Applying Case-Based Reasoning Technology for Product Selection and Customization in Electronic Commerce Environments

Sascha Schmitt, Ralph Bergmann

University of Kaiserslautern, Department of Computer Science, Artificial Intelligence – Knowledge-Based Systems Group, P.O. Box 3049, 67653 Kaiserslautern, Germany
Phone: ++49-631-205-3359, ++49-631-205-3361; Fax: ++49-631-205-3357
e-mail: {sschmitt|bergmann}@informatik.uni-kl.de

Keywords: Intelligent Sales Support, Product Selection, Customization, Case-Based Reasoning

Abstract
Case-based reasoning technology opens a very promising and enriching new field for electronic commerce applications. However, current CBR applications for sales support in electronic commerce are limited to fixed, unchangeable products. In order to implement sales support for configurable products, the customization of base products must be supported as well. For this purpose, an interactive operator-based customization approach from CBR can be applied and realized as a flexible system for business-to-business as well as business-to-consumer applications. Such an approach is currently under development as part of the ESPRIT project WEBSELL1.

1. Introduction

Electronic commerce offers a huge variety of application areas nowadays. Much effort is put into research and exploration of these areas. However, research to address one of the most important problems, responsive online sales support, has generated only few promising approaches. Most electronic catalogs and online shops do not explore the interactivity available on the Web (Timm & Rosewitz, 1998). Hence, their searching

1 Project No. 27068. The partners of WEBSELL are tecInno GmbH (Prime Contractor - Germany), Adwired AG (Switzerland), Interactive Multimedia Systems – IMS (Ireland), IWT Magazin Verlags GmbH (Germany), Trinity College Dublin (Ireland), and the University of Kaiserslautern (Germany).
and indexing capabilities are often only a little more useful than their printed counterparts.

Driven by the challenge to improve Internet-based electronic commerce, case-based reasoning (CBR) technology emerged as a new important possibility (Wilke, 1998; Wilke, Lenz, & Wess, 1998; Vollrath, Wilke, & Bergmann, 1998). The main role for CBR in electronic commerce is intelligent sales support, i.e., the task to select a product from a list of alternatives which is most appropriate for the demands of the customer. Currently, several of such applications have been realized and are accessible on the Web already. (Some examples are accommodation booking\(^2\), last minute trips\(^3\), selection of digital signal processors\(^4\), operational amplifiers search\(^5\)). However, these applications are limited to fixed products that can only be chosen as a whole; configurable products have not been supported, yet. For selling products that can be modified to some degree, means for customization are required. Examples of such products are personalized vacations, technical equipment, e.g., computers (Wilke, 1998), or designs such as in electrical engineering (Vollrath, 1998; Bergmann, Vollrath, & Wahlmann, 1998) and many more. Intelligent sales support for such products has to involve customization, i.e., tailoring base products to the customer’s wishes. Customers entering an electronic shop may have certain specific ideas about the product they are looking for. They specify their ideas, requirements, or wishes in a query. It will very likely happen that the products found do not exactly meet the customer’s wishes. Then, the retrieved products have to be customized to these wishes, so that in the end, they are as close as possible to the actual query.

To realize this customization step, CBR offers several possibilities (Hanney et al., 1995; Voss, 1996; Wilke & Bergmann, 1998). Since each of these methods has different advantages and shortcomings, an appropriate approach for customization must be selected carefully.

In this paper, we argue that an interactive approach to operator-based customization is appropriate for many electronic commerce applications. We will first introduce CBR technology in the following section 2. Then, section 3 highlights the basic requirements for a customization component from a customer’s and a shop-builder’s point of view. The ensuing section 4 presents our general operator-based customization approach. Finally, section 5 comments on the state of implementation of this approach and discusses related work.

The results presented in this paper have been achieved as part of the ESPRIT project WEBSELL\(^1\). WEBSELL aims at developing CBR techniques and tools for sales support for Internet-based electronic commerce.

---

\(^1\) http://www.mueritz.de/mofruk3.htm

\(^2\) http://www.reiseboerse.com/

\(^3\) http://wwwagr.informatik.uni-kl.de/~readee/

\(^4\) http://imsgrp.com/analog/query.htm

\(^5\) http://imsgrp.com/analog/query.htm
2. Case-Based Reasoning Support for Electronic Commerce

In this section, we describe CBR as the fundamental technology for our approach. This technology is first motivated in the ensuing subsection. Section 2 ends with general considerations about benefits of the technology for electronic commerce.

