Customer support for the web-based configuration of individualised products

Rosmary Stegmann*, Thomas Leckner, Michael Koch and Johann Schlichter

Department of Informatics
Technische Universitaet Muenchen
Boltzmannstr. 3, 85748 Garching, Germany
E-mail: stegmann@in.tum.de
E-mail: leckner@in.tum.de
E-mail: kochm@in.tum.de
E-mail: schlichter@in.tum.de
*Corresponding author

Abstract: The main task of mass customisation is the integration of the individual customer’s input into the manufacturer’s system of value creation. This function can be supported through information technology, such as web-based configurator tools. With such tools customers are enabled to virtually assemble a product. However, customers must also learn to deal with the complexity of the product model and of the configurator tool itself. This work introduces different approaches to overcome the problems of complexity and uncertainty associated with this customer tool interaction. Additionally, the prototype of a web-based configurator tool for products with many degrees of freedom is described from the user’s point of view.

Keywords: web-based product configuration; customer support; personalisation of product models; product recommendations; customer community; customer collaboration; customer profiles.


Biographical notes: Rosmary Stegmann is Computational Linguist at the Technische Universitaet Muenchen, Germany. She holds a master’s degree in Computational Linguistics and is currently taking her doctorate degree in Applied Informatics at same university. Since her master’s degree she has been working for SailLabs (Machine Translation software for Langenscheidt) and DaimlerChrysler Research since 1998. Her current interests are in personalisation and natural language processing for user profiling.

Thomas Leckner is a doctoral candidate and research employee in the Department of Informatics at the Technische Universitaet Muenchen, Germany. His diploma thesis was awarded the Werner von Siemens Excellence Award 2002. Currently, his focus is on product configuration, virtual communities, and customer collaboration for supporting customers during the task of individualising products.
1 Introduction and overview of related work

Whenever an enterprise decides to offer individualised products to its customers, it must deal with the customers’ individual needs. Since the customer himself defines the product he wants, communication and interaction with the customer is critical. The communication and interaction with the customer can clearly be facilitated through internet tools. Initial technical approaches for this purpose are online configurator tools (Sabin and Weigel, 1998; Felfernig et al., 2001), which can be found, for instance, on homepages of large automakers, computer vendors, and in the fashion industry.

In the multidisciplinary research effort SFB5821 we have been addressing the issue of supporting customers in configuring individualised products. In this chapter we first present an overview of related work (Section 1.1) and clarify our focus (Section 1.2). In Section 2, we then address how the envisioned configurator tool will support the customer. First, we discuss how automatic recommendations are created (Section 2.1), then how customer-to-customer collaboration can be supported (Section 2.2). Section 2.3 finally addresses the issue of user profiles needed for automatic recommendations and for community support. Section 3 presents the prototype implementation of the product specification tool that implements the aspects discussed above. We first motivate the system architecture (Section 3.1) and the design of the product model (Sections 3.2 and following), and then address the user interaction. In Section 4, we finally summarise our work and draw some conclusions.

1.1 Related work

Using configurators, customers can select different properties of a product by selecting the variant they desire. For the configuration of product variants there currently exists a broad base of methods and tools that mainly address modelling of product knowledge (Ardissono et al., 2001; Felfernig et al., 2001; Sabin and Weigel, 1998; Tiihonen et al., 1998). When extending the possibilities from selecting among variants to freely designing a product, researchers mention the possibilities for supporting customer driven innovation (von Hippel and Katz, 2002). However, such tools for freely customisable products with different degrees of freedom are not yet available. Indeed, there are CAD tools for product development. These, however, are too complex for the end customer.
Another issue that has not been investigated fully is the potential of mutual support of customers in the configuration of (complex) products (Piller et al., 2003). There are some generic reports dealing with the ways in which customers support each other (Leimeister et al., 2003; Terveen and Hill, 2001) that distinguish between synchronous and asynchronous collaboration. While synchronous collaboration mainly happens in small teams and is closely related to Groupware (Steinfeld et al., 1999; Turoff et al., 1993), asynchronous collaboration happens in large loosely coupled groups and is more related to ‘social navigation’ (Dieberger, 1997; Twidale et al., 1997).

Social navigation can support personalisation. Approaches to provide personalised views on product catalogues can be found in Ardissono et al. (2002) and Schubert and Ginsburg (2000). In these approaches personalisation does not address the user interface, but the content presented to customers. The spectrum of personalisation ranges from content-based methods to collaborative methods like those proposed in social navigation.

