Personalising On-Line Configuration of Products and Services

L. Ardissono[1] and **A. Felfernig**[2] and **G. Friedrich**[2] and **A. Goy**[1] and **D. Jannach**[2] and **M. Meyer**[3] and **G. Petrone**[1] and **R. Schäfer**[3] and **W. Schütz**[3] and **M. Zanker**[2]

Abstract. The sales of complex configurable products and services on the Web is challenged by the large number of technical details to be specified during the configuration process and by the fact that the customer base has heterogeneous requirements and knowledge about the items to be configured. Effective personalisation strategies are thus critical to the development of successful Web-based configuration systems.

This paper presents a framework for the management of personalised configuration in business-oriented domains. The main goal is to assist the user during the configuration task by suggesting suitable choices and providing her[4] with information about the products/services to be configured. Our framework integrates user modelling and personalisation techniques with constraint-based configuration techniques and is applied within the CAWICOMS prototype toolkit for the development of adaptive Web-based configuration systems.

1 INTRODUCTION

The provision of services on the Web is challenged by its heterogeneous customer base: electronic catalogs are visited by users differing in interests and knowledge about the products and services they search for. This raises the following issue: how can a Web-based system support these customers in finding the goods best fulfilling their needs? To face this issue, several Web-based systems offer one-to-one recommendation of items, given the customer's preferences [1, 2, 7, 10, 13, 15]. However, the recommendation techniques developed so far do not support the configuration of items, that is essential to comply with the customer's requirements when purchasing complex products, or registering for services. At the current stage, this type of activity can be performed by using non-personalised configuration systems offering a single, standard type of interface. Moreover, as the configuration of complex items would challenge the user with technical details, the systems available on the Web typically solve simple configuration tasks, which can be reduced to exploring a pre-determined set of already configured solutions; for instance, see [4, 5].

This paper presents the CAWICOMS[5] framework for the personalised configuration of products and services. The novel aspect of this work is the integration of an intelligent user interface in a configuration system, in order to support the user in the provision of technical details to be addressed during the configuration task: the user interface fills the gap between the system's point of view, focused on the implementation of the solutions, and the customer's one, focused on their properties. This approach supports the use of the configuration system by heterogeneous users such as inexperienced customers and technicians configuring items for third parties.

In several application domains, configuration systems are used by people, such as sales engineers and managers, filling specific roles in their own organisations. In our framework, this type of information is captured in the definition of the typical interests and expertise of the various user classes: the system exploits stereotypical user modelling techniques [14] to estimate the user's properties since the beginning of the interaction. Moreover, to take into account individual characteristics, dynamic user modelling techniques based on probabilistic reasoning techniques are applied to update such estimates, depending on the user's behaviour. The system also performs probabilistic inferences to reason about the user's requirements and customise the configuration process. Finally, a rule-based approach is used to tailor the interaction to the user's needs, by generating personalised pages, and to integrate business rules in the configuration process. Our framework has been applied to the development of a Web-based system supporting the configuration of high-technology products (telecommunication switches) and services (Internet Protocol Virtual Private Networks, IP-VPN) on the Web.

In this paper, Section 2 outlines one of the application scenarios guiding the development of our framework and Section 3 summarises the personalisation requirements on configuration we identified. Section 4 describes the CAWICOMS framework for the personalised configuration of items by specifying the typical flow of the interaction with the user (4.1), the strategies for customising the configuration task (4.2) and the inference techniques for reasoning about the user's interests and expertise (4.3). Section 5 outlines the system architecture and Section 6 ends the presentation.

2 APPLICATION SCENARIOS

The development of the CAWICOMS framework was guided by application scenarios from the telecommunication domain.

[1] Università di Torino, Italy.
[2] Computer Science and Manufacturing Research Group, University of Klagenfurt, Austria.
[3] DFKI, Saarbruecken, Germany.
[4] We refer to the user in a unique gender for readability purposes.
[5] CAWICOMS is the acronym for "Customer-Adaptive Web Interface for the Configuration of Products and Services with Multiple Suppliers" (www.cawicoms.org/). This work was funded by the EU through the IST Programme under contract IST-1999-10688.

