e-Everything: e-Commerce, e-Government, e-Household, e-Democracy

14th Bled Electronic Commerce Conference

Bled, Slovenia, June 25 - 26, 2001

A Formal Approach to Dialogs with Online Customers

Sascha Schmitt, Ralph Bergmann

Artificial Intelligence – Knowledge-based Systems Group
Dept. of Computer Science, University of Kaiserslautern
67653 Kaiserslautern, Germany
SSchmitt@informatik.uni-kl.de, Bergmann@informatik.uni-kl.de

Abstract

This article describes a formal approach for dynamically directed sales dialogs that avoids certain disadvantages found in online sales systems today. These disadvantages stem from an information deficit that we call a knowledge gap between the customer and the vendor. The suggested approach generates situation-dependently appropriate actions, e.g., asking the right questions, flexibly reacting on customer's decisions and also providing her with information needed. The dialog approach integrates the communication process in an accompanying search process. Furthermore, the structure of the product database is considered for question selection to focus on dialogs of reduced length, i.e. with respect to the number of questions to be asked.

1. Introduction

Product search in electronic sales environments, especially in online sites on the Internet, can become a hard business for potential buyers if adequate support from the system is missing. Customers are confronted with the different types of sites like portals, e-malls, e-shops that either deal with specific kinds or broad ranges of products. Customers entering one of these sites are often asked an also wide range of questions about the entire spectrum of available information on the electronic site. Inexperienced customers or those who do not exactly know what they want will run into difficulties. The consequence is that they leave the site frustrated because of different factors, e.g., there were

- too many questions,

- too difficult questions,
- redundant questions,
- wrong or no search results, etc.

A report on guided search on the Internet [6] found out that there are three major reasons why potential buyers have unsatisfying experiences:

1. Vendors build only point solutions,
2. E-Commerce sites implement search improperly, and
3. Shoppers struggle with poor search interfaces.

In this article, we will concentrate on the third point, which deals with the communication aspect between customer and sales system. One of the most important steps in an electronic sales process is the acquisition of customers' demands. This acquisition process should adapt to the customer and not require from the customer to adapt to the electronic sales system. Unfortunately, it seems to be state-of-the-art in too many e-shops where the customer is presented a form listing a number of product properties which she is supposed to fill in depending on her demands. This can be a time consuming, boring, and/or an even unsolvable task for the customer, especially if she is not familiar with the complexity of the products (e.g., personal computers). Of course, it would be worst case for both, vendors and potential buyers, if the system returned wrong results or no results after the acquisition phase (e.g., inconsistencies could also be a reason for no results).

We describe in this article a formal approach of a dynamic goal-directed sales dialog that tries to avoid the disadvantages mentioned by asking situation-dependently appropriate questions, flexibly reacting on decisions and also providing the customer with information needed. This dialog integrates the communication process in an ongoing search process. Furthermore, the structure of the product database is considered for question selection to focus on dialogs of short length, i.e. concerning the number of questions to be asked.

The ensuing section 2 will motivate the need for such a dialog definition by showing that, in general, a knowledge gap exists between vendors (i.e. in our case represented by the electronic sales system) and customers. This gap results from the different languages both parties speak, i.e. different ways to describe products or demands respectively. Section 3 introduces the model for user interaction in online sales scenarios and elicits the requirements for formal description of dialogs finally given in detail in section 4. Section 5 presents related work in the field of dialog processing and mediation and section 6 shows the ground for our future work.

2. The Knowledge Gap

Generally, for the specification of her demands, a customer would use vocabulary of a much higher level of abstraction than this is reflected by the predefined input forms that can be found in most of the electronic shops today. This situation happens because 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 [19, 20]. There is a knowledge deficit on both sides.

Seen from a methodological point of view, the phases of electronic sales share a relevant aspect as a characteristic: they are all knowledge-intensive [16]. Looking at EC scenarios from this knowledge perspective, it becomes clear that knowledge of different kind plays an important role, especially for

- *products*: properties, structure, areas of deployment, defects, prices, competitive products, competitors, availability, etc.
- *clients*: type, language, product expertise, demands, requirements, preferences, behavior, etc.
- *strategies*: information mediation, consulting, negotiation, sales, etc.

Typically, the vendors, distributors, or manufacturers possess the knowledge about their products. The customer possesses the knowledge about her individual needs. Neither the customer nor the vendor usually shares the other's knowledge. This is the kind of situation we call a "knowledge gap". During a sales transaction, the "knowledge gap" can be observed especially when it comes to the acquisition of customers' demands and finding adequate products for these demands. In brick-and-mortal companies, the human sales agent who makes use of her strategic knowledge to mediate between the potential buyer and the vendor bridges this knowledge gap. This mediation requires communication during which the knowledge is 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 respectively buyer system. In real sales situations, the knowledge gap exists, too. Nevertheless, it can be compensated or even be bridged by the better communication channels of human beings. In an online sales process, a virtual sales agent must realize this communication process.

The approach we are going to present does not base on the pretension that we want to simulate a real sales situation on a computer. In fact, we use the features, specialties, and possibilities of the media Internet to bridge the knowledge gap in an appropriate manner. One directive is always that the sales system should adapt to the customer and not vice versa.

