Product Configurators in Electronic Commerce – Extension of the Configurator Concept towards Customer Recommendation

Torben Hansen, Christian Scheer and Peter Loos

Author contacts:

Torben Hansen
Chemnitz University of Technology
Information Systems and Management
D-09126 Chemnitz, Germany
torben.hansen@s1998.tu-chemnitz.de

Christian Scheer
Johannes Gutenberg-University Mainz
Information Systems and Management
D-55099 Mainz, Germany
scheer@isym.bwl.uni-mainz.de
www.isym.de

Peter Loos, Dr., Professor
Johannes Gutenberg-University Mainz
Information Systems and Management
D-55099 Mainz, Germany
loos@isym.bwl.uni-mainz.de
www.isym.de

Abstract: Web-based product configurators are enablers of the concept of mass customization, as they allow customers to transform their perceptions regarding a desired product into a precise product specification. The customer chooses option values within set limits that best match his configuration goal. In this paper we present an approach to extend the traditional product configuration process with a consultation interface which offers option values on request. We describe both required architecture and process flow.

1. Methodology

Starting from the necessity to offer additional customer support in the product configuration, we depict the environment and functionality of product configurators in the electronic business-to-consumer commerce (cf. chapter 2). Besides several approaches to support the customer in the specification phase, we present the idea to generate recommendations for a customer decision (cf. chapter 3 and 4). Furthermore we discuss and evaluate different techniques to realize our idea in a deductive way of presentation (cf. chapter 5). As a result, we acquire an extended product configuration concept (cf. chapter 6) for customer recommendations including an extended configurator architecture (cf. chapter 7) and a process description (cf. chapter 8).

2. Product configurators in Electronic Commerce

In buy side markets, sellers integrate customers using the potentials of Electronic Commerce into their value chains. Customers become "prosumers" who are able to determine attributes of the desired product themselves within predefined limits.

The possibility of personalization can be realized within the **seller's process** using the concept of Mass Customization. Mass Customization, an oxymoron consisting of Mass Production and Customization, joins these two concepts in order to create products that offer "enough variety and customization" so that „nearly everyone finds exactly what they want" (Pine, 1993). Thereby basic product models[1] are created which can be modified regarding predefined attributes (or options) within predefined attribute values (or option values).[2] The configuration of an individual product is then realized by the customer in a special phase of customization. An additional degree of personalization can be achieved by offering user individual services, a unique product image, a variety of delivery options or even individually developed and produced product components. (Piller, 1998; Pine, 1998; Piller, 2001)

Within the **customer's process**, the buyer can choose from a range of offered options and option values. By combining desired options considering a defined product structure, the customer can construct an individual product. The interface between customer process and seller process includes all interactions between the two actors and can be supported using information and communication technology.

The early phase of product specification is of special importance regarding this interface, as it has direct influence on the succeeding phases in customer's process. The specification phase in the field of Mass Customization is supported by the use of **product configurators**. They allow the customer to transform customer's perceptions into a precise product specification using selection and assessment of offered options and option values within a predefined product model.

Symptomatically it can be said that the customer has the possibility to configure a huge number of products from a seller's view. In many cases though the customer aborts the configuration process by himself. There are several reasons stated in literature for why the customer behaves this way (Boston Consulting Group, 2000; Gesellschaft für Konsumforschung, 2001). Major problem areas include the lack of a customer desired option value regarding a specific attribute within the system as well as the inability of the customer to create definite preferences between certain option values. As a result, the user aborts the

---

[1] Will be explained in more detail in chapter 3 and 7.

[2] Basically, product configurators offer the functionality to combine product components and to determine product attributes both within predefined limits. In the following we describe both possibilities: the specification of options and option values.

configuration process and does not come up to the sales phase. At this point, the conceptional extension of the product configuration concept with a consultation interface would be useful. Doing so, the customer would be able to receive a recommendation “at the push of a button” on what option value to select.

3. Process and functionality of common product configurators

The basic functionality of a product configurator consists of gathering the requirements and specifications from the customer and of generating a product specification based on knowledge about the product configuration process.

The configuration process can be divided into three sub-processes (Maher, 1990; Brown, 1998):

1. Statement of construction: Gathering of user’s requirements regarding the desired product as well as the transformation into goals and initial terms concerning the configuration parameters.

2. Synthesis of construction: Iterative and stepwise specification of the configuration parameters regarding parameter attributes.

3. Evaluation of construction: Comparison of the selection decisions with the imposed requirements and goals. The sub-processes two and three run within a closed loop.

The result of the configuration process is a product description which specifies the product from customer’s perspective and displays customer’s perception. The construction synthesis is based in detail on a product model, which describes all options, option values, the structure of the options regarding the product to be constructed as well as sell side constraints[3]. The customer-specific product description (so called customer-specific product model) can be considered as an instance of the general product model. (Peltonen et al., 1994; Hedin et al., 1998; Tiihonen et al., 1998)

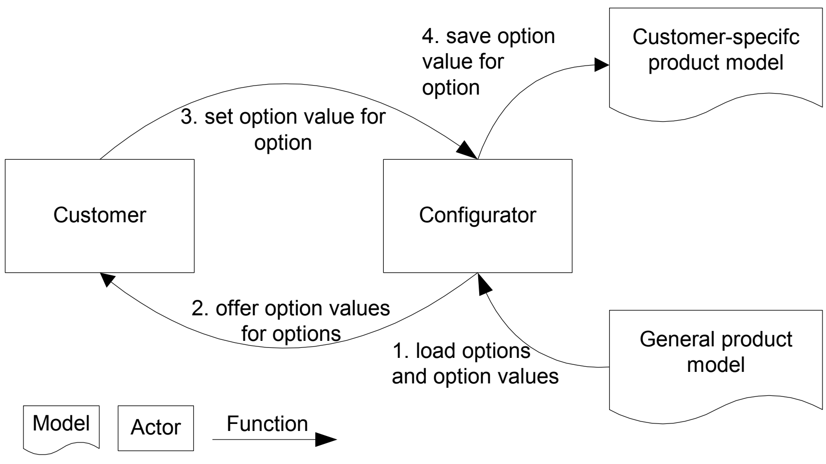

Figure I: Basic functionality of product configurators

Figure I illustrates the functionality of a configurator and displays the process of product configuration. The configurator offers options and option values of the general product model, requests value selections and stores them in customer-specific product model.

---

[3] Besides sell-side constraints, customer’s perceptions can also be formulated using contraints. E.g. maximal amount of money to spent.

