

December 2002

Customer Requirements Elicitation in Intelligent Electronic Sales Support Systems

Tilmann Fingerle
empolis GmbH

Sascha Schmitt
Artificial Intelligence - Knowledge Based Systems Group, Dept. Of Computer Science, University of Kaiserslautern

Carsten Tautz
empolis GmbH

Follow this and additional works at: <http://aisel.aisnet.org/bled2002>

Recommended Citation

Fingerle, Tilmann; Schmitt, Sascha; and Tautz, Carsten, "Customer Requirements Elicitation in Intelligent Electronic Sales Support Systems" (2002). *BLED 2002 Proceedings*. 10.
<http://aisel.aisnet.org/bled2002/10>

This material is brought to you by the BLED Proceedings at AIS Electronic Library (AISeL). It has been accepted for inclusion in BLED 2002 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Customer Requirements Elicitation in Intelligent Electronic Sales Support Systems

Sascha Schmitt

Artificial Intelligence – Knowledge Based Systems Group
Dept. of Computer Science, University of Kaiserslautern
Germany
SSchmitt@informatik.uni-kl.de

Tilmann Fingerle

empolis GmbH
Germany
Tilmann.Fingerle@empolis.com

Carsten Tautz

empolis GmbH
Germany
Carsten.Tautz@empolis.com

Abstract

The convenience of shopping on-line via the Internet has become a widely accepted view. An important aspect of automated electronic sales systems is communication with the customer. Nevertheless, customers encounter quite frequently user interfaces that are hard to use – either because they have to answer annoying or irrelevant questions or they are faced with technical jargon of manufacturers they are not able to understand. Once they have managed to articulate their needs, the system may return no products at all or a very long list of products satisfying the customers' needs. Recently, knowledge-based systems, and in particular case-based reasoning (CBR) systems, have been recognised to alleviate these problems. They provide means for intelligent dialogs and search mechanisms, which are also able to offer alternative products (avoiding the “sorry, no matching products found”

syndrome). In this paper, we suggest a framework enabling the characterisation and comparison of various strategies for eliciting customer needs. The framework is used to briefly describe and characterise advanced commercial, state-of-the-practise applications, which are available publicly. Furthermore, we present our tool, leveraging CBR technology, behind these applications.

1. Introduction

The ultimate goal of the electronic sales system is to find products for the customers such that their demands are fulfilled and they are willing to buy these products. However, many electronic sales sites do not provide responsive on-line sales support and the interactivity available on the Web is not explored. This situation becomes extremely problematic if the e-shop deals with complex products like computers, digital cameras, etc. On the one hand, the vendors have recognised intelligent customer support, i.e., elicitation of product wishes, navigation through product spaces, and product selection, as an important feature; but, on the other hand, the quality of the retrieval results, i.e. the products recommended by the system, is miserable [6]. Hagen's survey on search on the Internet among 50 e-shops [6] states that 68% of those shop providers consider search for extremely important on their Web sites and for another 22%, search is very important. Furthermore, the report also found out that 56% of the search engines do not find relevant information at all and 94% of them cannot handle synonyms or typing errors.

In our opinion, the problem starts even earlier with poor interaction facilities of the systems. Before anything can be searched the e-sales system has to carry on an automated dialog, which can be more or less sophisticated to elicit the customer's needs. Generally, for the specification of demands, a customer would use vocabulary of a much higher level of abstraction than this is reflected, e.g., by predefined input forms that can be found in most of the electronic shops today [11]. This situation occurs because of the fact that the customer would rather like to state a functionality to be fulfilled by the product than give a detailed description of the product to be found [13]. Typically, the vendors, distributors, or manufacturers possess the knowledge about their products. The customers possess the knowledge about their individual needs. Neither the customers nor the vendors usually share the others' knowledge completely; they may even use different vocabularies. This knowledge deficit is the kind of situation that can be called a "knowledge gap" [11]. During a sales transaction, the "knowledge gap" can be observed especially when it comes to the acquisition of customers' needs and finding adequate products for these demands. In bricks-and-mortar companies, the human sales agent who makes use of his strategic knowledge to mediate between the potential buyer and the vendor bridges this knowledge gap. This mediation requires communication during which the knowledge and information are transferred from the vendor to the

customer and vice versa. Here, the communication processes are very complex and they take place with many interactions between vendor respectively sales system and customer. In an on-line sales process, a virtual sales agent must realise this communication process, which is the preparation of the ensuing search process.

Both, information and knowledge are based on experience. This experience has to be managed and made accessible somehow. In this context, a new challenge for Case-Based Reasoning (CBR) technology was found with the aim to improve Electronic Commerce (EC) (e.g., [2], [3], [12], [14], [15]). CBR assumes products to be described by a set of attribute-value pairs (the product describing properties), e.g., stored as records in a database. The advantage of CBR over simple parametric search is its ability to consider knowledge (in the form of similarity measures) during retrieval, which enables to assess the suitability of a product with respect to a certain customer need. Thereby, it reaches much higher retrieval quality than competing approaches but at the price of investing into knowledge modelling. CBR can be used effectively whenever a domain model can be built for the goods and services to be offered by the system. Such a knowledge model can cover quite a large spectrum as the example Otto (see Section 4.2) shows.