3. Introduction to User Interaction

The ultimate goal of the electronic sales process, which we consider, is that the sales system finds a product for the customer such that her demands are fulfilled and she is willing to buy it, i.e. puts it into her electronic shopping basket. What we have to achieve therefore is communication and provision of information and knowledge between the vendor (the information provider) and the customer (the information user) in form of a dialog. Information and knowledge both are based on *experience*. This experience has to be managed and made accessible somehow. As a consequence, we insert the computer into the communication channel between vendor and customer, which causes two new communication channels:

- the communication of the vendor – seen from the view point above, now in the function of an experience provider – with the dialog system and
- the communication of the customer – now the experience user – with the dialog system.

This section addresses the second communication channel. This communication is of a bidirectional nature: The customer must be able to tell the dialog system about her demands and the dialog system must be able to communicate appropriate experience to this experience user.

We first present a more detailed view of the communication aspects in dialog management. Then, we discuss requirements concerning the user interaction and present the main issues to be considered during the design of the communication components.

3.1 The Basic Communication Data Flow

The communication between the customer and the dialog system involves both directions:

- *From the experience user (customer) to the dialog system*: During the *demand acquisition* phase, the problem must be captured and formalized and transferred to the experience management system as part of the sales system.
- *From the dialog system to the experience user*: During the experience presentation phase, the experience, or more precisely the lesson contained in the experience, must be presented in an appropriate form to the experience user.

3.2 Demand Acquisition

The task of the demand acquisition phase is to acquire sufficient information about the customer's demand in order to present to her one or several products that are useful to satisfy the demand. From a formal point of view, demand acquisition deals with specifying a problem $\mathsf{p} \in \mathsf{P}$, taking into account the representation language and vocabulary for the problem space P.

For this purpose, the demand (problem) acquisition software component must provide an appropriate user interface that shows input masks or asks specific questions. An important problem is to decide when to ask which question or when to show which input fields to the customer (cf. sections 4.5 and 4.6).

3.3 Requirements

The ultimate goal of user (customer) communication is to achieve efficient communication. Communication always involves a significant effort from the customer for entering information and for understanding the information presented to her. This effort should be kept as low as possible. Therefore, the following most important issues must be taken into account.

Small Number of Questions. During problem acquisition, the number of questions asked should be kept as small as possible. This relates to single questions raised directly, e.g., in a dialog window, and to questions that are answered implicitly by entering values into an input field of a questionnaire. Therefore, it is important to only ask questions that help selecting reusable experience. Asking irrelevant questions increases the communication effort without improving the accuracy of the results. A common problem with this requirement results from the fact that the relevance of a question often depends on many issues, such as the answers given to previous questions, the distribution of the products, the relevance (weight) of certain product properties, and so forth. Therefore, it is very difficult to find a static sequence of questions that avoids the problem of asking irrelevant questions. Dynamic questioning approaches can help to overcome this problem.

Comprehensible Questions. The questions asked must be understandable for the customer, i.e. they must be asked in a language and expressed in terms that she understands. Here, the problems can arise that different customers may have a different cultural or educational background, they can be domain experts or novices, etc. Depending on the spectrum of customers addressed by the sales approach it might be necessary to provide different questioning styles, multi-lingual access, and different modes for novices and experts.

Low Answering Cost of Questions. Answering a question always causes certain cost. Some questions cause low cost, i.e., they can be answered immediately while others may cause significant cost and may involve enacting certain examinations or investigations (e.g., in a medical diagnosis situation). When asking questions, the cost for answering them must be taken into account. Questions that cause low cost should be preferred over questions that cause high cost.

Comprehensible Question Clustering. Usually a series of questions is necessary to capture a demand. These questions are always clustered by some means: Questions can be clustered because they occur together on one page of a questionnaire or they can be clustered by the temporal sequence in which they are asked. Any clustering that occurs should be comprehensible for the customer. This means that the cluster should represent a concept known to the customer; sequences of questions should represent some „logical“ order.

3.4 Search Technology

The following main section will elicit that product search is an integral part for carrying on our communication process with the customer. Therefore, we want to build our formal ideas upon an adequate search technology. For support of structured content, *Case-Based Reasoning*, *CBR*, approaches turned out be best choice [22,21]. A number of commercial tools (by Acknosoft, Brightware, Inference, ServiceSoft, and tec:inno) implemented CBR methods and have proven success in many applications for product search in electronic shops. Like parametric search as available in tools from SAQQARA or Mercado, CBR assumes products to be described by a set of attribute values (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 wish. Thereby, it achieves much higher retrieval quality than competing approaches but at the price of investing into knowledge modeling.

4. A Formal Dialog Model

In this section, we introduce a general formal model for describing and relating the different dialog components and a dialog strategy. This model is a framework that allows to classify the detailed approaches presented in the subsequent sections. This model is inspired by dialog models that occur in diagnostics (see, e.g., [2]). Diagnostics can be considered as a problem acquisition task plus a classification task. Problem acquisition usually means to execute certain examinations that improve the information state of the object to be diagnosed. Examinations are selected according to whether they help to distinguish between several different hypotheses concerning the diagnosis.

4.1 Overview

A dialog can be modeled as a state machine. The state – we call it *dialog situation* – characterizes a particular instance in time of a particular dialog enacted between the user and the experience management system. It describes the current information state, i.e., all information gathered from the user up to this point in time, as well as all experience that is considered relevant for the problem as it stands at the moment (e.g., potential hypotheses). In a given dialog situation, a certain set of interactions (with the customer) can be initiated. We call these interactions *dialog interactions*. A dialog interaction can be asking a customer a particular question, presenting her a questionnaire, retrieving some products, presenting retrieved experience, etc. If a dialog interaction is executed, the current dialog situation is changed, which is of course the purpose of the interaction. For example, if a question is asked, the information state is extended by the new information gathered about a certain problem feature.