4. Functional extension of the configurator concept towards a consultation interface

Based on the problem of user's abortions within the configuration process (cf. chapter 2), it is necessary to extend the configurator concept with a consultation interface. This interface can be compared ideally to the consultation service in a traditional market, in which a human sales person is able to lead the configuration process based on his knowledge of the product model and customer's requirements. The sales person is able to suggest option values to the customer which go along with customer's configuration goals and the previous specifications.

This starting point is already considered in literature where different consultation approaches are described. They can be classified as follows:

- **Recommendation of product specifications based on user's preferences**: A complete product configuration is suggested based on collected data about the customer using a variety of inference techniques. (for example: Fridgen et al., 2000; Cunningham et al., 2001; Inakoshi et al., 2001)

- **Ranking according product properties**: Starting from different products which could be suitable to support customer's perceptions, product specifications are ranked according to customer selected properties. So called computer-assisted self-exlication (CASE). (for example: Popp, 1997)

- **User adaptive configuration process**: Several levels of detail regarding product configuration, user and dialogue interface are implemented in real time based on collected data about the customer or the customer's profile. (for example: Ardissono et al., 2001)

- **3D product configuration**: It is also possible to support a configuration interface in 3D where the customer can visually configure his product. (for example: Miller/Müller, 2000; Detken/Fikouras, 2003)

- **Integration of a human consultant**: Some approaches - mainly based in the field of financial electronical services - favour the interconnection of a human consultant with the support functionality of a configurator. (for example: Buhl et al., 1999; Aberg/Shahmehri, 2001)

- **Integration of an artificial consultant**: An artificial consultant (e. g. an avatar) offers additional support regarding navigation and desired information. (for example: Attardi et al., 1998; Wiegran/Koth, 2000). Also software agents are used to realize an consultant which searches for suitable products and negotiates with the seller side. (for example: João/Ramos, 2000)

These mechanisms are helpful to support the customer in the complete configuration process. Especially the recommendation and ranking of product specification can be regarded as an important way to support the customer in the specification phase. Nevertheless existing approaches focus on the complete product specification. They are not suitable to assist the customer in the specification of single options where the customer requires help concerning the choice between several option values. From our point of view, this imposes an additional possibility to support the customer in the configuration process or the specification phase.

The basic model of a product configurator therefore needs to be extended with a consultation interface which offers advice at the click of a button regarding a specific option (cf. figure II).

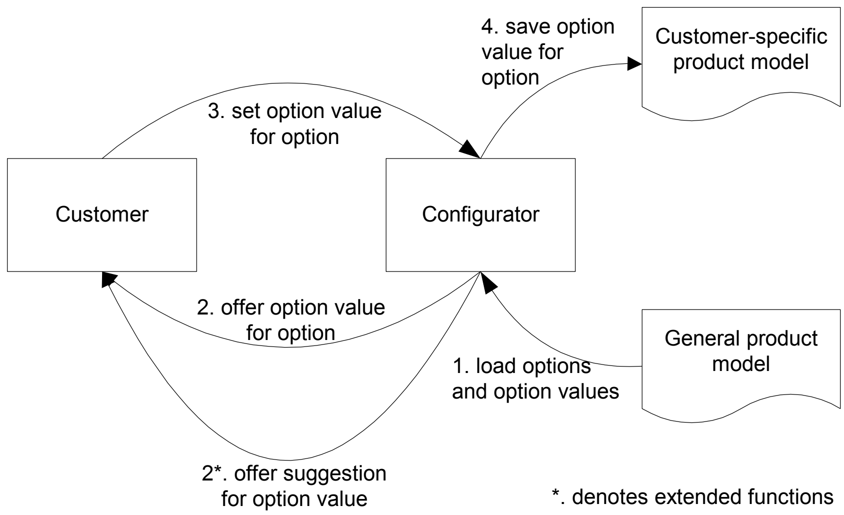

Figure II: The extended configuration process

This functional extension has far-reaching effects on the concept of configurators. Basically, the concept needs to be extended with some kind of representation of the customer within the system (a so called customer model), in order to be able to generate customer specific proposals. Depending on the used technique for consultation, it could be further necessary to use a representation of experience knowledge (a so called experience model) to generate predictions at all. (Scheer et al., 2003)

The highest impact on the extension of the configuration concept can be assigned to the techniques used for the generation of recommendations. Therefore it would be helpful to discuss possible techniques for the implementation of the consultation interface.

5. Techniques to realize the consultation interface in the configuration process

In the following we describe possible techniques to realize the consultation interface and evaluate them according to the special context of product configuration and specification.

To increase the transparency of the configuration process and to reduce the amount of process abortions initiated by the customer, there is a need to realize customer-individual support. This support is geared towards specific configuration situations, in which the customer requires help towards certain option values. In order for any mechanism to create a precise prediction, there is the need to learn about the preferences of the customer first. (Rosewitz/Timm, 1996; Terveen/Hill, 2001; Rashid et al., 2002) Therefore, two ways of **information retrieval** need to be distinguished initially (Lalmas, 1995; Rieger et al., 2000):

- **Customer active (qualitative) data collection**
- **Customer passive (quantative) data collection**

Customer active data collection: A registration process is used to query the customer about his preferences and goals so that the collected information can afterwards be used as basis for individual support. The provided information is explicitly issued from the customer, assumed to be correct, and therefore does not require any steps of reasoning. Drawbacks include the necessity of the customer to successfully complete the registration process in order to be able to begin with the option selection. Furthermore, the required information needs to be detailed, if customer help is supposed to assist regarding every option. These factors may lead to an abortion of the registration process and therefore of the configuration process as such. Exclusive use of active data collection in complex configuration situations is therefore not an appropriate way of learning about user's preferences due to a lack of detail causing a rather low quality prediction.

Customer passive data collection: The configuration goals of the user are assessed based on his behavior until he or she requests additional help. The behavior of the customer is assessed and his previous option value choices are used to describe his configuration goal. Passive data collection strategies include the use of cookies and clickstream analysis in the case of internet technology. By analyzing the customer's behavior, the uncertainty about the correctness of the provided information is avoided, additionally the user can start configuring right away. Problems occur, if the customer asks for support at the very beginning of the configuration process, since few information about his or her configuration goals are available. (Schafer et al., 1999; Rashid et al., 2002) In the following, we focus on customer passive data collection.

