

# Intelligent User Interfaces for Web-based Configuration Systems

L. Ardissono, A. Goy, and G. Petrone  
Dipartimento di Informatica  
Università di Torino, Italy  
{liliana, goy, giovanna}@di.unito.it

M. Holland  
BTEExact Technologies, UK  
matthew.holland@bt.com

R. Schäfer  
Ralph.Schaefer@r77.de

G. Friedrich and C. Russ  
Computer Science and Manufacturing Research Group  
University of Klagenfurt, Austria.  
{gerhard.friedrich, russo}@ifit.uni-klu.ac.at

## Abstract

We present a model for the integration of Intelligent User Interfaces and Configuration techniques. This model enhances the capabilities of on-line stores by supporting the development of configuration systems that assist customers in a personalised way, while they select the features of the products/services to be configured.

## 1. Introduction

Although efficient recommendation techniques have been developed to customise the suggestion of items off the shelf [5], 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. Indeed, efficient configuration systems have been developed, which can handle configurable items in large-scale domains; e.g., [4, 3]. 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.

Within the CAWICOMS<sup>1</sup> project, we have developed an intelligent user interface (the CAWICOMS frontend) for a configuration engine, aimed at mediating the interaction with the user. During the configuration process, the CAWICOMS frontend supports a non-technical elicitation of the user's requirements and provides personalised suggestions to help the user set the features of the item to be configured. Moreover, the system explains in a friendly way the possible failures in the configuration process. Finally, the system

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

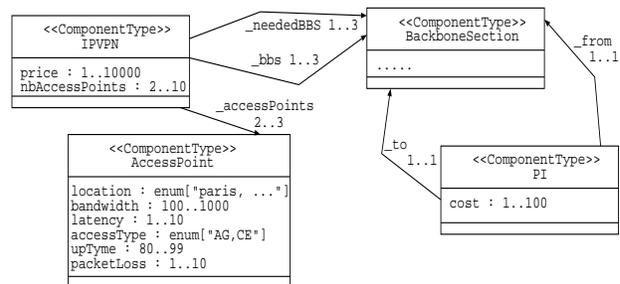


Figure 1. IP-VPN product model.

personalises the presentation of the configuration solutions by focusing on the most interesting features for the user. We have exploited the CAWICOMS frontend in two application domains: the configuration of (1) Telecommunications Switches and (2) of IP-based Virtual Private Networks (IP-VPNs). In the following, we use the latter domain as concrete example.

## 2. Knowledge representation

The technical knowledge about products and services exploited within the CAWICOMS system relies on the well known *component-port* approach for configuration [4]. 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 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). In addition, a knowledge base further constrains the allowed feature-value combinations, port connections, and instantiations of the component types. We refer to such conceptual

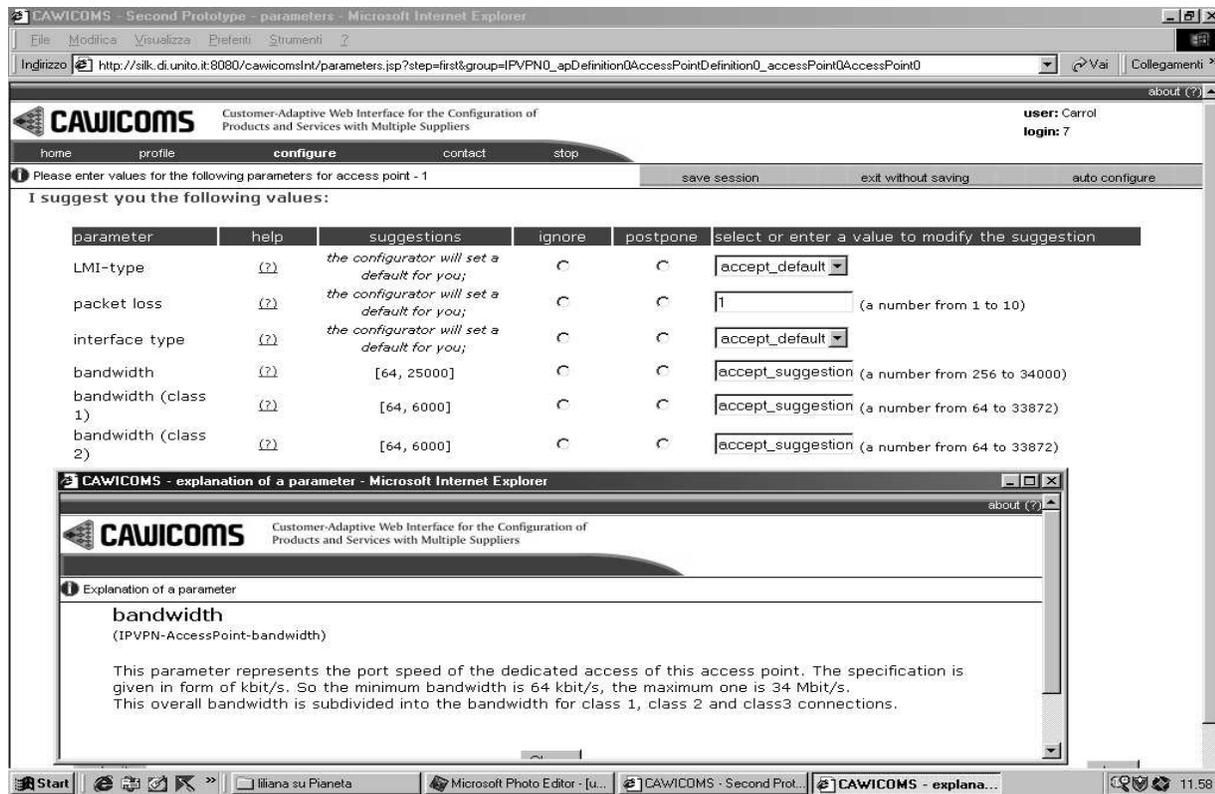


Figure 2. A step in the configuration of an IP-VPN.