Usually, there are many different applicable dialog interactions in a particular dialog situation. In order to fulfill the requirements introduced above, a *dialog strategy* is necessary that determines for a given dialog situation one dialog interaction to be executed. Processing a dialog with respect to a given strategy means to start at an initially empty dialog situation and successively execute the dialog interactions proposed by the strategy. Thereby, the dialog state is successively transformed into a successor state until the strategy indicates the termination of the dialog process. This view is formalized in the subsections below.

4.2 Dialog Situations

The following definition formally introduces a dialog situation as a triple.

Definition: *Dialog Situation Space and Dialog Situation*

> The *dialog situation space* $S = P \times Q \times H$ consists of the problem space P, a question state space Q, and a hypothesis space H. A *dialog situation* $s = (p,q,h)$ is an instance of the dialog situation space, i.e., $s \in S$.

The problem (demand) description p contained in the dialog situation is the aggregation of all problem attribute values collected so far. Initially, this problem description is empty. During the dialog more problem attributes are filled and the degree of incompleteness of the problem description is reduced. The question state q usually describes the history of the dialog (e.g., as a finite state variable) as well as information obtained other than the problem descriptions. For example, it also contains information concerning the customer classification such as whether the customer is a novice or a domain expert. This can have an influence on the questions asked or the language chosen. The hypothesis h contains current hypotheses concerning the reusability of experience. It is usually some subset of the product base, i.e., $H = \wp(C)$. It contains, e.g., the products currently rated best (highest similarity) with respect to the current problem description. Additionally, the products may be annotated with their similarity or some similarity interval.

4.3 Dialog Interactions

Dialog interactions are certain actions started by the dialog component to update the dialog situation. The formal definition is as follows:

Definition: *Dialog Interaction Space and Dialog Interaction*

> The *dialog interaction space* $I = \{i_1,\ldots,i_k\}$ consists of a set of *dialog interactions*. A dialog interaction is a function i_v: $U \rightarrow (S \rightarrow S)$ that determines from a certain user input (out of the set of possible user inputs U) a transformation function of the current dialog situation s into a successor situation s', i.e., $i_v(u)(s) = s'$.

There are different kinds of dialog interactions.

User Questions. The primary reason for dialog interactions is to extend the information about the problem. A customer interaction can be to ask one particular question that aims at determining the attribute value of a certain problem attribute. Then, the answer given by the customer is the customer input $u \in U$ and the transformation function determines the successor situation by assigning u to the respective attribute. Instead of asking a single question, a customer interaction can also present a whole questionnaire to the customer in which she can enter several answer values that are then assigned to the appropriate attributes.

Checking Consistency. Dialog interactions can also be used to model consistency checks for the information entered by the customer. If inconsistencies are determined the customer can be informed and she can be asked to correct mistakes. Such corrections lead to a change in the problem description.

Updating Hypotheses. Another type of dialog interaction is concerned with updating the hypothesis based on the current problem description. This interaction does not usually involve the customer but its execution triggers the retrieval and (possibly the adaptation) steps at the server side of the experience management system. The retrieval result is used to update the current hypotheses.

Presenting Experience. Finally, dialog interactions can also be used to initiate the experience presentation. When such an interaction is executed, the retrieved experience is presented to the customer. Additionally, feedback can be obtained about whether the presented experience is reusable or not. The problem description and the hypothesis are not updated by this interaction.

Besides the primary modifications of the dialog situations mentioned above (none for presentation interactions), they usually also update the question state. This is necessary in order to take care of the dialog history, e.g., to avoid asking questions twice. The way they are updated depends on the questioning strategy discussed below.

4.4 Dialog Strategy and Its Execution

The dialog strategy controls the whole dialog. It determines which interactions to be executed.

Definition: *Dialog Strategy*

> A *dialog strategy* is a function $\mathsf{strat}: \mathsf{S} \to \mathsf{I}$ that determines for a given current dialog situation the next dialog interaction to be executed.

The dialog strategy is the core of the dialog component. It is responsible for an efficient dialog and for the fulfillment of the requirements discussed in section 3.3. The following simple algorithm (Fig. 1) describes the top-level loop in which the dialog strategy is executed. It assumes a particular interaction that indicates the termination of the dialog.

In the following we discuss several approaches for user communication in relation to the just described formal framework. These approaches differ significantly in the dialog situation representation, the dialog interaction, and the dialog strategy that is used.

```
procedure DialogControl()
var
    sit: S
    interaction: I
    user_input: U
begin
    sit := (nil,nil,nil)
    repeat
        interaction := strat(sit) (* Determine next interaction *)
        Perform interaction and determine user_input if necessary
        sit := interaction(user_input)(sit) (* Determine successor situation *)
    until interaction = terminate
end
```

Figure 1: *Dialog Control Algorithm in Pseudo Code.*

5. Dialog Categories

5.1 Predefined Static Dialogs

The first category of approaches for user interaction is based on a predefined static dialog. The dialog interactions and strategy are manually modeled in advance by the developer of the experience management system behind the e-shop. We can distinguish different modeling approaches depending on the degree of flexibility provided by the modeling.

5.1.1 Three-Step Questionnaire-Based Problem Acquisition

The three-step questionnaire-based approach is the most simple and therefore also the most common one in many applications. The dialog strategy is a sequence of three predefined dialog interactions:

1. Present a questionnaire to the customer and obtain results.
2. Retrieve and adapt products.
3. Present the information found.