Once information is retrieved, the recommendation mechanism needs a strategy on how to transform customer information into precise option value **recommendations**. Generally, three common approaches[4] from the area of data mining can be depicted:

- **Content-based filtering**
- **Rule-based filtering**
- **Collaborative filtering**

Content-based filtering describes similarities between the data of an object and the corresponding user or customer profile (Basu et al., 1998; Meteren/Someren, 2000; Bridge, 2001). It can be used for website-recommendations or activities in which the main attributes of the object to be suggested consist of data. In the case of product configurators, a product consists of physical components instead of fully-indexable data, so that content-based filtering does not impose an adequate approach to use in a consultation interface. Furthermore problems occur with bugs in product description, for example with synonyms and homonyms. This is likely to take place in the specification of products by customers and sellers. Therefore content-based filtering seems to be not suitable for the consultation interface.

Rule-based filtering uses rules to describe coherences between a situation and a conclusion in this situation. It requires a large set of rules in order to make suitable statements about specific option values (Hayes-Roth, 1985). The construction of a rulebook creates the problem of having to manually derive several correct rules for every option value without creating contradictions. Furthermore new coherences have to be added manually or with the use of a separate data mining process. In most cases regarding the data overload, this is not a reasonable approach for the recommendation of products nowadays.

Collaborative filtering considers similarities not between objects themselves rather than between customers, who have used the system previously (Breese et al., 1998; Good et al., 1999; Sarwar et al., 2001). The similarity can be based on demographic customer data as well as the sequence of configuration and the chosen option values. Because of the implicit used experience knowledge of customers, collaborative filtering seems to be suitable in this product configuration context. Within collaborative filtering, two approaches can be sub-divided:

- **Memory-based filtering**: The comparison is done with the single profile in the database that matches the current customer best. The option value that the "nearest

[4] There are more approaches discussed in literature. For example, see the articles of Terveen and Hill or Burke (Terveen/Hill, 2001; Burke, 2002).

neighbor" has chosen for a specific option is applied to the current configuration run as an option value proposal. In large databases, the response times of the system will not be adequate, as every single past configuration run needs to be analyzed in search for the nearest neighbor. (Sarwar et al., 2000; Karypis, 2001; Sarwar et al., 2001)

- **Model-based filtering:** This approach abstracts the existing object preferences of the customer from the database and creates a model, which displays the relations between the object values based on customer's selection preferences (Ungar/Foster, 1998; Lin et al., 2001). Model-based filtering allows for faster responses in configuration situations. Predictions are not made by searching the entire database time-consumingly for similarities, but by analyzing the model which is highly abstracted and therefore of much smaller size, thus reducing search time. Model-based collaborative filtering can be realized using a variety of techniques, including association rule generation, bayesian networks or clustering techniques (Sarwar et al., 2001). Despite the problem that new models have to be generated at regular intervals to depict new or changed customer's preferences, the model-based filtering seems to be best suited as technique for the implementation of the consultation interface.

Furthermore it is possible that the generation of recommendations produces results which are unsatisfying in specific cases. Therefore a **validation** of the generated predictions can be added. By using methods which are based on active data collection, such as clustering techniques, a final comparison can be done in order to reconcile the model-based prediction with explicitly stated top-level configuration goals of the customer. (Scheer et al., 2003)

We conclude that the concept of collaborative filtering seems to be most promising to realize a consultation interface for product configuration given a certain minimal size of the database consisting of past configuration runs. Furthermore we prefer the model-based approach because of its better performance regarding configuration situations and databases. In order to determine which technique is applicable to realize this model-based collaborative approach, the usage site needs to be further analyzed and requirements need to be defined first. Finally a validation will be realized using clustering techniques.

The next chapters therefore deal with a conceptional extension of the traditional configurator concept and the related design decisions in order to present an architecture for a configurator with enhanced user support.

6. Extending the traditional configurator architecture with a consultation interface

The consultation interface is an extension of the traditional configurator architecture. This traditional architecture shown in figure III includes a description of the product structure and options (so called taxonomy) and configuration constraints in the general product model, the configuration functionality in the configurator as well as the active customer in his configuration process. The result of the configuration process is a customer-specific product model which describes the product specification in the view of the customer and can be regarded as an instance of the general product model.

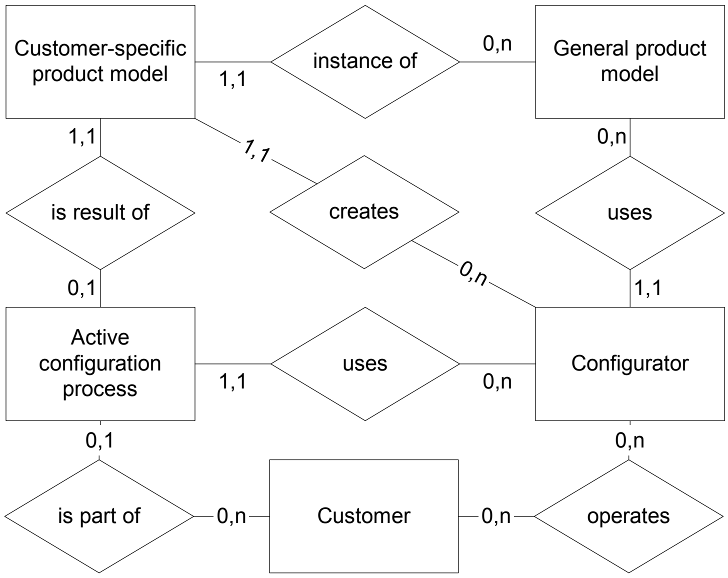

Figure III: Basic architecture of the traditional product configurator

The additional consultation interface shall enable the customer to request a recommendation for a single option at the push of a button that goes along with customer's configuration goals. This extension is aimed at keeping the customer within the configuration process by offering a recommendation in every configuration step if desired. At this point, a recommender system can be used. It creates based on all possible option values exactly one option value or a weighted list of option values which matches user's current configuration best. The traditional architecture of configurators therefore needs to be extended with a model consisting of customer's data and representing customer's situation. Furthermore coherences between customer's models and an "ideal" customer-specific product model have to be depicted in the so called experience model. As a result, the recommendation is done by assigning and propagating coherences of the experience model to a customer's model by the consultation interface.

This results in the following design decisions:

- Customer related data regarding user's demographics, preferences and requirements are stored in a **customer model** (Kobsa et al., 2001).

- Based on the presented approaches to realize a recommender system in chapter 5, we prefer the usage of a model-based collaborative approach as **consultation interface**. Content-based filtering is not appropriate for the presented situation as there is no data stored within the product model that would match customer's requirements syntactically. Traditional rule-based filtering causes problems with the amount of rules to be manually defined; additionally the rule set needs to be consistent. The basic approach of rule checking though is evaluated again in the context of artificial intelligence and automatic rule generation in chapter 7. Memory based filtering without usage of a model will not satisfy the needs of the system regarding the response time and overall system performance when used with large amounts of data.