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 in several aspects. For instance, we specified a *complexity level* of each feature, as an estimate of the degree of expertise required to understand the meaning of the feature. Moreover, we specified a *criticality level*, needed to identify features whose values should be explicitly acknowledged by the user (important features, from the configuration, or business point of view). Furthermore, we introduced the representation of properties aimed at evaluating the products and services from a qualitative point of view: the presence of this abstraction layer, on top of the level exploited by the configuration engine, enables the system to elicit the user requirements in a non-technical way and to translate them to technical requirements guiding the configuration process. As explained in [1], 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; see [6]. It should be noticed that the specific properties to be used

depends on the application domain; in the configuration of IP-VPNs, we defined properties such as the performance and the reliability of the switches/networks.

### 3. User-friendly configuration

The management of a user-friendly configuration process is based on the exploitation of the following types of information (see [1] and [2] for details):

- the representation of the properties of the products/services to be configured (see previous section);
- an individual model describing the user's expertise and interests in the product/service properties;
- rules for the personalised requirements elicitation;
- rules for the personalised presentation of the configuration results;

The CAWICOMS frontend manages an individual user model (UM) storing explicit and implicit information about the user interacting with the system. The explicit portion of the UM stores the user's characteristics, such as the nationality and the type of company she represents. The implicit portion of the UM describes the system's estimates

about the user's interests in properties of the configurable item, such as its reliability, performance, economy, and so forth. This portion of the UM also stores the estimates of the user's knowledge about the features of the configurable items. All the system's estimates are acquired by observing and interpreting the user's behavior during the interaction. For instance, the values selected by the user for the technical features of the product/service to be configured convey information about her interests in the properties of the final solution to be generated.

The estimates on the user's interests are exploited to steer the configuration process by identifying feature values and components that maximise the user's expected evaluation of the solution. Moreover, features can be set by applying individual defaults expressing specific user preferences for feature values, or personalised defaults describing business rules based on the customer's characteristics. Although the user can override all the suggestions proposed by the system, the provision of personalised suggestions is an effective way to reduce the number of features to be explicitly set by the user, therefore dramatically shortening the configuration sessions.

Figure 2 shows a page generated during the configuration of an access point (access point-1) for an IP-VPN. The system asks the user to specify some features, such as the packet-loss rate and the required bandwidth. As such features are rather technical, the system helps the user by suggesting suitable defaults: the first features should be set by applying standard configuration defaults, while personalised defaults are provided for the features concerning the bandwidth assigned to the access point. As shown in the pop-up window, the user may ask for a description of the meaning of each feature to be set.

Although not shown in the figure, the system may also ask the user about her preferences for the properties of an IP-VPN related to the features to be set.<sup>2</sup> Given the answer, the most suitable settings to achieve such properties are determined. Moreover, the user is allowed to delegate the system to set the values of the required features on her behalf or to postpone the selection of values ("ignore" and "postpone" buttons in Figure 2).

#### 4. Steering the configuration process

The intelligent user interface exploited in the CAW-ICOMS frontend mediates between user and configurator. In order to elicit user requirements and to present possible solutions, the interface exploits the services provided by a configuration problem solving component.

More specifically, we adopted the following view of these activities. The configurator starts from a set of user re-

<sup>2</sup>This is usually done when the the system believes that the user's domain expertise is not enough to set a technical feature in an informed way.

quirements 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. In order to put the user in control of the interaction, the elicitation of the user requirements 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 and other similar features. Then, she might configure the individual access points, the optional features, 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 preferences and requirements is available to the system, and the choices are not critical, the frontend may make the decision without involving the user.

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 configurator. This process may have three outputs: 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 which is needed to generate an inconsistency. This information is exploited by the user interface component to generate explanations for the user.
- In the *auto-complete mode* (see button "auto configure" at the top of Figure 2, right handside), 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 this 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.

We applied as a configuration engine the ILOG JConfigurator, which is a constraint satisfaction system [4]. The propagate mode corresponds in our case 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.

## 5. Explanations

The CAWICOMS architecture allows the generation of user adaptive explanations of conflicting requirements. In configuration sessions this feature 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.

We store in our knowledge base whether the user inputs could be retracted or not; e.g., the value of the feature packet loss could be retracted, whereas the feature 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 features with lower interest can be changed more easily. For instance, if I'm not very interested in the packet loss rate than 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. Conclusions

We presented a model for the integration of personalisation and interactive configuration in user-friendly configurators assisting the specification of complex products and services in the Web. The adaptation to the individual user's 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.

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] L. Ardissono, G. Friedrich, A. Goy, M. Holland, G. Petrone, C. Russ, and R. Schaefer. User-adaptive configuration of products and services. In *Proc. of the IJCAI Workshop on Configuration*, to appear, Acapulco, 2003.
- [2] L. Ardissono, A. Goy, M. Holland, G. Petrone, and R. Schaefer. Customising the interaction with configuration systems. In *Proc. 9th Int. Conf. on User Modeling*, pages 283–287, Pittsburgh, PA, 2003.
- [3] 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.
- [4] D. Mailharro. A classification and constraint-based framework for configuration. *AI in Engineering, Design and Manufacturing*, 12:383–397, 1998.
- [5] D. Riecken, editor. *Special Issue on Personalization*, volume 43. Communications of the ACM, 2000.
- [6] D. von Winterfeldt and W. Edwards. *Decision Analysis and Behavioral Research*. Cambridge University Press, Cambridge, UK, 1986.