In the remainder of this article, we first introduce a framework in Section 2 that lists ten categories, which in our opinion are most important to describe a dialog component in an electronic sales system. Most of the features mentioned are already state-of-the-practise, others are still state-of-the-art and currently under investigation. Section 3 describes the commercial tool empolis orange with respect to intelligent customer support and especially its facilities for dialogs. empolis orange has proven its strengths in numerous EC applications. Section 4 presents a small selection of commercial and demo applications, which are all available on-line. The focus lies on the customer interaction within those systems. Finally, Section 5 draws some conclusions.

2. Characteristics of Customer Demand Elicitation

All of the sites that sell different products have in common that they must find out which one of their products the customer is looking for respectively fulfills best the customer's needs. In this section, we present a framework for the characterisation of sales sites with respect to the dialog they realise.

2.1 Categories

We distinguish ten categories, which we consider to be the most important aspects of a dialog component. We describe their meaning in detail in the ensuing

subsections. Table 1 summarises the categories and also characterises the application examples we present in Section 4.

2.1.1 Domain

The *Domain* lists some general data about the sales site. The sales application can be either from the *sector* of Business-to-Consumer (B2C) or Business-to-Business (B2B). Furthermore, it will be embedded in an *environmental* context, so that the system is representing an e-shop, e-mall, e-marketplace, portal, or auction. The later described features and technologies strongly depend on the representation of the *products*. Especially for retrieval it is important if the products are described by text only or in a structured way, e.g., by attribute-value pairs. Also the number of different products offered is of special interest, e.g., for the retrieval method that can be used.

2.1.2 Dialog Representation

The *Dialog Representation* addresses the interface to the customer, which is abstracting from the underlying technology issues. We distinguish four basic ways how the customer will communicate with a sales system:

Free text. Customers are offered a text input box to be able to type in a description in natural language about their product needs. (see, e.g., [16], [21].) Of course, one can also think of the possibility to provide an interface for spoken language.

Form. This is probably the most common way to be found in e-sales on the Internet. The system displays a questionnaire with standard elements like selection lists, radio buttons, etc. (see, e.g., [20])

Questioning. The system generates single questions or groups of questions, which the customer is asked to answer sequentially. This is possibly the most flexible way of interaction as the system can influence many factors during the proceeding of the dialog. The highest degree of flexibility can be reached by systems that select their questions dynamically with respect to the current sales situation. These systems use some question selection strategy to adapt the questioning process depending on the information already given by the customer and also needed from the customer. (see, e.g., [16])

Listing. This is the simplest way of communication, which will be mainly found in pure catalog-based applications and is considered here just for the reason of completeness. Often, in such shops there is no or only very rudimentary search functionality provided (e.g., in [18]).

It is possible to combine those four techniques in parallel in one e-sales shop (e.g., [16]). So the customer can choose their preferred way of entering product needs. However, it is obvious that not every combination does make sense. A more important aspect is to combine these techniques sequentially. At a *first stage* of the interaction process, the system might identify many properties of the product the

customers are looking for from their natural language description as free text. To limit the number of candidate products, the system could ask more specific questions using a form or a questioning process at a *second stage*.

2.1.3 Dialog Behaviour

The *Dialog Behaviour* captures some aspects of the dialog proceeding. A *goal-directed* dialog tries to avoid redundant and irrelevant questions (e.g., asking for properties that the candidate products in the database do not have). Furthermore, depending on the customer's product knowledge and the degree of annoyance (cf. "nuisance level" in [10]), posing a certain question induces a *utility* for the system respectively for the customer. A dialog system has to balance between the information that can be gained by asking a question and the possible effects if the customer cannot answer the question, does not want to answer on it, or does not care about an answer. Such effects could increase the risk of the customer leaving the shop.

2.1.4 Structuring the Dialog Data

The behaviour of the dialog depends on the abilities of the sales agent to *Structure the Dialog Data*, i.e., information gained from the customer and also data of the domain, in an adequate way. We can distinguish *static* structures like simple lists or trees that are precompiled and *dynamic* ones based on entropy or similarity-influenced, which are able to react directly on changes in the product database [11]. Especially the latter ones are very interesting in our context. Entropy-based means that the next question will be determined by the maximum information gain. The maximum information gain is computed using the structured product descriptions. Those attributes (product properties) that allow to distinguish (discriminate) between the product descriptions most clearly are selected. The selection process also considers the attribute values already specified. Product specifications that hardly match the customer's (partial) query are ignored by the question selection process. A similarity-influenced method can be used in systems only, where the domain model contains the knowledge about product similarities (CBR systems). This approach will select the questions that lead to products with higher similarities to the customer's product wish, i.e. their specification (query).

An interesting aspect that comes into play is the possibility of such systems to use information for *learning* and therefore improve their behaviour either directly during the session with one customer or delayed by observing interactions with numerous customers.

2.1.5 Search

The first direct major outcome of the dialog with the customer has to be a query to start *searching* for adequate products. The *methods* for finding such products can

range from simple full text search to intelligent search technologies (e.g., CBR). The dialog with the customer can be processed more flexibly if the interaction process interweaves with the search process. Then, e.g., questions can be adapted to a current state of the product database.