- An **experience model** is used for representation of the concept of collaborative filtering. It depicts coherences between customer's models and the product model.

- A **control system** serves as the mediator between all components related to the generation of recommendations.

- As mentioned before we prefer the validation of generated recommendations in order to be able to eliminate improper recommendations in certain cases. We therefore require a

clustering engine which classifies all users into predefined categories and helps to search for wrong coherences. These categories are stored within a **user class model**.

The content of all models is stored within **XML-structures**. We use the SOX-extension for XML for the representation of object-oriented product models. The planned prototypical implementation is based on a Java-Servlet solution (in detail using the struts framework) which uses a XML-parser to realize both configurator as well as consultation interface in addition to processing XML-files.

7. Extended architecture and consultation process in detail

In the following, the design decisions as outlined in chapter 6 are presented in detail; figure IV displays the extension in comparison to the architecture as shown in figure III.

The **product model**[5] describes the options $Op = \{op_1, op_2, ..., op_o\}$, the corresponding option values $Opw(op_i) = \{opw_{i1}, opw_{i2}, ..., opw_{iw}\}$ and the structure of options $Op(op_i) = \{op_1, op_2, ..., op_o\}$ of the product to be customized. It also features constraints $Con = \{con_1, con_2, ..., con_c\}$ which are a result of technical or logistic requirements on the seller side. The result of the configuration process therefore needs to bring forward an instance of the taxonomy ($Op^* = \{op_1, op_2, ..., op_a\}$ whereas $Op^* \subseteq Op$ and $Opw^*(op_i) = \{opw_{i1}\}$ are valid while $Opw^*(op_i) \subseteq Opw(op_i)$) satisfies the given constraints. This instance is also considered to be the customer-specific product model or **product model instance** although no constraints are included. We prefer using an object-oriented product model because of its capabilities regarding inheritance and encapsulation (Peltonen et al., 1994; Hedin et al., 1998; Tiihonen et al., 1998) which offers advantages in modelling and maintenance of the product knowledge. By using constraints, the user is only offered these options which are actually available in the current configuration situation. As a result, no corrections of the chosen option values need to be manually applied afterwards.

The **customer model** stores user data which is necessary for determining the configuration goal of the user and creating the user profile. It consists fairly similar to the product model of a variety of data fields, selectable data components to be used with the data fields and constraints. In addition there are data fields which allow for arbitrary data input. The customer-specific customer model or **customer model instance** is created within a registration process prior to the begin of the actual product configuration. It is based on the customer model and includes all option values as selected by the user within the registration.

The **clustering engine** assigns the customer based on the information in his customer model instance to predefined categories which seem to match his configuration goal as stated exclusively within the registration best. This classification is later on used for validation of recommendations.

The **user class model** stores predefined categories along with **correlation ratios** which display the probability of each option value corresponding with the configuration goal of the user category. The amount of user categories depends on the desired degree of detail (Rijsbergen Van, 1979). Correlation ratios are determined by the frequency with which a certain option value was selected prior by users of the same user category.

The information about past configuration runs is stored as described earlier on within the product model entities. These product model entities are stored in the **experience model**

---

[5] Despite being an essential part of already the traditional architecture, the product model is described in this context in order to illustrate necessary coherences of the model to be created further on.

resembling the concept of model-based collaborative filtering. We impose the following requirements towards the capabilities of this model:

1. The user needs to be able to request a recommendation which matches his configuration goal as much as possible, anytime after having completed an initial configuration step.

2. The user is allowed to make selections in no specific order. It has to be possible, to leave out certain options and to return to answering these later on.

3. User selections within the configuration runs directly need to have an impact on the recommendations to avoid creating a static consultation interface which is exclusively based on registration data and does not include users choices in the configuration.

In order to create such an experience model, some kind of technique needs to be applied to the database in order to extract information and to store it in the experience model. Several approaches can be identified for doing so, all of them can be classified within the field of "machine learning" (Sarwar et al., 2001):

- **Bayesian networks**

- **Clustering techniques**

- **Association-rule-based approaches**

Bayesian networks are used for storing and processing of uncertain knowledge (Heckerman/Wellman, 1995; Jensen, 1996). Tree structures are the basis for the model creation, as objects are each represented as nodes within the tree. Each node has a table of probability attached which can be used to calculate the likelihood of the occurrence of the succeeding object. If a succeeding object only depends on one parent, there is a serial dependency among those two objects. Otherwise, every involved node needs to be considered in calculation. A general problem of bayesian networks in connection with product configurators is the lack of flexibility. If the customer is allowed to advance freely in the selection process, the tree structure either has to be constructed in a way that every object is interconnected, or that the tree generation needs to be done dynamically. In both cases, the response times of the system are inadequate for product recommendation.

Clustering techniques group customers into classes by identifying similarities between the demographics of these customers. Several methods can be used to realize this classification including the Bayes Classificator or Learning Vector Quantization Nets. (Ungar/Foster, 1998) There are two generic problems attached to this approach: There needs to be a huge amount of predefined user groups in order to generate precise predictions. This implies that there has to be an equally high amount of information gathered in order to classify the customer best. This again causes, as previously stated, the necessity of an elaborate registration process which needs to be avoided to keep the customer in the process. Also our requirement of not having a static consultation interface is violated, as all recommendations are set already before the user selection process even begins.

Association-rule based approaches focus on individual object relations within a database. Association rules have the purpose of identifying formerly unknown strong relations between objects and to classify these relations according to a set of parameters (Lin et al., 2001). Each object relation is characterized by values, which describe how close one object is interconnected to a second object. Based on the previous selections of a customer, the experience model consisting of strong association rules is queried concerning the appearance of any previously selected option value in combination with an option value on the other side of the association rule that satisfies the constraints set by the product model and can be used for prediction.

Considering the requirements stated earlier on, association rules seem best suited for the purpose of realizing a recommendation interface. In the field of association rules there are several approaches on how to differentiate between relevant and less relevant object relations. In the following, we will choose the confidence/support framework to measure qualitative and quantitative relevance (Agrawal et al., 1993).