Both for content-based and collaborative personalisation, information about the customer is needed. This information is stored in user profiles. There are several contributions to modelling user profiles (Fink and Kobsa, 2000; Kobsa, 2001). In the context of user profiles, the acquisition of information about customers is of central importance. Acquisition techniques are structured in explicit and implicit methods. Explicit acquisition methods ask the customer to enter all information explicitly in questionnaires or adaptive interviews (Stolze and Ströbel, 2001), while implicit methods focus on the automatic gain of user information through an analysis of user actions (Claypool et al., 2001).

1.2  Our approach

When needed, our approach supports the customer in configuring a freely customisable product by providing personalised recommendations during the configuration, both for whole products and for specific degrees of freedom. In order to enable and support the customisation of products, several aspects have to be addressed: product modelling, user profile acquisition, user interface issues, and personalisation strategies. In the following section, the main differences between our configurator tool and existing systems on e-commerce websites are described.

1.2.1 Products with many degrees of freedom for the customer

In conventional configurators the customer is guided through a rather small set of product variables and he can select from a set of predefined variants for each of these variables. In contrast to this, our aim is to build a system where the customer has enhanced and even more complex possibilities to adapt the product according to his individual preferences and needs. In consequence, the configurator tool rather aims at presenting to the customer the whole physical structure of a complex product with all its components, sub-components, and all properties of those components.

Therefore, the first fundamental requirement for our web-based configurator tool is the capability of handling products with many degrees of freedom. With the term ‘degree of freedom’ we describe those parts of the product which can be modified by the customer (see Section 3.2 and 3.3).
Additionally, we also had to determine methods of presenting such complex products to the customer in an appropriate manner. On the one hand, the customer needs an overview of the product and its structure. Thus, the customer must be able to navigate through the multitude of product possibilities. On the other hand, we had to enable the customer to influence the various degrees of freedom, ranging from a change in the product’s structure to the assignment of values for different types of component attributes. Also, some sort of configuration checking was necessary to evaluate the viability of the customer’s choices.

1.2.2 Additional support for the customer

The configuration task for complex products with lots of degrees of freedom can, of course, become quite tedious and time-consuming for the customer. Moreover, the effect of many degrees of freedom can discourage the customer due to his limited knowledge and experience in developing and customising products. In fact, too many choices may confuse the customer (Helander and Jiao, 2002).

In order to overcome the complexity during the configuration task, we defined additional support mechanisms and integrated them into our configurator. But in contrast to the approach of an advisory system introduced by Blecker et al. (2004) we did not want to hide the whole configuration complexity from the customer. Instead, the customer must manually configure the complex product, but is extensively supported by additional functionality.

For this purpose, on the one hand, we embedded statistical information into the configurator tool, presenting information for example on how often a certain value for a degree of freedom has been chosen in the past. On the other hand, we enhanced the configurator tool with personalisation techniques in order to give recommendations to the customer in an automatic way. Recommendations are given in various granularities (Stegmann et al., 2003) and are generated by different methods. Here we especially focused on user profile-based filtering. Thus, the capability to acquire, store and analyse user profiles had to be incorporated into our configurator tool. Our work on profile acquisition and management will not be described in detail in this paper, however. Please refer to Stegmann et al. (2004) for more details.

Additionally, we wanted to overcome shortcomings of single filtering approaches (Melville et al., 2002) through the dynamic combination of different filtering methods. We therefore included in our configurator tool the Pipelined Filter Combination approach that is described in more detail by Renneberg and Borghoff (2003).

1.2.3 Adaptability to different product models

A final important requirement for our web-based configurator tool was the openness towards product models. We did not build a tool which was restricted to a specific kind of product. Instead our aim was to build a model-based tool which could dynamically adapt to an underlying product model. In our approach the user interface of the configurator was supposed to be generated ‘on the fly’ while loading the respective product model. The product model itself describes the product’s structure and all the degrees of freedom through the use of conceptual building blocks, which are defined in a generic product meta model (Janitza et al., 2003; Leckner and Lacher, 2003).
2 Methods for supporting the customer

Before addressing the configurator tool itself in the following section, we first discuss the way in which the tool supports the customer. As presented in the previous section, there are two main support functionalities:

1. providing automatic recommendations
2. supporting customer collaboration.

In order to implement the concept of automatic personalised recommendations, information about the customer’s preferences needs to be accessible. In addition to describing the methods for supporting the customer we will also briefly describe how we collect, represent and store customer-related information in this section.

2.1 Automatic recommendations: personalisation of product models

To minimise the effort of manual product specification for the customer we apply the concept of personalisation. This means that the customer no longer has to deal with the complexity of the whole product model. Instead he deals with a restricted product model (see Section 3.3) in which only those choices which might be of interest to him, are visualised. Additionally, recommendations for these choices are generated and presented. After all, with applying methods of personalisation to every single degree of freedom, recommendations for full products that have not been selected and bought before are possible. Naturally, if the customer wishes, he can always access all degrees of freedom with all possible choices.

