User-Adaptive Configuration of Products and Services

L. Ardissono[1], G. Friedrich[2], A. Goy[1], M. Holland[3], G. Petrone[1], C. Russ[2] and R. Schäfer[4]
(1) Dipartimento di Informatica, Università di Torino, Italy
{liliana, goy, giovanna}@di.unito.it
(2) Computer Science and Manufacturing Research Group, University of Klagenfurt, Austria
{gerhard.friedrich, russo}@ifit.uni-klu.ac.at
(3)BTExact Technologies, UK, matthew.holland@bt.com
(4) Ralph.Schaefer@r77.de

Abstract

Configuration systems are key tools in the sales of complex services and products because they support the dynamic generation of solutions. However, such systems, typically designed for expert users, are too technical for the average customer of a Web store. As a solution to this issue, we present a model for the integration of Intelligent User Interfaces and Configuration techniques. This model supports the development of adaptive configuration systems, which help the user in the selection of the features to be set and present the solutions according to the user's interests and expertise.

1 Introduction

Although efficient recommendation techniques have been developed to customise the suggestion of items off the shelf [11; 13; 4; 6], such techniques do not support the adaptive configuration of items, which is essential to comply with the customer's requirements when purchasing complex products, or registering for complex services. The main problem is that the system has to tailor the problem solving process generating personalised items to the customer's preferences. Efficient configuration systems have been developed, which can handle configurable items in large-scale domains; e.g., [12; 10; 9]. However, such systems are designed for one typical user class, neglecting the fact that users differ in needs, knowledge about the product details, and expertise. Indeed, many configurators are product oriented in their communication process, ignoring the needs of large user groups (e.g., goal oriented customers or help seeking waverers) who cannot deal with (and are not interested in) product details. Product oriented configurators require that the user knows all the details about the items to be configured. One way to enhance the usability of such systems is to extend them with *user-adaptive interfaces* guiding the user through the configuration process in a personalised way [8; 1].

Within the CAWICOMS[1] project, we have developed an intelligent user interface (the CAWICOMS frontend) for a configuration engine, aimed at mediating the interaction with the user. The CAWICOMS frontend customises different aspects of the interaction. While the user selects the features of the item to be configured, the system assists her by eliciting requirements in a non-technical way and by possibly suggesting the values best fitting her requirements. Moreover, the system explains in a friendly way the possible failures in the configuration process, in order to help the user to choose different features. Finally, the system personalises the presentation of the configuration solutions by focusing on the most interesting information. We have exploited the CAWICOMS frontend in two application domains: the configuration of (1) Telecommunications Switches (i.e., hardware devices used to connect telecommunication networks) and (2) of IP-based Virtual Private Networks (IP-VPNs). In the following, we use the latter domain as concrete example.

Section 2 describes the representation of the domain-specific knowledge about products/services. Section 3 details the integration of configurators with intelligent user interfaces. Section 4 shows the management of personalisd configuration interactions. Section 5 describes our approach for generating explanations. Section 6 presents some evaluation results and Section 7 concludes the paper.

2 Representation of Knowledge about Products and Services

The CAWICOMS frontend reconciles the needs of the underlying configuration engine with the user's interaction requirements. The addition of an adaptive interface to a configuration engine requires that the implementation-oriented view of the configurable items has to be extended with a user-oriented view, focused on the properties offered by such items.

The technical knowledge about products and services exploited by the configuration engine underlying the CAWICOMS system relies on the well known *component-port* approach for configuration [12]. In that approach, configurable items are described, at the conceptual level, as sets of component types, related to one another by relations (ports). Components are further characterised by attributes corresponding to product features which are assigned to a specific value during problem solving. Usually a set of possible values constrains the valid assignments (e.g., packetLoss: 1..10). A knowledge base further constrains the allowed attribute-

[1]CAWICOMS is the acronym for "Customer-Adaptive Web Interface for the Configuration of Products and Services with Multiple Suppliers"; see http://www.cawicoms.org.

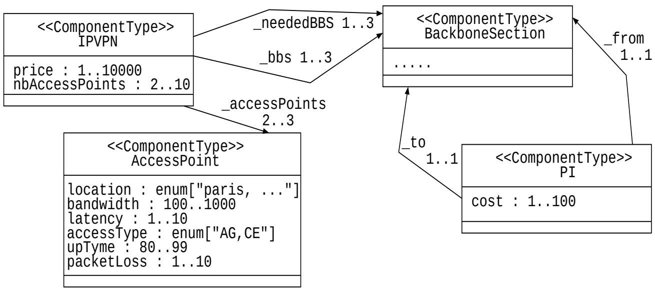

Figure 1: Portion of the IP-VPN product model.