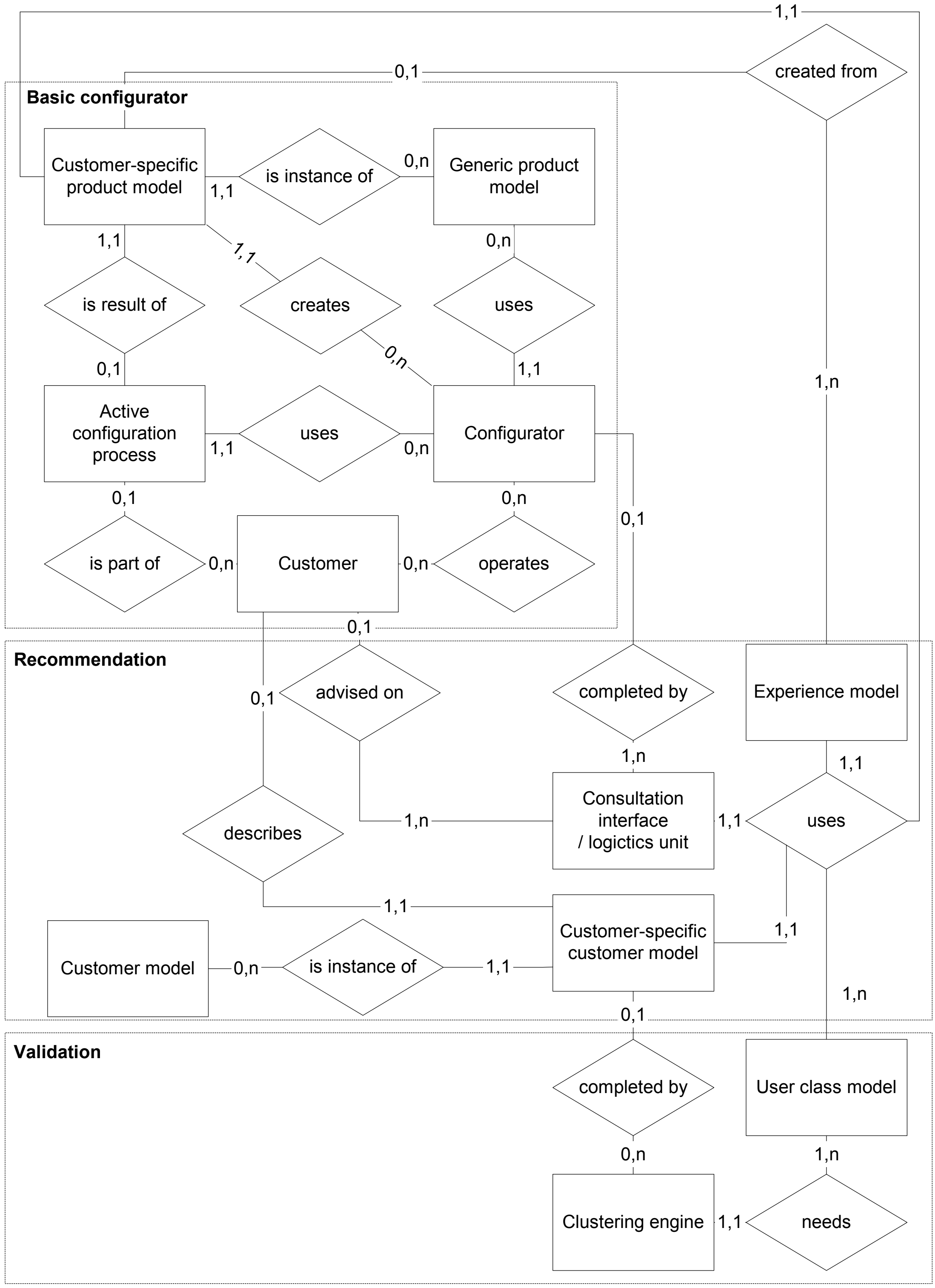

Figure IV: Extension of the architecture with a consultation interface and clustering engine

As all objects have relations with each other of some kind, there needs to be a filter which separates relevant from irrelevant object relations (Agrawal et al., 1993). We use the confidence/support framework to define requisites towards the object relations. In order to do so, we first need to equate traditional transactions as known from the field of shopping basket analysis with past product configuration runs. The value of the parameter support describes the frequency with which two objects appear together within all past configuration runs. The value of the parameter confidence describes the occurrence of configuration runs that include option value B among all configuration runs that also feature option value A. If both values pass the defined threshold, a strict association rule is created and the rule is stored within the experience model.[6]

The **control system** depicts the center of the recommendation process. It issues requests to the experience model, based on the customer data as stored in the customer model entities. Its queries include information about customer's configuration goals in order to extract knowledge from relevant past configuration runs. The control system also validates the query results with information stored in the user class model in order to finally recommend the option value, that matches user's configuration goal best.

The **consultation interface** contains all processes which are relevant for the generation of recommendations. Specifically, these are the registration process, the classification process, the transformation process, the recommendation process and the validation process.

- The **registration process** is basic for the validation of recommendations. The customer directly submits first-hand but superficial information about his configuration goals. The registration process is comparable to the configuration process in which the customer also creates his specific instance (product model instance) by selecting options based on the generic product model. Customer model entities don't necessarily have to be similar in structure as constraints within the customer model may lead to different outcomes and different options to be answered as a result of certain option value choices. The advantage of including constraints within the customer model is the possibility to get relevant information from the customer while keeping the interview short and down to the point.

- Within the **classification process**, the clustering engine assigns customer's profile to a certain user category. The clustering engine can work either rule-based or case-based; because of the different possible structures of the customer model entities and the predefined user categories that are known in advance, we prefer a rule-based approach. The clustering engine completes the customer model entities by adding a classification parameter to each instance resembling the user category the customer was assigned to. Each user category again is defined within the user category model which stores the correlation ratios for each category. Correlation ratios are numerical values between 0 and 1 that were calculated by statistical analysis for past configuration runs of users of the same user category (Rosewitz/Timm, 1996). These ratios help avoid unsuitable recommendations by comparing the explicitly stated configuration goals that are assumed to be correct with the recommendations based on the experience model.