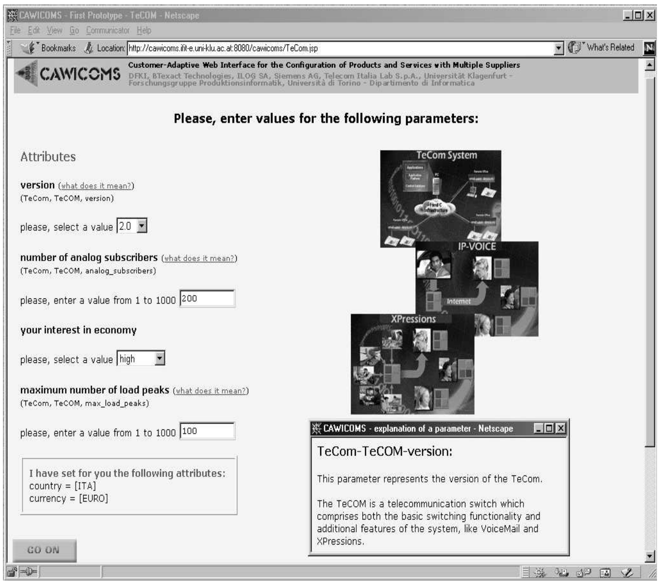

Figure 1. A personalised question page generated during a configuration step.

One is the configuration of telecommunication switches for next-generation public telephony. The core of the product solution is a *switching network* composed of a set of building blocks called *racks*, *frames* and *modules*. The number of these building blocks and their structure depends on the required performance characteristics and features specified by the customer. In order to specify this type of product, the current configuration systems pose up to hundreds of questions about parameters to their users, who can become overstrained and are, in general, unable to overview the configuration process. This is especially a problem when sales personnel lacks deep technical knowledge. The latter is often the case when companies recruit their sales force for new emerging markets or contract a third-party as an alternative sales channel. Therefore, configuration systems play a crucial role as a corporate knowledge management tool, where user specific knowledge presentation requires intelligence in the interface. In our application, we identified four user groups differing in the expertise about products and the frequency of system interaction: *sales engineers* have deep technical knowledge and typically want to drill down to configuration details and interact with a configuration system without any assistance. *Senior sales representatives* have good knowledge about the products to be configured and reasonable experience in using configuration systems. The *Junior sales representatives* category encompasses sales personnel with almost no experience, and/or low level of technical understanding. *Customers* cannot be assumed to have any training on the product. Especially when the configuration system is used as a medium to deliver product information, these users have to be provided with several explanations about the product characteristics and structure.

3 PERSONALISATION REQUIREMENTS

In order to take the user's interests and knowledge into account, a configuration system should fill the gap between the underlying representation of the product/service and the user's perception of such an entity. While an expert user is assumed to have precise knowledge about the features of a product/service and its structure, a novice one only perceives its most "external" aspects. For instance, a telecommunication switch is characterised by a large set of features, some of which are very technical, such as the number of trunk lines to be exploited. The novice user's view of the product may concern a subset of all these features: e.g., she may want to specify how many terminals will be connected to the switch.

We have identified a set of requirements concerning the personalisation of the interaction by interviewing sales representatives and technical engineers regularly using the configuration systems available to a telecommunication company and occasional users of on-line configuration systems. The most relevant issues follow.[6]

1. The configuration process may require the specification of a large set of data.
2. Depending on the user's expertise, the specification of the parameter values may be difficult, if not impossible, as the user might not know the impact of her selections on the configuration solution.
3. Most users are only interested in the cost and the usage characteristics of the solution, while they do not care about how it is implemented.
4. Some configuration parameters depend on specific customer features and should be automatically set: e.g., the customer's nationality could determine the currency for payments.
5. Other configuration parameters are so critical that the user must take the responsibility to set them. At least in these cases, she should be supported with specific information about the meaning of the parameters.
6. The user should be enabled to postpone some configuration decisions, when she is uncertain about the preferred value for a parameter.