2.1 The Problem with Ordinary Databases

CBR is closely related to the field of database technology. The motivation for the need of the relatively new CBR technology instead of standard database systems in electronic commerce environments comes from the fact that ordinary database systems do not provide sales support. Customers searching for product information on the Web will be dissatisfied with a huge database of products and a database search engine on top of it because for using the system one needs very special knowledge. Only a customer who knows the product part number is assured of finding the desired information. However, if the customer is not looking for a specific part but for a product with particular qualities, the success of the search is much less certain. Unless being very familiar with the contents of the database, the customer will get either no answer to the query or an excess of answers that might not be sorted in any usable way. The situation is even more frustrating if the customer does not know exactly what qualities to look for. An online catalog could provide intelligent sales support through a system that puts the knowledge of a shop assistant into its retrieval component. The system would require enough domain knowledge about the products to be able to aid customers in their searches. Such a system would not only satisfy the customers but also help the manufacturer or broker to sell products. Case-based reasoning is one way to implement it.

2.2 Case-Based Reasoning Technology

The basic idea of CBR is to solve new problems by comparing them to problems already solved (Aamodt & Plaza, 1994; Leake, 1996; Wilke, 1997; Lenz et al., 1998). The key assumption is that if two problems are similar, then their solutions are probably also similar. In electronic commerce, a problem is typically the assignment of a particular product to a set of demands (or requirements) stated by the customer. CBR systems are based on some measure of such similarity, i.e., products are selected based on the similarity of requirements.

2.2.1 CBR Concepts: Using Old Solutions for New Problems

Old problems and their solutions are stored in a database of cases — the case base. Often, the cases are stored as collections of attribute-value pairs, but for complex tasks it is useful to explicitly represent the hierarchical structure of the cases by describing them as structured objects, using inheritance, object decomposition, and possibly other relations between the object parts.

When a new problem has to be solved, the CBR system searches for the most similar old problem. The solution to this old problem can be adapted to more precisely meet the
requirements of the new problem. Figure 1 illustrates the steps taken in a case-based reasoning system.

![Figure 1. General processing within a CBR system.](image)

In our application scenario, the cases are descriptions of products. The problem description is a specification of a single product and possible demands the product can satisfy. The solution to the problem is an unambiguous reference to the product. For configurable products such as computers, automobiles, complex machines, and so on, the solution is not only the part number, but possibly the entire configuration. When a customer enters a query (perhaps into a query form), the query is regarded as a new problem and the CBR system tries to solve it by comparing it to the cases in the case base.

Two essential properties distinguish the approach described from ordinary database systems: cases are compared using special similarity measures and they can be further customized after retrieval by using adaptation techniques.

### 2.2.2 Similarity Measures

CBR systems have more specific domain knowledge built into them than ordinary database systems. The main part of this additional knowledge is implemented in a similarity measure, a function that assesses the similarity of a given query to the cases in the case base. The similarity values are ordinal values that are often normalized to the interval [0, 1]. A value of 0 means “does not satisfy the query at all” and a value of 1 says “that’s exactly what you asked for”.

To understand how such a similarity measure is used to find the best solutions for a given problem, consider the cases being represented as a fixed length vector of \( n \) attributes. These attributes can have numerical values or their values can be arranged to reflect some kind of order.

The problem space can be seen as a \( n \)-dimensional space where similar problems are placed closely together (see Figure 2). The term “close” is defined by the similarity measure. When a new problem is presented to the system, the similarity to all the cases in the case base is calculated\(^6\). The cases within a similarity range of a certain threshold (or sometimes a fixed number of similar cases) are then presented to the customer.

---

\(^6\) The number of visited cases can be optimized by suitable indexing techniques but their description is out of scope of this paper.
Figure 2. Cases distributed over an \( n \)-dimensional problem space. The circle around the query \( Q \) indicates a similarity threshold for the cases to be retrieved.

The usual approach to define such a similarity measure is to start with so-called local similarity measures for all the attributes. For numerical attributes a natural approach is to calculate the difference between the query value and the case’s value and normalize the result to the interval \([0, 1]\), for example, by applying the formula \( 1 - (d / (d + 1)) \) to the difference \( d \). But certain attributes require specialized functions because the similarity function must reflect the domain-dependent knowledge.