2.1.1 Generating recommendations

For generating recommendations three different methods are used: history-based filtering, rule-based filtering and collaborative filtering.

History-based filtering

To achieve history-based filtering, the customers’ buying histories are analysed. This approach assumes that customers are likely to select those values for the degrees of freedom, which the majority of other customers have previously selected.

Rule-based filtering

Based upon the customer profile, different values for the degrees of freedom of the product are chosen according to predefined rules. These rules can be defined, for example, by the marketing department of the enterprise.

Collaborative filtering

The collaborative filter is similar to the history-based filter in that previously purchased products are analysed. But as opposed to history-based filtering not all products are weighted equally. The value for a degree of freedom will be weighted according to the similarity of the previous buyer to the current customer. This means that in addition to the analysis of the previously purchased products there is also a comparison of customer
profiles. The higher the customer similarity, the greater the importance of the purchased products for collaborative filtering. The main idea of this filter is that similar minded customers will tend to buy similar products (Breese et al., 1998). The analysis of the similarity of customer profiles is performed by different database queries on the user profiles. In addition to demographic data and information about the customer’s name and address, information about the customer’s interests and ratings for different products are also stored in the profile. Both interests and ratings are represented as normalised float values of the interval [0…1]. In other words, a normalised value expresses the probability of the customer appeal or interest in the rated object.

On the basis of such normalised values the similarity between two profiles \( A \) and \( B \) can be calculated as follows:

Formula 1:

\[
\text{similarity}(A, B) = 1 - \frac{\sum_{i=1}^{n} \text{abs}(A(i) - B(i)) \cdot \text{weight}(i)}{\sum_{i=1}^{n} \text{weight}(i)}
\]

where \( A(i) \) are rated interests in the user profile of customer \( A \), and \( B(i) \) are rated interests in the profile of user \( B \). Whenever a customer does not explicitly give a rating, the standard value chosen here is 0.5. Also the different elements \( A(i) \) and \( B(i) \) can be weighted by \( \text{weight}(i) \) in accordance to their importance and relevance for the collaborative filter. Additionally the \( \text{weight}(i) \) is set lower when less customers have made an explicit selection for that specific element \( i \). In the end we achieve the similarity measure \( \text{similarity}(A, B) \in [0…1] \) of two customer profiles \( A(i) \) and \( B(i) \), which can be used to weight the relevance of a previously bought product with respect to the referred customer (Karacapilidis and Leckner, 2004).

2.1.2 The filtering pipeline

All the different filtering methods have their advantages and disadvantages. To combine different filtering methods we have introduced a Filtering Pipeline. Using the filtering pipeline, any combination of the filtering methods is possible for generating recommendations. The drawbacks of the different filtering methods can be balanced and it is possible to use particular filtering methods only for parts of the product model.

Another issue of the pipeline is that in the scenario of configurable products, we cannot simply recommend a value for one degree of freedom without taking into consideration the other degrees of freedom. Interdependencies between degrees of freedom might be possible, e.g., constraints might be violated whenever changing the value of a single degree of freedom. Therefore, when generating recommendations the whole product model must always be considered. In consequence, the input and output of every recommender is a complete product model, even if only the value of one specific degree of freedom has to be changed.

Obviously, when the whole product model is an input for the recommender, it is necessary to have a method of marking those parts of the product model which the recommender may change. In our approach we use a specific path-description to exactly point to those elements in the hierarchical product model the recommender
shall change. Additionally, we have fix-markers in the product model to allow the customer to explicitly mark degrees of freedom which must not be changed by the recommender system.

In a nutshell, there exist different external conditions, which have to be taken into account whenever generating a personalised product recommendation:

- constraints between degrees of freedom
- markers for those degrees of freedom which may be changed
- markers for those degrees of freedom that are fixed.

There are also differences in the quality of a recommendation: In some cases the system may decide very clearly what to recommend. But in other cases it might be difficult for the recommender to decide, because all possible values are about equally good or bad for the customer. In order to take care of such external conditions we have developed the concept of scoring functions for degrees of freedom. This approach is described in detail by Renneberg et al. (2004) and allows us to score the possible values for all degrees of freedom. On the bases of these scoring functions a maximum score for the single product configuration can be calculated, leading to a valid recommendation. Additionally, the scoring functions constitute the combination of different filtering methods, which can be specialised for specific parts of the product model (Renneberg and Borghoff, 2003).

2.2 Direct recommendations in customer communities

The personalisation methods described in Section 2.1 can help the customer to overcome the complexity of product configuration. Nevertheless there still remains the problem of uncertainties with regard to the product’s degrees of freedom and the related configuration decisions (Huffman and Kahn, 1998). Such uncertainties mainly result from lack of knowledge and experience. Whenever a customer purchases a highly individualised product, this product usually has never been produced before. Therefore, the customer cannot see, touch or evaluate the product before buying it. In consequence, the customer has to deal with uncertainties about the final product appearance, feel and function (Wind et al., 2002).