- The **transformation process** stores the customer model entities representing past configuration runs in the association-rule based experience model. In order to do so, common object relations need to be identified in the product model entities. An algorithm that can be used for this purpose for example is the Apriori-algorithm (Agrawal/Srikant, 1994). When the algorithm terminates, all candidates have been identified: All object relations that were found, appeared reasonably often together within the total amount of past configuration runs. Still it needs to be determined which relations are also to be

---

[6] Strict association rule in the meaning of a rule with a relevant content.

considered not only common but also relevant. Therefore, the value of the parameter confidence of each relation is compared to the defined threshold. All relations that pass are considered strict rules and are stored in the experience model in the form of x=>y.[7]

- The **recommendation process** is initiated by the control system and generates a recommendation based on the customer model instance and product model instance of the specific user and the experience model. Initially, all (strict) association rules within the experience model are analysed. If the partner of a rule contains an option value Opw $(op_i) = \{opw_{i1}, opw_{i2}, ...\}$ that is appendant to the option for which the recommendation request was issued by the user, the rule is filtered out for further examination (trigger => Opw (op_i)). In a second step, only these rules are kept, where the trigger of a rule is also identical with an option value that the customer has selected so far within the current configuration run and that therefore are stored already in the product model instance (Opw*(op_a) => partner). In order to be a possible candidate for recommendation, the partner in the association rule on the one hand needs to be associated with the option the recommendation was requested for. Also, the association rule needs to have a trigger, which resembles an option value that the customer has selected earlier on ((Opw*(op_a) => Partner), whereas $_a$ specifies the already selected option values). Among the group of rules that satisfy these constraints, the partner of the rule with the highest value of the parameter confidence would then be chosen for recommendation.

It is possible that the described approach for generation of recommendations produces results which are unsatisfying in specific cases, especially when the user starts configuring in detail without having selected any more general options. We therefore propose a validation process following after the recommendation process in order to double check the results using clustering techniques (Ungar/Foster, 1998). It needs to be determined whether the generated results match the explicitly stated top level configuration goal as laid out by the user in the registration process. For this purpose, the classification of the user into a user category is used. Each user category has correlation ratios for every option value assigned to them stating the probability of an option value matching the configuration goal. The selection of a recommendation value is therefore not only based on the highest confidence value but on the product of confidence value and correlation ratio.

8. The configuration process including the consultation interface as a whole

The complete configuration process including the consultation interface and clustering engine (shown in figure V) covers the following processes[8]:

0*. At first, the customer data is collected within a registration process and a classification of the customer into predefined user categories takes place. The correlation ratios for option values are calculated and stored within the user class model (classification process) and the experience model is built based on past product model entities (transformation process).

1. The customer starts out with the configuration. The configurator reads out the product model and offers all possible option values for each available option in several configuration steps.

2. The customer can now select an option value, go on to another option, or request a recommendation.

---

[7] x is also called the trigger, y the partner.
[8] Processes marked with a star were added in comparison to the traditional configuration process.

3. All customer option value selections are stored in the product model instance. The configurator reads out past selections and offers only these options and corresponding option values that satisfy the constraints included in the product model.

3*. If the customer requests a recommendation for a specific option, the control system reads out customer model instance, product model instance and the experience model (recommendation process). Strict association rules are filtered out, if past selections occur as trigger of the rule and possible option values as the partner of the rule. The partner in the relation with the highest value for the parameter confidence is selected for recommendation.

4*. In order to avoid bad recommendations, a user classification based on clustering techniques is used in combination with the correlation ratios within the user class model. Recommendations are now made by selecting the rule with the highest product of confidence value and correlation ratio. If no association rule was filtered out, the prediction can still be made exclusively on the basis of the correlation ratios.

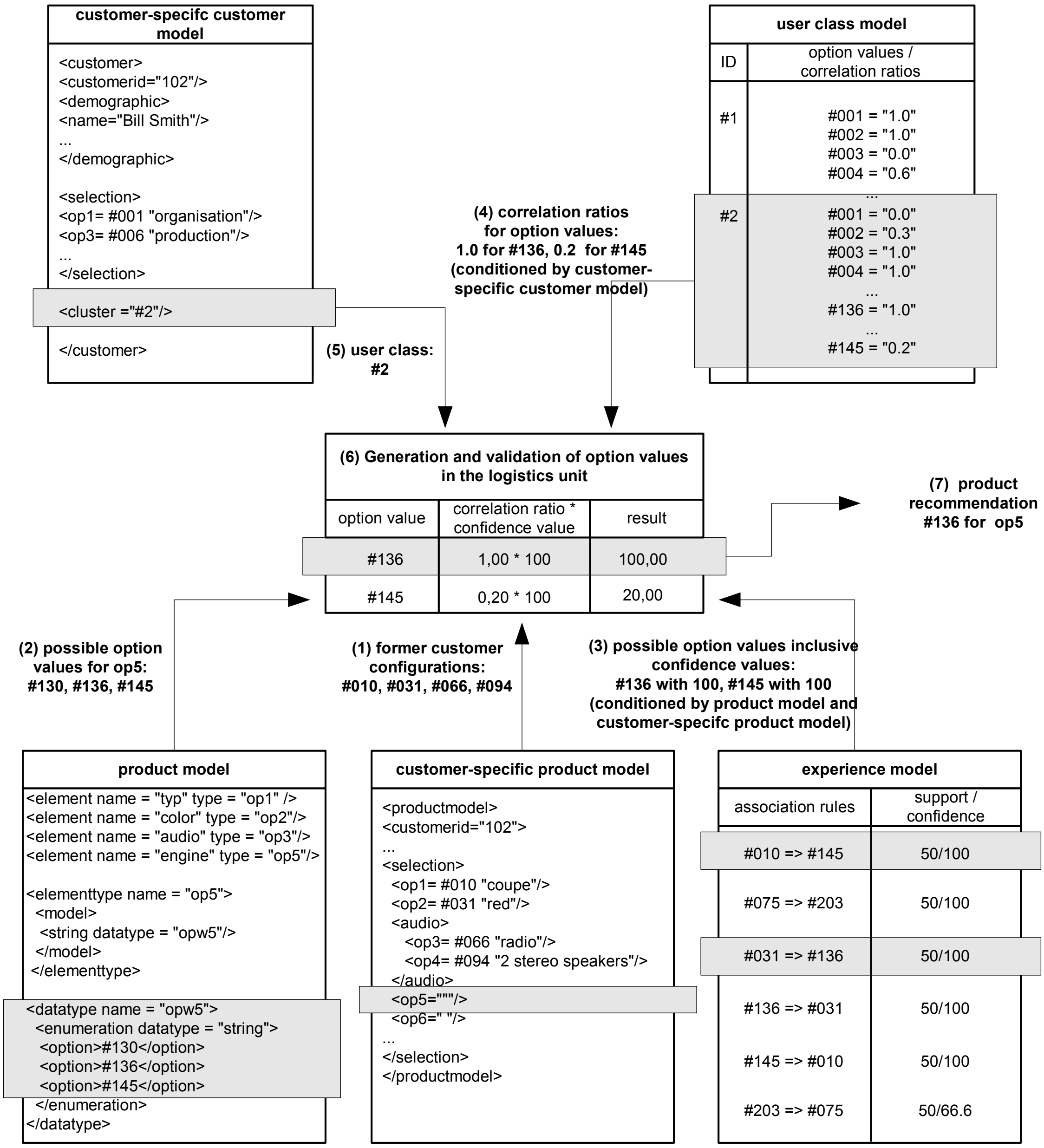

Figure V: The consultation interface and clustering engine for product configuration