4 PERSONALISATION OF CONFIGURATION SESSIONS

4.1 Interaction flow

The CAWICOMS system manages the interaction with the user as a dynamically generated sequence of configuration steps. During each step, the user is shown a form where some parameters are displayed, together with their domain (set of admissible values). For instance, Figure 1 shows a typical page generated by our system during the configuration of a telecommunication switch (TeCOM). The leftmost part of the page displays the list of questions the user is asked about and includes configuration parameters (e.g., version of the switch, number of analog subscribers) and information about the customer's requirements: her interest in the economy of the product, i.e., in how costly the solution is going to be. The user is asked to set values for such parameters and to submit the form to the configuration system ("GO ON" button), which propagates the values in a constraint network representing the partial solution. The propagation triggers domain reductions on other parameters.

[6] We also identified requirements on the presentation of configuration solutions, but we omit them for space reasons.

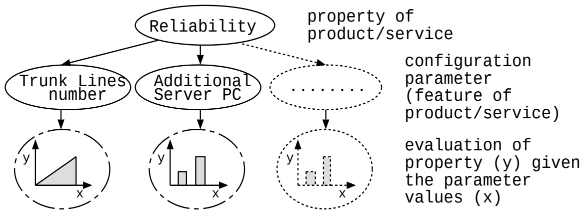

Figure 2. Relations between parameters and properties in the Frontend Model.

A help button ("what does it mean?" link) is available for each parameter to retrieve detailed information about its meaning: Figure 1 shows the explanation window for the "version" of the TeCOM. Moreover, at each configuration step, the system may set some parameters, by applying personalisation techniques. When this happens, the system shows these settings below the question list. For instance, in Figure 1, the system has set country and currency for the switch.

After each propagation step, the user is shown another list of parameters to be set, together with their current domains. When the constraint network evolves to a complete solution, where each parameter is set to one value, the system presents the solution. In contrast, if the user's choices generate a failure in the constraint propagation process, the configuration fails.

4.2 Personalisation techniques

4.2.1 Knowledge about products/services and users

The satisfaction of the requirements in Section 3 is based on the integration of user modelling, personalisation and flexible dialogue management techniques, as well on the use of a domain ontology describing personalisation-oriented information about products and services.

The technical knowledge about products and services is described in a *Product Model* supporting a conceptual, structured description of entities with features, components and constraints between components (see [6] and [3]). This model specifies the technical information needed by the configuration engine to generate solutions, but does not include high-level information typically addressed during the interaction with the user. This further type of information is stored in the *Frontend Model*, that extends the Product Model with data such as the explanation of the meaning of configuration parameters and an estimate of their technicality and of their criticality for the configuration task. The Frontend Model also stores the impact of parameters on the evaluation of the solution regarding different aspects and the difficulty of knowing such information. For example, in Figure 2, the number of trunk lines has positive impact on the reliability of a switch.

The system manages an individual user model storing information about the user's personal characteristics (nationality, enterprise type, etc.), her expertise about the products/services, and individual defaults (i.e., specific preferences for parameter values). The user model also describes the user's interests in different aspects of the product, such as its reliability and economy, corresponding to the properties defined in the Frontend Model. The personal characteristics are represented as $< feature, value >$ pairs. Instead, the estimates about the user's interests and expertise are represented as probability distributions on the values of variables associated to the knowledge items and the product properties: e.g., a variable represents the user's interest in the reliability of a configuration solution (see Section 4.3 for a description of the techniques applied to initialise and update the user model during the interaction with the user).