In real-world applications not all attribute values are numerical values. A case description very often contains boolean or symbolic attributes. In such cases, the local similarity measure can be described by a table that defines the similarities for all possible pairs of attribute values. Even more complex attribute types such as taxonomy types or complex objects are sometimes required.

Once the local similarity functions are defined, the global similarity of two cases must be derived from the local similarities. The usual way to do this is to apply a weighted sum to all the local similarities. Consider a query \( q \) is described by the attributes \( q_1, \ldots, q_n \) and a case \( c \) is described by \( c_1, \ldots, c_n \), where attribute values with corresponding indices belong to the same attribute. The similarity \( \sigma \) between \( q \) and \( c \) can be calculated from the local similarities \( \sigma_i \) as follows:

\[
\sigma(q, c) = \sum_{i=1}^{n} w_i \sigma_i(q_i, c_i)
\]

where

\[
\sum_{i=1}^{n} w_i = 1 \quad \text{and} \quad w_i \geq 0 \quad \text{for all} \; i.
\]

Although the weights \( w_i \) have to be assessed by a domain expert, they can be manipulated by the customer to express individual preferences.

This general method of defining a similarity measure as a weighted sum can be modified and refined in many ways. In general, the computation of the similarity measure depends on the data model and the type of application. A simple weighted sum is not always adequate. For example, nonlinear dependencies between the weights can cause the relevance of a certain feature to depend on the values of other features. CBR offers a number of possible solutions to account for such nonlinear similarity measures.

Complex measures, i.e., other than weighted sums, have the drawback of making it hard, if not impossible, to use specialized indexing techniques to improve retrieval times. However, the retrieval results can be very accurate.
Finding a suitable similarity measure is often the most critical part of the design and implementation of a CBR system. Once a good similarity measure has been found and implemented, maintaining a CBR system is rather easy. Typically, neither the similarity measure nor the domain model need to be changed for a long time compared to the knowledge bases of other types of expert systems. Often, the only part that undergoes significant changes over time is the case base.

2.2.3 Adaptation

Some CBR systems do not only support the retrieval process but they also adapt case solutions to the new problem and thus create new solutions. Of course, this requires the solution, here the product, to be configurable.

The knowledge needed to perform this kind of adaptation must be represented in some suitable form during the development of the CBR system. One possible representation is a set of rules that perform certain actions given that the required preconditions for the actions are valid.

Depending on the application domain, the adaptation process can be more or less complicated. To the authors’ knowledge there is currently no commercial CBR-based product catalog that supports adaptation. While many products cannot be adapted because they have no modifiable structure, adaptation is indispensable if the system must perform some kind of configuration task, like interactive configuration of personal computers or travel packages.

For an overview of adaptation principles see Wilke & Bergmann, 1998, for an overview of configuration technologies see Wilke et al., 1998.

2.3 Sales Support Benefits

In a typical sales support session, a customer enters a specification, i.e., parameter values, of the product wanted into the query form. The system then retrieves the ten best matches to the request. If the results do not exactly fit the parameters entered, the customer increases the priorities of the most important parameters. Again, the system displays the ten best matches to the refined query. If the results still do not satisfy the customer, more parameter slots previously left empty can be filled, thus further improving the quality of the returned results.

This is very similar to a real shop situation: a sales assistant learns the customer’s demands step-by-step until finally the product that best fits the customer’s needs is found. The real-world knowledge built into a CBR system is used to interpret the customer’s query and greatly enhances the quality of the retrieved data. Even the first request – typically a vague formulation – immediately produces usable results. And, if the system returns more than one answer, the results are ranked by their similarity to the query, making it easy for the customer to refine the request until satisfied with the results.
3. Requirements for Customization in Electronic Commerce

In this section, we analyze in some details the requirements for customization that occur in an electronic commerce application. Since it is very hard to formulate general requirements which hold for selling arbitrary products, our investigation is driven by general observations from a particular application domain: customization of vacations within a virtual travel agency. The domain of travels has a high volume of sales, this is why it plays an important role in the context of electronic commerce. However, we feel that these requirements should also hold for other kinds of products.