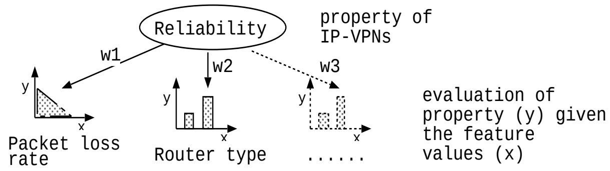

Figure 2: Evaluation of the reliability of an IP/VPN, given its technical features.

value combinations, port connections, and instantiations of the component types. We refer to such conceptual representation as the *product model*; Figure 1 shows a portion of the conceptual representation of IP-VPNs. The components reflect the internal organisation of the items, e.g., the physical elements assembled in a product. Individual items are described as instances of the conceptual model.

To support a user-oriented management of the configuration process, we have extended this representation with personalisation-oriented information. Specifically:

- We separated the features and components to be presented in the user interface from those to be exclusively handled by the configuration engine.
- We enriched the representation of features with the specification of the *type of information* they convey: e.g., technical, economic and other types of information. Moreover, we specified a *complexity level*, as an estimate of the degree of expertise required to understand the feature meaning. Furthermore, we specified a *criticality level*, needed to identify features whose value should be explicitly acknowledged by the user, and an *explanation* of their meaning.
- We grouped the features in homogeneous groups that intuitively correspond to *components, from the user's point of view*. The definition of such groups enabled us to provide the user with a structured view of the product/service, which may differ from the implementation-oriented representation specified in the product model (used by the configuration engine). For instance, the groups may be coarser-grained than the components in the product model, hiding implementation details.
- We introduced the representation of properties aimed at evaluating the items from a qualitative point of view. The specific properties to be used depend on the application domain. In the configuration of IP-VPNs, we defined properties such as the performance and the reliability of the switches/networks.

The properties are aimed at enabling the system to elicit functional requirements about products from the user. However, such requirements have to be translated to technical requirements guiding the configuration process. To this purpose, the defined properties have to be related to the technical features, in order to describe the impact of such features on the evaluation of items 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 of the item to be configured.

Figure 2 shows the evaluation of the reliability of an IP-VPN with respect to the packet loss rate, the type of router, and so forth. The impact of a set of features on the evaluation of a property is defined by following the approach defined in the Multiattribute Utility Theory [15]:

- For each property,[2] *Value functions* are defined to specify the dependencies between the individual features and the property values. Each value function provides a *value* for the property, representing the contribution to the property provided by the feature. For instance, Figure 2 shows that the packet loss rate negatively influences the reliability of an IP-VPN: the higher is the rate, the lower is the reliability of the network.
- The *value* of the individual features is combined to obtain the overall evaluation of the property by applying an integration function. We exploited the weighted sum of the values as an integration function (ad hoc functions can be defined for handling non additive cases).[3]

It should be noticed that, although the specification of the user-oriented information for a configuration domain is a knowledge intensive task, it is leveraged by a specialised editor developed by the CAWICOMS consortium; see [3]. Starting from the specification of the product model, the editor enables the service developer to extend components and attributes with information about their criticality, complexity level, and so forth. Moreover, the editor supports the definition of properties and specification of the relation between properties and technical features in the product model.

3 Integrating User Interaction with Configuration

The intelligent user interface mediates between user and configurator. In order to elicit user requirements and to present possible solutions (see Figures 3, 4 and 5), the interface exploits the services provided by the configuration problem solving component. However, to put the user in control of the interaction, the system enables her to choose which components of the product/service she wants to configure first, which ones she is not interested in specifying, and so forth.

[2]Properties denote dimensions in MAUT.

[3]Given the evaluation of a product/service with respect to its properties, the overall evaluation is determined by combining the property values in a weighted sum, according to MAUT.

In the CAWICOMS project we adopted the following general view of these activities. The configurator starts from a set of user requirements (e.g., needed components, preferred feature values) and searches for a complete and consistent solution such that all constraints of the product knowledge base are satisfied, all features are assigned, and no component instance is missing ([7] gives a formal definition). The interesting point is the elicitation of the user requirements which is performed by the user interface by following the user-oriented representation of products and services described in Section 2. The user is provided with a list of components to be configured and she may choose which ones she wishes to configure first. For instance, when configuring an IP-VPN, the user might start from the high-level specification of the network, by defining the number of access points. Then, she might configure the individual access points, and the backbone access, and so forth.

