and M.S. in computer science from the Vienna University of Technology. His e-mail address is gerhard.friedrich@uni-klu.ac.at.

Anna Goy is a researcher at the Dipartimento di Informatica of the Università degli Studi di Torino. She obtained a Ph.D. in cognitive science at the same university, where she participates in the research and development activity of the Intelligent User Interface Group. Her research interests include adaptive hypermedia and web technologies. Her e-mail address is goy@di.unito.it.

Dietmar Jannach works as a research assistant in computer science at the University Klagenfurt, Austria, where he obtained his M.S. and Ph.D. 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. His e-mail address is dietmar.jannach@uni-klu.ac.at.

Giovanna Petron is technical leader for research projects within the Intelligent User Interface Group at the Dipartimento di Informatica of the Università degli Studi di Torino. Her research interests include multiagent systems and web services. Her e-mail address is giovanna@di.unito.it.

Ralph Schäfer obtained his Ph.D. in computer science from Saarland University. He started working as a senior researcher at the German Center for Artificial Intelligence (DFKI). He was the project leader of the CAWICOMS project. His personal research interests include adaptive user interfaces, user modeling, and reasoning under uncertainty. His e-mail address is ralph.schaefer@r77.de.

Markus Zanker is a research and teaching assistant in computer science at the University Klagenfurt, Austria, where he obtained his Ph.D. and masters in applied informatics and 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. His e-mail address is markus.zanker@uni-klu.ac.at.

Call for Participation

AAAI-04 Student Abstract & Poster Program

Nineteenth National Conference on Artificial Intelligence
July 25-29, 2004 ■ San Jose, California
Sponsored by the American Association for Artificial Intelligence

The AAAI-04 committee invites submissions to the student abstract and poster program. The goal of this program is to provide a forum in which students can present and discuss their work during its early stages, meet some of their peers who have related interests, and introduce themselves to more senior members of the field. The program is open to all pre-Ph.D students. Nonstudent advisors or collaborators should be acknowledged appropriately, as coauthors or otherwise. However, students are requested to honor the spirit of the program by submitting only work for which they are primary investigators.

Summary of Dates
Authors must submit six (6) printed copies of a two-page abstract describing their research, to arrive at the AAAI office no later than January 23, 2004. We also request that authors submit the URL of a location where reviewers can access complementary material about the student's research. The URL is critical to reviewers because of the brevity of the hard-copy submission.

Notification of acceptance or rejection of submitted abstracts will be mailed to the author by March 19, 2004. Camera-ready copy of accepted abstracts will be due by April 6, 2004.

Summary of Format
Submissions must be printed on 8 1/2 x 11 inch or A4 paper using 12 point type (10 characters per inch for typewriters). Each page must have a maximum of 38 lines and an average of 75 characters per line (corresponding to the LaTeX article-style, 12 point). All abstracts must be no more than two pages, not including the bibliography. The first two pages must include the following: title; the primary author's full name, affiliation, postal address, phone number, URL (if available), and e-mail address; all coauthors' full names and affiliations; text; and any figures, tables, or diagrams. Up to one additional page may be used exclusively for the bibliography if necessary. Papers exceeding the specified length and formatting requirements are subject to rejection without review.

Submissions to AAAI-04 or Other Conferences
Students are free to submit abstracts for work reported in a regular paper submitted to the AAAI-04 or another conference, but not for work that has already been published. Abstracts will be accepted or rejected for the student session regardless of the outcomes of related paper submissions.

Publication
Accepted abstracts will be allocated two (2) pages in the conference proceedings. Students will be required to transfer copyright of the abstract to AAAI.

Poster Session
Accepted abstracts will be allocated presentation time and space in the student poster display area at the conference. Student authors of accepted abstracts must agree to prepare a poster representing the work described in their abstracts and to be available to discuss their work with visitors during their allocated time in the student poster display area.