3.1 Intelligent Sales Support for Vacations

We concentrate on an exemplary domain of travel sales which can be easily understood. Imagine an online catalog of a travel agency\(^7\). Every single product, i.e., the tour packages in the present domain example, is represented by a case in the travel case base. As an example, such a case could have the following description (in parts):

```plaintext
case1=( "Caribbean", /* region */
    "Winter", /* season */
    "Airplane", /* transportation */
    10, /* duration */
    1999.99, /* price */
    "Beach", /* holiday type */
    1508, /* journey code */
    "HolidayFlat", /* accommodation */
...
)
```

We consider a customer who would like to go on vacation in the Caribbean in February. Our traveler does not want to spend more than DM 2500. In addition to these requirements, she wants to learn surfing during her vacations. Our customer enters these itinerary preferences into the system by using a certain query form. These preferences are interpreted as a problem description. For this query, let us assume that our retrieval system finds one or more similar cases that best match the query, among other things, there is a journey to the Dominican Republic for DM 2000, but there is no surfing course included in this offer. At this point, we have reached the stage that the customization system has to be deployed. The customization component checks the hotel which would be booked for the journey and finds out that there is a variety of sports activities which also include surfing lessons for DM 400. If all requirements are fulfilled, e.g., the duration of the holiday is long enough to complete the surfing course, the customization component could offer these lessons to the customer. This transaction can be totally transparent to the customer. If the customer confirms the offer, the course will be added to the product and the resulting price is increased appropriately.

\(^7\) See http://wwwagr.informatik.uni-kl.de/~lsa/CBR/wwwcbbrApps.html for examples of existing commercial applications in this area.
3.2 Requirements from the Customer’s Point of View

The approach for a customization component within a CBR system is dependend on a couple of important requirements from both perspectives, the customer’s and the shop builder’s point of view. First, we extract these requirements for the point of view of the customer, i.e., the user of this system component:

- **Customer control:** The customer should control and guide the complete customization of the product.
- **No complete a-priory product specification:** The customer should not be asked to completely specify his desired product in detail at the beginning. Only during the customization process, the customer might get aware of her detailed product requirements; she should be asked to state these requirements only when needed.
- **Maintenance of product consistency:** The consistency of the overall product must be maintained during customization; means are required to ensure that only appropriate customizations can be chosen.
- **Presentation of customization options:** The customer must be made aware of possible ways to customize the product; this is a pre-requisite to enable customer control.
- **Adequate response time:** The system should respond in adequate time, i.e., on the one hand, the customer should be presented her control facilities in the current sales situation as quickly as possible and on the other hand, the customer’s change wish should be processed very fast.

3.3 Requirements from the Shop-Builder’s Point of View

The second category of requirements, besides the ones from the customer, are additional requirements from the shop-builder that enables her to efficiently develop electronic commerce applications that involve customization. These requirements refer to knowledge representation issues and the efficient use of resources. The following list summarizes the relevant aspects from a shop-builder’s point of view:

- **Flexible representation:** “Flexible” means for representing the customization knowledge are required. The shop-builder should be enabled to represent the diverse possibilities for product customization in a natural way. A rich, but easy to handle representation mechanism is desired.
- **Integration with case/product representation language:** The representation of the customization knowledge should be integrated seamlessly with the underlying product representation approach.
- **Efficient use of bandwidth on the Internet:** The limited bandwidth on the Internet should be used efficiently; the overall customization process including possible downloads of Applets and other data should be as quick as possible.
- **Support of business-to-business and business-to-consumer:** The customization approach should be appropriate for both business-to-business applications and for business-to-consumer applications. It would be desirable if the same application could serve both kinds of customers.
3.4 Illustration of the Requirements for the Customization of Vacations

The customer always has to decide on her own what she wants to do, i.e. she keeps the control of product selection and customization. This does not mean that it is necessary to exactly know beforehand what the product should look like. With respect to our travel domain and the example above, a statement like “I would like to go to the Caribbean” would be enough. The system will propose further specified journeys together with the possibilities to modify these offers on a more detailed level. Upto a certain degree, the system can restrict the choice of modifications (cf. 3.1). Nevertheless, giving the customer certain means of control, the overall customization process brings up the necessity for consistency checking of her actions because many of these actions cannot be forbidden in advance. E.g., it has to be checked if the duration of the additionally booked surfing course does not extend the duration of the vacation or that the trip should be booked for three people, so enough places have to be vacant. The latter example shows another important aspect. This is the fact that the system should be designed to serve the experts as well as the inexperienced users. That means that the system has to support the business-to-business level, e.g., between the travel organizer and the travel agency, and the business-to-consumer level between the travel agency and the traveler.