Our approach to handle these uncertainties is to embed the configurator tool into a virtual community platform. In such an environment the customer can take into consideration the opinions, experiences and recommendations of other customers. In fact customers are likely to take into account opinions and experiences of other people during the selection and configuration of a product, as reported in the EuroShoe Project (Piller et al., 2002). By doing so, on the one hand the customer can feel more confident of the correctness and usefulness of his individual selections and adaptations (Piller et al., 2003). One the other hand, the customer can ensure himself of the fact that his product is valued by others. He can, for example, look up how others have configured their products and how these products have been rated. Also it is possible, to explicitly ask others for their opinions about the current value settings in the customer’s product configuration. The access to the information space of a virtual community gives customers the chance to get an idea about what other customers think. Therefore the virtual community supports ‘knowledge transfer’ between customers (Ishida, 1998) by providing the single customer access to the knowledge and experience of others. Also the community space enables customers to give direct advice to each other.
2.2.1 Basic characteristics of virtual communities

Before we discuss different methods for community support and customer collaboration, some basic characteristics of virtual communities are described. Following the research of Hagel and Armstrong (1997) the most important criteria for a successful virtual community are:

- the support for different communication methods between community members
- the motivation of community members to actively create own content for the community
- the possibility for community members to establish and manage relationships to members
- the goal to build up loyal community members.

For the scenario of product configuration this means that customers are enabled to communicate with each other about products and product configurations, the actual topic of interest in a product community. Additionally, customers can establish relationships with other customers, they can be members of groups within the community and they can adopt certain user roles. For example in the community a customer could become an expert on specific parts of the product model. The communication and collaboration between customers can take place in synchronous and asynchronous ways, as described in the following.

2.2.2 Asynchronous collaboration

A first example for asynchronous collaboration is the access to previously configured products and components of other community members, which can be presented to the customer through ‘participatory catalogues’ (Schubert, 2000). Such catalogues are enriched with ratings and comments of community members. Additionally, its elements can be filtered and arranged in special orders, like for instance, by name of its authors, by average rating of certain groups etc. This way the customer gets information about what others have previously bought and what opinions other customers have about these products and their contained components. This information is helpful for making his individual configuration decisions.

Another example of asynchronous collaboration is to actively ask other community members for their advice, e.g., by sending e-mails to them. Other community members may reply either by sending their opinions to the customer as pure text or they can assume control for the configuration of certain parts of the product and provide concrete configuration recommendations to the questioner.

2.2.3 Synchronous collaboration

In addition to different forms of asynchronous collaboration, synchronous collaboration between customers is also possible. The idea is to provide some sort of a computer-based ‘shared workspace’ (Miles et al., 1993) to community members, which enables them to watch others configure parts of their products or even collaboratively design and develop individualised goods.
In this scenario the participating users form a temporary team and can simultaneously access the configurator tool to mutually adapt and configure degrees of freedom. For such a collaborative configurator, different sorts of awareness tools (Schlichter et al., 1998) are necessary that show which of the participating team members is currently working on which part of the product configuration. Additionally, the members of the collaborative configuration task can be supported by a private chat tool, which enables them to communicate with each other online.

We decide that only the customer himself has the permission to invite other customers to his team for collaboratively configuring his product. Accordingly, the shared workspace is only accessible for those users explicitly entitled by the initiating customer.

2.2.4 Phases of customer collaboration

Described above is a scenario of customer collaborative product configuration. The whole process of collaboration can be divided into four phases, all of which have to be supported by a collaborative configurator tool (Leckner, 2003): First of all (a) the initiation phase, where potential partners have to be recognised and identified. After this, (b) in the agreement phase customers need methods and tools for defining goals, agreeing upon and safeguarding them. An agreement about rewards for a positive collaboration and penalties for a faulty one is also possible. The next step is (c) the execution phase, where the actual collaboration takes place. Here, customers can support each other and exchange opinions and experiences in synchronous and/or asynchronous ways. Finally in (d) the termination phase the whole collaboration ends and the achieved goals and the quality of collaboration can be evaluated. This also means that previously agreed rewards and/or penalties will be executed.

2.2.5 Differences with existing approaches

As far as the authors know, existing product configurators do not provide any collaboration functionality to customers. In fact the customer is completely alone during the process of virtually configuring his product, facing the challenges discussed above. The approach of a collaborative product configurator enables customers to benefit from the knowledge and experiences of others.