The generation and validation of recommendations is visualized in Figure V. Based on selections within the current configuration run (1) and all possible option values for this specific option (2), one value needs to be preferred. Therefore all matching association rules are filtered within the experience model (3). The user class is read out of the customer-specific customer model (4) and the correlation coefficient of each candidate for recommendation is looked up (5). Then the recommendations based on the experience model are re-evaluated concerning the suitability for a member of the specific user class (6). As a result, one option value is returned as a recommendation (7). In figure V, extracts from XML-documents taken from a prototypical implementation are used for representation of the various models.

9. Summary

In this paper we have shown ways to enhance product configurators with consultation functionality. In detail we have presented and evaluated techniques and methods which can be used for realization. As a result, we have focussed on approaches within the field of collaborative filtering which seems best suited regarding the requirements of a consultation interface for product configurators. Based on this work, we then described our approach in realizing an efficient consultation interface allowing the user to immediately request support at any time in the configuration process and receiving precise recommendations for single option values that match users configuration goals.

References

Aberg, Johan / Shahmehri, Nahid (2001): An Empirical Study of Human Web Assistants: Implications for User Support in Web Information Systems, Proceedings of the SIG-CHI on Human factors in computing systems, Seattle, USA 2001.

Agrawal, Rakesh / Imielinski, Tomasz / Swami, Arun (1993): Mining Association Rules between Sets of Items in Large Databases, Proceedings of the 1993 International Conference on Management of Data, Washington D.C., USA 1993.

Agrawal, Rakesh / Srikant, Ramakrishnan (1994): Fast Algorithms for Mining Association Rules, Proceedings of the 20th International Conference of Very Large Data Bases, Kaufmann, M. 1994.

Ardissono, L. / Felfernig, A. / Friedrich, G. / Jannach, D. / Zanker, M. / Schäfer, R. (2001): Customer-Adaptive and Distributed Online Product Configuration in the CAWICOMS Project, Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI) - Workshop on Configuration, Seattle, USA 2001.

Attardi, Giuseppe / Cisternino, Antonio / Simi, Maria (1998): Web-based configuration assistants, Artificial Intelligence for Engineering Design, Analysis and Manufacturing, Vol. 12, No. 4, pp. 321-331.

Basu, Chunki / Hirsh, Haym / William, Cohen (1998): Recommendation as Classification, Proceedings of the Fifteenth National Conference on Artificial Intelligence, Madison, Wisconsin, USA 1998.

Boston Consulting Group (2000): Ecommerce Sites Still Failing Shoppers, http://www.plexusguam.com/NewsBytes.doc, 2002-11-16.

Breese, John S. / Heckerman, David / Kadie, Carl (1998): Empirical Analysis of Predictive Algorithms for Collaborative Filtering, Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, Wisconsin, USA, Morgan Kaufmann Publishers 1998.

Bridge, Derek (2001): Product Recommendation Systems: A New Direction, Proceedings of the Workshop Program at the Fourth International Conference on Case-Based Reasoning, Vancouver, British Columbia, Canada 2001.

Brown, David C. (1998): Defining configuring, Artificial Intelligence for Engineering Design, Analysis and Manufacturing, Vol. 12, No. 4, pp. 301-305.

Buhl, Hans Ulrich / Visser, Volker / Will, Andreas (1999): Virtualisierung des Bankgeschäfts, Wirtschaftsinformatik, Vol. 41, No. 2, pp. 116-123.

Burke, Robin (2002): Hybrid Recommender Systems: Survey and Experiments, User Modeling and User-Adapted Interaction, Vol. 12, No. 4, pp. 331-370.

Cunningham, Pádraig / Bergmann, Ralph / Schmitt, Sascha / Traphöner, Ralph / Breen, Sean / Smyth, Barry (2001): WEBSELL: Intelligent Sales Assistants for the World Wide Web, Künstliche Intelligenz, Vol. 2001, No. 1, pp. 28-32.

Detken, Kai-Oliver / Fikouras, Ioannis (2003): Lean Configuration: Interactive 3D Configuration for E-Commerce Environments, in: J. Gasos / K.-D. Thoben (ed.): E-Business Applications: Technologies for Tomorrow's Solutions, Berlin et al.: Springer. 2003, pp. 103-122.

Fridgen, Michael / Schackmann, Jürgen / Volkert, Stefan (2000): Preference Based Customer Models for Electronic Banking, Proceedings of the 8th European Conference on Information Systems (ECIS), Vienna, Austria, Gabler 2000.

Gesellschaft für Konsumforschung (2001): Online-Monitor Welle 7, Hamburg: G+J Electronic Media Service 2001.

Good, Nathaniel / Schafer, Ben J. / Konstan, Joseph / Borchers, Al / Sarwar, Badrul M. / Herlocker, Jonathan / Riedl, John (1999): Combining Collaborative Filtering with Personal Agents for Better Recommendations, Sixteenth National Conference on Artificial Intelligence 1999.

Hayes-Roth, Frederick (1985): Rule Based Systems, Communications of the ACM, Vol. 28, No. 9, pp. 921-932.

Heckerman, David / Wellman, Michael (1995): Bayesian Networks, Communications of the ACM, Vol. 38, No. 3, pp. 27-30.