4. The Approach to Customization

4.1 Interactive Customization

The stated requirements lead to the idea that instead of achieving complete automation of the customization process it makes more sense to give the flexible possibility to the customer to control the customization process according to her wishes. This requires a predefined interface to control the customization process, i.e., at certain stages in the process, the customer has to be able to tell the system how a product should be changed. Therefore, a central idea is to realize an interactive customization approach. Therefore, two major aspects have to be considered:

- The system has to find out which parts of the retrieved product have to be customized. A competent customer could determine the components of the product to be customized by her own. However, in a business-to-consumer scenario, customers cannot be assumed to be highly skilled. Hence, E-commerce systems should be able to guide and control the customer in her actions. It should be possible to define situations in which changing a part of a product makes sense. To describe such a situation, customization knowledge has to be used. E.g., if a customer wants to book an apartment house in the Caribbean, it must not be possible to book skiing lessons, too. Hence, the customization component should determine possible customization options, but the customer should decide which should be done.
10• If in the course of customization new parameters or sub-components have to be included into the product, they will have to be determined in co-operation with the customer. This allows the customer to specify her wishes more precisely only when required. However, the system should control the customer by validating the desired modifications.

4.2 Operator-based Customization

For our approach, we chose basically a way which modifies directly a retrieved “old” solution or case to adapt it to the new problem situation. We assume for our domain that there is only low interaction between the attributes of an alternative. The alternatives can be taken only completely, i.e., we will consider complete single transfer of products. An example is a customer looking for one complete journey. The journey has to be offered in a whole and should not be assembled from offers of several other tour organizers. Therefore, the customization process is divided into atomic units of changes. The customer has to decide which of these units should be processed. To model these atomic units, we have developed the concept of customization operators (see Fig. 3). The customization operators hold the whole customization knowledge and must be defined by the shop-builder. The first operator applied transforms the retrieved product into an intermediate product. The customer will apply sequentially more operators on the intermediate products until she reaches the target product (see Fig. 4). One could think of the possibility to allow the customer to specify these changes or transformations, respectively, in more detail by providing parameters after an operator is selected.

In most cases, an operator is applicable only in certain situations, i.e., for a certain product or intermediate product. The condition when an operator is applicable is part of the operator definition and must be defined by the shop-builder. The customer has to decide which of the applicable operators in a situation should be applied. By applying several operators, the target product will be successively customized according to the customer’s wish. The result of a customization is the changed target product.

This operator-based customization is an interactive variant of the formal adaptation model proposed by Bergmann & Wilke, 1998. Instead of a formal quality function that guides the operator application, the customer directly controls the application of the operator. Since the WEBSELL case representation is based on an

![Figure 3. Customization Operator.](image)

![Figure 4. Customization process as sequence of operators.](image)
object-oriented representation, our operator representation must be integrated with this object-oriented framework. We do not go into the details of this integration now.

4.3 Representation of Customization Operators

In this approach, the customization operators contain the complete customization knowledge. A customization operator at least consists of:

- A name which clearly identifies the customization process for the customer.
- A precondition that states for which products the customization is applicable. Basically, the precondition is defined over the attributes of the query product, and current intermediate products. Only when the precondition is fulfilled the operator can be applied.
- Parameters to specify the desired customization in more detail. The concrete parameter values have to be requested from the customer before the operator is actually applied.
- The action part specifies how the current intermediate product is changed. Actions can change slot values of the representation and add or delete new sub-products.

4.4 Example for Customization Operators

Now, we present a brief example for customization operators. It is an example for a customization operator which adapts the duration of the holiday to the query product.

- Name: Customizing the duration of the holiday according to the query.
- Precondition: The duration in the query is not equal to the duration in the intermediate product and the duration in the query product is not longer than the maximum duration possible for the hotel in the target product.
- Parameters: None.
- Action part: Set the duration in the intermediate product to the duration stated in the query.

It should be remarked that the example above is rather simplified just for the purpose of presentation. There are dependencies which also have to be considered as well as more actions to be performed. Generally spoken, dependencies between parts of a product have to be handled in the coding of the operator, e.g., in its precondition.