Compared with pure community support systems, the most important distinction is the link to the product model. The product model helps in structuring the community information space, since all community content is related to elements of the product model, e.g., ratings for components, opinions about degrees of freedom, etc. Additionally, for the purpose of collaboratively configuring a product, customers can ‘ad hoc’ agree upon mutual goals and execute a joint project, at least temporarily. Joint projects between community members are not typical for virtual communities and therefore mechanisms for negotiating, agreeing upon and safeguarding goals are necessary and important add-ons. Furthermore, the collaborative configurator differs from existing groupware applications (Greenberg and Roseman, 1996) which support the collaboration of professional designer teams in the B2B-area. As opposed to such tools the users of a collaborative product configurator normally do not know each other personally. And, even more important, members of the customer community have not agreed to an institutional secured goal, unlike users of most groupware systems.
2.3 Customer profiles
In order to generate personalised product recommendations, we need to model, acquire and store customer information. In this section, we briefly sketch our customer profile model, the import of existing profiles from other applications, and various methods for information acquisition while interacting with the customer.

2.3.1 Customer profile model
We developed a profile model that allows creating profile instances for new customers, which store all relevant personal information. The model is represented in XML as a semantically structured tree consisting of attribute-value pair nodes.

In the model, not only standardised basic personal information, lifestyle, interests and ratings of the user are represented, but also product and service related information relevant for our application domain. Since the profile model must interact with the various recommendation generation methods, these methods also had a strong influence on the model structure and contents. Rated items and interests, for instance, are necessary for the collaborative filtering method. Furthermore, the model must be flexible enough to be applicable to the three different information acquisition methods we developed (see below). It needs to be able to capture, for example, completely new and individual information on a customer, which is entered during a free-text natural-language dialog.

2.3.2 Importing customer profiles
In order to reduce the effort of providing personal information to the customer we have integrated the possibility of importing existing profiles and of allowing various applications to share profile information. The customer may enable access of the configurator to information he has entered for other applications (Koch, 2002; Koch and Möslein, 2003). A key aspect of this approach is that the customer is the owner of the profile information. He can modify the information at anytime and can define access rights for different applications and customers. For importing user profiles we propose a user profile management architecture that allows the storage and maintenance of user profile information independently from services which are using it (Koch and Wörndl, 2001; Koch, 2002). The core component in this architecture is a user profile repository service that stores information about an identity. It grants access to the profile information to the identity owner as well as to authorised service interfaces.

2.3.3 Acquisition and modification of customer information
For customers who want or have to create a user profile from scratch, we have developed a profiling tool that offers three different interaction modes among which the customer can choose according to his level of expertise and his personal preferences. The basic mode is especially suitable for novice users since it allows the entering of certain particularly important initial information in a well-explained questionnaire. The expert mode gives an overview of all possible user profile attributes and values and is also questionnaire-based. The dialog mode is a user-friendly natural-language interaction mode where the user answers questions asked by the system and which allows the user to enter his preferences in his own words (Stegmann et al., 2004). The profiling tool further allows the customer to see and modify his information at any time before, after or during the product configuration process.
3 The product specification tool

In this section, we present our configurator tool and describe its implementation of the support mechanisms that have been described in the previous sections. A potential customer may use the configurator to interactively specify his individual products. We first introduce our system architecture to explicitly show the structure and the most important modules of the configurator tool (Section 3.1). To explain the configuration functionality we also present our model of the product, including product components, and different types of degrees of freedom (Section 3.2). The tool integrates a product and component catalogue for easy browsing and visualises personalised product and component recommendations.

3.1 System architecture

For the discussion of the system’s functionality we roughly distinguish two different views onto the platform (see Figure 1):

1. the view of the product designer
2. the view of the customer.

The Product Designer, typically a member of the manufacturing company, is supported by the Product Model Editor and the Product Administration Tool. The Product Model Editor is a stand-alone tool that allows the product designer of a company to enter, store and modify product models and to visualise them in a user-friendly manner. The editor is based on the Protégé ontology editor and includes PAL, the Protégé axiom language, used for modelling and evaluating constraints. The Product Administration Tool constitutes the link between the Product Model Editor and the Web-based Product Configurator. Within the Product Administration Tool the product designer can provide additional information for the web application, such as images for components, textual descriptions for degrees of freedom, etc.

Figure 1 Platform architecture
For the customer the most important components of the platform are the Product Configurator, the Product Catalog, and the Profile Editor. After the login the Customer usually starts with the Product Catalog and selects the product model type he wants to configure. Then he enters the Product Configurator. He manually navigates through the product and adapts the values for the degrees of freedom in accordance to his individual needs and preferences. Also the customer may be supported by personalised recommendations of a product in form of restricted product models (see Section 3.3). Recommendations are given as a result of customer requests and are generated by the Recommender System, which controls the recommendation process and therefore needs the Product Model Manager, the Product Manager and the ID Manager to access the databases of product models, existing product configurations, and customer profiles.