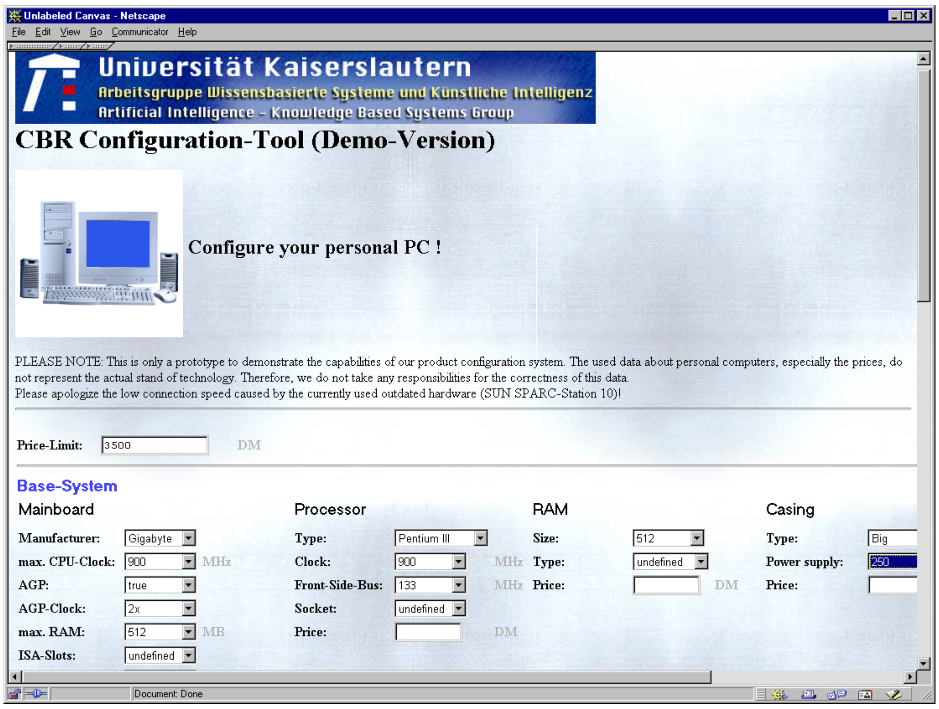
The questionnaire shows all problem attributes at a time. For each attribute, there is an appropriate way for entering a value of the respective value type of the attribute.

- Numeric attribute values are either entered by
    - an input field into which a number must be typed in, or
    - a slider that must be positioned to the right number.
- Symbolic attribute values can either be entered by

  - menus showing all possible values,
  - a set of radio buttons each of which is associated with a symbolic value, or
  - an input field into which the symbol value must be typed in.
- Textual attribute values are entered by an input field.

For hierarchical (object-oriented) representations, the questionnaire should be structured according to the object structure. However, this is only possible if a fixed object structure can be assumed. If the problem representation requires flexibility that allows to have arbitrary objects as instances in a relational slot, a dynamic questionnaire is required. Depending on the object class that is selected different input fields for the related attributes must be shown.

Figure 2 shows an example of a comprehensive questionnaire for obtaining a requirements specification for Personal Computers (PC) to be used for product experience selection in an e-commerce scenario [17]. This questionnaire is developed particularly for an expert customer who is aware of the meaning of the different attributes. For such a customer, these attribute descriptions are easily understandable and filling the appropriate fields is an effective way of communication. No further guidance is necessary and a long-lasting dialog would be disturbing for an expert customer who knows what she wants.


Unlabeled Canvas - Netscape
File Edit View Go Communicator Help
Universität Kaiserslautern
Arbeitsgruppe Wissensbasierte Systeme und Künstliche Intelligenz
Artificial Intelligence - Knowledge Based Systems Group
Document: Done

CBR Configuration-Tool (Demo-Version)

Configure your personal PC !

PLEASE NOTE: This is only a prototype to demonstrate the capabilities of our product configuration system. The used data about personal computers, especially the prices, do not represent the actual stand of technology. Therefore, we do not take any responsibilities for the correctness of this data.
Please apologize the low connection speed caused by the currently used outdated hardware (SUN SPARC-Station 10)!

Price-Limit: 3500 DM

Base-System

Mainboard			Processor			RAM			Casing	
Manufacturer:	Gigabyte		Type:	Pentium III		Size:	512		Type:	Big
max. CPU-Clock:	900	MHz	Clock:	900	MHz	Type:	undefined		Power supply:	250
AGP:	true		Front-Side-Bus:	133	MHz	Price:		DM	Price:	
AGP-Clock:	2x		Socket:	undefined						
max. RAM:	512	MB	Price:		DM					
ISA-Slots:	undefined									

Figure 2: *Query input form for PC configuration [17].*

5.1.2 Static Domain Specific Dialogs

The three-step questionnaire-based approach is a fixed dialog model used independently of the application domain. Only the questionnaire itself and the form of the experience presentation is domain specific. *Static domain specific dialogs* are modeled specifically for the domain at hand. They are static in the sense that they are modeled in advance by the developer of the experience management system and are not changed as a consequence of the available experience or experience with customer interactions (e.g., [14]).

Dialog Situations. The dialog situation is restricted to only contain the problem description and the question state. The problem description stores the currently acquired information about the problem. The question state is usually one or several state variables.

Dialog Interactions. Dialog interactions that represent user questions, consistency checks, and experience presentations are modeled particularly for the application domain. User questions can involve asking an individual question or presenting a small questionnaire asking for related problem attributes. Typically, such a questionnaire does not ask for all relevant problem attributes such as the example shown in Fig. 2 does.