Student Abstract Submissions & Inquiries
Please send abstracts to:

AAAI-04 Student Abstracts
American Association for Artificial Intelligence
445 Burgess Drive
Menlo Park, CA 94025-3442 USA

Registration and call clarification inquiries may be sent to ncai04@aaai.org. All other inquiries and suggestions should be directed to the Student Abstract and Poster Program Cochairs:

Avi Pfeffer
Division of Engineering and Applied Science
Harvard University
251 Maxwell Dworkin
33 Oxford Street
Cambridge, MA 02138-0001
Telephone: (617) 496-1876
Fax: (617) 496-1066
E-mail: avi@eecs.harvard.edu

Mark Craven
Department of Biostatistics and Medical Informatics
University of Wisconsin, Madison
1300 University Avenue
Madison, WI 53706
Telephone: (608) 265-6181
Fax: (608) 265-7916
E-mail: craven@biostat.wisc.edu

Call for Proposals

Intelligent Systems Demonstrations

Nineteenth National Conference on Artificial Intelligence
July 25-29, 2004 ■ San Jose, California

Sponsored by the American Association for Artificial Intelligence

The AAAI Intelligent Systems Demonstrations program showcases state-of-the-art AI implementations and provides AI researchers with an opportunity to show their research in action. Implemented intelligent systems allow us not only to experimentally validate AI research, but also to make AI research accessible to each other, to the broader scientific community, and to the public at large.

Researchers from all areas of AI are encouraged to submit proposals to demonstrate their systems. Submissions will be evaluated on the basis of their innovation, relevance, scientific contribution, presentation, and "user friendliness," as well as potential logistical constraints. This program is primarily to encourage the early exhibition of research prototypes, but interesting mature systems and commercial products are also eligible (commercial sales and marketing activities are not appropriate in the Intelligent Systems Demonstration program, and should be arranged as part of the AAAI-04 Exhibits program). Demonstrations that can be used by the audience and/or that interact with the audience are particularly encouraged.

Demonstration systems should be available as much as possible during the conference exhibition. Each demonstration will have a scheduled and advertised time during which it is the "featured" demonstration. Each accepted demonstration system must be attended by at least one knowledgeable representative (preferably an architect of the system) who will be available to answer in-depth technical questions at scheduled times.

Demonstration proposals must be made electronically using the forms at the AAAI web site. Please check www.aaai.org for further details after October 1, 2003. In addition to contact information, proposals must include the following, all of which must be submitted via the internet:

1. A two-page description in AAAI paper format of the technical content of the demo, including credits and references. These descriptions will appear in the conference proceedings, space permitting.
2. A 150-word summary of the demo in plain text. Please include title, demonstrator names, and affiliation(s). This summary will be used to compile a program for the demonstrations. Please try to keep the descriptions under the 150-word limit.
3. A demo storyboard of not more than six pages total or an informal video of the demo (in MPEG or Quicktime format), that describes how the demonstration will proceed (as opposed to the technical merits of the research being demonstrated). This is the committee's primary method of evaluating your proposal. Please emphasize the elements that make your demonstration exciting and interesting.
4. A detailed description of hardware and software requirements. Demonstrators are encouraged to be flexible in their requirements (possibly with different demos for different logistical situations). Please state what you can bring yourself and what you absolutely must have provided. Generally speaking, we can provide computer monitors and peripherals such as TVs and VCRs, as well as a network connection. Each demonstration will be assigned a booth in the Exhibit Hall.

Demo proposals must be received in their entirety including any supporting materials by Friday, February 20, 2004. Authors will be notified of acceptance by March 19, 2004.

We especially hope that authors of papers accepted for presentation at the conference technical program will be able to demonstrate their research in the AAAI Intelligent Systems Demonstration Program. To present a system demonstration, however, the authors must still submit a proposal conforming to the above requirements by the Demonstration program deadline. Submitters who wish to demonstrate intelligent mechanical systems that interact with the real world (aka "robots") should direct their efforts toward the Robot Exhibition.

If you have any questions or comments about the AAAI Intelligent Systems Demonstration program, we encourage you to address them to the program organizer, Chris Welty (welty@us.ibm.com).