Hedin, Görel / Ohlsson, Lennart / McKenna, John (1998): Product Configuration Using Object Oriented Grammars, Proceedings of the 8th International Symposium on System Configuration Management (SCM-8), Part of ECOOP'98, Brussels, Belgium 1998.
Inakoshi, Hiroya / Okamoto, Seishi / Ohta, Yuiko / Yugami, Nobuhiro (2001): Effective Decision Support for Product Configuration by Using CBR, Proceedings of the Fourth International Conference on Case-Based Reasoning (ICCBR), Workshop Case-based Reasoning in Electronic Commerce, Vancouver, Canada 2001.
Jensen, Finn Verner (1996): An Introduction to Bayesian Networks, New York: Springer 1996.
João, Maria / Ramos, Carlos (2000): An Agent-Based Model for an Electronic MarketPlace, in: Brian Stanford-Smith / Paul T. Kidd (ed.): E-business: Key Issues, Applications and Technologies, Amsterdam et al.: IOS Press. 2000, pp. 810-816.
Karypis, George (2001): Evaluation of Item-Based Top-N Recommendation Algorithms, Proceedings of the Tenth International Conference on Information and Knowledge Management (CIKM), Atlanta, Georgia, USA, ACM Press 2001.
Kobsa, Alfred / Koenemann, Jürgen / Pohl, Wolfgang (2001): Personalised hypermedia presentation techniques for improving online customer relationships, The Knowledge Engineering Review, Vol. 16, No. 2, pp. 111-155.
Lalmas, Mounia (1995): From a qualitative towards a quantitative representation of uncertainty in a situation theory based model of an information retrieval system, Information Retrieval, Logic and Uncertainty, Glasgow, Scotland 1995.
Lin, Weyling / Alvarez, Sergio / Ruiz, Carolina (2001): Efficient Adaptive-Support Association rule Mining for Recommender Systems, http://citeseer.nj.nec.com/483133.html, 2002-08-22.
Maher, Mary Lou (1990): Process Models for Design Synthesis, AI Magazine, Vol. 11, No. 4, pp. 49-58.
Meteren, Robin van / Someren, Maarten van (2000): Using Content-Based Filtering for Recommendation, Proceedings of ECML 2000 Workshop: Machine Learning in New Information Age, Barcelona, Spain, pp. 47-56.
Miller, Ann / Müller, Andrea (2000): FashionMe: The Future of Fashion Shopping over the Internet, in: Brian Stanford-Smith / Paul T. Kidd (ed.): E-business: Key Issues, Applications and Technologies, Amsterdam et al.: IOS Press. 2000, pp. 776-781.
Peltonen, Hannu / Männistö, Tomi / Alho, Kari / Sulonen, Reijo (1994): Product Configurations - An Application for Prototype Object Approach, Proceedings of the Eighth European Conference on Object-Oriented Programming (ECOOP), Bologna, Italy 1994.
Piller, Frank Thomas (1998): Kundenindividuelle Massenproduktion, WISU, Vol. 27, No. 8/9, pp. 875-879.
Piller, Frank Thomas (2001): Mass Customization: ein wettbewerbsstrategisches Konzept im Informationszeitalter, Wiesbaden: DUV / Gabler 2001.
Pine, B. Joseph II (1993): Mass Customization - The New Frontier in Business Competition, Boston, Massachusetts: Harvard Business School Press 1993.
Pine, B. Joseph II (1998): Einführung Mass Customization: Die Wettbewerbsstrategie der Zukunft, in: Frank Thomas Piller (ed.): Kundenindividuelle Massenproduktion: die Wettbewerbsstrategie der Zukunft, Munich, Vienna: Hanser. 1998, pp. 1-17.
Popp, Heribert (1997): Einsatz der Fuzzy-Technik in Industrie und Dienstleistungsbereich - ein Überblick, in: Jörg Biethahn/ Albrecht Hönerloh/ Jochen Kuhl / Volker Nissen (ed.): Fuzzy-Set-Theorie in betriebswirtschaftlichen Anwendungen, Munich: Vahlen. 1997, pp. 23-38.
Rashid, Al Mamunur / Albert, Istvan / Cosley, Dan / Lam, Shyong K. / McNee, Sean M. / Konstan, Joseph A. / Riedl, John (2002): Getting to Know You: Learning New User Preferences in Recommender Systems, Proceedings of the 2002 International Conference on Intelligent User Interfaces (IUI), San Francisco, USA, pp. 127-134.
Rieger, Bodo / Kleber, Anja / von Maur, Eitel (2000): Metadata-Based Integration of Qualitative and Quantitative Information Resources Approaching Knowledge

Management, Proceedings of the 8th European Conference on Information Systems (ECIS), Vienna, Austria 2000.
Rijsbergen Van , C. J. (1979): Information Retrieval. Department of Computer Science, University of Glasgow.
Rosewitz, M. / Timm, U. (1996): Basismechanismen für den Einsatz von Produktberatungskomponenten in Verbindung mit Elektronischen Produktkatalogen, 3. Workshop mit Minimesse "Elektronische Verkäufer mit und ohne Netz", Erlangen, Germany 1996.
Sarwar, Badrul / Karypis, George / Konstan, Joseph / Riedl, John (2000): Analysis of Recommendation Algorithms for E-Commerce, Proceedings of the 2nd ACM Conference on Electronic commerce, Minneapolis, Minnesota, USA, ACM Press 2000.
Sarwar, Badrul M. / Karypis, George / Konstan, Joseph / Riedl, John (2001): Item-based collaborative filtering recommendation algorithms, Proceedings of the 10th International World Wide Web Conference (WWW10), Hong Kong 2001.
Schafer, Ben J. / Konstan, Joseph / Riedl, John (1999): Recommender Systems in E-Commerce, Proceedings of the first ACM conference on Electronic commerce, Denver, Colorado, USA, pp. 158-166.
Scheer, Christian / Hansen, Torben / Loos, Peter (2003): Business Models to Offer Customized Output in Electronic Commerce, Journal of Integrated Computer-Aided Engineering, Vol. 10, No. 2, pp. 163-175.
Terveen, Loren / Hill, Will (2001): Beyond Recommenders Systems: Helping People Help Each Other, in: John M. Carroll (ed.): Human-Computer Interaction in the New Millennium, Boston et al.: Addison-Wesley. 2001, pp. 487-509.
Tiihonen, Juha / Lehtonen, Timo / Soininen, Timo / Pulkkinen, Antii / Sulonen, Reijo / Riitahuhta, Asko (1998): Modeling Configurable Product Families, Proceedings of the 4th WDK Workshop on Product Structuring, Delft, Netherlands 1998.
Ungar, L. / Foster, D. (1998): Clustering Methods For Collaborative Filtering, Proceedings of the Workshop on Recommendation Systems, Menlo Park, California, USA, AAAI Press 1998.
Ungar, Lyle H. / Foster, Dean P. (1998): Clustering Methods for Collaborative Filtering, Proceedings of AAAI Workshop on Recommendation Systems, Madison Wisconsin, USA, AAAI Press 1998.
Wiegran, Gaby / Koth, Hardy (2000): firma.nach.maß. Erfolgreiches E-Business mit individuellen Produkten, Preisen und Profilen, London et al.: Financial Times Prentice Hall 2000.