Modeling the Dialog Strategy with a Directed Graph. The modeling of the dialog strategy is the most crucial part of this approach. A simple but often appropriate method to model the strategy is to use a directed graph. The nodes of the graph are labeled with dialog interactions. When they are executed, a successor dialog situation results. The edges in the directed graph describe transitions from one dialog interaction to the next one. They are labeled with conditions on the successor dialog situation that results from the execution of the node from which the edge starts.

<u>Definition</u>: Dialog Strategy Graph

> For a given dialog interaction space I and a dialog situation space S, a dialog strategy graph is a directed labeled graph (N,E) in which each node $n \in N$ is labeled with a dialog interaction $i_n \in I$ and each edge $e \in E$ is labeled with a condition $c_e \subseteq S$. The conditions of outgoing edges from a node are disjoint, i.e.,
>
> $\forall e_1,e_2 \in E: e_1 = (n,n_1) \wedge e_2 = (n,n_2) \wedge n_1 \neq n_2 \rightarrow c_{e_1} \cap c_{e_2} = \emptyset$.

The dialog strategy graph contains one distinguished start node n_0.

The dialog strategy graph completely defines the dialog strategy. Its execution starts with the empty dialog situation[1] at the start node n_0. The dialog interaction indicated at a node (initially the start node) is executed and thereby a successor dialog situation is obtained. Then the conditions at the outgoing edges of the node are checked and the dialog proceeds with the node that can be reached from the edge

[1] The current node can be considered part of the question state.

whose condition is fulfilled. If no outgoing edges with fulfilled conditions are available, the question strategy terminates.

Examples of user communication components that are based on this approach are discussed in [15,3,18]. The responsibility for achieving an efficient dialog with respect to the requirements discussed in section 3.3 remains at the developer of the experience management system who defines the dialog strategy graph. A simple sequence of interactions is usually not sufficient since in most cases the relevance of a question depends on the answers to previous questions. The conditions at the edges of the graph express when a question is relevant.

Modeling the Dialog Strategy with Rules. An alternative way of modeling a dialog strategy is to use rules. Richter and Schmitt [13] suggest the use of so-called event-condition-action rules. These rules have the following form:

```
IF <Event> AND <Condition> THEN <Action>
```

The event part of the rule relates to an activity issued by the user during a dialog interaction. In the formal dialog model, such activities are encoded in the dialog state. The condition part of the rule is a condition over the current problem description. The action of the rule proposes the dialog interaction to be enacted.

Again, the responsibility for defining a rule base that models an efficient user communication is at the developers side. Like the conditions in the dialog strategy graph, the preconditions of the rules must not overlap, i.e., for every dialog situation only one rule should fire. If this property is not requirement a conflict resolution mechanism is required, i.e., by adding a priority value to the rules.

5.2 Dynamic and Adaptable Strategies

The previously discussed dialog strategies must be modeled by hand, which requires a significant knowledge acquisition effort. Changes in the product base will also require updating the dialog strategy. To overcome this problem, several approaches have been developed to realize dialog strategies implicitly. The basic idea behind this approach is to analyze the distribution of products in the product base to select questions according to their relevance for deciding the usability of a product. We distinguish different strategies

- according to the selection criterion used and
- according to whether the strategy is compiled into a dialog strategy graph or
- whether the selection criterion is interpreted dynamically when the dialog is enacted.

5.2.1 Criteria for Attribute Selection

Problem attributes to be asked to the user should be selected according to whether they contribute information that is relevant to decide among the usability of the products in the product base. The amount of dialog will be reduced to a minimum if irrelevant attributes are not asked. Several attribute selection criteria have been proposed in the machine learning and the case-based reasoning literature. Although, most of the approaches from machine learning do not aim at reducing the amount of dialog but at minimizing the size of generalized concept descriptions, the results can be transferred to a certain degree.

Information Gain Measure for Classified Products. The information gain measure has its origin in information theory [7] and was used in machine learning for attribute selection during the construction of decision trees [10,11]. It is based on the concept of *information content* of a message and the *entropy* of a set of items. The information conveyed by a message depends on its probability and can be measured in bits as minus the logarithm to base 2 of that probability. The higher the probability the lower is the information conveyed. If we have a set of k possible messages each of which occurs with the probability pr_i the event has an *expected information content*:

$$H(pr_1,\dots,pr_k) = -\sum_{i=1}^{k} pr_i \cdot \log_2(pr_i) \qquad \text{(Expected Information Content (Entropy))}$$

The information gain measure is traditionally used for classified products. Classified products represent examples of a classification task. The product lesson space is defined as a small set of possible classes. The number of classes is small with respect to the number of products available, i.e., there are usually several products that belong to the same class. If we have some set of products, with respect to our CBR approach (cf. section 3.4) called cases, $C = C_1 \cup \dots \cup C_k$ such that C_i contains only cases of class i, then the expected information content of C is:

$$H(C) = -\sum_{i=1}^{k} \frac{|C_i|}{|C|} \cdot \log_2\left(\frac{|C_i|}{|C|}\right) \qquad \text{(Expected Information Content (Entropy) of C)}$$

If C only contains cases of the same class, the expected information content (or the entropy) of C is 0 because we do not need any information to predict the class. If we partition the set of cases C according to an attribute A into m subsets $C = C^1 \cup \dots \cup C^m$ such that the attribute of all cases in C^j is v_j we can investigate the expected information content of each subset C^j. The expected conditional information content is

$$H(C \mid A) = \sum_{j=1}^{m} \frac{|C^j|}{|C|} \cdot H(C^j) \qquad \text{(Expected Conditional Information Content of C|A)}$$

This value is the expected information content required to determine the class of a case after we know the value of the attribute A. Given a set of cases C, we can now define the *information gain* obtained by knowing the value of the attribute A through:

$$Gain(A) = H(C) - H(C|A) \qquad \text{(Information Gain for Attribute A)}$$

In the traditional ID3 induction algorithm for decision trees [10] the next attribute used for partitioning the current set of cases is the one with the highest information gain. While this criterion is originally only defined for symbolic attributes, extensions to numeric attributes are discussed in connection with the C4.5 algorithm [11,12]; the continuous value range is partitioned into a finite set of intervals on which the standard definition is applied.