2.1.6 Information Used

Additional *Information* might be used to control the dialog and to also influence the product search. The sales system can store *profiles* about its customers. Such profiles can be elicited explicitly in advance and therefore the customer will be known to the system because they have to log in before using the system (see for example banking applications). Profiles can also be created implicitly by observing the customer's behaviour (e.g., click-streams on the Web pages) and their answers to the questions in the dialog with the aim to associate them to a specific class of users. Further, the dialog can allow the customers to specify certain preferences, the so-called *weights*. Also, here the system can ask explicitly how important the different properties of a product are; or, the weights are implicitly contained in the domain model. The use of *filters* gives the customer the possibility to formulate constraints (e.g., the travel can only be started after a certain date) if they are asked explicitly. It can also happen that the customer will evoke a kind of filtering effect by giving contradicting answers during the dialog. This is due to the dialog itself or to the search service to handle the problem adequately.

2.1.7 Product Presentation

A few *comfortable* features can also positively influence the *product presentation* at the end of the dialog. It should be possible for the customer to easily *refine* the query, i.e. to re-enter the dialog and specify properties again. Further, information about the customer can be used to *adapt* the product presentation, e.g. the system found out that it does not deal with an expert customer and so it does not directly display very detailed technical information about the found products. A feature of special benefit could be the *explanation* why the sales system is suggesting the products, that is to say, on which specifications from the customer the product selection depended.

2.1.8 Product Changes

A number of e-sales sites do not only deal with fixed, unchangeable products (say books for example) but also with products that can be further *adapted* or *configured* [14] to better fulfill the customers' demands. On the one hand, the adaptation and configuration processes can be executed by the system automatically, transparently for the customer. On the other hand, the system could suggest product change actions and let the customer take control by asking explicitly which action to perform. This introduces another dialog.

			Application		
			Neckermann	Otto	CarSmart24
Domain	sector	B2C	x	x	x
		B2B			
	environment	e-shop	x	x	x
		e-mail			
		e-marketplace			
		portal			
	products	auction			
textual descriptions			x	x	
structured with attributes		x		x	
		number	> 100.000	> 1.000.000	> 1000
Dialog Representation	1st stage	free text		x	x
		form			x
		questioning	x		x
	2nd stage	listing			
		free text			
		form	x		
		questioning		x	x
		listing			
Dialog Behavior	goal directed	redundancy avoided		x	x
		irrelevance avoided		x	x
	utility	user knowledge	x		
		user annoyance		x	x
Dialog Data Structuring	static	list	x		
		tree			
	dynamic	entropy-based		x	x
		similarity-based			
	learning	direct			x
		delayed			
Search	method	full text			
		intelligent	x		
interweaving with dialog			x	x	
Information Used	profiles	implicit			
		explicit			x
	weights	implicit			x
		explicit	x		x
	filters	implicit	x		
		explicit		x	
Product Presentation	comfort	query refining possible		x	x
		adapted			
		explanation			x
Product Changes	adaptation	automatic			x
		user-controlled			
	configuration	automatic			
		user-controlled			
User Encouragement	level	abstraction from attributes	x		
	help	specific help accessible		x	
		multimedia support			
	active support	recommendations			x
		error tolerance		x	
Implementation		knowledge acquisition	Prod., Cust.	Prod.	Prod., Cust.
		maintenance	Prod. Cat.	Catalogs	Cars
		technology	CBR	CBR	CBR

Table 1. Example applications described by our framework.

2.1.9 User Encouragement

What we call *user encouragement* summarises a couple of features to assist and encourage a customer to carry on the dialog with an e-sales agent. Customers will rather often prefer to specify their product needs on a more abstract *level* than the underlying domain model of the sales shop reflects. Especially unknowledgeable customers with respect to the domain (e.g., technical domains) will be happy if specific *help* functionalities are available, maybe in a textual description only or even supported by multimedia elements. The sales system can *actively support* the customers' buying process by providing product recommendations (based on collaborative filtering methods). An error detection (typos) and automatic correction suggestion can also help to improve the search process.

2.1.10 Implementation

The last category regards *implementation* issues to build a sales agent fulfilling the respective requirements. Different kinds of *knowledge* have to be considered and acquired from a human expert, e.g., about the products, customers, sales strategies, etc. The developing questions are what effort has to be spent to maintain the system (what will the product changes look like, how often do they occur, etc.) and what technology would be used most suitably.

2.2 Thoughts about Dialog Quality

The framework is neutral in the sense that it makes no judgement about the quality of the applications; it simply summarises the features and properties of the dialog service. Quality aspects would enter the scenario in the following way.

Each type of application could have ideal service from the customers' points of view. "Ideal" in our sense means to offer what is necessary, but not too much. However, there are costs and difficulties to be considered from the seller's point of view. Ideal service has to be provided at admissible costs, i.e. every desirable functionality should be available, but nothing is overdone.

3. empolis orange: a CBR Tool for Intelligent Sales Support

Empirical studies show that different customers prefer different user interfaces. For instance, Frakes & Pole show that this is true for searching reusable software component libraries [5]. As this type of search is not fundamentally different from

searching product catalogs, it can be assumed that there is no “one size fits all” solution for electronic sales agents.

If an electronic sales agent is not to be developed from scratch but by using a tool (to reduce development and maintenance effort), a toolbox is needed from which the sales agent can be constructed. This toolbox must provide choices according to the framework presented in the previous section. *empolis orange*, a commercial CBR tool developed by *empolis GmbH* [17] as the major outcome of the ESPRIT project WEBSSELL [3], has been designed with this principle in mind. In the following, the most important features and components of *empolis orange* for customer demand elicitation will be described, whereas the next section will present some exemplary solutions developed with this tool.