4.2.2 Personalisation strategies

The conceptual representation of products and services stored in the Frontend Model guides the system in the management of a structured configuration session, suggesting the configuration of the product/service one component after the other, in a possibly hierarchical order. However, the system enables the user to postpone the setting of parameters and to select the components that she wants to configure first. In this way, mixed-initiative dialogues are managed, where both the system and the user can guide the configuration process (requirement 6 in Section 3). The Frontend Model also supports the user during the setting of parameter values by providing her with explanations about the meaning of the parameters to be filled in (requirement 5).

Finally, the assessment of the user's interests and expertise, together with the exploitation of the information stored in the Frontend Model, supports the satisfaction of requirements 1 to 4, as it enables the system to automatically set parameters and to personalise the formulation of questions to the user. Given a configuration parameter to be filled in, alternative strategies can be used to identify the value(s) to be set and a personalisation module evaluates the alternatives, searching for the most promising one:

1. *If the criticality of the parameter is over a threshold, ask the user about the value to be set.*
2. *If the user model contains an individual default value for the parameter and the value is included in the current domain of the parameter, set the parameter accordingly.*
   The user model may contain individual defaults, as the system enables the user to set "long-lasting" preferences.
3. *If a personalised default matching the user's characteristics is available for the parameter and the intersection between the suggested values and the current domain of the parameter is not null, set the parameter to the intersection.*
   Personalised defaults represent business rules suggesting parameter settings based on customer's characteristics and are represented as production rules. The head of the rules specifies a possibly complex "and/or" condition on the user data. The consequent suggests a set of values for the requested parameter, together with the result of the evaluation of the head on the user model. For instance, in the interaction of Figure 1, a simple personalised default is applied that sets the "currency" parameter of a telecommunication switch to the appropriate currency (USD vs. Euro) on the basis of the user's nationality.
4. *If the parameter is related to some properties for which the user's estimated interest is low, set a standard (non personalised) default value consistent with the current domain.*
5. *If the user's estimated expertise is sufficient to choose a value for the parameter, ask her to set the preferred value, given the current domain.*
   This strategy relies on a comparison between the user's

expertise and the difficulty of the parameter in order to estimate the likelihood that the user will be able to answer the question [8].

6. *Given the parameter domain, select the best value and set it, on the basis of the user's interests in the product/service properties.*
   This strategy exploits the information in the user model to predict the preferred values for the parameter. The properties related to the parameter in the Frontend Model are used to focus on the corresponding user interests, which are analyzed to check whether a sufficiently substantiated prediction of the best value can be made. Section 4.3 describes the evaluation model ascribed to the user in our system.
7. *Elicit (if not yet done) information from the user about her interest in properties of the product/service that are influenced by the parameter to be set. Then, apply strategy 6 to possibly set the parameter values.*
   This strategy is applied to let the user self-assess her interests, when the information in the user model is not sufficient to perform any prediction.
8. *Postpone the parameter setting to a later stage of the configuration process (last resort).*

These strategies are sorted by priority because, whenever safe, automatic parameter settings are favoured over questions to the user. However, the selection of the strategy to be applied is a little more complex: while the evaluation of the first three strategies is binary (either they suit the current situation, or they do not), the other strategies rely on uncertain information. For instance, strategy 4 depends on the estimation of the user's interest in the properties related to the parameter in focus; similarly, strategy 5 is based on the probability that the user knows the meaning of the parameter. In order to take this uncertainty into account, the suitability of a strategy is evaluated, in the [0..1] range, and applicability thresholds are defined to rule out weak strategies.

For each parameter to be filled in, the personalisation module evaluates the strategies, according to their priority, and selects the first one exceeding the threshold. The selected strategy is applied to continue the interaction with the user, either by eliciting information from her, or by autonomously setting the value. The question pages submitted to the user reflect the fact that the parameters may be filled in using alternative strategies. For instance, in Figure 1, the user is questioned about parameters and interests; moreover, some parameters are set by the system.