This information gain measure for attributes has also been proposed as a strategy for question selection in case-based reasoning [5]. However, it has several disadvantages. First, it requires classified cases which are not always available; in the next section we discuss how to overcome this limitation. Second, it does not take into account the similarity measure. But the similarity measure also influences the relevance of an attribute. For example, an attribute that does not occur in the similarity measure must not be asked. Third, this approach does not take the answering cost of an attribute into account, which can be different for different attributes. Finally, it is not guaranteed that the clustering of questions, i.e. the sequence in which they are asked, is comprehensible to the customer.

Information Gain Measure for Unclassified Products. In order to overcome the limitation that products must be classified, Doyle and Cunningham [4] propose the use of clustering algorithms such as the *k-medoid algorithm* [8] to derive a classification of originally unclassified products. This algorithm selects k representative products, called medoids, and clusters the other products according to their similarity to the medoids. The medoids are selected such that the average similarity between the medoids and the products belonging to a cluster is high and the similarity to the products not belonging to the cluster is low. This clustering yields products that are classified according to a similarity measure. The information gain measure discussed before is then used for attribute selection on the now classified products.

Instead of clustering the products first, one can alternatively use the product identifier as a class label. This means that every product defines its own class. This is the same as turning each product of the product base into a medoid in the clustering algorithm. Given this approach, the entropy of a set of products (cases) C is simplified to

$$H(C) = \log_2(|C|)$$

and the information gain for an attribute A is simplified to

$$Gain(A) = -\sum_{i=1}^{m} \frac{|C^j|}{|C|} \cdot \log_2\left(\frac{C^j}{C}\right)$$

The clustering approach strongly depends on the number of clusters used, because questions are selected such that the problem can be assigned to a cluster. No questions are generated to differentiate the products within a cluster. On the other hand, if every product is turned into an individual cluster the relevance of the attributes that remain with respect to the similarity measure is ignored. Also, the answering costs for an attribute are not taken into account and the sequence of questions generated may not be comprehensible to the customer.

Similarity Influence Measure. A different approach that is better tailored to deal with unclassified products (cases) is to select attributes not on the basis of their information gain but on the basis of the influence of a known value on the similarity of a given set of cases. In an online shop it is desirable to present the customer a selection of products most similar to her query. It is therefore a reasonable strategy to first ask the attributes that have the highest influence on the similarity of the products stored in the product database.

We suggest a way to measure the influence on similarities by calculating the variance Var of similarities a query (problem) q induces on the set of candidate products C [9]:

$$Var(q,C) = \frac{1}{|C|} \cdot \sum_{c \in C} \left(sim(q,c) - \mu \right)^2 \qquad \text{(Variance of Similarities for a Query)}$$

Here, $sim(q,c)$ denotes the similarity of the query q and the product c, μ denotes the median of all similarities.

When asking a question the assigned value is not known in advance. It is therefore necessary to select the attribute only on the expected similarity influence $simVar$, which depends on the probability p_v that the value v is chosen for the attribute A:

$$simVar(q,A,C) = \sum_{v} p_v \cdot Var(q_{A \leftarrow v}, C) \qquad \text{(Expected Similarity Influence of an Attribute)}$$

$Var(q_{A \leftarrow v}, C)$ defines the similarity influence of assigning a value v to an attribute A of the query q. To simplify the computation of $simVar(q,A,C)$ it is possible to consider only the attribute values v that occur in the product set C. Then, the probability p_v for the value v can be calculated from the sample of products in C, i.e. $p_v = |C^v| / |C|$. (Here, it has to be remarked that at present the calculation of p_v only follows a heuristic. The distribution of values in the product database is certainly not the same like for the real customer buying behavior for products. However, without loss of generality, this function can easily be exchanged.)

In a dialog situation, the attribute with the highest expected similarity influence on the set of candidate products is selected. This strategy leads to the highest increase of knowledge about similarity thereby faster discriminating the product database in similar and dissimilar products.

Two common disadvantages of all previously discussed question selection criteria also hold for the similarity influence measure. First, they do not consider the answering cost of questions and they may lead to an incomprehensible question ordering. These issues are briefly discussed now.

5.2.2 Integrating Answering Cost

It often occurs that some questions are easier to answer than other questions. Questions may be difficult to answer because they require to make an expensive, time consuming, or disagreeable examination. We summarize these issues by introducing the cost of answering a question.

Definition:*Question Answering Cost*

> Let A be a problem attribute. The *cost of answering* the question concerning the value of the attribute A in the current situation is denoted by qc(A) ∈ [0,1[. Cost values are expressed as real values from the interval [0,1[where a higher value indicates higher cost.