If the customer’s profile does not include sufficient information for generating a personalised recommendation (see Section 2.1), the customer can either import an existing profile from another application or he can use the Profile Editor, which provides different modes of profile acquisition, as described in Section 2.3. Essential for both the profile acquisition and the generation of recommendations is the customer profile model, which has been described briefly in Section 2.3.

Profiles are stored in the ID-Repository and can be accessed and modified by the system, the customer and by other applications and customers that have been granted access rights. Product models and product configurations have been defined by the manufacturer (e.g., by the professional product designer via the Product Model Editor) or have been interactively created by customers. Product models and product configurations are stored in databases and the specific manager components support their maintenance. These managers also manage access rights for product models and configuration. For example, a customer can define which other customers are allowed to access his individual product configuration. Product models and configurations are also the bases for the dynamic generation of Product Catalogs, which give a personalised overview of available products and components to the customer and thus help to ease decisions during the configuration process. Product Administration Tool, Product Configurator, Product Catalog, Profile Editor, and community functionalities (see Section 2.2) are accessible via a web interface.

3.2 The product model

As stated by Tiihonen et al. (1998), a configurator tool ideally provides all product-related knowledge to the customer. Therefore, the tool must contain some representation of product-related knowledge, namely the so-called product model. The model describes the physical and logical structure of the product consisting of components and sub-components that can occur obligatorily, optionally or alternatively.

Additionally, the product model defines degrees of freedom, which are those elements of the model that can be directly modified by the customer. Examples are the product’s attributes and alternative/optional components. Each degree of freedom can have a range of valid values and a default setting. Certain restrictions and interdependencies between different degrees of freedom are also possible (Männistö et al., 2001).

The product model contains knowledge of interdependencies or physical restrictions for the different parts of the product; in fact it should contain in the ideal case all the actual knowledge about the product a human consultation expert would normally have.
The model focuses, however, on the representation of customer-relevant product characteristics. It does not include all technical details that are necessary for manufacturing a product based on this model. Figure 2 shows an example of a simple hierarchical product model that is structured as described above, and that can be used by our configurator tool.

**Figure 2** Example of a hierarchical product model

We suppose that the product model is created by a professional product designer as an abstract and extensible model of a product family using the Product Model Editor. Derived from the product model is the concept of so-called restricted product models, which represent completely or partially configured products and which are described in detail in the next section.

### 3.3 Restricted product models

#### 3.3.1 The concept of restricted product models

As described above, the degrees of freedom of a product have been predefined by a professional product model designer. For the configuration of his individualised product the customer is restricted to the possibilities provided to him by these degrees of freedom available in the product model. The degrees of freedom are the same for every customer and it is necessary to determine all degrees of freedom, before the configuration process can be completed.

The basic concept of restricted product models is the possibility for a specific customer to flexibly adapt the product’s restrictions to his individual preferences and needs. Through the use of restricted product models the system can provide different views of the product model to different customers. In consequence, the degrees of
freedom that are available and visualised for one customer may be quite different from the degrees of freedom of the same product model presented to another customer. The numeric interval attribute provides an intuitive example for the usage of restricted product models. Let us consider the possible values for the power of an engine, for example, which are constrained to [40...300 kW] in the product model. To adapt the model to a specific user we would show him possible values within the interval [110...200 kW]. But for another user, the restricted product model would contain the interval [80...130 kW] for the referred attribute.

In any case a restricted product model is a subset of the original model. This subset is created by using the customer’s profile. Therefore, a restricted product model can be considered a personalised view of the degrees of freedom of the product model.

Restricted product models are also of special relevance for the automatic generation of recommendations helping the user to overcome the complexity of the configuration and design process. A recommendation generated by the system will always be a restricted product model that reflects the degrees of freedom that best fit the customer. It is represented, for example, as a reduced number of components or attributes, as a changed attribute value range or as the setting of default values. The so-created restricted product model serves as a basis for the visualisation of this adapted view of the product model for the individual customer in our configurator tool as described in Section 3.5.

If a customer demands product features which cannot be realised within the degrees of freedom of the current restricted product model, the view of the model will gradually be enlarged back to the original product model, at least for the relevant degree of freedom. If the demanded feature cannot be realised even within the restrictions of the original product model, the model may be manually adapted by the product designer when possible and necessary.

### 3.3.2 Examples of restrictions

In the following we will illustrate how for available degrees of freedom an adapted view can be generated through the use of restricted product models. First, we discuss the degrees of freedom for the component tree: for alternative component models the restricted model may only provide a subset of the available alternatives for the specific component. In the extreme case the restricted model can reduce the number of alternative components to one. Then the customer has no choice left and degrees of freedom no longer exist. The same can be said for optional component models. By restricting an optional component model this degree of freedom can disappear such that the component is no longer optional, but is either permanently selected or permanently deselected.