4.3 Reasoning about the user's knowledge and interests

During the interaction with the user, the system estimates her interests and expertise by analyzing her observable behaviour: we use a probabilistic inference mechanism, namely *Bayesian networks* [12], to face the uncertainty affecting the interpretation of the system's observations. In order to estimate the user's interests, we have to ascribe her an evaluation process that she is supposed to apply in the evaluation of products and services. In an idealisation, we use Multi-Attribute Utility Theory (MAUT [18]) for this purpose. MAUT is a general evaluation scheme applied or, at least, compatible with the schemes used by several user modelling approaches for estimating the user's interests [16]. Many users are also familiar with MAUT, because it is used by consumer organisations for evaluating products. For example, in Germany, Stiftung Warentest uses MAUT for evaluating consumer products (e.g., digital cameras [19]).

According to MAUT, the *overall evaluation* of an object determines its utility for the user. Usually, an object can be evaluated by taking several aspects (*value dimensions*) into account. Moreover, not all the users are equally interested in the same aspects: e.g., suppose that a telecommunication switch is evaluated on the basis of its performance, reliability, and economy dimensions; then, some users may be more interested in performance and reliability and less in economy.

Formally, the overall evaluation of an item is expressed on a numerical scale, e.g., from 0 to 10, and is defined as the weighted addition of the object's evaluation on its relevant value dimensions [18].[7] A weight is associated with each dimension to describe the user's interests. The more interested the user is, the larger the weight is.

Similar to the overall evaluation, the evaluation of the object on a dimension d is based on a weighted addition of the evaluation of the attributes relevant for d. In our configuration task, an attribute corresponds to a parameter to be set and is characterised by a list of levels, each one associated to a parameter value. A numerical scale is defined to quantify the levels of an attribute and an *evaluation function* maps evaluation values onto the attribute levels. For example, regarding reliability dimension, a guaranteed uptime of 99% yields 10, whereas an uptime of 50% yields 2. For simplicity, we assume that the evaluation functions of the attributes and their weights are fixed for all users. The described weights and the evaluation functions are defined in the Frontend Model; see the diagrams at the bottom of Figure 2 in Section 4.2.1.

The weights associated with the dimensions are determined by applying a probabilistic approach and are represented as a probability distribution. At the beginning of the interaction with the user,[8] these weights are roughly estimated by using stereotypical knowledge about users [14]. A set of stereotypes define categories of users (such as the representatives of a small company) and the related weights. These stereotypes are activated based on the user's personal characteristics. Then, the user's observed behaviour in typical situations is interpreted in order to update these estimates. The following situations can be processed:

- Self assessment: especially at the beginning of the interaction, the system may ask the user about her interests in the properties of the product/service. The user's answer reflects her self-assessment, which is very likely related to her interests, but this fact should not be taken for granted because she might misunderstand the meaning of the terminology used by the system.
- During the configuration process, the user can change the parameter values that the system proposed as defaults by applying personalisation strategies. This type of action suggests that the user believes that the change has a positive impact on the overall evaluation of the solution. In other words, she believes that the new parameter settings cause a positive shift in the evaluation of the item to be configured,

[7] Other possibilities for aggregation are described by [18].

[8] If the system does not already have an individual user model acquired in previous interactions with the user.

with respect to the evaluation with the values proposed by the system.

- After generating a configuration solution, the system presents it. Then, the user has to decide whether accepting the solution or looking for an alternative one. If she accepts it, her behaviour can be interpreted as evidence that her overall evaluation of the solution is good.

For each of these situations, a Bayesian network has been specified which reflects the above described dependencies. These specifications are domain independent: at runtime, the actual network for processing the situation with the parameters involved is created and used for the interpretation of the user's behaviour. This interpretation results in an update of the probability distributions representing the weights of the dimensions corresponding to the user's interests.