During this search process, decisions must be drawn, e.g., which value should be assigned to a feature or which component type should be instantiated. Some of these decisions could be made by the user (henceforth, possible user decisions). The rest of these choices must be made autonomously by the configurator. Note that we are talking about *possible user decisions* because if enough information about the user's interests and requirements is available to the system, and the choices are not too critical, the frontend may make the decision without involving the user at all.

The user interface enables the configuration system to work in two modes:

- In the *interactive mode* (propagate mode), requirements are elicited from the user and passed to the configuration system. This process may have three types of output:
  a) a complete and consistent configuration (if the user's decisions were sufficient to generate it);
  b) a set of user decisions to be made (i.e., features and their allowed values);
  c) if the user's requirements are inconsistent, sets (*conflict sets*) of conflicting requirements together with the part of the configuration knowledge base needed to generate an inconsistency. The information is used to generate explanations for the user; see Section 5.
- In the *auto-complete mode* all decisions are made by the configurator. The output is either a complete and consistent configuration or a set of conflicts. The user is allowed to invoke the auto-complete mode at any stage of the interaction with the system. Thus, she may delegate the system to configure the whole product/service, or she may work in interactive mode for a while, to configure the components she is most interested in, and let the configuration system do the rest of the job.

The communication flow between the intelligent user interface and the configurator is the following:

1. The configurator receives a set of requirements in form of instances of the product model. Some of these requirements correspond to user inputs, some of them may be generated by the intelligent user interface on its own. In addition, we specify whether the configurator should run in the propagate mode or in the auto complete mode.
2. Start the configuration process.
3. If the requirements are inconsistent with the configuration knowledge base return the conflict sets (at least one or all if run time allows).
4. If the configurator has computed a complete and consistent solution output this solution; otherwise output the possible user decisions.

The process goes on until a complete and consistent configuration is found. The search engine optimises configuration solutions by applying heuristics. These heuristics support decisions aimed at finding configurations which better satisfy the user's interests and at improving the search process. The found solution is the best solution (from the configurator point of view) the configurator can provide. In principle it is possible to generate a set of consistent configurations and present these configurations to the user. However, depending on the application, this could lead to an information overload for the user. In the CAWICOMS applications we decided to present only the best solution found so far, in order to keep the complexity of user interaction low.

In the CAWICOMS project we applied a constraint satisfaction system as a configuration engine [10; 12]. The propagate mode corresponds to the constraint propagation phase which computes the domain reduction of the possible feature values. The auto complete mode corresponds to the solution construction process where after a constraint propagation phase we search for a complete value assignment of the constraint variables.

4 Adaptive User Interaction

The CAWICOMS frontend adapts the interaction style to the user by customising the elicitation of requirements and the presentation of information. The methodologies underlying these activities are outlined in the following subsections.

4.1 Management of the User Model

The CAWICOMS frontend manages an individual user model (UM) describing the user interacting with the system. The UM stores information about user characteristics, such as the nationality. Moreover, the UM stores the system's estimates about the user's interests in the properties of the configurable item (reliability, performance, economy, etc.). Finally, the UM stores the estimates of the user's knowledge about the configurable items.

During the interactive configuration sessions, the estimates of the user's interests and knowledge are updated by interpreting her behaviour. The system sometimes asks the user to explicitly self-assess her own expertise and interests. More interesting, the user's interests can be estimated as an attempt to maximise the utility of a configuration solution by means of actions, such as the rejection of the system's suggestions and the acceptance or rejection of a proposed configuration; see [14; 5] for details. Finally, the user's expertise can be estimated by identifying her information needs. For example, downloading the pages that explain the meaning of features