It seems important to state that *empolis orange* uses CBR as its central component to realise fuzzy search. But *empolis orange* is more. Additional innovative and advanced functionalities are built around the CBR kernel. Its goal is to take CBR as the powerful technology it is and to lift restrictions that come along with CBR by adding tools to analyse text, compute dialogs or execute strict rules.

empolis orange, the Open REtrieval ENGINe, is able to convert automatically textual product descriptions and free-text query entries into a structured representation, i.e., attribute-value pairs, based on a knowledge model. This means that a minimum of maintenance effort is needed. If products of a new catalog are to be offered, all that is needed is to run this automatic conversion. The knowledge model itself does not need to be redefined (except if new product categories are added). The model does not only define the structural descriptions of products and how to transform textual descriptions into this structured form, but also the behaviour of the user dialog and the retrieval process.

The customer interaction can be controlled in various ways. First, a free-text query input field can be processed using the *orange:TexMiner*. Any text entered in this field will be transformed into attribute values as defined by the knowledge model. This way, *empolis orange* fulfills an important role of a real sales agent: To translate between the customers’ language/vocabulary and the manufacturers’ or experts’ language. An optional spell checker can be used to tolerate typos. In addition, the customer’s profile (if available) can be used to pre-select some of the attribute values. Moreover, *orange:Profiler* can be used to initialise a query based on known customer profiles.

Secondly, a dialog can be initiated. In this dialog, *empolis orange* will pose only questions for attributes for which no values have been specified so far (by the free-text query input). Two major parameters control the dialog: (a) the number of questions to be displayed at one time and (b) the selection strategy (either sequentially or entropy-based; see subsection 2.1.4). For each question, possible answers (i.e., attribute values) are presented to the customer. These attribute values are automatically constrained. For example, imagine that an on-line store offers among other products clothes and washing machines. If the customer has already specified a particular product group (e.g., a pair of pants), the system will offer only

manufacturers of apparel (more specifically, manufacturers of pants), but for instance no manufacturers of washing machines.

The specification of customer needs is done alternately with a retrieval step until the customer is satisfied with the result. Retrieval is done using similarity functions. Retrieval behaviour can be influenced using filters and weights (cf. subsection 2.1.6). Both can be defined in three different ways: (a) statically as part of the knowledge model, (b) dynamically by the system based on the customer needs (this behaviour is specified in the knowledge model using rules), and (c) dynamically by the customers themselves. Available product descriptions can be adapted using rules. For example, in a used car shop, a particular car model may not be available in the color the customer demands. However, it is possible to re-paint the car. This will satisfy the customer's demand, but at the same time increase the price of the car.

Finally, the result is displayed. The display can be enhanced by an explanation, telling why a particular product is part of the result. This can become important to the customers because the similarity-based retrieval may result in products not satisfying their demands to 100%. In this way, *empolis orange* acts like a real sales agent in a store that would always offer an alternative even if none of the available products satisfied the customer demands to 100%.

A special authoring environment, called *orange:Creator*, allows the definition of the knowledge model including dialog and retrieval behaviour. Fig. 1 shows a screenshot where the dialog behaviour is defined for an example of a used car shop.

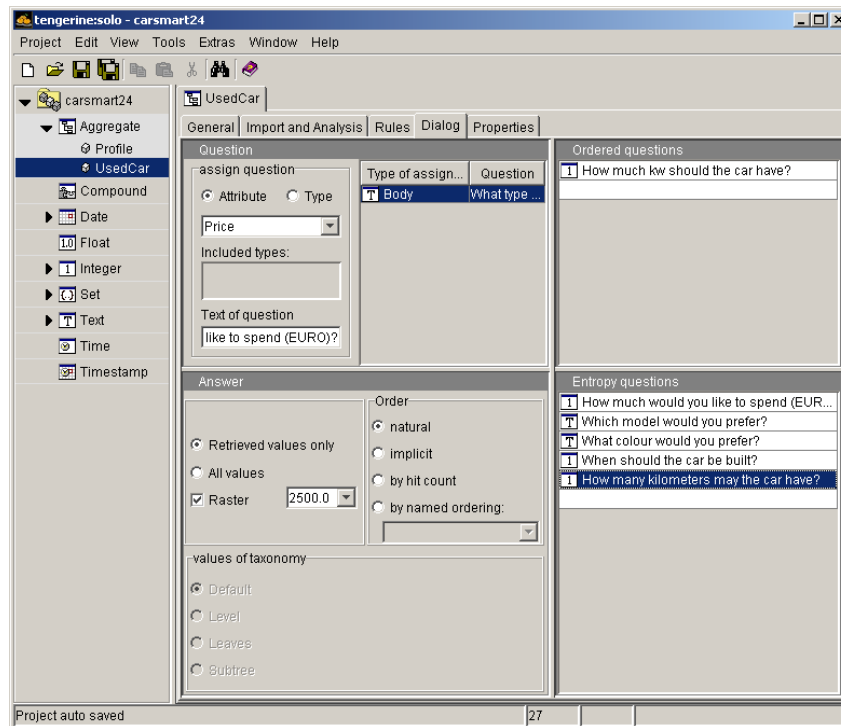


Fig. 1. *Defining the Dialog Behaviour (orange:creator screenshot).*

4. Application Examples

This section describes some of the sales applications developed with empolis orange.