4.5 Processing of Customization Operators

To enable operator-based customization, mechanisms are required to apply operators and possibly to undo operators that have been already applied. For the current intermediate product, the precondition of all available operators must be checked. Those operators that are applicable must be identified and presented to the customer. If the customer selects an operator, she must be asked for the values of the operator parameters. If the values entered by the customer fulfill the parameter condition, then the action part of the operator is applied. This means that all actions are performed on the current intermediate product. This results in the next intermediate product, which is again presented to the customer. Then, she can either accept this product and terminate the customization or new customization operators can be offered to further modify the
product. Additionally, the customer should also be enabled to undo previously performed customization steps. Therefore, the customization component must keep track of the computed intermediate products and operators. If one operator is retracted, all operators that have been applied afterwards must be checked whether they are still applicable. In case they are not, it could be necessary to retract such operators as well.

5. Discussion

We now comment on the current state of realization of the described approach and discuss related work.

5.1 State of Realization

The described approach to customization is completely realized in a prototypical implementation (Meyfarth, 1997) and was developed on top of a commercial CBR system (CBRWorks). However, this implementation neither follows an agent architecture nor does it support a client-server architecture like needed for Internet-based electronic commerce. Therefore, a new detailed system design and specification of the customization system was developed as part of the WEBSELL project (see WEBSELL Project Deliverable D3.1&D4.1, 1998). A first implementation of the described architecture is expected by early summer 1999.

5.2 Related Work

Nowadays, there are many electronic commerce applications on the Web. Today, only a very small number of these applications uses CBR techniques for product retrieval, apart from that fact, there are even not many systems which offer possibilities to customize the products. In the following, a couple of examples are shown and compared to our approach.

PersonaLogic. PersonaLogic presents on their web pages a variety of product showcases. All applications have in common that the system asks the customer many questions about her needs. At every point in this acquisition process, the customer can ask for the products sorted out so far. Answering all questions only limits the number of products presented because the products get more and more specified by each question. Customization is not supported since it is not possible to modify the concrete products found.

Vobis. The Vobis Web page offers a PC configuration system. The customer starts with the selection of a pre-configured computer. Then, the single components can be exchanged which is the customization aspect. The customer first selects the kind of component, e.g., “keyboard” and then she can choose from a list of keyboards. This

---

8 http://www.personalogic.com/home/demo/demo.stm
9 http://www.vobis.de
configuration system can be compared to our approach very well, as selecting a component class is equal to selecting an operator. For certain components it is possible to enter parameters like for the number of components that should be built in. A major shortcoming of the Vobis system is that it does not do sufficient consistency checks or application control for its operators, respectively. Besides this, the Vobis application can be realized as a particular domain of our approach.

transtec. In the transtec\textsuperscript{10} solution for PC configuration, the customer first chooses the kind of system she wishes to configure, i.e., Workstation, Notebook, etc. There, she will find several pre-configured computers which can be further customized by exchanging or adding components (there is no real retrieval of initial products). Unlike the Vobis solution, it is not possible to enter any parameters for more detailed specifications. Another difference compared to our approach, the transtec customization system does not do consistency checking either.

Dell. The Dell\textsuperscript{11} computer configuration system also starts with a pre-configured machine. In contrast to the previously presented configuration systems, only certain components can be customized, others are fixed, e.g., the processor type. Compared to our approach, it is not possible to enter parameters. The Dell system does not do automatic consistency checking. Some of the components get labels which tell the customer that a consistency check or compatibility check respectively is needed.

To the best knowledge of the authors, there is currently no electronic commerce application on the Web that supports completely our form of customization. One or more of the aspects like parameters, checking for applicability of an adaptation step, retraction possibilities, handling of dependencies between customization steps, etc. are simply not considered. However, most of the described applications (for limited PC configuration) can be realized and significantly improved by the customization approach presented in this paper. Also, the implementation effort for these applications would be drastically reduced by using our generic architecture.

5.3 Optimization and Maintenance Aspects

In this work, we did not go into details concerning optimization criteria, e.g., to minimize the costs of vacation. It should be mentioned that this is a basic problem. One way to treat this problem leads via the similarity measures. Depending on the cases, there is the possibility to encode minimization criteria in the appropriate similarity function.

This paper also does not touch the wide area of administration and maintenance of the product case base. Especially the domain of PC configuration underlies frequent changes (e.g., prices, new hardware components, etc.). Therefore, the interested reader may refer to Heister & Wilke, 1998.

\textsuperscript{10}http://www.transtec.de
\textsuperscript{11}http://www.dell.com/client/home/index.htm
References