The user's expertise is estimated by applying an approach based on [8]. Due to the limited space constraints, we only sketch it: if a user is observed to click on a help button, she probably does not know the implications of the parameter and therefore her expertise is assumed to be probably low. If we observe that the user knows the implications of a parameter (e.g., because she specifies a parameter value), her expertise is probably high.

5 SYSTEM ARCHITECTURE

The CAWICOMS system is based on a modular, distributed architecture, where a specialised module is associated with each main task to be carried out during the interaction with the user: e.g., configuration, user modelling, personalisation and generation of the Web pages. The system is implemented in Java and exploits standard software development environments. In particular, the user interface consists of a sequence of Web pages, implemented as JSPs [17], whose content is dynamically generated on the basis of the interaction context and the application of the personalisation strategies. The JSPs run within an Apache Web Server. Moreover, some of the modules use specialised engines for the execution of the tasks they are devoted to. For instance, the configuration system is based on the ILOG JConfigurator engine [6, 9, 11].

6 CONCLUSIONS

We have presented the personalisation facilities offered by CAWICOMS, a framework for the Web-based configuration of products and services. These facilities allow tailoring the interaction style to the individual user and also support her in the configuration of the product/service best suiting her needs. The personalisation of the interaction is based on the integration of a user-oriented view of the configuration task with the technical level at which configuration systems usually work. This result is achieved by integrating constraint-based configuration techniques with user modelling, personalisation and dialogue management strategies. In addition, a domain ontology describing personalisation-oriented information about the items to be configured is used.

We have applied this framework to the development of a prototype system working on a subset of the telecommunication switches domain and we have performed a first test of the personalisation facilities offered by this prototype with a limited number of users having different background and occupation. These users appreciated such facilities, especially the personalised setting of parameters, as this sensibly speeds up the configuration process. However, they wanted to be able to control the system's decisions, possibly overriding them. For this reason, we have modified the user interface to produce editable personalised suggestions, which are typically suitable for the user, but can be modified, if she does not agree with the system's decisions. The users also appreciated the system's explanation capabilities, although only partially developed, because they enhance the comprehension of the configuration process. Finally, other requests for a more flexible management of the interactions came. For instance, the management of reconfiguration, with its implications (corrections of previous parameter settings, revision of a configuration solution, recovery from a configuration failure), was considered essential and is part of our future work.

REFERENCES

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

[2] BroadVision, 'Broadvision', http://www.broadvision.com.

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

[4] FIAT, 'buy@fiat: la tua prossima auto.', http://www.buy@fiat.com, (2001).

[5] Fidelity Investments, 'Insurance center @fidelity.', http://www400.fidelity.com:80/, (2001).

[6] ILOG, 'ILOG JConfigurator', http://www.ilog.com/products/jconfigurator/, (2002).

[7] Net Perceptions Inc., 'Net perceptions', http://www.netperceptions.com.

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

[9] U. Junker, 'Preference-programming for configuration', in *IJ-CAI Configuration Workshop*, pp. 50–56, Seattle, (2001).

[10] 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).

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

[12] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann Publishers, San Mateo, CA, 1988.

[13] *Special Issue on Recommender Systems*, eds., P. Resnick and H.R. Varian, volume 40, Communications of the ACM, 1997.

[14] E. Rich, 'Stereotypes and user modeling', in *User Models in Dialog Systems*, eds., A. Kobsa and W. Wahlster, 35–51, Springer Verlag, Berlin, (1989).

[15] *Special Issue on Personalization*, ed., D. Riecken, volume 43, Communications of the ACM, 2000.

[16] R. Schäfer, 'Rules for using multi-attribute utility theory for estimating a user's interests', in *Proc. 9. GI-Workshops: ABIS-Adaptivität und Benutzermodellierung in interaktiven Softwaresystemen*, (2001).

[17] Inc. Sun Microsystems, 'Java Server Pages', http://java.sun.com/products/jsp/.

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

[19] Stiftung Warentest, 'Digitalkameras: Pixeljagd', *test*, (6), (2000).