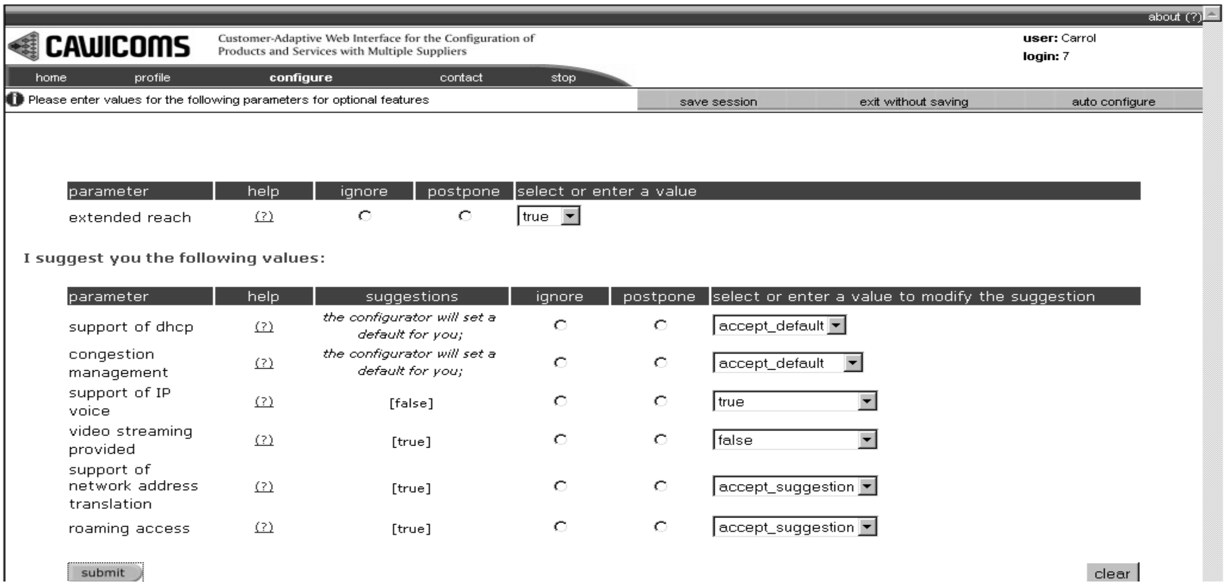

Figure 3: A step in the configuration of an IP-VPN.

Figure 4: Elicitation of preferences for IP-VPN properties.

and components of items is interpreted as evidence that the user is not familiar with that type of information.

4.2 Personalisation of the Interaction

The management of a user-friendly configuration process is based on the personalised elicitation of requirements about the item to be configured and the personalised presentation of the configuration results.

Personalised Requirements Elicitation

The estimates on the user's current interests in the item's properties, such as its reliability, are used to steer the configuration process by proposing feature values and components that maximise the user's expected evaluation of the solution. In this way, the system reduces the number of decisions she has to make. The system also uses other personalisation strategies. For example, individual defaults expressing preferred values can be applied, or personalised defaults describing business rules based on the customer's characteristics. These personalisation strategies, described in [2], are represented as declarative rules with priority.

Figure 3 shows a page generated during the configuration of the optional services of an IP-VPN (such services are represented as a component in the user-oriented representation of the service). The system asks the user to specify whether the company wants the extended reach to Eastern countries. Moreover, the system suggests to set the dhcp and the congestion management protocol as standard configuration defaults. Furthermore, personalised defaults are suggested for the support of IP voice, the provision of video streaming, the support for network address translation, and the roaming access. The suggestions can be overridden, as the user did with the IP voice and video streaming. Moreover, the user is allowed to delegate the system to set the values of the required features on her behalf ("ignore" button), or to postpone the selection of values ("postpone" button). As shown in Figure 4, the system may also elicit user requirements by questioning her about her preferences for properties (e.g., reliability) of an IP-VPN. Given the answer, the most suitable settings to achieve such properties are determined, on the basis of the functional knowledge available to the system (see Figure 2).

Personalised Presentation of Solutions

The system personalises the presentation of a solution to be proposed to the user (e.g., a configured IP-VPN), by focusing the description on the information best fitting her interests and expertise. Presentation strategies split the list of features to be presented into a subset to be shown in the main presentation page, and two subsets to be linked as technical and supplementary information. Such strategies are evaluated and applied to each individual feature, and select the presentation depending on its criticality, the user's interests and expertise. Similar to the personalised elicitation rules, the presentation strategies are managed by a rule-based engine as prioritised rules. We show the rules in decreasing priority order.
1) Show very critical features in the main presentation page.
2) If the feature is related to a property important for the user, then it must be shown in the main presentation page.
3) Link as technical information the technical features related to properties for which the user's estimated interest is moderate (the *type of information* facet is used to distinguish technical features from other data).
4) Link as "more information" the features related to properties for which the user's estimated interest is low.