4.1 Neckermann Product Advisor

Neckermann Versand AG is one of the biggest mail order companies in Germany with a highly frequented on-line shopping portal [19]. To support their customers buying complex products, e.g., camcorders, personal computers, or washing machines, Neckermann decided to implement an easy to use CBR-based questionnaire. The idea is to provide on-line customers with a shopping situation very similar to the real world: the customers are being asked relevant questions for a specific product and then provide the system (like a virtual sales person) with answers that describe their individual needs best.

The solution requires knowledge about:

1. the customers' state of knowledge,
2. the relevant questions,
3. actions or values that result from a given answer.

The following example illustrates all steps that are necessary to elicit the customer demands for finding a suitable washing machine (see Fig. 2).

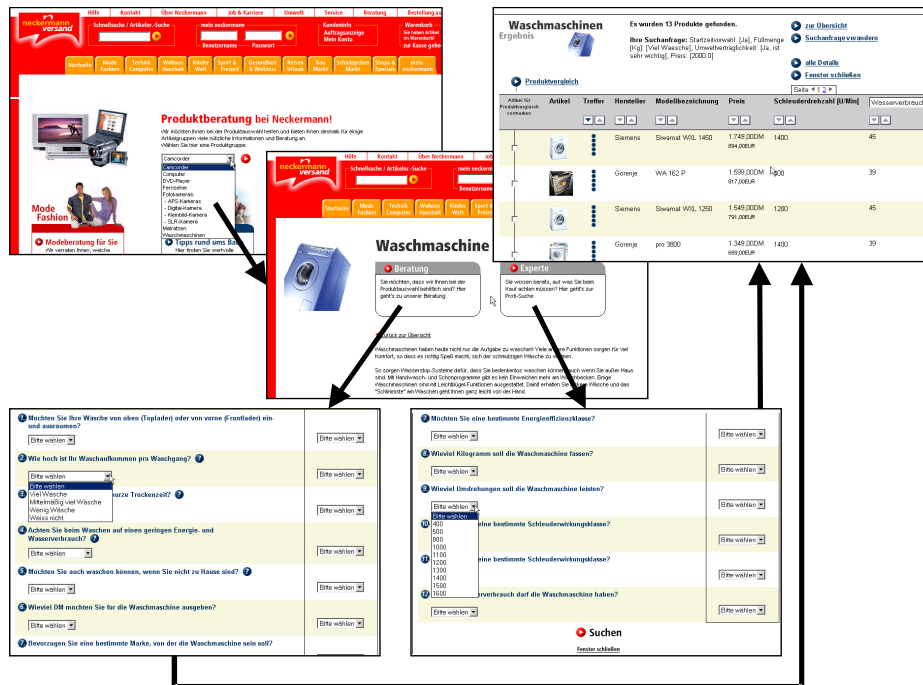


Fig. 2. The Dialog of the Neckermann Product Advisor.

The elicitation of the customer needs starts with the product category, e.g., camcorders, cameras, refrigerators, washing machines, or even mattresses (upper left of Fig. 2). Neckermann recognised that there are two main types of customers: experienced customers with expert knowledge and less experienced customers (casual customers). Therefore, as a service to the customers, Neckermann decided to offer two questionnaires: Casual and expert (middle of Fig. 2). After having selected the level of knowledge, the customers are being asked a couple of questions to get to know their specific needs using a questionnaire. The lower left screen of Fig. 2 shows a casual customer giving an answer to the question of whether she usually has lots of clothes, an average amount of clothes or only few clothes to wash. The lower right part of Fig. 2 shows the alternate questionnaire for expert customers. In contrast to the casual questionnaire, all questions in the expert section are directly related to a specific feature of a washing machine. In this example, the customer is asked, how fast the drum is supposed to be able to spin.

Both kinds of customers have the possibility of expressing the importance of the criteria such as price and brand (right column in the questionnaires). The customers are able to tell to the application, which one of the asked questions are of higher, regular, or lesser importance. This corresponds to talking to a real sales agent who will consider the various degrees of importance for advice. Finally, the results are displayed (upper right of Fig. 2).

The Neckermann solution uses weights, rules, filters, and similarity to implement intelligent retrieval.

4.1.1 Weights

A product, e.g., a washing machine, is described by a set of abstract attributes such as price, manufacturer, water consumption, additional features, etc. Some of these attributes are assumed to be of higher importance for customers than others. That means that price and manufacturer usually are more important than energy or water consumption. However, different people set various priorities. Therefore, Neckermann offers the opportunity to individualise the importance. Three levels of importance are available: high (doubles the default importance), normal (default), low (bisects the predefined importance).

The various degrees of importance are implemented as weights. Default weights are defined in the underlying knowledge model. However, the customer can change them as explained above.