Answering costs are not considered in any of the previously discussed question selection criteria. However, if the answering costs are known they can be easily integrated in any of the previously introduced criteria as follows (the criterion select_without_cost(A) can be, e.g., Gain(A) or simVar(p,A,C) [9]) :

$$select\_with\_cost(A) = \frac{select\_without\_cost(A)}{1 - qc(A)}$$

(Cost Sensitive Attribute Selection)

However, the problem of determining the attribute cost remains. They can also vary from customer to customer or from situation to situation. Hence, adaptive approaches are desirable that automatically determine attribute costs from customer behavior.

5.2.3 The Problem of Comprehensible Question Clustering

The second problem ignored by any of the known question answering approaches is that they can cause incomprehensible question orderings. This is due to the fact that relationships between different questions are not considered in any of the criteria. We can express the relatedness of two questions by using a similarity measure as follows:

Definition: *Question Similarity*

> Let A_1 and A_2 be two problem attributes. The *similarity of the two questions* asking for the values for A_1 and A_2 is denoted by $sim(A_1,A_2) \in [0,1]$. The higher the similarity value the more are the two questions related.

Such relationships among attributes are already modeled in an experience management system if object-oriented modeling is used. The class hierarchy can be regarded also as a hierarchy for questions. Hence, it should be avoided to switch too often from an attribute of one class to an attribute of a class at a distant location in the class hierarchy. By interpreting the class hierarchy as a taxonomy, we can measure the similarity of attributes (or the similarity or relatedness of questions) by using the inter-class similarity, i.e. $sim(A_i,A_j)=sim_{inter}(C_i,C_j)$ where C_i is the object class in which the attribute A_i is defined.

The attribute similarity can then be integrated into the attribute selection criterion as a correction term as follows:

$$select\_with\_pref(A) = select\_without\_pref(A) \cdot (1-\alpha+\alpha \cdot sim(A, A_{pref}))$$

(Similar Attribute Preference)

Here, A_{prev} denotes the attribute that has been asked in the previous question. The parameter $\alpha \in [0,1]$ specifies the influence of the attribute similarity on the selection criterion. The higher the parameter α is, the higher is the influence of the attribute similarity. Again, this correction can be used with any of the previously discussed attribute selection criteria.

6. Related Work

Recent commercial surveys have shown that product search in electronic shops leaves customers frustrated in a huge number of cases. Experiments in leading e-shops turned out that 92% of customers' searches produced bad results [6]. This is due to the fact that all current search approaches are oriented towards the vendor's language and not towards the different languages of the customers. This is not a big problem when the customer knows the available products and the language in which the vendors speak about their products. However, if the customer needs real guidance this approach is inappropriate. Recent standardization activities, such as the German eCl@ss[2] initiative, which tries to normalize product descriptions for e-procurement, try to bridge the knowledge gap by telling the customer the vendor's language. Here again, the customer is supposed to adapt to the vendor; the problem is not solved but just handed over to the customer. This approach is also inappropriate for complex products or dynamic areas since the standardized product language must either be adapted and communicated to the customer frequently or it remains too abstract to enable high quality product recommendation.

Representing the theoretical background, the basis for our considerations is definitely provided by the work of Quinlan [10,11,12] who introduced the information gain measure for attribute selection during the construction of decision trees. Doyle and Cunningham [4] have adopted this as a basis for their incremental clustering approach. Their approach lacks of a couple of drawbacks compared to ours. As already mentioned in section 4.6, do they require classified products, which are not always available. Furthermore, the approach does not take into account the similarity measure. But the similarity measure also influences the relevance of an attribute (e.g., an attribute that does not occur in the similarity measure must not be asked). Last but not least, this approach does not take the answering cost of an attribute into account, which can be different for different

[2] http://www.eclass.de/

attributes. Finally, it is not guaranteed that the sequence in which questions are asked is comprehensible to the customer.

The SeTA[3] (Servizi Telematici Adattativi - User Adaptive Web-Based Systems) project is a prototype system that can be used to define adaptive Web stores, focusing on the individualization of the interaction with the customers, i.e. on tailoring the interaction with customers to their preferences and needs [1]. SeTA is based on a multi-agent architecture, where several specialized agents co-operate to the management of a dynamic, virtual store that personalizes the interaction with the customer depending on her interests, needs and domain expertise.

An example for the implementation of an entropy-based approach is the Carsmart[4] demo application working on the domain of used car sales. The system issues a short sequence of questions. The first question asks about the customer's most important attribute, e.g., make, body, or price. Depending on the answer, the respective attribute is asked in the second question (let us assume the make of the car was chosen) together with the problem attribute that contains the model of the car. The third question asks now for more detailed information about the importance of other problem attributes. Finally, the two most important and yet unknown attributes (e.g., price and mileage) are asked.

7. Conclusions & Future Work

The dialog process is working at a stage that prepares a query to the product database. The more information was gained from the customer the better will be the search results. The fact that the lack of current search technologies is their inability to bridge the knowledge gap made us develop the presented formal framework for a dynamic dialog approach. Of course, the idea of carrying on a dialog with a customer is nothing new. Our innovative part is the interweaving of customer communication directly with the search process. CBR as a search technology fully supports this idea.

The next steps of this research have already been started: the practical deployment of our approach. Unfortunately, by time of this paper written, there are no results ready for presentation. We estimate to have a demo prototype version of our system by automn 2001. We can already build upon a system called READEE[5] that realizes a mediation level to bridge the knowledge gap. It is an application in the B2B field dealing with the domain of reuse of complex electronic designs. This system does not yet work on basis of a dynamic dialog adapting to answers given by the customer like we suggested in this paper. The dialog will cause an action depending on its current state (situation).