The presentation is structured according to the "user-oriented components" defined in the user view; see Section 2.

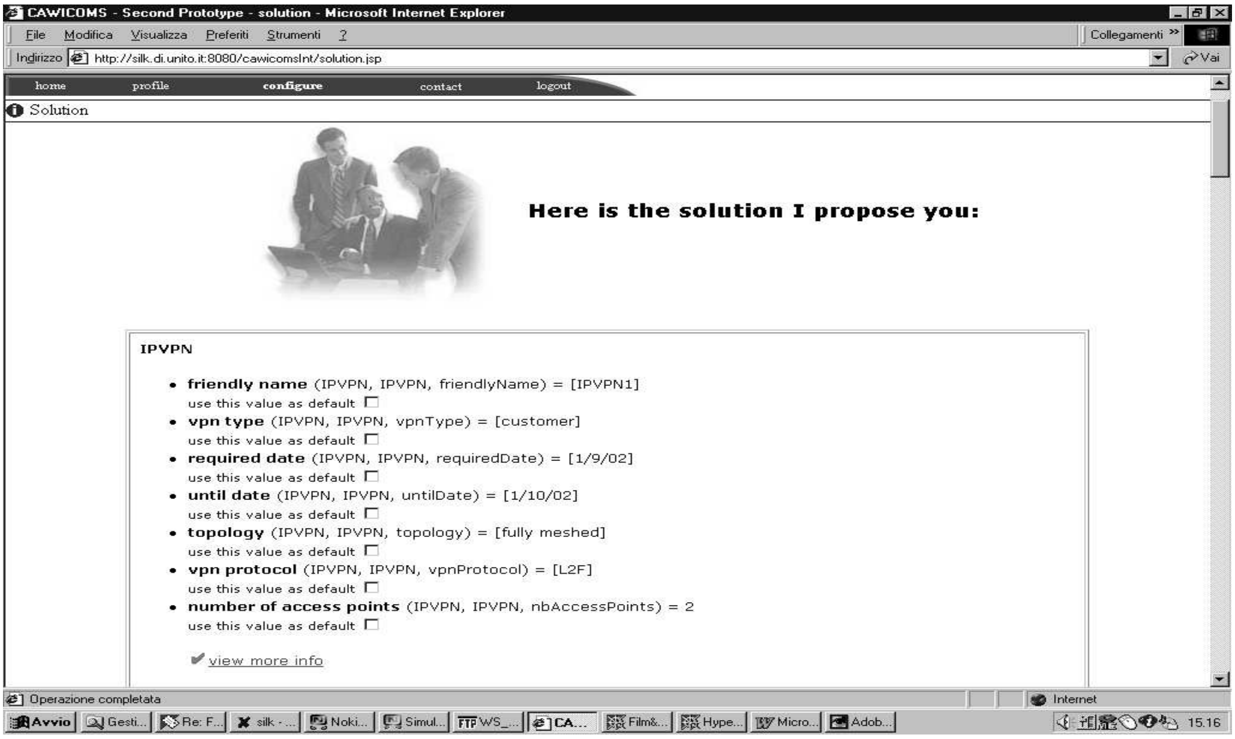

Figure 5: Portion of the presentation of a configuration solution.

For instance, Figure 5 shows a portion of a page generated by our system for the presentation of a configured IP-VPN. The page shows the general features of the IP-VPN, which are grouped in the main component of the user-oriented service representation. The presentation is organised in a compact way, by listing the values of each feature. Similar presentations are generated for the other service components.

5 Explanations

The CAWICOMS architecture allows the generation of user adaptive explanations of conflicting requirements. A non user adaptive approach was proposed by [7]. In configuration sessions this is a key for success because otherwise the user has no hint why a certain set of requirements does not lead to any valid solution. In case the user formulates requirements such that no consistent and complete solution can be computed, we further analyse the conflict sets returned by the configurator and present only the relevant part of these conflict sets. Note that in case the computation of all conflicts is not possible, we base our processing on the conflicts found so far. Every conflict gives a hint to the user regarding necessary changes.

We store in our knowledge base whether the user inputs could be retracted or not; e.g., the value of the packet loss could be retracted, whereas the user age could not. Similar, we distinguish between hard and soft constraints where the user has the possibility to relax soft constraints whereas hard constraints are fix. In the presentation of the conflicts we exploit the expertise level of the user in order to find the right level of detail. Moreover, we exploit the user's interests. We assume that user input which is linked to technical aspects with lower interest can be changed more easily. For instance, if I'm not very interested in the packet loss rate then I might agree more easily on changing the value for this feature. Consequently, we can attract the user's attention to the relevant parts in the revision process.