4.1.2 Rules and Filters

To improve the quality of retrieval, rules are used. More specifically, they are used to:

- **Translate between customer and manufacturer language.** Often, the customer formulates demands in terms on how they want to apply/use the product. Normally, these kinds of demands do not correspond to any technical specifications of the manufacturer. For instance, in the casual mode, the customer has to answer some fuzzy questions such as "How frequently do you use your washing machine per week?". The answers to this question (more than 10 times, 5 to 10 times, less than 5 times) are not directly connected to an attribute of a washing machine. Therefore, a rule is set up that derives specific values from a fuzzy answer. In the case of Neckermann, the answer "More than 10 times" results in two values: (a) energy consumption less than 0.96 kWh per wash and (b) water consumption less than 40 l per wash. These values are then used to search for a suitable washing machine.
- **Complete queries.** In reality, the sales persons know (or at least should know) the customer and their needs. For example, if the customer wants to have a washing machine with a delay timer, the sales person should know that the

customer needs a water control system, too (to be able to wash when not being at home). Neckermann implemented the knowledge about their customers using rules, in this case with a rule that adds the additional value "water control system" to the set of features the washing machine should have.

- **Set filters.** Filters are used to limit the offered results regarding a given criterion. For instance, if the customer does not want to spend more than a certain amount of money, Neckermann decided to prevent the system from offering a product that is two or three times more expensive.

4.1.3 Similarity Definition

Similarities are necessary to recommend alternatives to the customer. A typical situation is that the customer wants to have a specific brand. In case the shop does not sell products of this brand but has comparable products available, the sales person would offer these products as an alternative. empolis orange rates the relevance of each product in regarding the elicited customer needs. The relevance is displayed as part of the result (see number of dots in the third column of the result screen in Fig. 2).

4.2 Otto

Otto is the world's largest mail order business group. One of their strategies is to utilise new technological developments early and to invest in successful, promising new media. The Otto business group has two on-line shops utilising the advanced orange technology: Shopping 24 (a virtual mall consisting of more than 30 shops; [21]) and Otto [20] itself.

Fig. 3 shows the dialog of Otto. The dialog starts with a free-text entry. This text is analysed. In the particular example shown (in German), the customer is looking for a blue pair of pants for his girl friend, but has mistyped "pants". Otto decided to include a spell checker which offers alternative terms to the customer. In this case, the correct spelling is "Hose". In the last dialog step shown, the electronic sales agent repeats the query in its own words ("pants in the color blue for girl friend"). This provides the opportunity to refine the query (in the example, the system asks for the manufacturer), and displays articles at the bottom of the screen using their textual descriptions (not shown in Fig. 3). Note that the system recognises that "girl friend" means "women's apparel". The generated questions are entropy-based. This means that the customer will not be asked to answer questions for which he has already provided answers. Moreover irrelevancy is avoided. In the example shown, only manufacturers of women pants (which are sold by Otto) are presented to the customer.

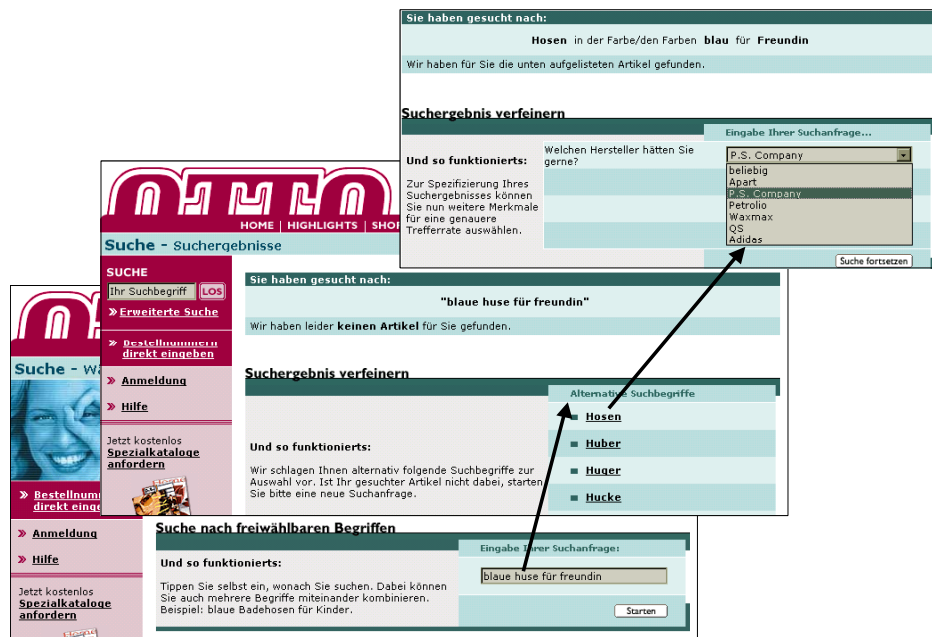


Fig. 3. Exemplary Dialog of the Otto Application.

4.3 CarSmart 24

CarSmart 24 [16] has been developed as an on-line application to demonstrate the features of empolis orange. The application supports several customer interaction schemes, one of which shall be discussed in detail in this paper. The dialog considers weights (customers' importance ratings). It is shown in Fig. 4 below.

The dialog begins with asking for the most important criterion. In the second step, the customer is asked for a desired value. In the example, these are the make and the model. The intelligent interface restricts the models based on the make selected by the customer (e.g., the system will not offer any Mercedes models if the customer has already picked BMW as the make). In step three, the system asks the customer to rate the importance of some other features (shown at the bottom of the figure). It does not ask for the make/model since this criterion has already been selected to be the most important one. In the example, the customer selects the year and mileage of the used car to be (very) important. This is the reason why the system asks in step 4 for the values of these criteria. Finally the system displays the retrieval results (upper right screen in Fig. 4). Cars found are displayed in textual form based on their structured description. The explanation component marks the results. For example, "BMW 315" is displayed in green because it matches perfectly the customer needs. The "185,000 km" is displayed in blue because it deviates from the specified "25,000 km". A more elaborate explanation is given if the customer moves the mouse pointer over the highlighted text.