[3] http://www.di.unito.it/~seta/

[4] http://live.tecinno.de/projects/carsmart24com/

[5] http://wwwagr.informatik.uni-kl.de/~readee/

Another important aspect to investigate will be the extension of our approach by machine learning methods in the dialog. We want to use statistical methods to determine especially the costs for a question. It might happen that based on the information gain measure a certain question will be asked, but the customer will not be able to answer it. Then, there would not be any information gain and the costs of asking this question are very high. We will try to predict these costs and to adapt them by time.

References

[1] L. Ardissono, A. Goy. Tailoring the InteractionWith Users in Electronic Shops. In: *Proc. of the 7th Conference on User Modeling, UM'99,* Banff, Canada. Springer, 1999.

[2] P. Cunningham, B. Smyth. A Comparison of model-based and incremental case-based approaches to electronic fault diagnosis. In: *Proc. of the Case-Based Reasoning Workshop,* AAAI-1994.

[3] P. Cunningham, R. Bergmann, S. Schmitt, R. Traphoener, B. Smyth, P. Mac AnUltaigh. WEBSELL: Intelligent Sales Assistants for the World Wide Web. In: *KI – Zeitschrift für Künstliche Intelligenz,* Issue 1, 2001.

[4] M. Doyle, P. Cunningham. A Dynamic Approach to Reducing Dialog in On-Line Decision Guides. In: *E. Blanzieri, L. Portinale (Eds.): Advance in Case-Based Reasoning, Proc. of the 5th European Workshop, EWCBR-2000.* Springer LNCS, LNAI 1898, pp. 49-60, 2000.

[5] M. H. Göker, C. A. Thompson. The Adaptive Place Advisor: A Conversational Recommendation System. In: *M. H. Göker (Ed.): Proc. of the 8th German Workshop on Case-Based Reasoning, GWCBR-2000,* DaimerChrysler Research, Ulm. 2000.

[6] P. Hagen. Guided Search For eCommerce. The Forrester Report. January 1999.

[7] E. B. Hunt, J. Martin, P. J. Stone. Experiments in Induction. Academic Press. 1966.

[8] L. Kaufman, P. J. Rousseuw. Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley & Sons, NY, US, 1990.

[9] A. Kohlmaier, S. Schmitt, R. Bergmann. A Similarity-based Approach to Attribute Selection in User-Adaptive Sales Dialogs. To appear in: *D.W. Aha, I. Watson, Q. Yang (Eds.): Case-Based Reasoning Research and Development. Proc. of the 4th International Conference on Case-Based Reasoning, ICCBR'01,* Vancouver, Canada. Springer LNAI, 2001.

[10] J. R. Quinlan. Induction of decision trees. Machine Learning, 1(1), pp. 81-106. 1986.

[11] J. R. Quinlan. C4.5 Programs for Machine Learning. Morgan Kaufmann. 1993.

[12] J. R. Quinlan. Improved Use of Continuous Attributes in C4.5. Journal of Artificial Intelligence Research, 4, pp. 77-90. 1996.

[13] M. M. Richter, S. Schmitt. Kundenmodellierung und Dialogführung: Eine Herausforderung für eCRM. In: *A. Eggert, G. Fassott (Eds.): eCRM – Management der Kundenbeziehungen im Internet-Zeitalter”.* Schäffer-Poeschel Verlag, 2001.

[14] M. Rosewitz, U. J. Timm. Editor für elektronische Produktberatung. Wirtschaftsinformatik, 40(1), pp. 21-28. 1998.

[15] S. Schmitt, D. Jerusalem, T. Nett. An Acquisition Framework to Implement Questioning Strategies for Query Building for CBR Systems. In: *M. H. Göker (Ed.): Proc. of the 8th German Workshop on Case-Based Reasoning, GWCBR-2000*, DaimerChrysler Research, Ulm. 2000.

[16] S. Schmitt, B. Schneider: Einsatzpotentiale der KI im Electronic Commerce. KI - Zeitschrift für Künstliche Intelligenz, Special Issue: Electronic Commerce, Issue 1/01, 2001.

[17] A. Stahl, R. Bergmann, S. Schmitt. 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*, Bled, Slovenia, June 19-21, 2000. Volume 1: Research.

[18] U. J. Timm, M. Rosewitz. Electronic Sales Assistance for Product Configuration. In: *G. J. Doukidis, J. Gricar, J. Novak (Eds.): Electronic Commerce in the Information Society. Proc. of the 11th International Bled Electronic Commerce Conference*, Bled, Slovenia, June 8-10, 1998. Volume 1: Research. 1998.

[19] I. Vollrath. 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, EWCBR-98,* Dublin, Ireland. Springer LNCS; Vol. 1488: LNAI, pp. 136-147, 1998.

[20] W. Wilke. Knowledge Management for Intelligent Sales Support in Electronic Commerce. DISKI 213, Infix Verlag, 1999.

[21] W. Wilke, R. Bergmann, S. Wess. Negotiation During Intelligent Sales Support with Case-Based Reasoning. In: *Proc. of the 6th German Workshop on Case-Based Reasoning, GWCBR'98*, Germany, 1998.

[22] W. Wilke, M. Lenz, S. Wess. Case-Based Reasoning for Electronic Commerce. In: *Lenz et al. (Eds.): Case-Based Reasoning Technology from Foundations to Applications*, Springer, 1998.

[23] S. Schulz. CBR-Works: A State-of-the-Art Shell for Case-Based Applications. In: *E. Melis, (Ed.): Proc. of the 7th German Workshop on Case-Based Reasoning, GWCBR'99,* Würzburg, Germany. 1999.