In Figure 3 the component tree of a product model is illustrated and two different restricted models for this product model are shown. In Picture 1 the whole product model is visualised. Picture 2a shows the Restricted Product Model 1, a very simple, strongly restricted form of the original model. The degrees of freedom are severely reduced from those seen in the original product model, especially since in the original model each component has associated with it a set of attributes representing further degrees of freedom. Finally, Picture 2b shows the Restricted Product Model 2, another valid restriction for the product model in Picture 1. Here, only one alternative component model and one optional component model have been removed compared to the original product model in Picture 1.
Restricted product models can also influence the attribute models. For an enumerated set model, for instance, the use of restricted models is quite similar to the alternative component models. The number of possible attribute values will be restricted and in the extreme the degree of freedom will vanish, \textit{i.e.}, the attribute is fixed at a specific value.

More complicated are the circumstances for the numeric interval model. Numeric intervals are constrained by an upper and a lower boundary. In a first step the creation of a restricted product model will simply adapt these boundaries to define a subinterval. This idea is illustrated as Restriction 1 in Figure 4.

Another option is the possibility of splitting the original interval into two or even more independent intervals, each of them being a subinterval of the original. The restricted model of such a numeric interval model will be the union of all the independent subintervals. This kind of restriction of a numeric subinterval is illustrated as Restriction 2 in Figure 4.

For this kind of restriction we either need a new type of attribute model, or we have to use the logical operator OR to connect the independent subintervals. We decided on the second option, because from our point of view it is not practical for restricted product models to create new types of degrees of freedom. In consequence, a restricted product model no longer consists of a single interval model only. Instead it is a more complex concept in that it can consist of a set of interval models connected by logical operators.
3.4 Visualising the product structure and degrees of freedom

After having introduced the underlying concepts of our configurator tool, we will show how these concepts are visualised for the customer in the configuration interface. Since our product model is designed from a customer’s point of view rather than from a manufacturing point of view, we mainly visualise those components, features and functionalities that can be manipulated by and are of particular interest to the customer. However, since we need to offer many degrees of freedom to enable real individualisation rather than mere choice between predefined variants, our visualisation methods need not only facilitate customer usage, but also allow a complete overview of the product structure and product details.

As mentioned earlier, our tool is not bound to one product model type only, i.e., several types can exist from which the customer may choose (Leckner et al., 2004). We use the product and component catalogue as a means of visualising the appropriate information. The product and component catalogue helps the user to browse different product types and select a specific type and model as the basis for his individual adaptations. This also reduces complexity for the customer, since it does not require the customer to specify a complete product from scratch.

3.4.1 Navigation through the product

In our approach the product model is structured as a tree of product components (see Section 3.2), which represents a complete product. With the configurator tool the customer can navigate through this component tree. He can view and manipulate details of each component in separate parts of the browser. The navigation is supported by the navigation tree and the navigation bar (see Figure 5).

Figure 5  Navigation tree and navigation bar
The navigation tree allows the customer to browse through the subcomponents of the current component. The navigation tree is supported by the navigation bar on top, which indicates the actual position in the product as a component path. We have included the navigation bar, because we wanted to offer an easy way to move through the component hierarchy and also to get back to parent components or to the root component. The customer’s navigation through the product can be performed easily by clicking on a component name either in the navigation tree or in the navigation bar. The names of all components within navigation tree and bar are labelled with hyperlinks, which are created automatically while loading the respective webpage. The webpage itself is also created dynamically taking into account the ID of the actual product model, the component model ID, and the user ID of the customer.

3.4.2 Details of a component model

After having explained the navigation through the product structure, let us now consider the detail view of a product component which the customer has chosen in the navigation tree. The detail view is subdivided into three parts (see Figure 6):

1. The top part shows the name and the component or product ID, a picture and a short description of the component or product as well as the last date of configuration, the person who configured it, and the status of the correctness of the configuration.

2. The middle part describes and provides link tabs for different kinds of support methods which can be obtained.

3. The bottom part shows the degrees of freedom of this component, e.g., all attributes and types of possible values such as selection lists or selection ranges.

Figure 6  Detailed component view
Within this view the customer always has an overview of possible value ranges while changing attribute values manually. After value changes have been made, it is necessary to click the button ‘update values’ or press the return key to send the changes to the server.

### 3.5 Visualisation of recommendations

The fundamental idea of our prototype tool is to support the customer by providing methods of personalisation. Recommendations are generated, which can be given both for a complete product and for specific degrees of freedom only. Whenever a recommendation is provided, the customer has to decide whether or not to apply it to his own product configuration. Recommendations may have different effects in our prototype system and therefore we also need different ways of visualising them, which are illustrated in Figure 7.