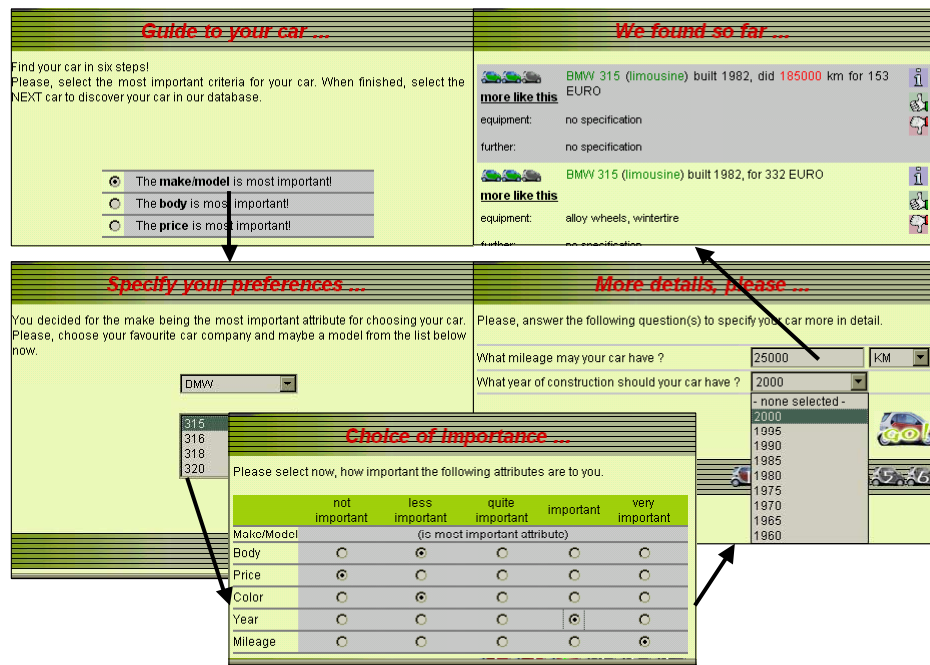


Fig. 4. Exemplary Dialog in CarSmart24.

At this point, the customers have several options:

- They can further refine the query (not shown in the figure).
- They can search for similar cars (“more like this”). In this case, the description of the selected car is used to start a new query.
- They can give feedback to the system affecting future retrieval, that is, CarSmart24 supports profiling. Clicking on “thumb up” will store the selected car as the default query. Clicking on “thumb down” will place the attribute values of the selected car in a negative list. This negative list is used as a filter for future queries. The filter settings can also be edited manually (see Fig. 5).

CarSmart24 is able to suggest an initial query even if the customer visits the Web site for the first time. This is done by asking each customer for some personal information and preferences like age, sex, preferred car style (comfortable, compact, luxurious, sportive), car usage (private only, business only, both), etc. If a new customer registers, the most similar profile is retrieved using the personal information and preferences. Then, the default query associated with the most similar profile is used.

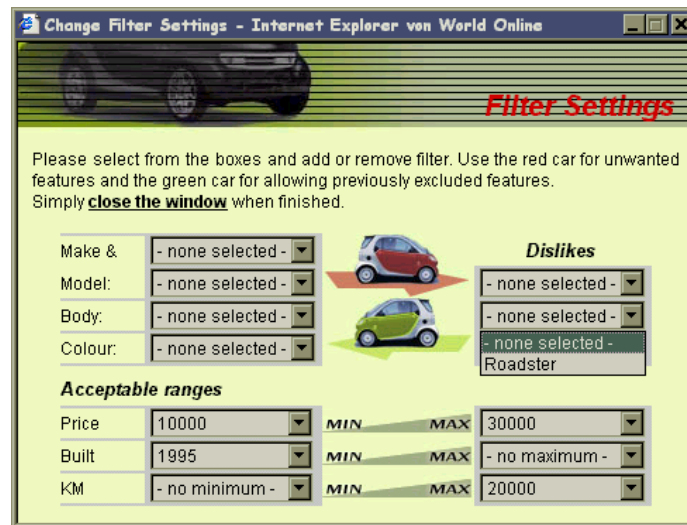


Fig. 5. Manual Setting of Filters.

5. Conclusions

Starting from a framework allowing the characterisation and comparison of interaction strategies for eliciting customer requirements, we described different factors to be considered for systems carrying on automated sales dialogs with customers. Communication with the customer is the important first step for successful product recommendation as it initiates the search process. Moreover, the processes of interaction and search are closely linked together and interweaving. We introduced our tool *empolis orange*, which allows building intelligent sales support systems based on CBR technology, providing much of the functionality encouraged by our framework. The potentials and powers of *empolis orange* are illustrated focussing on customer demand elicitation by two commercial state-of-the-practice and one demo applications as examples.