6 Evaluation

In order to test the adaptive interface, we used role-playing exercises involving a small number of 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 in the form of a questionnaire.

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 scenarios 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 (Network Global Services) who employ a CAWICOMS configuration system. We developed 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 this evaluation were satisfactory. The total number of questionnaires analysed was 30. Approximately half of the test subjects were novices, with 10% in the "Intermediate" bracket. Our test users particularly liked the facilities designed to help novice users: 55% liked the suggestion facility and 35% explicitly praised the auto-configure one, which was equally liked by novice and expert users. 70% of users felt that the adaptive aspects of the system improved the user experience.

7 Conclusions

We discussed the integration of personalisation and Web-based configuration in the development of Web-based configuration systems. In particular, the present paper was focused on the coordination of an intelligent user interface with the underlying configuration engine needed to generate configuration solutions; the generation of personalised interactions with the user, and the generation of explanations of possible failures during the configuration process. The adaptation to individual user requirements enhances the usability of configuration systems. Moreover, it contributes to the management of mass-customisation in e-commerce, as it supports the satisfaction of individual needs at the cost of mass production.

Our approach complements and enhances existing customer relationship management (CRM) systems. In case we can identify a user, we can download the information provided by the CRM database to replace the stereotypical user information. In addition, we can upload into the CRM database the information we gained about a specific user during a configuration session. This integration improves the overall communication with the user because customers frequently hop between different sales channels (e.g., internet, sales rep, and customer care centre) and improve their knowledge about interests and expertise. In addition, we can exploit the information about (un)realisable customer requirements to enhance the product build blocks (i.e., the product and service portfolio) in order to discover new market opportunities.

Our model supports the suggestion of personalised choices during the configuration process, the explanation of failures, the provision of extra-information about the items to be configured and the personalised presentation of solutions. The personalised user interface for the configuration system is based on the dynamic generation of the Web pages, which are created on the basis of contextual information, such as the content of the user model and the type of information to be displayed. As the UM is continuously revised by the system, a reactive adaptation of the interaction is supported, so that the system can acknowledge changes in the user's expertise and interests. We have applied our model within the CAWICOMS workbench for the development of adaptive Web-based configuration systems.

This work was funded by the EU through the IST Programme under contract IST-1999-10688. We thank the other partners of the CAWICOMS project (Telecom Italia Lab, ETIS, ILOG SA, University of Klagenfurt) for their contribution to the design and development of the CAWICOMS configuration system.

References

[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*, 225–229, Lyon, 2002.

[3] L. Ardissono, A. Felfernig, G. Friedrich, D. Jannach, R. Schaefer, and M. Zanker. A framework for rapid development of advanced web-based configurator applications. In *Proc. 15th Conf. ECAI*, 618–622, Lyon, 2002.

[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. Schaefer. Customising the interaction with configuration systems. In *Proc. 9nd Int. Conf. on User Modeling*, to appear, Pittsburgh, PA, 2003.

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

[7] A. Felfernig, G. Friedrich, D. Jannach, and M. Stumptner. Consistency-based diagnosis of configuration knowledge bases. *Proc. 14th Conf. ECAI*, 146–150, 2000.

[8] 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*, 10(2-3):209–249, 2000.

[9] 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.

[10] U. Junker. Preference-programming for configuration. In *IJCAI Configuration Workshop*, 50–56, 2001.

[11] A. Kobsa, J. Koenemann, and W. Pohl. Personalized hypermedia presentation techniques for improving online customer relationships. *The Knowledge Engineering Review*, 16(2):111–155, 2001.

[12] D. Mailharro. A classification and constraint-based framework for configuration. *AI in Engineering, Design and Manucturing*, 12:383–397, 1998.

[13] *Special Issue on Recommender Systems*, volume 40. Communications of the ACM, 1997.

[14] W. Schütz and R. Schäfer. Bayesian networks for estimating the user's interests in the context of a configuration task. In *Proc. UM2001 Workshop on Machine Learning for User Modeling*, 23–36, 2001.

[15] D. von Winterfeldt and W. Edwards. *Decision Analysis and Behavioral Research*. Cambridge University Press, Cambridge, UK, 1986.