**Figure 7** Visualisation of recommendations

One effect of a recommendation can be the concrete value setting for a degree of freedom. In this case the colour of the affected attribute’s or component’s name will be changed and the former value of the degree of freedom will be indicated below the newly set value. Another possible effect is the removal of values for a degree of freedom (e.g., hiding alternative components from the navigation tree, because they do not seem to fit the customer’s preferences). In that case the number of visible degrees of freedom will be indicated and the customer is enabled to return to the full model when wanted. Note that in each case the customer can determine whether to accept the whole recommendation or only parts of it or whether to return to the former values, larger visible value ranges, or the full range of components, etc. Thus, the intention of generating recommendations is never to impose the system’s will on the customer, but to assist and advise him during his configuration task by reducing complexity, when wanted.
3.6 Supplementary support for the customer

As described above, the customer’s decisions during the configuration process are supported by recommendation mechanisms. In addition to this, supplementary support functionalities have been integrated into our configurator tool: First, browsing the product and component catalogue is supported by the functionality ‘similar components’ and ‘similar products’. This functionality can be seen as a kind of ‘smart browsing’ and is especially useful if the customer has already set certain values of a component and wishes only to quickly review the configurations used by other customers for the same component. For example, if the customer has set some values for the component spray gun and clicks on ‘similar components’, he obtains a list of spray guns which have been configured by other customers in the past and which are similar to the attribute values chosen for his spray gun so far (see Figure 8).

**Figure 8** Similar components

The degree of similarity is indicated by percentages for each similar component that has been found. If he wishes, the customer can click one of the listed components and a new window opens (bottom of the right side in Figure 8), in which the customer can browse the component and accept it for his own product, by pressing the button ‘apply this component for my product’. The customer is supported during the configuration of single degrees of freedom. For each degree of freedom, either components or attributes, the user can obtain an overview of other customer preferences. Figure 9 shows two examples of this kind of support:

1. On the left side percentage values illustrate the customer preferences for alternative components previously selected in the navigation tree.

2. On the right side the top three of all formerly configured values are shown for an attribute in the detail view of a component.
Figure 9  Indication of customer selection preferences

The two functionalities described in this chapter are initial and rudimentary realisations of community concepts (see Section 2.2), which can help the customer to overcome uncertainties while manually configuring a product. We also use these functionalities to empirically evaluate the correctness of our recommender system and especially the history-based filtering method described in Section 2.1.

4  Summary and conclusion

We have presented a new approach for a web platform with integrated configurator tool and customer community, which supports customers during the complex process of individualising products. The problem we address is that configuring a product with many degrees of freedom can become quite difficult, frustrating and time consuming for the customer. With this contribution, we introduce different concepts to support the customer during configuration. Our work is located in the area of personalisation and mass customisation of complex mechatronic products.

Offering support with a configurator tool for individualised products resembles a tightrope walk. On the one hand, we want to offer as many degrees of freedom as possible for individualisation. On the other hand, we want to keep complexity low and offer the high reliability and predictability the customer is accustomed to, with standard products. In our opinion, a customer in this situation greatly profits not only from product recommendations and ease of product configuration (still offering the degrees of freedom he desires) but also from cooperation and exchange with other customers.

Thus, we have presented two methods of customer support:
1  automatically generated product and component recommendations
2  direct recommendations among customers.

For the automatic generation of recommendations we have, first, developed a comprehensive user model together with explicit profiling methods. Second, we have developed the concept of restricted product models and implemented different filtering techniques as well as a configurable filtering pipeline to combine these techniques. To enable customers to mutually support each other we combined the configurator tool with functionality of community support systems. In the virtual community, knowledge space customers can exchange opinions and experiences.

To illustrate the different support mechanisms we have presented a prototype system of a configurator tool in this contribution. In addition to a discussion of the user interface of the system, an overview of our system architecture was given in order to show the overall coherence of different modules. Most of the described work has been implemented but has not yet been tested empirically in a real e-commerce setting. There
are still topics left for further enhancement and worthwhile investigation. Currently we are working on the improvement of methods for customer profile acquisition. The importance of this is clear, considering that the quality of automatic recommendations depends directly on the quality of available customer information. Currently available methods for explicit profiling are not yet satisfactory and hard to employ in practice. In addition, we are further investigating methods of mutual customer support and collaboration during their configuration tasks.

Acknowledgement

This work is part of a larger research effort towards the local production of individualised products funded by the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG) within the programme SFB582 – ‘Marktnahe Produktion individualisierter Güter’. See http://www.sfb582.de/ for more information about this research effort.

References


**Notes**

1 See www.sfb582.de

2 http://protege.stanford.edu/