Looking at the examples, the framework also identified improvement potentials. While the mostly used strategies for question selection in the dialog with the customer have their origin in entropy-based methods [4] considering special aspects of EC and so often concentrating only on dialog length, especially CBR systems can make use of their system-inherent knowledge of similarity to also concentrate on dialog quality [9]. That means each step in the questioning process brings the system closer to the recommendation of a product that best suits the customer's wish. In certain domains, such systems can even show better performance than their only entropy-based counterparts [8]. Another important factor to be considered by the dialog systems should be the cost of posing a question [1]. The cost is determined by the degree of comprehensibility and the probability that the customer

wants to answer the question. Unfortunately, such systems are only state-of-the-art and not state-of-the-practise at present.

References

- [1] D. Annacker, S. Spiekermann, M. Strobel. (2001). e-Privacy: Evaluating a New Search Cost in Online Environments. In: B. O'Keefe, C. Loebbecke, J. Gricar, A. Pucihar, G. Lenart (Eds.): e-Everything: e-Commerce, e-Government, e-Household, e-Democracy. Proc. of the 14th Bled Electronic Commerce Conference, Bled, Slovenia. Vol. 1: Research.
- [2] R. Bergmann, S. Breen, M. Göker, M. Manago, S. Wess. (1999). Developing Industrial Case-Based Reasoning Applications. The INRECA-Methodology. LNAI 1612, Springer.
- [3] P. Cunningham, R. Bergmann, S. Schmitt, R. Traphöner, S. Breen, B. Smyth. (2001). WEBSSELL: Intelligent Sales Assistants for the World Wide Web. In: R. Weber, C. Gresse von Wangenheim (Eds.): Proc. of the Workshop Program at the 4th International Conference on Case-Based Reasoning, ICCBR-2001, Vancouver, Canada. Workshop 3: Case-Based Reasoning in Electronic Commerce.
- [4] M. Doyle, P. Cunningham. (2000). A Dynamic Approach to Reducing Dialog in One-Line Decision Guides. In: E. Blanzieri, L. Protionale (Eds.): Advances in Case-Based Reasoning. Proc. of the 5th European Workshop on Case-Based Reasoning, EWCBR 2000, Trento, Italy. LNAI 1898, Springer.
- [5] W B. Frakes, T. P. Pole. (1994). An empirical study of representation methods for reusable software components. IEEE Transactions on Software Engineering, 20(8):617–630, August 1994.
- [6] P.R. Hagen. (2000). Must search stink? The Forrester Report, June 2000.
- [7] P.R. Hagen. (1999). Guided Search For eCommerce. The Forrester Report, January 1999.
- [8] A. Kohlmaier, S. Schmitt, R. Bergmann. (2001). Evaluation of a Similarity-based Approach to Customer-adaptive Electronic Sales Dialogs. In: S. Weibelzahl, D. Chin, G. Weber (Eds.): Empirical Evaluation of Adaptive Systems. Proc. of the workshop held at the 8th International Conference on User Modeling, Sonthofen, Germany.
- [9] A. Kohlmaier, S. Schmitt, R. Bergmann. (2001). A Similarity-based Approach to Selection in User-Adaptive Sales Dialogs. 2001. In: D.W. Aha, I. Watson (Eds.): Case-Based Reasoning Research and Development. Proc. of the 4th International Conference on Case-Based Reasoning, ICCBR-2001, Vancouver, Canada. LNAI 2080, Springer.

- [10] B. Raskutti, I. Zuckerman. (1997). Generating Queries and Replies during Information seeking Interactions. In: *International Journal of Human Computer Studies*, 47(6).
- [11] S. Schmitt, R. Bergmann. (2001). A Formal Approach to Dialogs with Online Customers. In: B. O'Keefe, C. Loebbecke, J. Gricar, A. Pucihar, G. Lenart (Eds.): *e-Everything: e-Commerce, e-Government, e-Household, e-Democracy. Proc. of the 14th Bled Electronic Commerce Conference*, Bled, Slovenia. Vol. 1: Research.
- [12] A. Stahl, R. Bergmann, S. Schmitt. (2000). A Customization Approach for Structured Products in Electronic Shops. In: S. Klein, B. O'Keefe, J. Gricar, M. Podlogar (Eds). *Electronic Commerce - The End of the Beginning. Proc. of the 13th Bled Electronic Commerce Conference*. Vol. 1: Research.
- [13] I. Vollrath. (1998). Reuse of Complex Electronic Designs: Requirements Analysis for a CBR Application. In: B. Smyth, P. Cunningham (Eds.): *Advances in Case-Based Reasoning, Proc. of the 4th European Workshop on Case-Based Reasoning, EWCBR-98*, Dublin, Ireland. LNAI 1488, Springer.
- [14] W. Wilke. (1998). *Knowledge Management for Intelligent Sales Support in Electronic Commerce*. Dissertation, University of Kaiserslautern, Germany. Diski Verlag.
- [15] W. Wilke, M. Lenz, S. Wess. (1998). Intelligent Sales Support with CBR. In: M. Lenz, B. Bartsch-Spörl, H.D. Burkhard, S. Wess. (1998). *Case-Based Reasoning Technology. From Foundations to Applications*. Springer.
- [16] CarSmart24 demo: <http://live.tecinno.de/projects/carsmart24com/>
- [17] empolis Knowledge Management GmbH, Germany.
<http://www.km.empolis.com/>
- [18] www.hugoboss.com
- [19] www.neckermann.de
- [20] www.otto.de
- [21] www.shopping24.de